Bull. Nov. Comp. Center, Comp. Science, 17 (2002), 63–73

Gas-lattice simulation of high viscous fluid flows*

Yu. Medvedev

1. Introduction

A classical approach to the fluid flows simulation comprises constructing differential equations, transforming them into a finite difference scheme, and obtaining the solution on computer. This approach has two disadvantages: accumulation of rounding off errors during the large number of iterations and a long time of calculations on the fine-cell grid. Discrete models are free from the above shortages. A wide class of discrete models referred to as Cellular Automata (CA) allows simulating many physical processes: diffusion, heat conduction, chemical reactions, fluid flows. Such modeling fluid flows are also called gas-lattice models. There are two well-known basic gas-lattice models in the 2D space: the HPP model (Hardy, Pazzis, Pomeau) and the FHP model (Frish, Hasslacher, Pomeau) [1]. The above gas-lattice models simulate flows of fluids with a small Reynolds number with good accuracy.

As for the 3D gas-lattice models, none of them is found to meet sufficient condition of structural symmetry. Instead, the projections of pseudo-4D models onto a 3D space are used to simulate the 3D flows. This is a basic pseudo-4D FCHC model (Face Centered Hyper Cubic) [2]. But, due to its high computational complexity it is considered to be impractical.

So, there are two problems to be discussed. First, to construct a model, at least, a 2D one, which could adequately describe flows with larger Reynolds numbers. And, second, it is necessary to find a true 3D model simulating the flows in a 3D space, perhaps not so accurate the FCHC but practically acceptable.

In this work, an attempt is made to solve the above problems. In Section 2, the concepts of a gas-lattice model are introduced, the basic 2D models is considered, and a modification aimed at raising a admissible Reynolds number range is proposed. Also, in this section some tests carried out with basic and modified 2D models, are described. A 3D gas-lattice model with 12 neighbors and one rest-particle is proposed in Section 3. The software package allowing the simulation of the 3D flows is presented in Section 4.

*Supported by the Russian Foundation for Basic Research under Grant 01.00.00026.

Results of the experiments, carried out with the package, are discussed.

2. The 2D model investigation results

The 2D space is presented in these models by a cellular array. The spatial points are associated with the automaton cells. There are some hypothetical particles moving in the array. The rules of their behavior conserve the total mass and momentum. The number of particle movement directions is 4 for the HPP model, 6 for the FHP model (Figure 1), and 24 for the FCHC model [2, Figure 5].

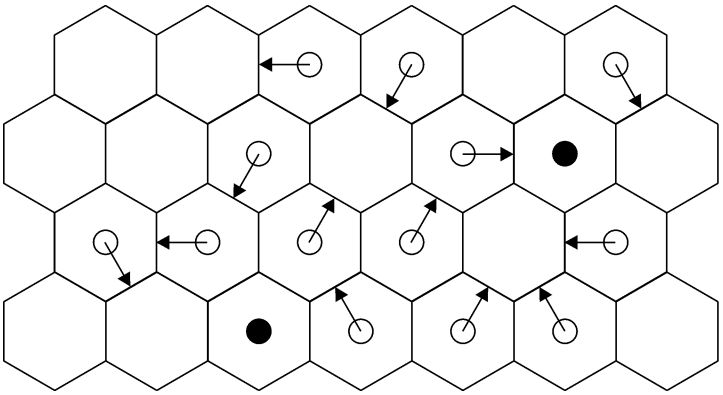

Figure 1. The FHP model

Each time of cellular automaton functioning is divided into two sub-times: collision and propagation. A change of particles velocity directions occurs in the phase of collision according to some rules. In each cell particles collide regardless of particles in other cells. At the sub-time propagation, each particle moves by one cell along its velocity vector. For providing conservation laws it is sufficient to meet the following conditions.

1. Masses of all particles are identical and equal to 1.
2. Velocities of particles are identical in size and can differ in direction only, modules of velocity being 0 or 1.
3. The number of directions of motion is finite and complies with the number of neighbors of each cell.
4. Particles are located in cells of the automaton, moreover, in each cell there cannot be more than 1 having identical velocity vector.
5. Collision rules must be chosen in such a way that the total mass and momentum of all particles, involved in the collision, should not change.

