Flow modelling with lattice gas cellular automata

Mr. S. Schreiber

March 26, 2014

Abstract

The use of lattice in flow modelling has become increasingly popular over the last couple of decades and during that time a lot of models have been developed. We'll be investigating one of the more widely used models i.e. FHP. We will also briefly discuss the HPP model. The following will be discussed for each model: history, assumptions, the collision rules, the propagation rules, calculation of the mass and momentum density, time complexity analysis, applications.

1 Introduction

Lattice-Gas Cellular Automata (LGCA) and lattice Boltzmann methods (LBM) have become widely used for fluid simulations. Lattice-Gas Cellular Automata (LGCA), the predecessor to LBM, is a good starting point to understand how these models function, how they are defined, what they can be used for and why these discrete models lead to the Navier-Stokes equation. In section 2 provides background information on cellular automata and lattice gas cellular automata. In section 3.1 we discuss the HPP model and in section 3.2 we discuss the FHP models.

2 Background

2.1 Cellular automata

Informally a cellular automaton is a system made up of many discrete cells, usually a M by N grid. Each one of these cells can be in one of a finite number of states. Each cell or automaton may only change state on a fixed interval and only according to a fixed set of rules. These rules depend on the cells own value and that of this neighbours. A cellular automaton can formally be defined by

Definition A cellular automaton is a 4-tuple, $C = \{S, s_0, f, G\}$, where

1. S is a finite set of state,
2. s_0 is the initial state,
3. G is the neighbourhood,
4. $f : S^n \to S$.

Two popular neighbourhoods are the von Neumann and Moore. The von Neumann neighbourhood for range r can be defined by

$$G_{(x_0,y_0)} = \{(x, y) : |x - x_0| + |y - y_0| \leq r\}.$$

The Moore neighbourhood for range r can be defined by

$$G_{(x_0,y_0)} = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\}.$$

A popular example of a cellular automata that makes use of a Moore neighbourhood with r = 1 is "Conway's game of life".

2.2 Lattice gas cellular automata

Lattice gas cellular automata is a cellular automata where the transition functions f is split into two phases i.e. propagation phase and collision phase. Also each cell has n number of local sites, where n is the number of directly connected neighbours. Each site is associated with a specific velocity vector $\mathbf{c}_i$ $(i = 0, 1, 2, \ldots, n)$. Velocity vectors are the edge's between cells and their neighbours and is of unit length. Sites may be empty or occupied by at most one particle. This pauli exclusion principle is characteristic for all lattice-gas cellular automata.

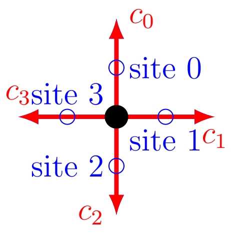

Figure 1: Here is an example of a single cell with 4 velocity vectors and 4 empty sites.

As previously mentioned the transition function f is divided into two phases propagation and collision. The collision phase only applies rules that act on a single cell i.e. the local sites. The propagation phase on the other hand applies rules between cells. What those rules are will vary between implementations and will be discussed later.

3 Lattice-Gas Cellular Automata (LGCA)

3.1 HPP

3.1.1 History

The HPP model was proposed by Hardy, de Pazzis and Pomeau in 1973. This was the first LGCA model and received its name from the initials of the authors. It is the simplest LGCA model and is only mentioned for its historical importance and not for its application because it does not lead to the Navier-Stokes equation in the macroscopic limit, this is due to the insufficient degree of rotational symmetry of the lattice.

3.1.2 Model Description

The HPP model is a LGCA that make use of a square lattice. Each cell has 4 velocity vectors $\mathbf{c}_i$ where $i \in \{0, 1, 2, 3\}$. Each velocity vector is associated with one site, see Fig 1. All particles have the same mass m and are indistinguishable. A site can only be in one of two states occupied or empty. A site can not be occupied by more than one particle (Pauli exclusion principle). It will lead to equilibrium distributions of Fermi-Dirac type for the mean occupation of the cells.The two phase will now be discussed.

