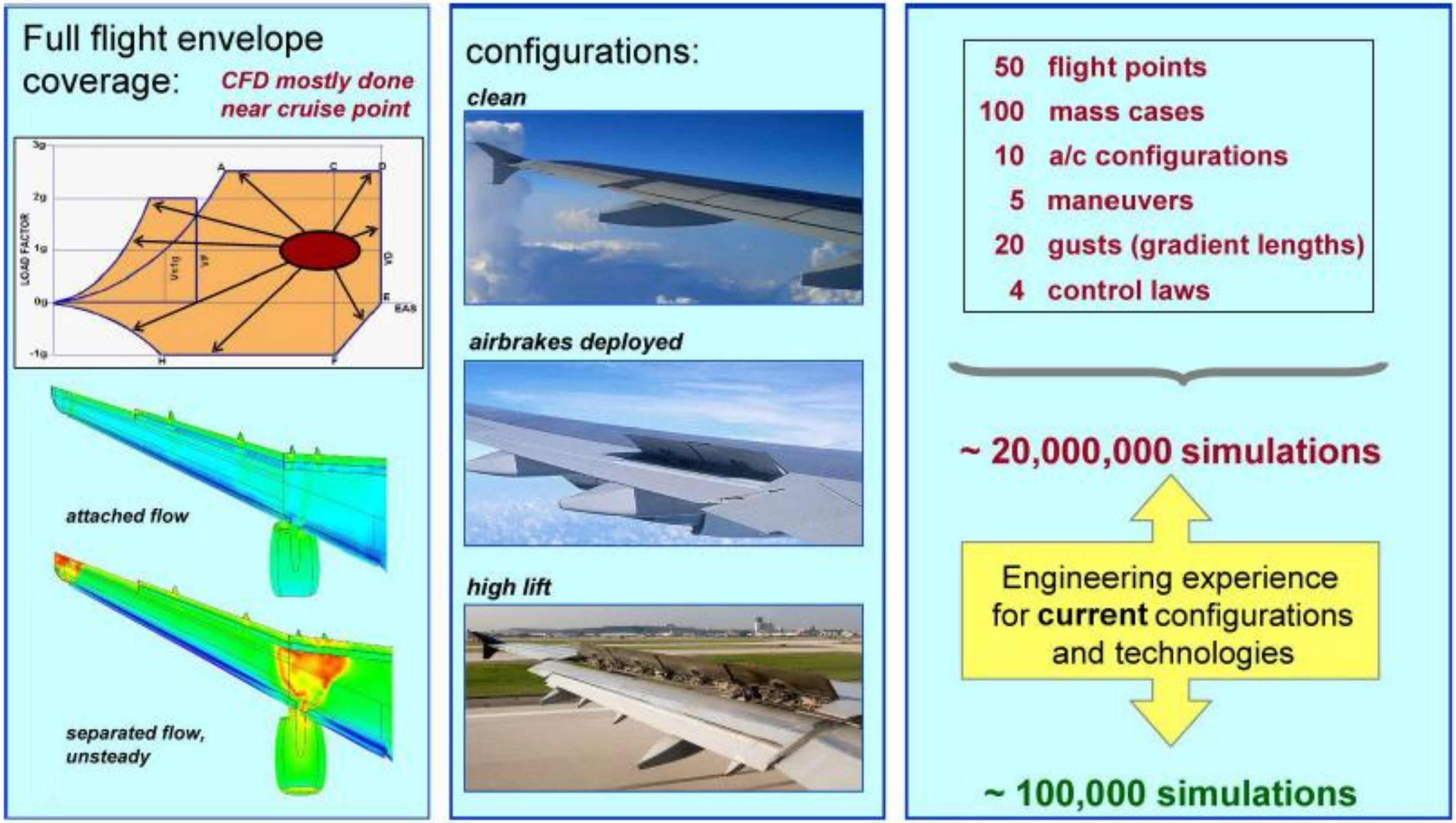
Cloud Computing mit mathematischen Anwendungen

Dr. habil. Marcel Kunze
Engineering Mathematics and Computing Lab (EMCL)
Institut für Angewandte und Numerische Mathematik IV
Karlsruhe Institute of Technology (KIT)

11. Numerical Simulation and Optimization

Vision: Digital Aircraft



Source: DLR

- Example: Airbus 380

How can we scale to this Dimension?

