Lattice Gas Automata: FHP Model

Mateusz Bancewicz
student of Physics
at Wroclav University

April 28, 2015

Abstract

The flow of a fluid is described by Navier-Stokes equation.[1]Unfortunately, it can be solved analytically only for small group of problems. One of the solvable problems is Poiseuille flow through a channel. In this project I will use FHP model to get Poiseuille profile of fluid velocity and compare it to the analytic solution. Moreover I will check if the pressure gradient is constant in channel and whether mass and momentum is conserved. Finally, I will investigate a fluid flow behind a plate and try to get von Karman vortex street.

1 Model description

The idea of lattice gas automata (LGA) is that if essential features of real fluid are maintained on microscopic level then the correct physical behavior will be reproduced at a macroscopic scale. LGA models have two basic assumptions:

- Local conservation of particles at each time step
- Local conservation of momentum at each time step

The FHP model is characterized as follows:

- A hexagonal lattice of nodes,
- each node has six (seven in FHP-III) cells representing particles with six different velocities (and rest particle),
- cells can be occupied by at most one particle,
- all particles have the same mass,
- the evolution in time proceeds according to predetermined update rules,
- only particles of single node are involved in collision.

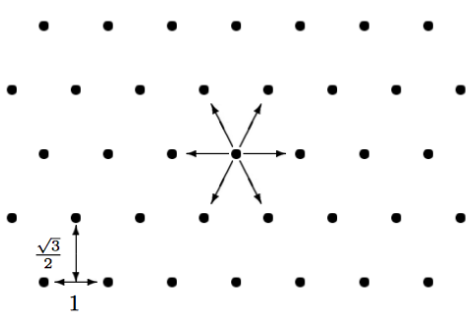

Figure 1: Hexagonal lattice of FHP model with six velocity vectors in each node.[2]

[1] Navier-Stokes equation description
[2]Dieter A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Methods - An Introduction*

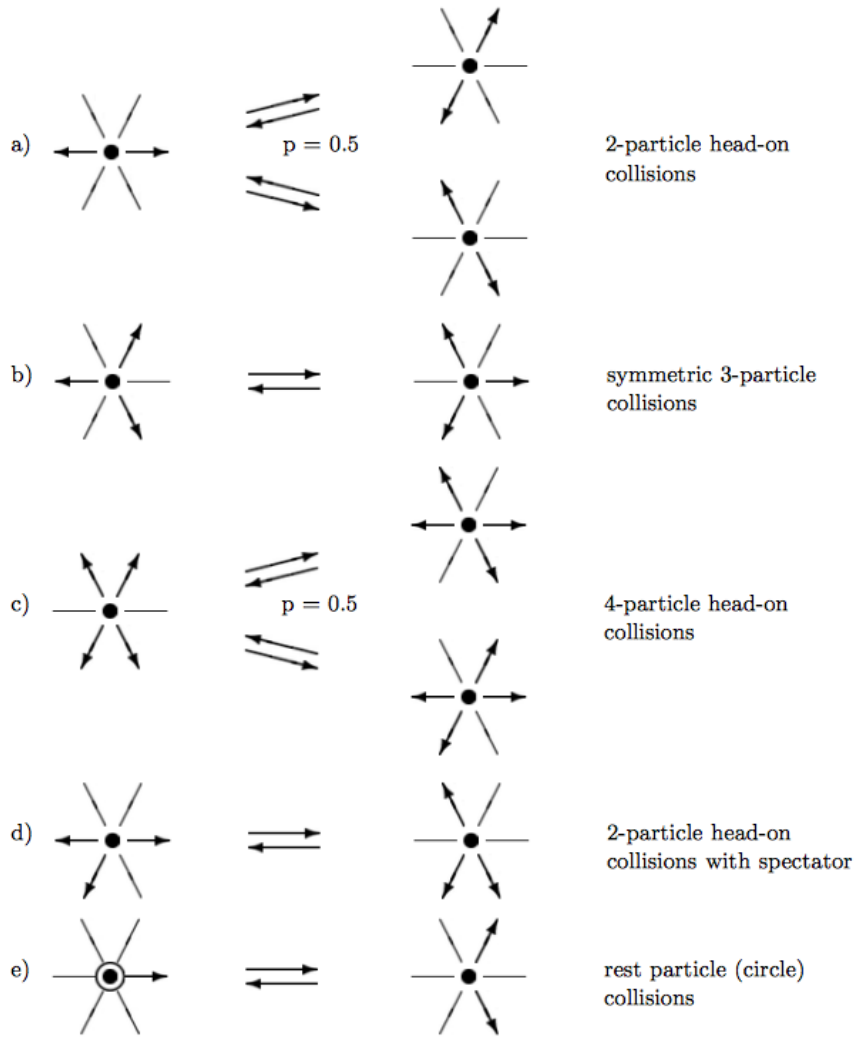

Figure 2: Collision rules in FHP-III model, FHP-I includes only a) and b).[2]