2.1. The FHP model

The FHP model is based on the 2D CA with a hexagonal structure, and each cell has exactly six neighbors (see Figure 1). Each cell is finite probabilistic automaton. Its state is a Boolean vector, in which each of its 6 components shows the existence or the absence of a particle with the corresponding direction of velocity vector. Consequently, the range of a cell state vector is the number of neighboring cells plus the number of rest-particles allowed in a cell. For instance, an FHP model with one rest-particle has a seven-range state vector.

The transition function of the automaton is defined by the collision rules of particles. For the FHP model these rules are shown in Figure 2. Arrows mark directions of the particles velocity vectors. Black circles mark the rest-particles. Three rules, shown in Figures 2a, 2b, and 2c are deterministic because their next states are uniquely defined by the states before the collision. In Figure 2d, a probabilistic rule is shown. Each one is realized with probability 1/2. So, the FHP model is probabilistic, and its cell is a probabilistic finite automaton.

For obtaining a full set of transition rules one needs to add to those shown in Figure 2 all the cases symmetrical to them. In addition, one needs to add to them all identical collisions not changing the cell state.

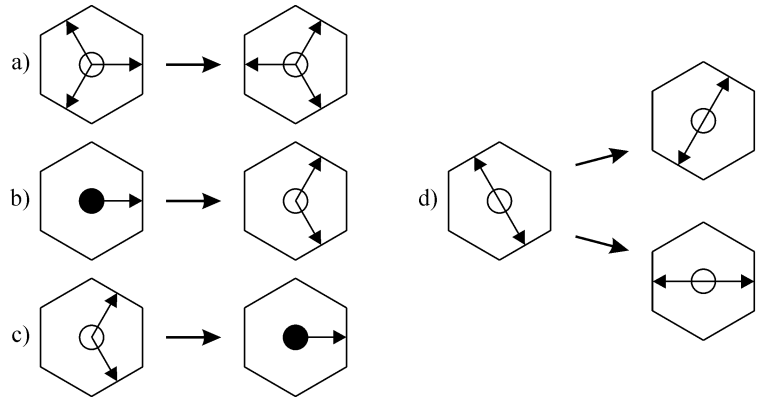

Figure 2. FHP model collision rules

In simulation experiments (in real life), the most interest is not in the hypothetical particles with their velocities of motion, but in averaged values of their velocities. They are obtained as a sum of all the particle velocity vectors over a certain vicinity, and the resulting vectors showed comply with the macroscopic velocity of a real flow at the corresponding points. Since the CA has a hexagonal structure, the vicinity has also the shape of the hexagon (Figure 3).

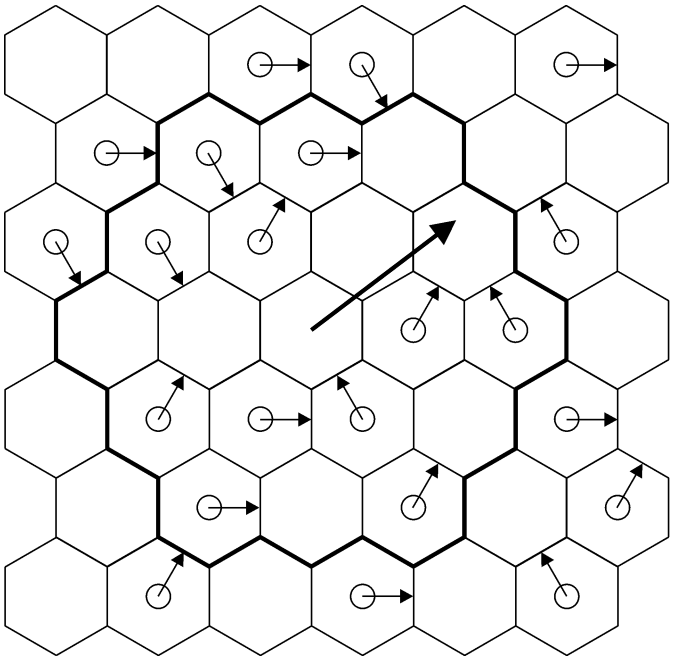

Figure 3. Averaged velocity vector

2.2. The boundary conditions

Simple boundary conditions are among the advantages of gas-lattice models. The cells differentiation according to types is a basis of these conditions. There are three following types of cells used in the model: working cells, wall cells, and particle–source cells. All cells of the same type have identical collision rules. Cells of different types have different collision rules, propagation rules being the same for all types of cells.

To simulate a fluid flow one needs to involve a new type of cells called sources. Each cell of such a type can generate particles with all possible velocity vectors with certain probability. One can obtain a source of the even flow of particles by aligning source cells. By changing the probability of generating particles, one can vary the flow intensity.

