PARALLEL CFD 2011
23rd International Conference on
Parallel Computational Fluid Dynamics 2011
MAY 16-20 BARCELONA

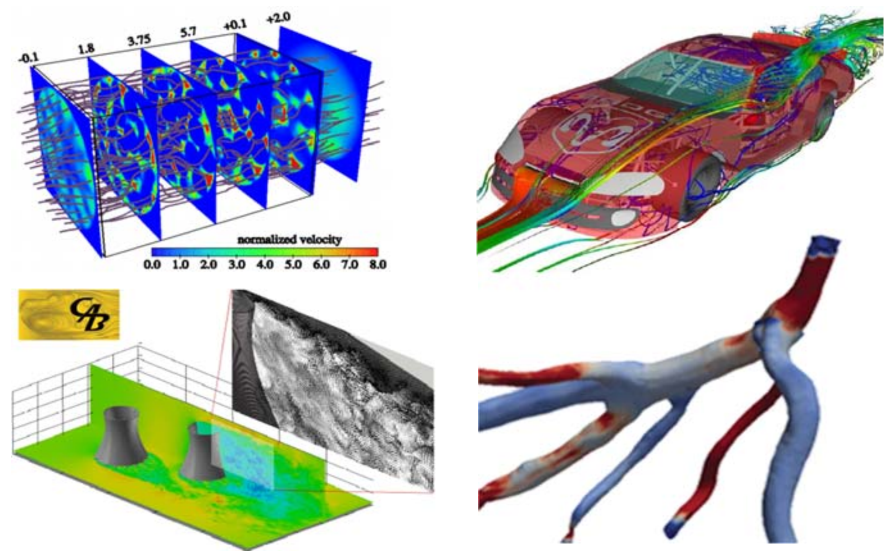
Brief introduction to LBM

Thomas Zeiser
Regionales Rechenzentrum Erlangen (RRZE)
University of Erlangen-Nuremberg (FAU), Germany
thomas.zeiser@rrze.uni-erlangen.de

This work is financially supported by:
FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
KONWIHR II
Federal Ministry of Education and Research
through grant SKALB (01IH08003)

Examples of flows calculated using LB solvers

Figures by courtesy of EXA GmbH, LS CAB-Braunschweig, Thomas Zeiser; Succi et al.



What is the “Lattice Boltzmann Method”?

I’m only talking about the stream-collide LB type with explicit time-stepping!

- **The fluid mechanist:**
  it’s a way to solve the Navier-Stokes equations
  in the nearly incompressible limit
  (“nearly incompressible” like in Chorin’s method of artificial compressibility)
- **The physicist:**
  it’s a very special discretization of the Boltzmann equation
- **The computer theorist:**
  it’s related to cellular automata
- **The computer scientist:**
  it’s a Jacobi-type iteration scheme with low computational intensity

Micro & macro world: Conservation laws on different levels

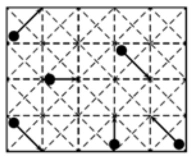
macroscopic description ← *ensemble averaging* ← microscopic description

averaging

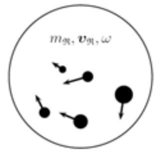
simplification abstraction

“mesoscopic” description

continuum mechanical laws;
field values; Eulerian point of view
→ Navier-Stokes equations



Lattice Gas Automata (LGA)
→ based on individual discrete particles

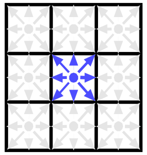


Discrete particles
Lagrangian point of view
→ Newton’s laws

Distribution functions
laws of statistical physics;
kinetic theory of gases
→ Boltzmann equation

Lattice Boltzmann methods (LBM)
Calculation of the spatial and temporal evolution of a particle distribution function in a discrete phase space

- Evolved from "cellular automata models"



