3. Lattice-gas cellular automata

3.1 The HPP lattice-gas cellular automata

The first lattice-gas cellular automata (LGCA) was proposed in 1973 by Hardy, de Pazzis and Pomeau. It is named HPP after the initials of the three authors. The HPP model is the simplest[1] *LGCA which will be discussed in some length. Today HPP is of interest mainly for historical reasons*[2] *because it does not lead to the Navier-Stokes equation in the macroscopic limit. In the current chapter specific coding techniques for lattice-gas cellular automata*[3] *like multi-spin coding will be introduced.*

3.1.1 Model description

HPP is a two-dimensional lattice-gas cellular automata model over a square lattice. The vectors $\boldsymbol{c}_i$ ($i = 1, 2, 3, 4$) connecting nearest neighbors (compare Fig. 3.1.1) are called *lattice vectors* or *lattice velocities*. More precisely, the lattice velocities are given by the lattice vectors divided by the time step Δt which is always set equal to 1. So lattice vectors and lattice velocities have different dimensions but the same numerical values. The meaning of the $\boldsymbol{c}_i$ can be easily recognized from the respective context. At each *site* (*node*) there are four cells (Fig. 3.1.2) each associated to a link with the nearest neighbor. These cells may be empty or occupied by at most one particle. This *exclusion principle* (*Pauli principle*) is characteristic for all lattice-gas cellular automata. It will lead to equilibrium distributions of *Fermi-Dirac*[4]

[1] The model of Boghosian and Levermore (1987) for Burger's equation in one spatial dimension will not be discussed here.

[2] There are still some applications: Chopard and Droz (1991) use HPP as a random generator.

[3] There are only few papers which give hints to specific coding techniques for lattice-gas cellular automata; see, for example, Kohring (1991), Wolf-Gladrow and Vogeler (1992), and Slone and Rodrigue (1997).

[4] Fermi-Dirac distributions are well known from quantum mechanics. Particles with half-odd-integer spins like the electron, proton, or neutron are called fermions; they obey Fermi-Dirac distributions.

D.A. Wolf-Gladrow: LNM 1725, pp. 39–138, 2000.

type for the mean occupation of the cells. All particles have the same mass m (which will be set to 1 for simplicity) and are indistinguishable.

The evolution in time is deterministic and proceeds as an alternation of local *collisions* $\mathcal{C}$ (only particles at the same node are involved) and *streaming* $\mathcal{S}$ (also called *propagation*) along the appropriate links to the nearest neighbors. The *evolution operator* $\mathcal{E}$ is defined as the composition of collision and streaming:

$$\mathcal{E} = \mathcal{S} \circ \mathcal{C}. \tag{3.1.1}$$

To each particle a momentum of magnitude $m\boldsymbol{c}_i$ is assigned. The collision should conserve mass and momentum while changing the occupation of the cells. For HPP there is only one collision configuration. When two particles enter a node from opposite directions and the other two cells are empty a *head-on collision* takes place which rotates both particles by 90° in the same sense (compare Fig. 3.1.3). All other configuration stay unchanged during the collision step. In passing we note that twofold application of the collision operator leads back to the initial configuration:

$$\mathcal{C}^2 = \mathcal{I}, \tag{3.1.2}$$

where $\mathcal{I}$ is the *identity operator.*

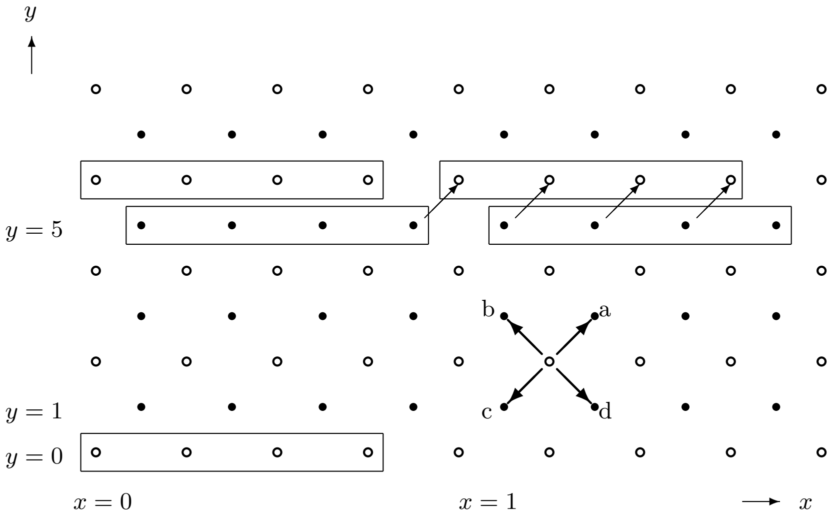
The HPP model respects a particle-hole symmetry, i.e. the operator $\mathcal{F}$ - which interchanges particles and holes - commutes with the evolution operator $\mathcal{E}$. As a consequence of this symmetry the model has similar properties at low and corresponding high mass densities (Hardy et al., 1973).

At each time step particles are interchanged between the sub-lattice consisting of points with even indices (the 'white' sub-lattice) and the sub-lattice consisting of points with odd indices (the 'black' sub-lattice; imagine a chessboard). Therefore there exist two decoupled particle populations on the lattice. This decoupling is characteristic for the square lattice (compare the decoupling of solutions at even and odd time steps - sometimes called the *chessboard instability* - in finite differences; see, for example, Orszag, 1971 or Rood, 1987).

As already noted above the HPP model does not obey the desired hydrodynamic equations (Navier-Stokes) in the macroscopic limit. We will prove later on that this deficit is due to the insufficient degree of rotational symmetry of the lattice. Certain tensors composed of products of the lattice velocities - so-called *lattice tensors* - are not isotropic over the square lattice. See Section 3.3 for an extensive discussion of these tensors. This anisotropy would manifest itself, for example, in the flow past a non-rotational symmetric obstacle in that the drag depends on the relative orientation of the obstacle with respect to the lattice.

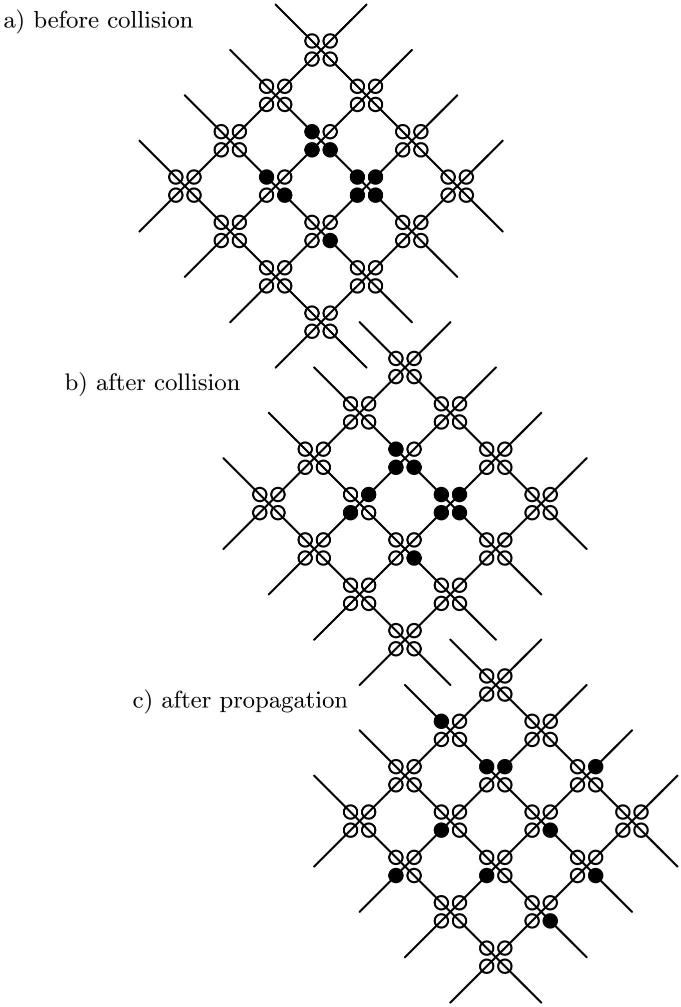
In addition to mass and momentum there exist additional conserved quantities for the HPP model. For example, the difference in the number of particles parallel and anti-parallel to a lattice axis does not change by collisions

Fig. 3.1.1. *The square lattice of the HPP model. The four arrows labelled by a, b, c, and d indicate the lattice velocities* $\mathbf{c}_i$*. Particles are interchanged between the black (small points) and the white (small circles) sub-lattices (chess board).*



or propagation. These '*spurious invariants*' are undesirable because they restrict to a certain degree the dynamics of the model and have no counterpart in the real world.

Fig. 3.1.2. *HPP: collision and propagation. Filled circles denote occupied cells and open circles empty cells. a) Part of the lattice before collision (e.g. after propagation); there is only one collision configuration (two particles in opposite cells at the same node; on the left). b) After collision (e.g. before propagation): the configuration of the cells at node on the left side has changed. c) After propagation: all particles have moved along the links to their nearest neighbors (the lattice outside the part shown was assumed to be empty, e.g. no propagation of particles from 'outside').*

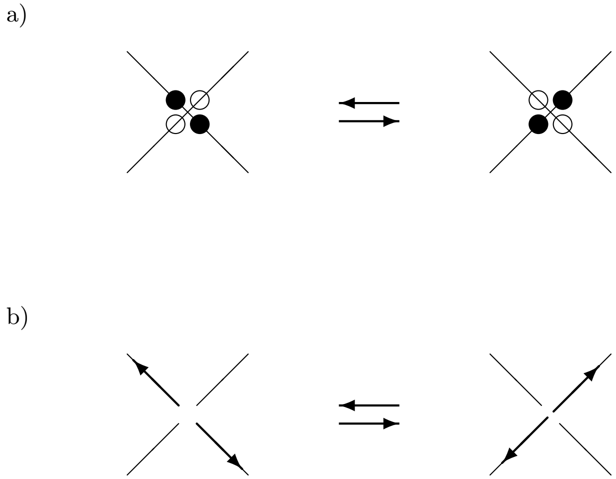


a) before collision

b) after collision

c) after propagation

Fig. 3.1.3. *HPP: collisions. a) There is only one collision configuration (head-on collision) for HPP: two cell on opposite links are occupied and the two other cells are empty. After collision the formerly empty cells are occupied and vice versa. b) Same as a) but showing the associated momentum vectors. Both momentum vectors are rotated by* 90°. *Mass and momentum are conserved.*



3.1.2 Implementation of the HPP model: How to code lattice-gas cellular automata?

The coding techniques discussed here will be also applicable to other LGCA like FHP or PI. The FCHC model requires a different approach (discussed in Section 3.5).

FORTRAN or C? Which programming language is best suited for the coding of LGCA? Of course there will be no unique answer to this question and often discussions with various people resemble religious controversies[5]. I propose the following criteria:

– The computer codes should be portable, i.e. programming in machine specific language (Assembler) is excluded.

– The computation demand of hydrodynamic problems is usually large. Therefore the model should be written in a language for which optimizing compilers (vectorization, parallelization) are available on super-computers. Thus BASIC or PASCAL are excluded and one has to choose between C or FORTRAN.

The specific coding techniques for LGCA can be applied in C as well as in FORTRAN. Comparison of computation time shows only a very small advantage of C (Wolf-Gladrow and Vogeler, 1992). The code of the collisions is much easier to grasp in C than in FORTRAN. Nevertheless, the 'translation' from C to FORTRAN is straightforward (compare Table 3.1.1).

Multi-spin coding. The most important technique for LGCA is *multi-spin coding*[6]. The exclusion principle makes it possible to describe the state of a cell by one bit which is set to 0 if the cell is empty and to 1 if it is occupied. There is no special data type for bits either in C nor in FORTRAN. However, several bits (32 on Sun-Workstation; 64 on CRAY-J90) can be packed into one unsigned (C) or integer (FORTRAN) variable. In standard C there exist bit-operators which act bitwise[7] on whole unsigned variables. In FORTRAN you may find bit-functions with the same effects. These bit-functions do not belong to the FORTRAN standard but are available on almost all machines.

The core of the HPP program, namely the coding of collision and streaming, encompasses only a few lines. Here is the code in C:

[5] I was taught that 'FORTRAN is a dead language' already in the 70ies.

[6] Although the term 'multi-spin coding' which is related to the spins of Ising models has been coined by Creutz et al. in 1979, this technique has been described already in the article of Hardy et al. (1976) and it was first mentioned by Friedberg and Cameron (1970).

[7] Example: decimal 65 | 39 reads in binary notation on an 8-bit machine (01000001) | (00010111) = (01010111) and therefore 65 | 39 = 103.

Table 3.1.1. *Some elements and constructions in C and FORTRAN.*

C	FORTRAN	Remarks
`a & b`	`iand(a,b)`	bit operators are
`a | b`	`ior(a,b)`	standard in C;
`a ^ b`	`ieor(a,b)`	bit functions are
`~a`	`not(a)`	(almost always available)
		extensions in FORTRAN
`a<<3`	`ishift(a,3)`	left shift of bits by
		3 positions
`a>>4`	`ishift(a,-4)`	right shift
`a = a | (1<<3)`	`a = ibset(a,4)`	set 4th bit
`#define A 5`	`parameter (A=5)`	in C: global define
`unsigned a[3]`	`integer a(3)`	1D array with 3 elements
`a[0],a[1],a[2]`	`a(1),a(2),a(3)`	elements
`for(i=0;i<3;i++){`	`do i=1,3`	loops
`}`	`enddo`	
`a % b`	`mod(a,b)`	a modulo b

Table 3.1.2. *Bit-operators in C*: & *and;* | *inclusive or;* $\wedge$ *exclusive or;* $\sim$ *not.*

a	b	a & b	a | b	a $\wedge$ b	$\sim$ a
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

```
/* ---  grid   ---

    b         a
     \       /
      \     /
       \   /
         +
       /   \
      /     \
     /       \
    c          d              */

    last = LENGTH - 1;    /* --- shift last bit --- */
    XMAX1 = XMAX - 1;
    YMAX1 = YMAX - 1;

                       /* -----   collision   ----- */
for(x=0; x<XMAX; x++)
for(y=0; y<YMAX; y++) {
change = ( (a1[x][y] & c1[x][y] & ~(b1[x][y] | d1[x][y])) |
           (b1[x][y] & d1[x][y] & ~(a1[x][y] | c1[x][y])) )
          & nsb[x][y];
a2[x][y] = a1[x][y] ^ change;
b2[x][y] = b1[x][y] ^ change;
c2[x][y] = c1[x][y] ^ change;
d2[x][y] = d1[x][y] ^ change; }

/* -----   propagation in a-direction   ------ */

for(x=1; x < XMAX1; x++) {
for(y=1; y < YMAX1; y += 2) {

/*  note: brackets are necessary,
    because in C the + has a higher priority than the >>  */

/* black to white: */
a1[x][y]  = (a2[x][y-1] >> 1) + (a2[x-1][y-1] << last);
a1[x][y+1] = a2[x][y]; }}      /* white to black: */
```

A few comments on the code are now in order:

- Each unsigned variable can store *LENGTH* (= 64 on CRAY computers) bits. The number of grid points is *XMAX*LENGTH* in x-direction and *YMAX* in y-direction (compare Fig. 3.1.1).

- The states of all cells are stored in the two-dimensional (2D) arrays *a1, b1, c1, d1*, where a, b, c, d assign the different lattice directions (lattice velocities). *a1[0][0]* contains the 'a-bits' of the nodes from 1 to *LENGTH* of the first line.
- The location of obstacles is stored in the 2D array *nbs* (non-solid bit): bits in *nbs* are set to 1 outside of the obstacles and 0 otherwise.
- In the first loop a variable *change* is calculated. It contains the information whether or not a collision will happen: the bits in *change* are set to 1 if the configurations $(a, b, c, d) = (1010)$ or (0101) are present and the node is located outside of obstacles.
- Subsequently, the bit arrays $a1, b1, c1, d1$ are concatenated by 'exclusive or' with the variable *change*. This changes the state bits (interchange of 0 and 1; compare Table 3.1.2). The results of this operation are stored in the auxiliary arrays $a2, b2, c2, d2$. The introduction of these auxiliary arrays ensures a very fast updating on vector computers. These arrays are also useful in the propagation step.
- In Fig. 3.1.1 the propagation is shown only for the a-directions (upwards and to the right). In the program listing the propagation in a-direction is shown only for the inner nodes. The propagation for nodes on the boundaries has to be treated separately according to the appropriate boundary conditions.
  - The propagation from the white to the black sub-lattice consists of a storage in different arrays.
  - The propagation from the black to the white sub-lattice is more involved. To simplify the following discussion let us assume that number of bits per integer is only 4 (the parameter *LENGTH* in the code). The propagation of *a2[0][5]* and *a2[1][5]* to *a1[1][6]* (compare Fig. 3.1.1) will be considered. First all bits of *a2[1][5]* will be shifted to the right whereby the rightmost bit drops out. The resulting void at the left boundary of *a2[1][5]* will be filled up automatically by a 0. Yet, this position must be occupied by the rightmost bit of the neighbor element *a2[0][5]*. To isolate this bit all bits in *a2[0][5]* are shifted to the left by *LENGTH*-1 (= 'last' in the code) digits whereby zeros fill up from the right. Now *a2[1][5]* after a right shift and *a2[0][5]* after *LENGTH*-1 left shifts could be concatenated by exclusive or. But the addition of these two shifted integers yields the same result and is faster on some computers due to chaining[8] (compare Kohring, 1991 and Wolf-Gladrow and Vogeler, 1992).
    Example: *LENGTH* = 4; *last* = 3;

[8] Chaining is the process of passing the output of one vector operation directly as input into another vector operation. As soon as the first element of the first operation's result is output, the second operation can begin. This allows partial overlapping of vector instruction execution.

a2[0][5] = (1001); *a2[1][5]* = (1011);
a2[1][5] >> 1 = (0101); *a2[0][5]* << *last* = (1000)
a1[1][6] = (1101)

3.1.3 Initialization

Before the evolution of the LGCA can start the various arrays have to be initialized. At time $t = 0$ the bits are set by random processes with probabilities such that the mean values over a large number of nodes (typically 32 times 32 or 64 times 64) approximate the given initial values for mass and momentum density. Thus the question arises, how to choose appropriate probabilities for given mass and momentum density?
The state of the LGCA is fully described by the Boolean fields $n_i(t, \boldsymbol{r}_j)$ where the index i which runs from 1 to 4 (or alternatively from a to d) indicates the directions, n_i is the *occupation number* which may be 0 or 1, t is the (discrete) time and $\boldsymbol{r}_j$ are the coordinates of the nodes. Mean occupation numbers N_i are calculated by averaging over neighboring nodes

$$N_i(t, \boldsymbol{x}) = \langle n_i(t, \boldsymbol{r}_j) \rangle \, . \tag{3.1.3}$$

The mean occupation numbers can take on values between 0 and 1. Mass $\rho(t, \boldsymbol{x})$ and momentum density $\boldsymbol{j}(t, \boldsymbol{x})$ are defined by

$$\rho(t, \boldsymbol{x}) = \sum_{i=1}^{4} N_i(t, \boldsymbol{x}) \tag{3.1.4}$$

and

$$\boldsymbol{j}(t, \boldsymbol{x}) = \rho \boldsymbol{u} = \sum_{i=1}^{4} \boldsymbol{c}_i N_i(t, \boldsymbol{x}) \tag{3.1.5}$$

($\boldsymbol{u}$ is the flow velocity) with the lattice velocities $\boldsymbol{c}_i$

$$\begin{aligned} \boldsymbol{c}_1 &= \frac{1}{\sqrt{2}}(1, 1) & (3.1.6)\\ \boldsymbol{c}_2 &= \frac{1}{\sqrt{2}}(-1, 1) \\ \boldsymbol{c}_3 &= \frac{1}{\sqrt{2}}(-1, -1) & (3.1.7)\\ \boldsymbol{c}_4 &= \frac{1}{\sqrt{2}}(1, -1) \end{aligned}$$

which obey

$$\sum_{i=1}^{4} \boldsymbol{c}_i = 0 \quad \text{(lattice symmetry!)} \tag{3.1.8}$$

and

$$\sum_{i=1}^{4} c_{i\alpha} c_{i\beta} = 2\delta_{\alpha\beta}, \tag{3.1.9}$$

where the Latin indices refer to the lattice vectors and run from 1 to 4 whereas the Greek indices assign the cartesian components of the vectors and therefore run from 1 to 2. This convention will be used also in all other chapters.

The theoretical background for the calculation of the equilibrium occupation numbers will be developed not until the next chapter (the section on the FHP model contains some results relevant for HPP). Instead a simple ansatz for N_i will be made which is linear in ρ and $\boldsymbol{j}$

$$N_i = \xi\rho + \eta \boldsymbol{c}_i \boldsymbol{j}. \tag{3.1.10}$$

The coefficients ξ and η can be calculated from the constraints (3.1.4) and (3.1.5):

$$\begin{aligned} \rho &= \sum_{i=1}^{4} N_i \\ &= \sum_i \xi\rho + \eta \boldsymbol{j} \underbrace{\sum_i \boldsymbol{c}_i}_{=0} \\ &= 4\xi\rho \end{aligned} \tag{3.1.11}$$

which yields $\xi = 1/4$;

$$\begin{aligned} \boldsymbol{j} &= \sum_{i=1}^{4} \boldsymbol{c}_i N_i \\ &= \xi\rho \underbrace{\sum_i \boldsymbol{c}_i}_{=0} + \sum_i \eta \boldsymbol{c}_i (\boldsymbol{c}_i \boldsymbol{j}) \\ &= 2\eta \boldsymbol{j} \end{aligned} \tag{3.1.12}$$

thus $\eta = 1/2$ and therefore

$$N_i = \frac{\rho}{4} + \frac{1}{2} \boldsymbol{c}_i \boldsymbol{j}. \tag{3.1.13}$$

I.e., for $\boldsymbol{j} = 0$ the occupation numbers are independent of direction (they can still depend on location which corresponds to pure density perturbations)

whereas non-vanishing momenta imply occupation numbers which vary with direction.
The Boolean arrays n_i will be initialized with probabilities[9] p_i such that the N_i give the desired distributions of ρ and $\boldsymbol{j}$ when summed up according to Eqs. (3.1.4) and (3.1.5). The relations between the Boolean arrays n_i and the mass and momentum density are illustrated in Fig. 3.1.4.

Fig. 3.1.4. *Relations between microscopic (Boolean arrays n_i) and macroscopic (mass and momentum density) level.*

$$\begin{tikzcd}
\underbrace{n_i}_{\substack{\in \{0,1\} \\ \text{Boolean} \\ \text{occupation number}}} \arrow[r, shift right, "\text{averaging}"'] & \underbrace{N_i}_{\substack{\in \mathbb{R} \\ \text{mean} \\ \text{occupation number}}} \arrow[l, shift right, "\text{throw dice}"'] \arrow[r, Leftrightarrow, "\text{definition}"] & \underbrace{\substack{\rho \\ \mathbf{j}}}_{\substack{\text{mass and} \\ \text{momentum} \\ \text{density}}}
\end{tikzcd}$$

Exercise 3.1.1. (*)
Construct occupation numbers N_i which vary with direction but yield $\boldsymbol{j} = 0$.

Exercise 3.1.2. (**)
How long does it take for the distribution (3.1.13) to relax toward equilibrium distribution? What does the relaxation time constant depend on?

3.1.4 Coarse graining

The calculation of mean values for mass and momentum density is called coarse graining. Although it is possible to average over space, time or a combination of space and time, spatial coarse graining is much faster than the

[9] Unfortunately there is no standard for random generators. Portable random generators can be found for example in ‘Numerical Recipes’ (Press et al., 1992a,b). Different types of random generators (multiplicative congruential, shift-register and lagged Fibonacci) are discussed by Slone and Rodrigue (1997).

other alternatives. For the purpose of coarse graining the domain is divided into a number of subdomains which are large enough (usually 32 times 32 or 64 times 64 nodes) to obtain reliable (low noise) averages and small enough as to allow a large 'physical domain' under the constraint of a given limit of core memory.

The following more technical notes can be skipped in a first reading. The main computational load is the counting of the 1-bits in the unsigned (integer) arrays. On some computers a fast routine for counting the 1-bits in an integer is available. On the CRAY, for example, this is the FORTRAN function POPCNT (population count). Nothing similar is available in C. But one can use the FORTRAN function in C. Include the following lines into the code

```
#include <fortran.h>
fortran int POPCNT();    /* counts the number of 1-bits
                            in a 64 bit word */
```

and apply POPCNT to unsigned variables:

```
int n;
unsigned u;

u = 7;
n = POPCNT(u);
```

If no machine specific population count is available one can use the following routine which applies a *look-up table* (see Kohring, 1991, for the FORTRAN version):

```
/* ------- popcount for 32-bit unsigned ------ */

unsigned lu16[65536];   /* global */

void makelu16()
{
/* ----- make look-up table ----- */
   int ilu,ibi;
   for(ilu=0; ilu<65536; ilu++) {
      lu16[ilu] = 0;
      for(ibi=0; ibi<16; ibi++) {
         if( (ilu & bits[ibi]) > 0) lu16[ilu] += 1;
      }
   }
}   /* ---   end of makelu16   --- */

int popcount(unsigned u)
{
/* --- count bits --- */
   int pop;
   pop = lu16[(u&65535)] + lu16[( (u>>16)&65535 )];
   return(pop);
}   /* --- end of popcount --- */
```

3.2 The FHP lattice-gas cellular automata

In 1986 Frisch, Hasslacher and Pomeau showed that a lattice-gas cellular automata model over a lattice with a larger symmetry group than for the square lattice yields the incompressible Navier-Stokes equation in the macroscopic limit. This model with hexagonal symmetry is named FHP according to the initials of the three authors. The discovery of the symmetry constraint was the start for a rapid development of lattice-gas methods. The theoretical foundations where worked out by Wolfram (1986) and by Frisch et al. (1987). Within the following years many extensions and generalizations (FCHC for 3D simulations, colored models for miscible and immiscible fluids) were proposed. These models allow a wide range of applications.

After concentrating more on coding techniques in the HPP chapter the focus of the current section will be on the *theory* of lattice-gas cellular automata. Especially the equilibrium distribution and the macroscopic equations will be derived.

3.2.1 The lattice and the collision rules

The FHP lattice is composed of triangles (compare Fig. 3.2.1). It is invariant under rotations by $n \cdot 60°$ modulo 360° (*hexagonal symmetry*) about an axis through a node and perpendicular to the lattice plane. At each node and each link to the nearest neighbor there is a cell which may be empty or occupied by at most one particle (*exclusion principle*). All particles have the same mass m (set to 1 for simplicity) and are indistinguishable. The state of a node can be described by six bits. The exclusion principle leads to an equilibrium distribution of the mean occupation numbers of *Fermi-Dirac* type. Each cell is associated with a *lattice vector* $\boldsymbol{c}_i$ which connects a node with its nearest neighbor in direction i. The lattice vectors $\boldsymbol{c}_i$ are also called *lattice velocities* because the time step Δt is always set to 1 in lattice-gas cellular automata and therefore $\boldsymbol{c}_i$ and $\boldsymbol{c}_i/\Delta t$ have the same numerical values. Because all particles have the same mass $m = 1$, $\boldsymbol{c}_i$ is also the particle's momentum. The respective meaning of $\boldsymbol{c}_i$ can be recognized from the context.
As for HPP there are 2-particle *head-on collisions* (compare Fig. 3.2.2). The initial state[10] $(i, i+3)$ can be transformed into one of two different final states $(i+1, i+4)$ or $(i-1, i+2)$ (rotation by 60° to the left or right) while conserving mass and momentum density. If one chooses always one and the

[10] The description of the state of a node will be given in terms of indices of the occupied cells whereby the cell index $i > 0$ is understood as modulo 6. The index $i = 0$ will be assigned to rest particles (see below).

