Bull. Nov. Comp. Center, Comp. Science, 39 (2016), 1–15

Simulation performance versus stochasticity in large-scale cellular automata models

Olga Bandman

Abstract. Due to a growing interest in chemical and biological phenomena, simulation of reaction-diffusion processes on micro-level becomes urgently wanted. Asynchronous cellular automata are promising mathematical models to be used as a base for creating computer simulation systems, which gives reason for the investigation of their capability. In particular, since the micro-level simulation requires a very large size of cellular automata, the performance of simulation is important, especially, because parallel implementation is inevitable. In this connection, the dependence of simulation performance on the CA stochasticity (the degree of randomness) is studied in this paper. Much attention is being given to the contradiction between the stochasticity and the parallelization efficiency. It is shown how the proper choice of the CA stochasticity may help to achieve the acceptable performance. The results of a 3D process of diffusion limited aggregation are presented to assert the dependencies obtained.

1. Introduction

Due to availability of huge computational power, the simulation becomes an essential part both in scientific investigations and in the new technologies design. Meanwhile, conventional mathematical models, based on the differential calculus, are sometimes not capable of simulating nonlinear, dissipative processes on micro- or nano-level of resolution, which is requested in many chemical, biological and microelectronic investigations. A large class of such tasks deals with "reaction–diffusion" processes and, accordingly, they are represented by "reaction–diffusion" (RD) models. In continuous mathematics such models are given by parabolic partial differential equations, usually having nonlinear reaction terms. Numerical solution of such a kind of equations requires some transformations to be done for obtaining a lin-

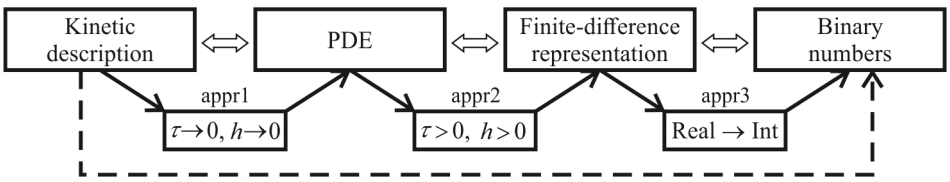

Figure 1. Comparison of approximations in PDE numerical solutions process and in cellular automata evolution computation

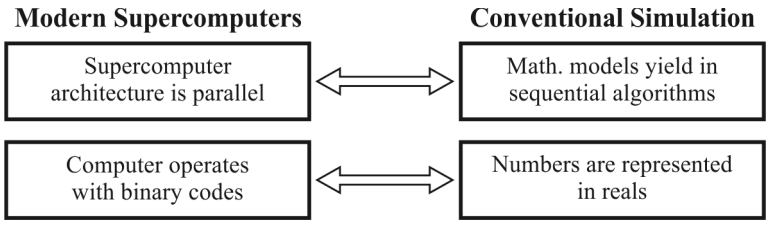

Figure 2. Contradictions between conventional mathematics and supercomputer architecture properties

ear approximation, and overcoming difficulties when carrying out parallel implementation (Figure 1).

Problems arise from the incompatibility between continuous mathematical models intended for a sequential solution, on the one hand, and discrete data and parallel operations on modern computers, on the other hand (Figure 2). These contradictions stimulate the development of new approaches to spatial dynamics modeling [1], which are based on the following principles:

- time, space, and variables should be integers (including Boolean), real numbers allowed only for computing auxiliary values (probabilities, conditions);
- all time-independent operations should be explicitly expressed in the model;
- a computer-simulated algorithm should allow for visualization of the resulting spatial function in real time.

Among the existing computer simulation methodologies of reaction–diffusion phenomena completely or partly based on the above statements, the following methods are most known: kinetic Monte Carlo methods in chemistry [2, 3] and in microelectronics [4], probabilistic cellular automata in material science [5–7], self-organizing CA in biology [8, 9]. Actually, all of them may be considered as asynchronous cellular automata [10] simulating phenomena which are composed of movements and transformations [11] of a certain kind of micro-entities, such as molecules or nano particles either real or abstract. Accordingly, a model should consist of a number of simple local operators being applied to randomly chosen sites of discrete space. Organization of particles interactions in space and time (the operation model) is not necessarily completely asynchronous. Particularly, it may be characterized by a balance between deterministic and random parts of computation [12]. Since the simulation is usually aimed at studying an unknown mechanism of substance behavior, it may be supposed that the introduction of a certain determinism may be admissible, because it reduces the computation time both in sequential and in parallel implementation. This paper is aimed at

clarifying how the ratio of deterministic to stochastic operations influence the performance of the reaction-diffusion computer simulation, especially, in the case of parallel implementation.

This paper is organized as follows. The next section deals with formal definitions and formal problem statements. In the third section, the notion of stochasticity is introduced, and the impact of CA stochasticity on computation time is studied. The fourth section concerns to the dependence of parallel implementation efficiency on the CA stochasticity. The simulation results of parallel implementation of 3D diffusion limited aggregation are presented to give an impression of real efficiency values.

