

Simulation and Scientific Computing II

Lattice Boltzmann Method (LBM)

D. Bartuschat, K. Pickl
Lehrstuhl für Systemsimulation

Sommersemester 2011

- Historical Background
- Cellular Automata
- Lattice Boltzmann Method
- Lattice Boltzmann Algorithm

Classical fluid simulation approach: the Navier-Stokes (NS) equations

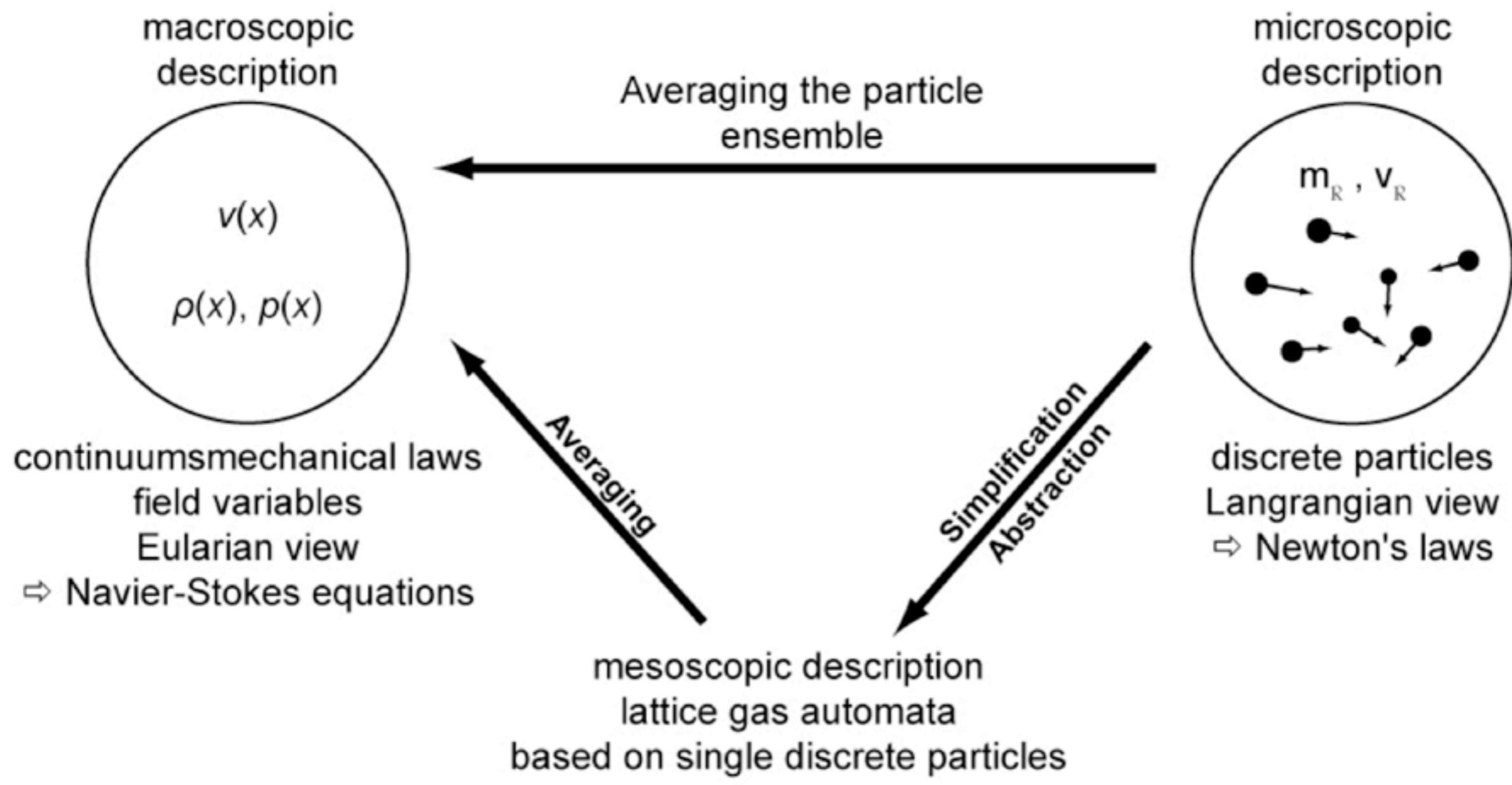
$$\frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \text{grad}) \vec{u} + \text{grad} p = \frac{1}{Re} \Delta \vec{u} + \vec{g}$$
$$\text{div } \vec{u} = 0 \quad (\text{incompressible})$$

- Direct discretization of the Navier-Stokes equations
- Non-linear velocity terms
- Global pressure (poisson) equation → large LSE

⇒ Search for alternative fluid simulation approaches

- Starting in the 1980s, the *Lattice Gas methods* were used to simulate the behavior of gases
 - Special case of cellular automata
 - Based on fluid mechanical principles
 - Time and space discretization ⇒ lattice
 - Fluid particles are positioned in certain lattice sites
 - May only move in certain, fixed directions
- Derivation of the lattice Boltzmann method using statistical averaging procedure
 - Simulation of fluid particle ensembles (not single particles)
 - Use of distribution functions





The Lattice Boltzmann method

Calculation of the spatial and temporal development of a particle distribution function in the discrete phase space



- For modeling and simulating
 - time dependent,
 - complex systems
- Discrete model of regular grid of cells
- Each cell has a state, depending on states of neighboring cells
- Intrinsically parallel
- Introduced by von Neumann (around 1940) to investigate self-reproducing systems, e.g. in biology.
- Famous: Conway's game of life



- „Cellular automaton“ invented by Cambridge mathematician John Conway
- Game became widely known when mentioned in article published by Scientific American in 1970.
- Consists of a collection of cells which can live, die or multiply, based on a few mathematical rules
- Depending on initial conditions, cells form various patterns throughout the course of the game

(From <http://www.bitstorm.org/gameoflife/>)



- For a space that is „populated“:
 - Each cell with one or no neighbors dies, as if by loneliness
 - Each cell with four or more neighbors dies, as if by overpopulation
 - Each cell with two or three neighbors survives
- For a space that „empty“ or „unpopulated“
 - Each cell with three living neighbors becomes populated in next generation
 - Other empty cells stay empty.
- Game of life - Demo

(From <http://www.bitstorm.org/gameoflife/>)



- regular grid (lattice)
 - cells (lattice cells)
 - covering portion of d-dimensional space
- set $S(x,t) = \{S_1(x,t), S_2(x,t), \dots, S_n(x,t)\}$ of Boolean variables depending on position and time
- local rule (state transition rule)
 - specifies the time evolution
 - $S_i(x,t) = R_i(S(x,t), S(x+d_1,t), S(x+d_2,t), \dots, S(x+d_q,t))$
 - depending on state variables of previous time step in a *neighborhood* (2D: *von-Neumann-Neighborhood or Moore-Neighborhood*).
- Extensions:
 - probabilistic rules
 - integer or real valued states, ...