2 Model implementation

To represent each node we need six (FHP-I) or seven (FHP-III) boolean values for each cell. We can use 8-bit **unsigned char** and operate on its bits. The easiest way to implement collisions is to use lookup tables.[3] Index of lookup table represents input state, the value of table stands for output state.

Let us analyze collision a) from Figure 3. We have two particles in cells 0 and 3 as an input state (integer value of this node is $2^0 + 2^3 = 9$) after collision we have particles in cells 1 and 4 (integer value 18) or 2 and 5 (integer value 40). Since there are two possible out state values we need to create two dimensional lookup table, first collisionTable[0][9] = 18, second collisionTable[1][9] = 40. The size of lookup table is 2 x 128. The bounce-back rule on the top and bottom boundaries were implemented as a bit operation

$$n_i = n_{i+3},$$

modulo 6.

FHP-I model has only two types of collision hence I decided to implement it as a single equation of bit operations which determines cells occupation in next time step.

[3] James M. Buick, *Lattice Boltzmann Methods in Interfacial Wave Modelling*

$$
\begin{aligned}
& n_i\left(t+1, \boldsymbol{r}+\boldsymbol{c}_i\right)=n_i(t, \boldsymbol{r}) \wedge \\
& \quad\{[(n_i \wedge n_{i+1})\,\&\,(n_{i+1} \wedge n_{i+2})\,\&\,(n_{i+2} \wedge n_{i+3}) \\
& \quad \&\,(n_{i+3} \wedge n_{i+4})\,\&\,(n_{i+4} \wedge n_{i+5})] \\
& \quad \mid [n_i \& n_{i+3} \& \sim(n_{i+1} \mid n_{i+2} \mid n_{i+4} \mid n_{i+5})] \\
& \quad \mid [\xi \& n_{i+1} \& n_{i+4} \& \sim(n_i \mid n_{i+2} \mid n_{i+3} \mid n_{i+5})] \\
& \quad \mid [\sim \xi \& n_{i+2} \& n_{i+5} \& \sim(n_i \mid n_{i+1} \mid n_{i+3} \mid n_{i+4})]\ \}
\end{aligned}
$$

Figure 3: Bit equation for FHP-I model.[2]

It is convenient to represent hexagonal lattice as a rectangular lattice with redefined links.

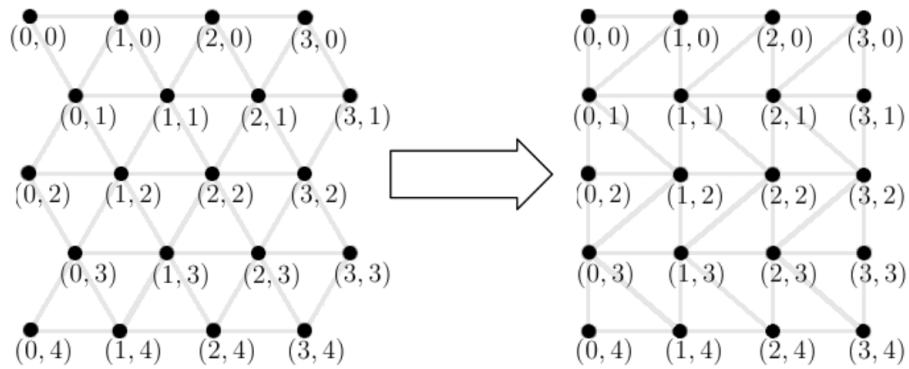

Figure 4: Transition from hexagonal lattice to rectangular lattice.[4]

One can see that nearest neighborhood depends on the line index. To reproduce hexagonal symmetry on rectangular lattice, we must define different links depending on whether line index is even or not (see the code below).

```
//for even lines
directions[0][0] = Int2(1,0);
directions[0][1] = Int2(0,-1);
directions[0][2] = Int2(-1,-1);
directions[0][3] = Int2(-1,0);
directions[0][4] = Int2(-1,1);
directions[0][5] = Int2(0,1);
//for odd lines
directions[1][0] = Int2(1,0);
directions[1][1] = Int2(1,-1);
directions[1][2] = Int2(0,-1);
directions[1][3] = Int2(-1,0);
directions[1][4] = Int2(0,1);
directions[1][5] = Int2(1,1);
```