2. Cellular automata formal representation

2.1. The main definitions. A classical CA [13] is usually defined by a triple

$$\aleph = \langle A, X, \theta(\boldsymbol{x}) \rangle, \tag{1}$$

where $A = \{0, 1\}$ is a Boolean *state alphabet*, $X = \{\boldsymbol{x}_i : \ i = 0, 1, \ldots\}$ is a *set of cell names*, a cell being a pair $(a, \boldsymbol{x})$, $a \in A$, $\boldsymbol{x} \in X$, and a set of cells $\Omega = \{(a_i, \boldsymbol{x}_i) : \ i = 0, 1, \ldots, |X|\}$ being called a *cellular array*; $\theta(\boldsymbol{x})$ is a transition rule, usually given as a Boolean function of the state neighborhood of a cell named $\boldsymbol{x} \in X$. A transition rule $\theta(\boldsymbol{x})$ is represented in the form of the substitution

$$\theta(\boldsymbol{x}) : S(\boldsymbol{x}) \xrightarrow{\text{cond}} S'(\boldsymbol{x}), \tag{2}$$

where $S(\boldsymbol{x}) = (u_0, \ldots, u_n)$ and $S'(\boldsymbol{x}) = (v_0, \ldots, v_m)$, $u, v \in A$, $m \leq n$, are *local configurations* expressed as vectors of cell states in the vicinity of $\boldsymbol{x}$ defined by its *underlying template* $T(\boldsymbol{x})$, "cond" is an external condition of the substitution applicability,

$$T(\boldsymbol{x}) = \{\boldsymbol{x}, \boldsymbol{x} + d_1, \ldots, \boldsymbol{x} + d_{n-1}\}, \tag{3}$$

where d_j is a shift distance in the naming space, such that the cell $(\boldsymbol{x} + d_j)$ has the state u_j. A maximum value of all d_j, $j = 1, \ldots, n-1$, is referred to as *template radius* $R(T)$. The underlying templates $T(\boldsymbol{x})$ and $T'(\boldsymbol{x})$ of the local configurations $S(\boldsymbol{x})$ and $S'(\boldsymbol{x})$, respectively, are in the following ratio:

$$T'(\boldsymbol{x}) \subseteq T(\boldsymbol{x}). \tag{4}$$

Substitution (2) is applicable to a cell $\boldsymbol{x}$ if the following condition is satisfied:

$$(S(\boldsymbol{x}) \subset \Omega(t)) \quad \& \quad (\text{cond} = \text{true}). \tag{5}$$

Application of $\theta(\boldsymbol{x})$ to $\boldsymbol{x} \in X$ yields in substituting the first m components $(u_0, \ldots, u_{m-1})$ of $S(\boldsymbol{x})$ in (2) for the corresponding state values $(u'_0, \ldots, u'_{m-1})$ of $S'(\boldsymbol{x})$, $m = |T'(\boldsymbol{x})|$, u'_j being values of the *transition functions*

$$u'_j = f_j(u_0, \ldots, u_n). \tag{6}$$

Transition functions may be of any type: constant, arithmetic, or threshold. The only requirement is that the value should remain in A.

Given an initial cellular array $\Omega(0) = \{(a_i, \boldsymbol{x}_i) : \ a_i \in A, \ \boldsymbol{x}_i \in X\}$, the CA starts to evolute. The evolution

$$\Sigma = \Omega(0), \ldots, \Omega(t), \Omega(t+1), \ldots$$

is an iterative process, where each iteration is a transition from $\Omega(t)$ to $\Omega(t+1)$ resulting from application $\theta(\boldsymbol{x})$ to all $\boldsymbol{x}_i \in X$, which is defined as a global operator $\theta(X)$.

For providing functioning correctness, $\theta(X)$ should satisfy the following condition: no pair of substitution application should aim at adjusting the same cell at the same time, i.e.,

$$T'(\boldsymbol{x}) \cap T'(\boldsymbol{y}) = \emptyset \quad \forall \boldsymbol{x}, \boldsymbol{y} \in X, \ \boldsymbol{x} \neq \boldsymbol{y}, \tag{7}$$

where $T'(\boldsymbol{x})$ and $T'(\boldsymbol{y})$ are underlying templates for the right-hand sides of (2) in $\theta(\boldsymbol{x})$ and $\theta(\boldsymbol{y})$.

2.2. Modes of operation. The process of global operator execution transferring $\Omega(t)$ into $\Omega(t+1)$ may be organized in different ways called *modes of operation* and labeled further by the index ρ. The following modes are mainly used in the CA simulation.

Synchronous mode $\boldsymbol{\sigma}$ prescribes the following actions to be done to execute $\theta(X)$:

1. The next states u' of all $\boldsymbol{x} \in X$ are computed in any order according to (6) and saved in an additional array $\Omega'(t)$;
2. All cells at once are adjusted by replacing $\Omega'(t)$ by $\Omega(t+1)$.

