# Large Scale Simulations using lattice Boltzmann methods

Jonas Tölke
Institut für Computeranwendungen im Bauingenieurwesen
TU Braunschweig

# Outline

- Lattice Boltzmann method
- Parallelization
- Different Examples
- GPU Programming

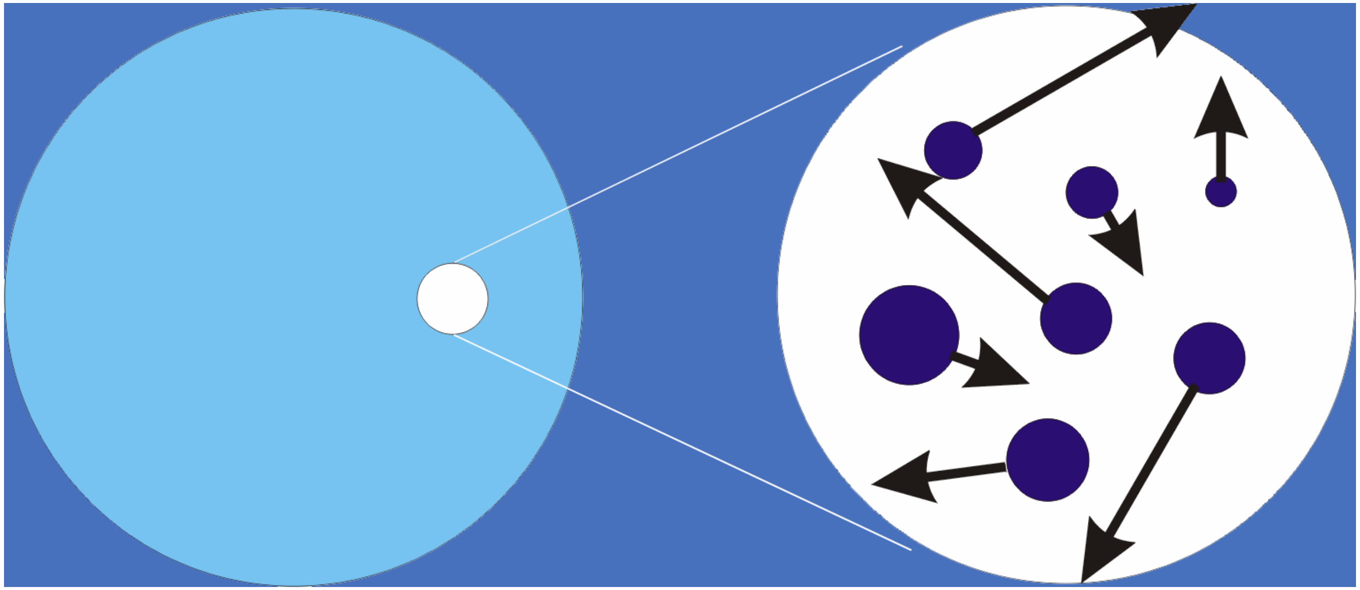
## Computational Fluids

Fluid (Continuum)

$$\mathrm{p}, \nu, \vec{\mathrm{u}}(\vec{\mathrm{r}}, \mathrm{t}), \rho$$

Particle Ensemble

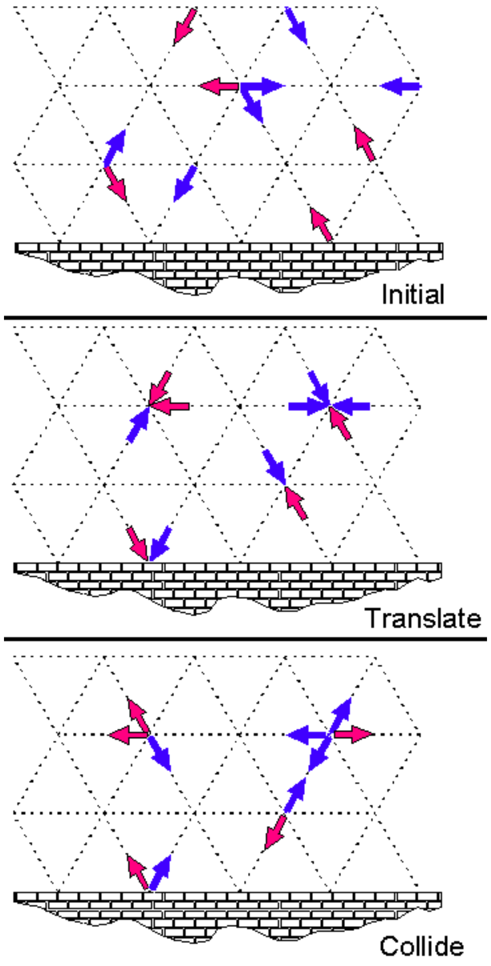
$$\mathrm{m_i}, \tau, \vec{\mathrm{v}}_\mathrm{i}(\vec{\mathrm{r}}, \mathrm{t})$$



$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\nabla \otimes \vec{u} = -\nabla p^* + \nu \Delta \vec{u}$$

$$m_i \frac{\partial v_i}{\partial t} = \sum_{i \neq j} \frac{\partial V(\|\vec{r}_i - \vec{r}_j\|)}{\partial \vec{r}_j}$$

# Minimal kinetic model: Lattice-Boltzmann Automata




*Wolfram 86, Hasslacher 86*

$$f_i(t+\Delta t, \mathbf{x}+\mathbf{e}_i\Delta t) = f_i(t,\mathbf{x}) + \quad {}_i, \qquad i = 0,\ldots,b-1$$

$$\square = \mathrm{M}^{-1}\,\boldsymbol{k}$$

- *mass fractions f*
- *regular grid*
- *propagation*
- *collision*

# Collision operator

transformation into moment space:

$$m = \mathsf{M}f := (\rho, \rho u_x, \rho u_y, e, p_{xx}, p_{xy}, h_x, h_y, \epsilon)$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & c & 0 & -c & 0 & c & -c & -c & c \\ 0 & 0 & c & 0 & -c & c & c & -c & -c \\ -2\,c^2 & c^2 & c^2 & c^2 & c^2 & 4\,c^2 & 4\,c^2 & 4\,c^2 & 4\,c^2 \\ 0 & c^2 & -c^2 & c^2 & -c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c^2 & -c^2 & c^2 & -c^2 \\ 0 & -c^3 & 0 & c^3 & 0 & 2\,c^3 & -2\,c^3 & -2\,c^3 & 2\,c^3 \\ 0 & 0 & -c^3 & 0 & c^3 & 2\,c^3 & 2\,c^3 & -2\,c^3 & -2\,c^3 \\ c^4 & -2\,c^4 & -2\,c^4 & -2\,c^4 & -2\,c^4 & 4\,c^4 & 4\,c^4 & 4\,c^4 & 4\,c^4 \end{bmatrix}$$

Cascade: *Geier 06*

$$k_0 = k_1 = k_2 = 0$$
$$k_e = s_e \left(e - 3\,\rho\,(u_x^2 + u_y^2)\right)$$
$$k_{xx} = s_\nu\,(p_{xy} - \rho\,u_x\,u_y)$$
$$k_{xy} = s_\nu \left(p_{xx} - \rho\,(u_x^2 - u_y^2)\right)$$
$$k_{h_x} = s_h \left(h_x - 6 u_y k_{xy} \left(\frac{1}{s_\nu} - \frac{1}{s_h}\right)\right) + s_h \left(-u_x \left(\frac{1}{2}\left(e - \frac{k_e}{s_h}\right) - \frac{3}{2}\left(p_{xx} - \frac{k_{xx}}{s_h}\right)\right)\right)$$
$$k_{h_y} = s_h \left(h_y - 6.0 u_x k_{xy} \left(\frac{1}{s_\nu} - \frac{1}{s_h}\right)\right) + s_h \left(-u_y \left(\frac{1}{2}\left(e - \frac{k_e}{s_h}\right) + \frac{3}{2}\left(p_{xx} - \frac{k_{xx}}{s_h}\right)\right)\right)$$
$$k_\epsilon = s_\epsilon \left(\epsilon - 27 u_x^2 u_y^2 + k_e \left(\frac{1}{s_e} - \frac{1}{s_\epsilon}\right) + \frac{3}{2}(u_x^2 + u_y^2)\left(e - \frac{k_e}{s_\epsilon}\right)\right) + s_\epsilon \left(-\frac{9}{2}(u_x^2 - u_y^2)(p_{xx} - \frac{k_{xx}}{s_\epsilon}) + 36 u_x u_y (p_{xy} - \frac{k_{xy}}{s_\epsilon})\right) + s_\epsilon \left(-6 u_x (h_x - \frac{k_{h_x}}{s_\epsilon}) - 6 u_y (h_y - \frac{k_{h_y}}{s_\epsilon})\right)$$