3 Poiseuille flow

3.1 Analytic solution

Let us assume that we have unidirectional flow in x direction. This implies that $v_y \equiv 0$ and $v_z \equiv 0$. Moreover we are interested in incompressible flows, thus from continuity equation we get

$$\frac{\partial v_x}{\partial x} = 0,$$

[4]Sebastian Szkoda, *Implementacja modelu FHP w technologii NVIDIA CUDA*

so v_x is function of three variables:

$$v_x = v(y, z, t).$$

Taking above into account we can write down Navier-Stokes equation for this problem

$$\frac{\partial v}{\partial t} - \nu \left(\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) = -\frac{1}{\varrho} \frac{dp}{dx} \tag{1}$$

where ν is kinematic viscosity, ϱ is density and $\frac{dp}{dx}$ is pressure gradient.

We are considering steady and laminar flow, since it is two-dimensional case and our flow doesn't change in time $v_x = v(y)$, where y = [0,h] (h is the height of the channel). The flow is sustained by the pressure gradient $\frac{dp}{dx}$. Now our equation is

$$\frac{\partial^2 v}{\partial y^2} = -\frac{1}{\nu \varrho} \frac{dp}{dx}$$

General solution to the above equation is

$$v(y) = \frac{1}{2\nu\varrho} \frac{dp}{dx} y^2 + C_1 y + C_2,$$

if we consider no-slip boundary conditions ($v(0) = v(h) = 0$) we can calculate constants and get the final equation for velocity profile

$$v(y) = -\frac{1}{2\nu\varrho} \frac{dp}{dx} y(h - y),$$

if the pressure gradient is constant in channel we should get parabolic profile of the velocity.

3.2 Simulation results

To reproduce analytic solution I simulated flow through thin channel of size 452 x 110. The measurement starts after 6,000 time steps and lasts for 3000 more. Then all the quantities are averaged in time and in case of velocities also in space. To start and sustain the flow I created two reservoirs (16 x 110) with constant density at the inlet and the outlet. In FHP models pressure p can be calculated as follows $p = c_s^2 \cdot \varrho$, where c_s is speed of sound in particular model ($c_s = \sqrt{\frac{3}{7}}$ in FHP-III). The pressure difference drives the flow in desired direction. First thing to do was to check if the pressure gradient is constant in the channel (we expect pressures linear dependence of x).

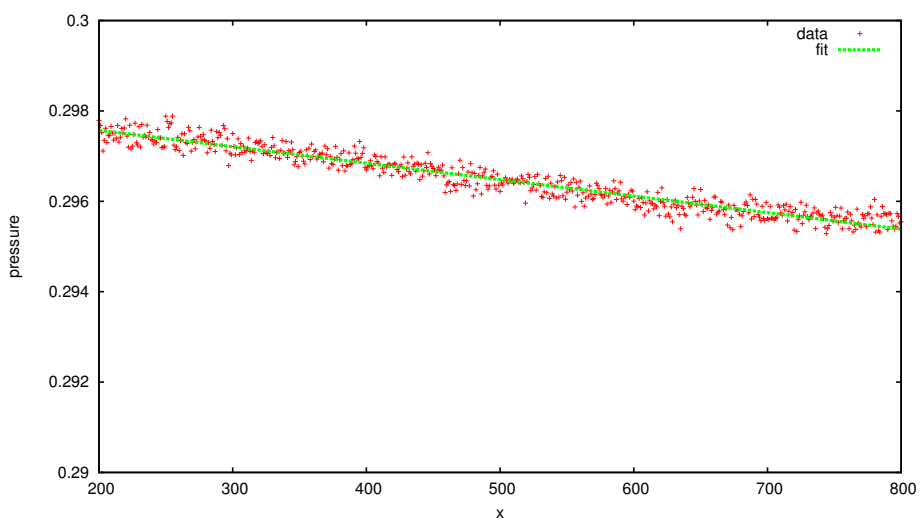

Figure 5: Pressure dependence of x. Density per cell at the inlet set to 0.25 and 0.15 at the outlet. Slope of this function represents pressure gradient.