Classical CA are synchronous. The reason is that synchronous computations are well organized and fit the synchronous way of computer operation. In synchronous CA, the correctness condition (7) is satisfied, when each substitution application adjusts a single cell, i.e.

$$|T'_j(\boldsymbol{x})| \leq 1, \tag{8}$$

where T'_j is the underlying template of $S'(\boldsymbol{x})$ of $\theta(\boldsymbol{x})$ in (2). This yields a strong constraint in the CA synthesis process, but simplifies parallel implementation of a large-scale CA on a cluster.

Asynchronous mode $\boldsymbol{\alpha}$ is used when a natural phenomenon on micro or nano level is under simulation, the α-mode offers the following algorithm of a global operator application:

1. A cell $\boldsymbol{x} \in X$ is chosen with probability $p = 1/N$, $N = |X|$;
2. If the condition of its applicability (5) is satisfied, $\theta_i(\boldsymbol{x})$ is applied to $\boldsymbol{x}$, adjusting the cell states from $T'_i(\boldsymbol{x})$ immediately;
3. The global operator is considered to be executed, when Steps 1 and 2 are applied $|X|$ times, completing the tth iteration.

In reality, the asynchronous computation process is not divided into iterations, the notion of iteration is accepted conditionally for making the comparison with synchronous case more conceptual. Since asynchronous computation is by definition sequential, it is always correct when implemented on a single computer. The problem arises when it is allocated on a parallel computer system. In this case the block-synchronous mode should be used.

Block-synchronous mode $\boldsymbol{\beta}$ combines synchronous and asynchronous modes as a single algorithm of the global operator execution by representing the iteration as a number of sequential synchronous stages, with a random choice of a state to be executed. The β-mode was introduced in [14] to make the asynchronous CA parallelization efficiency be more acceptable relying on the fact that the synchronous CA parallelization is efficient, which is in contrast to the asynchronous case.

The block-synchronous mode is based on constructing a partition $\Pi = \{\Pi_1, \ldots, \Pi_m\}$ of X, as follows:

1. A block $B(\boldsymbol{x})$ containing m adjacent cells is chosen, such that

$$|B(\boldsymbol{x})| = m, \quad B(\boldsymbol{x}) \supseteq T'(\boldsymbol{x}); \tag{9}$$

2. A partition $\Gamma = \{B(\boldsymbol{x}_1), \ldots, B(\boldsymbol{x}_M)\}$ is constructed having $M = N/m$ congruent blocks $B(\boldsymbol{x}_i)$, such that for all $i, j \in \{1, \ldots, M\}$

$$\forall (i \neq j) : B(\boldsymbol{x}_i) \cap B(\boldsymbol{x}_j) = \emptyset, \qquad \bigcup_{i=1}^{M} B(\boldsymbol{x}_i) = X; \tag{10}$$

3. A partition $\Pi = \{\Pi_1, \ldots, \Pi_m\}$ orthogonal to Γ is determined, containing the subsets, that being processed synchronously in any order, execute the computation of $\theta(X)$ correctly.

The correctness of β-mode results from the following property of the orthogonal partitions Γ and Π:

$$|\Pi_k(\boldsymbol{x}) \cap B_j(\boldsymbol{x})| = 1, \quad k = 1, \ldots, m, \quad j = 1, \ldots, M, \tag{11}$$

which provides that any pair of cells $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ belonging to one and the same Π is separated by the distance $d_{ij} > R(T)$, hence, can be updated simultaneously, i.e. synchronously, without violating (7).

Investigation of computational properties of block-synchronous CA [15, 16] has shown that block-synchronous CA are significantly less time consuming than asynchronous CA, enhancing their use in sequential cases as well. The gain in time is due to the difference in random generator calls numbers, which are in α-case and in β-case as follows:

$$E_{RG,\alpha} = N, \quad E_{RG,\beta} = m. \tag{12}$$

Taking into account the fact that the time of a RG call may be two or three orders more than that needed for a substitution application, savings in computer time with β-mode may be significant.

3. Complex cellular automata models

3.1. A composed global operator of a complex CA. A classical conception of CA is unconvincing to be used for simulating complex phenomena, particularly, reaction–diffusion processes because of the following reasons:

In complex processes, several species are involved. Hence, the CA alphabet (usually a symbolic one) should be expanded, and the global operator $\Theta(X)$ should be composed of several substitutions, being referred to as *θ-composition*. Accordingly, a CA with a complex $\Theta(X)$ is called the *complex CA*, otherwise it is called the *simple CA*. Reaction–diffusion CA are always complex ones, containing at least two substitutions: θ_{reac} and θ_{diff}.

The behavior of natural phenomena is usually completely or partly stochastic both with respect to time and space. In fact, it is not known exactly how the process of particles interactions proceeds in a natural phenomenon. Hence, when designing a CA-model, the researcher should have the opportunity to use different types of *θ-composition* and chose the most plausible one.