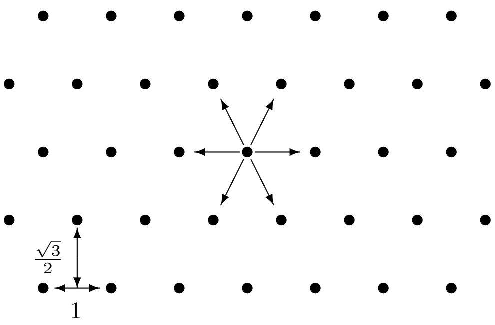
same final state the model becomes *chiral*[11]: it is not invariant with respect to spatial reflections (parity transformation). This is an undesired property because the hydrodynamic equations do not break parity symmetry. To restore reflection symmetry on the macroscopic level the choice between the different final states will be made by a random process with equal probabilities for rotation to the left and to the right. Thus in contrast to HPP the FHP model encompasses *nondeterministic rules.*
The generation of random numbers is a time consuming process. Therefore a *pseudo-random choice* is used where the rotational sense changes by chance for the whole domain from time step to time step (i.e. only one random number per time step has to be generated) or the sense of rotation changes from node to node but is constant in time (i.e. random numbers have to be generated only in the initial step for all nodes).
The 2-particle collisions conserve not only mass and momentum but also the difference of the number of particles that stream in opposite directions (the same invariant as for HPP). This additional invariant has no counterpart in 'real world hydrodynamics' and therefore is called a *spurious invariant.* It further restricts the dynamic of the lattice-gas cellular automata and can lead to deviations from hydrodynamic behavior on the macroscopic scale. The invariance of the particle differences can be destroyed by *symmetric* 3*-particle collisions* which conserve mass and momentum (compare Fig. 3.2.2). 2- and 3-particle collisions form a minimal set of collisions for FHP. This version of FHP is called *FHP-I.* Introduction of additional collisions like 4-particle collision, 2-particle collision with spectator and collisions including rest particles lead to various variants (*FHP-II*, *FHP-III*: see, for example, Frisch et al., 1987 and Hayot and Lakshmi, 1989). The corresponding macroscopic equations all have the same form (*universality theorem*) and differ only in their viscosity coefficients. As a rule of thumb the viscosity coefficient decreases with increasing number of collisions.

The 3-particle collisions destroy a spurious invariant. Unfortunately no method exists to detect all invariants of a given lattice-gas cellular automata model. Of course this is an unsatisfactory situation in the light that the invariants play an essential role in the equilibrium distributions of the mean occupation numbers. Indeed Zanetti (1989) found *spurious invariants* for all variants of the FHP model. Fortunately these *staggered invariants* are not set to values above a certain noise level by the usual initialization procedure and obviously are not generated by interactions with obstacles. So they do not influence the macroscopic dynamic too much. A discussion of the *Zanetti invariants* will be given in Section 3.8.

Frisch et al. (1987) give a discussion of a unified theory for the lattice-gas cellular automata HPP, FHP and FCHC. This excellent paper may be heavy fare for the beginner. The following discussion is restricted to the FHP model

[11] The word *chiral* derives from the Greek word for 'hands'.

Fig. 3.2.1. *The triangular lattice of the FHP model shows hexagonal symmetry. The lattice velocities* $\boldsymbol{c}_i$ *are represented by arrows.*

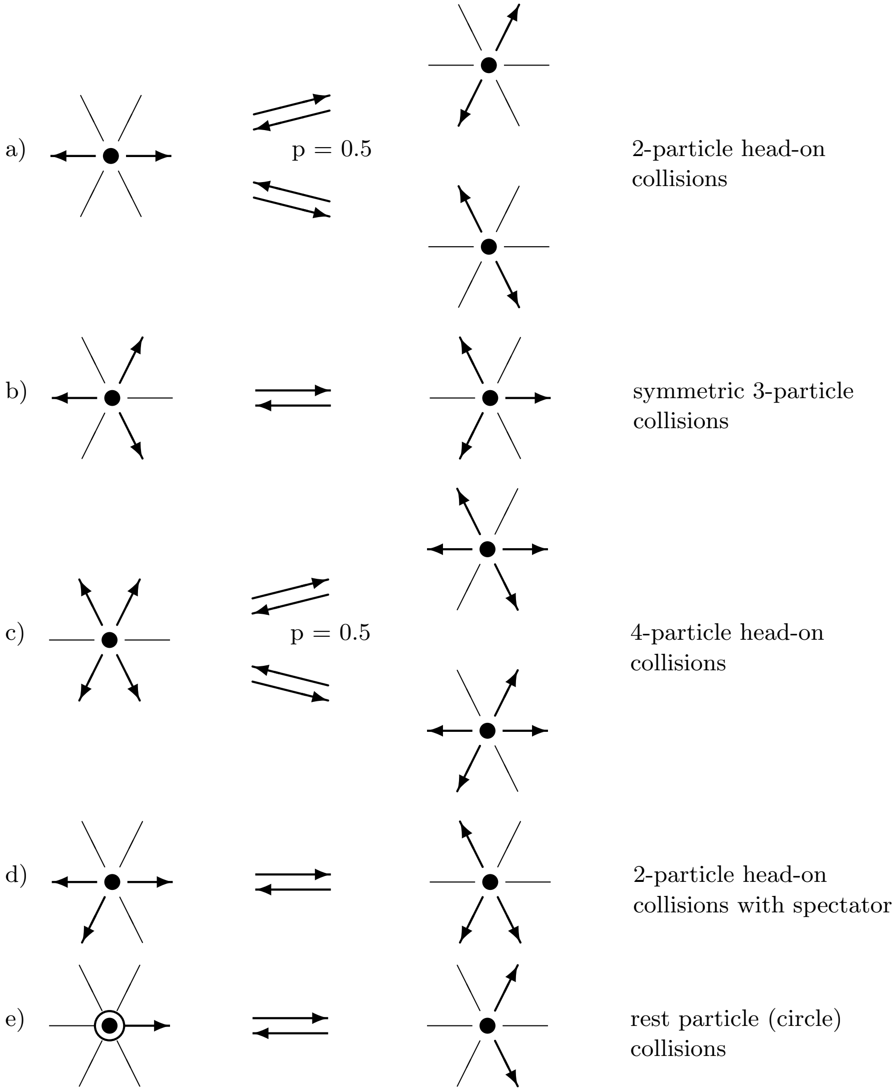


without rest particles (FHP-I) but from time to time more general results from Frisch et al. (1987) will be quoted.

The essential properties of the FHP model read:

1. The underlying regular lattice shows *hexagonal symmetry.*
2. *Nodes* (also called *sites*) are linked to six nearest neighbors located all at the same distance with respect to the central node.
3. The vectors $\boldsymbol{c}_i$ linking nearest neighbor nodes are called *lattice vectors* or *lattice velocities*
$$\boldsymbol{c}_i = \left(\cos\frac{\pi}{3}i, \sin\frac{\pi}{3}i\right), \quad i = 1, ..., 6. \tag{3.2.1}$$
with $|\boldsymbol{c}_i| = 1$ for all i.
4. A cell is associated with each link at all nodes.
5. Cells can be empty or occupied by at most one particle (*exclusion principle*).
6. All particles have the same mass (set to 1 for simplicity) and are indistinguishable.
7. The evolution in time proceeds by an alternation of *collision* $\mathcal{C}$ and *streaming* $\mathcal{S}$ (also called *propagation*):
$$\mathcal{E} = \mathcal{S} \circ \mathcal{C}, \tag{3.2.2}$$
where $\mathcal{E}$ is called the *evolution operator.*
8. The collisions are strictly *local*, i.e. only particles of a single node are involved.

Fig. 3.2.2. *All possible collisions of the FHP variants: occupied cells are represented by arrows, empty cells by thin lines.*


a)
p = 0.5
2-particle head-on collisions
b)
symmetric 3-particle collisions
c)
p = 0.5
4-particle head-on collisions
d)
2-particle head-on collisions with spectator
e)
rest particle (circle) collisions

$\boldsymbol{r}$ is the position vector of a node and $\boldsymbol{r}+\boldsymbol{c}_i$ are the position vectors of its nearest neighbors. Each pair of lattice vectors $(\boldsymbol{c}_i, \boldsymbol{c}_j)$ is associated with an element of the isometric group[12] $\mathcal{G}$ which transforms $\boldsymbol{c}_i$ into $\boldsymbol{c}_j$ by a rotation of $n \cdot 60°$. The first three moments of the lattice velocities $\boldsymbol{c}_i$ read:

$$\sum_i \boldsymbol{c}_i = 0 \quad \text{(symmetry of the lattice!)} \tag{3.2.3}$$

$$\sum_i c_{i\alpha} c_{i\beta} = 3\delta_{\alpha\beta} \tag{3.2.4}$$

$$\sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0 \tag{3.2.5}$$

where the sum over i always runs from 1 to 6. The *Latin indices* refer to the lattice vectors and run from 1 to 6 whereas the *Greek indices* assign the cartesian components of the vectors and therefore run from 1 to 2. To each node a *bit-state* $n(\boldsymbol{r}) = \{n_i(\boldsymbol{r}), i = 1, ..., 6\}$ will be assigned where the $n_i \in \{0, 1\}$ are Boolean variables. The *streaming* (*propagation*) is defined by

$$n_i(\boldsymbol{r}) = \mathcal{S} n_i(\boldsymbol{r} - \boldsymbol{c}_i). \tag{3.2.6}$$

Collisions take place synchronously at every node and transform the *initial state* of a node $s = \{s_i, i = 1, ..., 6\}$ into the *final state* $s' = \{s'_i, i = 1, ..., 6\}$ according to the collision rules described above. A certain *transition probability*

$$A(s \to s') \geq 0 \tag{3.2.7}$$

is assigned to each pair of initial and final state. The transition probabilities satisfy the *normalization*

$$\forall s : \quad \sum_{s'} A\,(s \to s') = 1 \tag{3.2.8}$$

The only combinations of six real numbers a_i which fulfill the constraints

$$\forall s, s' : \quad \sum_i (s'_i - s_i)\, A\,(s \to s')\, a_i = 0 \tag{3.2.9}$$

are linear combinations of 1 (for all i) and $\boldsymbol{c}_i$, i.e. mass and momentum conservation. This is indeed the case for FHP. The *Zanetti invariants* are *non-local invariants*.

The transition probabilities are invariant with respect to each element of the *isometric group* $\mathcal{G}$:

$$\forall g \in \mathcal{G},\, \forall s, s' : \quad A\,(g(s) \to g(s')) = A\,(s \to s')\,. \tag{3.2.10}$$

[12] Isometries are mappings g which keep the distances $d(\alpha, \beta)$ of arbitrary points α, β invariant: $d(g(\alpha), g(\beta)) = d(\alpha, \beta)$. Rotations and reflections are isometries.

The FHP model fulfills the *detailed balance*

$$\boxed{A(s \to s') = A(s' \to s),} \tag{3.2.11}$$

i.e. the probabilities for each collision and its inverse collision are equal. For the derivation of the equilibrium distribution (compare Theorem 3.2.1) the weaker condition of *semi-detailed balance* (or *Stueckelberg condition*[13])

$$\boxed{\forall s' : \quad \sum_{s} A(s \to s') = 1} \tag{3.2.12}$$

is sufficient.

There are $2^6 = 64$ different states for a node of the FHP-I model. The transition probabilities $A(s \to s')$ form a 64×64 *transition matrix* $A_{ss'}$. The two constraints (3.2.8) and (3.2.12) are equivalent to the statement that the sums of each line or each column of $A_{ss'}$ are equal to 1.

Exercise 3.2.1. (*)
How does the transition matrix $A_{ss'}$ look like? Write down the submatrix which contains the 2- and 3-particle collisions.

Exercise 3.2.2. (**)
Prove: the 2- and 3-particle collisions locally (at a single node) conserve only mass and momentum.

Exercise 3.2.3. (*)
Consider a system with three possible states (a, b, c) and transition probabilities
1.)

$$\begin{aligned} A(a \to b) &= 0.5 \qquad & A(b \to a) &= 0.5 \\ A(b \to c) &= 0.5 \qquad & A(c \to b) &= 0.5 \\ A(c \to a) &= 0.5 \qquad & A(a \to c) &= 0.5 \end{aligned}$$

or 2.)

$$\begin{aligned} A(a \to b) &= 0.8 \qquad & A(b \to a) &= 0.2 \\ A(b \to c) &= 0.8 \qquad & A(c \to b) &= 0.2 \\ A(c \to a) &= 0.8 \qquad & A(a \to c) &= 0.2 \end{aligned}$$

[13] Stueckelberg (1952) showed that this condition is sufficient to prove the H-theorem.

or 3.)

$$\begin{aligned} A(a\to b) &= 0.8 & \qquad A(b\to a) &= 0.3 \\ A(b\to c) &= 0.7 & \qquad A(c\to b) &= 0.4 \\ A(c\to a) &= 0.6 & \qquad A(a\to c) &= 0.2. \end{aligned}$$

Calculate the distribution $(a,b,c)_t$ for large t given the initial value $(a,b,c)_{t=0} = (1,0,0)$ for all three cases. In which cases is detailed or semi-detailed balance fulfilled?

3.2.2 Microdynamics of the FHP model

For hydrodynamics certain averaged quantities like mass and momentum density are of interest. To understand the behavior of these averaged quantities a description of lattice-gas cellular automata in terms of statistical mechanics will be formulated. First the analogue to the *Hamilton equations* in classical statistical mechanics will be considered.

Boolean description of the microdynamics. The state of the cells of the lattice-gas cellular automata is described by the *Boolean*[14] arrays $n_i(t,\boldsymbol{r})$:

$$n_i(t,\boldsymbol{r}) \;=\; \begin{cases} 1 & \text{if cell i is occupied} \\ 0 & \text{if cell i is empty} \end{cases}$$

where $\boldsymbol{r}$ and t indicate the discrete points in space and time. The time evolution in terms of Boolean arrays reads:

$$\begin{aligned} &n_i\left(t+1,\boldsymbol{r}+\boldsymbol{c}_i\right) = n_i(t,\boldsymbol{r}) \wedge \\ &\quad \{[(n_i \wedge n_{i+1})\,\&\,(n_{i+1}\wedge n_{i+2})\,\&\,(n_{i+2}\wedge n_{i+3}) \\ &\quad \&\,(n_{i+3}\wedge n_{i+4})\,\&\,(n_{i+4}\wedge n_{i+5})] \\ &\quad \mid [n_i \& n_{i+3} \& \sim (n_{i+1} \mid n_{i+2} \mid n_{i+4} \mid n_{i+5})] \\ &\quad \mid [\xi \& n_{i+1} \& n_{i+4} \& \sim (n_i \mid n_{i+2} \mid n_{i+3} \mid n_{i+5})] \\ &\quad \mid [\sim \xi \& n_{i+2} \& n_{i+5} \& \sim (n_i \mid n_{i+1} \mid n_{i+3} \mid n_{i+4})]\;\} \end{aligned} \tag{3.2.13}$$

where the occupation numbers n_i on the right side refer to the input-states at node $\boldsymbol{r}$ and time t. The symbols

[14] A short introduction to the Boolean algebra can be found in Appendix 6.1.

$$\begin{aligned} \& &= \text{AND} \\ | &= \text{OR (inclusive or)} \\ \wedge &= \text{XOR (exclusive or)} \\ \sim &= \text{NOT.} \end{aligned} \tag{3.2.14}$$

have been used. ξ is a Boolean random variable which determines the sense of rotation for the 2-particle collisions (denoted 'xi' in the program code listed below). Eq. (3.2.13) looks rather nasty to people who do not use Boolean algebra every day. The interpretation is however simple. Eq. (3.2.13) states that the state of cell i at node $\boldsymbol{r} + \boldsymbol{c}_i$ at time $t + 1$ is given by $n_i(\boldsymbol{r}, t)$ when no collision happens or (exclusive!) $n_i(\boldsymbol{r}, t)$ is changed by collision before its value is propagated to $\boldsymbol{r} + \boldsymbol{c}_i$. The following collisions are taken into account:

1. A symmetric 3-particle collision happens at node $\boldsymbol{r}$ at time t when the n_j ($j = 1, ..., 6$) are alternating occupied and empty. When a 3-particle collision happens and n_i is occupied at input cell i will change to the empty state, whereas in a 3-particle collision with n_i empty initially cell i will be occupied after collision.
2. If a 2-particle collision happens where on input cells i and $i + 3$ are occupied (and all other cells empty) n_i will change to 0.
3. If a 2-particle collision happens where on input cells $i + 1$ and $i + 4$ are occupied (and all other cells empty) and the Boolean random variable ξ is 1 such that the rotation of the outgoing particles is in the clockwise direction (otherwise n_i would not change!).
4. If a 2-particle collision happens where on input cells $i + 2$ and $i + 5$ are occupied (and all other cells empty) and the Boolean random variable ξ is 0 such that the rotation of the outgoing particles is in the counterclockwise direction.

The four different cases correspond to the last five lines on the right hand side of Eq. (3.2.13). n_i is changed if any one of the collisions happens. Therefore the various collision terms are connected by inclusive or. The exclusive or between $n_i(\boldsymbol{r}, t)$ and the collision terms enforces a change when any one of the collisions happens.

The coding of the FHP model is based on the Boolean formulation of the microdynamics. The code in C reads as follows (*nsbit* is the non-solid bit which is set to 1 in the fluid and to 0 inside obstacles; it is used here to suppress collisions inside obstacles):

```
/* loop over all sites */

    for(x=0; x<XMAX; x++) {
    for(y=0; y<YMAX; y++) {
      a = i1[x][y];
      b = i2[x][y];
      c = i3[x][y];
      d = i4[x][y];
      e = i5[x][y];
      f = i6[x][y];

/* two-body collision
   <-> particles in cells a (b,c) and d (e,f)
       no particles in other cells
   <-> db1 (db2,db3) = 1                   */

   db1 = (a&d&~(b|c|e|f));
   db2 = (b&e&~(a|c|d|f));
   db3 = (c&f&~(a|b|d|e));

/* three-body collision <-> 0,1 (bits) alternating
   <-> triple = 1 */

   triple = (a^b)&(b^c)&(c^d)&(d^e)&(e^f);

/* change a and d
   <-> three-body collision triple=1
      or two-body collision db1=1
      or two-body collision db2=1 and xi=1   (- rotation)
      or two-body collision db3=1 and noxi=1 (+ rotation)
    <-> chad=1                                  */

   xi = irn[x][y];       /* random bits */
   noxi = ~xi;

   nsbit = nsb[x][y];    /* non solid bit */

   cha = ((triple|db1|(xi&db2)|(noxi&db3))&nsbit);
   chd = ((triple|db1|(xi&db2)|(noxi&db3))&nsbit);
   chb = ((triple|db2|(xi&db3)|(noxi&db1))&nsbit);
   che = ((triple|db2|(xi&db3)|(noxi&db1))&nsbit);
   chc = ((triple|db3|(xi&db1)|(noxi&db2))&nsbit);
   chf = ((triple|db3|(xi&db1)|(noxi&db2))&nsbit);
/* change: a = a ^ chad  */

   k1[x][y] = i1[x][y]^cha;
   k2[x][y] = i2[x][y]^chb;
   k3[x][y] = i3[x][y]^chc;
   k4[x][y] = i4[x][y]^chd;
   k5[x][y] = i5[x][y]^che;
   k6[x][y] = i6[x][y]^chf;

/* collision finished */
}}
```

Exercise 3.2.4. (**)
Write the analogue of Eq. (3.2.13) for the case where rest particle collisions (compare Fig. 3.2.2 e) are included.

Arithmetic description of the microdynamics. The transition from the Boolean to the arithmetic formulation of the microdynamics can be achieved by formal substitutions (compare Table 3.2.1). However, this procedure is too laborious. Instead the collision function Δ_i is introduced by

$$\boxed{n_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) = n_i(t, \boldsymbol{r}) + \Delta_i.} \tag{3.2.15}$$

The collision function can be constructed according to the following recipe:

- For each collisional configuration, i.e. one that will lead to a change of the occupation numbers, a product of the n_i (if $n_i = 1$) respectively $(1 - n_i)$ (if $n_i = 0$) for $i = 1, ..., b$ (b total number of lattice velocities) is written. This product yields 1 if the specific configuration is given and 0 otherwise.
- For nondeterministic collision rules (FHP) the products defined above are multiplied by a random variable ξ, $(1 - \xi)$ or 1, respectively.
- The products receive positive or negative sign according to whether n_i increases or decreases when changing from input (before collision) to output (after collision) state.
- All products are added up.

To give an example let us consider the HPP model. There are only two collisional configurations, namely $(1, 0, 1, 0)$ and $(0, 1, 0, 1)$. The corresponding products read

$$n_{i+1}n_{i+3}(1 - n_i)(1 - n_{i+2}) \quad \text{and} \quad n_i n_{i+2}(1 - n_{i+1})(1 - n_{i+3})$$

where n_i increase in the former case and decreases in the latter. Thus the collision function is given by

$$\Delta_i = n_{i+1}n_{i+3}(1-n_i)(1-n_{i+2}) - n_i n_{i+2}(1-n_{i+1})(1-n_{i+3}).$$

Table 3.2.1. *'Translation' of Boolean expressions into arithmetic formulas.*

a	b				
		AND	OR	XOR	NOT
		$a\&b$	$a \mid b$	$a \wedge b$	$\sim a$
		$a \cdot b$	$a+b-a\cdot b$	$a+b-2\cdot a\cdot b$	$1-a$
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

The analogous procedure yields for the FHP-I model

$$\begin{aligned} \Delta_i(n) \;=\;& n_{i+1}n_{i+3}n_{i+5}(1-n_i)(1-n_{i+2})(1-n_{i+4}) \\ & -n_i n_{i+2}n_{i+4}(1-n_{i+1})(1-n_{i+3})(1-n_{i+5}) \\ & +\xi n_{i+1}n_{i+4}(1-n_i)(1-n_{i+2})(1-n_{i+3})(1-n_{i+5}) \\ & +(1-\xi)n_{i+2}n_{i+5}(1-n_i)(1-n_{i+1})(1-n_{i+3})(1-n_{i+4}) \\ & -n_i n_{i+3}(1-n_{i+1})(1-n_{i+2})(1-n_{i+4})(1-n_{i+5}) \end{aligned} \qquad (3.2.16)$$

where $n = \{n_i, i = 1, ..., 6\}$. The conservation of mass and momentum at each node can be expressed as follows

$$\forall n: \quad \sum_{i=1}^{6} \Delta_i(n) = 0 \qquad (3.2.17)$$

and

$$\forall n: \quad \sum_{i=1}^{6} \boldsymbol{c}_i \Delta_i(n) = 0. \qquad (3.2.18)$$

These equations imply corresponding conservation laws for the Boolean arrays n_i

$$\sum_i n_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) = \sum_i n_i(t, \boldsymbol{r}) \qquad (3.2.19)$$

and

$$\sum_i \boldsymbol{c}_i n_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) = \sum_i \boldsymbol{c}_i n_i(t, \boldsymbol{r}). \qquad (3.2.20)$$

Exercise 3.2.5. (*)
Derive Eq. (3.2.17).

3.2.3 The Liouville equation

Now lattice-gas cellular automata will be considered from the viewpoint of statistical mechanics. In classical statistical mechanics the deterministic description of systems with many degrees of freedom by Hamiltonian equations is abandoned and replaced by a probabilistic approach (*Gibbs' ensemble*: instead of deriving equilibrium values from the long time limit of a single system, one calculates them as mean values over a large hypothetical set of 'equivalent' systems [15]; some authors prefer to avoid the introduction of ensembles: see, for example, Ma, 1993). One can proceed quite similar for lattice-gas cellular automata. The microscopic description of the FHP model encompasses already probabilistic elements (choice of the sense of rotation for 2-particle collisions). To avoid confusion because of these two different types of probabilities the reader should - at least for a while - consider the microdynamics as a deterministic process (like it is indeed for HPP).
Consider a lattice $\mathcal{L}$ of final extend with periodic (cyclic) boundary conditions. The *phase space* Γ (Gibbs) is defined as the *set of all possible states* $s(.)$ of the lattice $\mathcal{L}$. At time $t = 0$ an ensemble of initial states is given with probabilities $P\left(0, s(.)\right) \geq 0$ which add up to 1:

$$\sum_{s(.)\in\Gamma} P\left(0, s(.)\right) = 1.$$

Each member of the ensemble evolves according to the microdynamics of the lattice gas. This implies the conservation of probabilities

$$P\left(t+1, \mathcal{S}s(.)\right) = P\left(t, \mathcal{C}^{-1}s(.)\right) \tag{3.2.21}$$

or

$$P\left(t+1, \mathcal{E}s(.)\right) = P\left(t, s(.)\right). \tag{3.2.22}$$

Eq. (3.2.22) is called the *Liouville equation* because of its close analogy to the Liouville equation in classical statistical mechanics (compare page 139).
In the case of nondeterministic microdynamics like for FHP the Liouville equation has to be replaced by the more general *Chapman-Kolmogorov equation*

$$P\left(t+1, \mathcal{S}s(.)\right) = \sum_{s(.)\in\Gamma} \prod_{\boldsymbol{r}\in\mathcal{L}} A\left(s(\boldsymbol{r}) \to s'(\boldsymbol{r})\right) P\left(t, s(.)\right). \tag{3.2.23}$$

[15] Gibbs (1902) writes in his preface: "For some purposes, however, it is desirable to take a broader view of the subject. We may imagine a great number of systems of the same nature, but differing in the configurations and velocities which they have at a given instant, and differing not merely infinitesimally, but it may be so as to embrace every conceivable combination of configuration and velocities. And here we may set the problem, not to follow a particular system through its succession of configurations, but to determine how the whole number of systems will be distributed among the various conceivable configurations and velocities at any required time, when the distribution has been given for some one time."

Exercise 3.2.6. (*)
How many different states are possible a) at a single node and b) on a lattice with N nodes? When does this number exceed 10^{10}?