3.3 Mass and momentum conservation

After relaxation time there are only small fluctuations around equilibrium state of mass and momentum.

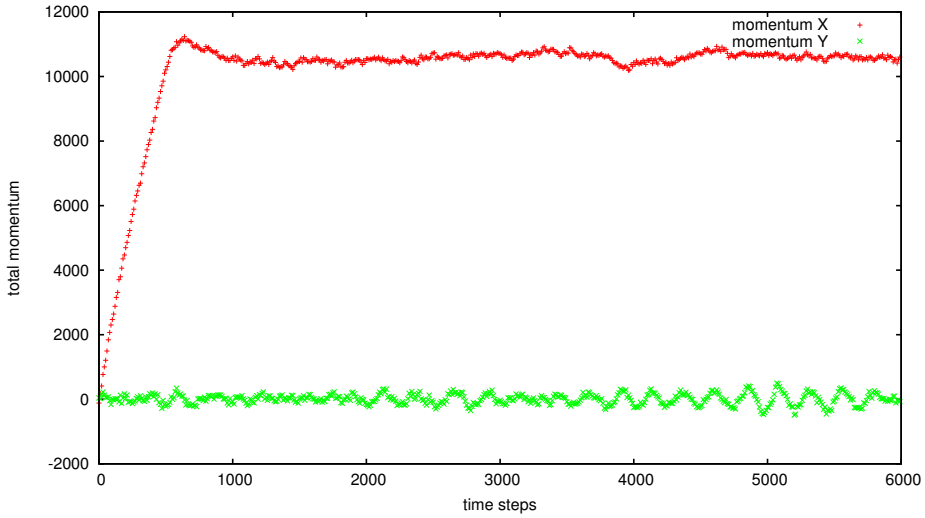

Figure 6: Total momentum (x and y components plotted separetely) of our system in time.

3.4 Poiseuille profile

To get the desired profile I divided my domain to area of size 140 x 12. After averaging velocities in space and time I fitted quadratic function to the obtained data.

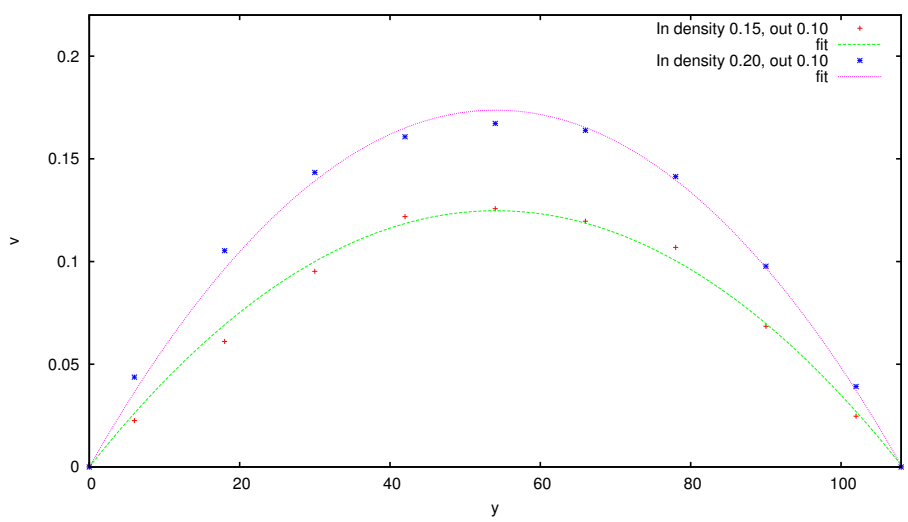

Figure 7: Velocity (x component) dependence of y in a channel of size 420 x 108

One can see that our result is very satisfying, the standard uncertainty of parabolas parameters was around 1,5%.

4 Flow behind a plate

To obtain anything similar to von Karman vortex street[5] we need to increase our domain significantly. Let me introduce a Reynolds number

$$Re = \frac{u \cdot L}{\nu}$$

where u is the velocity at the inlet, L is the characteristic length (height of the plate) and ν is the kinematic viscosity (in FHP models it depends only on density). We need this number to be greater than 70. The easiest way to increase Reynolds number is to have a bigger plate. In my simulations I chose L = 128. Since we need to reduce the effect of boundaries our domain height should be much larger than L, therefore domain width must be greater. This leads to much bigger lattice than in Poiseuille problem.