A formal representation of a complex CA is similar to the classical one, differing in interpretation of two concepts of (1). Namely, an alphabet is allowed to contain several integers and symbols, and the algorithm of computing the transition to a next iteration, referred to as *global mode of operation* is defined not only with $\rho \in \{\alpha, \beta, \sigma\}$ determining the way of cell choosing to be updated, but also by θ-composition indexed by $\mu \in \{\delta, \gamma\}$, which indicates to the way of choosing $\theta \in \Theta$ that may be deterministic (δ) or random (γ),

$$\Theta(X) = \Phi_{\rho,\mu}(\theta_1(x), \ldots, \theta_n(x)). \tag{13}$$

The following types of the global mode of operation are mainly used.

Stochastic global mode $\boldsymbol{\Phi_{\alpha,\gamma}}$ is in wide use, being the idealization of a maximally randomized representation of the particles behavior on micro level. In [3], they are called Monte Carlo simulation method; in [11], they form a class of stochastic CA. A global operator of such a CA is computed as follows.

1. A cell $\boldsymbol{x} \in X$ is chosen with probability $p = 1/|X|$;
2. A substitution $\theta_j \in \Theta$, $j = 1, \ldots, n$, is randomly determined according to the probabilities p_j, computed using the θ intensities $r_1, \ldots, r_n$ as follows:
$$p_j = \frac{r_j}{r_1 + \ldots + r_n}, \tag{14}$$
and immediately applied to $\boldsymbol{x}$ with probability p_j;
3. Every iteration consists of $|X| \times |\Theta| = Nn$ repetitions of Steps 1 and 2.

Block-synchronous stochastic mode $\boldsymbol{\Phi_{\beta,\gamma}}$ is used in large-scale CA to be implemented on a supercomputing cluster, or, sometimes, when computing time is wanted to be reduced. The computation of $\Theta(X)$ is performed as follows:

1. A block $B(\boldsymbol{x})$ containing m adjacent cells is chosen, such that
$$B(\boldsymbol{x}) \supseteq T'_\Sigma(\boldsymbol{x}), \quad T'_\Sigma(\boldsymbol{x}) = \bigcup_{i=1}^{n} T'_j, \quad j = 1, \ldots, n, \tag{15}$$
where T'_j is the underlying template of $\theta_j \in \Theta$;
2. The partitions $\Gamma = \{B(\boldsymbol{x}_1), \ldots B(\boldsymbol{x}_M)\}$ and orthogonal to it $\Pi = \{\Pi_1, \ldots, \Pi_m\}$ are constructed according to (10);
3. At each subset Π_i, $i = 1, \ldots, |M|$, for all cells $\boldsymbol{x} \in \Pi$, the following is done n times: a substitution is chosen according to probabilities (14) and is immediately executed.

Sequential global mode $\boldsymbol{\Phi_{\sigma,\mu}}$ is a superposition of global operators of simple $\theta_j(X)$, each of them operating in its own mode, i.e.,

$$\Theta(X) = \theta_{n,\rho_n}(\theta_{n-1,\rho_{n-1}}(\ldots \theta_{1,\rho_1}(X))). \tag{16}$$

Here θ_{j,ρ_j}, $j = 1, \ldots, n$, are chosen in a deterministic order, while each of them operates as a simple global operator in its own mode of operation ρ_j.

3.2. The concept of CA stochasticity. For investigating the interdependence between the stochastic character of a CA behavior and its simulation performance, a quantitative measure of CA stochasticity is essential. To introduce the concept, let us denote the total number of elementary operations per iteration as E, the number of randomly chosen operations — as E_{rand} and the number of deterministic operations — as $E_{\text{det}} = E - E_{\text{rand}}$. An operation is considered to be randomly chosen if its application requires access to a random number generator (RNG). So, the asynchronous choice of a cell $\boldsymbol{x} \in X$ is a randomly chosen operation, no matter how many random numbers are used for obtaining $\boldsymbol{x}$, since $\boldsymbol{x}$ may be represented by two or three coordinates in a discrete space. A randomly chosen substitution $\theta_j \in \Theta$ is also a randomly chosen operation. However, the deterministic application of a probabilistic θ is not randomly chosen, although it requires the use of RNG.

Let us define a *stochasticity* λ as a fraction of randomly chosen operations in computing $\Theta(X)$, $\lambda = E_{\text{rand}}/E$.

Stochasticity depends both on the mode of CA operation (ρ) and on the type of θ-composition (μ). In a simple asynchronous CA $\aleph_\alpha$ the total number of operations per iteration is $E_\alpha = 2N$, $N = |X|$, because there is N applications of θ to N randomly chosen cells. Hence, $E_{\alpha,\text{rand}} = E_{\alpha,\text{det}} = N$. In a simple block-synchronous CA $\aleph_\beta$, $E_\beta = N + mM = 2N$, including only m randomly chosen operations.