- CA require an access pattern similar to elliptic PDE solvers
 - local data access patterns
 - Jacobi-like scheme
 - von Neumann = 5-point stencil
 - Moore = 9-point stencil
- Update rules homogeneous, i.e. do not depend on location or time
spatial or temporal behavior exclusively determined by definition of the state variables

- CA are used to model physical systems
 - Biology, ecology, physics, chemistry, material sciences
 - Example: grass-rabbit-fox predator-prey system by M.Eigen - DEMO
 - Easy to create models
 - but weak verification (usually, but not for the LBM)
- CA also of interest in (theoretical) computer science
 - e.g. in terms of complexity theory, CA are universal computing machines
 - a model of parallel computation
 - links to Neural nets, cryptography

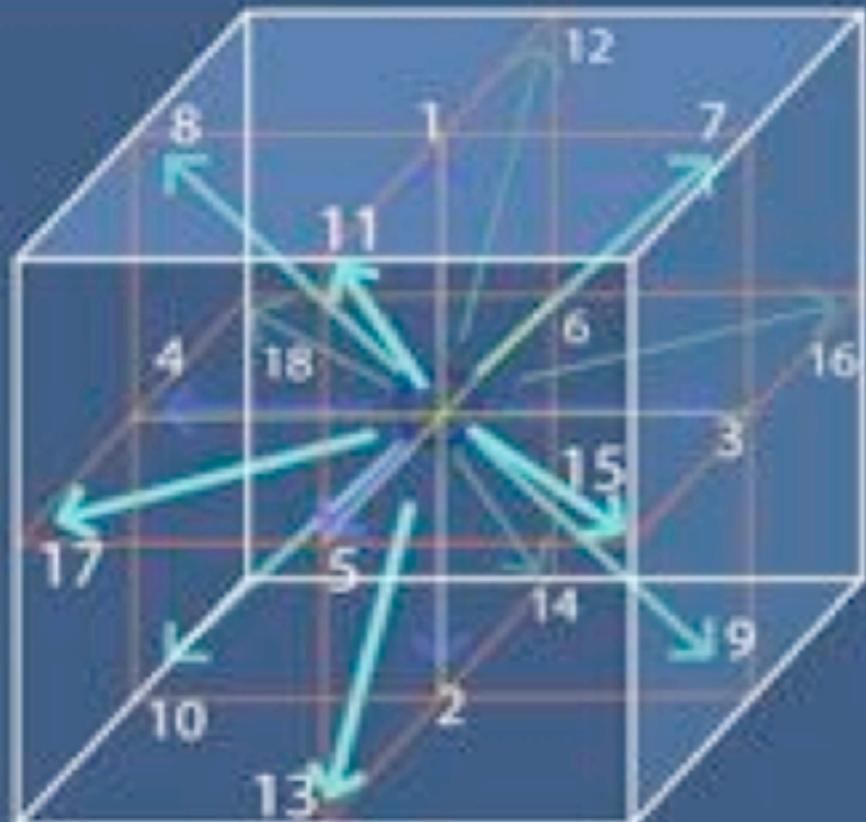
- Each cell of a CA can contain gas particles or be empty
- Gas particles have a velocity
- In each time step, the particles move (according to their velocity) to neighboring cells (stream step)
- Multiple particles in cells collide, possibly randomized
- only discrete particles necessary
- discrete set of velocities
- discrete positions



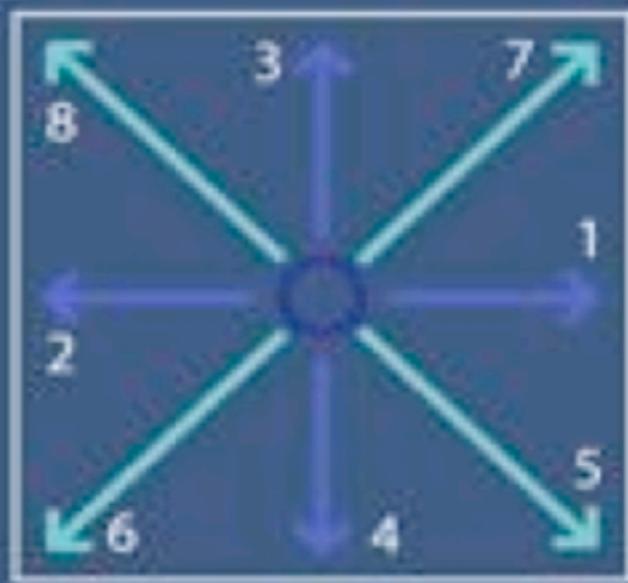
- collision rules must satisfy elementary physical principles
 - conservation of mass
 - conservation of momentum
 - conservation of energy
- Lattice Gas Method can provide a physically correct representation of flow phenomena,
- but only in the limit when the mesh size goes to zero
- averaging over many cells necessary to recover physically meaningful quantities
- In practice very fine meshes necessary
- very compute intensive

- Real valued representation of particles (density)
- still discrete velocities and positions
- stream step and collide step
- Alternative to classical Computational Fluid Dynamics (CFD) as taught in NuSiF
 - use PDE (Navier-Stokes equations or simplified model, e.g. Stokes or Euler equations)
 - discretize PDE by either
 - Finite Differences
 - Finite Volumes
 - Finite Elements
 - Solve (large) systems of (nonlinear) algebraic equations