- **Extension of the Parallel Programming Model**
  - Fully asynchronous
  - Dataflow driven
  - Implicitly load balanced
  - Exclusively uses local locks instead of global barriers
  - Should work well with large multi-core CPUs

```
for(k = 0; k < KDIM; k++)
   for (j = 0; j < JDIM; j++)
      for (i = 0; i < IDIM; i++) {};
BARRIER;
for(k = 0; k < KDIM; k++) …
```

```
for(k = get_k_index(); k != -1; k = get_next_k_index(k))
   for (j = 0; j < JDIM; j++)
      for (i = 0; i < IDIM; i++) {};
 for(k = get_k_index(); k != -1; k = get_next_k_index(k)) …
```

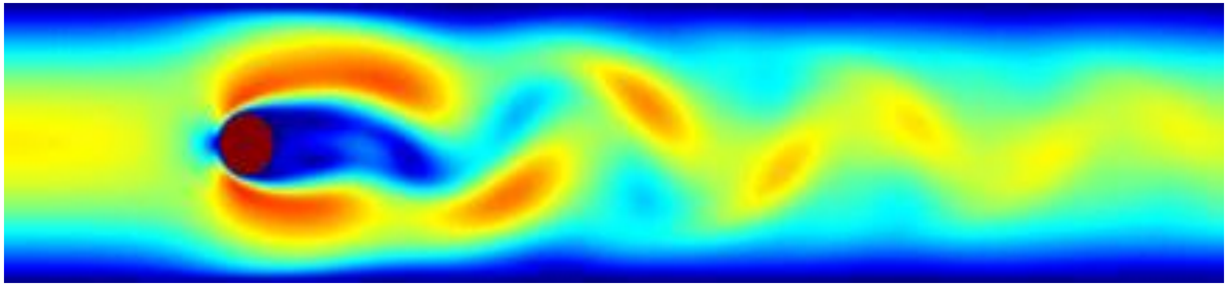
J.Jägersküpper (DLR), C.Simmendinger (T-Systems SfR). A Novel Shared-Memory Thread-Pool Implementation for Hybrid Parallel CFD Solvers, EuroPar 2011, (to appear).

Computational Fluid Dynamics (CFD)

- Predicting the behavior of fluids
- Important in e.g. aerospace-, civil-, or mechanical engineering
- Numerical simulation of fluid flows
  - Traditional method: solves the conservation equations of macroscopic properties (i. e., mass, momentum, and energy)
  - Lattice Boltzmann method (LBM): models the fluid consisting of fictitious particles, which perform consecutive propagation and collision processes over a discrete lattice
  - LBM is efficient when dealing with complex boundaries and the incorporation of microscopic interactions

What are Lattice Boltzmann Methods (LBM)?

- Lattice Boltzmann methods are numerical techniques for the simulation of fluid flows.
- Solve the incompressible, time-dependent **Navier-Stokes** equation numerically
- Represent **complex physical phenomena**, ranging from multiphase flows to chemical interactions between the fluid and the borders
- Molecular description of a fluid incorporating physical terms
- A collection of resources on the lattice Boltzmann method can be found on the website of **LBmethod**.