3.2.4 Mass and momentum density

In the framework of the probabilistic description *ensemble mean values* $q\,(n(t, ...))$ for *observables* q are defined by

$$\langle q\,(n(t, ...))\rangle := \sum_{s(.)\in\Gamma} q\,(s(.))\,P\,(t, s(.))$$

By far the most important observables are the *mean occupation numbers*

$$N_i(t, \boldsymbol{r}) = \langle n_i\,(t, \boldsymbol{r})\rangle$$

which are used to define the mass

$$\rho(t, \boldsymbol{r}) := \sum_i N_i(t, \boldsymbol{r})$$

and momentum density

$$\boldsymbol{j}(t, \boldsymbol{r}) := \sum_i \boldsymbol{c}_i N_i(t, \boldsymbol{r}).$$

These quantities are defined with respect to nodes and not to cells or area[16]. The density per cell d is calculated by division of ρ by the number of cells per node b ($= 6$ for FHP-I):

$$d = \frac{\rho}{b}. \tag{3.2.24}$$

The flow velocity is defined by the (non-relativistic) relation *momentum density = mass density · velocity*:

$$\boldsymbol{j}(t, \boldsymbol{r}) = \rho(t, \boldsymbol{r})\,\boldsymbol{u}(t, \boldsymbol{r}).$$

Of course the microscopic conservation equations (3.2.19) and (3.2.20) imply conservation of the averaged quantities

$$\sum_i N_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) = \sum_i N_i(t, \boldsymbol{r}), \tag{3.2.25}$$

$$\sum_i \boldsymbol{c}_i N_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) = \sum_i \boldsymbol{c}_i N_i(t, \boldsymbol{r}) \tag{3.2.26}$$

[16] The area per node is $\sqrt{3}/2$ (compare Fig. 3.2.1).

3.2.5 Equilibrium mean occupation numbers

After many definitions in the previous subsections now one of the main results of the theoretical analysis of the FHP model will be derived, namely the *equilibrium occupation numbers* N_i^{eq}. Frisch et al. (1987) proved the following theorem which is valid for HPP, FHP and FCHC:

Theorem 3.2.1. *(Frisch et al., 1987)*
The following statements are equivalent:

1. *The N_i^{eq}'s are a solution of equation (3.2.23).*
2. *The N_i^{eq}'s are a solution of the set of b equations*

$$\forall i = 1, ..., b: \quad \Delta_i(N) := \sum_{ss'} \left(s_i' - s_i\right) A\left(s \to s'\right) \prod_j N_j^{s_j} \left(1 - N_j\right)^{(1-s_j)} = 0. \tag{3.2.27}$$

3. *The N_i^{eq}'s are given by the Fermi-Dirac distribution*

$$N_i^{eq} = \frac{1}{1 + \exp\left(h + \boldsymbol{q} \cdot \boldsymbol{c}_i\right)} \tag{3.2.28}$$

where h is a real number and $\boldsymbol{q}$ is a D-dimensional vector.

Proof. The complete proof is given in Appendix C of Frisch et al. (1987). Here only the step from 2. to 3. will be discussed. The semi-detailed balance condition and the nonexistence of spurious invariants has to be taken into account. In the following the superscript '*eq*' will be dropped in order to keep the notation simple.
Define

$$\breve{N}_i := \frac{N_i}{1 - N_i}$$

and

$$\Pi := \prod_{j=1}^{b} (1 - N_j).$$

Eq. (3.2.27) may be written as

$$\frac{\Delta_i}{\Pi} = \sum_{ss'} (s_i' - s_i) A(s \to s') \prod_j \breve{N}_j^{s_j} = 0. \tag{3.2.29}$$

Multiply Eq. (3.2.29) by $\breve{N}_i$, sum over i and use

$$\sum_i (s_i' - s_i) \log \breve{N}_i = \log \frac{\prod_j \breve{N}_j^{s_j'}}{\prod_j \breve{N}_j^{s_j}}$$

to obtain

$$\sum_{ss'} A(s \to s') \log \left(\frac{\prod_j \breve{N}_j^{s'_j}}{\prod_j \breve{N}_j^{s_j}} \right) \prod_j \breve{N}_j^{s_j} = 0. \tag{3.2.30}$$

Semi-detailed balance

$$\sum_{s} A(s \to s') = \sum_{s'} A(s \to s') = 1$$

implies that

$$\sum_{ss'} A(s \to s') \left(\prod_j \breve{N}_j^{s_j} - \prod_j \breve{N}_j^{s'_j} \right) = 0. \tag{3.2.31}$$

Combining Eqs. (3.2.30) and (3.2.31), one obtains

$$\sum_{ss'} A(s \to s') \left[\log \left(\frac{\prod_j \breve{N}_j^{s'_j}}{\prod_j \breve{N}_j^{s_j}} \right) \prod_j \breve{N}_j^{s_j} + \prod_j \breve{N}_j^{s_j} - \prod_j \breve{N}_j^{s'_j} \right] = 0. \tag{3.2.32}$$

The relation ($x > 0$, $y > 0$)

$$y \log \frac{x}{y} + y - x = - \int_x^y \log \frac{t}{y}\, dt \le 0 \tag{3.2.33}$$

where equality being achieved only when $x = y$ will be exploited. The left hand side of (3.2.32) is a linear combination of expressions of the form (3.2.33) with nonnegative weights $A(s \to s')$. For it to vanish, one must have

$$\prod_j \breve{N}_j^{s_j} = \prod_j \breve{N}_j^{s'_j} \quad \text{whenever} \quad A(s \to s') \neq 0.$$

This is equivalent to

$$\forall s, s' : \quad \sum_i \log(\breve{N}_i)(s'_i - s_i) A(s \to s') = 0. \tag{3.2.34}$$

Eq. (3.2.34) means that $\log(\breve{N}_i)$ is a collision invariant. Now assuming that only mass and momentum are conserved and no spurious invariants exist, one concludes that

$$\log(\breve{N}_i) = -(h + \boldsymbol{q} \cdot \boldsymbol{c}_i),$$

which is the most general collision invariant (a linear combination of the mass invariant and of the D momentum invariants). Reverting to the mean populations $N_i = \breve{N}_i/(1 + \breve{N}_i)$, one obtains (3.2.28). **q.e.d.**

Calculation of the Lagrange multipliers at small flow speeds. The determination of the Lagrange multipliers[17] h and $\boldsymbol{q}$ is constrained by the conserved quantities

$$\rho = \sum_i N_i = \sum_i \frac{1}{1+\exp\left(h+\boldsymbol{q}\cdot\boldsymbol{c}_i\right)} \tag{3.2.35}$$

and

$$\rho\boldsymbol{u} = \sum_i N_i\boldsymbol{c}_i = \sum_i \frac{\boldsymbol{c}_i}{1+\exp\left(h+\boldsymbol{q}\cdot\boldsymbol{c}_i\right)}. \tag{3.2.36}$$

These equations apply also for HPP where the problem of the determination of the Lagrange multipliers can be reduced to a cubic equation (Hardy et al., 1973). In contrast to HPP, for the FHP model explicit solutions are known only in a few special cases.

The Lagrange multipliers can be calculated by an expansion for small *Mach numbers* $M_a := U/c_s$, i.e. for speeds $U = |\boldsymbol{u}|$ well below the sound speed c_s. The somewhat lengthy calculations in Appendix 6.2 apply to FHP and HPP.

The equilibrium distributions for FHP-I read

$$\boxed{N_i^{eq}(\rho,\boldsymbol{u}) = \frac{\rho}{6} + \frac{\rho}{3}\boldsymbol{c}_i\cdot\boldsymbol{u} + \rho G(\rho)Q_{i\alpha\beta}u_\alpha u_\beta + \mathcal{O}\left(u^3\right)} \tag{3.2.37}$$

with

$$G(\rho) = \frac{1}{3}\frac{6-2\rho}{6-\rho} \tag{3.2.38}$$

and

$$Q_{i\alpha\beta} = c_{i\alpha}c_{i\beta} - \frac{1}{2}\delta_{\alpha\beta}. \tag{3.2.39}$$

The term quadratic in $\boldsymbol{u}$ will lead to the nonlinear advection term in the Navier-Stokes equation. This is the reason to expand the N_i^{eq} up to second order.

Frisch et al. (1987) have shown that the generalization of (3.2.37) to (3.2.39), namely

$$N_i^{eq}(\rho,\boldsymbol{u}) = \frac{\rho}{b} + \frac{\rho D}{c^2 b}c_{i\alpha}u_{i\alpha} + \rho G(\rho)Q_{i\alpha\beta}u_\alpha u_\beta + \mathcal{O}\left(u^3\right) \tag{3.2.40}$$

with

$$G(\rho) = \frac{D^2}{2c^4 b}\frac{b-2\rho}{b-\rho} \tag{3.2.41}$$

and

$$Q_{i\alpha\beta} = c_{i\alpha}c_{i\beta} - \frac{c^2}{D}\delta_{\alpha\beta}. \tag{3.2.42}$$

(b number of cells per node, D dimension) is valid for HPP, FHP-I, FHP-II, FHP-III and FCHC.

[17] The Fermi-Dirac distribution (3.2.28) could be derived from a *maximum entropy principle* whereby the constraints of mass and momentum conservation are coupled to the entropy (Shannon) by Lagrange multipliers like h and $\boldsymbol{q}$. This method will be discussed in some detail in Section 4.3 and applied in Section 5.2.

Exercise 3.2.7. (*)
Specify the equilibrium distribution for HPP and compare it with the linear distribution (3.1.13) which was used for initialization in Section 3.1.

3.2.6 Derivation of the macroscopic equations: multi-scale analysis

The derivation of macroscopic equations by *multi-scale analysis* is one of the most demanding topics in the whole book. In the current section the expansion will be followed up to first order only. The calculation of all second order terms is too involved to be shown here in detail. The multi-scale analysis will be discussed again in a special section devoted to the Chapman-Enskog expansion (Section 4.2) and will be applied up to second order in Section 5.2 in the context of lattice Boltzmann models.
In the former subsection the distribution functions for a *global* (homogeneous) equilibrium were derived. The interesting aspects of fluid flows and of nature in general lie, however, in its variations in space and time. One can think of the 'real world' as a *patchwork of thermodynamic equilibria* whose parameters like mass, momentum or energy[18] density show slow changes in space, such that every point can be characterized by the *local* values of mass, momentum and energy density.

For FHP equilibrium mean occupation numbers N_i^{eq} have been derived which depend continuously on tunable parameters, namely the mean values of the conserved quantities mass and momentum density. At the beginning of a numerical simulation a distribution of mass and momentum density will be initialized which varies on large (compared to the lattice unit[19]) spatial scales ϵ^{-1} (measured in lattice units; ϵ is a small number). In the course of time three phenomena can be distinguished with respect to their characteristic time scales:

1. *Relaxation toward local equilibrium* with time scale ϵ^0: very fast. Few collision are necessary to reach local equilibrium (compare Fig. 3.2.6). Note that the number of collisions per time interval is a function of mass density. Thus the characteristic time scale ϵ^0 is large at low and high mass densities where collisions are rare.
2. Sound waves (perturbations of mass density) and advection with time scale ϵ^{-1}: fast, but slower than relaxation toward local equilibrium.
3. Diffusion with time scale ϵ^{-2}: distinctly slower than sound waves and advection.

[18] Energy does not play a role for FHP and many other LGCA because it does not exist as an independent quantity or because it is not conserved by certain collisions whereas so-called thermal LGCA include an energy equation (compare Section 3.7).

[19] Lattice unit = distance between neighboring nodes = 1 for FHP

Let us consider, for example, spatial variations on the scale of $\epsilon^{-1} = 100$ lattice units which may be due to an obstacle in a flow (compare the von Karman vortex street shown in Fig. 3.6.5). The relaxation toward local equilibrium proceeds on a time-scale of order $\epsilon^0 = 1$, i.e. in a few time steps[20]. Sound waves[21] and advection show time scales of the order of $\epsilon^{-1} = 100$ whereas diffusion with time-scales of $\epsilon^{-2} = 10\ 000$ is much slower.

The microdynamics of the lattice-gas cellular automata contain all of these phenomena. We will now apply the so-called *multi-scale technique* to 'pick out' the processes of interest here, namely the hydrodynamic modes. Corresponding to the differentiation given above one introduces three time variables:

$$\begin{aligned} &t_\star && \text{(discrete)} \\ &t_1 &=& \ \epsilon t_\star \\ &t_2 &=& \ \epsilon^2 t_\star \end{aligned}$$

where the last two scales will be considered as continuous[22] variables. With respect to space only two scales have to be distinguished because sound waves and advection as well as diffusion act on similar spatial scales:

$$\begin{aligned} &\boldsymbol{r}_\star && \text{(discrete)} \\ &\boldsymbol{r}_1 &= \epsilon \boldsymbol{r}_\star \quad & \text{(continuous)} \end{aligned}$$

$N_i^{(0)}$ will refer to mean occupation numbers for given local values of ρ and $\boldsymbol{u}$. The $N_i^{(0)}$ are given by the *global* form of the equilibrium distributions (3.2.37) but where ρ and $\boldsymbol{u}$ are the *local* values of mass density and velocity, i.e. $N_i^{(0)}(\boldsymbol{r}, t) \equiv N_i^{eq}(\rho, \boldsymbol{u})$. The actual occupation numbers $N_i(t, \boldsymbol{r})$ are close to the equilibrium values and therefore can be expanded about $N_i^{(0)}$:

$$N_i = N_i^{(0)}(t, \boldsymbol{r}) + \epsilon N_i^{(1)}(t, \boldsymbol{r}) + \mathcal{O}\left(\epsilon^2\right) \tag{3.2.43}$$

Terms of higher than linear order in ϵ will be neglected. The linear corrections do not contribute to the local values of mass and momentum density:

$$\sum_i N_i^{(1)}(t, \boldsymbol{r}) = 0 \quad \text{and} \quad \sum_i \boldsymbol{c}_i N_i^{(1)}(t, \boldsymbol{r}) = 0 \tag{3.2.44}$$

which follows from

[20] At reasonable mass densities, because otherwise there are not enough collisions.

[21] Impressive sound waves are created, for example, when an obstacle like the plate in Fig. 3.6.5 is suddenly put into an initially homogeneous flow.

[22] Over longer time scales a thing or two are smoothed out.

$$\rho = \sum_i N_i(t, \boldsymbol{r}) \quad = \quad \sum_i N_i^{(0)}(t, \boldsymbol{r}) \quad \text{and} \qquad (3.2.45)$$

$$\boldsymbol{j} = \sum_i \boldsymbol{c}_i N_i(t, \boldsymbol{r}) \quad = \quad \sum_i \boldsymbol{c}_i N_i^{(0)}(t, \boldsymbol{r}). \qquad (3.2.46)$$

In the *Chapman-Enskog expansion* of the Boltzmann equation (Chapman, 1916, 1918; Enskog, 1917; Chapman and Cowling, 1970; Cercignani, 1990) ϵ is the *Knudsen number* K_n, i.e. the ratio between the mean free path length l and the characteristic length scale of the system L which can be the diameter of an obstacle (for example: flow past a sphere) or the size of the whole domain. The hydrodynamic (continuous) regime is characterized by small Knudsen numbers whereas finite size effects play a role at Knudsen number of order 1 or higher (Knudsen flows).
The microscopic conservation laws (3.2.25) and (3.2.26) are the starting point for the multi-scale analysis. The mean populations after collision and propagation are expanded up to second order in ϵ around its values before the collision step:

$$\begin{aligned} \left\{ \begin{matrix} 1 \\ \boldsymbol{c}_i \end{matrix} \right\} N_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) \quad = \quad & \left\{ \begin{matrix} 1 \\ \boldsymbol{c}_i \end{matrix} \right\} [\, N_i(t, \boldsymbol{r}) + \partial_t N_i + c_{i\alpha} \partial_{x_\alpha} N_i \\ & + \frac{1}{2} \partial_t \partial_t N_i + \frac{1}{2} c_{i\alpha} c_{i\beta} \partial_{x_\alpha} \partial_{x_\beta} N_i \qquad (3.2.47) \\ & + c_{i\alpha} \partial_t \partial_{x_\alpha} N_i + \mathcal{O}\left(\partial^3 N_i\right) \,] \end{aligned}$$

In what follows the fast (local) relaxation processes will be neglected in the theoretical description because one is only interested in the hydrodynamic behavior of the lattice-gas cellular automata. The derivations in time and space are substituted in Eq. (3.2.47) according to the scalings given above:

$$\partial_t \quad \longrightarrow \quad \epsilon \partial_t^{(1)} + \epsilon^2 \partial_t^{(2)} \qquad (3.2.48)$$

$$\partial_{x_\alpha} \quad \longrightarrow \quad \epsilon \partial_{x_\alpha}^{(1)}. \qquad (3.2.49)$$

Insertion of Eqs. (3.2.43), (3.2.47), (3.2.48) and (3.2.49) into Eqs. (3.2.25) and (3.2.26) leads to

$$\begin{aligned} & \sum_i \left\{ \begin{matrix} 1 \\ \boldsymbol{c}_i \end{matrix} \right\} [N_i(t+1, \boldsymbol{r}+\boldsymbol{c}_i) - N_i(t, \boldsymbol{r})] \\ = \quad & \sum_i \left\{ \begin{matrix} 1 \\ \boldsymbol{c}_i \end{matrix} \right\} \Big[\epsilon \partial_t^{(1)} N_i^{(0)} + \epsilon^2 \partial_t^{(1)} N_i^{(1)} + \epsilon^2 \partial_t^{(2)} N_i^{(0)} + \epsilon^3 \partial_t^{(2)} N_i^{(1)} \end{aligned}$$

$$
\begin{aligned}
&+\epsilon c_{i\alpha}\partial_{x_\alpha}^{(1)}N_i^{(0)}+\epsilon^2 c_{i\alpha}\partial_{x_\alpha}^{(1)}N_i^{(1)}+\frac{1}{2}\epsilon^2\partial_t^{(1)}\partial_t^{(1)}N_i^{(0)}+\frac{1}{2}\epsilon^3\partial_t^{(1)}\partial_t^{(1)}N_i^{(1)}\\
&+\epsilon^3\partial_t^{(1)}\partial_t^{(2)}N_i^{(0)}+\epsilon^4\partial_t^{(1)}\partial_t^{(2)}N_i^{(1)}+\epsilon^4\partial_t^{(2)}\partial_t^{(2)}N_i^{(0)}+\epsilon^5\partial_t^{(2)}\partial_t^{(2)}N_i^{(1)}\\
&+\frac{1}{2}\epsilon^2 c_{i\alpha}c_{i\beta}\partial_{x_\alpha}^{(1)}\partial_{x_\beta}^{(1)}N_i^{(0)}+\frac{1}{2}\epsilon^3 c_{i\alpha}c_{i\beta}\partial_{x_\alpha}^{(1)}\partial_{x_\beta}^{(1)}N_i^{(1)}+\epsilon^2 c_{i\alpha}\partial_t^{(1)}\partial_{x_\alpha}^{(1)}N_i^{(0)}\\
&+\epsilon^3 c_{i\alpha}\partial_t^{(1)}\partial_{x_\alpha}^{(1)}N_i^{(1)}+\epsilon^3 c_{i\alpha}\partial_t^{(2)}\partial_{x_\alpha}^{(1)}N_i^{(0)}+\epsilon^4 c_{i\alpha}\partial_t^{(2)}\partial_{x_\alpha}^{(1)}N_i^{(1)}\Big]\\
=\;& 0.
\end{aligned}
$$

To first order in ϵ one obtains

$$\partial_t^{(1)}\sum_i N_i^{(0)}+\partial_{x_\beta}^{(1)}\sum_i c_{i\beta}N_i^{(0)}=0 \tag{3.2.50}$$

and

$$\partial_t^{(1)}\sum_i c_{i\alpha}N_i^{(0)}+\partial_{x_\beta}^{(1)}\sum_i c_{i\alpha}c_{i\beta}N_i^{(0)}=0 \tag{3.2.51}$$

or

$$\partial_t^{(1)}\rho+\partial_{x_\beta}^{(1)}\left(\rho u_\beta\right)=0 \quad \text{(continuity equation)} \tag{3.2.52}$$

$$\partial_t^{(1)}\left(\rho u_\alpha\right)+\partial_{x_\beta}^{(1)}P_{\alpha\beta}^{(0)}=0 \tag{3.2.53}$$

where

$$
\begin{aligned}
P_{\alpha\beta}^{(0)} &\equiv \sum_i c_{i\alpha}c_{i\beta}N_i^{eq}\\
&= \rho\delta_{\alpha\beta}+\underbrace{\rho G(\rho)T_{\alpha\beta\gamma\delta}^{(\mathrm{MA})}u_\gamma u_\delta}_{\longrightarrow\ \text{advection term}}+\mathcal{O}\left(u^4\right)
\end{aligned} \tag{3.2.54}
$$

is the momentum flux tensor in first approximation. The momentum advection tensor[23] $T^{(MA)}$ is a tensor of 4th rank

$$T_{\alpha\beta\gamma\delta}^{(\mathrm{MA})}=\sum_i c_{i\alpha}c_{i\beta}Q_{i\gamma\delta}. \tag{3.2.55}$$

It is isotropic and given by (compare Section 3.3, Eq. (3.3.10)):

$$
\begin{aligned}
T_{\alpha\beta\gamma\delta}^{(\mathrm{MA})} &= \sum_i c_{i\alpha}c_{i\beta}Q_{i\gamma\delta}=\sum_i c_{i\alpha}c_{i\beta}\left(c_{i\gamma}c_{i\delta}-\frac{1}{2}\delta_{\gamma\delta}\right)\\
&= \frac{3}{4}\left(\delta_{\alpha\gamma}\delta_{\beta\delta}+\delta_{\alpha\delta}\delta_{\beta\gamma}-\delta_{\alpha\beta}\delta_{\gamma\delta}\right).
\end{aligned}
$$

[23] The name is derived from the fact that this tensor is part of the nonlinear advection term. It occurs, however, also in the dissipative terms of the Navier-Stokes equation.

Accordingly the components of the momentum flux tensor in first approximation read

$$\begin{aligned} P_{xx}^{(0)} &= \rho G(\rho)\left(u^2 - v^2\right) + \frac{\rho}{2} \\ P_{yy}^{(0)} &= \rho G(\rho)\left(v^2 - u^2\right) + \frac{\rho}{2} \\ P_{xy}^{(0)} &= P_{yx}^{(0)} = \rho G(\rho) 2uv \end{aligned} \tag{3.2.56}$$

whereas the momentum flux tensor in the 'real world' (i.e. in the Navier-Stokes equation) is

$$\begin{aligned} P_{xx} &= \rho u^2 + p \\ P_{yy} &= \rho v^2 + p \\ P_{xy} &= P_{yx} = \rho uv. \end{aligned} \tag{3.2.57}$$

Identification of $\frac{\rho}{2}(1 - g(\rho)\boldsymbol{u}^2)$ with the pressure p leads to

$$\begin{aligned} P_{xx}^{(0)} &= \rho g(\rho) u^2 + p \\ P_{yy}^{(0)} &= \rho g(\rho) v^2 + p \\ P_{xy}^{(0)} &= P_{yx}^{(0)} = \rho g(\rho) uv \end{aligned} \tag{3.2.58}$$

which looks similar to the momentum flux tensor (3.2.57) except for the factor $g(\rho) = G(\rho)/2$. For small values of $\boldsymbol{u}^2$ the pressure is given by the 'isothermal' relation

$$p = \frac{\rho}{2} = c_s^2 \rho \tag{3.2.59}$$

with the sound speed $c_s = 1/\sqrt{2}$. It can be shown that invariance under *Galilei transformations* constrains the *g-factor* to be equal to 1 (compare Exercise 3.2.8). Here

$$g(\rho) = \frac{3-\rho}{6-\rho} \tag{3.2.60}$$

is always smaller than 1 (actually smaller than 1/2). Similar expressions apply to other lattice models. Thus this g-factor *breaks* the Galilean invariance. The deviation of g from 1 is caused by the underlying lattice which is only invariant under certain discrete translations but not under arbitrary Galilei transformations. It will be shown later that this 'disease' can be cured when other distribution functions (Boltzmann instead of Fermi-Dirac) are applied.

This is the case in lattice Boltzmann models. The occurance of Fermi-Dirac distributions in LGCA is a consequence of the exclusion principle. Therefore if one sticks to this essential feature of any LGCA the g-disease can be treated only symptomatically[24], namely by a *rescaling of time*

$$t \quad \longrightarrow \quad \frac{t}{g(\rho)}. \tag{3.2.61}$$

The equation for the incompressible regime can be derived from the compressible equation by ignoring all density variations except in the pressure term[25]. Setting ρ to the constant and uniform value ρ_0 and applying the rescaling of time leads to

$$\rho_0 g(\rho_0)\frac{\partial \boldsymbol{u}}{\partial t} + g(\rho_0)(\boldsymbol{u}\boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla}\left(\frac{\rho}{2} - \frac{\rho_0}{2}g(\rho_0)\boldsymbol{u}^2\right)$$

or

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla}P \tag{3.2.62}$$

with the kinematic pressure

$$P = \left(\frac{\rho}{2\rho_0 g(\rho_0)} - \boldsymbol{u}^2\right) \tag{3.2.63}$$

Eq. (3.2.62) is the Euler equation (Navier-Stokes without dissipation) of the FHP model. Eq. (3.2.52) is the continuity equation for the mass density ρ. It will not change when terms of order ϵ^2 are included.

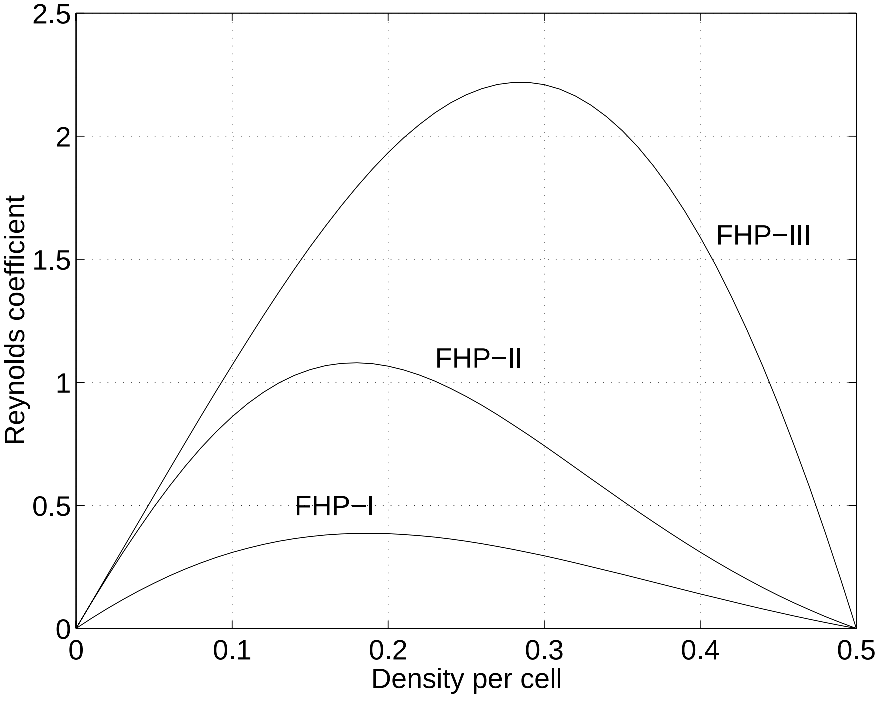
The terms of order ϵ^2 are calculated by Frisch et al. (1987) and Hénon (1987b). Adding up terms of order ϵ and ϵ^2 while neglecting terms of order ϵu^3, $\epsilon^2 u^2$ and $\epsilon^3 u$ leads in the same incompressible limit as discussed before to the Navier-Stokes equation

$$\boxed{\begin{array}{rcl} \boldsymbol{\nabla}\cdot\boldsymbol{u} & = & 0 \\ \partial_t \boldsymbol{u} + (\boldsymbol{u}\boldsymbol{\nabla})\boldsymbol{u} & = & -\boldsymbol{\nabla}P + \nu\nabla^2\boldsymbol{u} \end{array}} \tag{3.2.64}$$

where ν is the (scaled) kinematic shear viscosity ($\nu = \nu^{(u)}/g(\rho_0)$; the unscaled shear viscosity $\nu^{(u)}$ for different FHP models is given in Table 3.2.2; compare also Fig. 3.2.4). Thus we have attained the object of our desire: the incompressible Navier-Stokes equation!