Hence, stochasticity of all types of simple CA are as follows:

$$\lambda_\alpha = \frac{1}{2}, \quad \lambda_\beta = \frac{m}{2N}, \quad \lambda_\sigma = 0. \tag{17}$$

From the the definitions of global operator modes, given in Section 3.1, the stochasticity of all type of complex CA are easily obtained.

For $\Phi_{\alpha,\gamma}$ random choice of $\boldsymbol{x} \in X$ and random choice of $\theta_j \in \Theta$, $j = 1, \ldots, n$, are done $n = |X| \times |\Theta|$ times plus the same quantity of deterministic θ- applications, which comprises the total number of operations $E = 2Nn$, a half of them being randomly chosen.

For $\Phi_{\beta,\gamma}$, the total number of operations is also $E = 2Mmn = 2Nn$, among them $E_{\text{rand}} = mn$ are randomly chosen, yielding $\lambda_{\beta,\gamma} = 1/2M = m/2N$.

For sequential global mode $\Phi_{\rho,\delta}$, the stochasticity is calculated as a sum of λ_j of all simple $\theta_{j,\rho_j}(X)$. Particularly, when all $\theta_{j,\rho_j}(X)$ are asynchronous, $\lambda_{\alpha,\delta} = n/2$.

Hence, stochasticity of all types of complex CA is as follows:

$$\lambda_{\alpha,\gamma} = \frac{1}{2}, \quad \lambda_{\beta,\gamma} = \frac{m}{2N}, \quad \lambda_{\rho,\delta} = \sum_{j=1}^{n} \lambda(\theta_{j,\rho_j}). \tag{18}$$

4. Parallel implementation of complex CA

4.1. Parallelization efficiency via stochasticity. Parrallel implementation of cellular automata models are based on the domain decomposition method, which consists of dividing the cellular array Ω into N parts, referred to as *subdomains*, $\Omega = \omega_0 \cup \ldots \cup \omega_{N-1}$, and allocate them onto N processors, working in parallel. In order to provide interprocessor data exchange the subdomians have to be supplemented by V peripheral cells, called *shadow cells*,

$$V = 2DR_{\Sigma}H^{D-1}, \tag{19}$$

where $D \in \{1, 2, 3\}$ is the domain dimension, R is the united template radius (15), and H is the linear size of the domain. Let us assess weak parallelization efficiency as follows:

$$\eta_{\rho,\mu} = \frac{t_1(\omega)}{t_N(\Omega)} = \frac{t_1(\omega)}{t_1(\omega) + l_{\rho,\mu}t_{\text{exch}}}, \tag{20}$$

where $t_N(\Omega)$ is the time needed for executing one iteration in N subdomains using N processors ($|\Omega| = N|\omega|$), and $l_{\rho,\mu}$ is the number of data exchanges, t_{exch} is the time needed for transmitting data from one processor to another. So, $l_{\rho,\mu}$ causes a decrease in efficiency, being dependent on the global mode of operation $\Phi_{\rho,\mu}$. According to (7), a simple synchronous CA $\aleph_{\sigma}$ requires inter-processor data exchange only once per iteration, i.e. $l_{\sigma} = 1$. In the asynchronous case, each peripheral cell adjustment should be transferred to the adjacent processor immediately (19), hence $l_{\alpha} = V$, which is absolutely unacceptable, and, hence, β-mode with m exchanges is used instead. Summarizing, this yields

$$l_{\sigma} = 1, \quad l_{\alpha} = V, \quad l_{\beta} = m. \tag{21}$$

From (21) it follows that synchronous CA are ideal for parallel implementation and asynchronous CA should not be allocated on clusters without transforming into the block-synchronous mode, the block-size m being the indicator of the balance between stochasticity and parallelization efficiency. With allowance for (20) it should be stated that *parallelization efficiency may be achieved at the expense of stochasticity.* Graphically, it is represented in Figure 3, where these two characteristics are shown for the block-synchronous mode depending on m.

When simulating a large-scale complex phenomenon thought of in terms of several types of particles interacting with different rates asynchronously in discrete space, the most appropriate model seems to be a stochastic global mode of operation. But allocating the task on a computer cluster, changes the preference towards the block-synchronous mode, where the choice of the block size m is crucial for achieving a proper balance between the process

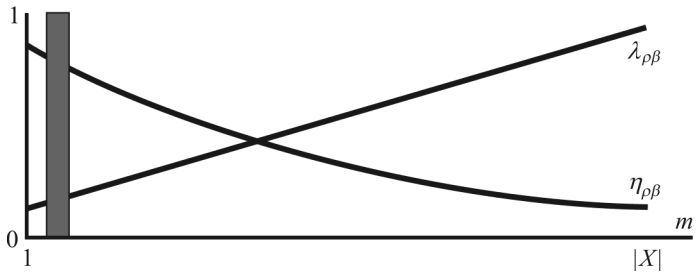

Figure 3. The dependence of block-synchronous CA stochasticity and parallelization efficiency on the number of data exchanges per iteration. A zone of preferable values of m is shown in dark color

stochasticity and parallelization efficiency. The responsibility for a proper choice of m is on the model designer, who has to assess the role of stochasticity in the mathematical representation of a phenomenon and to make a decision.

