3-D Cellular Automata Model of Fluid Permeation through Porous Material *

Olga Bandman

Supercomputer Software Department
ICM&MG, Siberian Branch, Russian Academy of Sciences
Pr. Lavrentieva, 6, Novosibirsk, 630090, Russia
E-mail: bandman@ssd.sscc.ru

Abstract. A 3D Cellular Automata (CA) model for simulating fluid permeation through porous material with complex morphology is developed and investigated. The model is a composition of two interacting CA: one, simulating fluid convection, induced either by gravitation force or by external pressure, and another — simulating fluid surface leveling by diffusion. Both CA process the same discrete space, their operation being separated in time and space, which simplifies essentially parallel implementation. The CA model is tested on an example of water permeation through soil. Results of its parallel implementation on a multiprocessor with distributed memory are presented. A tomographic digitized representation of a 3D soil sample was kindly given by Prof.Wim Cornelis.[1] The simulation program was implemented on the cluster of Siberian Supercomputer Center of Siberian Branch of Russian Academy of Science.

1 Introduction

In connection with new technologies of production and implementation of porous materials new claims are set to computer simulation of processes in them. Dealing with porous media in the same manner that with bulk materials, does not satisfy neither designers of new composites, nor researchers of soil fertility. In order to understand how the porous material interacts with the permeating fluid, it is necessary to make allowance for internal properties of the material, among which morphology and interaction character between pore walls and incoming fluid are the most important. Unfortunately, conventional mathematics, based on partial differential equations, cannot be helpful, because of impossibility to describe pore borders by means of continuous functions [1,2]. This is why, several attempts are made to develop discrete methods of fluid permeation through porous media, because they "do not fear intricate boundary conditions". Such

* Supported by (1)Presidium of Russian Academy of Sciences, Basic Research Program 15-9 (2013), (2) Siberian Branch of Russian Academy of Sciences, Interdisciplinary Project 47, (3) Russian Fund for Basic Research grant 11-01-00567a

[1] Prof. Wim Cornelis, SoPHy, Department Soil Management, Ghent University, Coupure links 653, 9000 Gent, Belgium.

methods are based on Lattice-Gas [3] and Lattice-Boltzmann [4] discrete hydrodynamics, conventional cellular automata [5,6] being also considered. Among them, the models based on Lattice-Gas principles are predestined to simulate laminar flows. They are appropriate when the flow constantly goes through a piece of porous material, as it happens in polymer membranes or in carbon electrodes of hydrogen energetic cells [1]. But if transient processes are of interest, such that fluid permeation to make the material damp, or, conversely, evaporation to make it dry, then Lattice-Gas based models occur to be stiff, especially for parallel implementation [7]. Hence, more adaptable models are needed, which are capable to simulate all kinds of particles moving and interacting with solid pore walls.

The described version of a CA model simulates three forms of particle movements: forced convection, diffusion (surface leveling) and phase transition (evaporation). In fact, it is a discrete version of the convection-diffusion partial differential equation [8], the latter being capable to work in linear channels only. Being very simple, the proposed model is flexible enough: modification of CA transition functions allows to add the hydrophilous soaking, drainage and other processes.

The paper consists of Introduction, four sections, Conclusion, and list of references. In the Introduction the motivation of the investigation is explained. The second section presents formal definitions. In the third — the computation algorithm is given. The results of simulation are shown in the fourth (sequential version) and fifth (parallel version) sections and discussed in the Conclusion.

2 Formal Representation of a CA Model

CA is a set of identical computing units, that are represented as pairs $(u, \mathbf{x})$, called *cells*, where $u \in A$ is a *cell state* from the alphabet A, $\mathbf{x} \in X$ is a *name*, usually given by a vector $\mathbf{x} = (i, j, k)$ from a set of coordinates of a finite 3–dimensional discrete space X. A set of names

$$T(\mathbf{x}) = \{\mathbf{x}, \mathbf{x} + \mathbf{a}_1, \ldots, \mathbf{x} + \mathbf{a}_{n-1}\}, \tag{1}$$

where $\mathbf{a}_j$ is a shift vector, $n = |T(\mathbf{x})|$, is called a *template*. The cells named from $T(\mathbf{x})$ form a *local configuration*

$$S(\mathbf{x}) = \{(u_0, \mathbf{x}), (u_1, \mathbf{x} + \mathbf{a}_1), \ldots, (u_{n-1}, \mathbf{x} + \mathbf{a}_{n-1})\}. \tag{2}$$

The set of cells $\Omega = \{(u_i, \mathbf{x}_i) | u \in A, \mathbf{x} \in X, \mathbf{x}_i \neq \mathbf{x}_j\}$ is referred to as a *cellular array*, and a list of states of cells from Ω — as a CA *global state* $\Omega_A = (u_1, u_2, \ldots, u_{|X|})$.