Please note that the FHP models do not possess an energy equation. In the FHP-I model mass and (kinetic) energy conservation are essentially identical. In models with rest particles, some collisions do not respect conservation of

Fig. 3.2.3. *The Reynolds coefficient R_* as a function of the density per cell d for FHP-I, FHP-II and FHP-III. FHP-I: 2-particle and symmetric 3-particle collisions; FHP-II: additional collisions involving rest particles and 2-particle collisions with spectator; FHP-III: additional 4-particle collisions (collision saturated model). The Reynolds coefficient is defined as $R_* := \dfrac{R_e}{L \cdot M_a} = \dfrac{c_s g(d)}{\nu(d)}$; $R_*^{\max} = \max_d (R_*) = R_*(d_{max})$; d is the density per cell (Eq. 3.2.24), c_s sound speed, $g(d)$ density dependent g-factor, and $\nu^{(u)}$ (unscaled) kinematic shear viscosity.*



(kinetic) energy. lattice-gas cellular automata with energy equations will be discussed in Section 3.7.

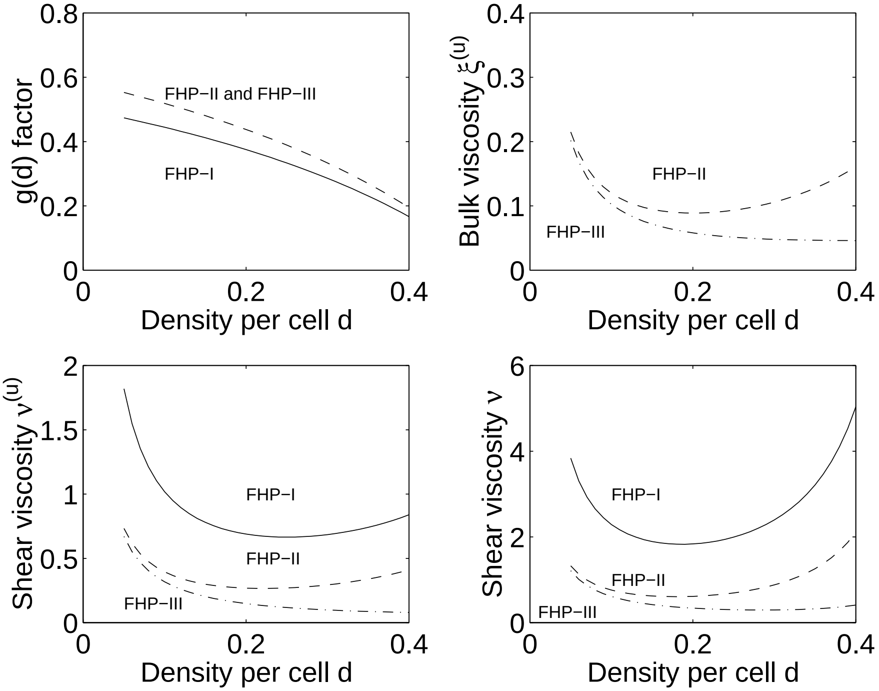
[24] The symptomatic treatment of the g-disease does not solve all problems. D'Humières et al. (1987) have shown that vorticity is advected at speed $g(d)u \neq u$. In order to fix this problem they have proposed a model with 8 bits over the FHP lattice (see Subsection 3.2.10 for details).

[25] For some fine points see Majda, 1984.

Table 3.2.2. *Analytical values for three different FHP models.* $\nu^{(u)}$ *(unscaled) kinematic shear viscosity;* $\eta^{(u)}$ *(unscaled) kinematic bulk (compressional) viscosity (taken from Frisch et al., 1987).* d *is the density per cell (Eq. 3.2.24).*

	FHP-I	FHP-II	FHP-III
Number of cells	6	7	7
c_s	$1/\sqrt{2}$	$\sqrt{3/7}$	$\sqrt{3/7}$
$g(d)$	$\frac{1}{2}\frac{1-2d}{1-d}$	$\frac{7}{12}\frac{1-2d}{1-d}$	$\frac{7}{12}\frac{1-2d}{1-d}$
$\nu^{(u)}$	$\frac{1}{12}\frac{1}{d(1-d)^3}-\frac{1}{8}$	$\frac{1}{28}\frac{1}{d(1-d)^3}\frac{1}{1-4d/7}-\frac{1}{8}$	$\frac{1}{28}\frac{1}{d(1-d)}\frac{1}{1-8d(1-d)/7}-\frac{1}{8}$
$\eta^{(u)}$	0	$\frac{1}{98}\frac{1}{d(1-d)^4}-\frac{1}{28}$	$\frac{1}{98}\frac{1}{d(1-d)}\frac{1}{1-2d(1-d)}-\frac{1}{28}$
$R_*^{\max}$	0.387	1.08	2.22
d_{max}	0.187	0.179	0.285

Fig. 3.2.4. *The g-factor, unscaled kinematic bulk viscosity ($\xi^{(u)}$), and the unscaled ($\nu^{(u)}$) and scaled (ν) kinematic shear viscosity as functions of the density per cell (d) for the models FHP-I, FHP-II and FHP-III (compare Table 3.2.2). The bulk viscosity vanishes in FHP-I.*



Exercise 3.2.8. (**)

Show that the generalized (g-factor) substantial derivative

$$\frac{\partial \boldsymbol{u}}{\partial t} + g\,(\boldsymbol{u}\boldsymbol{\nabla})\,\boldsymbol{u} \tag{3.2.65}$$

is invariant under Galilei transformations if and only if $g \equiv 1$.
Hint: it will be sufficient to consider the special Galilei transformation

$$\begin{aligned} x' &= x + ct; \quad c = const \\ y' &= y \\ z' &= z \\ t' &= t \end{aligned} \tag{3.2.66}$$

Exercise 3.2.9. (**)
Calculate the components of the momentum flux tensor $P^{(0)}_{\alpha\beta}$ for HPP from the formula

$$\rho G(\rho) T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta} + \frac{c^2}{D}\rho\delta_{\alpha\beta}. \tag{3.2.67}$$

What's wrong with the advection term?

3.2.7 Boundary conditions

The coding of boundary conditions (BC) is an essential part of the LGCA (and any other numerical) method. There are at least five different types of BC:

1. *Periodic* BC are often used even if it is not realistic because they are so easy to code.
2. *Inflow* BC (example: channel flow).
3. *Outflow* BC (example: channel flow) can be very difficult to deal with, especially when waves try to leave the model domain (compare, for example, Orlanski, 1976, Røed and Smedstad, 1984, and Stevens, 1991).
4. *No-slip* BC ($\boldsymbol{u} = 0$) apply to solid boundaries (walls, obstacles).
5. *Slip* BC, i.e. the velocity component normal to the boundary and the normal derivative of the tangential component vanish ($u_n = 0$ and $\partial u_t/\partial n = 0$), apply to solid boundaries where the frictional force adjacent to the wall is not resolved.

Even when constraints are formulated only for the velocity one usually also requires conservation of mass[26]. Whereas in channel flows a small violation of mass conservation could be tolerable, because each fluid element leaves the domain after some time anyway, such a violation is not acceptable for flows in closed domains where a small but steady leakage, for example, would lead to an empty basin after a while.

The coding of the first two types of BC is obvious not only for FHP but for any kind of LGCA or LBM and therefore needs no further comment. Since the beginning of simulations with FHP the following heuristic procedures for the implementation of no-slip and slip conditions are used (d'Humières and Lallemande, 1987):

– No-slip: Collision on boundary points are skipped. Instead the incoming particles are turned around by 180°. In the next propagation step they will

[26] However, momentum is most often not conserved at the boundaries.

leave the node in their former incoming direction. The mean value over in-state and out-state yields $\boldsymbol{u} = 0$. This flipping of the incoming particles is also called *bounce-back rule.* The kinetic equation for this implementation of the no-slip condition reads

$$n_i(\boldsymbol{x}_b + \boldsymbol{c}_i, t + 1) = n_{i+3}(\boldsymbol{x}_b, t)$$

where $\boldsymbol{x}_b$ is a boundary point and as usual the indices are understood modulo 6.

This *node-oriented implementation* of the no-slip BC has the advantage that it is independent of the orientation of the wall (for alternative implementations see Rem and Somers, 1989, Cornubert et al., 1991, and Ziegler, 1993).

- Slip: Collisions on boundary points are skipped. Instead the incoming particles are reflected like a light ray (*specular reflection*) at the wall. The mean value over in-state and out-state yields $u_n = 0$ and the tangential momentum of the particles is conserved. The kinetic equation for this implementation of the no-slip condition reads

$$n_i(\boldsymbol{x}_b + \boldsymbol{c}_i, t + 1) = n_{i-3}(\boldsymbol{x}_b, t).$$

Only several years later Cornubert et al. (1991) have investigated discrepancies of the distribution functions in the boundary layer due to these kind of implementations of boundary conditions. For FHP they found anisotropic Knudsen layers adjacent to obstacles. The effective boundary of obstacles is not identical with the node locations but lies somewhat outside the outer nodes of the obstacle. The precise location depends on the direction (anisotropic) and is smaller than the distance to the next neighbor node in the fluid region. Especially for small obstacles this effect has to be taken into account.

3.2.8 Inclusion of body forces

In principle body forces, i.e. a change of momentum, can be applied on the macroscopic or microscopic level. The macroscopic method, however, requires at each time step the calculation of mean values (coarse graining), change of the momentum $\boldsymbol{j}$ and re-initilization. Of course this procedure is computationally much too demanding. Thus only a microscopic method is feasable.

Body forces $\boldsymbol{F}$ which may vary in space and time but are independent of the flow velocity $\boldsymbol{u}$ can be realized by flipping particles with velocity $-\boldsymbol{c}_i$ into particles with velocity $\boldsymbol{c}_i$ (for forces parallel to $\boldsymbol{c}_i$). The probability of this flipping has to be proportional to the magnitude of $\boldsymbol{F}$.

The following results are from Appendix D in Frisch et al. (1987). Boolean transition variables $\xi'_{ss'}$ are defined such that their mean values

$$\langle \xi'_{ss'} \rangle = B\,(s \to s') \tag{3.2.68}$$

are a set of transition probabilities associated to the body-force. The transition probabilities satisfy normalization

$$\sum_{s'} B(s \to s') = 1 \tag{3.2.69}$$

and mass conservation

$$\sum_i (s'_i - s_i) B(s \to s') = 0 \quad \forall s, s', \tag{3.2.70}$$

but *do not* satisfy momentum conservation (which is desired of course!), semi-detailed balance and $\mathcal{G}$-invariance. In case I (body-force $\boldsymbol{f}$ independent of velocity) they are further constrained by

$$\boldsymbol{f} = \sum_{s,s',i} \boldsymbol{c}_i (s'_i - s_i) B(s \to s') \left(\frac{d}{1-d}\right)^p (1-d)^b \tag{3.2.71}$$

where $p = \sum_j s_j$ and b is the number of cells per node. If $\boldsymbol{f}$ is space and/or time dependent, so are the $B(s \to s')$'s.

In case of a force linear in velocity

$$f_\alpha = C_{\alpha\beta} U_\beta \tag{3.2.72}$$

the $B(s \to s')$'s are constrained by

$$0 = \sum_{s,s',i} \boldsymbol{c}_i (s'_i - s_i) B(s \to s') \left(\frac{d}{1-d}\right)^p (1-d)^b, \quad p = \sum_j s_j, \tag{3.2.73}$$

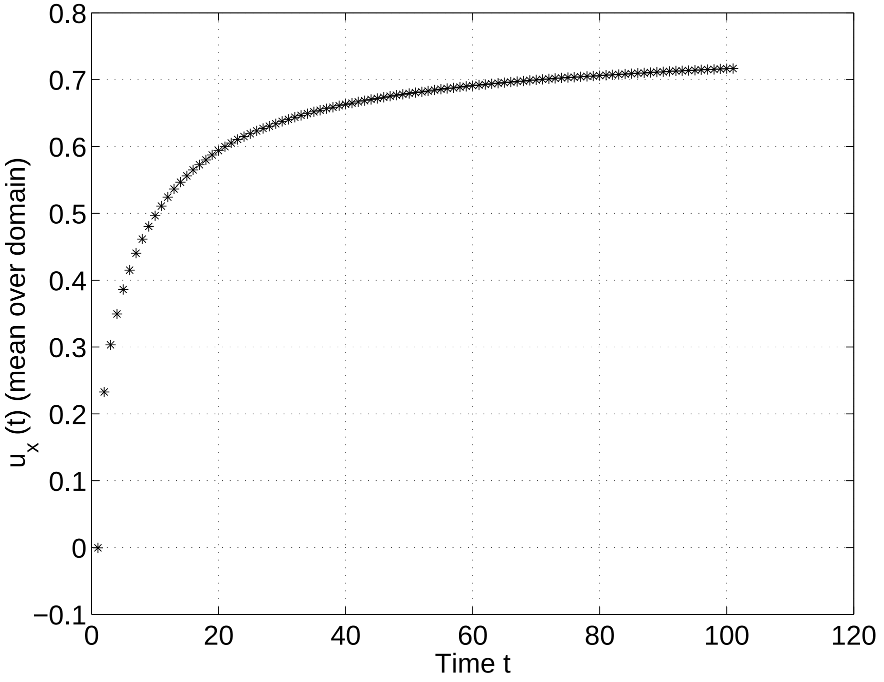
and

$$C_{\alpha\beta} = \frac{D}{c^2} (1-d)^{b-1} \sum_{s,s',i} c_{i\alpha} (s'_i - s_i) B(s \to s') \left(\frac{d}{1-d}\right)^p \sum_j s_j c_{j\beta} \tag{3.2.74}$$

where $p = \sum_j s_j$.

In order to illustrate the method let us consider the simplest example, namely time-independent homogeneous forcing in the direction of a particular lattice velocity, say in x-direction. We can flip particles with velocity $-\boldsymbol{c}_6 = \boldsymbol{c}_3 = (-1, 0)$ into particles with velocity $\boldsymbol{c}_6 = (1, 0)$ whenever this is possible while leaving all other particles unchanged (compare Fig. 3.2.5). Already after one time step the domain mean x-velocity, $u_x(t)$, increased from zero (its initial value) to approximately 0.2. This extreme acceleration leads to high Mach numbers after a few time steps. Thus in simulations of incompressible flow problems the flipping rate has to be lowered.

Fig. 3.2.5. *Inclusion of body-forces: The plot shows the increase of domain mean x-velocity, $u_x(t)$, due to flipping of particles with velocity $-\boldsymbol{c}_6 = \boldsymbol{c}_3 = (-1, 0)$ into particles with velocity $\boldsymbol{c}_6 = (1, 0)$ whenever this is possible (FHP-I, $d = 0.3$).*

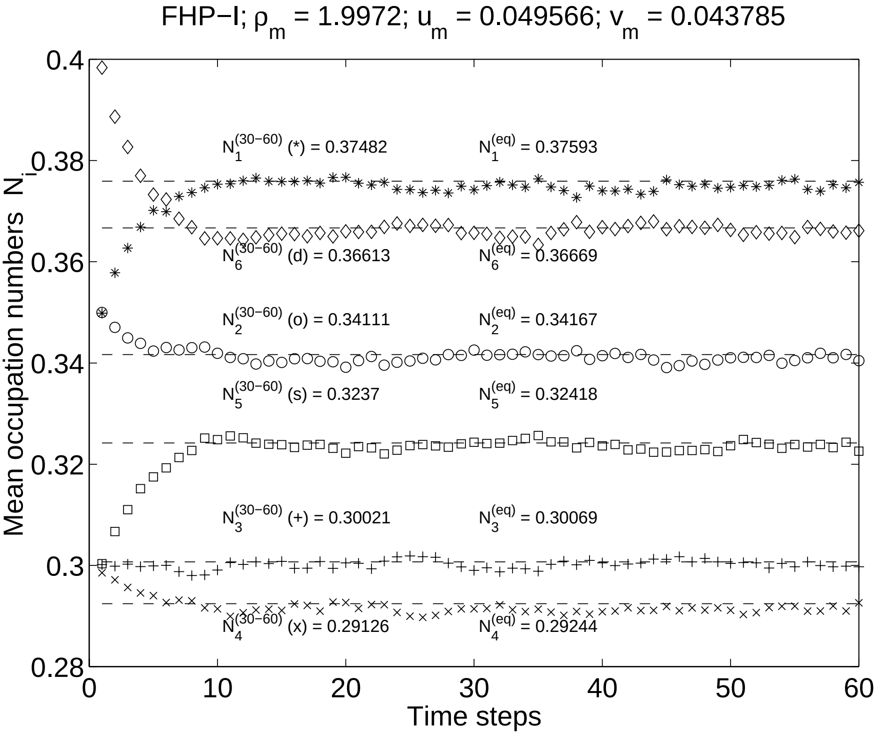


3.2.9 Numerical experiments with FHP

Here the results of some numerical calculations will be discussed. The code written in C is accessible (see the web address given in the Preface).

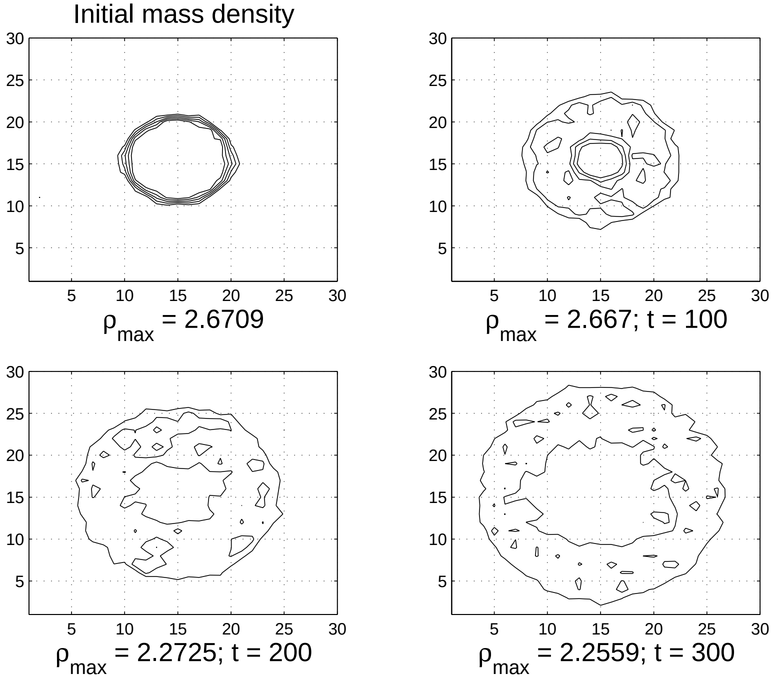
1. Relaxation toward equilibrium (compare program 'exper1.c'): The FHP-I model is initialized with a distribution which is spatially homogenous but far from equilibrium. At a density $\rho \approx 2$ enough collisions occur in order to drive the occupation numbers toward their equilibrium values (compare Fig. 3.2.6).

Fig. 3.2.6. *Relaxation toward equilibrium for FHP-I: The model is initialized with a distribution which is spatially homogenous but far from equilibrium. The domain size is* 320 *times* 320 *nodes. For simplicity periodic boundary conditions are applied. At a density* $\rho \approx 2$ *enough collisions occur in order to drive the occupation numbers toward their equilibrium values (dashed lines) which are almost reached after* 10 *time steps. The mean values over the time levels* 30 *to* 60 *compares very well with the theoretical values according to equilibrium distribution (3.2.37).*



2. Propagation of *sound waves*: Here sound waves are excited by a density perturbation (compare Fig. 3.2.7 upper left). The waves propage isotropically with a speed of $c_s = 1/\sqrt{2}$.

Fig. 3.2.7. *Propagation of sound waves: The FHP-I model is initialized with vanishing velocity and constant density (≈ 2) except for a positive radial symmetric density perturbation near the center of a domain with* 960 *times* 960 *nodes. The plots show the mean densities calculated over subdomains (macrocells) of* 32 *times* 32 *nodes at four different time levels. The values of the contour lines are always* $2.1, 2.2, 2.3, 2.4, 2.5$. *The propagation of the density perturbation is isotropic and the speed of propagation is consistent with the sound speed* $c_s = 1/\sqrt{2}$ *(in 300 time steps the sound will propagate over a distance of* $300/32/\sqrt{2} \approx 13$ *sidelengths of macrocells).*



3. Flow past an obstacle: Our next goal is the simulation of the flow past an obstacle. Wether the flow is laminar or turbulent depends on the Reynolds number R_e which is defined by

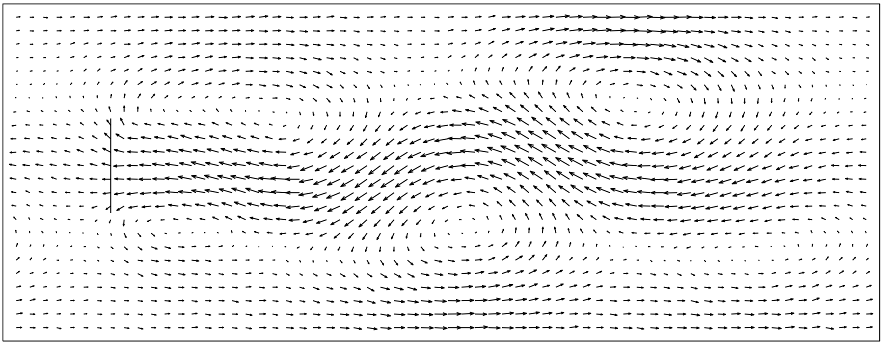
$$R_e := \frac{U \cdot L}{\nu} \tag{3.2.75}$$

where U is a characteristic flow speed (usually the flow speed far upstream of the obstacle), L is a characteristic spatial scale of the obstacle (for an obstacle in the form of a circular cylinder, for example, L is the radius or diameter of the cross-section), and ν is the kinematic shear viscosity. From experiments it is known that eddies are formed and shedded when the Reynolds number becomes larger than about 50. A so-called von Karman vortex street will build up (compare the beautiful pictures in the book of van Dyke, 1982).
How can one simulate a flow with $R_e \approx 90$ with FHP-II? The flow speed has to be small compared to the sound speed $c_s^{(\text{FHP-II})} = \sqrt{3/7} \approx 0.65$. So $u = 0.2$ is a good value. The density per cell d should be below 0.5 (the scaling factor $g(d)$ vanishes at $d = 0.5$) and is chosen here as $d = 2/7$ which results in a density per node $\rho = 2$. The scaled kinematic viscosity at this density is approximately 0.8. The only free parameter is the size of the obstacle L which can be calculated from

$$L = \frac{R_e \cdot \nu}{u} = \frac{88 \cdot 0.88}{0.2} = 388 \tag{3.2.76}$$

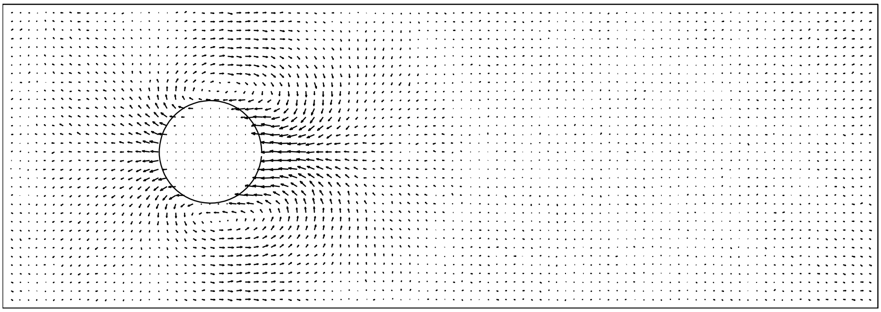
(see results in Fig. (3.2.8)).

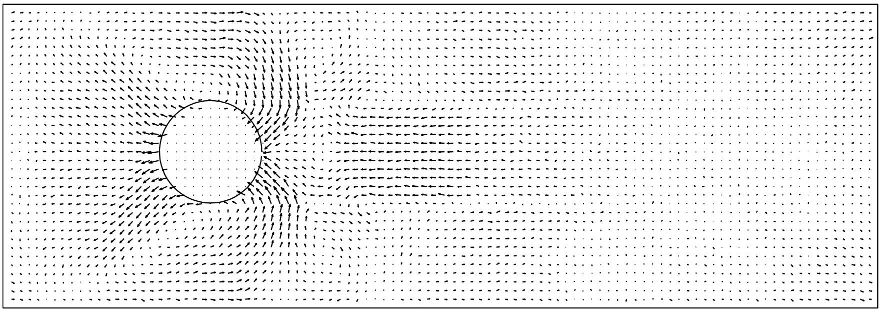
Fig. 3.2.8. *Flow past a plate. The FHP-II model is initialized with a homogenous flow in x-direction with speed* $u = 0.2$ *(in lattice units). The lattice encompasses* 4096 *times* 1792 *nodes. The width of the plate is* 388. *At a density per node* $\rho = 2.1$ *the (scaled) kinematic viscosity is* 0.88. *Thus the Reynolds number is* 88. *The figure shows the flow minus the mean flow velocity after* $t = 260,000$ *time steps.*



Please note that the type of boundary conditions are essential. If one applies periodic boundary conditions the simulation addresses flow past a periodic array of cylinders. In contrast to flow past a single cylinder the flow may be steady even at a Reynolds number of 100 (Gallivan et al., 1997).

Fig. 3.2.9. *Flow past a circular cylinder. Lattice with* 6400 *times* 2560 *nodes. The uppermost plot shows the result after* 20000 *time steps for the FHP-I model (the flow velocity averaged over the whole domain has been substracted in both plots in order to make the pertubations better visible). If* 3*-particle collisions are left out the flow field looks quite different (lower plot).*





Exercise 3.2.10. (*)
Consider typical flow velocities in the atmosphere and oceans. At which spatial scales the Reynolds numbers are 1, 1000 and 10^6?

3.2.10 The 8-bit FHP model

D'Humières et al. (1987) proposed a model with two populations (n_0, m_0) that represent rest particles. With 2 bits up to 3 rest particles can be described: (n_0, m_0) = (0,0) no, (1,0) one, (0,1) two and (1,1) three rest particles. All collisions conserving mass and momentum are included ('collision saturated'). The collisions a) to c) in Table 3.2.2 proceed independent of the number of rest particles. The collisions leading to creation and destruction of rest particles are all included, except a few cases which take place with probability x, y, or z (compare Tables 3.2.10 and 3.2.11). It can be shown that for $x = 0.5$, $y = z = 0.2$, there exists a value for d for which $g(d) = 1$ and $\mathrm{d}g(d)/\mathrm{d}d = 0$. D'Humières et al. recommend a model with $x = 1/2$, $y = z = 1/6$ which gives $g = 1.0$ at $d = 0.21$. The model violates semi-detailed balance (compare Exercise 3.2.11). Numerical experiments show that vorticity is advected close to the flow speed U_0. The 8-bit-FHP model is probably already too complex to allow coding with bit-operators. The use of look-up tables makes it much more clumpsy than the 7-bit-FHP models.