If the computation time is the most important parameter, then m should be taken as small as possible, i.e. $m_{\min} = |T'_{\Sigma}|$, its impact to the performance being two-fold: increasing parallelization efficiency (20) and decreasing the time for RG calls (12).

4.2. The simulation results. A reaction–diffusion process identified as *Diffusion Limited Aggregation* (DLA), hereinafter called *aggregation*, has been chosen as an example for the simulation. The process may be described as a set of organic particles wandering randomly in a reservoir filled with water. In the reservoir there are also a number of immobile heavy structures (nuclei) allocated inside the reservoir. When a walking particle occurs close enough to an immobile one, it becomes immobile as well. The process looks like a kind of a growing tree, forming a fractal structure (Figure 4).

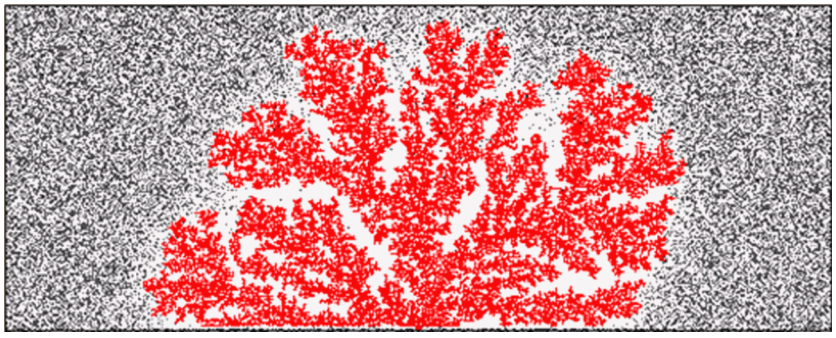

Figure 4. A snapshot of simulation a 2D version of aggregation process $\Phi_{\alpha,\gamma}(\theta_{\text{diff}}, \theta_{\text{stick}})$ initialized by a single nuclei

There is no continuous function or differential equation representing the aggregation process. Its first mathematical model has been expressed in terms of interactions and displacements of particles [17], being then directly mapped onto a computer. Later on, the process was simulated by a reaction-diffusion CA, where diffusion is given as a random walk, and reaction — as transformation of a walking particle to an immobile one (sticking process). The model is used for simulating a wide range of phase transition processes such as electric galvanization [18], formation of crystal structure [19], growing of settlements [20], etc.

The DLA is simulated by a complex CA $\aleph_{\rho,\mu} = \langle A, X, \Theta(X)\rangle$, where $A = \{0, 1, b\}$, 0 meaning a water cell, 1 — a cell with a mobile particle, and b — a cell with an immobile particle; $X = \{\boldsymbol{x}\}$, $\boldsymbol{x} = (i, j)$ for 2D processes, and $\boldsymbol{x} = (i, j, k)$ for 3D ones; $\Theta(X) = \Phi_{\alpha,\gamma}(\theta_d(\boldsymbol{x}), \theta_r(\boldsymbol{x}))$, $\theta_d(\boldsymbol{x})$ simulating mobile particles wandering, $\theta_r(\boldsymbol{x})$ simulating sticking.

The particles wandering process is simulated by a well-known rule of asynchronous *naive* diffusion CA [21, 22],

$$\begin{aligned}\theta_d(\boldsymbol{x}) : \{(u_0, \boldsymbol{x} + \boldsymbol{a}_0), \ldots, (u_l, \boldsymbol{x} + \boldsymbol{a}_l), \ldots, (u_d, \boldsymbol{x} + \boldsymbol{a}_d)\} \xrightarrow{p_d} \\ \{(u_l, \boldsymbol{x} + \boldsymbol{a}_0), \ldots, (u_0, \boldsymbol{x} + \boldsymbol{a}_l), \ldots, (u_d, \boldsymbol{x} + \boldsymbol{a}_d)\},\end{aligned} \tag{22}$$

and the sticking process is simulated by the substitution

$$\theta_r(\boldsymbol{x}) : \{(1, \boldsymbol{x}), (b, \boldsymbol{x} + \boldsymbol{a}_l)\} \xrightarrow{p_r} \{(b, \boldsymbol{x},)\}, \tag{23}$$

where $l \in \{1, \ldots, 2d\}$ is a randomly chosen number, $\boldsymbol{a}_l$ are orts shifting $\boldsymbol{x}_l$ to its lth neighbor, p_d and p_r are probabilities of an application θ_d and θ_r, respectively.

A two-dimensional version of the CA with $\aleph_{\alpha,\gamma}$ has been performed for selecting proper parameters for the large-scale 3D experimental simulation, namely, probabilities p_d and p_r, and the initial density $D(0)$ of mobile particles. As a result, the following parameters have been adopted: $D(0) = \langle u\rangle_{t=0} = 0.5$, $p_d = 0.9$, $p_r = 0.1$ (see Figure 4).