CA functioning is determined by *operator* $\Theta(X)$ that may be a composition of several more simple operators:

$$\Theta(X) = \Phi(\Theta_1(X), \ldots, \Theta_n(X)), \tag{3}$$

which in their turn, are composed of *substitutions*

$$\theta(\mathbf{x}) : S(\mathbf{x}) \rightarrow S'(\mathbf{x}), \tag{4}$$

where $|S(\mathbf{x})| \geq |S'(\mathbf{x})|$. The first $m' = |S'(\mathbf{x})|$ cells in $S(\mathbf{x})$ comprise the *base* of $\theta(\mathbf{x})$, and the remaining $(m - m')$ cells play the role of a *context*.

A substitution $\theta(\mathbf{x})$ is applicable to a cell $(u, \mathbf{x}) \in \Omega$, if $S(\mathbf{x}) \subseteq \Omega$, a cell with a variable state being applicable, if the range of u is in A. Application of $\theta(\mathbf{x})$ implies replacing base cells states $(u_j, \mathbf{x} + \mathbf{a}_j) \in S'(\mathbf{x})$ by

$$u'_j = f_j(u_1, \ldots, u_n), \quad n = |S(\mathbf{x})|, \quad j = 0, \ldots, |S'(\mathbf{x}|, \tag{5}$$

where $f_j(u_1, \ldots, u_n)$ – is a *transition function*. The context cells states remain unchanged.

Application $\theta(\mathbf{x})$ to all $\mathbf{x} \in X$ comprises a *global operator* $\Theta(X)$, which executes the transformation of $\Omega(t)$ into $\Omega(t + 1)$, performing an *iteration*. In order to avoid conflicting situations leading up to data loss, the global operator should satisfy the following correctness condition [11]:

$$T_k(\mathbf{x}) \cap T_m(\mathbf{y}) = \emptyset, \quad \forall \mathbf{x}, \mathbf{y} \in X, \quad \forall k, m \in \{1, \ldots, l\}, \quad l = |\Theta(\mathbf{x})|. \tag{6}$$

There are several modes of executing the global operator $\Theta(X)$, the main of them are synchronous and asynchronous ones.

Synchronous mode implies the following sequence of actions: (1)for all $(u, \mathbf{x}) \in \Omega(t)$ new states are computed by (5); (2) in all cells $(u, \mathbf{x}) \in \Omega(t)$ the states $u(\mathbf{x})$ are replaced by $u'(x)$. To satisfy correctness condition (6) in synchronous mode the substitutions (4) of $\Theta(X)$ should have a single cell base, i.e.

$$|S'(\mathbf{x})| = 1 \quad \forall \theta_i \in \Theta(\mathbf{x}), \tag{7}$$

Asynchronous mode implies the following way of local operator application. (1) with probability $p = 1/|X|$ a cell $(u, \mathbf{x}) \in \Omega$ is chosen; (2) $\Theta(\mathbf{x})$ is applied to a chosen cell and the base cells states $(u, \mathbf{x}) \in S(\mathbf{x})$ are immediately replaced by the corresponding $(u', \mathbf{x}) \in S'(\mathbf{x})$. By condition, it is agreed that $|X|$ repetitions of (1) and (2) comprise an iteration. Such an agreement is helpful, because it is in accordance with the synchronous mode and with a definition of a step in kinetic Monte-Carlo method [9].

Since in asynchronous CA local operator is applied sequentially, condition (6) is always met. The problem of correct computation arises only if the algorithm of asynchronous CA is executed in parallel on several processors [10,12].

Ordered asynchronous mode is also frequently used. It is a modification of the asynchronous mode, when $\theta(\mathbf{x})$ is applied sequentially to the ordered cell set.