Let us consider the border cells transition rules operation in more detail. There are three ways of reflecting the particles from obstacles. The simplest of them is the inverse direction reflection (Figure 4a). A more complicated approach follows the optical reflection laws (Figure 4b), where one needs to take into account the position of neighboring cells of the obstacle. The third one enables a particle to be reflected in any possible direction (where there is no obstacle) with equal probability (Figure 4c). One also may use a compound approach which is a superposition of the above three ways of reflection with predetermined probability of each.

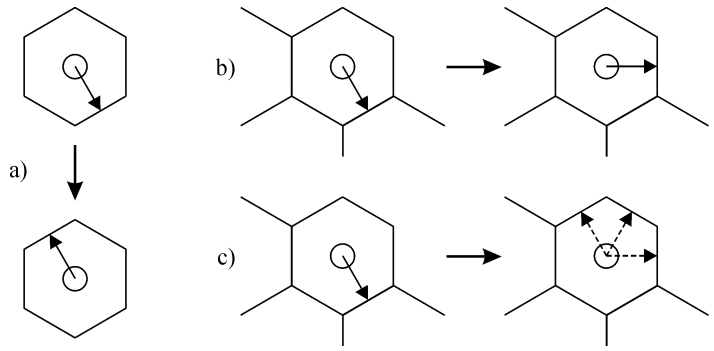

Figure 4. Collisions: a) backward; b) according to the optics law; c) compound

2.3. Simulation of less viscous fluids

The basic gas-lattice model is limited by small Reynolds numbers. A modification of the basic model increasing the Reynolds number, which allows simulating less viscous fluids, is given below [3].

The essence of the proposed modification is to allow a particle, being alone in a cell to change its direction of motion with predetermined probability (Figure 5). The momentum conservation law, certainly, is violated, but when averaging is done over a sufficient number of particles, it is approximately followed.

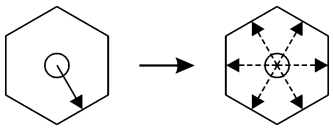

Figure 5. Changing velocity vector of a single particle

Certainly, a strict analytical proof of validation of such a model is a very hard, and possibly, undecidable problem. However it is possible to check the correctness of our assumption by performing numerical experiments and comparing their results with the earlier known.

2.4. Experiments with the FHP models

A program model is realized in C programming language under X-Window shell OS UNIX. It is possible to investigate both the basic and the modified models. The program model in question consists of a system kernel, executing transition functions of elementary automata of the model and functions of velocity averaging, as well as of a graphic user interface, displaying in real time the motion of particles in the CA and the velocity vector field. A global state of the CA is saved in the ALT format [4], and the velocity field is saved in the BMP format. It is possible to vary the probability of single particles deflection from the rectilinear direction according to the given viscosity and other parameters during the simulation process.

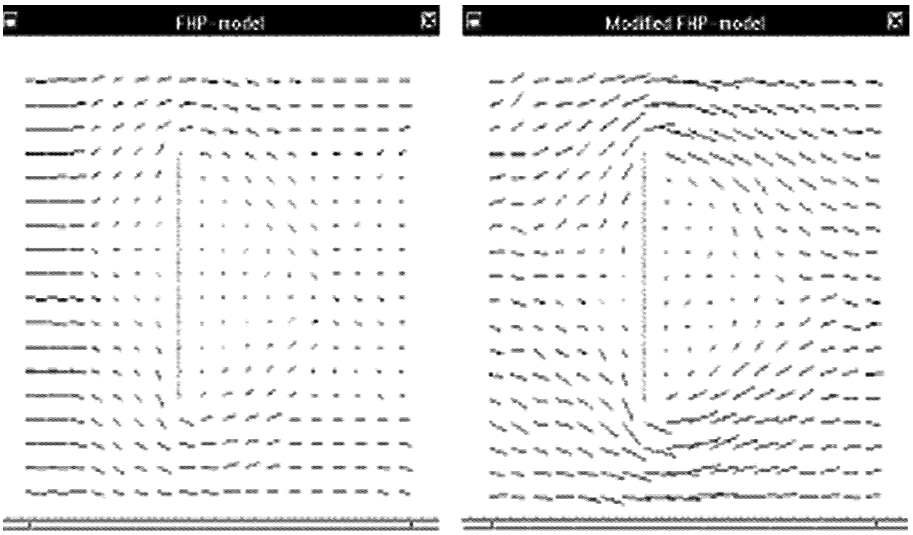

Figure 6. Velocity field of the flow (the FHP model)