MRT: *Humieres 92, Lallemand 00*

$$k_0 = k_1 = k_2 = 0$$
$$k_3 = k_e = s_e \left(e - 3\,\rho\,(u_x^2 + u_y^2)\right)$$
$$k_4 = k_{xx} = s_\nu \left(p_{xx} - \rho\,(u_x^2 - u_y^2)\right)$$
$$k_5 = k_{xy} = s_\nu\,(p_{xy} - \rho\,u_x\,u_y)$$
$$k_6 = k_{hx} = s_h\,h_x$$
$$k_7 = k_{hy} = s_h\,h_y$$
$$k_8 = k_\epsilon = s_\epsilon \epsilon,$$

$s_\nu, s_e, s_h, s_\epsilon$: Relaxation rates

# Multi- scale analysis with computeralgebra

- grid
- collision-operator
- momenten
- Taylor expansion
- asymptotic expansion
- sort by order

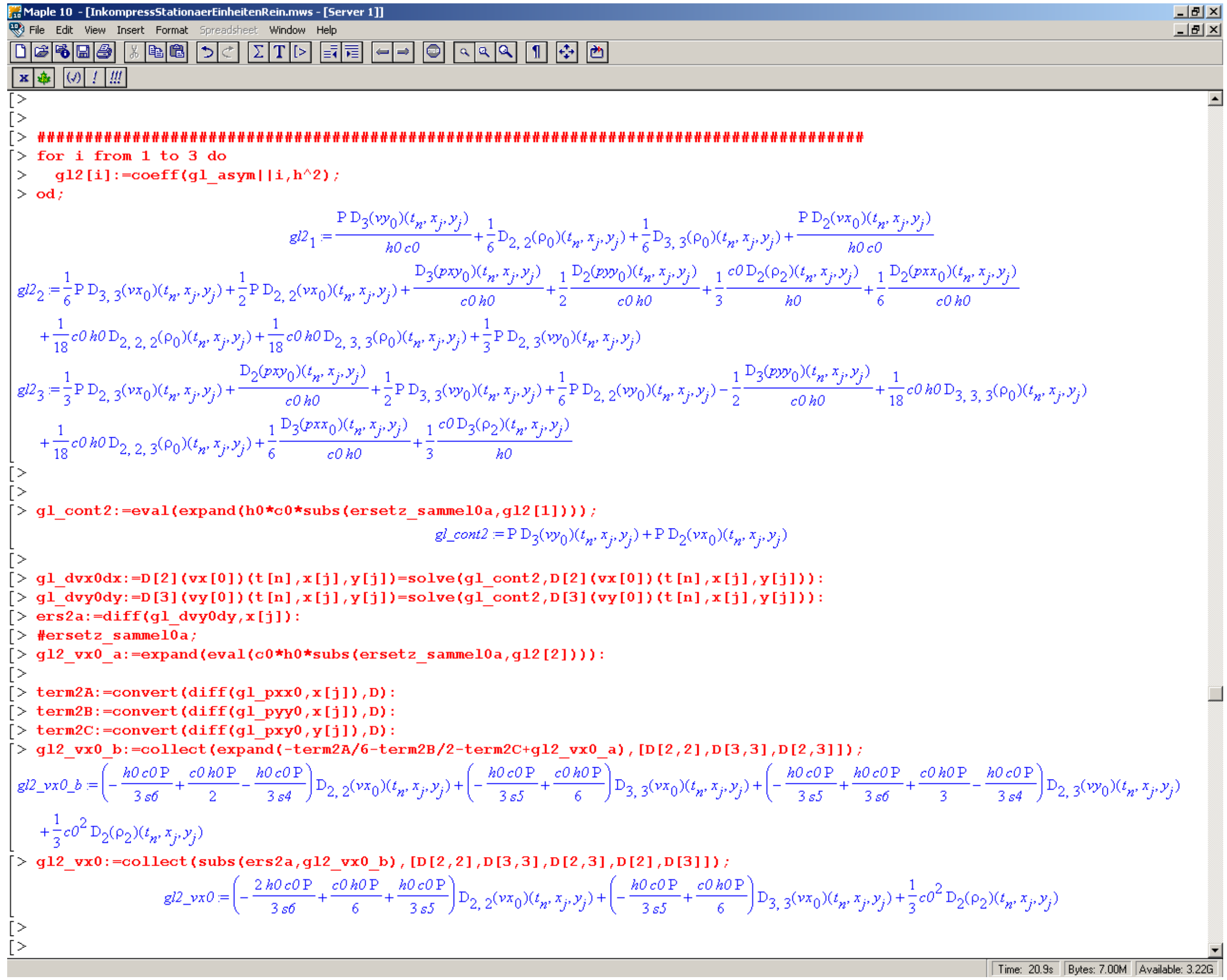
$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\nabla \otimes \vec{u} + \nabla p = \nu \Delta \vec{u}$$

viscosity:

$$\nu = c^2 \Delta t \left(\frac{1}{3 s_\nu} - \frac{1}{6}\right)$$

pressure:

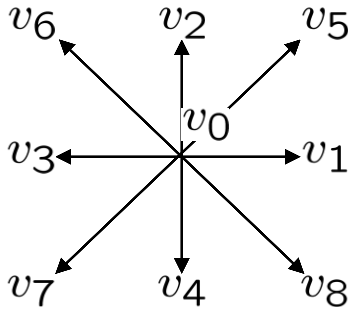
$$p = \frac{c^2}{3}\rho$$



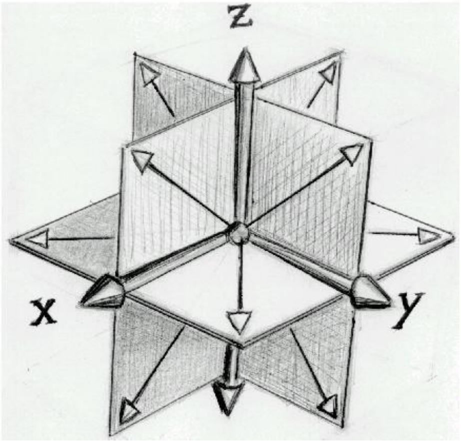
# Transport phenomena with Lattice-Boltzmann Automata

- Navier-Stokes (d2q6, d2q7, d2q9, d3q13, d3q15, d3q19, d3q27)
- advection-diffusion (d2q5,d2q9,d3q7,...)
- suitable for transport equation
- applications:
  - fluid mechanics (large range of Reynolds numbers)
  - aero-acoustic
  - traffic flow
  - …..