Besides these modes any other one is also admissible, if it fits the phenomenon under simulation and satisfies (7). The mode of functioning is an essential parameter of a CA, i.e. if two CA differ only by functioning modes, then their

evolutions may be quite different. Thus, the unambiguous definition of a CA is a four tuple

$$\aleph = \langle A, X, \Theta, \rho \rangle, \tag{8}$$

where ρ is the mode of operation. These four notions define the algorithm of CA functioning. But, they do not define the character of its behavior: one and the same CA with different initial global states may generate quite different evolutions. The bright example is the well known CA referred to as "Conway's Game of Life "[13]. So, a CA model of a specific process is further given as a five-tuple

$$\mathbf{A} = \langle A, X, \Theta(X), \rho, \Omega(0) \rangle. \tag{9}$$

3 The Algorithm of Simulation Water Permeation through Soil

In the CA model under consideration the superposition of several CA is used [14], which is represented as a composed CA (8), whose notions have the following interpretation.

State alphabet $A = \{0, 1, 2, 3\}$, where 0 is interpreted as a void space, 1 — as solid substance of pore walls, 2 — as a water particle, 3 — as soaked solid, inherent to hydrophilous materials.

Discrete space X is a Cartesian lattice

$$X = \{(i, j, k) : i = 0, \ldots, I; j = 0, \ldots, J; k = 0, \ldots, K\}. \tag{10}$$

Operator $\Theta(X)$ is a global composition

$$\Theta(X) = \Theta_C(X) \circ \Theta_D(X) \circ \Theta_H(X), \tag{11}$$

where $\Theta_C(X)$, $\Theta_D(X)$, and $\Theta_H(X)$ are operators of convection, diffusion, and hydrophilicity, respectively. Global superposition implies application of each global operator to the result of the preceding one.

Convection operator $\Theta_C(X)$ is in its turn a global superposition

$$\Theta_C(X) = \Theta_G(X) \circ \Theta_E(X), \tag{12}$$

where $\Theta_G(X)$ and $\Theta_E(X)$ are single substitution operators, each consisting of $\theta_G(i, j, k)$ and $\theta_E(i, j, k)$, for gravitation and evaporation, respectively.

$$\theta_G(i, j, k) : \{(1, (i, j, k))(2, (i, j, k-1))\} \xrightarrow{p_G} \{(2, (i, j, k)(1, (i, j, k-1)\} \tag{13}$$

simulates the propagation of water particles along the gravitation direction k down, letting void particles pass up. Such a movement is simulated by K sequential steps in ordered asynchronous mode, at each l-th step ($l = 1, \ldots, K$) (13) being applied synchronously to all cells having as a third coordinate $k = K - l$. Probability in (13) may be taken as $p_G = 1$, since propagation rate under gravitation is assumed to be larger than that under evaporation.

The substitution

$$\theta_E(i,j,k) : \{(2,(i,j,k))(1,(i,j,k-1))\} \xrightarrow{p_E} \{(1,(i,j,k)(1,(i,j,k-1)\} \quad (14)$$

simulates evaporation, which is a phase transition process, leading to a decrease of total water amount. Probability p_E in general case is the function of k, allowing for the differentiation of evaporation intensity in the vicinity of the surface and in deep layers in the soil. $\theta_E(i,j,k)$ is applied in ordered asynchronous mode along the kth axis, $(k = 0,\ldots,K-1)$, at each k step applying $\theta_E(i,j,k)$ synchronously to all cells having $(i,j) \in \{I \times J\}$.

Diffusion operator $\Theta_D(X)$ simulates the process of density equalization, which implies leveling the free surface of water, whatever it might be: in caverns, horizontal pores and over the soil surface. In order to conform the rates of diffusion and convection operators, the first should be applied n times in sequence, n being large enough to provide water surface smoothing.

$$\Theta_D(X) = (\Theta_d(X))^n, \quad (15)$$

where $\Theta_d(X)$ contains one probabilistic substitution $\theta_D(i,j,k)$ called *naive diffusion* [15], based on a five-point local configuration

$$\begin{gathered} S(i,j,k) = \{(u_0,(i,j,k)),(u_1,(i-1,j,k)),(u_2,(i,j+1,k)),(u_3,(i+1,j,k)), \\ (u_4,(i,j-1,k))\} = \{(u_l(i,j,k))_l : l = 0,\ldots,4\}. \end{gathered} \quad (16)$$

$\theta_D(i,j,k)$ executes the exchange of states between the cell $(2,(i,j,k))$ and one out of those its neighbors in k-th plane, whose state $u_l = 1$, i.e.

$$\begin{gathered} \theta_D(i,j,k) : \{(2,(i,j,k))(1,(i,j,k)_l)\} \xrightarrow{p_D} \{(1,(i,j,k))(2,(i,j,k)_l\}, \\ u_l, u_0 \in \{1,2\}, \quad l = 1,2,3,4. \end{gathered} \quad (17)$$

$\theta_D(i,j,k)$ is applied sequentially to all kth planes, $k = 0,\ldots,K$. In each k-th plane it is applied like a 2D asynchronous naive diffusion operator with probability $p_D = 1/m$, where m is the number of cells in $S(i,j,k) \setminus (u_0,(i,j,k))$ with void states. When pore walls are not smooth, and the neighborhood of $(u_0,(i,j,k))$ contains a pore wall cell, then $p_D < 1/m$, which implies, that some water particles may be adhered to the wall.