I simulated flow in domain of size 3072 x 1088. The velocity field was obtained by averaging 32 x 32 blocks of nodes in 1, 2, 4, 16 and 100 time steps. The density per cell at the inlet was 0.30 and 0.10 at the outlet. The x component of velocity at the intlet was 0.25. Kinematic viscosity for obtained mean density per cell (0.1933) is 0.1536, thus the Reynolds number Re = 208.

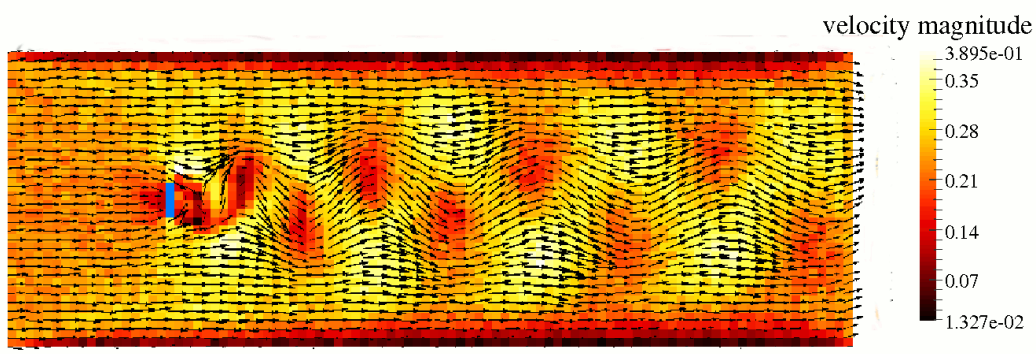

Figure 8: Velocity field after 40,000 time steps

Even though calculated Reynolds numbers is high enough, I did not get the vortex street. Probably the domain of simulation was too small and effects of boundaries disturbed flow. In literature [5] size of domain is 6400 x 3000 (the channel is almost 3 time wider). However one can see characteristic patterns in velocity magnitude field that looks similar to ones from von Karman vortex street. Averaging in different time intervals did not change velocity field significantly. Velocity field on Figure 8 was averaged in two time steps.

[5] D.A Wolf-Gladrow, R. Nasilowski, A. Vogeler *Numerical simulations of hydrodynamics with a pair interaction automaton in two dimensions*

5 Appendix

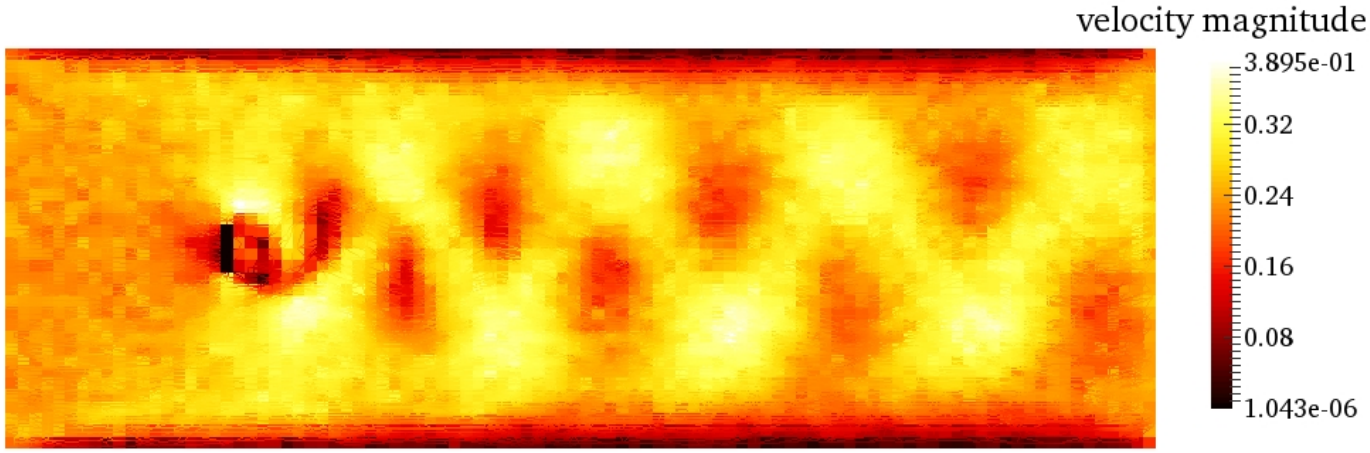

Figure 9: More "artistic" way of visualizing velocity magnitude field (looks better after zooming a little bit).