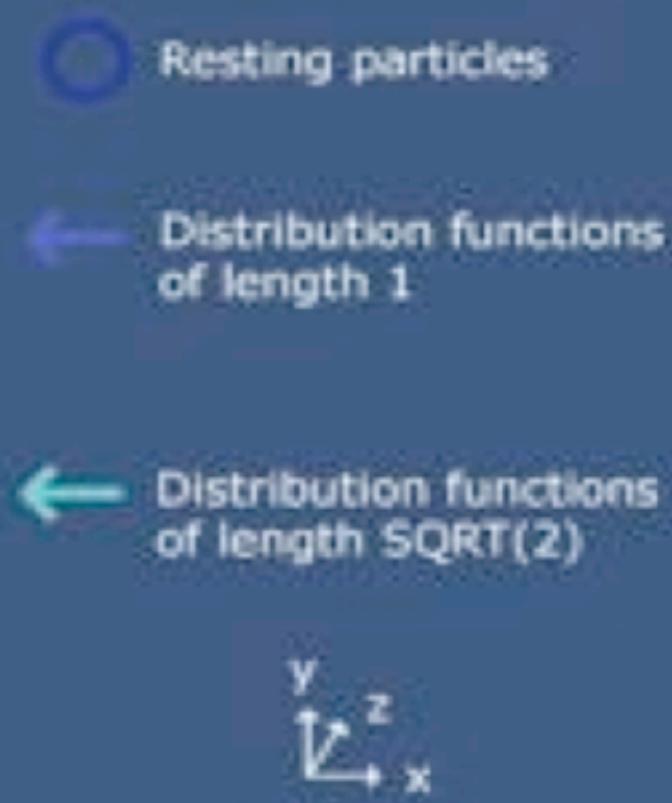
for fluid flow simulations



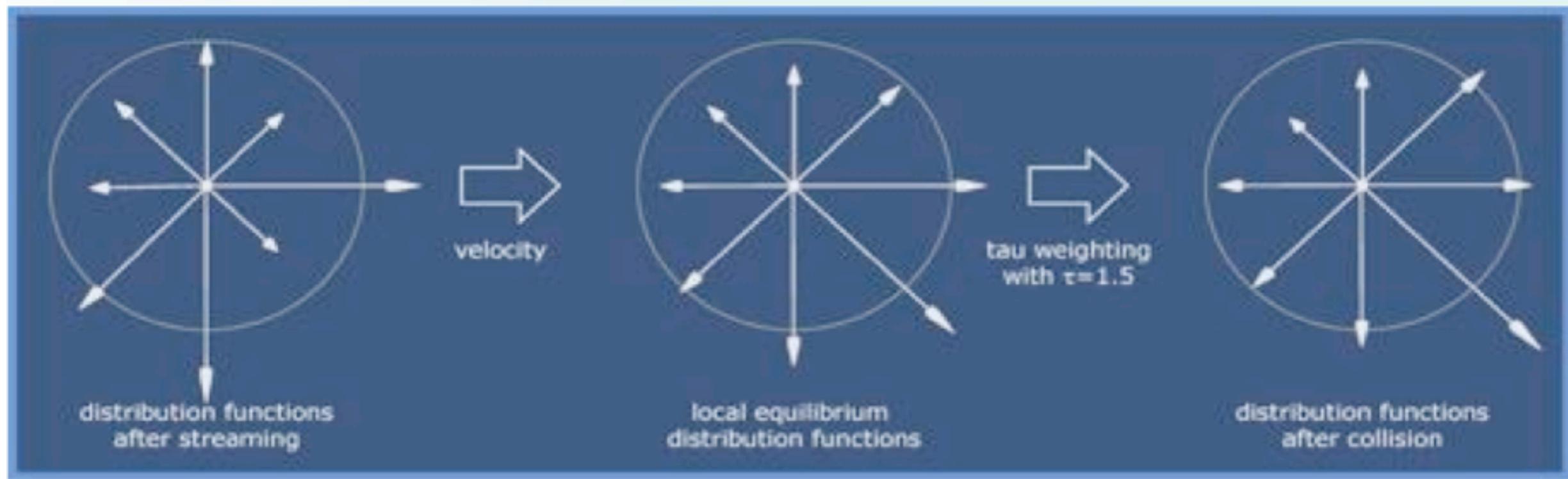
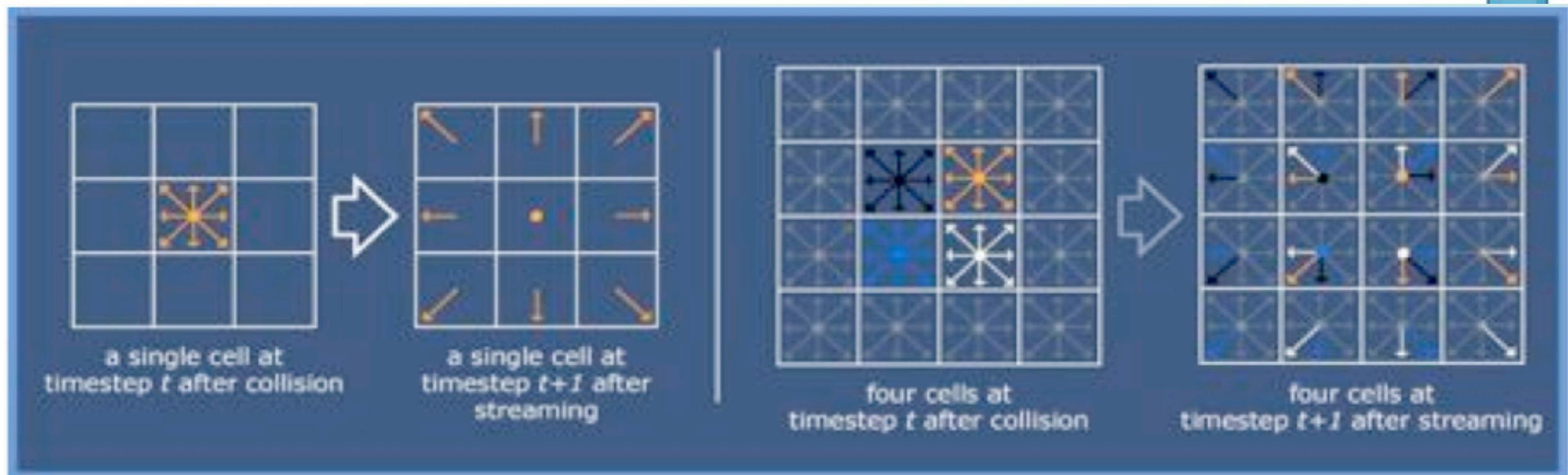
D3Q19 lattice cell



D2Q9 lattice cell



y
z
x



- Simple algorithm
- Easy to parallelize
- Easy to adapt, e.g. for
 - complicated geometries
 - time-varying geometries
 - (some) additional physical or chemical effects

- Compute-intensive
 - Some extensions (stretched grids, adaptive grids) not available / not yet developed / complicated
 - Only few commercial software (PowerFlow/Exa)

- $n_i(x,t)$ = number of particles at lattice point x , time t , moving in direction i
- $n_i(x,t)$ is a positive integer, often only the values 0 and 1 are allowed
- kinetic equation:
$$n_i(x + c_i \Delta t, t + \Delta t) = n_i(x, t) + \Delta_i$$
- c_i : lattice velocity, Δt time step (often = 1)
- Δ_i : collision function

- Expand the right hand side using Taylor's theorem for small Δt :

$$n_i(x + c_i \Delta t, t + \Delta t) = n_i(x, t) + \Delta t \frac{\partial n_i}{\partial t} + c_i \Delta t \nabla n_i + \mathcal{O}(\Delta t^2)$$

- Neglecting higher order terms this gives:

$$\frac{\partial n_i}{\partial t} + c_i \nabla n_i = \frac{\Delta_i}{\Delta t}$$

- This is the Boltzmann transport equation.

- Both lattice gas and lattice Boltzmann use models with
 - a discrete time step Δt ,
 - a discrete lattice of points,
 - a fixed number of directions / lattice velocities.
- Lattice gas uses a *discrete* (positive integer) number of particles (in each cell).
- Lattice Boltzmann uses *positive real numbers* to represent some average number of particles traveling at a lattice point in any of the lattice directions.
 - These numbers are called *distribution functions*.
 - Mathematically they are *expected* values of the statistical distribution functions for the lattice point.

- $f_i(x, t) \geq 0$ real valued (i.e. a floating point number):
average number of particles traveling at node x at time t in direction c_i

- Density:

$$\rho = \sum_{i=0}^{N-1} f_i$$

- Momentum density:

$$\rho u = \sum_{i=0}^{N-1} f_i c_i$$

- The Boltzmann equation:

$$\frac{\partial f}{\partial t} + \xi \nabla_x f + K \nabla_\xi f = Q(f, f), \quad f = f(t, x, \xi).$$

- describes the dynamic of particle position probability in phase space:

phase space (6D) = position space (3D) + velocity space (3D) .

- Complicated integro-differential equation due to its collision term.

Assumptions for Q : binary collisions and molecular chaos.

$f(t, x, \xi)$ is the probability to encounter particles with the continuous microscopic velocity ξ at position x at time t .

- Modeling of the complex collision term \Rightarrow Krook equation,
BGK (Bhatnagar-Gross-Krook) equation:

$$\frac{\partial f}{\partial t} + \xi \nabla_x f = -\frac{1}{\tau} (f - f^{(0)}).$$

- with τ : relaxation time depending on the viscosity of the simulated fluid,
- $\omega = \frac{1}{\tau}$: Collision frequency [0.5, 1.95]
- Simplified collision operator used, as considering particle-particle collisions would be very complicated.

$$\frac{\partial f}{\partial t} + \xi \nabla_x f = -\frac{1}{\tau} (f - f^{(0)})$$

TECHNISCHE
FAKULTÄT

Knudsen number: $Kn = \frac{\text{mean free path length}}{\text{characteristic length scale}}$.

- If ($Kn = \varepsilon \ll 1$) and in case of very small deviations from the local equilibrium, f in relation to the microscopic velocity space can be approximated with only a few degrees of freedom.
- The physical approximation of the distribution function f with values f_i at N collocation points, which move with the velocity c_i , is

$$f(t, x, \xi) \implies f(t, x, \xi) = \tilde{f}_i(t, x).$$

$$\frac{\partial f}{\partial t} + \xi \nabla_x f = -\frac{1}{\tau} (f - f^{(0)})$$