Hydrophilicity operator $\Theta_H(X)$ is used if the soil has hydrophilous inclusions. It simulates the process of soaking some pore wall cells during the permeation and the reverse process of drainage from soaked cell during the process of water evaporation. So, it is a superposition of two global operators:

$$\Theta_H(X) = \Theta_S(X) \circ \Theta_Q(X),$$

where $\Theta_S(X)$ simulates soaking process, and $\Theta_Q(X)$ — the drainage.

The soaking operator contains a single substitution

$$\begin{gathered} \theta_S(i,j,k) : \{(1,(i,j,k))(2,(i,j,k)_l)\} \xrightarrow{p_S} \\ \{(1,(i,j,k))(3,(i,j,k)_l\} \\ (i,j,k)_l \in T(i,j,k), \end{gathered} \quad (18)$$

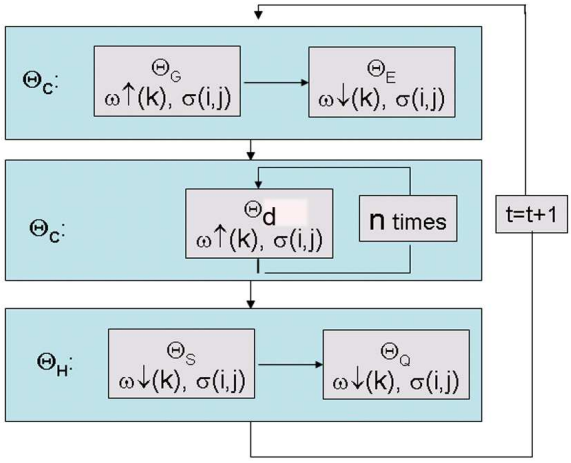

Fig. 1. The schematic representation of the algorithm of porous material damping process simulation: $\omega \uparrow (k)$ stands for ordered asynchronous mode along k-axis up, $\omega \downarrow (k)$ stands for ordered asynchronous mode along k-axis down, $\alpha(i,j)$ stands for asynchronous mode over (i,j) plane, $\sigma(i,j)$ stands for synchronous mode over (i,j) plane.

which is applied in synchronous ordered mode along kth axis and synchronously in each k-th 2D plane. The drainage operator consists of a substitution

$$\theta_Q(i,j,k) : \{(3,(i,j,k))(2,(i,j,k)_l)\} \xrightarrow{p_Q} \{(1,(i,j,k))(2,(i,j,k)_l\} \\ (i,j,k)_l \in T(i,j,k), \tag{19}$$

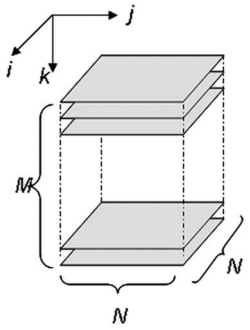

Fig. 2. A 3D cellular array structure of the porous sample

which converts a soaked cell again into solid with the probability p_Q. The probabilities p_S and p_Q depend both on soil properties and on CA parameters, and should be determined by simulation for each type of soil. The above four notions (A, X, Θ, ρ) define the composed CA, whose functioning is represented by the algorithm shown in (Fig.1). To construct the CA model of a particular task, the initial global cellular array should be known as well, which may be obtained from the data that describe the medium morphology under simulation.

4 Results of Sequential Version Implementation

In our case the initial cellular array $\Omega(0)$ is obtained by transforming the digitized representation of a sample of soil of the size $10,304 \times 10,304 \times 21,88892$ mm^3. It was packed into 1480 files, each having the size 700×700 bytes (Fig.2). So, the linear dimension of a cell is $h = 14.72$ mkm, and the cardinality of the cellular array $|X| = 725.2 \cdot 10^6$. The transformation of given files into $\Omega(0)$ was done by creation of a 3D Boolean array of size $I \times J \times K$, $(I = J = 700, K = 1480)$, whose cells $(0,(i,j,k))$ correspond to solid, and cells $(1,(i,j,k))$ correspond to pores.