Exercise 3.2.11. (*)
Show that the 8-bit FHP model of d'Humières et al. (1987) violates semi-detailed balance.

Exercise 3.2.12. (***)
Repeat the simulation of the flow past a cylinder with the FHP-II model. Compare the computational time for constant Reynolds number by varying the upstream velocity or the size of the system. Analyze the differences in the flow pattern when applying the random or the chiral two-particle head-on collision.

Fig. 3.2.10. *FHP-8-bit model: collisions involving rest particles; occupied cells are represented by arrows, empty cells by thin lines, the number of rest particles is indicated by the number in the central circle.*

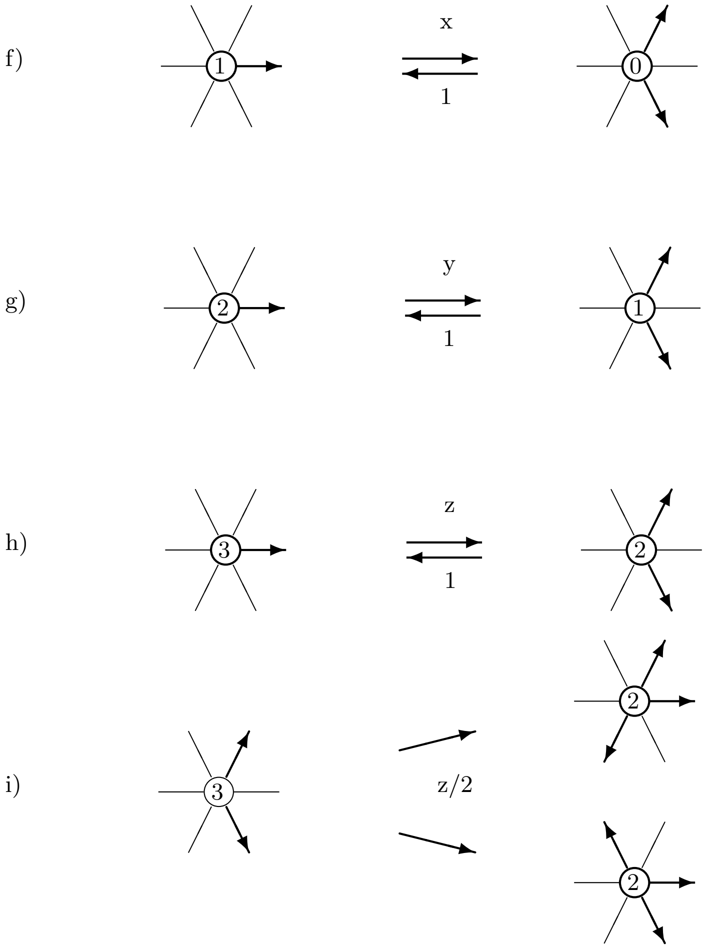
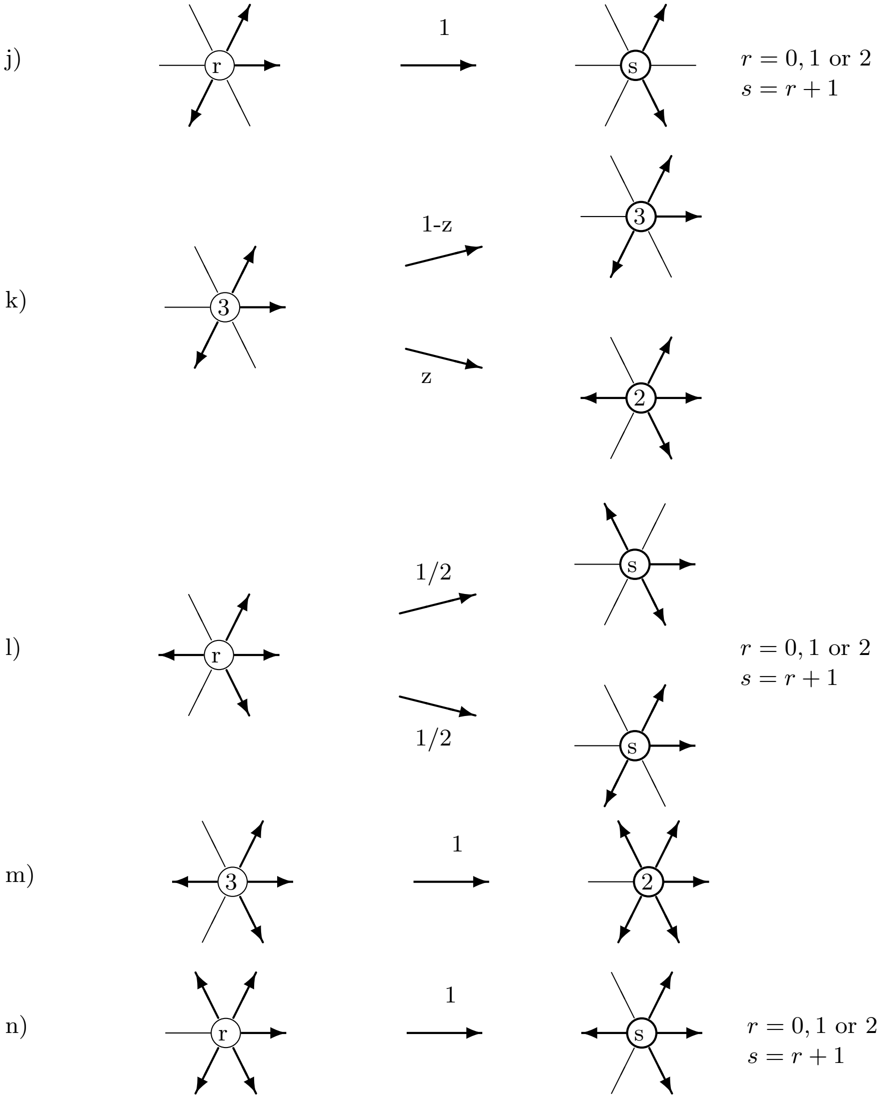


Fig. 3.2.11. *FHP-8-bit model: collisions involving rest particles (continued); occupied cells are represented by arrows, empty cells by thin lines, the number of rest particles is indicated by the number in the central circle.*



3.3 Lattice tensors and isotropy in the macroscopic limit

All lattice-gas cellular automata are based on extremely discretized phase spaces. Nevertheless it is to be expected that the spatial discretization will be smoothed out on scales which are much larger than the grid spacing[27]*. Things are less obvious for the discretization of the velocities. Calculation of mean values over an angular direction is restricted to a range of 2π which contains only a few velocities (six for FHP). The multi-spin coding yields tensors which consist of the components of the lattice velocities. These tensors are invariant with respect to elements of the associated finite symmetry group but in general not with respect to arbitrary orthogonal transformations (including continuous rotations). A sufficient condition for 'reasonable' macroscopic equations encloses the isotropy of lattice tensors (to be defined below) of 2nd and 4th rank. The lattice tensors with odd rank vanish because of the symmetry of the lattices.*

Group theoretical methods allow quite general propositions concerning the isotropy of tensors of this special form (Wolfram, 1986). Discussion of group theoretical concepts is outside the scope of this work. Explicit expressions for the most general isotropic tensors will be given and lattice tensors and generalized lattice tensors for the various lattice-gas cellular automata will be calculated. Note that the results apply also to the lattice Boltzmann models discussed later on in Section 5.

3.3.1 Isotropic tensors

Definition: A tensor $T_{\alpha_1\alpha_2...\alpha_n}$ of nth rank is called *isotropic* if it is invariant with respect to arbitrary orthogonal transformations O (rotations and reflections)

$$T_{\alpha_1\alpha_2...\alpha_n} = T_{\beta_1\beta_2...\beta_n} O_{\alpha_1\beta_1} O_{\alpha_2\beta_2} ... O_{\alpha_n\beta_n}. \tag{3.3.1}$$

The most general isotropic tensors up to 4th rank are provided by the following theorem.

Theorem 3.3.1. *(Jeffreys and Jeffreys, 1956; Jeffreys, 1965)*

1. *There are no isotropic tensors of rank* 1 *(vectors).*
2. *An isotropic tensor of rank* 2 *is proportional to $\delta_{\alpha\beta}$.*
3. *An isotropic tensor of rank* 3 *is proportional to $\epsilon_{\alpha\beta\gamma}$*[28]*.*

[27] There is still the problem of a selected reference system which violates Galilean invariance and which can produce problems in the macroscopic limit.

[28] *Levy-Civita symbol* $\epsilon_{\alpha\beta\gamma}$: $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$, $\epsilon_{132} = \epsilon_{321} = \epsilon_{213} = -1$, and zero otherwise.

4. There are three different (linear independent) tensors of rank 4

$$\delta_{\alpha\beta}\delta_{\gamma\delta}, \quad \delta_{\alpha\gamma}\delta_{\beta\delta}, \quad \delta_{\alpha\delta}\delta_{\beta\gamma},$$

which can be combined to the most general form

$$T_{\alpha\beta\gamma\delta} = a\delta_{\alpha\beta}\delta_{\gamma\delta} + b\delta_{\alpha\gamma}\delta_{\beta\delta} + c\delta_{\alpha\delta}\delta_{\beta\gamma}, \tag{3.3.2}$$

where a, b and c are arbitrary constants.

A proof of the theorem can be found in Jeffreys (1965).
Isotropic tensors of rank $n \geq 4$ consist only of products of second rank δ tensors (for example: $\delta_{\alpha\beta}\delta_{\gamma\delta}\delta_{\epsilon\zeta}$ and all tensors that result from cyclic permutation of indices) when n is even or of products of δ and ϵ tensors when n is odd.

In two dimensions the isotropic tensor of rank 4 (3.3.2) has the following non-vanishing components:

$$\begin{array}{rcl} T_{1111} = T_{2222} & = & a+b+c, \\ T_{1122} = T_{2211} & = & a, \\ & & \\ T_{1212} = T_{2121} & = & b, \\ T_{1221} = T_{2112} & = & c. \end{array} \tag{3.3.3}$$

In particular, the tensor $\delta_{\alpha\beta\gamma\delta}$ is non-isotropic ($\delta_{\alpha\beta\gamma\delta}$ is 1 if all indices are equal and 0 otherwise; it is a generalization of the Kronecker symbol $\delta_{\alpha\beta}$). The same is true for $\delta_{\alpha\beta\gamma\delta\epsilon\zeta}$.

3.3.2 Lattice tensors: single-speed models

Let us define the following tensors of rank n

$$\boxed{L_{\alpha_1\alpha_2...\alpha_n} = \sum_i c_{i\alpha_1}c_{i\alpha_2}...c_{i\alpha_n}} \tag{3.3.4}$$

where $c_{i\alpha_\nu}$ are the cartesian components of the lattice velocities $\boldsymbol{c}_i$. We will call these tensors the *lattice tensors*. Because of their special structure they are invariant with respect to the symmetry group of the lattice and they are symmetric in all of their indices. From these symmetries it follows that these tensors can have a maximal number N of independent components of

$$N = \binom{n+D-1}{n} = \frac{(n+D-1)!}{n!(D-1)!}$$

where D is the (spatial) dimension. Example: A symmetric tensor T of rank 2 in two dimensions has at most $N = \binom{3}{2} = \frac{3!}{2!1!} = 3$ independent components:

$$T = \begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

In lattice-gas cellular automata with one speed (HPP, FHP, FCHC) the momentum advection tensor (MAT) of 4th rank $T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta}$ (compare Eq. 3.2.55) occurs in the macroscopic form of the momentum balance. It can be rewritten in terms of the lattice tensors of rank two and four:

$$\begin{aligned} T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta} &= \sum_i c_{i\alpha} c_{i\beta} Q_{i\gamma\delta} \\ &= \sum_i \left[c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} - \frac{1}{2} c_{i\alpha} c_{i\beta} \delta_{\gamma\delta} \right] \\ &= L_{\alpha\beta\gamma\delta} - \frac{1}{2} L_{\alpha\beta} \delta_{\gamma\delta}. \end{aligned} \tag{3.3.5}$$

A sufficient condition for the isotropy of $T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta}$ is the isotropy of $L_{\alpha\beta}$ and $L_{\alpha\beta\gamma\delta}$. $T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta}$ is non-isotropic if $L_{\alpha\beta}$ is isotropic while $L_{\alpha\beta\gamma\delta}$ is non-isotropic. Other combinations do not occur in the models considered. In what follows we use the notation $DkQb$ of Qian et al. (1992) where k is the spatial dimension and b is the number of lattice velocities (including $\boldsymbol{c}_0 = 0$ for rest particles).

Square lattice: HPP (D2Q4). Lattice velocities:

$$\boldsymbol{c}_i = \left(\cos \frac{2\pi i}{4}, \sin \frac{2\pi i}{4} \right) \qquad i = 1, 2, 3, 4.$$

The lattice tensor of rank 2

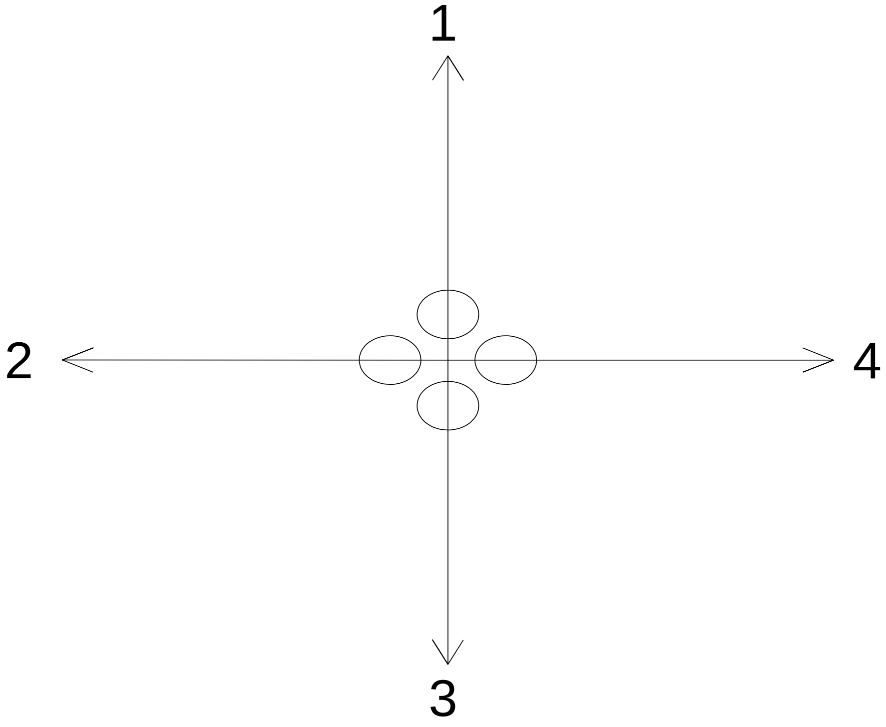
$$L^{HPP}_{\alpha\beta} = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 2\delta_{\alpha\beta}$$

is isotropic.
The lattice tensor of rank 4

$$L^{HPP}_{\alpha\beta\gamma\delta} = 2\, \delta_{\alpha\beta\gamma\delta} \tag{3.3.6}$$

is non-isotropic. As a consequence the HPP model fails to yield the Navier-Stokes equations in the macroscopic limit.

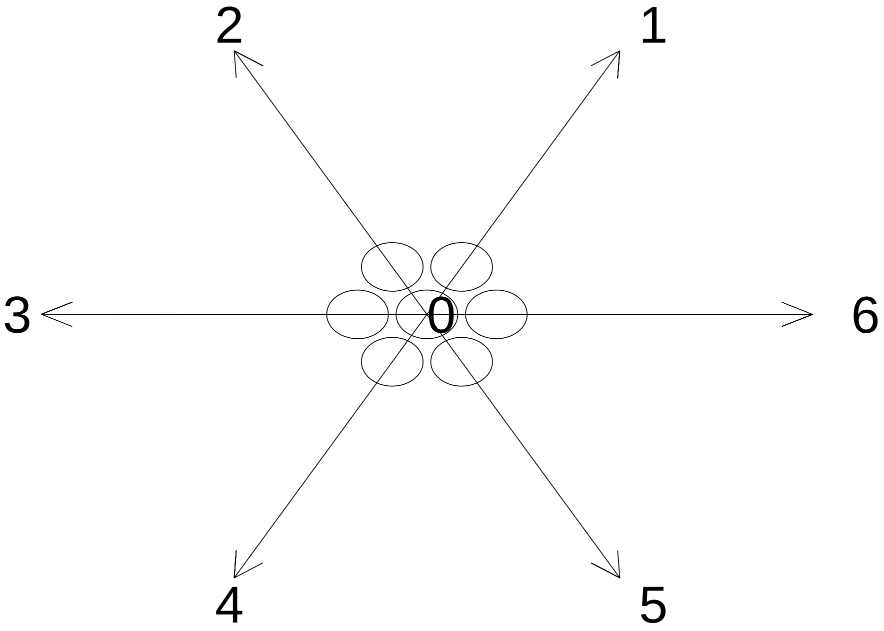
Fig. 3.3.1. *The lattice velocities of the HPP (D2Q4) lattice.*



Triangular lattice: FHP (D2Q7). The triangular FHP lattice has hexagonal symmetry. Lattice velocities:

$$\begin{aligned} \boldsymbol{c}_0 &= (0,0) \\ \boldsymbol{c}_i &= \left(\cos\frac{2\pi i}{6}, \sin\frac{2\pi i}{6}\right) \quad i=1,...,6 \end{aligned} \tag{3.3.7}$$

Fig. 3.3.2. *The lattice velocities of the FHP (D2Q7) lattice.*



The lattice tensors of rank 2

$$L^{FHP}_{\alpha\beta} = 3\,\delta_{\alpha\beta} \tag{3.3.8}$$

and rank 4

$$L^{FHP}_{\alpha\beta\gamma\delta} = \frac{3}{4}\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right) \tag{3.3.9}$$

are isotropic. The momentum advection tensor reads

$$T^{(\mathrm{MA})}_{\alpha\beta\gamma\delta} = \frac{3}{4}\left(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma} - \delta_{\alpha\beta}\delta_{\gamma\delta}\right). \tag{3.3.10}$$

Thus, FHP yields the Navier-Stokes equation in the macroscopic limit.

FCHC (D4Q24). The investigation of the lattice tensors of the FCHC model is left to the reader (Exercise 3.3.1).

Exercise 3.3.1. (**)
The 24 lattice velocities of the FCHC model are given by

$$\begin{gathered}(\pm1,\pm1,0,0),\quad (\pm1,0,\pm1,0),\quad (0,\pm1,\pm1,0),\\ (\pm1,0,0,\pm1),\quad (0,\pm1,0,\pm1),\quad (0,0,\pm1,\pm1).\end{gathered}$$

Calculate the lattice tensors of rank 2 and 4 and show that they are isotropic.

3.3.3 Generalized lattice tensors for multi-speed models

Later on multi-speed[29] lattice-gas cellular automata and multi-speed lattice Boltzmann models will be discussed. The associated lattice tensors of rank 4 are usually non-isotropic because the symmetry group of the corresponding lattices is not large enough (note that the situation is different for the multi-speed FHP model where the symmetry of each single-speed sub-lattice is large enough). But isotropy of 4th rank tensors can be recovered by introducing weights w_i for the different speeds. These *generalized lattice tensors*

$$\boxed{G_{\alpha_1\alpha_2\ldots\alpha_n} = \sum_i w_i c_{i\alpha_1} c_{i\alpha_2} \ldots c_{i\alpha_n}} \tag{3.3.11}$$

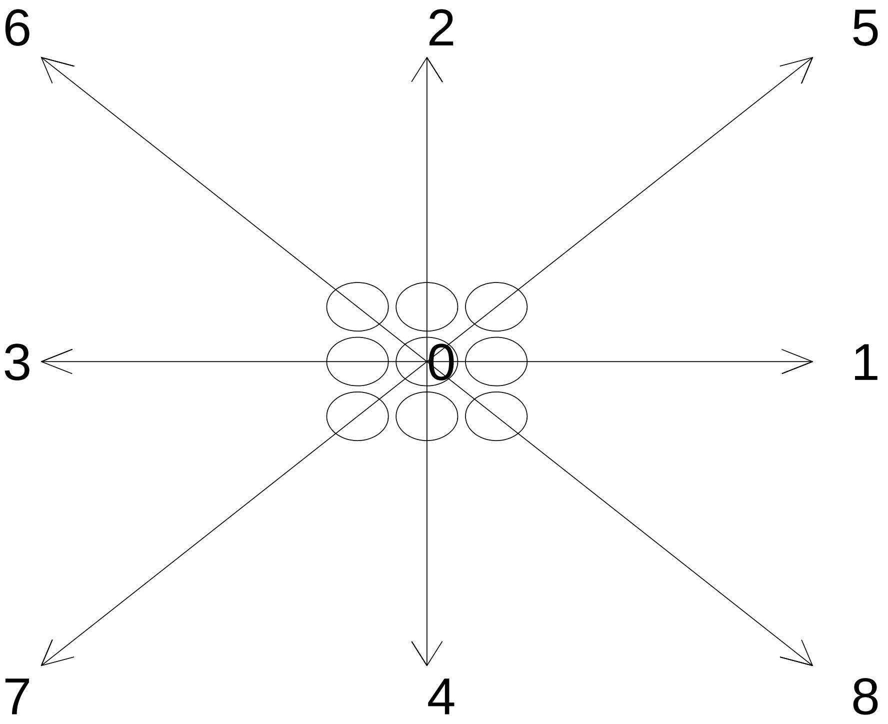
occur naturally in the multi-scale analysis of multi-speed models. The weights correspond to different occupation numbers for the different speeds in the global equilibrium with vanishing macroscopic velocities.
The following sub-sections can be skipped in the first reading and should be revisited when the appropriate multi-speed model is discussed.

[29] Models with several different speeds have been encountered before, namely the FHP variants with rest particles (FHP-II, FHP-III). Particles with vanishing speed have no influence on the isotropy of the lattice tensors. Therefore only models with different *non-vanishing* speeds are called multi-speed models.

D2Q9. Lattice velocities:

$$\begin{aligned} \boldsymbol{c}_0 &= (0,0), \\ \boldsymbol{c}_{1,3}, \boldsymbol{c}_{2,4} &= (\pm 1, 0), \quad (0, \pm 1), \\ \boldsymbol{c}_{5,6,7,8} &= (\pm 1, \pm 1). \end{aligned} \tag{3.3.12}$$

Fig. 3.3.3. *The lattice velocities of the D2Q9 lattice.*



The lattice tensor of rank 2

$$L^{(D2Q9)}_{\alpha\beta} = 6\delta_{\alpha\beta}$$

is isotropic. The lattice tensor of rank 4 with the following non-vanishing components

$$\begin{aligned} L^{(D2Q9)}_{1111} = L^{(D2Q9)}_{2222} &= 6 \\ L^{(D2Q9)}_{1122} = L^{(D2Q9)}_{2211} &= 4 \\ L^{(D2Q9)}_{1212} = L^{(D2Q9)}_{2121} &= 4 \\ L^{(D2Q9)}_{1221} = L^{(D2Q9)}_{2112} &= 4 \end{aligned}$$

is non-isotropic.