d2q9-Model




d3q19-Model

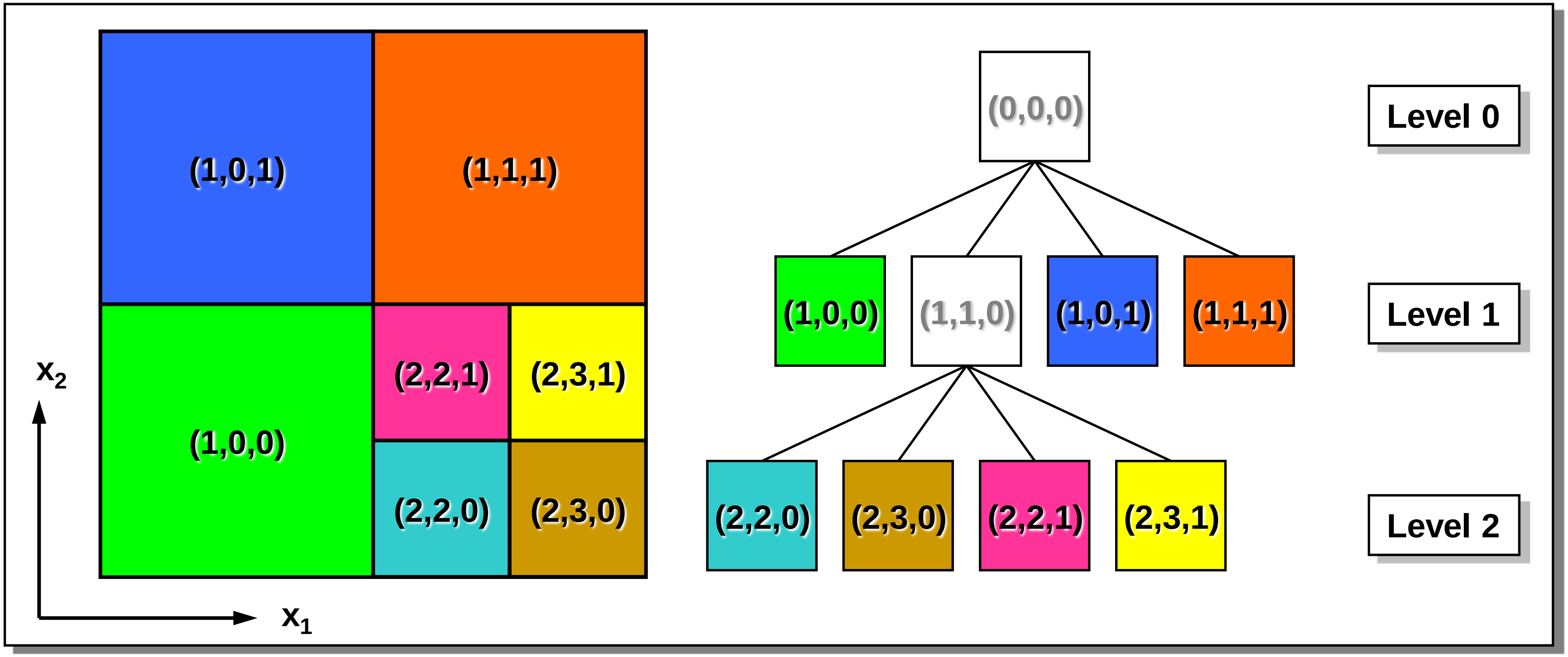


# Computational aspects

- basic cells: squares and cubes
- coupled space and time resolution
- convergence properties:
  LBE second-order accurate with respect to the corresponding solution of incompressible Navier-Stokes flow
- because of their explicit nature and local stencil LBE models are very well suited for vectorization and parallelization
- stress tensor locally available
- weakly compressible scheme (no Poisson equation is solved for the pressure)
- no numerical viscosity
- very efficient explicit time stepping scheme (high cell Reynolds-number)
- hydrodynamic boundary conditions are introduced for distributions
- conservative scheme for mass and momentum

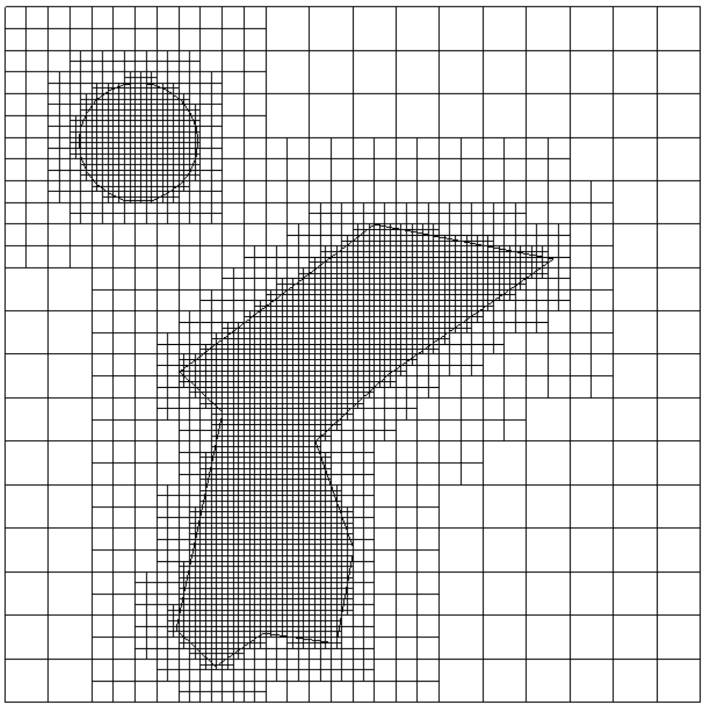
# Lattice Boltzmann Automata on tree type grids