A sectional view of the soil morphology is shown in Fig.3. Obtaining digitized representation of porous medium morphology is a separate problem. Usually, there are no real sample, and the digitized representation is to be synthesized, provided its characteristics are given. A few methods of pore morphology computer representation construction are known (a good review may be found in [16]). Particularly, there is a method based on totalistic CA evolution [1].

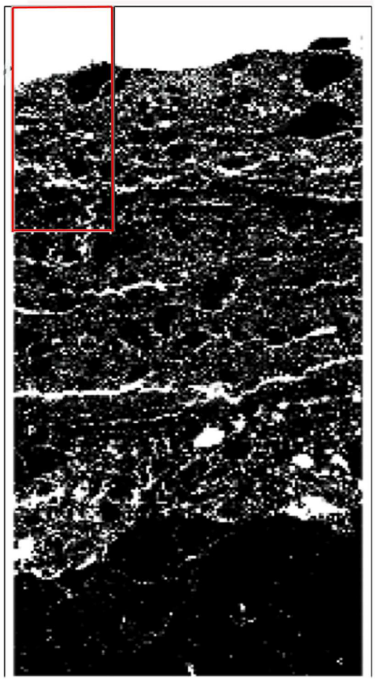

Fig. 3. The cross-section ($j = 400$) snapshot of the soil sample under investigation. The rectangle at the left top side shows the small fragment for sequential test.

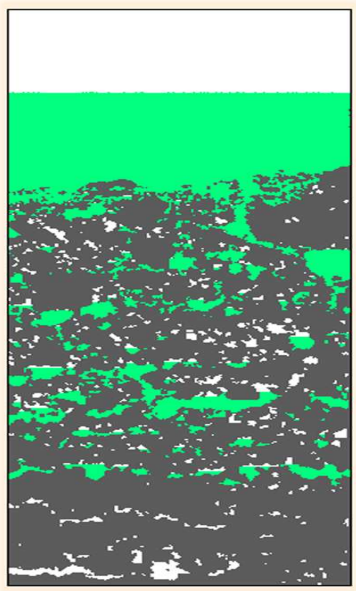

Fig. 4. The cross-section ($j = 50$) snapshot of a small fragment at $t = 50000$ with $p_E = p_H = 0$.Grey cells correspond to water.

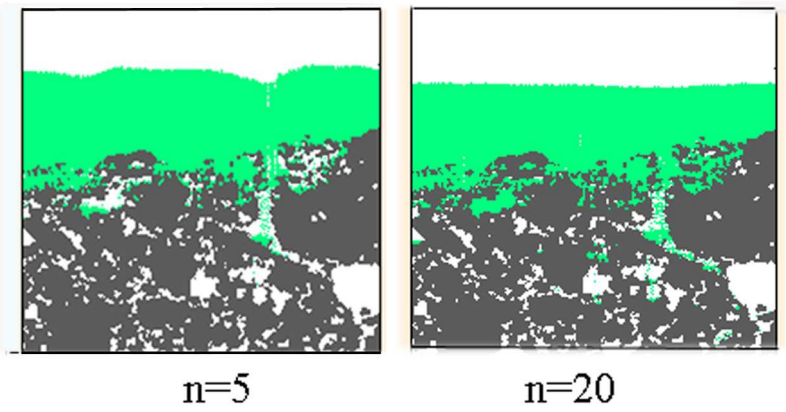

Fig. 5. The upper parts of the cross-section of the small fragment of the sample at $t = 1000$ with $n = 5$ and $n = 20$.

The investigation of computational characteristics of the CA model was done in two stages. At the first stage computational experiments were performed on a small fragment of a sample using a sequential version of the algorithm (Fig.1). At the second stage capabilities of the model were studied by a series of experiments on the whole sample using 14 processors in parallel. Testing the sequential algorithm was performed on the small fragment $200 \times 200 \times 500$ (Fig.3). The program was implemented on Intel Core-i7 (2,66 GHz). Time of an iteration is 0,4 sec. The test aimed to assess water penetration rate and to range the CA model parameters values, which can be obtained only by experimental simulation. Such parameters are the following.

1. Water penetration rate is decreases in time. The depth achieved by the water during T=50000 iterations is approximately 100 cells (Fig.4. Later on permeation rate becomes very slow.

2. Coefficient n in (15), that represents the ratio of diffusion and convection intensities. Its value is obtained by running the program many times, each time increasing the value of n, until the water free surface becomes quite smooth (Fig.5). Coefficient n is the invariant of the a CA model of a concrete process [17], for each new model it should be determined over again.