Introducing the weights $w_i = 1$ for speed 1 ($i = 1, ..., 4$) and $w_i = 1/4$ for speed $\sqrt{2}$ ($i = 5, ..., 8$) leads to isotropic generalized lattice tensors of rank 2

$$G_{\alpha\beta}^{(D2Q9)} = 3\delta_{\alpha\beta} \tag{3.3.13}$$

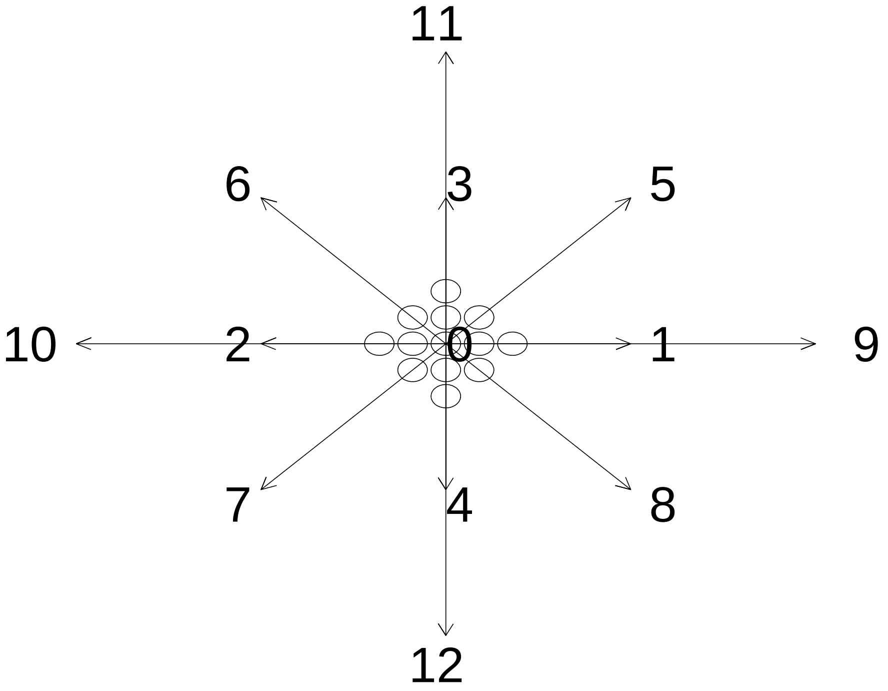
and rank 4

$$G_{\alpha\beta\gamma\delta}^{(D2Q9)} = \delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}. \tag{3.3.14}$$

D2Q13-WB. Weimar and Boon (1996)

$$\begin{array}{rcll} \boldsymbol{c}_0 & = & (0,0) & \text{rest particle} \\ \boldsymbol{c}_{1,2},\, \boldsymbol{c}_{3,4} & = & (\pm 1, 0),\, (0, \pm 1) & \text{1-particles} \\ \boldsymbol{c}_{5,6,7,8} & = & (\pm 1, \pm 1) & \sqrt{2}\text{-particles} \\ \boldsymbol{c}_{9,10},\, \boldsymbol{c}_{11,12} & = & (\pm 2, 0),\, (0, \pm 2) & \text{2-particles} \end{array} \tag{3.3.15}$$

Fig. 3.3.4. *The lattice velocities of the D2Q13-WB lattice.*



The lattice tensor of rank 2

$$L_{\alpha\beta}^{(D2Q13-WB)} = 14\,\delta_{\alpha\beta}$$

is isotropic whereas the lattice tensor of rank 4

$$L_{\alpha\beta\gamma\delta}^{(D2Q13-WB)} = 4\,(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) + 26\,\delta_{\alpha\beta\gamma\delta}$$

is non-isotropic. The weights $w_i = 4$ for speed 1, $w_i = 5$ for speed $\sqrt{2}$, and $w_i = 1$ for speed 2 lead to isotropic generalized lattice tensors of rank 2

$$G_{\alpha\beta}^{(D2Q13-WB)} = 36\,\delta_{\alpha\beta} \tag{3.3.16}$$

and rank 4

$$G_{\alpha\beta\gamma\delta}^{(D2Q13-WB)} = 20\,(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})\,. \tag{3.3.17}$$

D2Q21. The D2Q21 lattice was introduced by Fahner (1991).

$$\begin{array}{rcll}
\boldsymbol{c}_0 & = & (0,0) & \text{rest particle} \\
\boldsymbol{c}_{1,2},\ \boldsymbol{c}_{3,4} & = & (\pm1,0),\ (0,\pm1) & \text{1-particles} \\
\boldsymbol{c}_{5,6,7,8} & = & (\pm1,\pm1) & \sqrt{2}\text{-particles} \\
\boldsymbol{c}_{9,10},\ \boldsymbol{c}_{11,12} & = & (\pm2,0),\ (0,\pm2) & \text{2-particles} \\
\boldsymbol{c}_{13,\ldots,16},\ \boldsymbol{c}_{17,\ldots,20} & = & (\pm2,\pm1),\ (\pm1,\pm2) & \sqrt{5}\text{-particles}
\end{array} \tag{3.3.18}$$

The lattice tensor of rank 2

$$L_{\alpha\beta}^{(D2Q21)} = 34\,\delta_{\alpha\beta}$$

is isotropic whereas the lattice tensor of rank 4

$$L_{\alpha\beta\gamma\delta}^{(D2Q21)} = 36\ (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) - 2\,\delta_{\alpha\beta\gamma\delta}$$

is non-isotropic. The weights $w_i = 2$ for speed 1 and $w_i = 1$ otherwise lead to isotropic generalized lattice tensors of rank 2

$$G_{\alpha\beta}^{(D2Q21)} = 36\,\delta_{\alpha\beta} \tag{3.3.19}$$

and rank 4

$$G_{\alpha\beta\gamma\delta}^{(D2Q21)} = 36\ (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \tag{3.3.20}$$

D3Q15. Multi-speed lattice Boltzmann models in 3D will be defined in Section 5.3. The lattice D3Q15 has the following lattice velocities

$$\begin{array}{rcll}
\boldsymbol{c}_0 & = & (0,0,0) & \text{rest particle} \\
\boldsymbol{c}_{1,2},\ \boldsymbol{c}_{3,4},\ \boldsymbol{c}_{5,6} & = & (\pm2,0,0),\ (0,\pm2,0)\,(0,0,\pm2) & \text{2-particles} \\
\boldsymbol{c}_{7,\ldots,14} & = & (\pm1,\pm1,\pm1) & \sqrt{3}\text{-particles}
\end{array} \tag{3.3.21}$$

Fig. 3.3.5. *The lattice velocities of the D2Q21 lattice.*

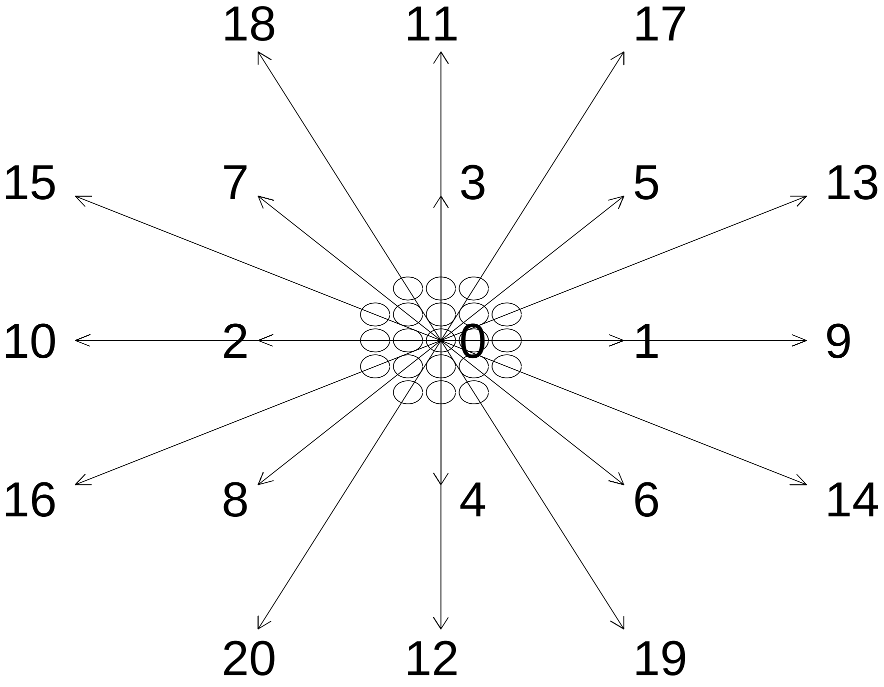
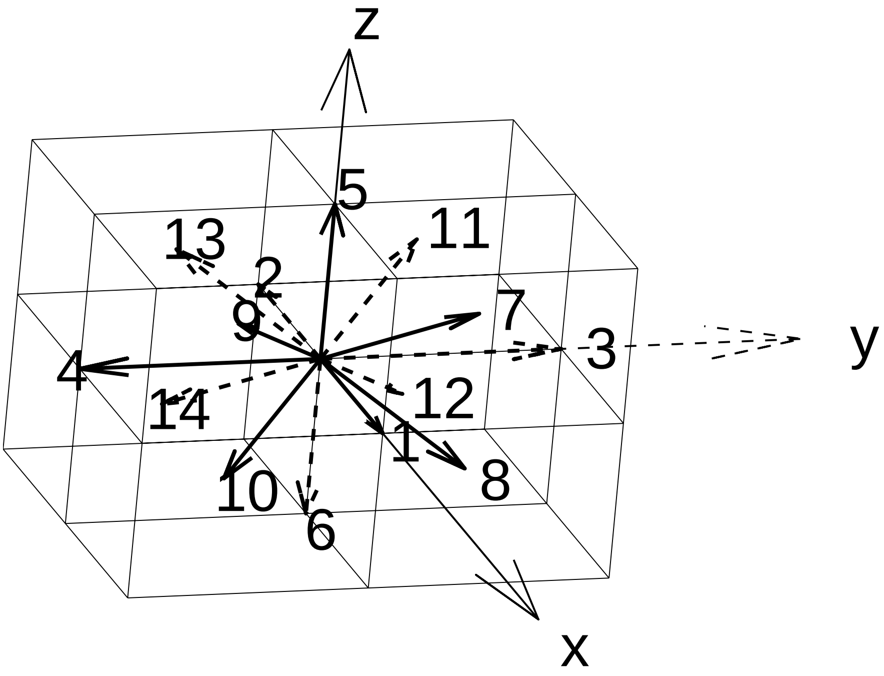


Fig. 3.3.6. *The lattice velocities of the D3Q15 lattice.*



The lattice tensor of rank 2

$$L_{\alpha\beta}^{(D3Q15)} = 16\,\delta_{\alpha\beta}$$

is isotropic whereas the lattice tensor of rank 4

$$L_{\alpha\beta\gamma\delta}^{(D3Q15)} = 8\,(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) + 16\,\delta_{\alpha\beta\gamma\delta}$$

is non-isotropic. The weights $w_i = 2$ for speed $\sqrt{3}$ and $w_i = 1$ for speed 2 lead to isotropic generalized lattice tensors of rank 2

$$G_{\alpha\beta}^{(D3Q15)} = 24\,\delta_{\alpha\beta} \tag{3.3.22}$$

and rank 4

$$G_{\alpha\beta\gamma\delta}^{(D3Q15)} = 16\,\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right). \tag{3.3.23}$$

D3Q19. The model D3Q19 has the following lattice velocities

$$\begin{aligned} \boldsymbol{c}_0 &= (0,0) \\ \boldsymbol{c}_{1,2},\, \boldsymbol{c}_{3,4},\, \boldsymbol{c}_{5,6} &= (\pm 1,0,0),\, (0,\pm 1,0)\, (0,0,\pm 1) \\ \boldsymbol{c}_{7,\ldots,10},\, \boldsymbol{c}_{11,\ldots,14},\, \boldsymbol{c}_{15,\ldots,18} &= (\pm 1,\pm 1,0),\, (\pm 1,0,\pm 1),\, (0,\pm 1,\pm 1). \end{aligned} \tag{3.3.24}$$

The lattice tensor of rank 2

$$L_{\alpha\beta}^{(D3Q19)} = 10\,\delta_{\alpha\beta}$$

is isotropic whereas the lattice tensor of rank 4

$$L_{\alpha\beta\gamma\delta}^{(D3Q19)} = 4\,(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) - 2\,\delta_{\alpha\beta\gamma\delta}$$

is non-isotropic. The weights $w_i = 2$ for speed 1 and $w_i = 1$ for speed $\sqrt{2}$ lead to isotropic generalized lattice tensors of rank 2

$$G_{\alpha\beta}^{(D3Q19)} = 12\,\delta_{\alpha\beta} \tag{3.3.25}$$

and rank 4

$$G_{\alpha\beta\gamma\delta}^{(D3Q19)} = 4\,\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right). \tag{3.3.26}$$

3.3.4 Thermal LBMs: D2Q13-FHP (multi-speed FHP model)

For thermal lattice Boltzmann models (Navier-Stokes plus energy equation) isotropic lattice tensors up to rank 6 are required.

$$\begin{aligned} \boldsymbol{c}_i &= (0,0) & & i = 0 \\ \boldsymbol{c}_i &= \left(\cos\frac{2\pi k}{6}, \sin\frac{2\pi k}{6}\right) & & i = 1,2,...,6; \quad k = i \\ \boldsymbol{c}_i &= 2\left(\cos\frac{2\pi k}{6}, \sin\frac{2\pi k}{6}\right) & & i = 7,8,...,12; \quad k = i - 6 \end{aligned} \tag{3.3.27}$$

Fig. 3.3.7. *The lattice velocities of the D3Q19 lattice.*

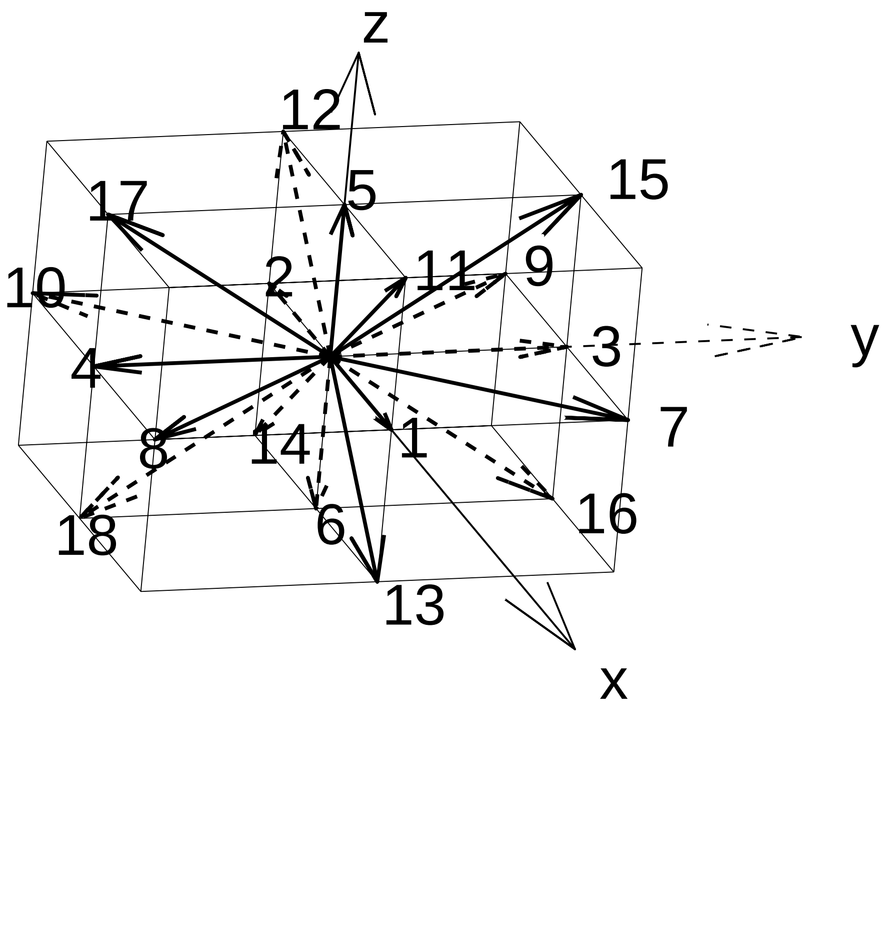
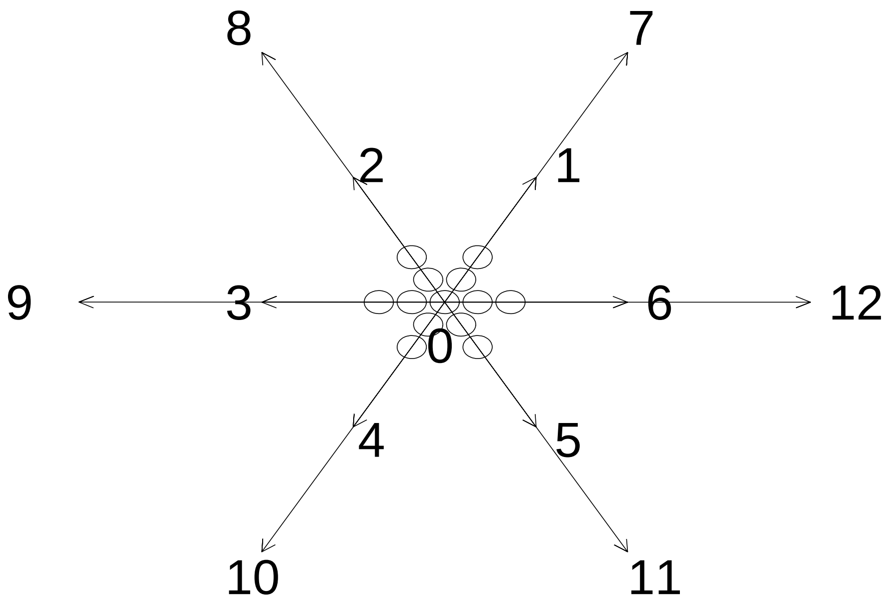


Fig. 3.3.8. *The lattice velocities of the D2Q13-FHP (multi-speed FHP) lattice.*



The lattice tensors of rank 2

$$L_{\alpha\beta}^{(D2Q13-FHP)} = 51\delta_{\alpha\beta} \qquad (3.3.28)$$

and of rank 4

$$L_{\alpha\beta\gamma\delta}^{(D2Q13-FHP)} = 15\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right) \qquad (3.3.29)$$

are isotropic.

3.3.5 Exercises

Exercise 3.3.2. (*)
Prove Theorem 3.3.1 for the special case of tensors of rank 2 in two dimensions. Do reflections play any role?

Exercise 3.3.3. (*)
Show that isotropic tensors of the same rank and dimension form a linear space.

Exercise 3.3.4. (*)
Prove that there are no isotropic tensors of rank 1.

Exercise 3.3.5. (**)
Prove (in 2D) that $T_{\alpha\beta\gamma\delta} = \delta_{\alpha\beta\gamma\delta}$ is not isotropic. Note: $\delta_{\alpha\beta\gamma\delta}$ is 1 when all indices are equal and 0 otherwise. Thus it is the generalization of the Kronecker symbol $\delta_{\alpha\beta}$.

Exercise 3.3.6. (**)
Prove (in 2D) that $T_{\alpha\beta\gamma\delta} = \delta_{\alpha\beta}\delta_{\gamma\delta}$ is isotropic.

3.4 Desperately seeking a lattice for simulations in three dimensions

Frisch, Hasslacher and Pomeau (1986) found out that in addition to mass and momentum conservation lattice-gas cellular automata for the Navier-Stokes equation must reside on a grid with 'sufficient' symmetry. The importance of this insight can hardly be overestimated. Thus the first task in the development of a LGCA for simulations in 3D is to find a lattice with appropriate symmetry. This is not as easy as in 2D.

3.4.1 Three dimensions

In close analogue to 2D where lattice vectors $\boldsymbol{c}_i$ were defined by the corners of regular polygons let us define $\boldsymbol{c}_i$ in 3D by the corners of regular polytopes. In three dimensions only a few regular polytopes, namely the *Platonic solids*[30], lead to lattice vectors of equal length. There are exactly five Platonic solids: tetrahedron, hexahedron (cube), octahedron, dodecahedron and icosahedron (compare Table 3.4.1 and Fig. 3.4.1).

The cube and the octahedron are *dual* to each other in the following sense: the mid-points of the faces of a cube yield the corners of an octahedron and the mid-points of an octahedron yield the corners of a cube. In the same sense dodecahedron and icosahedron are dual to each other. The tetrahedron is self-dual: its dual solid is also a tetrahedron.

Rotations and reflections which transform the solid onto itself form a group which is referred to as the symmetry group of the solid. Each rotation or reflection which transforms the solid onto itself does the same thing with the dual solid embedded. Therefore dual regular polytopes show identical symmetries and their corresponding symmetry groups are isomorphic (Ledermann, 1985a). There may be further polytopes with the same symmetry group which, however, are not regular.

In order to be useful as building blocks of a lattice-gas cellular automata, a polyhedron must show a large enough symmetry group (this constraint is fulfilled only by the dodecahedron and its dual partner the icosahedron) and in addition must fill the whole space[31]. The cube is the only Platonic solid whichs completely fills the space without gaps (Ledermann, 1985a). Thus in 3D there is no polyhedron which respects all constraints.

[30] Definition of Platonic solids: convex polytopes, bounded by regular congruent polyhedrons and with equal number of edges meeting at each corner.

[31] Actually it is required that all corners (which are nodes of the lattices) are connected to an equal number of nodes by the lattice vectors $\boldsymbol{c}_i$. Gaps between the polytopes could lead to nodes with a smaller number of nearest neighbors. As an analog in 2D consider the parqueting of the plane with octagons.

Fig. 3.4.1. *The five Platonic solids.*

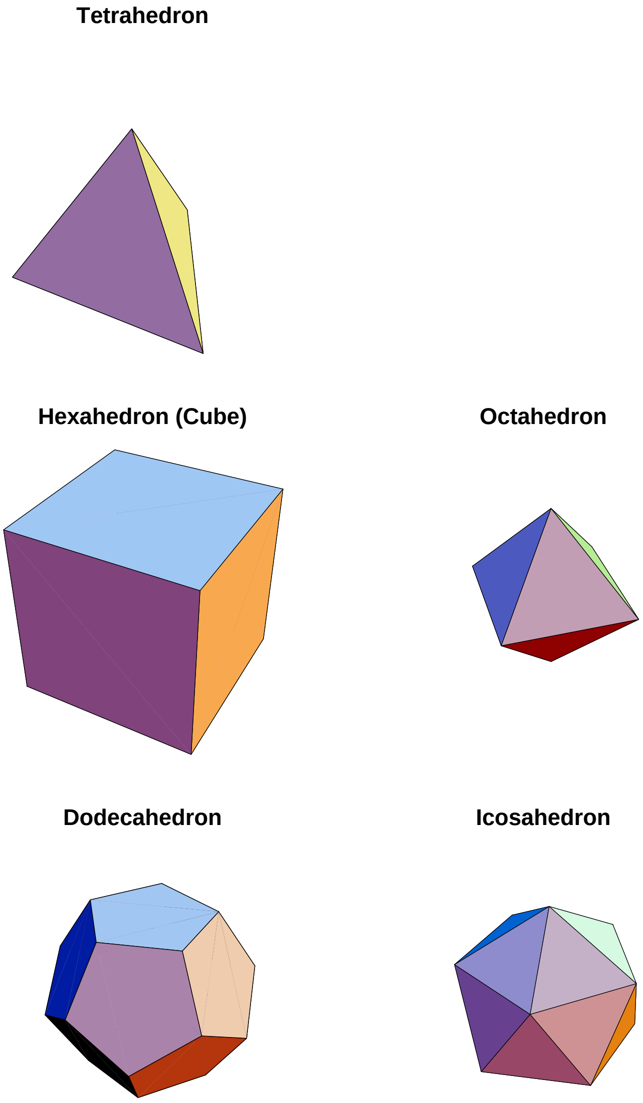


Table 3.4.1. *The five Platonic solids: number of faces F, edges E and corners C. Euler's polyhedron theorem: $F - E + C = 2$*

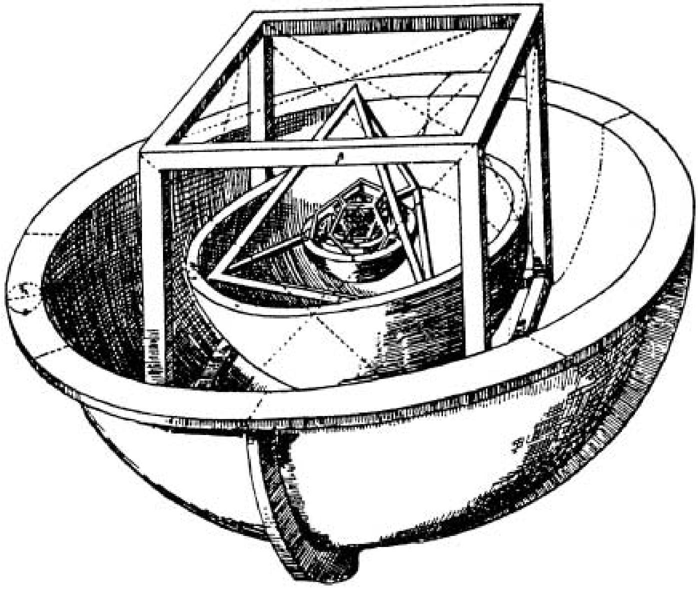
polyhedron	F	E	C	dual solid
tetrahedron	4	6	4	tetrahedron
hexahedron (cube)	6	12	8	octrahedron
octrahedron	8	12	6	hexahedron (cube)
dodecahedron	12	30	20	icosahedron
icosahedron	20	30	12	dodecahedron

Table 3.4.2. *Lattice tensors of ranks* 2 *to* 6 *for the five Platonic solids: isotropic (+) or not (−) (from Wolfram, 1986). The number of lattice vectors is equal to the number of corners C.*

polyhedron	C	2	3	4	5	6
tetrahedron	4	+	−	−	−	−
hexahedron (cube)	8	+	+	−	+	−
octrahedron	6	+	+	−	+	−
dodecahedron	20	+	+	+	+	−
icosahedron	12	+	+	+	+	−

Historical remark: In ‘Mysterium Cosmographicum’ (1596) Johannes Kepler (1571-1630) has suggested that the distances of the then known planets - Mercury to Saturn - are constrained by the Platonic solids which are alternatingly inscribed and circumscribed to spheres. (compare Fig. 3.4.2).

Fig. 3.4.2. *In ‘Mysterium Cosmographicum’ (1596) Johannes Kepler (1571-1630) has suggested that the distances of the then known planets - Mercury to Saturn - are constrained by the Platonic solids which are alternatingly inscribed and circumscribed to spheres. The six spheres correspond to the six planets, Saturn, Jupiter, Mars, Earth, Venus, Mercurius, separated in the order by cube, tetrahedron, dodecahedron, octahedron and icosahedron (adapted from Weyl, 1989).*



3.4.2 Five and higher dimensions

A further possibility is to find lattices in higher dimensions with sufficient symmetry. The quantities of interest can then be obtained by appropriate projections from higher dimensions down to 3D. In five and higher dimensions there are only three regular polytopes[32] for each dimension, namely

[32] These three regular polytopes exist also in lower dimensions.

the *simplex*[33], the *hypercube*[34] and its dual solid. The corresponding lattice tensors of rank 4 are isotropic only for $D < 3$ for the simplex and only for $D < 4$ for the hypercube and its dual solid.

3.4.3 Four dimensions

The last chance lies in four dimensions where in addition to the simplex, the hypercube and its dual solid there exist further three regular polytopes which can be characterized by the Schläfli symbols[35] $\{3,4,3\}$, $\{3,3,5\}$, and $\{5,3,3\}$. The $\{3,4,3\}$-polytop is referred to as *face-centered hypercube* (FCHC). It has 24 corners with coordinates which are permutations of $(\pm 1, \pm 1, 0, 0)$. The corresponding lattice tensors are isotropic up to 4th rank inclusively. Thus FCHC is the lattice searched for! The 24 lattice vectors of FCHC read

$$\begin{array}{ccc} (\pm 1, \pm 1, 0, 0), & (\pm 1, 0, \pm 1, 0), & (\pm 1, 0, 0, \pm 1), \\ \\ (0, \pm 1, \pm 1, 0), & (0, \pm 1, 0, \pm 1), & (0, 0, \pm 1, \pm 1). \end{array} \tag{3.4.1}$$

The projections of FCHC into 3D space are shown in Figures 3.4.3 and 3.4.4. There are 12 velocities with $c_{i4} = 0$ (Fig. 3.4.3) and two times 6 velocities with $c_{i4} = \pm 1$ (Fig. 3.4.4).

The other regular polytopes $\{3,3,5\}$ and $\{5,3,3\}$ are dual to each other. The $\{3,3,5\}$ polytope has 120 corners and the corresponding lattice tensors are isotropic up to 8th rank inclusively. The $\{3,3,5\}$ and $\{5,3,3\}$ polytopes have not been mentioned by Wolfram (1986) as alternatives for applications in lattice-gas automata. One reason at least is simplicity: FCHC has much less corners than the $\{3,3,5\}$ polytop.
Further reading on regular polytopes: Coxeter (1963).

[33] The set of all vectors $\boldsymbol{x}$ which respect the constraints

$$\boldsymbol{x} = \lambda_1 \boldsymbol{x}_1 + ... + \lambda_r \boldsymbol{x}_r, \quad \boldsymbol{x}_i \in \mathbf{R}^D, \quad \lambda_i \in \mathbf{R}, \quad \lambda_i \geq 0, \quad \sum_{i=1}^{r} \lambda_i = 1,$$

are referred to as simplex $[\boldsymbol{x}_1, ..., \boldsymbol{x}_r]$. Despite of some degenerated cases 2, 3, and 4 points lead to a line segment, a (planar) triangle and a tetrahedron as simplices (Ledermann, 1985a, p. 108).

[34] The hypercube γ_D in D dimensions has 2^D corners $\pm \boldsymbol{e}_1 \pm ... \pm \boldsymbol{e}_D$. Its $2 \cdot D$ 'surfaces' are $(D-1)$-dimensional hypercubes. γ_1 is a line segment, γ_2 a square, and γ_3 a cube (Ledermann, 1985b). $\boldsymbol{e}_i = (0, ..., 0, 1, 0, ..., 0)$ are the standard base vectors.

[35] Compare Appendix 6.5.