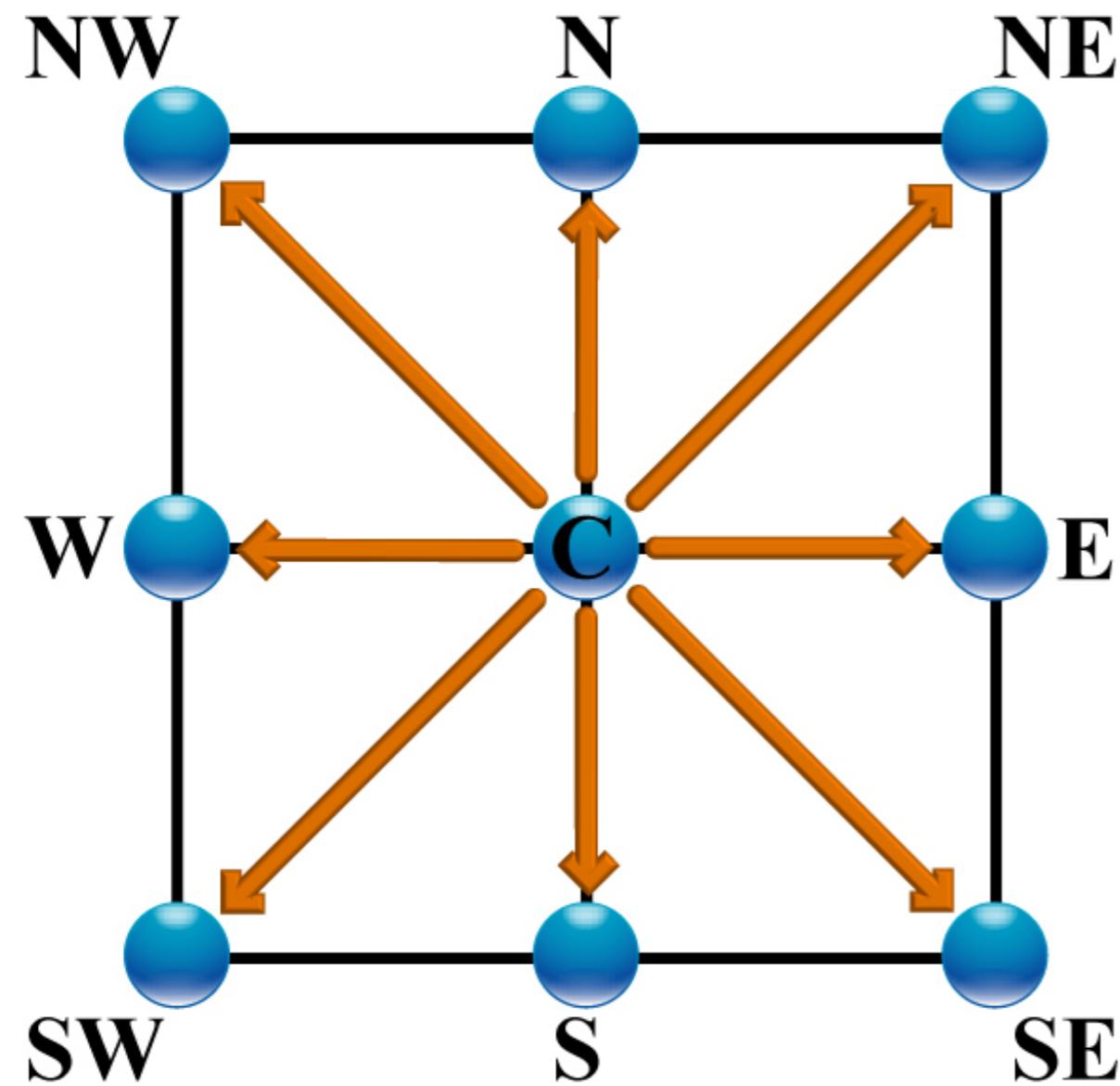
- Using this approach, we can derive the *velocity discrete Boltzmann equation*:

$$\frac{\partial f_i}{\partial t} + c_i \nabla_x f_i = -\frac{1}{\tau} (f_i - f_i^{eq})$$

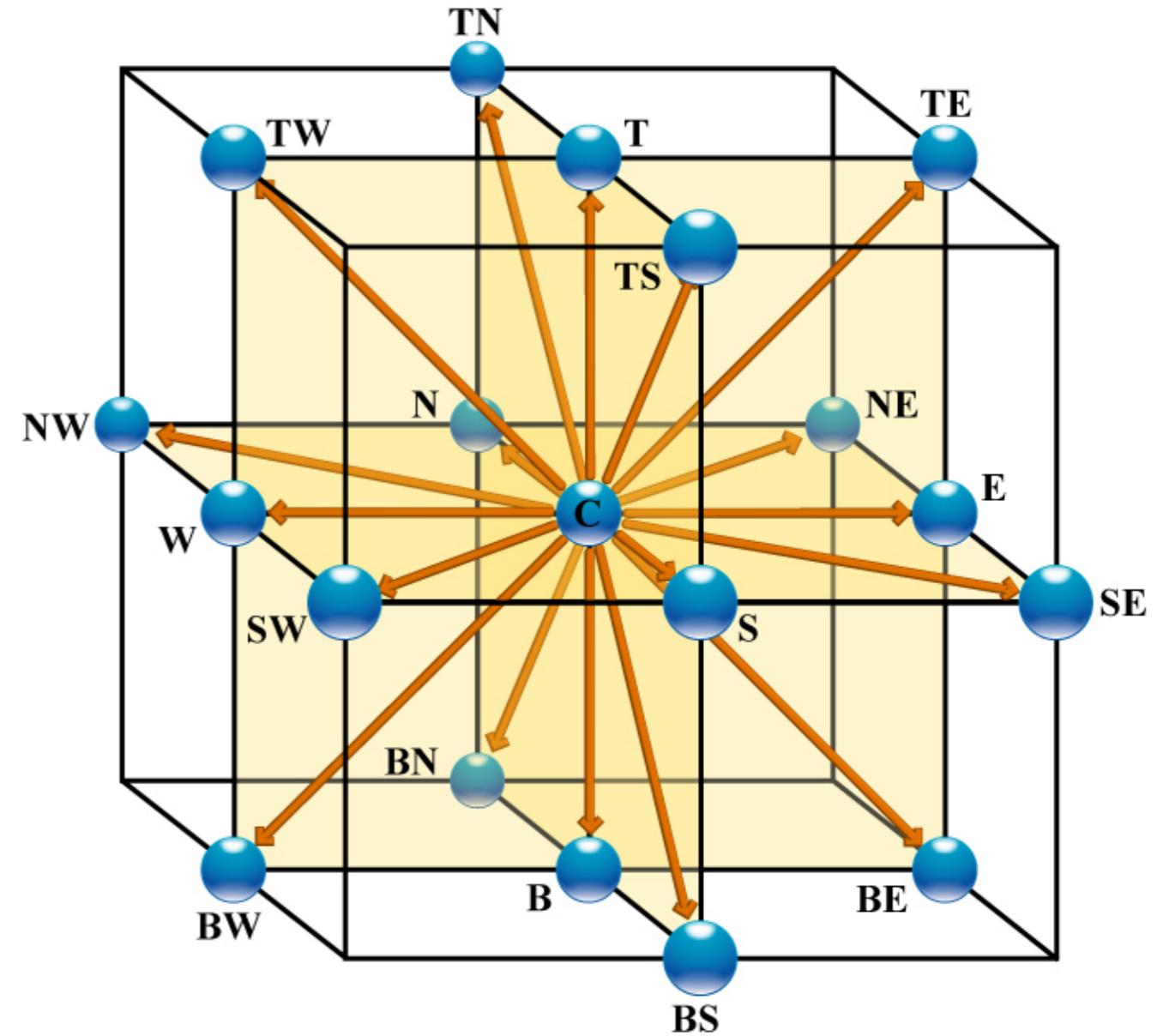
c_i : Discrete lattice velocities

f_i^{eq} : local equilibrium distribution functions

Lattice Boltzmann models for the 2D and 3D case:



D2Q9 model



D3Q19 model

$$\frac{\partial f_i}{\partial t} + c_i \nabla_x f_i = -\frac{1}{\tau} (f_i - f_i^{eq})$$

- Time discretization \Rightarrow explicit forward Euler
- Spatial discretization along the lattice velocities c_i

$$\frac{f_i(x, t + \Delta t) - f_i(x, t)}{\Delta t} + \|c_i\|_2 \frac{f_i(x + \Delta x_i, t + \Delta t) - f_i(x, t + \Delta t)}{\|\Delta x_i\|_2} = -\frac{1}{\tau} (f_i - f_i^{eq}).$$

- Explicit lattice Boltzmann equation:

$$f_i(x + c_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} (f_i - f_i^{eq}).$$