In Fig.5 two snapshots of the sample fragment cross-section with $j = 50$ are shown. It is seen that with $n = 5$ the surface of the water has not enough time to become smoothed, and $n = 20$ is suitable. The value of n affects the simulation time essentially. That is why it should be taken as small as possible, but large enough to satisfy the smoothing condition.

3. The domain of possible values of evaporating probability p_E. Although the rate of drainage may be assessed basing on physical considerations, the only way of obtain p_E for a certain CA model is computer simulation. The simulation experiments (Fig.6) showed that even with $p_E = 0,01$ the process of damping is strongly affected.

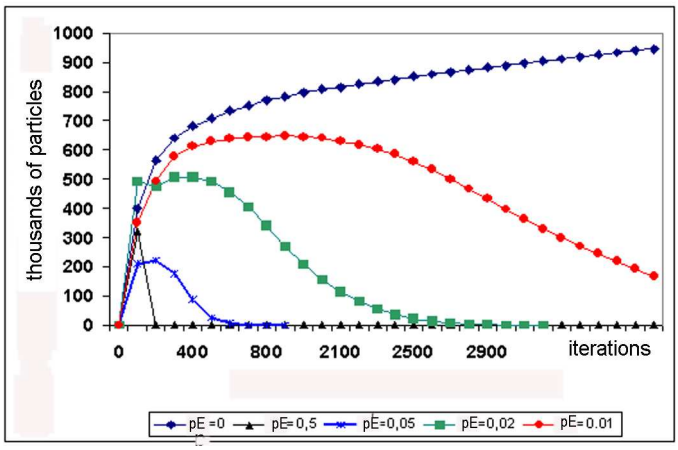

Fig. 6. Dependence of water amount in the soil with different evaporation intensity

The main peculiarity of the proposed 3D CA model is that its convection and diffusion components operate in separated dimensions: convection operates in one-dimension space along k-th axis, diffusion and soaking-drainage — in 2D planes, orthogonal to it. Actually, a method "divide and conquer" is used, that simplifies essentially the programming, especially the parallel version. It is hardly possible to assess how much this simplification decreases the accuracy of simulation, because the object of investigation cannot be precisely defined: porous materials morphology differs from sample to sample and depends from many external factors, such as dampness, temperature etc..

5 Results of Parallel Version Implementation

The architecture of the proposed model predefines the domain decomposition method for parallel processing. Convection being directed along k-th axis determines the direction of the computation domain dissection. The whole domain is decomposed into $m \times n$ subdomains. Each subdomain contains $I/m \times J/n \times K$ cells, and is allocated to a process. Interprocess data exchange is performed after each iteration in all k-th planes along i-th and j-th axes.

The well known problem of asynchronous CA parallelization necessitates to transform asynchronous CA into a block-synchronous one [10]. The procedure of such a transform is applied to all k-th planes $X_k = \{(i,j,k) : i = 0,\ldots,I; j = 0,\ldots,J;\}$, $X_k \subset X$, as follows.

1. X_k is decomposed into $|T| = 9$ nonintersecting subsets $X_k = X_k(0) \cup \ldots X_k$ in such a way, that according to correctness condition (6), intersection of their cells neighborhoods is empty.
2. At each iteration operators $\Theta_D(X)$ and $\Theta_H(X)$ are applied in nine successive stages, at each stage to all $(i,j) \in X_k(l)$, $l = 0,\ldots,8$.

3. Exchange between adjacent parts of X_k is performed after each stage.

The whole cellular array with the size $700 \times 700 \times 1480$ was decomposed into 14 domains, each of the size $350 \times 100 \times 1480$, which were allocated onto two 8-core nodes Intel Xeon 5540, 2.53 GHz (Nehalem) of the cluster NKS-30 in Siberian Supercomputer Center. Initial conditions are as follows: soil pores are empty, and over the soil there is a certain amount of water. Boundary conditions along i and j axes are periodic.

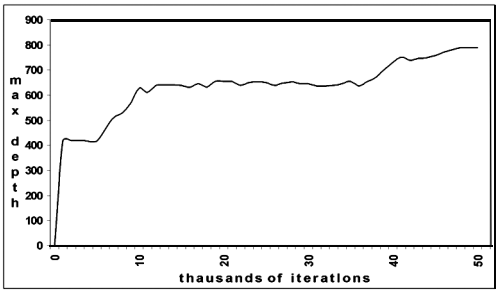

Fig. 7. Dependence of maximum permeation depth on time with $p_E = 0, p_H = 0$.