One of the experiments, conducted with the FHP models, is the simulation of the fluid flows with different viscosity around a straight wall. This experiment was aimed comparing the velocity fields given by the basic and the modified FHP models. The CA array represented a camera with one wall being a straight source of the even fluid flow, and the opposite one being opened (Figure 6). A thin plane was placed in the middle of the camera across the flow but not reaching the borders of the camera. The basic FHP model generates a laminar flow velocity field (Figure 6a). The modified model generates a velocity field with vortices behind the obstacle (Figure 6b), which shows that the modified model is capable to simulate fluid flows with Reynolds number higher than that simulated by the basic one.

3. A 3D gas-lattice model

No appropriate 3D model has been found. Instead, the pseudo-4D models, whose projections onto the 3D space represent the 3D flows. Using the pseudo-4D models involves a huge amount of calculations (the table size of automaton transition is 2^{27} with 24 neighbors and 3 rest-particles), which cannot be done even on modern supercomputers.

Let us introduce a 3D flow CA model with cells having 12 neighbors (the size of automaton transition table is 2^{13} with 1 rest-particle). It may be used for simulating a large-scale flows on currently available computers.

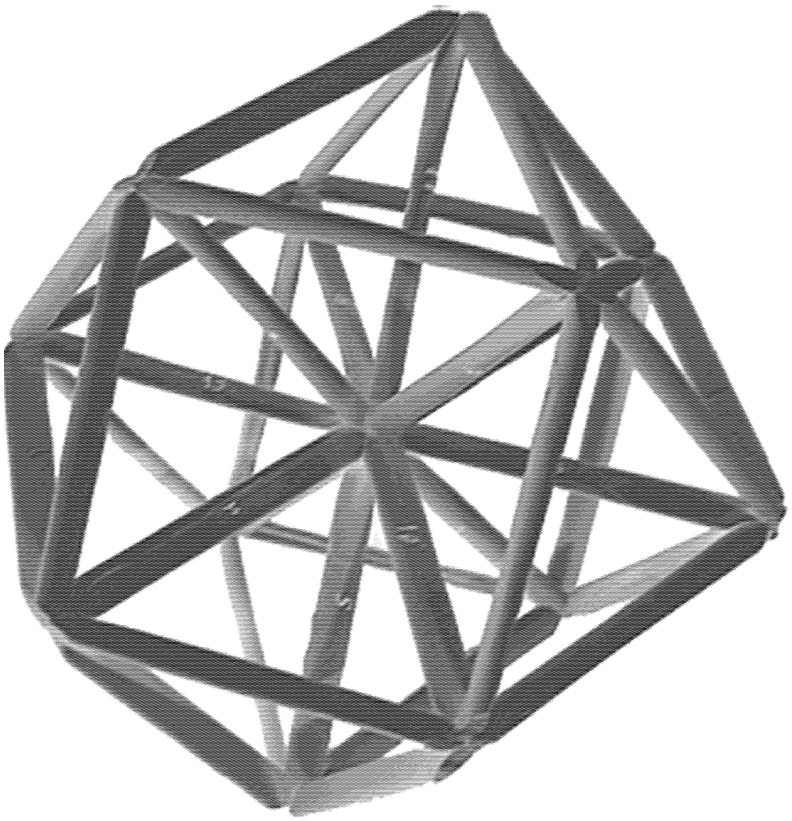

Figure 7. The PCRD model

The automaton array structure is based on the polar complex of the rhombic dodecahedron (PCRD) shown in Figure 7. It is quite simple in realization and sufficiently well describes the 3D flows.

3.1. Cellular automaton structure

Let us dispose the PCRD structure with the cell centers coordinates i, j, k ($0 \le i < l$, $0 \le j < m$, $0 \le k < n$) onto a Cartesian lattice with the coordinates x, y, z. Then the coordinates (x_i, y_j, z_k) of any node (i, j, k) of the PCRD follow the formulas:

$$x_i = i + (k \bmod 2)/2, \quad y_j = j + (k \bmod 2)/2, \quad z_k = (k + (k \bmod 2)/2)\sqrt{2}.$$

If the rhombic dodecahedrons with unitary distance between opposite faces fill a 3D space, then the above node locations correspond to the coordinates of the dodecahedrons centers. Hence, the neighboring cells are those, having a common dodecahedron face. It is interesting to note, that

elements of the lattice with such a spatial location comply with the centers of tightly packed balls.