- The LBM is a second order scheme

$$f_i(x + c_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} (f_i - f_i^{eq}).$$

- This is the discrete lattice Boltzmann equation for the *single time relaxation approximation*.
- Relaxation constants:
 - $\tau = 1$: The distribution functions are exactly set to the equilibrium distribution.
 - $\tau = 2$: The distribution functions are midway between incoming distribution and the equilibrium distribution.
- In realistic simulations, outside forces must still be included (e.g. gravity).
- The equilibrium distribution must be found from physical arguments.

$$f_i(x + c_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} (f_i - f_i^{eq}).$$

- f_i^{eq} is the *equilibrium distribution function*
- defines the local particle distribution for a certain velocity u and a certain density ρ when the fluid has reached a state of equilibrium

$$f_i^{eq} = f_i^{eq}(\rho, u) = t_i \rho \left(1 + \frac{3}{c^2} c_i u + \frac{9}{2c^4} (c_i u)^2 - \frac{3u^2}{2c^2} \right)$$

c_i : discrete lattice velocity

t_i : direction dependent weighting factor

- Weighting factor t_i for the D2Q9 model:

$$t_i = 4/9 \quad \text{for } i = C$$

$$t_i = 1/9 \quad \text{for } i = N, E, S, W$$

$$t_i = 1/36 \quad \text{for } i = NW, NE, SW, SE$$

- Weighting factor t_i for the D3Q19 model:

$$t_i = 1/3 \quad \text{for } i = C$$

$$t_i = 1/18 \quad \text{for } i = N, E, S, W, T, B$$

$$t_i = 1/36 \quad \text{for } i = \text{all 12 others.}$$

$$f_i(x + c_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} (f_i - f_i^{eq})$$

The equation is solved in two steps:

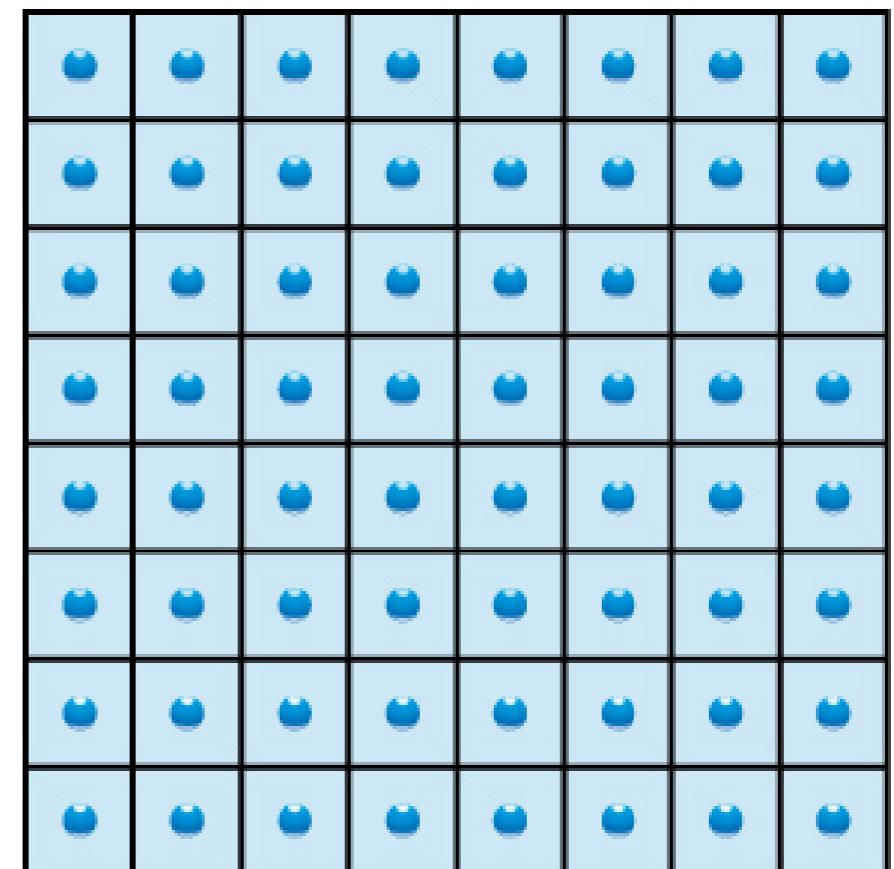
- The *collide* step

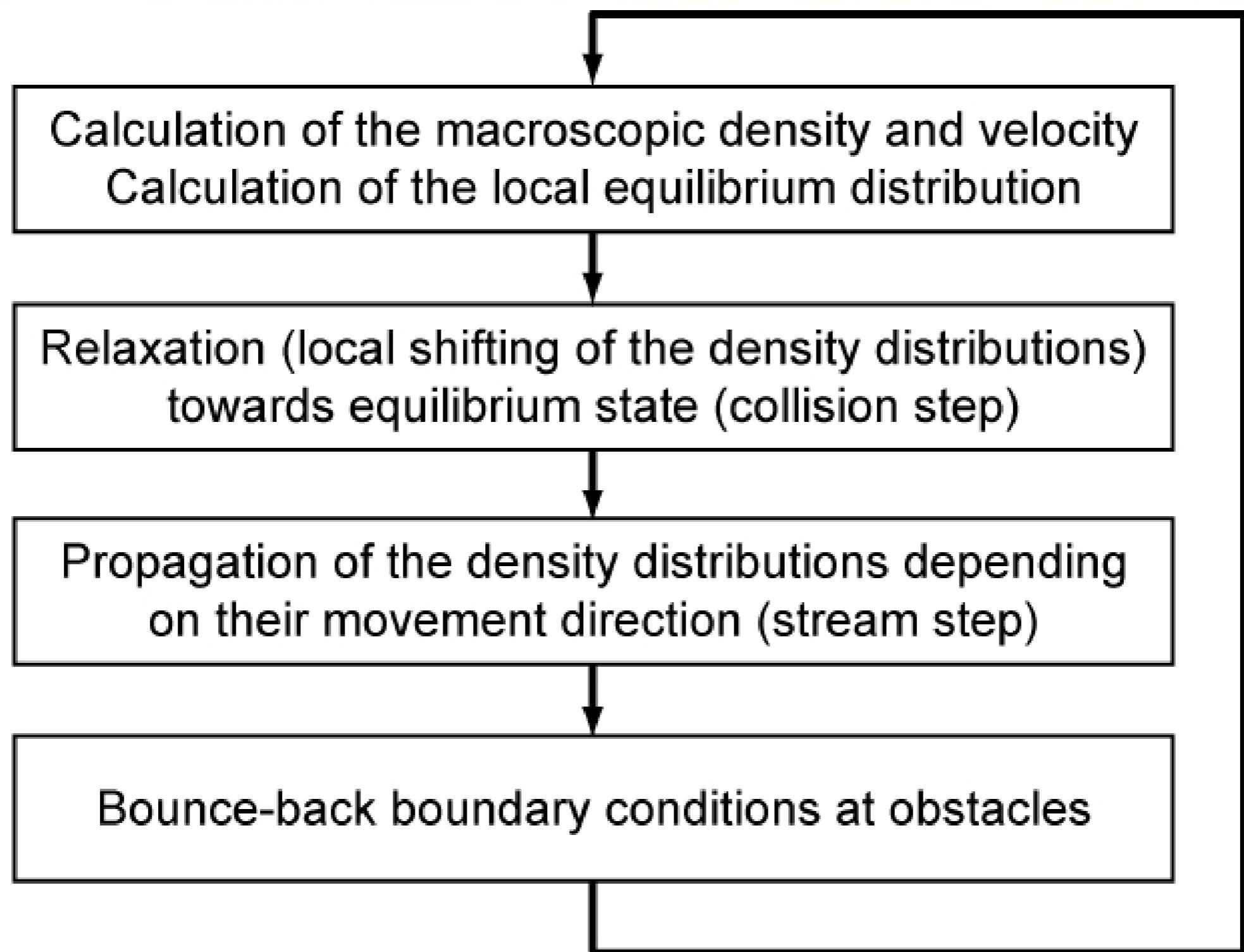
$$\tilde{f}_i(x, t + \Delta t) = f_i(x, t) - \frac{1}{\tau} (f_i - f_i^{eq})$$