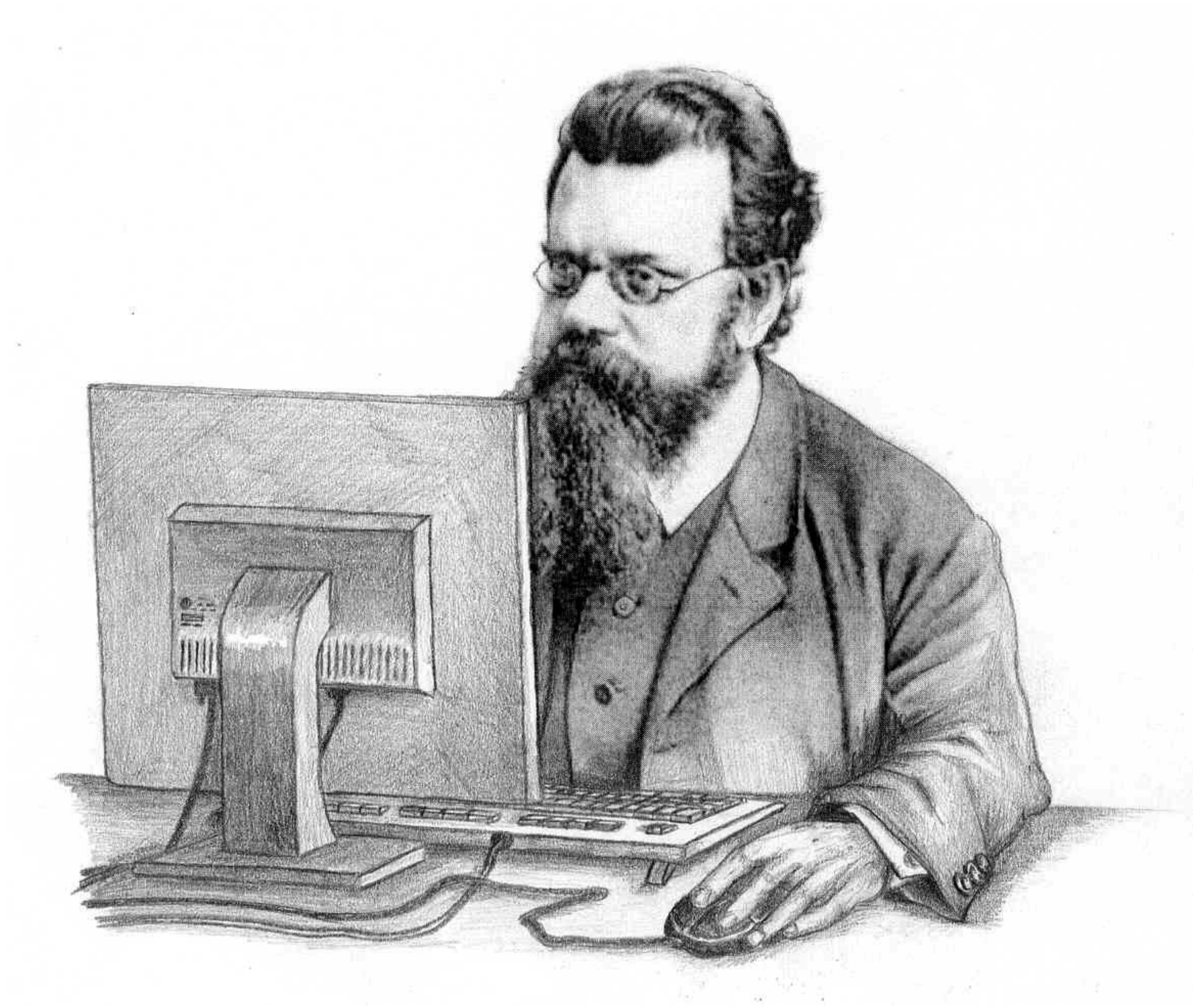
http://www.lbmethod.org/animations/karman.gif

2D flow past a cylinder
Source:
http://www.lbmethod.org

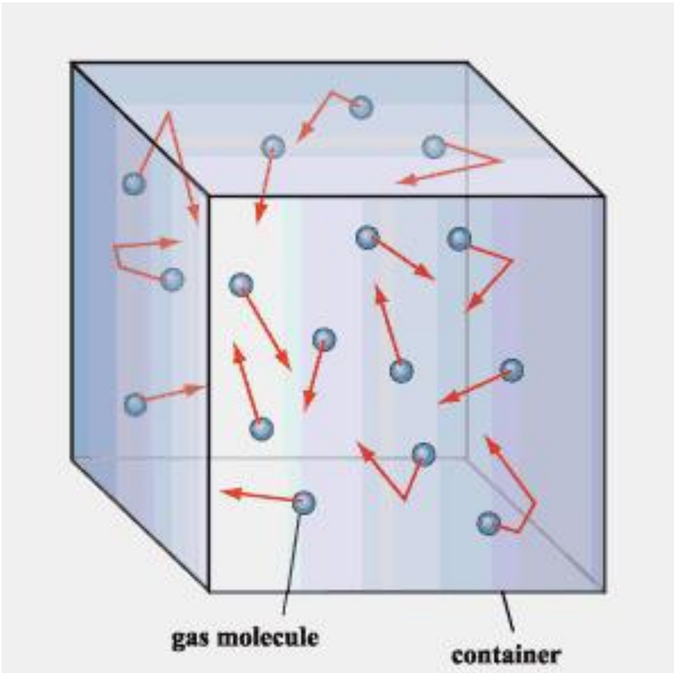
Where does LBM come from?

- Discrete representations of the Boltzmann equation (Kinetic gas theory)
- **Boltzmann equation** is the analogue of the **Navier-Stokes equation** at a molecular level, where it describes the evolution of the **probability distribution function** for a molecule to be present at a given point in the space of positions and velocities, the 6-dimensional phase space.
- The molecular model is able to capture transport phenomena such as friction, diffusion and temperature transport and derive the corresponding transport coefficients.