3.1.3 Collision phase

This is a basic model and thus only has one collision rule, all other configurations are treated as if no collisions have occurred. The collision rules only consider local sites and only has a local effect. The notation to select the nth site is s_n. If s_i is equals to the opposite site s_j where $j = (i + 2) \bmod 4$ and the other two sites are empty. Then s_k will take the value of s_i and s_d will take the value of s_j where k and d are the indices of the open sites. The sites s_j and s_i are then set to empty.

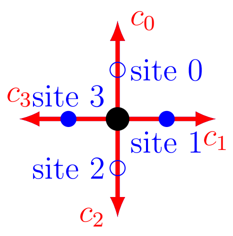

Figure 2: Before Collision

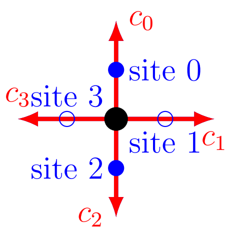

Figure 3: After Collision

3.1.4 Propagation phase

The propagation phase only considers the local neighbours of a cell and not the cell it self. The propagation phase has no effect on the local neighbours of a cell, only on the cell it self. The cell's neighbours will be denoted as n_i where $i \in \{0, 1, 2, 3\}$, the cell it self does not count as a neighbour. Neighbour n_0 is the top neighbour, neighbour n_1 is tot the right, neighbour n_2 is below and neighbour n_3 is tot the left of the cell. The propagation rules work as follow: If the cell's neighbour n_i has a occupied site s_i then the particle in that site moves to the cell's site s_i. A thing to keep in mind while applying the propagation phase is, that it will require two states/matrices one that stores the previous results and one that will store the new results.

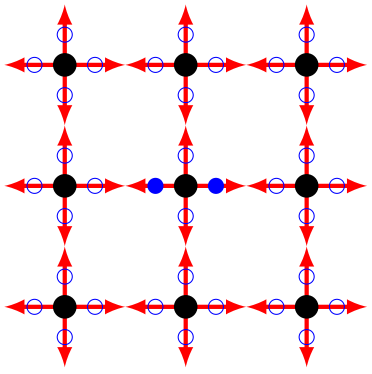
Figure 4: Initial configuration, before collision

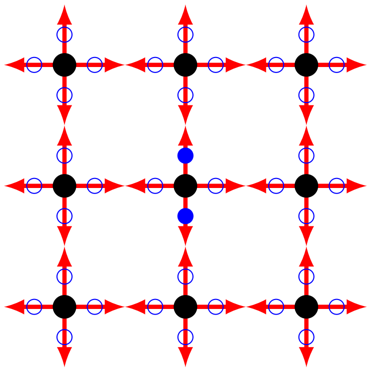
Figure 5: After Collision and before propagation

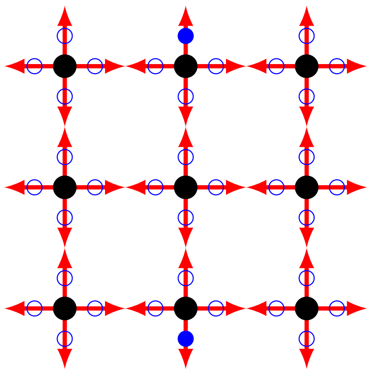
Figure 6: After propagation

3.1.5 Boundary conditions

We will discuss two approaches to handle the boundary condition i.e. wrap around (also known as repeating) and reflection. Both conditions only have an effect on the propagation phase and not on the collision phase. Also when generating a lattice grid and using the reflection boundary condition do not generate particles in the border sites[1] this will create problems.

Wrap around This approach can easily be explained by firstly placing copies of the current lattice all around it self i.e all 4 positions. The center lattice is the original while the other 4 lattices are copies. Secondly by running the propagation phase on the new lattice. All the changes that

[1]Border sites are the sites right next to the border and not all 4 sites of the border cell

happen in the center lattice must then be applied to the original[2] lattice. All of the changes that happen in copies are ignored. This effect can also easily be achieved by making use of the module operator.