The 3D large scale computer simulation was carried out on the cellular array of $1600 \times 1600 \times 1600$ size (Figure 5). The initial global state $\Omega(0) = \{(u, \boldsymbol{x}) : \ \boldsymbol{x} \in X\}$ contained uniformly distributed "ones" and "zeros", and 16 nuclei $2 \times 2 \times 2$ cell cubes with $u = b$. The whole array was allocated on 64 cores of the cluster, each processing a subdomain having $400 \times 400 \times 400$ cells. To provide data exchange between processes, each subdomain was enlarged up to $402 \times 402 \times 402$ cells, the added shadow cells serving for sending-receiving data. The simulation was performed for the following types of global modes of operation: asynchronous stochastic ($\Phi_{\alpha,\gamma}$), block-synchronous stochastic ($\Phi_{\beta\,\gamma}$) and asynchronous sequential ($\Phi_{\alpha,\delta}$). The computation lasted up to $t = 10{,}000$ iterations. The time per

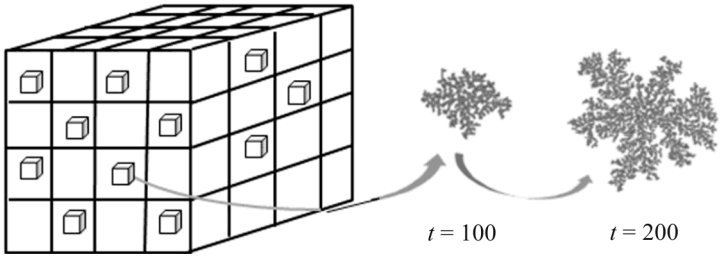

Figure 5. A schematic picture of simulation of 3D diffusion-limited aggregation initialized by 16 nuclei, the whole task being allocated in 64 processors

iteration for each case was calculated as an average over the simulation time: $t_{\text{iter}} = T_{10000}/10000$.

The aim of the simulation was twofold:

1. By simulating the process on a single processor (sequentially) to determine what is the difference in the simulation time between asynchronous stochastic mode (γ, α) and block-synchronous stochastic mode (γ, β);
2. By simulating the process on a cluster, to find out how the parallelization efficiency depends on the stochasticity of a model.

From (12) and (20) it follows, that in both cases the crucial model parameter affecting the computation time is the value of m in β-mode of operation. An intense desire to reduce the simulation time yields in choosing a minimal $m = |T'_\Sigma|$. On the other hand, Monte Carlo methods [3], as well as the asynchronous CA models [7], are based on the principle of maximal stochasticity. There is no theoretical answer to the question in what extent it is admissible to vary stochasticity without violating a given simulation accuracy. Although, some attempts have been made to resolve the contradiction. Thus, in [16] and [15] it was shown, that the simulation results for different modes of operations of one and the same reaction–diffusion CA are quite similar, i.e. the difference of evolutions $\Omega(t)$ are within the limits of statistical uncertainty. However, the physical time of an iteration execution significantly differs, due to the different amount of random generator calls (12). The above considerations allow one to deduce, that since the general problem of admissible stochasticity variations is not resolved, and, moreover, it is not proved, the maximal stochasticity shows the best correlation with a real phenomenon, then let us leave the choice of m on the responsibility of the model developer.

The difference in the simulation time for asynchronous and block-synchronous modes obtained by using CA with the above-given parameters on

Table 1. The time per iteration for asynchronous and block-synchronous modes obtained on a single processor, $p_d = 0.9$, $p_r = 0.1$, $|X| = 400 \times 400 \times 400$

Global mode	α, γ	β, γ	α, δ	β, δ
t_{iter}, s	59.75	13.4	77.2	14.6

Table 2. Dependence of parallelization efficiency (η) and stochasticity (λ) on the block size $m = R(T)$, obtained by simulation $\aleph_{\alpha,\gamma}$ with $p_d = 0.9$, $p_r = 0.1$, $|X| = 1600 \times 1600 \times 1600$ on 64 cores of a cluster

m	9	25	49	81
η	0.883	0.732	0.5	0.457
λ	$0.089 \cdot 10^{-6}$	$1.9 \cdot 10^{-6}$	$14.35 \cdot 10^{-6}$	$64.87 \cdot 10^{-6}$

a single processor (Table 1) shows that the block-synchronous case is about 5 times faster.

The dependence of parallelization efficiency and stochasticity, obtained by simulation using CA $\aleph_{\alpha,\gamma}$ with the above parameters on 64 cores of cluster NKS-30 of the Siberian Supercomputer Center is presented in Table 2.