- The *streaming* step

$$f_i(x + c_i \Delta t, t + \Delta t) = \tilde{f}_i(x, t + \Delta t)$$

- Simulation domain discretization in „voxels“ (2D: squares, 3D: cubes)
- Each voxel/cell contains N distribution functions (2D: 9, 3D: 19)

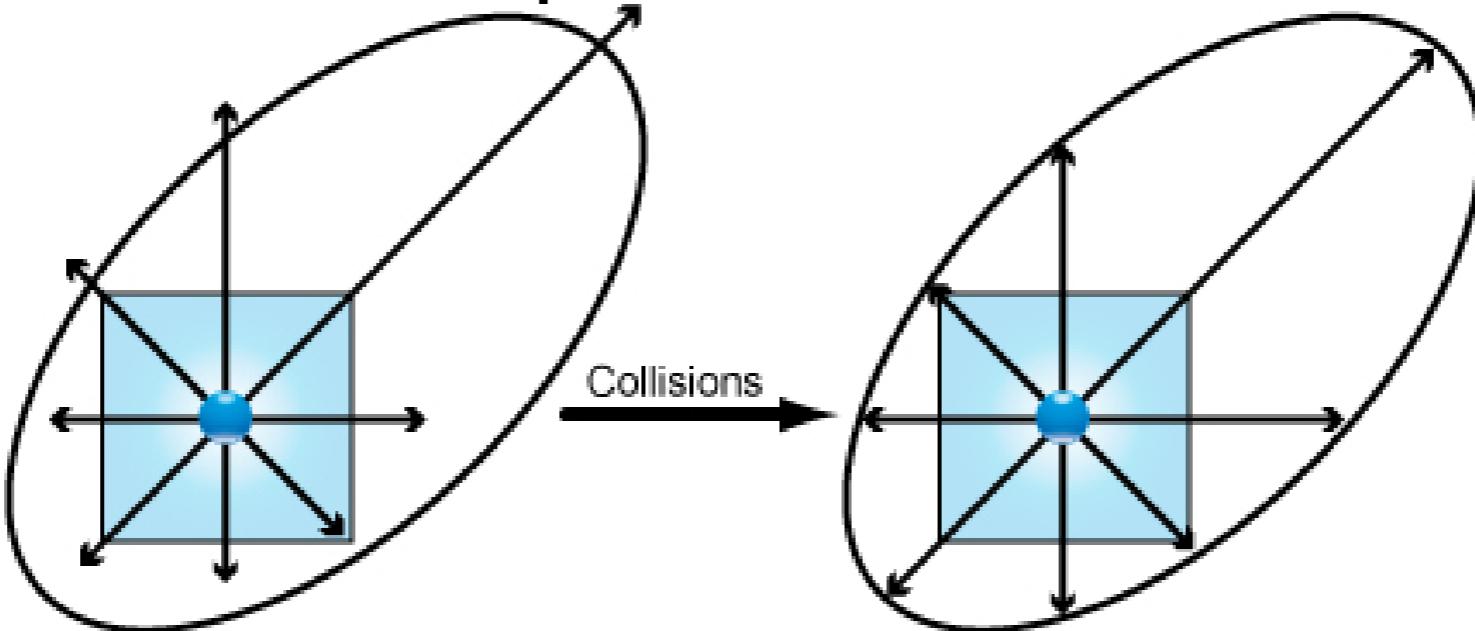




Two possible update orders:

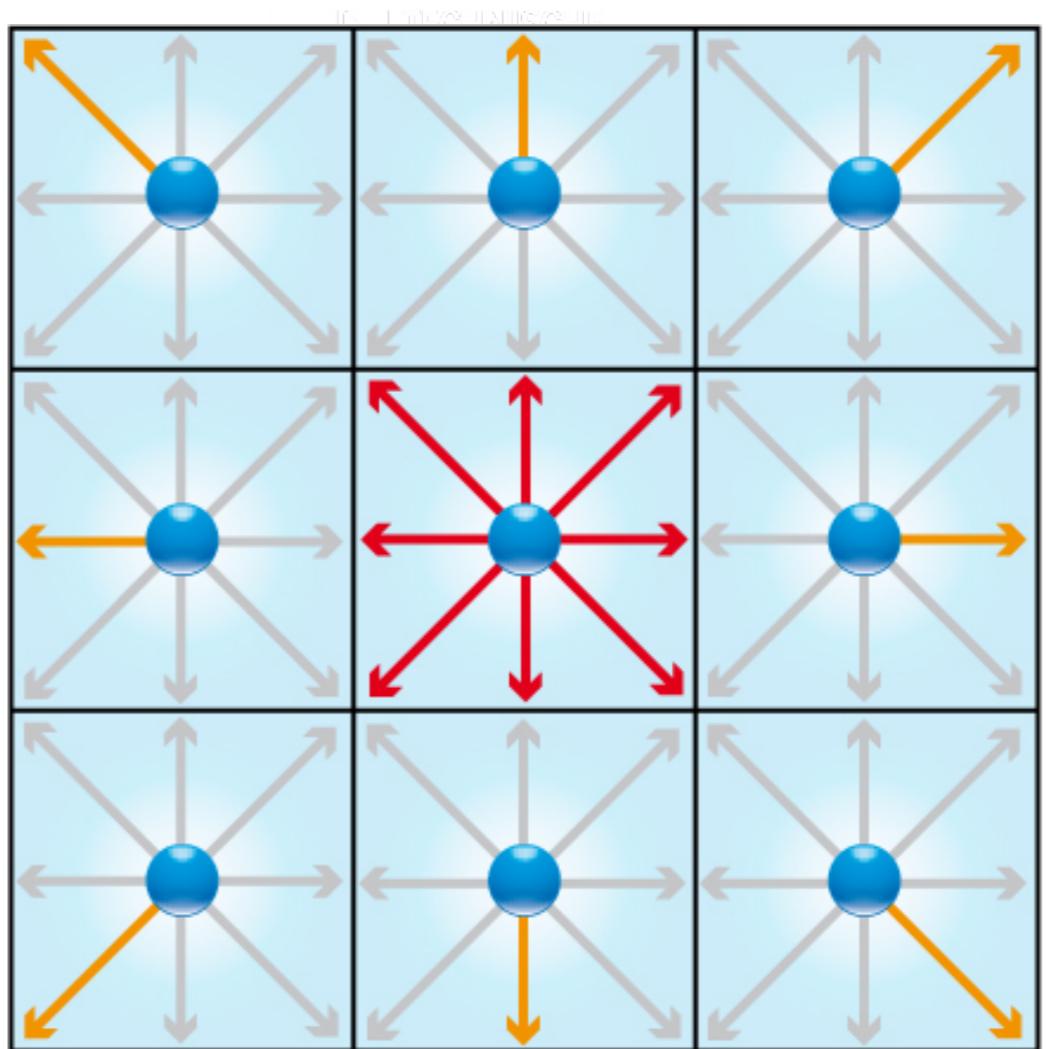
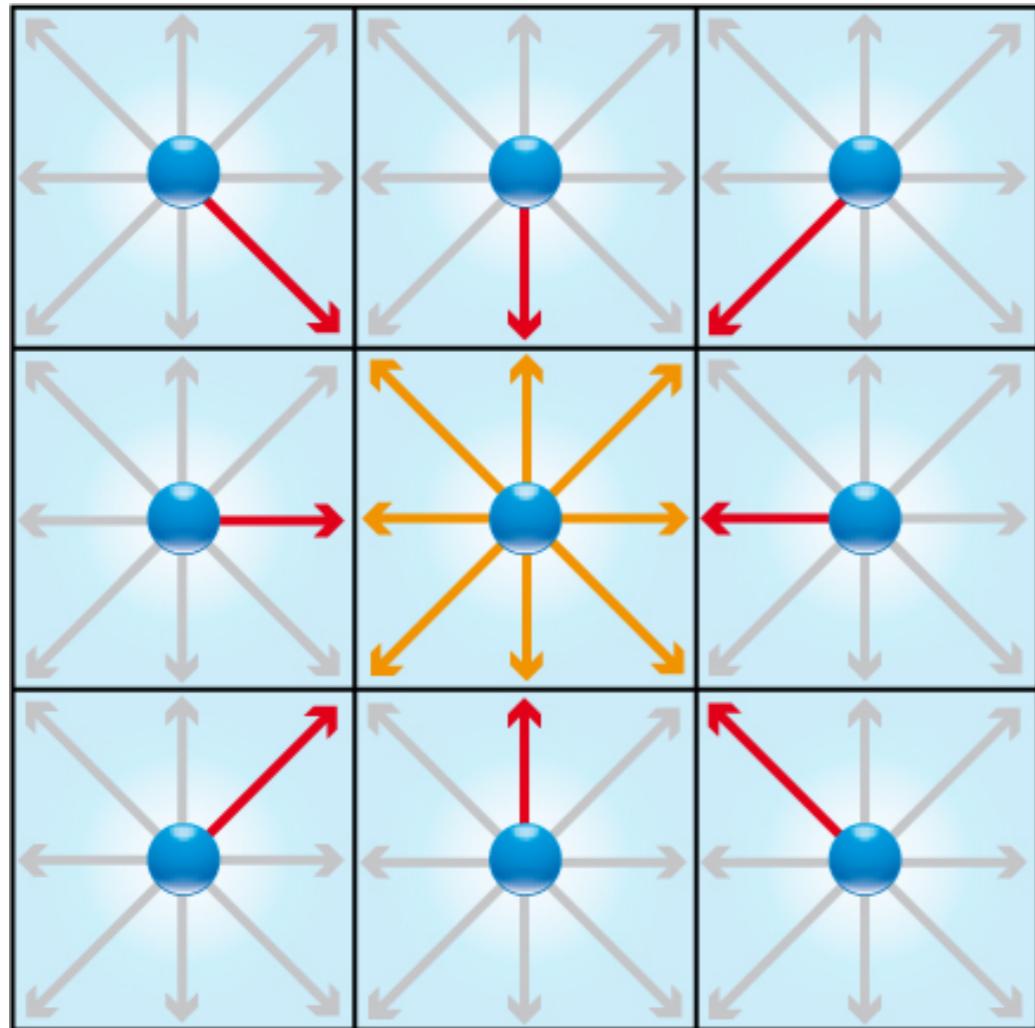
- Stream-collide:
The distribution functions are first streamed to the corresponding neighboring cells. Then a collision step is calculated (pull implementation)
- Collide-stream:
First, a collision step is calculated, after which the new distribution functions are streamed (push implementation)