- Physical basis: the Boltzmann equation

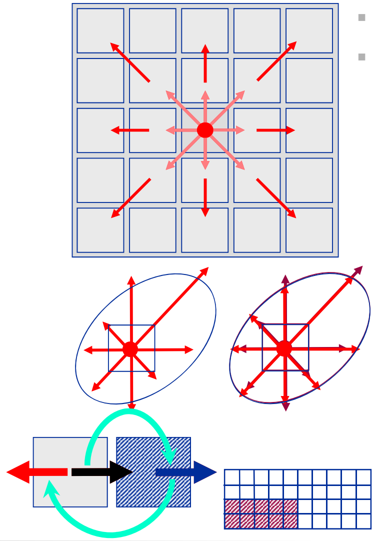
$$\frac{\partial f}{\partial t} + \boldsymbol{\xi}\frac{\partial f}{\partial \boldsymbol{x}} + \boldsymbol{K}\frac{\partial f}{\partial \boldsymbol{\xi}} = \mathcal{Q}(f,f)$$

- 1) approximation of the collision process by the BGK relaxation
- 2) physical discretisation → „velocity discrete Boltzmann eq."
- 3) numerical discretisation of spatial and temporal derivatives

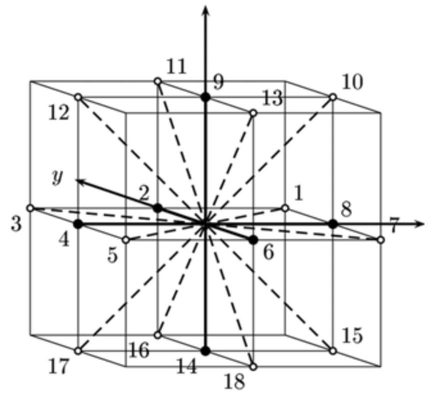
- ⇒ explicit lattice Boltzmann equation

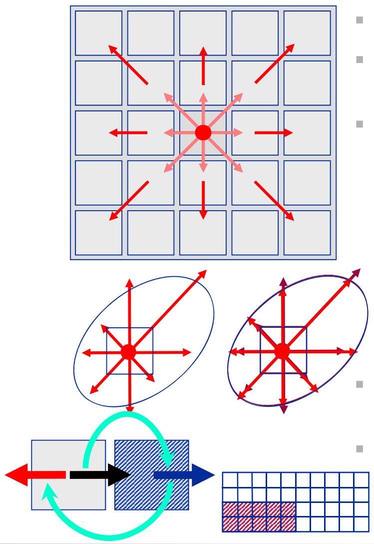
$$f_i(\boldsymbol{x}+\boldsymbol{c}_i,\, t+1) - f_i(\boldsymbol{x},\, t) = -\frac{1}{\tau}(f_i - f_i^{eq})$$

- satisfies the Navier-Stokes equations
  in the nearly incompressible limit with 2nd order accuracy
- numerical and computational advantages



- The domain is discretised into „voxel"
- in each voxel N distribution functions f_i are stored





- The domain is discretised into „voxel"
- in each voxel N distribution functions f_i are stored
- from time step to time step, these distribution functions move to the next cell according to their microscopic velocity direction (propagation)
  - the magnitude of the microscopic velocities is fixed by the lattice
  - however, the fraction of the particle density moving in all the directions is different
- through the collisions (relaxation) equilibrium is approached
- „bounce back" at solids means particle distributions which would enter a solid cell are put back to the original cell, but with opposite microscopic velocity

The evolution of the particle distribution functions f_i at each lattice cell is calculated in discrete time steps.

- Calculation of macroscopic flow quantities and the equilibrium distribution
- „Collision" relaxation (redistribution) of the particle distributions towards equilibrium
- „Propagation" of the distribution functions according to their directions to the next cells
- „Bounce back" at solid walls

computationally intensive, however purely local

memory operations only – however, involves only next-neighbor communication

$$\rho = \sum_i f_i$$

$$\rho\boldsymbol{u} = \sum_i \boldsymbol{c}_i f_i$$

$$f_i^{eq}(\boldsymbol{x}_*,\, t_*) = t_p\rho\left\{1 + \frac{\boldsymbol{c}_i\boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i\boldsymbol{u})^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{c_s^2}\right\}$$

$$\frac{1}{\tau}\Big(f_i(\boldsymbol{x}_*,\, t_*) - f_i^{eq}(\boldsymbol{x}_*,\, t_*)\Big) = \mathcal{Q}$$

with $\nu = \frac{2\tau-1}{6}$

$$f_i(\boldsymbol{x}_*+\boldsymbol{c}_i,\, t_*+1) = f_i(\boldsymbol{x}_*,\, t_*) - \mathcal{Q}$$