Ludwig Boltzmann (1844-1906)



Kinetic Gas Theory (Boltzmann Equation)



- Free elastic particles
- No external forces
- Constant velocity
- Large number (ca. 10^{20})

Boltzmann equation

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \frac{\partial f}{\partial \mathbf{x}} = \Omega(f, f')$$

Particle probability density function **Ω**

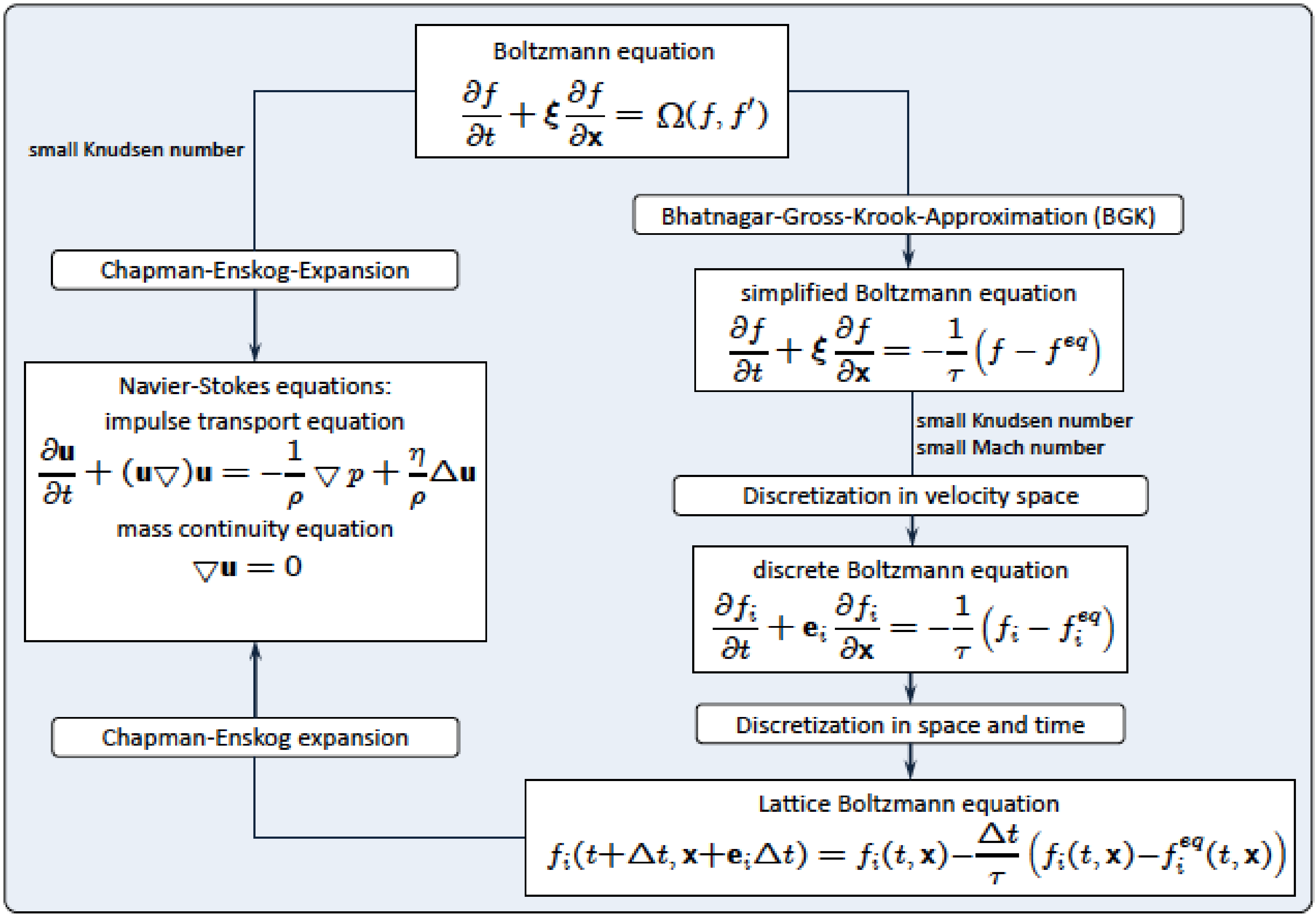
f = Mass density

$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$: Position

$\xi = (\xi_1, \xi_2, \xi_3)$: Velocity

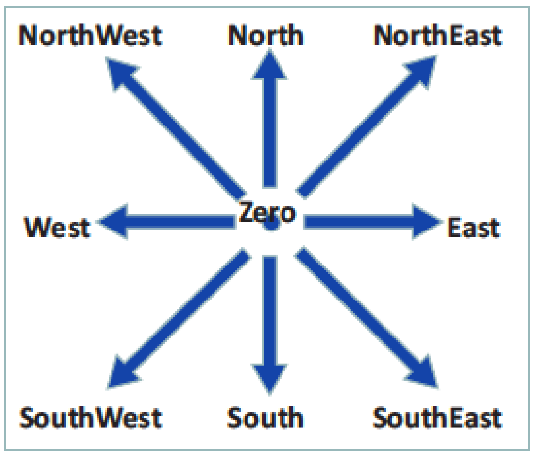
t : Time

Lattice Boltzmann Equation

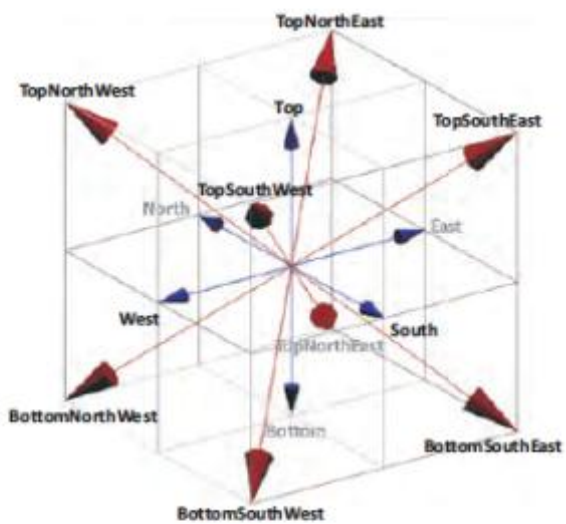


Discretization

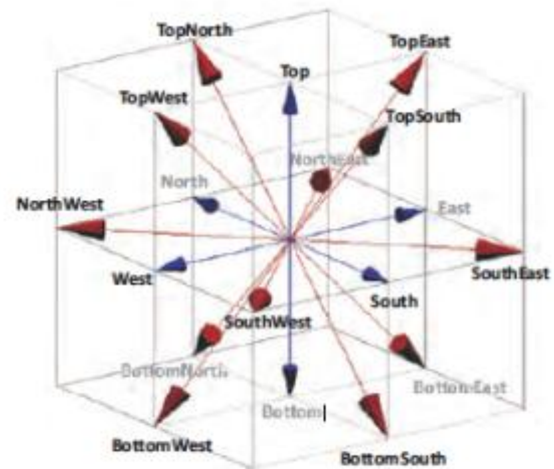
- Discretization on a lattice with stencils of dimension d and number of microscopic velocities q: DdQq