The directions of particle velocities are determined by the polar complex of the rhombic dodecahedron, which is a collection of beams, coming out from the dodecahedron center perpendicular to its faces. The number of beams is 12, and they pass through the neighboring cell centers, i.e., all the lines connecting the neighboring cell centers comply with the polar complexes of the corresponding dodecahedrons. One node of the lattice is shown in Figure 7 as a cross point of lines inside a cell. It is connected with 12 neighboring nodes by lines. Also, the connections between each pair of the neighboring nodes, each having the neighbor of the center node, are shown.

3.2. Collision rules

There are different versions of collision rules in different modifications of the PCRD models, but in each case, of course, conservation of mass and momentum under each collision must be satisfied. One of the versions is a full set of rules with one rest-particle. Since each cell in the lattice has 12 neighbors and one rest particle, the number of collision rules being 2^{13}. Let us construct all output states for any i-th rule of these 8192 input states of a transition table, total mass and momentum of particles in the cell being not changed. This set of n_i output states ($i = 1, 2, \ldots, 8192$) contains all the collision rules with realization probability equal to $1/n_i$ for each case. There is an experimental proof (by exhaustive search) that the probability of values is the interval from 1 to 56 for different input conditions.

4. The PCRD model program realization

4.1. Software package scheme

A software package consisting of four modules is shown in Figure 8. It has been created for the investigation of the CA models of fluid flows. In the package, the following data structures are used.

- *The CA global state* is a set of states of all the cells, including information of the cell type (working, wall, source cells) and about particles (the number of particles and directions of velocity vectors of each of them). The CA global state is iteratively changed during the experiment.
- *Velocity field projections* consist of three arrays, whose elements represent velocity vectors projections of three Cartesian planes: OXY, OYZ, OXZ.

The software package consists of the following programming blocks.

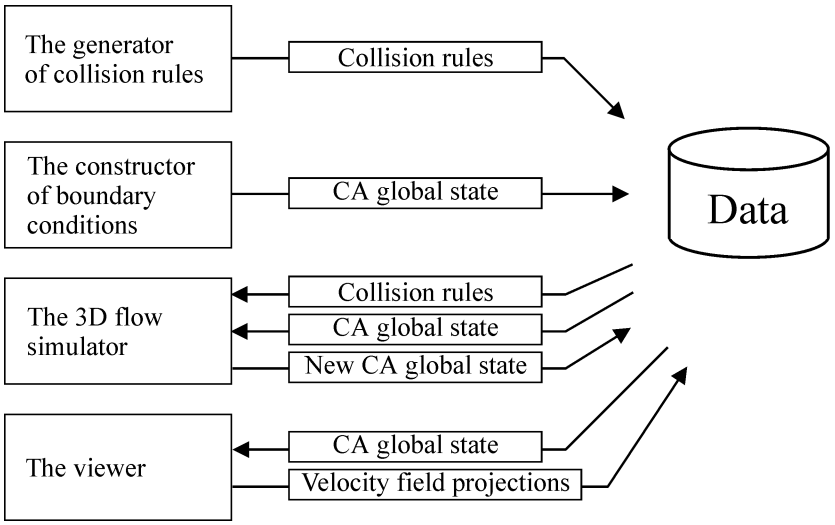

Figure 8. Program package scheme

- *The generator of collision rules* forms a set of the rules, which are used by the fluid flow simulator. The generator of collision rules is run once when the experiment begins.
- *The constructor of boundary conditions* forms an initial global state of the automaton. It is used once in the beginning of the experiment.
- *The 3D flow simulator* converts a current CA global state to a new one using transition rules. It operates iteratively with a given number of cycles.
- *The viewer* computes averaged velocity vectors of particles and constructs a velocity field of the flow. It also displays three orthogonal projections of the field on the computer monitor.

The simulation has been performed as follows. At first, the generator of collision rules and the constructor of boundary conditions create a set of collision rules and the initial CA global state. Then the simulator iteratively executes the CA transition rules. The viewer is run at any time we need to observe the CA evolution.