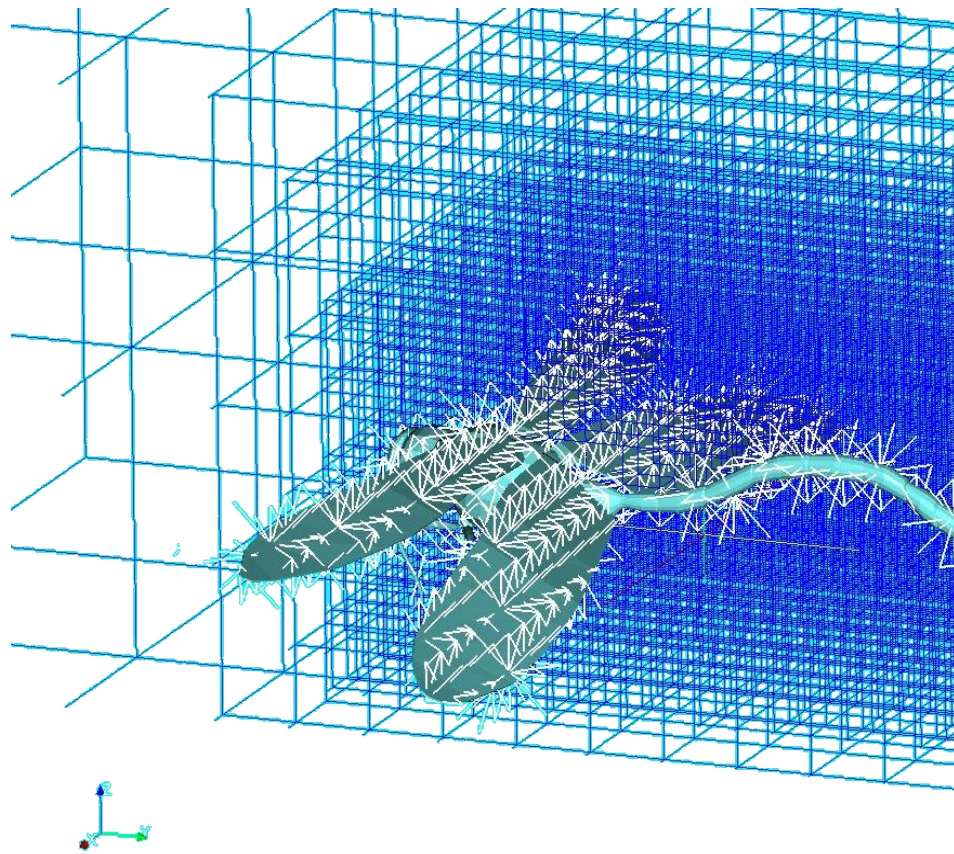
recursive bipartition of unit cell



# Tree type grids

- Quadtree in 2D
- Octrees in 3D

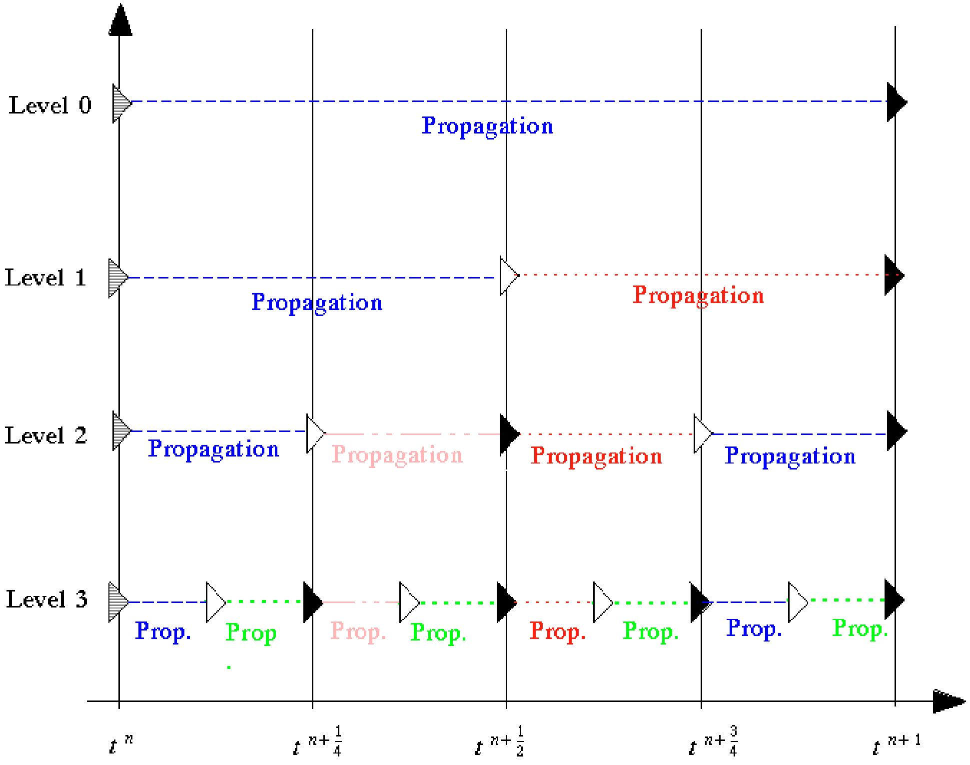


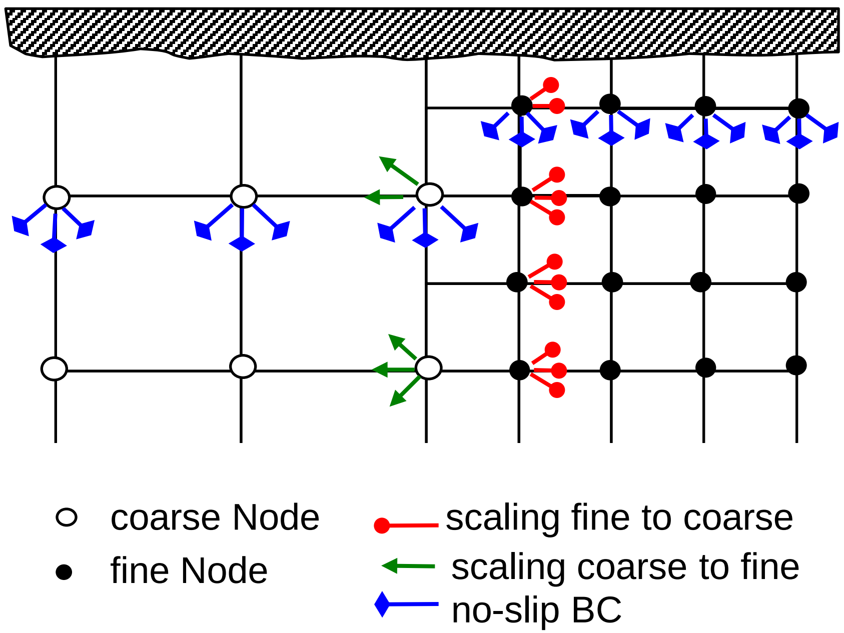


# Tree type grids

- nested time stepping
- scaling of mass fraction
- interpolation in time and space

*Filippova 98*



# Parallel computing / High performance computing

- complex geometry
- complex physics
- turbulence (Re^(11/4))
- computational steering
- real-time
- faster than real-time
- many different design cases

# Systems we use

## Aman at CAB:

| | |
|---|---|
| Processors: | 120 |
| Peak-performance: | 350 GFlop/s |
| Total size of memory : | 250 GByte |
| Direct Attached Disks: | 4000 GByte |

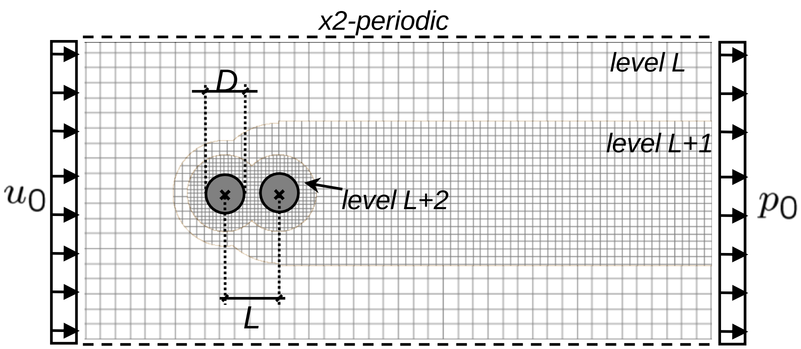
## HLRB II in Muich:

| | |
|---|---|
| Processors: | 4096 |
| Peak-performance: | 26.2 TFlop/s |
| Linpack Performance: | 24.5 TFlop/s |
| Total size of memory : | 17.5 TByte |
| Direct Attached Disks: | 300 TByte |

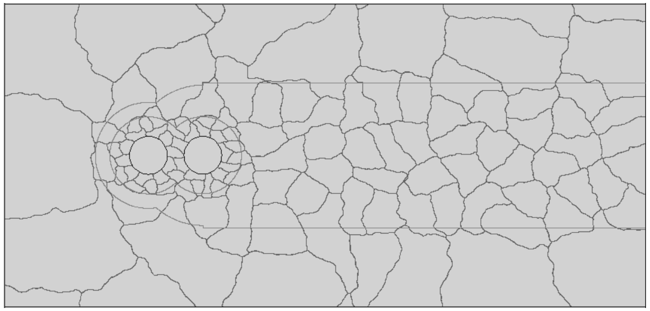
# Tree type grid parallel a priori

- space filling curves (Peano, Hilbert) for optimization of cache-access
- partition based on  graph theory (work load / communication, METIS)

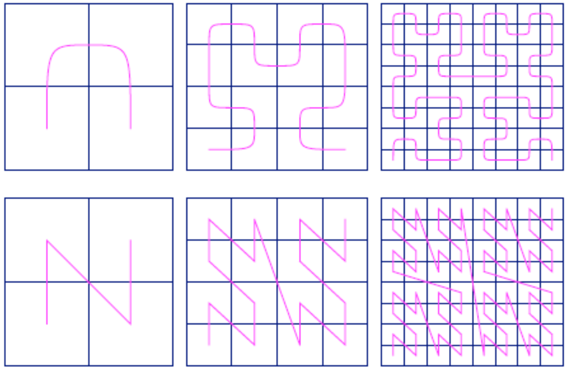
Grid:




Partition for108 Prozessors:



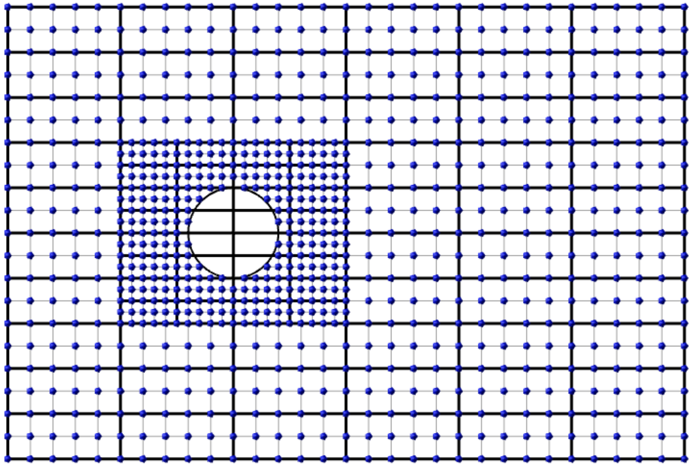
space filling curves:

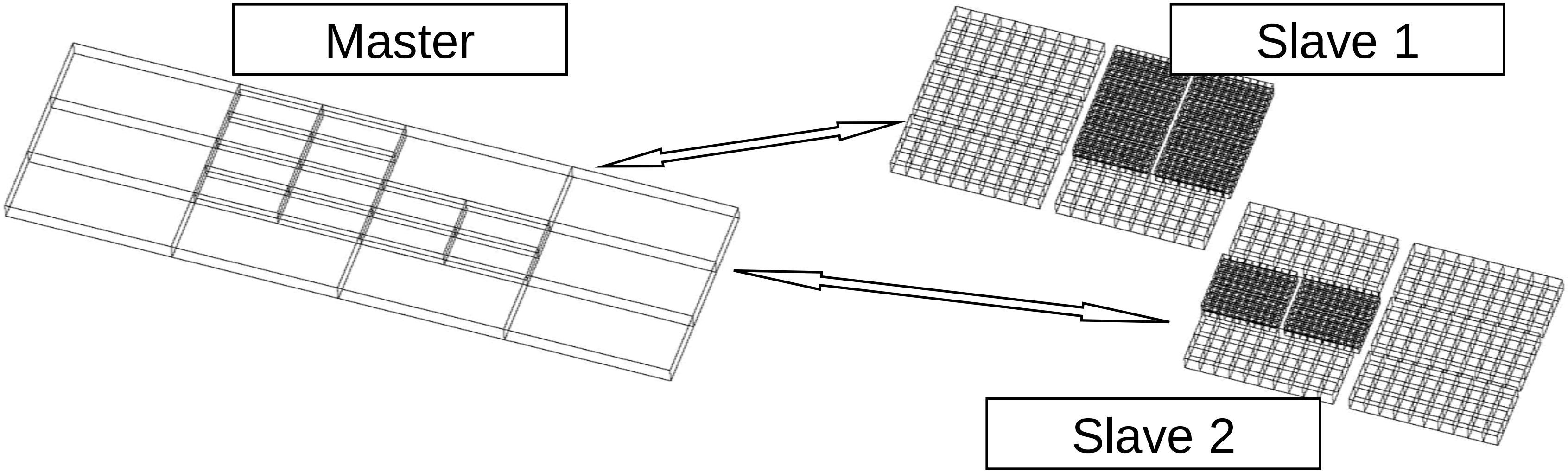


*top*: Peano-Hilbert „U“-ordering
*bottom*: Morton „N“-ordering

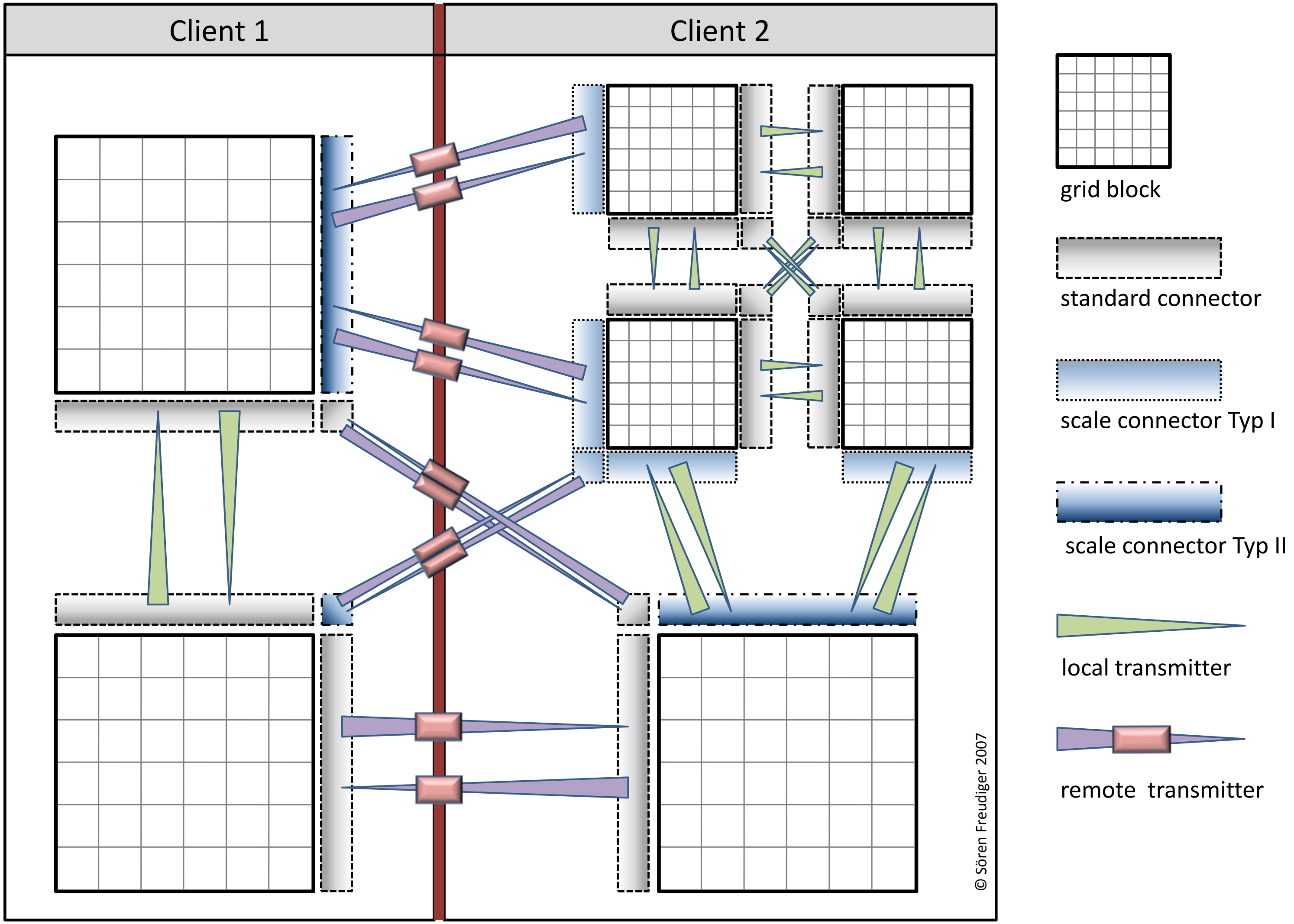
# Adaptive parallel tree type grids

- Adaptive Mesh Refinement (AMR)
- blocks on each level
- dynamic (de)allocation of blocks
- compromise performance and flexibility