Summary: From Boltzmann to lattice Boltzmann

Boltzmann equation $\frac{\partial f}{\partial t} + \vec{\xi}\frac{\partial f}{\partial \vec{x}} + \vec{K}\frac{\partial f}{\partial \vec{\xi}} = Q(f,f)$

BGK approximation and neglecting external forces

$$\frac{\partial f}{\partial t} + \vec{\xi}\frac{\partial f}{\partial \vec{x}} = -\frac{1}{\tau}\left(f - f^{(0)}\right)$$

Physical discretisation $\Rightarrow$ velocity discrete Boltzmann eq.

$$\frac{\partial f_i}{\partial t} + \vec{c}_i\frac{\partial f_i}{\partial \vec{x}} = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

Numerical discret.: Finite differences for spatial / temporal derivatives

$$\frac{f_i(\vec{x}_*, t_*+\Delta t) - f_i(\vec{x}_*, t_*)}{\Delta t} + \vec{c}_i \frac{f(\vec{x}_*+\Delta\vec{x}_i, t_*+\Delta t) - f(\vec{x}_*, t_*+\Delta t)}{\Delta \vec{x}_i} = -\frac{1}{\tau}\cdots$$

Explicite lattice Boltzmann equation (using $\Delta\vec{x}_i = \vec{c}_i\Delta t,\ \Delta t = 1$)

$$f_i(\vec{x}_*+\vec{c}_i, t_*+1) - f_i(\vec{x}_*, t_*) = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

„The" lattice Boltzmann method ?!

Boltzmann equation
$\frac{\partial f}{\partial t} + \xi\frac{\partial f}{\partial x} + K\frac{\partial f}{\partial \xi} = Q(f,f)$

small Knudsen number

Chapman-Enskog-Expansion

continuity equation
$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0$$

Navier-Stokes equations (j=1,2,3)
$$\rho\left(\frac{\partial u_j}{\partial t} + u_i\frac{\partial(\rho u_j)}{\partial x_i}\right) = -\frac{\partial p}{\partial x_i} - \frac{\tau_{ij}}{\partial x_i} + \rho g_j$$

Chapman-Enskog-Expansion

Bhatnagar-Gross-Krook-Approximation

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

small Knudsen number
small Mach number

discretisation of momentum space

velocity discrete BGK equation
$\frac{\partial f_i}{\partial t} + c_i\frac{\partial f_i}{\partial x} = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$

discretisation of space and time

Lattice Boltzman equation (LBGK)
$$f_i(t_*+\Delta t, x_*+c_i\Delta t) = f_i(t_*, x_*) - \frac{\Delta t}{\tau}\left(f_i(t_*, x_*) - f_i^{eq}(\rho(x_*, t_*), u(x_*, t_*))\right)$$

Final remarks: 1991 / 2011

- ***David H. Bailey***
  Supercomputing Review, August 1991, p. 54-55
  "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers"

1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this fact.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on Crays.
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If MFLOPS rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
9. Quote performance in terms of processor utilization, parallel speedups or MFLOPS per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

2011 – Fooling the masses

- **Scalability vs. Performance !?**

  Only time-to-solution matters

- **Strong vs. Weak scaling !?**
- **Single vs. Double precision !?**
- **2D or 3D !?**
- **Base line system / algorithm / implementation !?**

- http://blogs.fau.de/hager/category/fooling-the-masses/
- http://blogs.fau.de/hager/files/2010/07/thirteen-ways-eihecs6.pdf