The program consists of three parts:

1. reading initial digitized sample representation from the file and dividing it into 14 subdomains;
2. executing the algorithm, computing at each iteration the required characteristics (depth of penetration, quantity of water in pores, rate of evaporation, etc.);
3. writing the obtained results into files.

The simulation experiments were performed up to T=50000 iterations aimed to investigate the evolution of permeation process under the following conditions.

- Pure permeation process without evaporation and soaking ($p_E = 0$, p_H=0), aiming to determine maximal depth water particle may reach (Fig. 7). It is seen, that the process is not monotonous due to stratified morphology of the soil.
- Permeation process affected by evaporation ($p_E = 0.001, p_H = 0$), in this case decrease of water amount in soil is also of interest, together with the maximum depth (Fig.8).
- Permeation process in the soil with hydrophilous inclusions ($p_H = 0.001$, $p_E = 0$), water amount in the soil and maximum depth of permeation (Fig.9).
- Permeation process affected both by evaporation ($p_E = 0.001$) and hydrophilicity ($p_S = 0.001, p_Q = 0$) (Fig.10).

Table 1 summarize the results.

Analysis of the above results shows that:

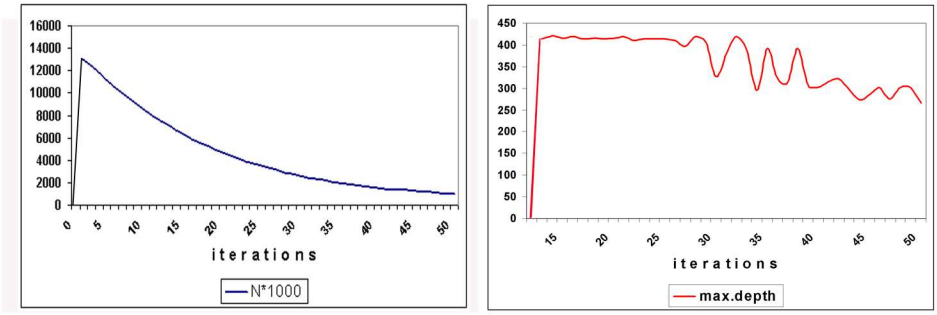

Fig. 8. Dependence of amount of water particles (thousands of particles in pores) and of maximum permeation depth on time (thousands of iterations), with evaporation intensity $p_E = 0.001$, and no hydrophilous inclusions $p_H = 0$

.

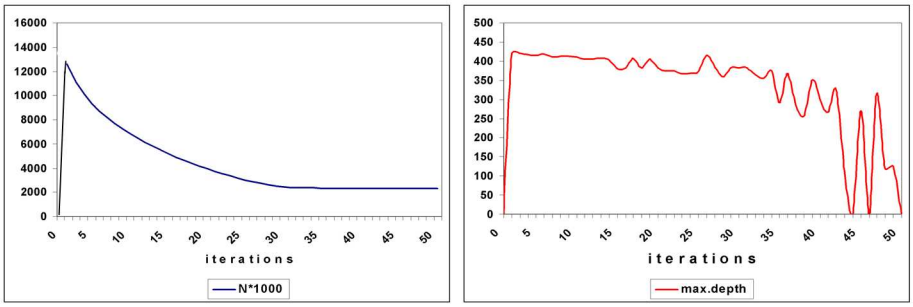

Fig. 9. Dependence of amount of water particles (thousands of particles in pores) in the soil and of maximum permeation depth) on time (thousands of iterations) without evaporation $p_E = 0$, and with hydrophilous inclusions $p_H = 0.001$.

Table 1. Maximum depth of water permeation in soil with different intensities of evaporation and hydrophilicity at T=50000

	$p_E = 0$,$p_H = 0$	$p_E = 0,001$,$p_H = 0$	$p_E = 0$,$p_H = 0.001$	$p_E = 0.001$,$p_H = 0.001$
N*1000	13572	986	2340	734
max.depth	789	267	0	0

1. rate of water permeation decreases essentially in the deep layers of soil:
2. evaporation of water strongly affects permeation process;
3. a small amount of hydrophilous inclusions may block up the permeation.