D2Q9



D3Q15

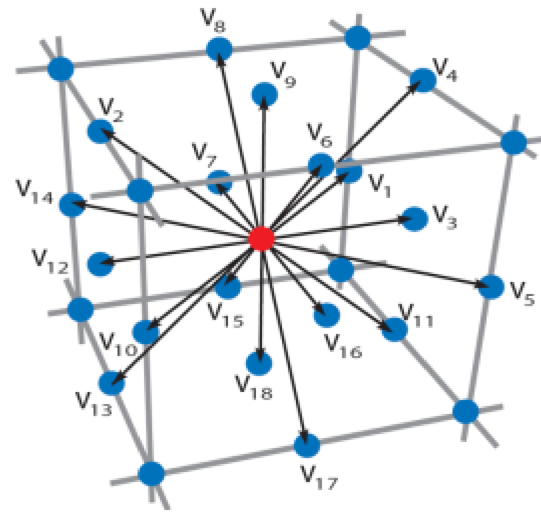


D3Q19

Lattice Boltzmann Methods (LBM)

KIT
Karlsruhe Institute of Technology

D3Q19



Idea: coupling model parameter $h \in \mathbb{R}_{>0}$ with discretisation parameter: *Lattice DdQq*

Macroscopic moments:

density $\rho = \sum_{i=0}^{q-1} f_i$, velocity $\rho \boldsymbol{u} = \sum_{i=0}^{q-1} \boldsymbol{v_i} f_i$

Time loop $t = t_0, t_0 + h^2, t_0 + 2h^2, ..., t_1$

Position space loop $\boldsymbol{r} \in \Omega_h$

(1) Collision $\tilde{f}_i(t, \boldsymbol{r}) = f_i(t, \boldsymbol{r}) - \frac{1}{3\nu + 1/2}\left(f_i(t, \boldsymbol{r}) - M_{f_i}^{eq}(t, \boldsymbol{r})\right)$

(2) Streaming $f_i(t + h^2, \boldsymbol{r} + h^2 \boldsymbol{v_i}) = \tilde{f}_i(t, \boldsymbol{r})$