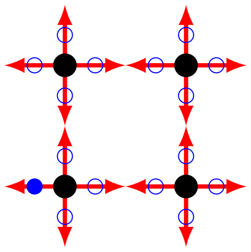
Figure 7: Initial configuration

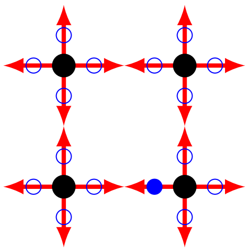
Figure 8: After propagation with repeating boundary condition.

Reflection This is a easy and quick approach to handle the boundary conditions. When a particle collides with the boundary it changes its direction by 180°, note by moving a particle to the apposite site is the same as rotating by 180°. It then propagates in that direction.

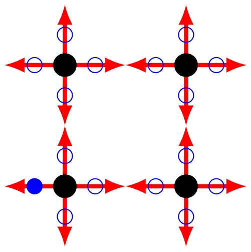
Figure 9: Initial configuration

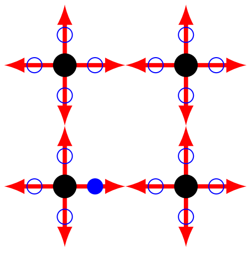
Figure 10: Step one: Turn 180°

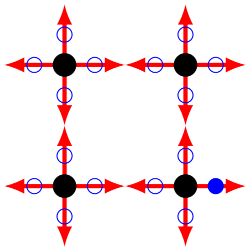
Figure 11: Step 2: Move in that direction.

3.1.6 Mass and Momentum Densities

The LGCA can be fully describe by the boolean fields $n_i(t, \mathbf{r}_j)$, where $i \in \{0, 1, 2, 3\}$, $\mathbf{r}$ is the position of the cell and t is the discrete time. $n_i(t, \mathbf{r}_j)$ means the boolean value of the ith site at position $\mathbf{r}$ at time t. Before calculating the mass and momentum densities we require the mean occupation

[2]What is meant by original lattice is the lattice that was created after the collision phase has executed

numbers. The mean occupation number for site i at position $\mathbf{x}$, is the average of $n_i(t, \mathbf{r}_j)$ where $\mathbf{r}_j$ are the the positions of the neighbours of $\mathbf{x}$

$$N_i(t,\mathbf{x}) = \frac{\sum_{j=0}^{3} n_i(t,\mathbf{r}_j)}{4}.$$

The mass density for time t and position $\mathbf{x}$ is defined as follow:

$$\rho(t,\mathbf{x}) = \sum_{i=0}^{3} N_i(t,\mathbf{x}).$$

The momentum densities for time t and position $\mathbf{x}$ is defined as follow:

$$\boldsymbol{j}(t,\mathbf{x}) = \rho\boldsymbol{u} = \sum_{i=0}^{3} \mathbf{c}_i N_i(t,\mathbf{x}).$$

Given ρ and $\boldsymbol{j}$ the mass and momentum densities we can calculate N_i assuming a linear relationship

$$N_i = \xi\rho + \eta\mathbf{c}_i\boldsymbol{j}.$$

Where ξ and η are both constants and can be calculate as follow:

$$\rho = \sum_{i=0}^{3} N_i$$

$$\rho = \sum_{i=0}^{3} \xi\rho + \eta\boldsymbol{j}\sum_{i=0}^{3}\mathbf{c}_i$$
$$\rho = 4\xi\rho + \eta\boldsymbol{j}\mathbf{0}$$
$$\rho = 4\xi\rho$$

which yields $\xi = \frac{1}{4}$;