5. Conclusion

The dependence of computational performance on stochasticity of asynchronous reaction–diffusion CA-models is investigated. The investigation aims at finding a compromise between the simulation efficiency and inherent natural stochasticity of simulated phenomena. In order to quantitatively assess these properties the notion of the CA-model stochasticity is introduced as a fraction of randomly chosen elementary operations in the asynchronous CA evolution. It is shown, that inducing some amount of synchronization by using the block-synchronous mode of operation it is possible to control the simulation performance. It is most important in large-scale simulation tasks implemented on a parallel computer architecture. The information may be essentially helpful at the stage of the CA-model development, when the elementary operations interaction should be determined with allowance for parallel implementation conditions. The results of a computational experiment are presented to give the impression of concrete relations between stochasticity values and parallelization efficiency.

References

[1] Hoekstra A.G., Kroc J.K., Sloot P.M.A. Simulating Complex Systems by Cellular Automata. — Springer, 2010.

[2] Monte Carlo Methods in Chemical Physics / D.M. Ferguson, J.I. Siepmann, D.G. Truhlar, eds. — Willey and Sons Inc., 2007. — (Advances in Chemical Physics; 105).

[3] Matveev A.V., Latkin E.I., Elokhin V.I., Gorodetskii V.V. Turbulent and stripes wave patterns caused by limited CO_{ads} diffusion during CO oxidation over Pd(110) surface: kinetic Monte Carlo studies // Chem. Eng. J. — 2005. — Vol. 107. — P. 181–189.

[4] Nurminen L., Kuronen A., Kaski K. Kinetic Monte Carlo simulation on patterned substrates // Phys. Rev. — 2000. — B63, 035407. — P. 3–15.

[5] Chatterjee A., Vlaches D.G. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods // J. Comp. Aided Mater. Des. — 2007. — Vol. 14. — P. 253–308.

[6] Desai R.C., Kapral R. Dynamics of Self-Organized and Self-Assembled Structures. — Cambridge University Press, 2009.

[7] Kireeva A. Parallel implementation of totalistic cellular automata model of stable patterns formation // Lect. Notes in Comp. Sci. — 2013. — Vol. 7979. — P. 347–360.

[8] Echieverra C., Kapral R. Molecular crowding and protein enzymatic dynamics // Phys. Chem. — 2012. — Vol. 14. — P. 6755–6763.

[9] Vitvitsky A. CA-model of autowaves formation in the bacterial MinCDE system // Lect. Notes in Comp. Sci. — 2015. — Vol. 9251. — P. 246-250.

[10] Bandini S., Bonomi A., Vizzari G. What do we mean by asynchronous CA? A reflection on types and effects on asynchronicity // Lect. Notes in Comp. Sci. — 2010. — Vol. 6350. — P. 385–395.

[11] Bandman O., Kireeva A. Stochastic cellular automata simulation of oscillations and autowaves in reaction–diffusion systems // Siberian J. Num. Math. — 2015. — Vol. 18, No. 2. — P. 255–274 (In Russian).

[12] Bandman O. Functioning modes of asynchronous cellular automata simulating nonlinear spatial dynamics // Appl. Discr. Math. — 2015. — No. 1. — P. 110–124 (In Russian).

[13] von Neumann J. Theory of Self-Reproducing Automata / A.W. Burks, ed. — Urbana, USA: University of Illinois Press, 1960.

[14] Bandman O. Parallel simulation of asynchronous cellular Automata Evolution // Lect. Notes in Comp. Sci. — 2006. — Vol. 4173. — P. 41–48.

[15] Elokhin V.I., Sharifulina (Kireeva) A.E. Simulation of heterogeneous catalytic reaction by asynchronous cellular automata on multicomputer // Lect. Notes in Comp. Sci. — 2011. — Vol. 6873. — P. 204–209.

[16] Bandman O. Contradiction between parallelization efficiency and stochasticity in cellular automata models of reaction-diffusion phenomena // Lect. Notes in Comp. Sci. – 2015. — Vol. 9251. – P. 135–149.

[17] Witten T.A.Jr., Sander I.M. Diffusion-limited aggregation, a kinetic critical phenomenon // Phys. Rev. Lett. — 1981. — Vol. 47, No. 19. — P. 1400–1403.

[18] Ackland G.J., Tweedie E.S. Microscopic model of diffusion limited aggregation and electro deposition in the presence of leveling molecules // Phys. Rev. E. — 2006. — 73, 011606 (arXiv.org > cond-mat > arXiv:cond-mat/0508675).

[19] Bogoyavlenskiy A., Chernova N.A. Diffusion-limited aggregation: A relationship between surface thermodynamics and crystal morphology // Phys. Rev. E. — 2000. — No. 2. — P. 1629–1633.

[20] Batty M., Longley P. Urban growth and form: scaling, fractal geometry, and diffusion-limited aggregation // Environment and Planning. — 1989. — Vol. A, No. 21. — P. 1447–1472.

[21] Toffolli T., Margolus N. Cellular Automata Machines: A New Environment for Modeling. — USA: MIT Press, 1987.

[22] Bandman O. Cellular automata diffusion models for multicomputer implementation // Bull. Novosibirsk Comp. Center. Ser. Comp. Science. — Novosibirsk, 2014. — Iss. 36. — P. 21–31.