M.Krause, EMCL

OpenLB: Overview

openlb.org

2D and 3D fluid flow simulations based on LBM

- Computational framework:
  - Open Source (GPL2)
  - C++, object oriented, template-based
- Features in latest release 0.6:
  - Multi-physics models (multiphase, thermal)
  - Hybrid parallelisation (MPI & OpenMP)
  - Automated build-in pre-processing:
    - Voxelisation from STL-files
    - Setting of boundary conditions
  - XML interface for input parameters
  - Visualization (built-in and VTK)
- Current project partners:

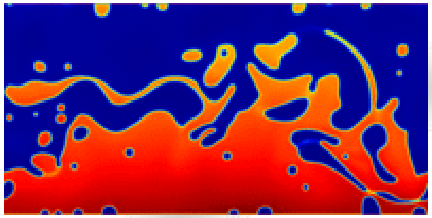
  EMCL/KIT (Karlsruhe), Mevis (Bremen)

2009

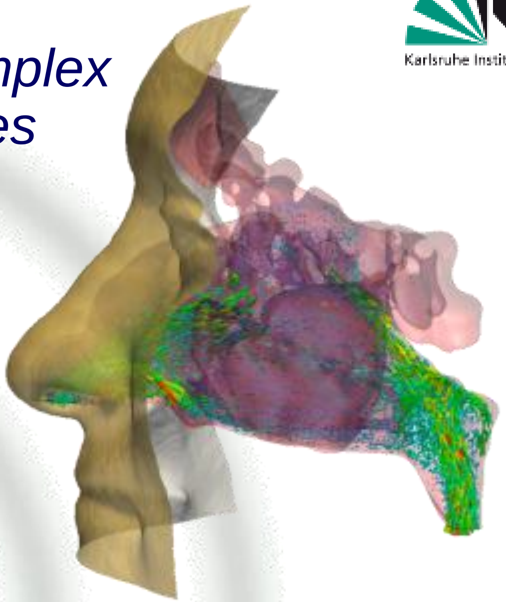
2007

M.Krause, EMCL

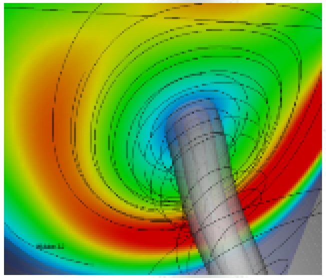
OpenLB: Applications



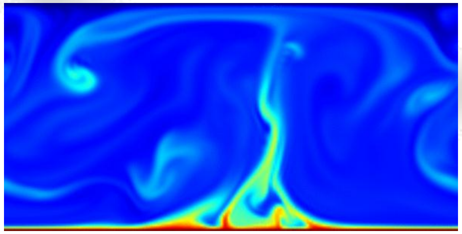
Multiphase flows



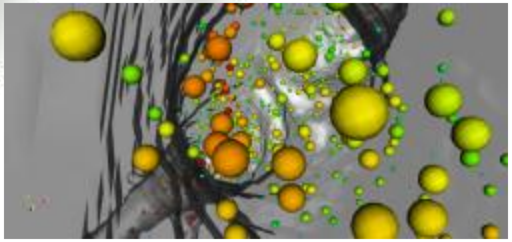
Flows in complex geometries



Turbulent flows



Thermal flows



Particle flows

M.Krause, EMCL

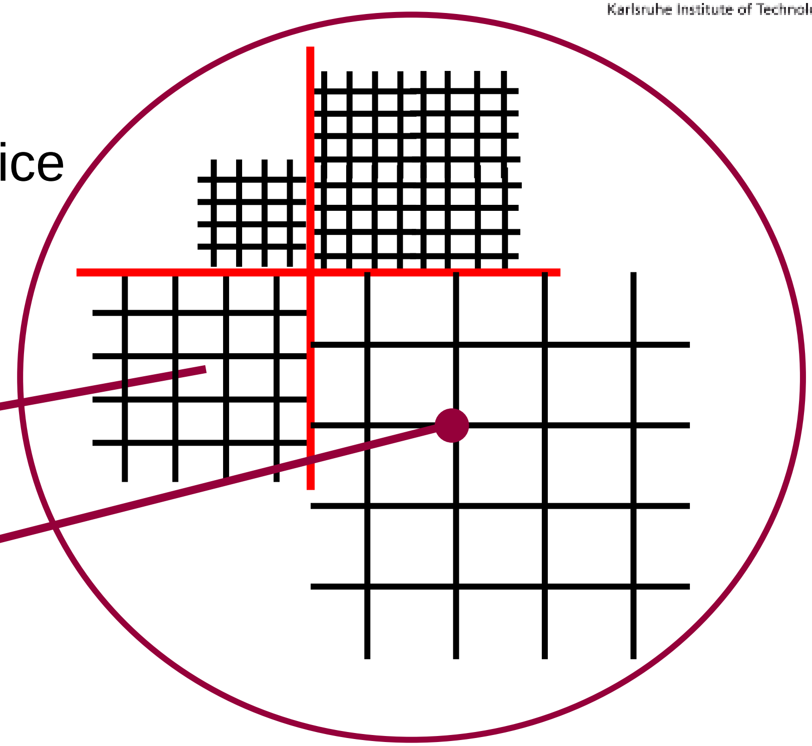
Data Structure Design

SuperLattice

Multi-block ansatz

BlockLattice

Cell



Data:

- Distribution function f_i

Methods:

- Compute density
- Compute velocity
- Compute stress tensor
- Execute collision step

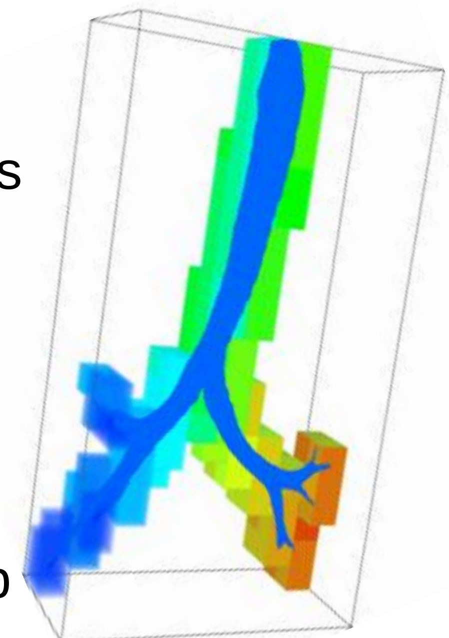
Example: distinction between fluid cell and boundary cell

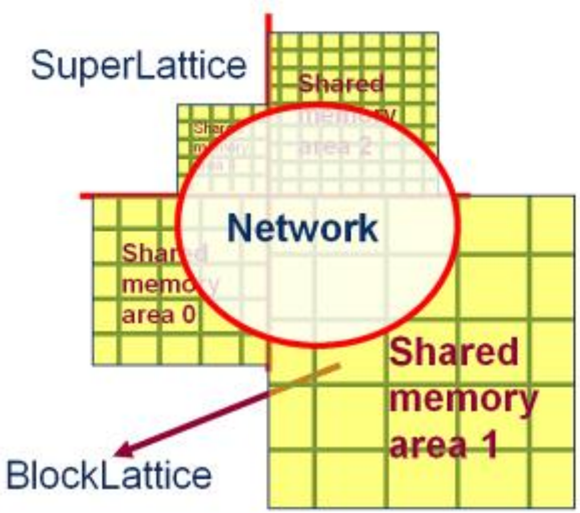
Example: inhomogeneous fluids

M.Krause, EMCL

Parallelisation: Hybrid Concept

- For distributed and/or shared memory platforms
- Based on domain decomposition
- Sparse multi-block representation of geometries
- OpenMP for intra-block parallelism
- MPI for inter-block parallelism





Time loop

Collision step

> Blocking

> Communicating

> Writing to ghost cells

Streaming step

M.Krause, EMCL

Hausaufgabe 10: OpenLB

Installieren Sie das OpenLB-System entsprechend der Anleitung auf http://www.openlb.org/. Instanzieren Sie mit AWS zunächst ein Amazon Linux System (m1.large), basierend auf dem Image ami-8e1fece7.