Fig. 3.4.3. *The projection (along the 4th axis) of the lattice velocities of the FCHC lattice for $c_{i4} = 0$.*

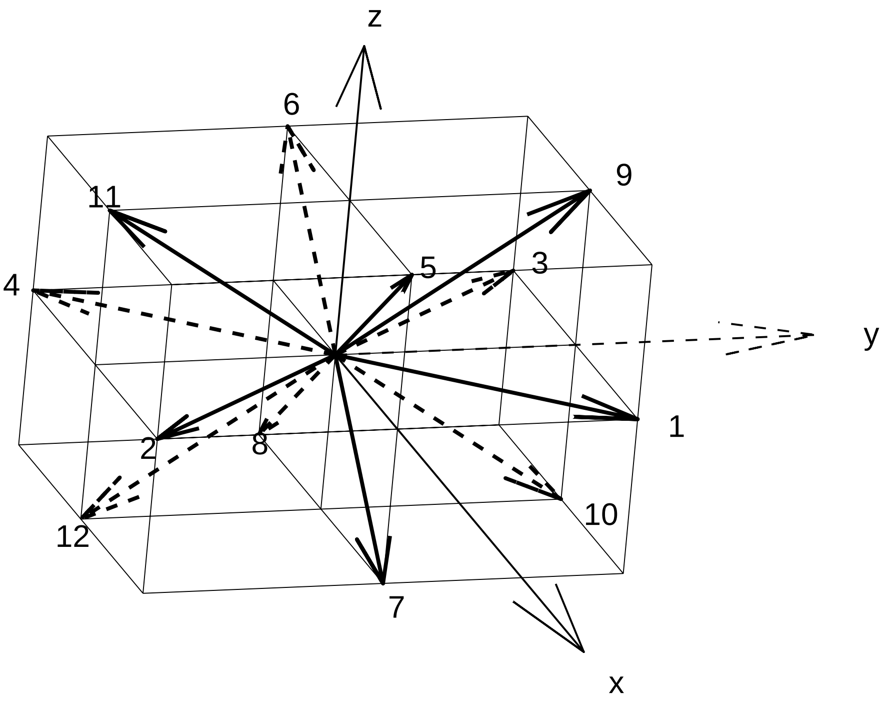
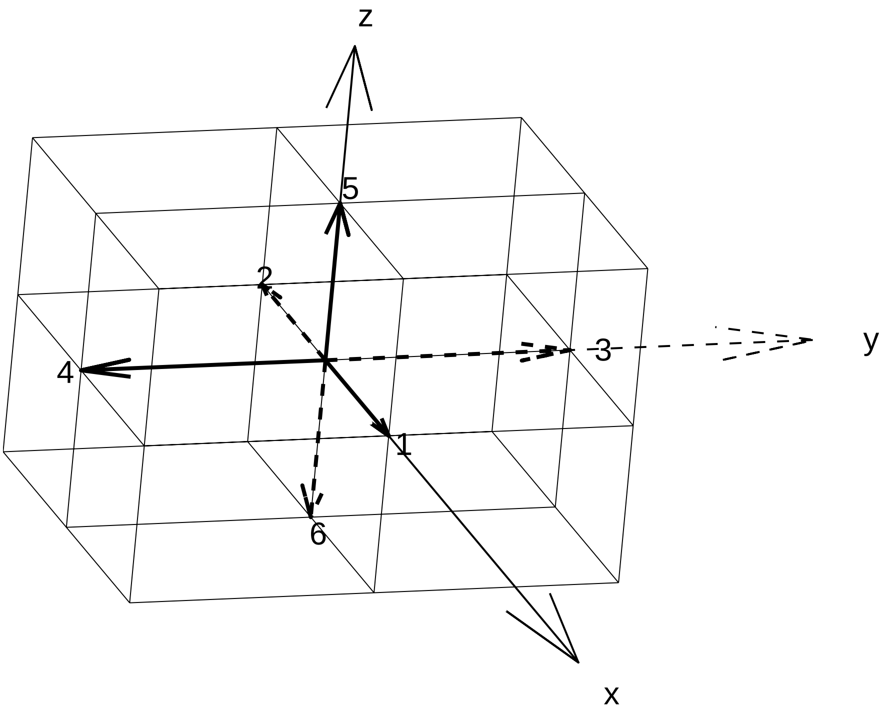


Fig. 3.4.4. *The projection (along the 4th axis) of the lattice velocities of the FCHC lattice for* $c_{i4} = \pm 1$.



Exercise 3.4.1. (***)
Is it possible to build a space-filling lattice from the regular polytop $\{3,3,5\}$ alone?

Exercise 3.4.2. (***)
Prove the polyeder theorem of Leonard Euler (1707-1783): The number C of corners plus the number F of faces equals the number E of edges plus 2:

$$C + F = E + 2.$$

Exercise 3.4.3. (**)
Prove that the Platonic solids are the only regular polyhedrons in 3D. Hint: Apply the polyeder theorem of Euler.

3.5 FCHC

D'Humières, Lallemand and Frisch (1986) proposed the face-centered hypercube (FCHC) as a lattice with sufficient symmetry for hydrodynamic simulations in 3D but without specifying collision rules. Those were given for the first time by Hénon (1987a) and later on modified by Rem and Somers (1989), Somers and Rem (1989) and others.

The 24 vectors of the FCHC lattice are listed in Section 3.3. The collision rules have to respect the following constraints:

1. The number of particles is conserved in each collision (conservation of mass).
2. The momentum is conserved in each collision.
3. There are no conserved quantities except of mass and momentum (no spurious invariants).
4. The exclusion principle is valid: at each node there sits at most one particle per lattice velocity (or per cell).
5. The collision rules share the symmetry of the lattice or, in other words, they are invariant under arbitrary transformations by elements of the isometric (symmetry) group $\mathcal{G}$.
6. The collisions respect the semi-detailed balance.

The FHP-model with six lattice velocities (FHP-I) has only $2^6 = 64$ different states per node. Therefore the collision rules could be derived 'by hand'. On the contrary, for FCHC there are $2^{24} = 16\ 777\ 216$ different states. Thus the collision rules have to be specified by an automatic algorithm.

3.5.1 Isometric collision rules for FCHC by Hénon

Despite the constraints given above there is space for almost unnumerable many different collisions. Therefore Hénon (1989a) introduced further constraints (referred to as *Hénon constraints*):

1. Every collision is an isometry[36] (isometric collision rules). Motivation:
   - An isometry is simpler than an arbitrary transformation.
   - The general constraints 1 and 4 are fulfilled automatically.

[36] Isometries are mappings g which keep the distances $d(\alpha, \beta)$ of arbitrary points α, β invariant: $d(g(\alpha), g(\beta)) = d(\alpha, \beta)$. Rotations and reflections are isometries.

- All collisions of HPP and FHP (without rest particles) are isometries.

2. The isometry depends only on the total momentum.
3. The isometry which is to be applied will be chosen by random out of the optimal (with respect to low viscosity, see below) isometries.

Despite the first two Hénon constraints there exists in addition to the identity at least one nontrivial isometry for each total momentum.
The various isometries contribute differently to the shear viscosity which should be kept as low as possible. Hénon could quantify these contributions and thereby classify the optimal isometries. This isometric group $\mathcal{G}$ contains 1152 elements. It can be created by five generators.
In order to keep the collisions easy to take at a glance, Hénon introduced a kind of normal form of the components of momentum q_1, q_2, q_3, q_4. Note that this is not a further constraint to the collision rules. For every state there exists an isometry which transforms that state into a state (*normalized momenta*) with

$$q_1 \geq q_2 \geq q_3 \geq q_4 \quad \text{and} \quad (q_4 = 0 \quad \text{or} \quad q_1 + q_4 < q_2 + q_3). \tag{3.5.1}$$

Thereby the problem of constructing collision rules for FCHC is simplified considerably because there exist only 37 different normalized momenta.
Each collision proceeds by successive application of three isometries:

1. Transformation of the initial state by an isometry $g \in \mathcal{G}$ into the state with the appropriate normalized momentum.
2. Application of an optimal isometry M (proper collision).
3. Transformation with $g^{-1} \in \mathcal{G}$ back to the original coordinates.

Thus the total transformation reads $g^{-1} \circ M \circ g$.

3.5.2 FCHC, computers and modified collision rules

Despite the extreme simplification of the original problem of constructing collision rules for FCHC by Hénon the coding of these rules require the introduction of a very large *look-up table* that contains the final state (after collision) for each initial state (before collision). The necessary memory can be estimated as follows. The coding of a single state requires 24 bits. On a CRAY two states can be packed into one word (8 bytes = 64 bits). Thus for the storage of $2^{24} = 16\ 777\ 216$ initial and the same number of final states 8 times 16 777 216 bytes or approximately 130 Mbytes are required. This is a severe obstacle for computers with small memories.
Relatively few applications of the FCHC model have been published. A paper of Chen et al. (1991c) on the flow through porous media is especially remarkable. The calculations were performed on a CRAY-YMP (core size of a few

hundred Mbytes). The size of the look-up table which actually should contain 16 million entries could be somewhat reduced by exploiting the hole-particle symmetry. The propagation has been coded in CRAY-Assembler. Despite all these machine-specific measures the update rates per node are not higher than those of the PI model (compare Section 3.6) in 3D which is coded in standard C (Wolf-Gladrow and Vogeler, 1992). For a fair comparison of the two models, however, the different values of the shear viscosities have to be taken into account.

The collision rules were simplified quite drastically by Rem and Somers (1989) even risking the violation of the semi-detailed balance. The resulting look-up table requires only 40 kbytes. Computer experiments show good agreement with theoretical predictions such as Fermi-Dirac distribution and the shear viscosity as a function of density. Yet, the shear viscosity is three time higher than its optimal value.

In addition to his isometric collision rule Hénon has proposed a purely random rule: The final state will be randomly chosen out of all states with the same mass and momentum as the initial state. The values of the shear viscosity for both rules proposed by Hénon are comparable (Hénon, 1987a).

3.5.3 Isometric rules for HPP and FHP

Hénon applied his method for the construction of isometric collision rules also to the HPP and the FHP model (without restparticles). For HPP one obtains collision rules which are identical to those of the original formulation: the head-on collision is the only collision. The model is too simple to allow additional collisions. On the other hand the Hénon constraints are not too restrictive to forbit all nontrivial transformations.

For the FHP model the isometric collision rules read

$$\begin{aligned}
(i, i+3) &\rightarrow (i+1, i+4) \quad \text{or} \quad (i-1, i+2), \\
(i, i+2, i+4) &\rightarrow (i+1, i+3, i+5) \quad \text{or} \quad (i, i+2, i+4), \\
(i-1, i, i+2) &\rightarrow (i-2, i, i+1), \\
(i+1, i+2, i+4, i+5) &\rightarrow (i, i+2, i+3, i+5) \quad \text{or} \\
&\qquad (i, i+1, i+3, i+4),
\end{aligned}$$

i.e. the two-particle, the three-particle, the two-particle with observer and the four-particle collisions. The only difference in relation to the original formulation (compare Section 3.2) occurs for the three-particle collision: according to the isometric rules the particle velocities will be changed with a probability of 1/2 instead of 1 as in the original rules. This can be interpreted as an indication that the three-particle collision does not contribute much to the reduction of the shear viscosity. This collision was introduced to destroy a spurious invariant.

3.5.4 What else?

- Implementation of the FCHC model: in the 4th dimension the model encompasses only two ‘layers’ and periodic boundary condition.
- The fourth component of the momentum behaves like a passive scalar (Frisch et al., 1987).
- Variants of the FCHC model are listed in Table 3.5.1.
- Further reading: Rivet et al. (1988), Shimomura et al. (1988), Hénon (1989), Cancelliere et al. (1990), Dubrulle et al. (1990), Ladd and Frenkel (1990), Vergassola et al. (1990), Rivet (1991), Benzi et al. (1992), Hénon (1992), van der Hoef et al. (1992), Verheggen (1992), van Coevorden et al. (1994), Adler et al. (1995).

Exercise 3.5.1. (**)
Classify the 64 different states at each node of the FHP-I model according to particle number and momentum.

Exercise 3.5.2. (**)
How many different 3-momenta can be realized in the FCHC model?

Concluding remark: If one restricts oneself to models with a single lattice speed there exists only the FCHC model for hydrodynamic simulations in 3D. The collision rules of the FCHC model are much more complicated than those of the FHP model. Later on we will discuss multi-speed models as an alternative to FCHC.

Table 3.5.1. *Variants of the FCHC model (compare Dubrulle et al., 1990). Roughly, the Reynolds coefficient R_*^{max} measures the inverse viscosity in lattice units.*

Name	Rest particles	Semi-detailed balance	$R_*^{\max}$ Boltzmann	$R_*^{\max}$ measured	References
FCHC-1	0	Yes	2.00	2.0	Hénon (1987), Rivet (1987)
FCHC-2	0	Yes	6.44	-	Hénon (1989)
FCHC-3	0	Yes	7.13	6.4	Hénon (1989), Rivet (1988a,b)
FCHC-4	0	Yes	7.57	-	Hénon (1989)
FCHC-5	3	Yes	10.71	-	Hénon (1989)
FCHC-6	0	No	17.2	-	Dubrulle (1988)
FCHC-7	3	No	(∞)	7.9	Dubrulle et al. (1990)
FCHC-8	3	No	99.7	13.5	Dubrulle et al. (1990)

3.6 The pair interaction (PI) lattice-gas cellular automata

The FCHC model for hydrodynamic simulations in 3D has complicated collision rules and thus requires large look-up tables. Extension to problems with a free surface or to magneto-hydrodynamics seems to be extremely involved. In 1989 Nasilowski proposed a lattice-gas cellular automata which runs in 2D over a square lattice as well as in 3D over a cubic grid. In contrast to FHP or FCHC the state of a cell of the PI model is characterized by $D + 1$ (D spatial dimension) bits instead of only one. The interaction[37] *at a node consists of a succession of interactions between pairs of cells (thus the name PI = pair interaction). The splitting into pair interactions allows an efficient coding with bit-operators (C) or bit-functions (FORTRAN) also in 3D.*

A complete discussion of the pair interaction lattice-gas cellular automata has been given by Nasilowski (1991). Here only the main ideas of the PI-approach will be explained and consequentely the presentation is restricted to the two-dimensional case. The extension to 3D is straightforward.

3.6.1 Lattice, cells, and interaction in 2D

The PI model in 2D is based on the square lattice (compare Fig. 3.6.1). As usual the development in time proceeds by an alternating sequence of local interaction (only cells of a single node are involved) and propagation to the nearest neighbor nodes. The lattice splits into two sub-lattices[38]:

– At even time levels the particles reside on nodes with even indices (white circles).

– At odd time levels the particles reside on nodes with odd indices (black circles).

As for HPP, FHP and FCHC there is a cell on each link at each node. The state of a cell of the PI model is characterized by $D+1$ bits n_J ($J = 0, 1, ..., D$) where D is the dimension: n_0 is called the *mass bit* and n_j ($j = 1, ..., D$) are the *momentum bits*. By convention in this section uppercase indices run from 0 to D whereas lowercase indices run from 1 to D. The momentum bits are subject to the constraint

[37] We speak of interaction instead of collision because some of the rules of the PI lattice-gas cellular automata cannot be described as collisions between particles.

[38] Points with a combination of an even and an odd index will never be occupied by particles. Therefore they will not be called nodes and they are not shown in Fig. 3.6.1.

$$n_j \leq n_0 \tag{3.6.1}$$

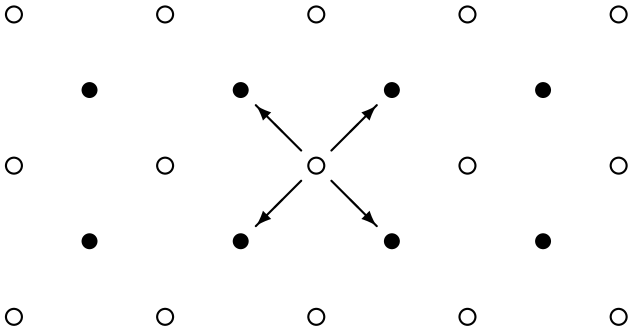
which can be interpreted as 'the momentum of empty cells vanishes'. The vector linking two neighboring nodes is termed $\boldsymbol{c}$ (lattice velocity). The components of $\boldsymbol{c}$ take on the values 1 or -1. The links to the neighboring nodes and the corresponding cells of a node will be labelled a, b, c, d (compare Fig. 3.6.2). The corresponding lattice velocities are given by $\boldsymbol{c}_a = (1, 1)$, $\boldsymbol{c}_b = (-1, 1)$, $\boldsymbol{c}_c = (-1, -1)$, $\boldsymbol{c}_d = (1, -1)$. The *momentum* $\boldsymbol{m}$ is defined *component-wise*:

$$m_{j'} := n_{j'} v_{j'}, \quad j' = 1, ..., D \tag{3.6.2}$$

(remark: no summation convention here!). This definition is rather unusual because in general the momentum does not point to the same direction as the velocity. This can be illustrated by considering all possible states of cell a:

1. Mass bit $n_0 = 0 \rightarrow$ all momentum bits vanish (according to eq. 3.6.2): the cell is empty.
2. Mass bit $n_0 = 1$, all momentum bits vanish: rest particle.
3. $n_0 = n_1 = 1$, $n_2 = 0$: particle with x-momentum only.
4. $n_0 = n_2 = 1$, $n_1 = 0$: particle with y-momentum only.
5. $n_0 = n_1 = n_2 = 1$: particle with momentum in the diagonal direction; this is the only case where $\boldsymbol{m}$ and $\boldsymbol{c}$ point to the same direction.

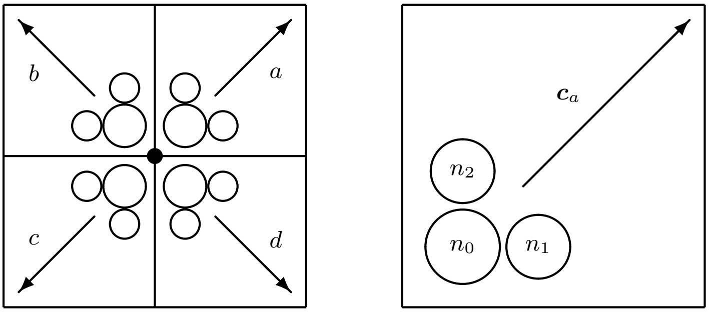
Fig. 3.6.1. *The sub-lattices of the PI lattice-gas cellular automata in 2D*



What was Nasilowski's motivation to introduce the somewhat strange definition of the momentum? There are at least two good reasons:

1. The component-wise definition of the momentum allows a splitting of the interaction into pair interactions (see below).

Fig. 3.6.2. *Structure of the nodes and cells of the PI lattice-gas cellular automata.*

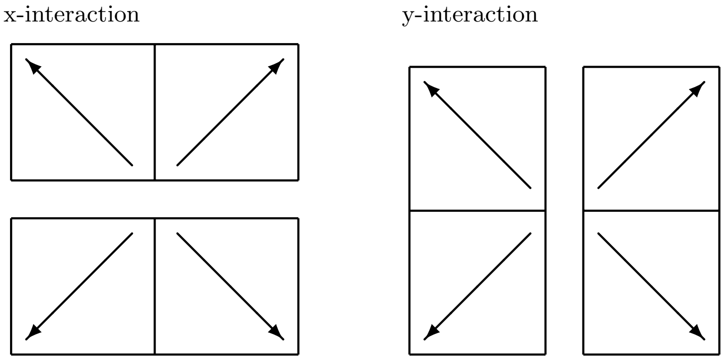


2. The symmetry of the square lattice is not sufficient to assure the isotropy of 4th rank lattice tensors (compare HPP and Section 3.3 on lattice tensors). The component-wise definition of the momentum introduces a *new degree of freedom* in that 'the momentum can fluctuate with respect to the direction of velocity'. This freedom may open a new route to isotropy.

The interaction is composed of the following sequence of pair interactions (compare Fig. 3.6.3):

1. Interaction in x-direction between the cells a and b and between the cells c and d
2. followed by interaction in y-direction between the cells a and d and between the cells b and c.

Fig. 3.6.3. *PI: interaction between pairs of cell first in x-, then in y-direction.*



The pair interaction rules are designed according to the maxim 'whatever is not forbidden is allowed'. Nasilowski (1991) formulates three constraints:

1. The interaction must conserve mass and momentum (as all good lattice-gas cellular automata should do).
2. The interaction must be reversible, i.e. the mapping between initial and final state is *one-to-one*. This allows the calculation of the statistical equilibrium by applying the Gibbs formalism.
3. The interaction should yield a maximal change of the state of a node. The identity fulfills the first two constraints but leads to spurious (additional) invariants as, for example, the particle number on each diagonal of the lattice. Such invariants will produce deviations from the hydrodynamic behavior in the macroscopic limit and therefore should be avoided.

The rules given by Nasilowski (1991, p. 107) obey the first and second constraint and encompass all allowed pair interactions (compare also Fig. 3.6.4) which most probably lead to the maximal change.

3.6.2 Macroscopic equations

The rather lengthy calculations of the equilibrium distribution and the multi-scale expansion has been given by Nasilowski (1991). The first order terms of a multi-scale expansion and the rescaling of certain quantities leads for $\rho_0 = 1/2$ to the continuity equation

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.6.3}$$

and the Euler equation

$$(\partial_t + \boldsymbol{u}\nabla)\boldsymbol{u} + \nabla P = 0 \tag{3.6.4}$$

where

$$P = \frac{p}{\rho_0} = \frac{4}{9}\rho \tag{3.6.5}$$

is the kinematic pressure. The hydrodynamic velocity $\boldsymbol{u}$ is related to the momentum density $\boldsymbol{q}$ by

$$\boldsymbol{u} = \frac{8}{9}\boldsymbol{q} \tag{3.6.6}$$

where the hyper-momentum density $\boldsymbol{q}$ is defined component-wise by

$$q_J = 2^{-D} \sum_{\boldsymbol{v}} v_J \langle n_{\boldsymbol{v}\,J} \rangle \tag{3.6.7}$$

with the hyper-velocity $\boldsymbol{v} = (v_0, v_1, ..., v_D) = (1, \boldsymbol{c})$. In particular, $\boldsymbol{q} := (q_1, ..., q_D)$ is the momentum density, and $\rho := q_0$ is the mass density. The viscosity resulting from the second order terms of the multiscale expansion is anisotropic: a tensor of 4th order instead of a scalar.

Fig. 3.6.4. *Pair interaction in horizontal (x-) direction: all possible configurations and changes. Open cycles denote empty cells, filled cycles denote occupied cells, and arrows indicate the momentum.*

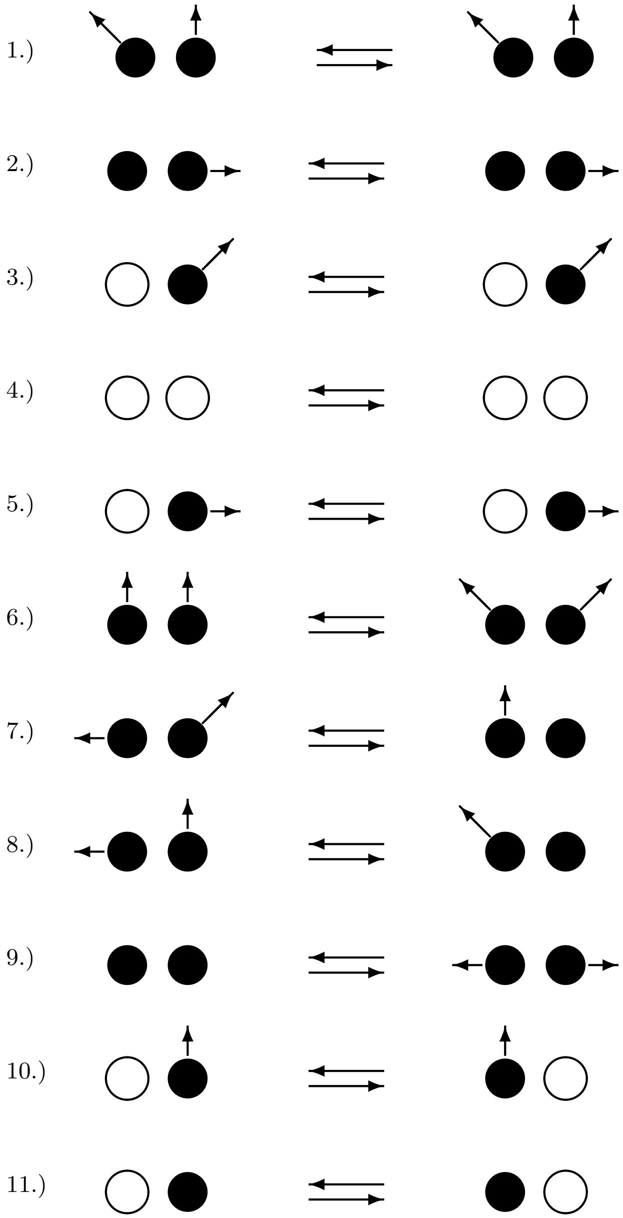
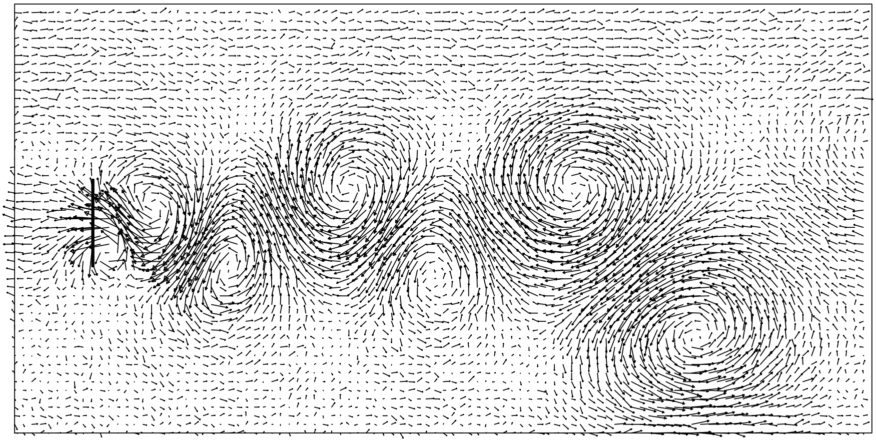


Fig. 3.6.5. *Simulation with PI-LGA of a Karman vortex street in 2D at a Reynolds number of* 80 *(Wolf-Gladrow et al., 1991): flow past a plate with upstream on the left. The figure shows the perturbation of the velocity after* 80,000 *time steps. The homogeneous flow field was subtracted to make the eddies clearly visible. The lattice consists of* 6400 *times* 3200 *nodes.*



3.6.3 Comparison of PI with FHP and FCHC

The PI-LGCA has some disadvantages compared with FHP and FCHC. PI can only be applied at the particular mass density $\rho_0 = 1/2$ and the viscosity is non-isotropic. On the other hand, the advantages are impressive. The pressure does not depend explicitly on the velocity and simulations in 3D are possible with portable code (Wolf-Gladrow and Vogeler, 1992). As an example Vogeler and Wolf-Gladrow (1993) have calculated drag coefficients for flows past obstacles in two and three dimensions.

Exercise 3.6.1. (*)
PI in 2D and 3D: How many different states are possible at a single node? How large is the number of states on a lattice with only four nodes?

Exercise 3.6.2. (**)
Does the result of the interaction depend on the order of the pair interactions?

Exercise 3.6.3. (**)
Find nontrivial rules for a pair interaction between the cell a and c and between b and d ('*diagonal pair interaction*').