6 Conclusion

The proposed convection-diffusion CA model of fluid permeation through porous material is remarkable by the fact that its convection and diffusion parts are sep-

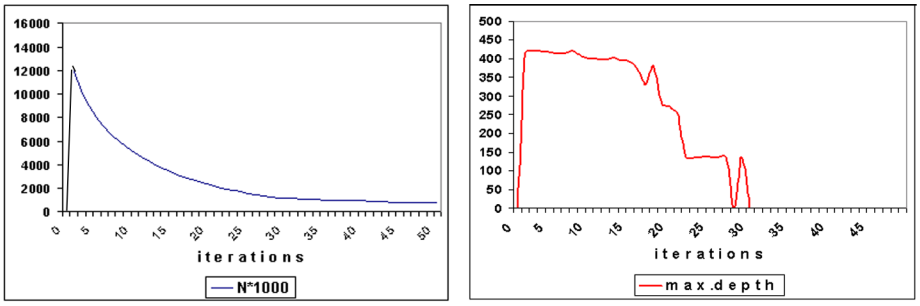

Fig. 10. Dependence of amount of water particles (thousands of iterations)in the soil and of maximum permeation depth on time with evaporation intensity $p_E = 0.001$, hydrophilicity $p_H = 0.001$.

arated both in time and in space. The last feature helps essentially parallel implementation, because it reduces asynchronous computations to a 2D case. The feature which makes the model advantageous relative to Lattice-Gas models is the probabilistic character of propagation operators which allows to differentiate water-solid interaction nature: sticking water particles to the pore wall, soaking and evaporation. Simulation experiments performed with the soil sample show the qualitative similarity of CA evolution to the expected behavior. To obtain quantitative characteristics in real physical values the scaling coefficients are needed. Unfortunately, they may be obtained only by studying the real material properties and comparing them to simulation results.

References

1. Bandman, O.: Using cellular automata for porous media simulation. J.of Supercomputing, 57(2), 121–131 (2011)
2. Keller, L.M. et al.: 3D geometry and topology of pore pathways in Opalinus clay: Implications for mass transport. Appl.Clay Science, 52,85–95 (2011)
3. Rothman, B.H., Zaleski, S. Lattice-Gas Cellular Automata. Simple Models of Complex Hydrodynamics. Cambridge Univ. Press (1997)
4. Hidemitsu, H. : Lattice Boltzmann Method and its Application to Flow Analysis in Porous Media. *R&D* Review of Toyota CRDL 38, N 1, 17–25 (2007)
5. Simulating Complex Systems by Cellular Automata. Understanding complex Systems (A.G.Hoekstra et al.. eds), Springer: Berlin (2010)
6. Bandman, O.L.: Cellular-Automata models of Spatial Dynamics Simulation. In: System informatics, issue 10, Novosibirsk: SBRAS Press, 58–116 (2006) (in Russian)
7. Frish, U., Hasslacher, B., Pomeau, Y.: Lattice-gas Automata for Navier-Stokes equation. Phys. Rev. Letters, 56, 1505–1508 (1986)
8. Sahimi, M.: Flow phenomena in rocks: from continuum models to fractals, percolation, cellular automata, and simulation annealing. Review in Modern Physics,65(4), 1393–1534 (1993)

9. Jansen A.P.J.: An Introduction to Monte-Carlo Simulation of Surface Reactions arXiv:cond-mat/0303028v1[stat-mech](2003)
10. Bandman, O.: Parallel Simulation of Asynchronous Cellular Automata Evolution. LNCS 4173, Proc.ACRI-2010, 41–48, Berlin: Springer (2006)
11. Achasova, S.,Bandman, O.: Correctness of parallel Processes. Novosibirsk: Nauka, 1999) (in Russian)
12. Kalgin. K.: Parallel implementation of asynchronous cellular automama on 32-core computer. Siberian J.Num.Math. 15(1), 55–65, Novosibirsk: SBRAS Press, (2012)
13. Gardner. M.:Mathematical Games - the fantastic combinations of John Conway's new solitaire game "life"". Scientific American 223. pp. 120-123,(1970)
14. Bandman, O.: Cellular Automata Composition Techniques for Spatial Dynamics Simulation. In: Simulating Complex Systems by Cellular Automata. Understanding complex Systems (A.G.Hoekstra et al.. eds), 81-115, Berlin: Springer(2010)
15. Toffolli, T., and Margolus, N.: Cellular Automata Machines: A new Environment for Modeling. MIT Press, USA (1987)
16. Ramirez, A., Jaramillo, D.E.: Porous media generated by using an immiscible Lattice-Gas model. Computational Material Science 65, 12, 157–164 (2012)
17. Basndman, O.: Invariants of cellular automata reactioon-diffusion models. Applied Discrete Mathematics, N3, 55–64 (2012)