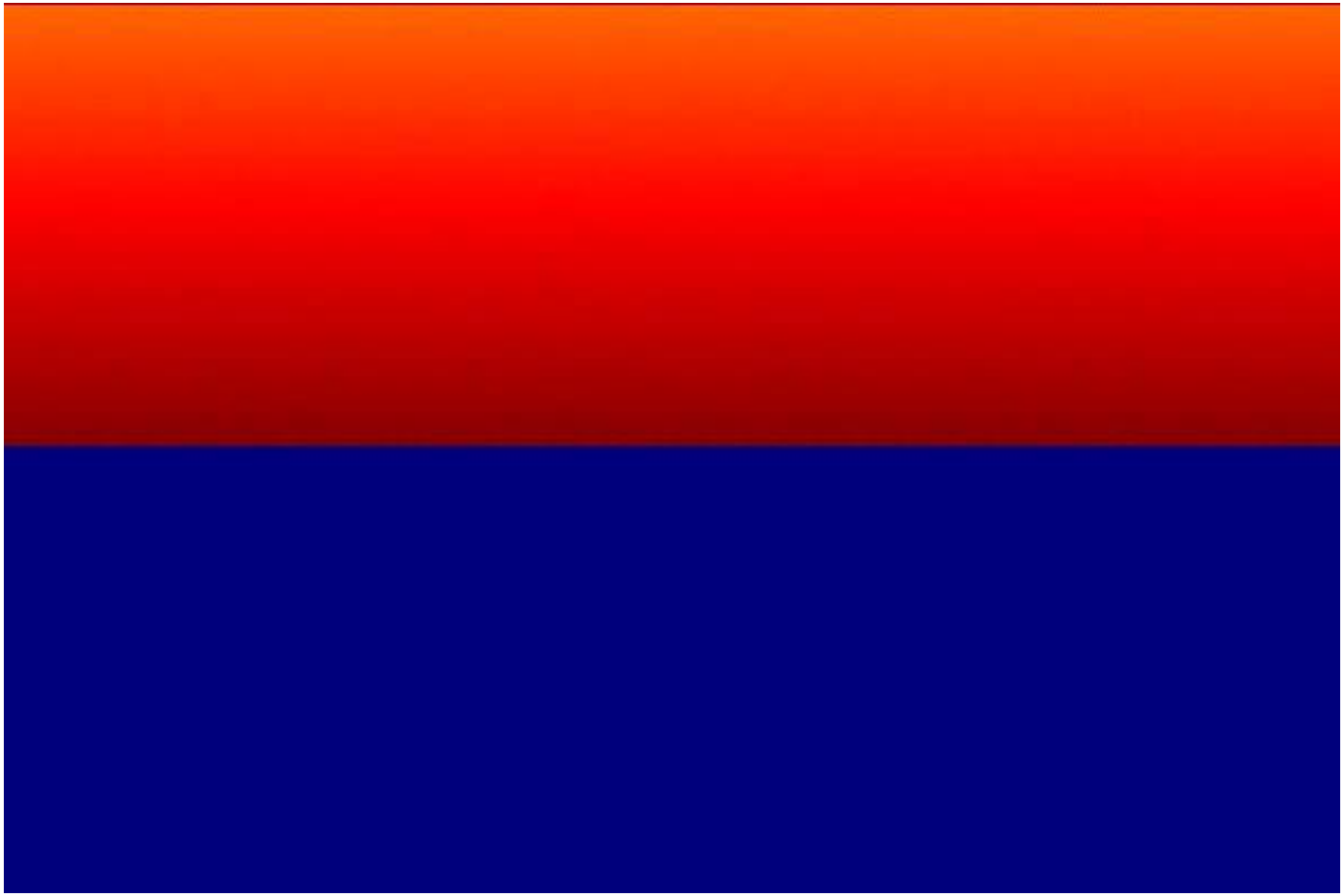
- Einloggen auf dem Server mit
  ssh –i <keyname>.pem ec2-user@<server-ip>
- Installation der Requirements
  sudo yum install mpi-devel;
  sudo yum install blas-devel;
  sudo yum install gcc-c++;
  sudo yum install make
- Laden des OpenLB-Pakets
  wget http://www.openlb.org/openlb/download.php?cat=30_Download\&file=olb-0.6r0.tgz
- Auspacken des Pakets
  tar -zxf *olb-0.6r0.tgz; cd olb-0.6r0
- In Makefile.inc die folgenden Optionen aktivieren:
  CXX := mpicc
  PARALLEL_MODE := MPI
- Übersetzen des Pakets
  export MPI_HOME=/usr/lib64/openmpi
  export PATH=$MPI_HOME/bin:$PATH
  export LD_LIBRARY_PATH=$MPI_HOME/lib:$LD_LIBRARY_PATH
  make

Hausaufgabe 10: OpenLB

Probieren Sie nun einige der Beispiele aus und stellen Sie die Ergebnisse grafisch dar.

- Installation des Grafikpakets und des Filmkonverters
  sudo yum install imageMagick
  sudo yum install ffmpeg
- Wechsel in das Beispielverzeichnis. Durchführen einer Simulation. Erzeugen eines Films aus den einzelnen gif-Bildern.
  cd examples
  cd cavity2d; make
  mpirun –np 8 cavity2d
  cd tmp
  convert –delay 5 *.gif cavity2d.mpg
- Die vti-Files können auch mit dem Programm Paraview studiert werden:http://www.paraview.org/

Example



- Rayleigh-Taylor instability in 2D, generated by a heavy fluid penetrating a light one. http://cloudvorlesung.s3.amazonaws.com/multiComponent2d.mpg

Good to know

- **All course info is on http://studium.kit.edu**
- **http://www.math.kit.edu/mitglieder/lehre/cloudcomp2011s/**


©johnlund.com