$$\boldsymbol{j} = \sum_{i=0}^{3} \mathbf{c}_i N_i$$

$$\boldsymbol{j} = \xi\rho\sum_{i=0}^{3}\mathbf{c}_i + \sum_{i=0}^{3}\eta\mathbf{c}_i(\mathbf{c}_i\boldsymbol{j})$$
$$\boldsymbol{j} = \mathbf{0} + 2\eta\boldsymbol{j}$$

thus $\eta = \frac{1}{2}$ and therefore

$$N_i = \frac{1}{4}\rho + \frac{1}{2}\mathbf{c}_i\boldsymbol{j}.$$

3.1.7 Time complexity analysis

The Time complexity analysis was done for an unoptimized implementation of HHP. It ignores the set-up and boundary costs. Let the lattice size be n by n and the number of occupied sites be l. The collision phase checks each cell to see if a collision has occurred. There are n^2 cells and two configurations for each cell. Thus the collision phase has $4n^2$ comparisons. The propagation phase checks the 4 neighbours cells to see if any particles will propagate to the current cell which also takes $4n^2$ comparisons. Thus the running time of an unoptimized version of HPP for one simulation is $O(n^2)$ and for n simulations is $O(n^3)$.

3.2 FHP

3.2.1 History

The FHP model was proposed by Frisch, Hasslacher and Pomeau in 1986 and received its name from the initials of the authors. This was the first model that showed it is possible for a LGCA to yield the Navier-Stokes equation in the macroscopic limit. This was a big discovery and lead to the rapid development of lattice-gas cellular automata. There where three FHP models developed FHP-I, FHP-II and FHP-III. The main difference between each of the models are the collision rules that are used. Each model contains its predecessors collision rules. So FHP-III contains all of FHP-II's collision rules and FHP-II contains all of FHP-I's collision rules. The FHP model is invariant under rotations by n * 60° modulo 360° (hexagonal symmetry).

3.2.2 Model Description

The FHP model is a LGCA that make use of a hexagonal lattice. Each cell has 6 velocity vectors $\mathbf{c}_i$ where $i \in \{0, 1, 2, 3, 4, 5\}$ with 60° between each vector. Also each velocity vector is associated with a site see Fig 3.2.4. All particles have the same mass m and are indistinguishable. A site can only be in one of two states occupied or empty. A site can not be occupied by more than one particle (Pauli exclusion principle). It will lead to equilibrium distributions of Fermi-Dirac type for the mean occupation of the cells.

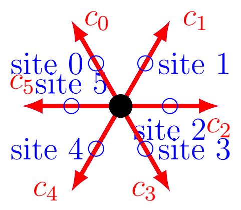

Figure 12: Here is an example of a single hexagonal cell with 6 velocity vectors and 6 empty sites.

The velocity vectors must have a symmetrical property i.e. $\sum_{i=0}^{5} \mathbf{c}_i = \mathbf{0}$. Also each vector should be of unit length.

3.2.3 Collision rules

The collision rules for FHP are a lot more complex than HPP. FHP also introduced non-determinism in some of the rules because if this was not done the model would become chiral. This is an undesired property because the hydrodynamic equations do not break parity symmetry. The 2-particle collisions conserve mass, momentum and also an addition property[3], the same as in HPP. The addition property has no real world counterpart. This is called a spurious invariant. This spurious invariant is eliminated when 3-particle collisions are used. Let s_i and c_i respectively denote the ith site and vector of the current cell where $i \in \{0, 1, 2, 3, 4, 5\}$.

Rule 1 (2-particle head-on collisions):
This collision rule was added in FHP-I. When two opposite sites are occupied and the rest of the sites are empty, then a collision has happened. Now are two configurations to choose from. The choice between the two configurations are done randomly with a probability of 0.5. The rule can be defined as: If

$(s_i = s_{i+3}) \wedge (s_i = 1) \wedge (s_j = 0) \wedge (j \neq i) \wedge (j \neq (i+3))$ for $i, j \in \{0, 1, 2, 3, 4, 5\}$

then with p = 0.5 choose

$$(s_{i-1} := s_i), (s_{i+2} := s_{i+3})$$

or

$$(s_{i+1} := s_i), (s_{i+4} := s_{i+3}).$$

[3]Conserves the difference of the number of particles that stream in opposite direction

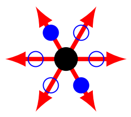

Figure 13: Before collision.