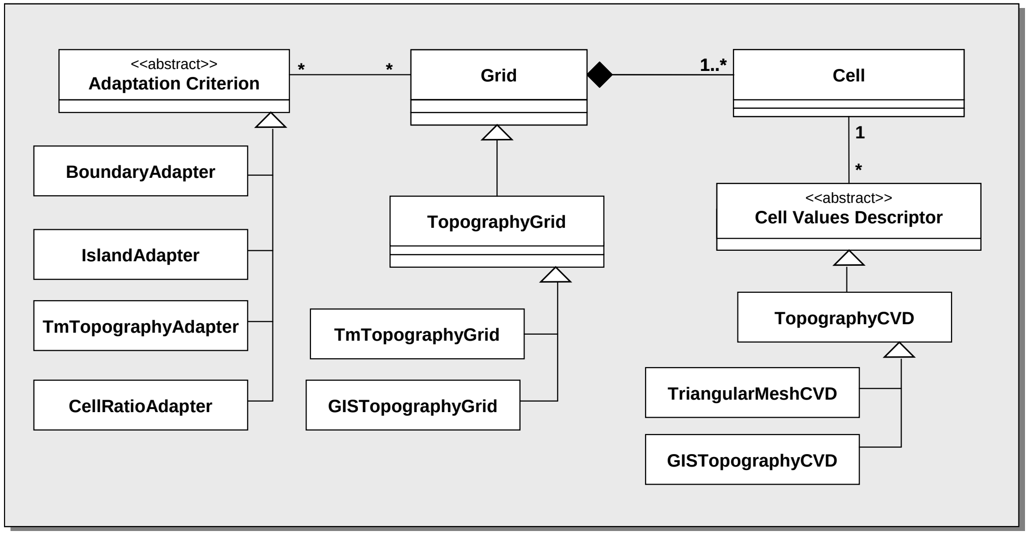


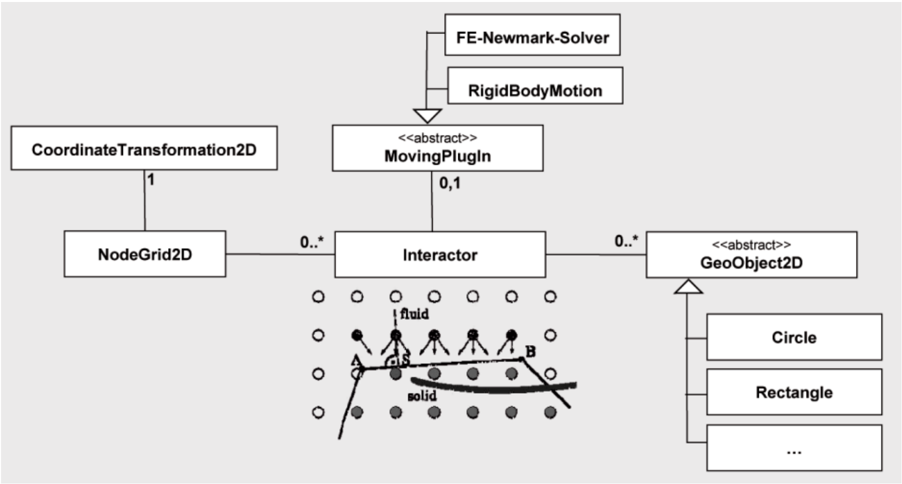


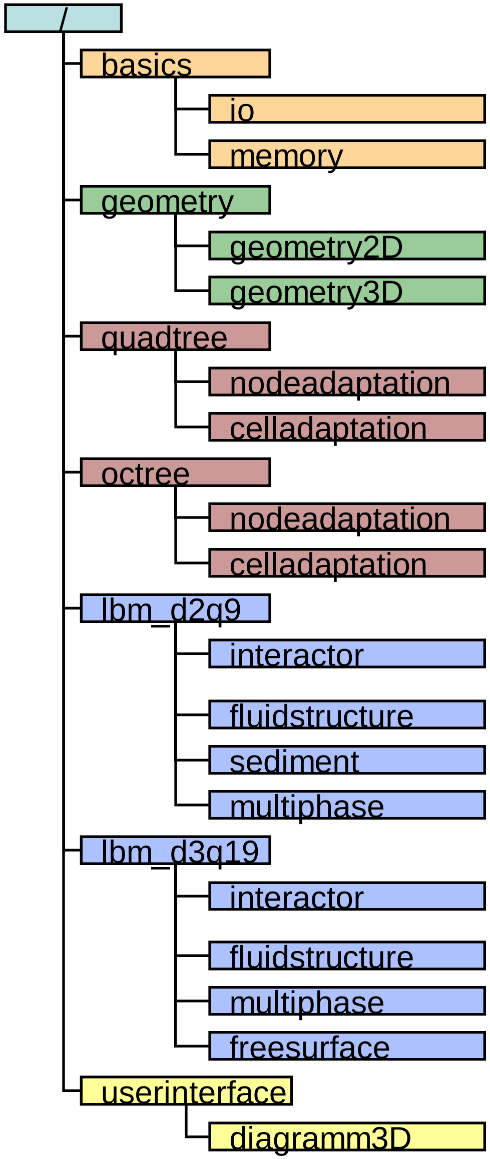
# Adaptive parallel tree type grids



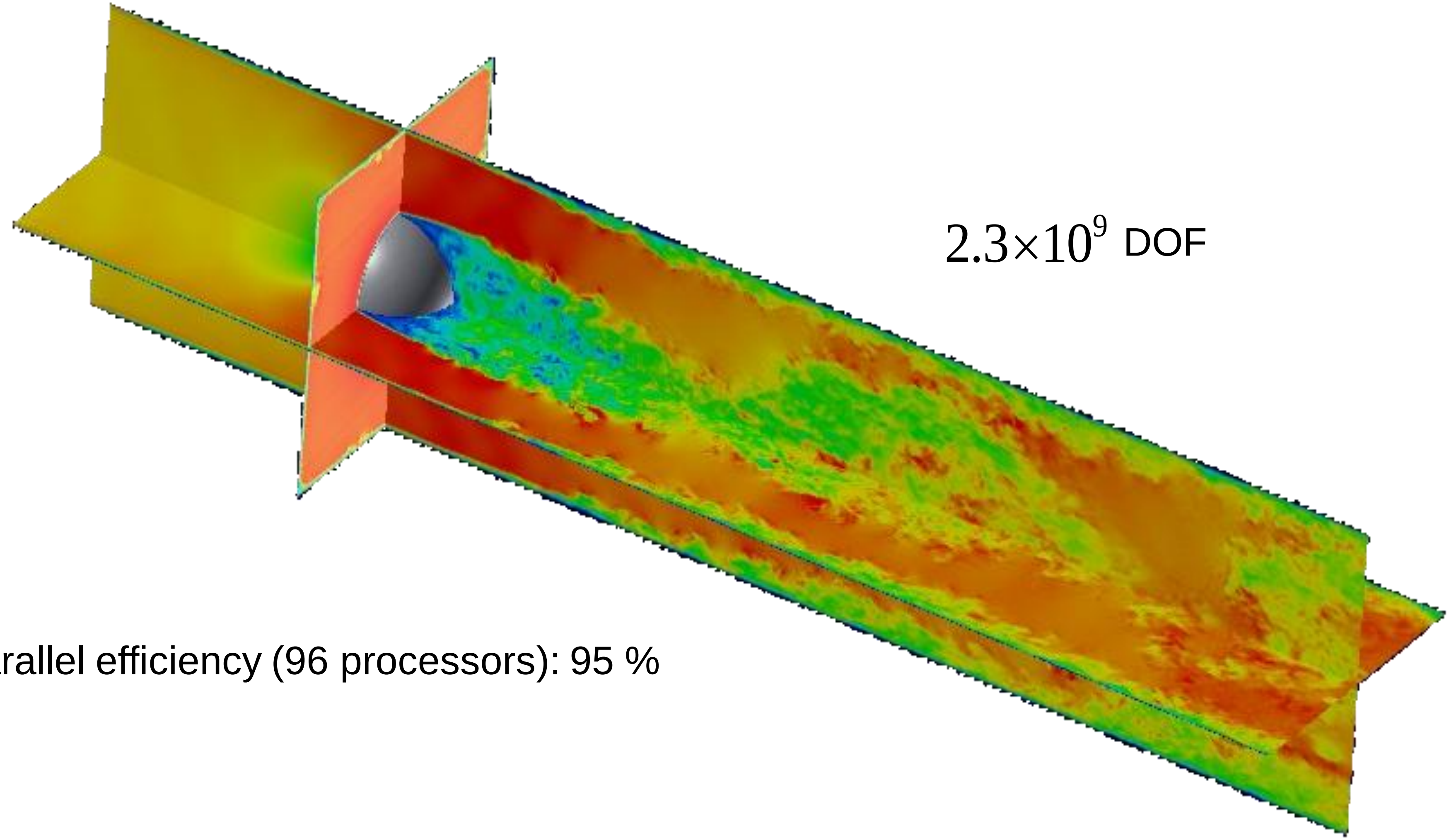
# *Virtual Fluids* / LB – Multiphysics library