4.2. Experiments

Experimental investigations of the CA models are a very important factor for confirming or refusing their validation, revealing inaccuracies, indicating ways of their correction. One of the experiments carried out with the PCRD model has been conducted as follows. The constructor of boundary

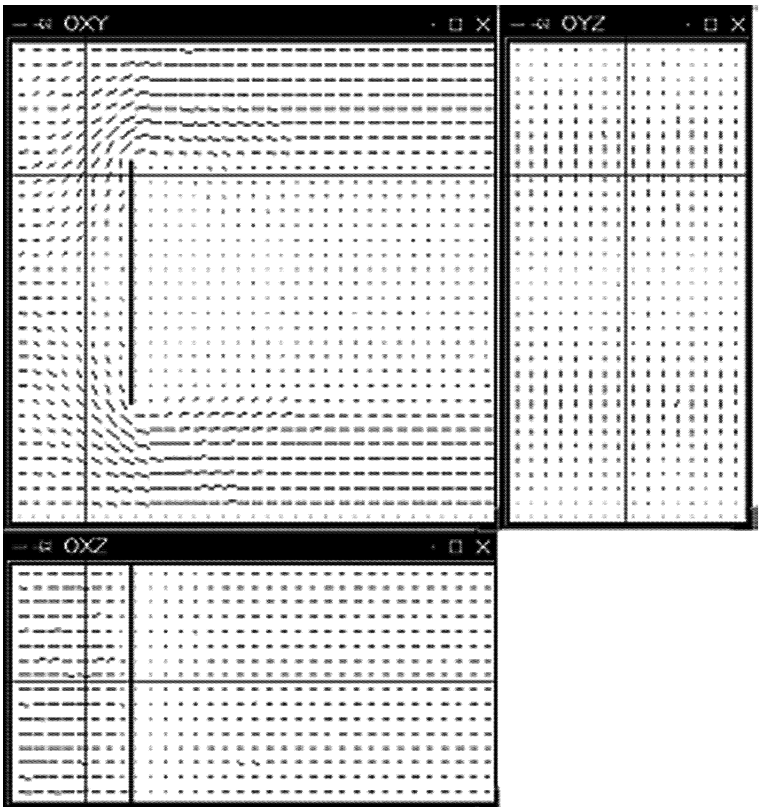

Figure 9. Velocity field of the flow (the PCRD model)

conditions has created an initial CA global state. The wall cells were located along the four faces of parallelepiped, two opposite faces being left opened. The source cells were located along one of the opened sides, its opposite side serving for fluid running out. A flat obstacle was placed inside the parallelepiped across the flow, touching its two opposite faces and not touching two other ones, forming two slots.

A velocity flow field has been obtained after a sufficient number of iterations for the determination of a stationary flow. Three cross-sections of the parallelepiped are shown in Figure 9. They display three projections of the 3D velocity flow field on the sections. The boundary conditions for this experiment have been chosen so that one of the cross-sections of the parallelepiped complied with the previous experiment, carried out for a flat model. It is seen that the velocity field projection on the plane OXY of the PCRD model in Figure 9 is similar to the velocity field of the FHP model in Figure 6a. Thus, the PCRD model in the 3D space possesses the same characteristics as the FHP model in the 2D space, i.e., it can be assumed that it simulates fluid flows, described by the Navier–Stokes equations.

5. Conclusion

The CA models were proposed for simulation not so long ago, but recently they are in importance. Amongst 2D models, which have been recently studied, two basic ones have been distinguished. All the other known models are modifications of the basic ones. One of such modifications is proposed, and the results of its investigation are presented. A new 3D model, referred to as the PCRD is also proposed and studied by simulation. The program package for the experimental investigation is described. The future work will be to find the relations between the PCRD parameters and the physical values of the fluid flow, simulated by it.

References

[1] Medvedev Yu. Discrete methods solving some fluid dynamics problems // Proceedings of OVS conference, Ekaterinsburg, 1998 (in Russian).

[2] Frisch U. et al. Lattice gas hydrodynamics in two and three dimensions // Complex Systems. – 1987. – Vol. 1. – P. 649–707.

[3] Medvedev Yu. Modification of a 2D gas-lattice model // NCC Bulletin, Special issue. – Novosibirsk: NCC Publisher, 1999. – P. 82–87.

[4] Pogudin Yu. Simulation of fine-grained parallel algorithms with the ALT system // Proceedings of First International Workshop on Distributed Interactive Simulation and Real Time Applications. Jan. 9–10, 1997, Eilat. IEEE Computer Society. – 1997. – P. 22–27.