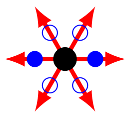

Figure 14: First possible outcome.

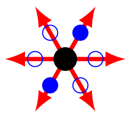

Figure 15: Second possible outcome.

Rule 2 (symmetric 3-particle collisions):
This collision rule was added in FHP-I. When three particles collide at a 120° their directions are rotated by a 180°. The rule can be defined as: If

$$(s_i = s_{i+2} = s_{i+4}) \wedge (s_{i+1} = s_{i+3} = s_{i+5}) \wedge (s_i \neq s_{i+1}) \text{ for } i \in \{0,1,2,3,4,5\}$$

then if $s_i = 0$

$$(s_i := s_{i+1}), (s_{i+2} := s_{i+3}), (s_{i+4} := s_{i+5})$$

else

$$(s_{i+1} := s_i), (s_{i+3} := s_{i+2}), (s_{i+5} := s_{i+4}).$$

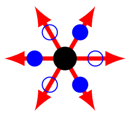

Figure 16: Before collision.

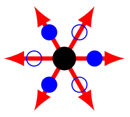

Figure 17: After collision.

Rule 3 (4-particle head-on collisions):
This collision rule was added in FHP-II. This rule is similar to that of rule 1 in the sense that their are two possible outcomes with equal probability and both rules work with head-on collisions. The difference is rule 1 only applies for one head-on collision where rule 3 applies for two head-on collisions that are next to each other. The rule can be defined as follow: If

$$(s_i = s_{i+3}) \wedge (s_i = 1) \wedge (s_{i+1} = s_{i+4}) \wedge (s_{i+1} = 1) \text{ for } i \in \{0,1,2,3,4,5\}$$

then with p = 0.5 choose

$$(s_{i-1} := s_i), (s_i := s_{i+1}), (s_{i+2} := s_{i+3}), (s_{i+3} := s_{i+4})$$

or

$$(s_{i+1} := s_i), (s_{i+2} := s_{i+1}), (s_{i+4} := s_{i+3}), (s_{i+5} := s_{i+4}).$$

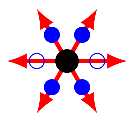
Figure 18: Before collision.

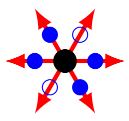
Figure 19: First possible outcome.

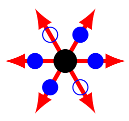
Figure 20: Second possible outcome.

Rule 4 (2-particle head-on collisions with observer): This collision rule was added in FHP-II. This rule is similar to rule 2 to but introduces a observer particle. The placing of this addition observer particle is not import because as previously stated all of the rules are invariant under rotations by n * 60° modulo 360°(hexagonal symmetry), but observer particle and the open site opposite of the observer particle must remain stationary and unchanged during the collision . The two colliding particles move to the two remaining open positions. The rule can be defined as: If

$$(s_i = s_{i+3}) \wedge (s_i = 1) \wedge (s_j = 1) \wedge (j \neq i) \wedge (j \neq (i+3)) \wedge (j \neq k) \wedge (i \neq k) \wedge (k \neq (i+3))$$

$$\text{for } i, j, k \in \{0, 1, 2, 3, 4, 5\}$$

then

$$(s_{k+3} := s_i), (s_k := s_{i+3}).$$

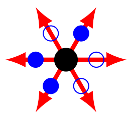
Figure 21: Before collision.

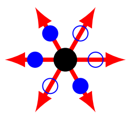
Figure 22: After collision.

Rule 5 (rest particle (circle) collisions):
This collision rule was added in FHP-III. This collision rule introduces rest particles into a LGCA. We will not be using this rule or discuss it because I do not agree that the rule conserves momentum.