# turbulent flow around a sphere

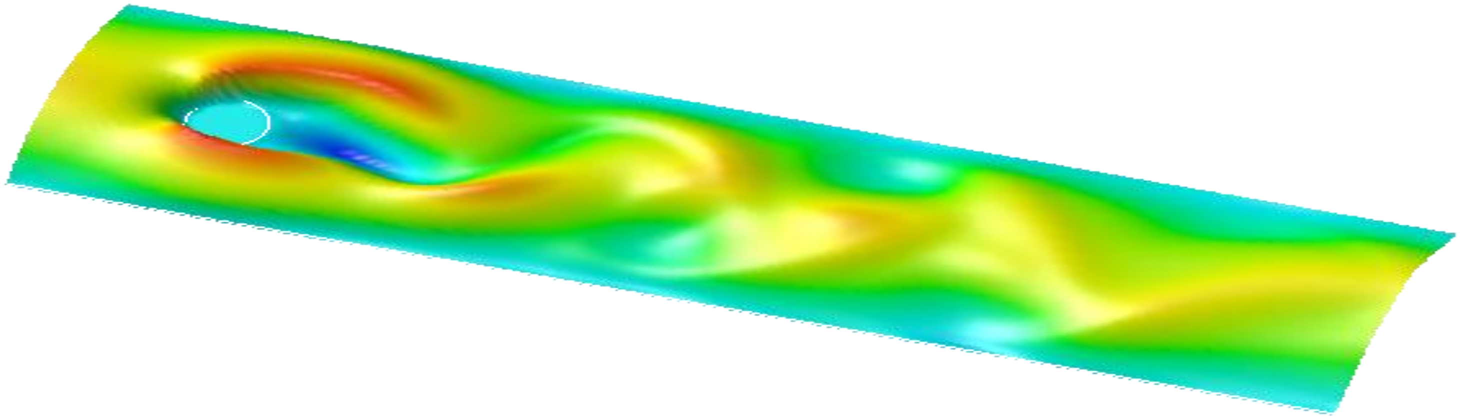


$2.3\times10^{9}$ DOF

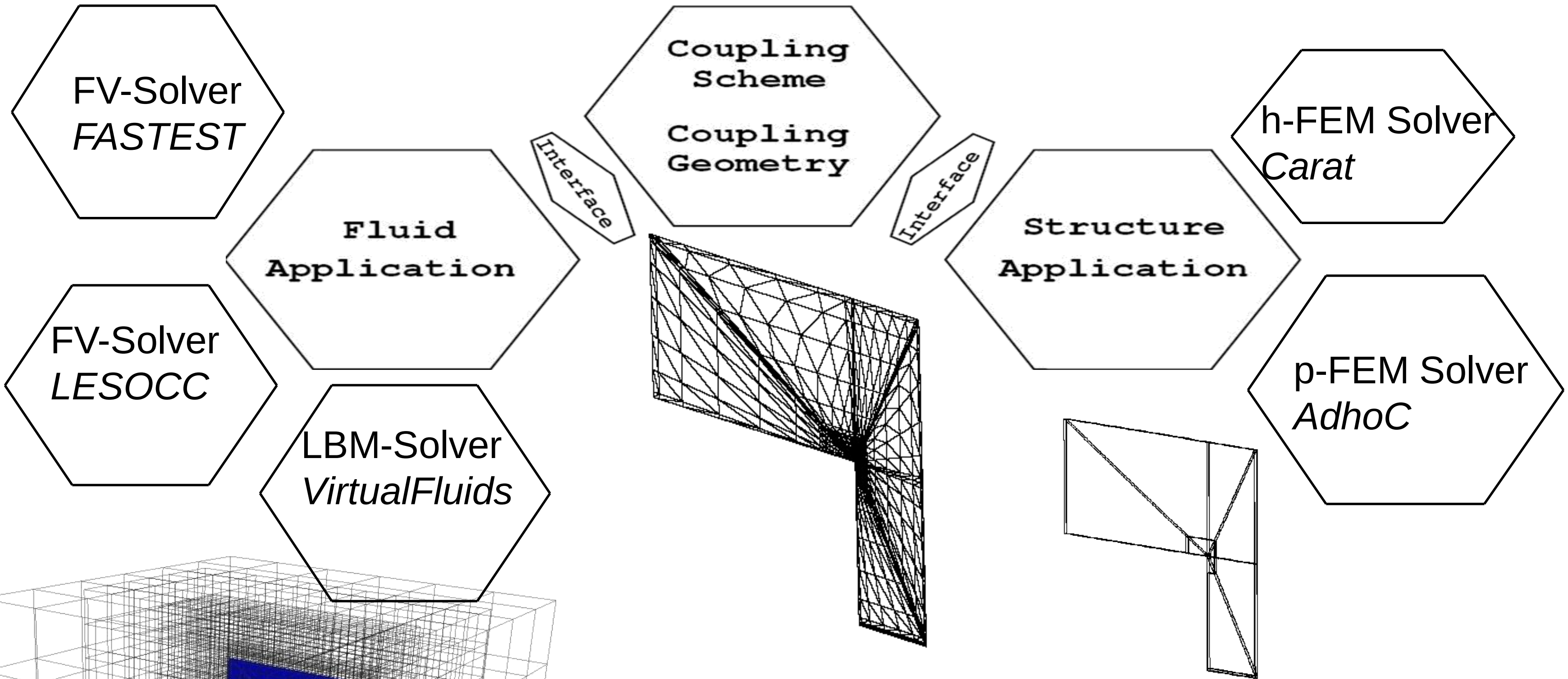
parallel efficiency (96 processors): 95 %

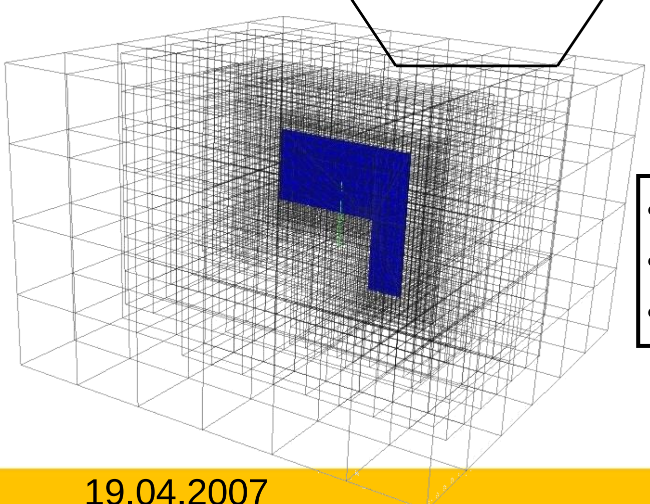
# Fluid-Structure-Interaction with Lattice-Boltzmann-Methods on tree type grids

DFG-Research group 493: fluid-structure-interaction: Modeling, Simulation, Optimization



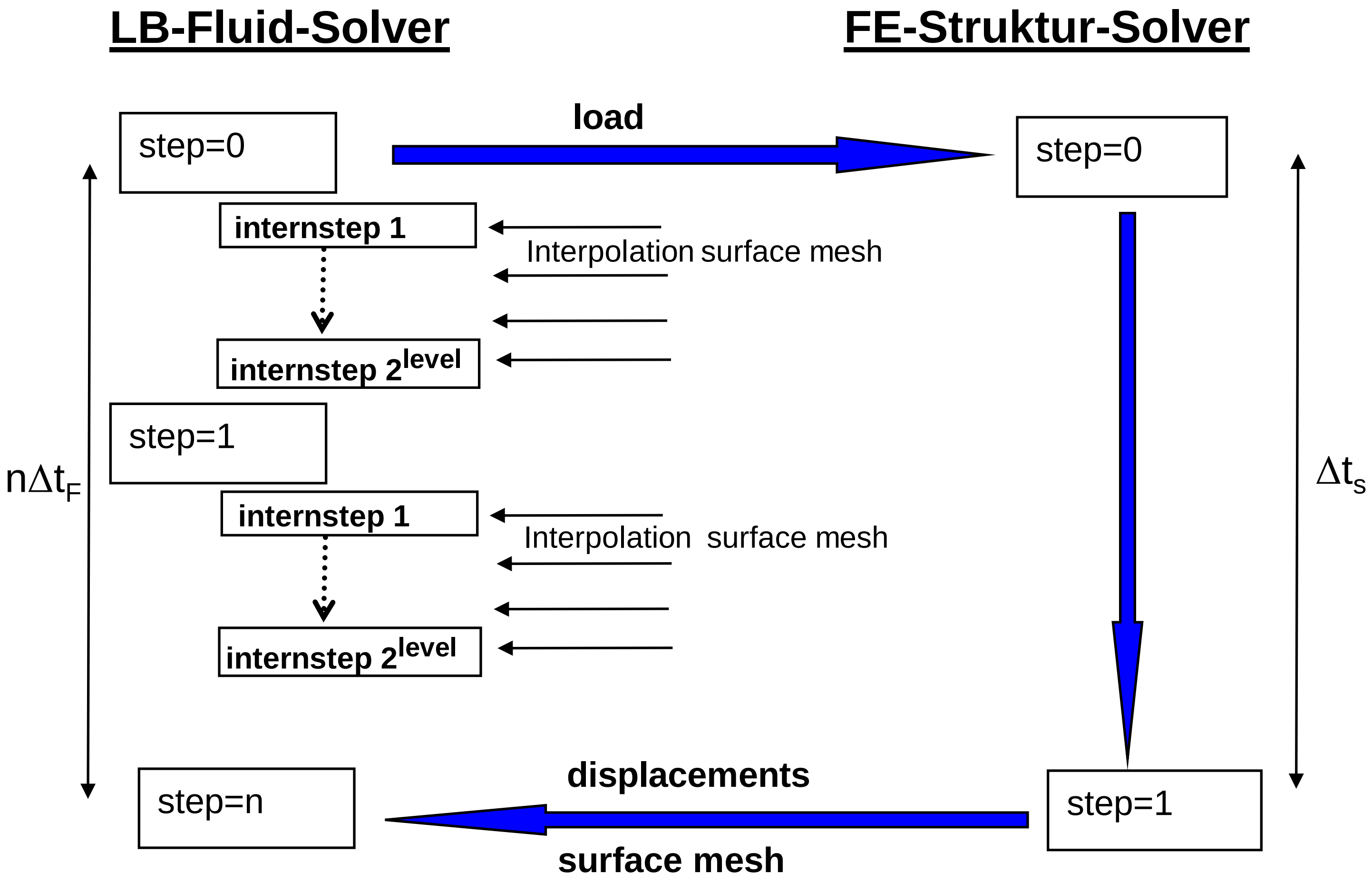
# Coupling of Structure- and Fluid Solver