3.6.4 The collision operator and propagation in C and FORTRAN

C:

```
        IXM = 32;      IXM1 = IXM - 1;
        IYM = 1024;    IYM1 = IYM - 1;

        LAST = LENGTH - 1;    /* the last bit */




   /*  -----   interaction on 1. sub-lattice   -----  */
   /*  -----   interaction: x-direction        -----  */
   /*  -----      pair a <--> b                -----  */

   for(ix=0; ix < IXM; ix++)
   for(iy=1; iy < IYM1; iy++) {

     ab0 = a10[ix][iy] & b10[ix][iy];
     ba0 = a10[ix][iy] ^ b10[ix][iy];
     nab1 = ~(a11[ix][iy] | b11[ix][iy]);
```

```
    chab0 = ba0 & nab1;

    chab1 = ab0 & ~(a11[ix][iy] ^ b11[ix][iy]);

    chab2 = ((ba0 & nab1) | ab0) & (a12[ix][iy] ^ b12[ix][iy]);

  /*  array_1 --> array_2  */

      a20[ix][iy] = a10[ix][iy] ^ chab0;
      b20[ix][iy] = b10[ix][iy] ^ chab0;

      a21[ix][iy] = a11[ix][iy] ^ chab1;
      b21[ix][iy] = b11[ix][iy] ^ chab1;

      a22[ix][iy] = a12[ix][iy] ^ chab2;
      b22[ix][iy] = b12[ix][iy] ^ chab2; }

  /*  ... interactions c <-> d, a <-> d, b <-> c ...      */

  /*        propagation:         */

  /* ---   a-direction   --- */

  for(ix=0; ix < IXM; ix++)
  for(iy=0; iy < IYM; iy++) {
     a10[ix][iy] = a20[ix][iy];
     a11[ix][iy] = a21[ix][iy];
     a12[ix][iy] = a22[ix][iy]; }

  /* ---   b-direction   --- */

  for(ix=0; ix < IXM1; ix++)
  for(iy=0; iy < IYM; iy++) {
   b10[ix][iy] = (b20[ix][iy]>>1) + (b20[ix+1][iy]<<LAST);
   b11[ix][iy] = (b21[ix][iy]>>1) + (b21[ix+1][iy]<<LAST);
   b12[ix][iy] = (b22[ix][iy]>>1) + (b22[ix+1][iy]<<LAST); }
```

FORTRAN:

```
      IXM = 32
      IXM1 = IXM - 1
      IYM = 1024
      IYM1 = IYM - 1

c     last bit
      LAST = LENGTH - 1

c  -----   interaction on 1. sub-lattice   -----
c  -----   interaction: x-direction    -----

c ----------------- pair (a,b)

      DO iy=1,IYM
      DO ix=1,IXM

      ab0 = iand(a10(ix,iy),b10(ix,iy))
      ba0 = ieor(a10(ix,iy),b10(ix,iy))
      nab1 = not(ior(a11(ix,iy),b11(ix,iy)))

c --- change ?

      chab0 = iand(ba0,iand(nab1,sb1(ix,iy)))
      chab1 = iand(ab0,not(iand(ieor(a11(ix,iy),
     1                 b11(ix,iy)),sb1(ix,iy))))
      chab2 = iand(ior(iand(ba0,nab1),ab0),
     1        iand(ieor(a12(ix,iy),b12(ix,iy)),sb1(ix,iy)))

c --- set new values

      a20(ix,iy) = ieor(a10(ix,iy),chab0)
      b20(ix,iy) = ieor(b10(ix,iy),chab0)
      a21(ix,iy) = ieor(a11(ix,iy),chab1)
      b21(ix,iy) = ieor(b11(ix,iy),chab1)
      a22(ix,iy) = ieor(a12(ix,iy),chab2)
      b22(ix,iy) = ieor(b12(ix,iy),chab2)

      ENDDO
      ENDDO

c        ... interactions c <-> d, a <-> d, b <-> c ...

c                      propagation:

c ---   a-direction   ---

      DO iy=1,IYM
      DO ix=1,IXM
         a10(ix,iy) = a20(ix,iy)
         a11(ix,iy) = a21(ix,iy)
         a12(ix,iy) = a22(ix,iy)
      ENDDO
      ENDDO

c ---   b-direction   ---

      DO iy=1,IYM
      DO ix=1,ixm1
         b10(ix,iy) = ishft(b20(ix,iy),-1)
     1              + ishft(b20(ix+1,iy),LAST)
         b11(ix,iy) = ishft(b21(ix,iy),-1)
     1              + ishft(b21(ix+1,iy),LAST)
         b12(ix,iy) = ishft(b22(ix,iy),-1)
     1              + ishft(b22(ix+1,iy),LAST)
      ENDDO
      ENDDO
```

3.7 Multi-speed and thermal lattice-gas cellular automata

The macroscopic equations derived from the microdynamics of lattice-gas cellular automata include 4th rank tensors which are composed of the lattice velocities and therefore are referred to as lattice tensors (compare Section 3.3 and especially Eq. (3.3.4)). The isotropy of these tensors, which depend on the symmetry of the underlying lattice, is an essential condition to obtain the Navier-Stokes equations. Hasslacher (1987) has shown that models with several different non-vanishing lattice speeds may be equivalent to models with a single speed but larger symmetry group. The resulting 4th rank tensors include speed dependent weights and are referred to as generalized lattice tensors. By appropriate choice of the weights these tensors can be isotropic for relative low symmetry of the lattice. In contrast to single speed models the pressure in multi-speed models does not depend explicitly on the flow velocity. The collision rules can be chosen such that in addition to mass and momentum also kinetic energy will be conserved. Such models are called thermal LGCA.

Models with several different speeds have been encountered before, namely the FHP variants with rest particles (FHP-II, FHP-III). Particles with vanishing speed have no influence on the isotropy of the lattice tensors. Therefore only models with different *non-vanishing* speeds are called *multi-speed models.*

3.7.1 The D3Q19 model

D'Humières, Lallemand, and Frisch proposed already in 1986 - together with FCHC - a multi-speed model in 3D with 19 lattice velocities (Eq. 3.3.24) and three speeds (0, 1, and $\sqrt{2}$). The collisions shall conserve mass ρ, momentum $\boldsymbol{j}$ and kinetic energy density $\rho\varepsilon_K$ which are defined as follows

$$\begin{aligned}
\rho &= \sum_I N_I \\
\boldsymbol{j} = \rho\boldsymbol{u} &= \sum_I N_I \boldsymbol{c}_I \\
\rho\varepsilon_K &= \sum_I N_I \frac{c_I^2}{2}
\end{aligned}$$

where the index $I = (\sigma, i)$ indicate the speed ($\sigma = 0, 1, 2$ for the rest particles and particles with speed 1 and $\sqrt{2}$, respectively) as well as the direction i.

The exclusion principle and semi-detailed balance lead to Fermi-Dirac distributions for the mean occupation numbers

$$N_I = f_{FD}(Q_I) = \frac{1}{1+\exp(Q_I)}$$

whereby

$$Q_I = \alpha + \beta \boldsymbol{c}_I \cdot \boldsymbol{u} + \gamma \boldsymbol{c}_I^2$$

is a linear superposition of the collision invariants. The Lagrange multipliers α, β, and γ can be calculated by expansion for small Mach numbers (compare analogous calculations for FHP).

The D3Q19 model is the first lattice-gas cellular automata with an energy term in the distribution function. This raises the question why similar terms do not occur in the other models (FHP, FCHC, PI).

- For FHP without rest particles mass and kinetic energy density are identical:
$$2\rho\varepsilon_K = \sum_i N_i \underbrace{\boldsymbol{c}_i^2}_{=1} = \sum_i N_i = \rho$$
- FHP with rest particles: the collisions including rest particles do not conserve kinetic energy.
- PI: the kinetic energy density can be defined by
$$\rho\varepsilon_K := \sum_i \frac{\boldsymbol{p}_i^2}{2m}$$
(particle mass $m = 1$); some of the interactions do not conserve energy.
- FCHC: same as for FHP, i.e. the kinetic energy density is proportional to the mass density for the version without rest particles and the collisions including rest particles do not conserve kinetic energy.

The advection term of the macroscopic momentum equation of the D3Q19 model contains the following 4th rank tensor

$$T_{\alpha\beta\gamma\delta} = \frac{\beta_0^2}{2} \sum_I \underbrace{d_\sigma(1-d_\sigma)(1-2d_\sigma)}_{=\,w_\sigma} c_{I\alpha} c_{I\beta} c_{I\gamma} c_{I\delta}$$

where d_σ are the equilibrium occupation numbers at vanishing flow speed (the zeroth order term of the Taylor expansion) which depend only on the lattice speed and not on direction. The new feature - compared to single speed models - is the occurrence of the weights

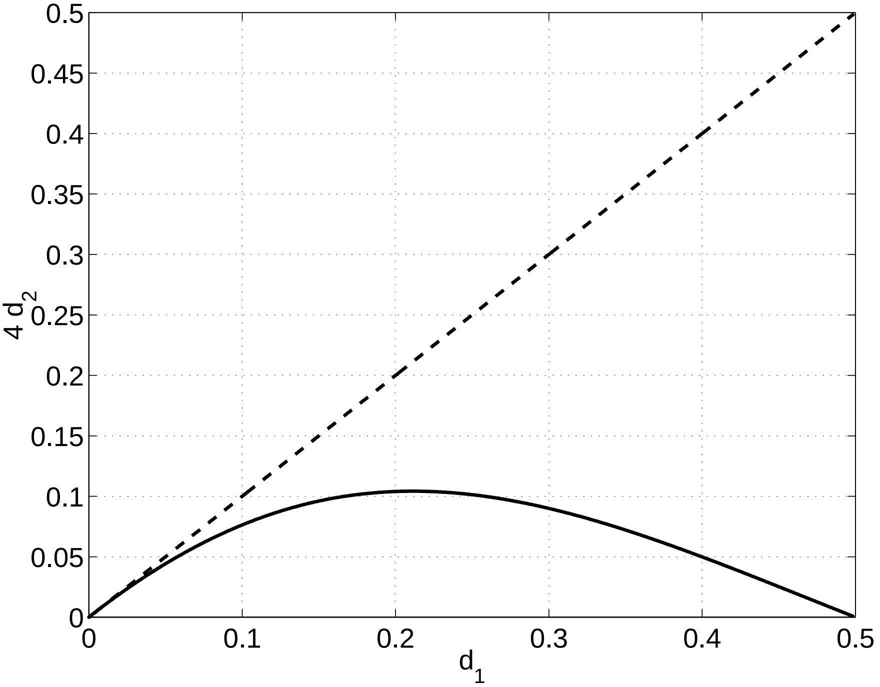
$$w_\sigma = d_\sigma(1-d_\sigma)(1-2d_\sigma)$$

which can influence the transformation properties (isotropy) of $T_{\alpha\beta\gamma\delta}$ (of course the coefficient $\beta_0^2/2$ does not play a role). The tensor $T_{\alpha\beta\gamma\delta}$ of the D3Q19 model is isotropic if

$$d_1(1-d_1)(1-2d_1) = 4d_2(1-d_2)(1-2d_2) \tag{3.7.1}$$

(d'Humières et al., 1986; compare also Section 3.3), i.e. the occupation numbers of speed-1 and speed-$\sqrt{2}$ particles must respect a certain ratio. In other words, the tensor is isotropic only at a certain 'temperature' where the right number of high energy states are excited. For small densities ($d_\sigma \ll 1$) relation (3.7.1) can be approximated by $d_1 = 4d_2$, i.e. the occupation numbers of cells with speed 1 must be four times as high as for cells with speed $\sqrt{2}$ (compare Fig. 3.7.1 for the ratio of d_2/d_1 for finite values of d_1). Thus a certain non-isotropy of the occupation numbers ensures the isotropy of the tensor $T_{\alpha\beta\gamma\delta}$.

Fig. 3.7.1. *The figure shows one solution of the cubic equation* $d_1(1-d_1)(1-2d_1) = 4d_2(1-d_2)(1-2d_2)$ *(solid line). For small values of* d_1 *it can be approximated by* $d_1 = 4d_2$ *(broken line).*



Exercise 3.7.1. (*)
Which interactions of the PI model violate energy conservation?

Exercise 3.7.2. (***)
Propose collision rules for D3Q19.

3.7.2 The D2Q9 model

Chen et al. (1989) reduced the multi-speed model of d'Humières et al. (1986) to 2D. They proposed collision rules and performed some numerical experiments. The model encompasses 9 lattice velocities:

$$\begin{array}{lcll} \boldsymbol{c}_0 & = & (0,0) & \text{rest particle} \\ \boldsymbol{c}_{1,2} & = & (\pm 1, 0) & \text{1-particles} \\ \boldsymbol{c}_{3,4} & = & (0, \pm 1) & \text{1-particles} \\ \boldsymbol{c}_{5,6,7,8} & = & (\pm 1, \pm 1) & \sqrt{2}\text{-particles.} \end{array}$$

All particles have the same mass. Only collisions between two particles are considered (compare Fig. 3.7.2):

1. Head-on collision of two particles with speed $c = 1$: as for HPP both particles are rotated (in the same sense) by 90°.
2. Head-on collision of two particles with speed $c = \sqrt{2}$: as for HPP both particles are rotated (in the same sense) by 90°.
3. Collision between a $\sqrt{2}$-particle and a rest particle: two particles with speed $c = 1$ leave the node under ±45° with respect to the incoming $\sqrt{2}$-particle. The inverse process is also allowed.
4. Collision between a $\sqrt{2}$-particle and a particle with speed $c = 1$ from different quadrants: the $\sqrt{2}$-particle will be rotated by 90° whereas the velocity of the 1-particle will be reversed (identical to rotation by 180°).

All these two-particle collisions conserve mass, momentum, and kinetic energy. The third type of collisions changes the number of particles with given speed. The tensor of 4th rank in the advection term reads

$$\begin{aligned} T_{\alpha\beta\gamma\delta} &= const. \cdot (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \\ &\quad + 2\left[d_1(1-d_1)(1-2d_1) - 4d_2(1-d_2)(1-2d_2)\right]\delta_{\alpha\beta\gamma\delta}. \end{aligned}$$

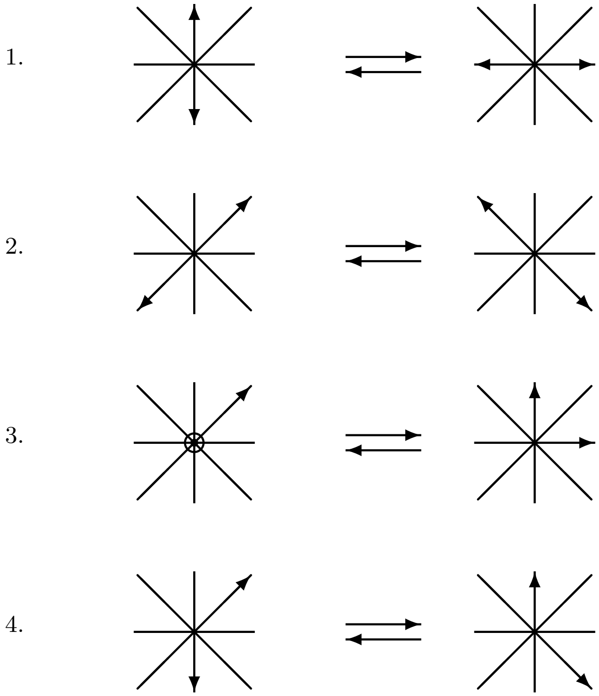
The second part of the term, which destroys isotropy, vanishes under the same condition (3.7.1) given by d'Humières et al. (1986). In the limit of small occupation numbers one obtains $d_1/d_4 = 4$ and a kinetic energy density $\varepsilon_K = 1/3$. In the same limit the pressure does not depend explicitly on the flow velocity. For finite mass densities, however, there occurs an $\boldsymbol{u}^2$ term as in FHP.

Furthermore, the model allows simulation of pure heat conduction problems ($\boldsymbol{u} = 0$). However, this is possible with simpler lattice-gas cellular automata (see, for example, Chopard and Droz, 1988 and 1991) or with ‘classical’ methods like finite differences. Biggs and Humby (1998) summarize their discussion of thermal LGA as follows:
“Thermal LGA has not been applied to problems of any real significance. The only simulation of note is that of Chen et al. (1991a) who briefly considered the Bénard convection problem. The relative paucity of non-isothermal LGA studies is perhaps not surprising given the relative immaturity of this level of LGA model. The models have, however, developed to a point where serious application can be contemplated provided substantial validation work is undertaken.”

Exercise 3.7.3. (**)
Show that the collision rules of Chen et al. (1989) contain all possible two-particle collisions.

Exercise 3.7.4. (**)
Find three-particle collisions which conserve mass, momentum, and energy.

Fig. 3.7.2. *Collision rules proposed by Chen et al. (1989) for the D2Q9 model. The rest-particle in rule three is indicated by a circle.*



3.7.3 The D2Q21 model

Fahner (1991) proposed a model with 21 lattice velocities in 2D (compare Section 3.3 and especially Eq. (3.3.18)). In addition to the 9 velocities of the D2Q9 model Fahner included the speeds 2 and $\sqrt{5}$. The number of possible states per node of $2^{21} = 2\,097\,152$ is almost comparable with that of FCHC. Fahner proposed the following collision rule which reminds on Hénon's random rule for FCHC:

– At each node mass, momentum, and kinetic energy will be calculated (these quantities characterize the 'macrostate' of each node).

– The final microstate of each node will be chosen among all microstates which are compatible with its macrostate.

To keep the number of collision states low and thus the look-up table small, Fahner took into account only collisions with a maximum of five particles involved. The number of these collisions is maximized by initializing a mean mass density with two particles per node.

The equilibrium occupation numbers obey a Fermi-Dirac distribution

$$N_i = \frac{1}{1 + \exp(-\beta\mu + \beta\epsilon_i - \boldsymbol{q}\boldsymbol{c}_i)} \tag{3.7.2}$$

where $\boldsymbol{c}_i$ are the lattice velocities, $\epsilon_i = c_i^2 \in \{0, 1, 2, 3, 4, 5\}$ are single-particle energies and $\boldsymbol{q}$, β, μ are Lagrange multipliers. The Lagrange multipliers can be calculated by expansion for small Mach numbers. Explicit expressions can be found in Fahner (1991).

3.7.4 Transsonic and supersonic flows: D2Q25, D2Q57, D2Q129

Kornreich and Scalo (1993) proposed multi-speed models similar to that of Fahner (1991) but with 25, 57, and 129 lattice velocities in 2D. The Lagrange multipliers were calculated numerically instead of applying the usual expansion for small Mach numbers. The resulting *numerical distribution functions* were used to determine the pressure tensor. Kornreich and Scalo found that the velocity dependent term in the expression for the pressure does not increase as fast as it can be predicted by applying the 'expanded' distributions. On the contrary, this term decreases again when the (macroscopic) flow velocity $\boldsymbol{u}$ is near one of the (microscopic) lattice velocities $\boldsymbol{c}_i$. Thus, in principle simulations of transsonic and supersonic flows are possible. These models are not without problems. The collision rules are complicated already for the D2Q25 model despite the fact that Kornreich and Scalo considered only two-particle collisions. The memory demand is enormous and the viscosity is larger than for FHP.

3.8 Zanetti ('staggered') invariants

In addition to local - at one node, at one time level - invariants there may exist some invariants which involve quantities at different nodes and different time levels. Zanetti (1989, p. 1539) gives a simple example:
"The presence of these new invariants can be easily understood by using a trivial one-dimensional example. Let $g(x)$ be the linear momentum of the particles present at site x, define $G_e(t) = \sum_{x,\text{even}} g(x,t)$, $G_o(t) = \sum_{x,\text{odd}} g(x,t)$ as the total momentum of the particles on even or odd sites, and let the collision rules conserve the momentum and the number of particles at each site. Since the particles can only hop between nearest neighbors, G_e and G_o are exchanged at each time step. The dynamics of this one-dimensional model allows three conserved quantities: M, $G_e + G_o$, and $H = (-1)^t(G_e - G_o)$. The first two are the usual number of particles and the total linear momentum; the third is due to our extremely simplified dynamics."

3.8.1 FHP

Zanetti (1989) has found three[39] staggered invariants of the FHP-III model:

$$H_i = (-1)^t \sum_{\boldsymbol{r} \in \Omega} (-1)^{\boldsymbol{b}_i \cdot \boldsymbol{r}} \boldsymbol{c}_i^{\perp} \cdot \boldsymbol{j}; \quad i = 1, 2, 3, \tag{3.8.1}$$

where $\boldsymbol{c}_i^{\perp}$ is obtained by rotating $\boldsymbol{c}_i$ by $\pi/2$ counterclockwise, $\boldsymbol{b}_i = (2/\sqrt{3})\boldsymbol{c}_i^{\perp}$ is the reciprocal space vector perpendicular to $\boldsymbol{c}_i$ and $\boldsymbol{j} = \sum_k \boldsymbol{c}_k n_k(\boldsymbol{x}, t)$ is the microscopic momentum density.
According to Zanetti the validity of (3.8.1) "can be verified by inspection". More precisely it may be shown (see Exercise 3.8.1) that the H_i are conserved by the combination of any single collision plus propagation and for collisionless propagation. Because the H_i are linear in the n_k this provides a complete proof of the invariance. The Zanetti invariants are also valid for FHP-I and FHP-II which follows as a direct consequence of the proof sketched above.

3.8.2 Significance of the Zanetti invariants

Additional invariants further restrict the dynamics by changing the equilibrium distributions which are functions of all invariants. Fortunately, the procedures usually applied for initialization create very small values of H_k (noise). Consequently, in the simulation reported so far there is no clear indication of the presence of the staggered-momentum density $h_i(\boldsymbol{r}, t) = (-1)^t(-1)^{\boldsymbol{b}_i \cdot \boldsymbol{r}} \boldsymbol{c}_i^{\perp} \cdot \boldsymbol{j}$ $(i = 1, 2, 3)$. But 'pathological' initial conditions as, for

[39] Note: $H_{i+3} = -H_i$ for $i = 1, 2, 3$

example, $\langle\rho\rangle = const$, $\langle\boldsymbol{j}\rangle = 0$, $\langle h_1\rangle = 0 = \langle h_3\rangle$ and $\langle h_2\rangle = h_0 \sin(2\pi y/W)$ lead to an exitation of an oscillatory mode in j_x due to a coupling of the hydrodynamic modes with the additional (kinetic) modes of the lattice gas (Zanetti, 1989).

Further reading: Kadanoff et al. (1989), d'Humières et al. (1989, 1990), Bernardin (1992), Qian (1997). Ernst (1991) calculates the contribution of the unphysical modes to the stress tensor for FHP.

Exercise 3.8.1. (**)
Consider a lattice with 3 times 3 sites which is empty except for the site $(2,2)$ in the middle of the domain. Calculate the Zanetti invariants 1.) with the initial ($t = 0$) configurations of Fig. 3.2.2 at site $(2,2)$ and 2.) after collision and propagation ($t = 1$). In addition prove that H_i are conserved by a single propagating particle.

Exercise 3.8.2. (**)
Qian et al. (1992) proposed a LGCA for diffusion in 1D with four velocities c_i (D1Q4) and two different masses m_i:

$$\begin{aligned} c_i &= \{2c, -2c, c, -c\} \\ m_i &= \{m, m, 2m, 2m\} \end{aligned}$$

There is only one possible collision which conserves mass, momentum and energy: the head-on collision between a fast (hot) particle with mass m and a slow (cold) particle with mass $2m$ which leads to a reversal of their velocities. For a domain with L sites where L is divisible by four and periodic boundary conditions find three spurious invariants staggered in space and time and one additional spurious invariant staggered in space only.

Exercise 3.8.3. (**)
Can you find staggered invariants for HPP?

Exercise 3.8.4. (***)
Try to find staggered invariants for PI.

Exercise 3.8.5. (***)
Try to find staggered invariants for D2Q9.

Exercise 3.8.6. (***)
Modify the usual procedure to initialize a given distribution of ρ, $\boldsymbol{j}$ and staggered-momentum densities h_i (the microscopic densities corresponding to H_i).

3.9 Lattice-gas cellular automata: What else?

Further reading:
Textbooks: Rothman and Zaleski (1997), Rivet and Boon (to be published).
Proceedings and reviews:

- Proceedings of the workshop on Large Nonlinear Systems, Complex Systems, Vol. 1, No. 4, 1987.
- Monaco (1989)
- Doolen (1990)
- Ernst (1991)
- Chen et al. (1995)
- Boon et al. (1996)
- Biggs and Humby (1998)

Bibliographies: Doolen (1990) and J. Stat. Phys. 68 (3/4), 611-667, 1992.

Further topics:

- Mode-mode coupling, long time-tails, divergence of transport coefficients in 2D: Kadanoff et al. (1989), Frenkel and Ernst (1989), d'Humières et al. (1989), McNamara (1990), van der Hoef and Frenkel (1990), Ernst (1991).
- Flows in 3D: Bernsdorf et al. (1999), van Genabeek and Rothman (1999).
- Flow through porous media: Balasubramanian et al. (1987), Rothman (1988), Kohring (1991a,b,c), Knackstedt et al. (1993), Gutfraind et al. (1995), van Genabeek and Rothman (1996), Koponen et al. (1996, 1997), Krafczyk et al. (1998), Matsukama (1998), Matsukuma et al. (1998), Niimura (1998), Waite et al. (1998).
- Multiphase flows ('colored models') in 2D: Rothman and Keller (1988), Stockman et al. (1990), Gunstensen and Rothman (1991), Kougias (1993), Rothman and Zaleski (1994, review article), Emerton et al. (1997), Matsukama (1998), Peng and Ohta (1997), Stockman et al. (1997), Tsumaya and Ohashi (1997), Weig et al. (1997), Ebihara et al. (1998), Sehgal et al. (1999).
- Multiphase flows in 3D: Rem and Somers (1989), Olson and Rothman (1995, 1997).
- Flow of granular media: Karolyi et al. (1998), Manna and Khakhar (1998).

- Flow in dynamical geometry: Hasslacher and Meyer (1998)
- Poisson solver: Chen, Matthaeus and Klein (1990).

- Magnetohydrodynamics (MHD): Chen and Matthaeus (1987), Hatori and Montgomery (1987), Montgomery and Doolen (1987), Chen et al. (1991), Succi et al. (1991), Chen et al. (1992), Martinez et al. (1994), Isliker et al. (1998), Takalo et al. (1999).
- Relativistic flows: Balazs et al. (1999).
- Chemical reactions, reaction-diffusion equations: Weimar et al. (1992), Weimar (1997), Decker and Jeulin (1997), Vanag and Nicolis (1999).
- Burgers equation: Boghosian and Levermore (1987), Cheng et al. (1991), Nishinari and Takahashi (1998).
- Generalized semi-detailed balance condition: Chen (1995, 1997).
- Beyond the Boltzmann approximation: Boghosian (1995).
- Pattern formation: Chen et al. (1995), Bussemaker (1996), Deutsch and Lawniczak (1999).
- Integer lattice gases, Digital Physics: Boghosian et al. (1997), Chen, Teixeira and Molvig (1997), Teixeira (1997).
- Quantum mechanics: Boghosian and Taylor (1997), Boghosian and Taylor (1998), Succi (1998), Yepez (1998).
- Further reading: Kohring (1992a,b), Hashimoto and Ohashi (1997), Suarez and Boon (1997), Tribel and Boon (1997), Buick (1998), Hashimoto et al. (1998), Lahaie and Grasso (1998), Masselot and Chopard (1998b), Nicodemi (1998), Tsujimoto and Hirota (1998), Ujita et al. (1998).