3.2.4 Propagation rules

The propagation phase only considers the local neighbours of a cell and not the cell it self. The propagation phase has no effect on the local neighbours of a cell, only on the cell it self. The cell's neighbours will be denoted as n_i where $i \in \{0, 1, 2, 3, 4, 5\}$, the cell it self does not count as a neighbour. Neighbour n_0 is the top left neighbour and continues clockwise. The propagation rules work as follow: If the cell's neighbour n_i has a occupied site s_i then the particle in that site moves to the current cell's site s_i.

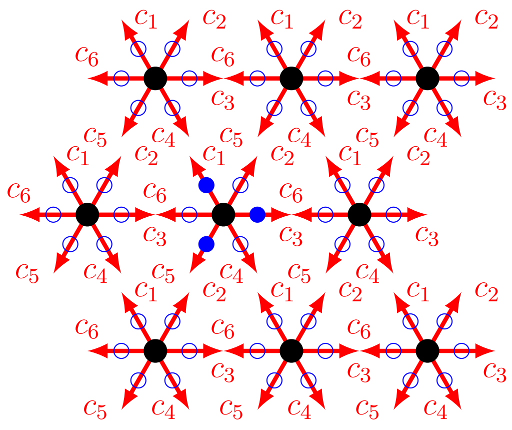

Figure 23: Before collision phase

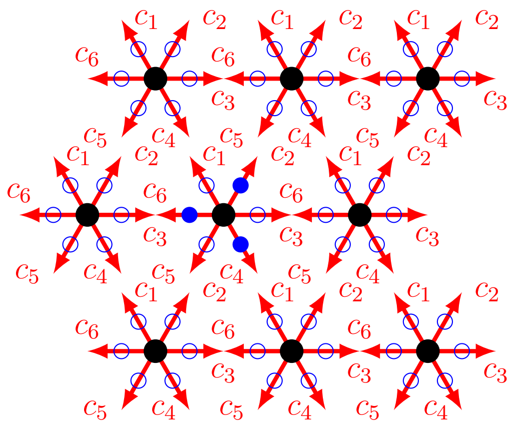

Figure 24: After collision phase and before propagation phase.

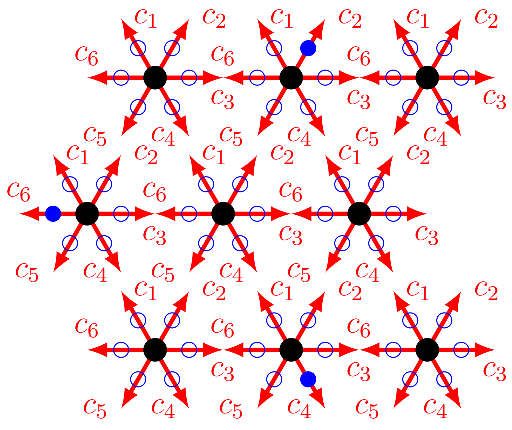

Figure 25: After propagation phase.

3.2.5 Time complexity analysis

The Time complexity analysis was done for an unoptimized implementation of FHP-II. This is just to give a basic idea of the possible time complexity and is not the optimal solution. Assume the size of the lattice is n by n and that initialization and border collision costs are ignored. The number of comparisons that are made in the collision phase is $16n^2$. The number of comparisons that are made in the propagation phase is $6n^2$. Thus the running time of an unoptimized version of FHP-II model for one simulation is $O(n^2)$ and for n simulations is $O(n^3)$

3.2.6 Applications

- Simulating rain in games.
- Simulating rivers in games.
- 2D fluid simulations.
- Network congestion simulation.

3.2.7 defects

The hexagonal shape is hard to work with. There is no way to detect all of the spurious invariants of a given lattice-gas cellular automata model. It was show that all FHP models have non-local spurious invariants.

References

[1] Wolf-Gladrow, D. A. 2000. Lattice-gas cellular automata and lattice Boltzmann models. New York: Springer.