Performing a collision step



- Models various fluid particle interactions (collisions, ...)
- Weighting between the equilibrium distribution functions and the distribution functions form the previous time step
- Relaxation towards equilibrium state
- Preservation of mass and momentum

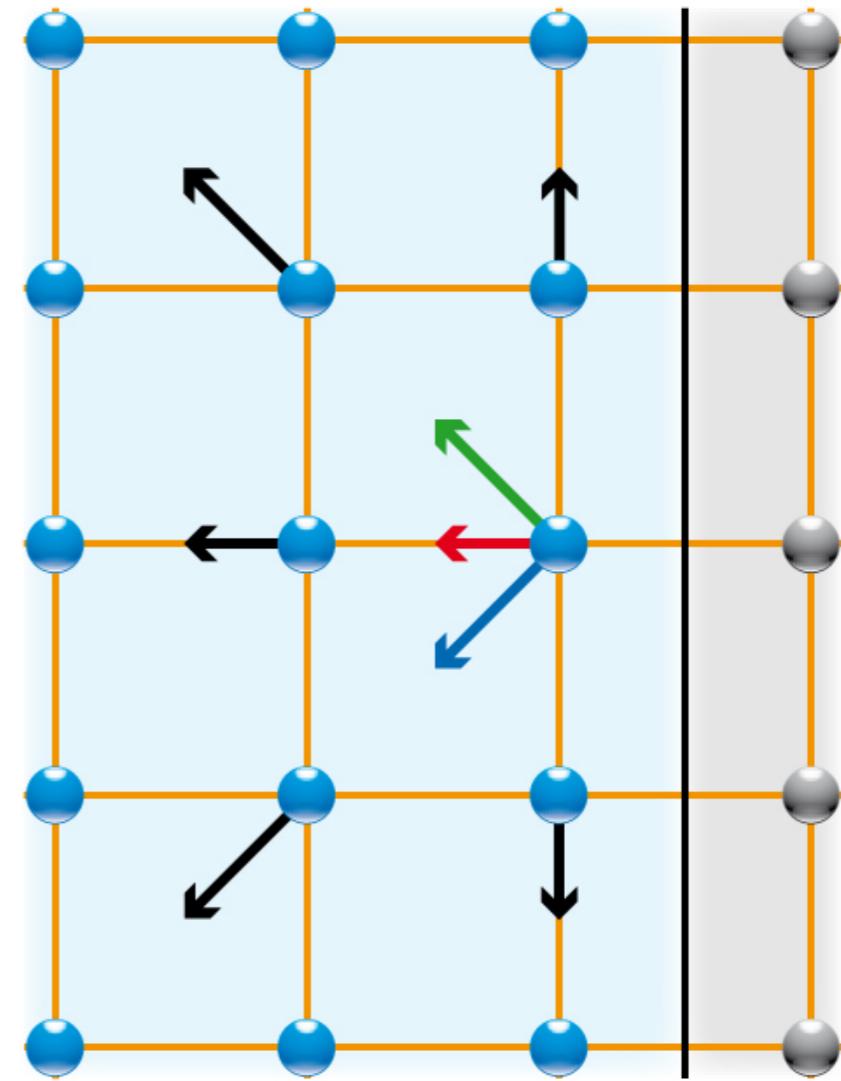
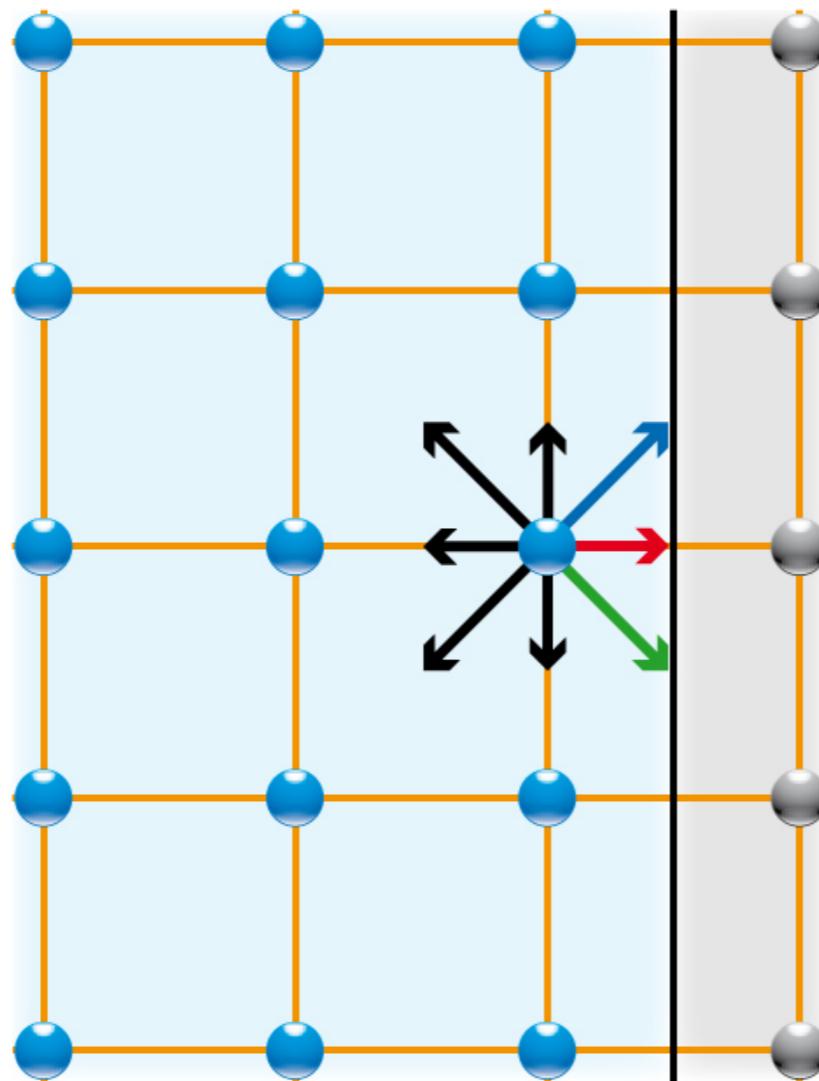
Performing a streaming step



Treatment of boundary conditions

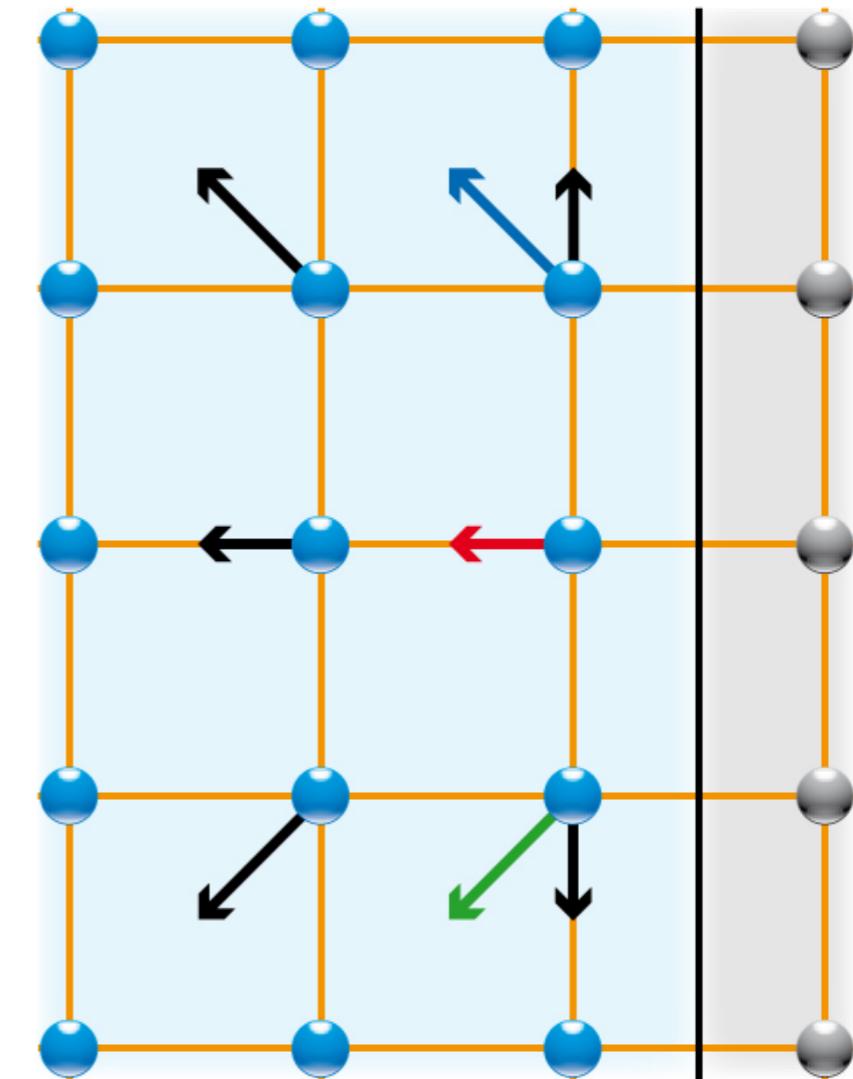
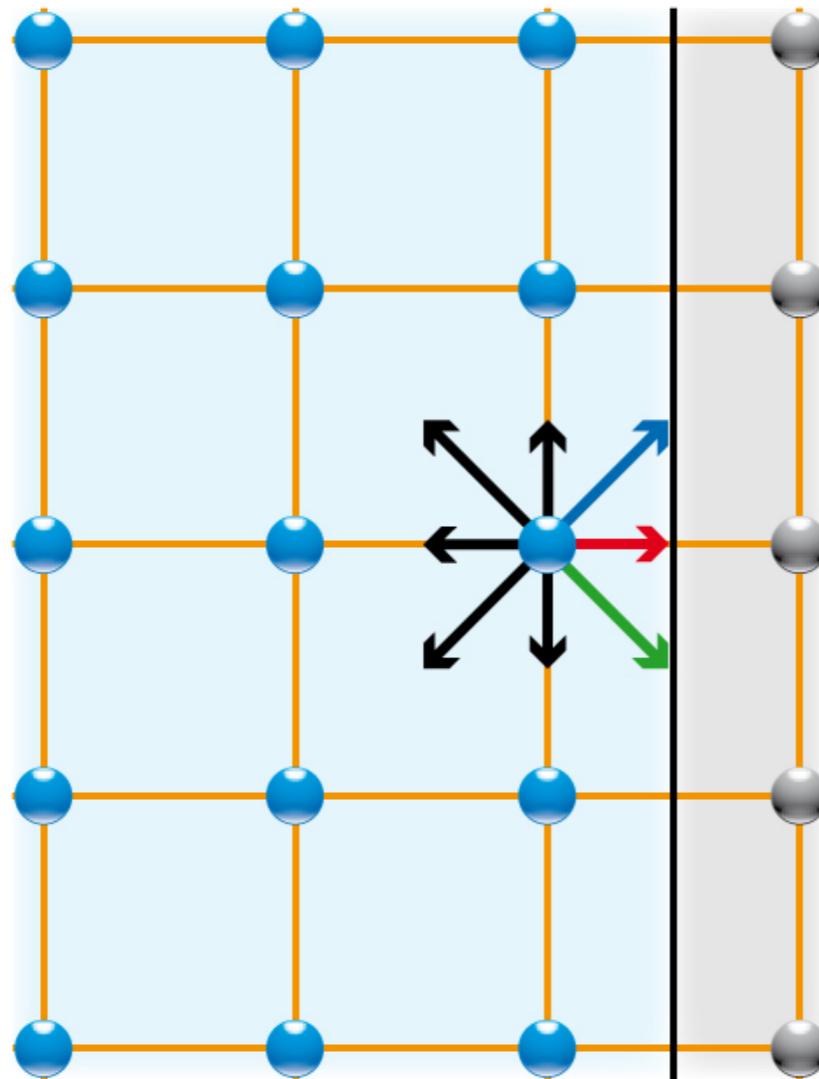
- *No-slip* boundary conditions:

- Used in case of real wall (friction)
- Fluid velocity at the wall is reduced to zero



- *Free-slip boundary conditions:*

- Used in case of symmetric boundary conditions
- Fluid velocity normal to the wall is reduced to zero



■ *Moving no-slip boundary conditions*

- Movement of the wall involving friction induces a flow
- The added momentum is modeled by an additional term during the no-slip treatment

$$f_{\bar{\alpha}}(x, t) = f_{\alpha}(x, t) - 2t_i \rho \frac{3}{C^2} c_{\alpha} u_w$$

α : direction towards the wall

$\bar{\alpha}$: direction from the wall

t_i : same direction dependent parameter as for the equilibrium distributions

ρ : fluid density in the fluid cell near the wall

u_w : velocity of the wall

Example Problem: Lid-driven Cavity