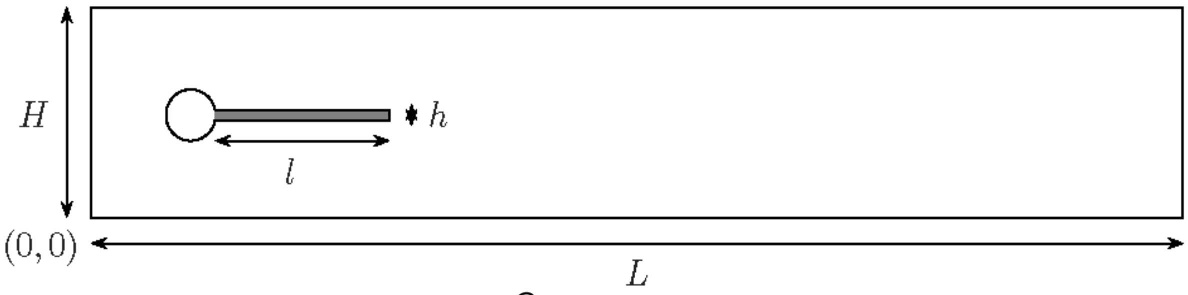



- distributed environment
- transformation loads / displacements
- coupling through moving triangle mesh

# Coupling algorithm (explicit) / Mapping of surface mesh

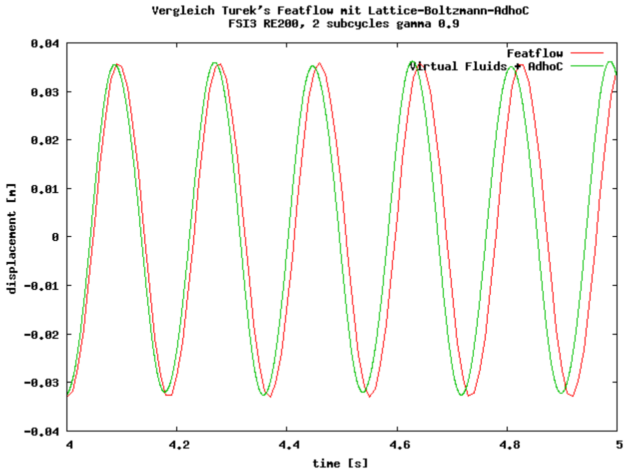


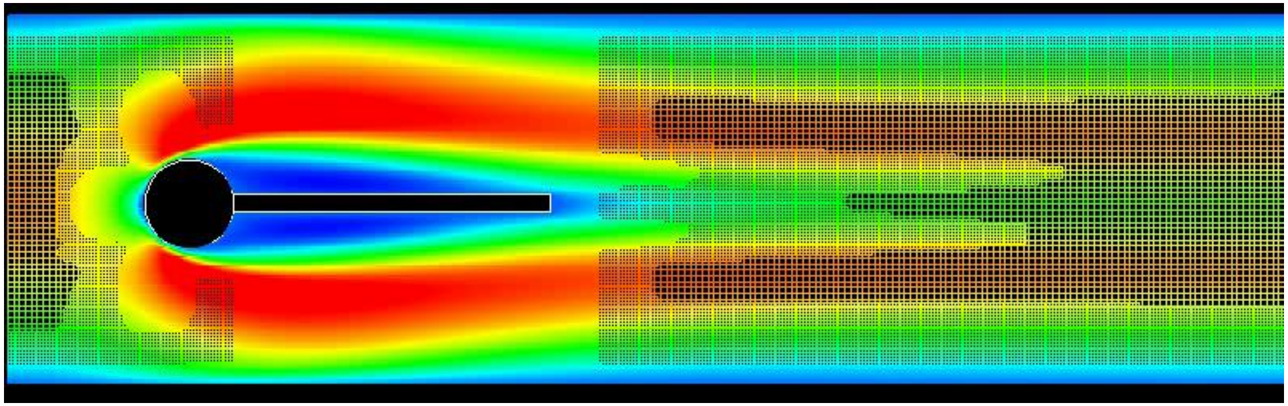
# Benchmark Fluid-Structure-Interaction (Turek/Hron)




$\rho_f = 1000[\mathrm{kg\,m^{-3}}]$
$\rho_s = 10000[\mathrm{kg\,m^{-3}}]$
$\nu_f = 1/1000[\mathrm{m^2 s^{-1}}]$
$\nu_s = 0.4[-]$

$E = 1.4E6[\mathrm{Pa}]$
$\bar{U} = 1[\mathrm{m\,s^{-1}}]$



# Multiphase flow in bioreactors

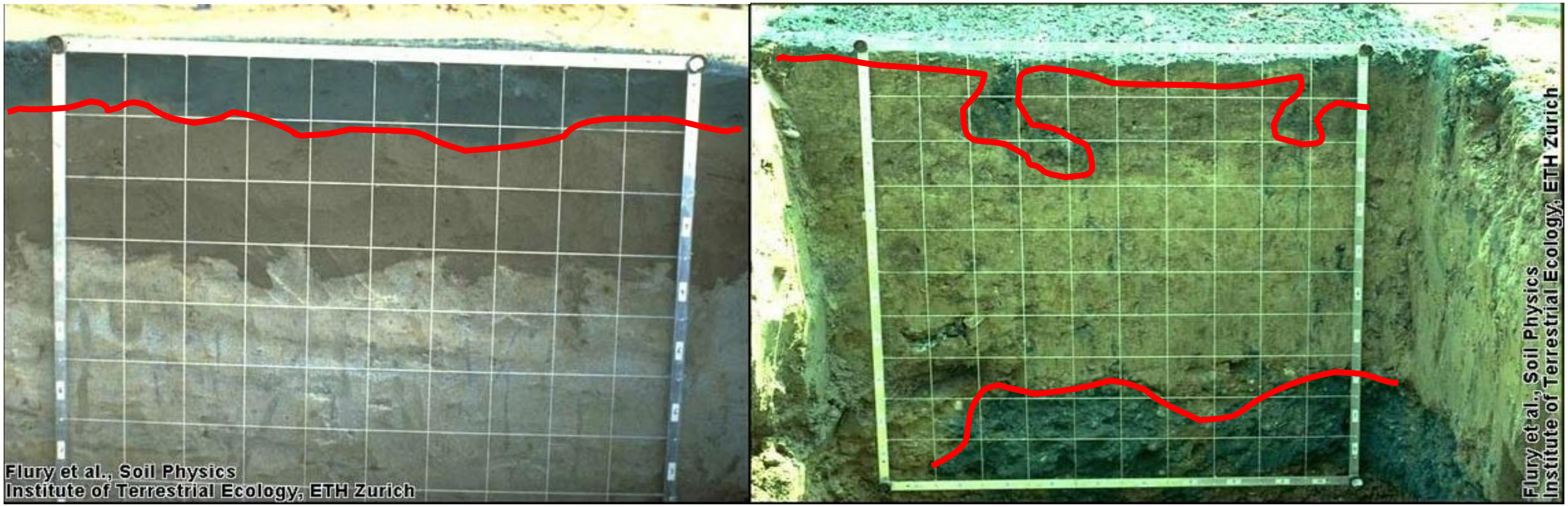
Simulation of bubble

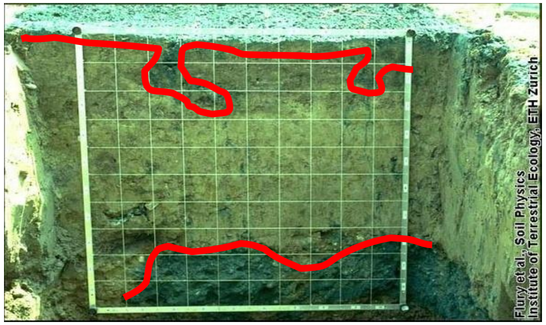
Simulation of bioreactor

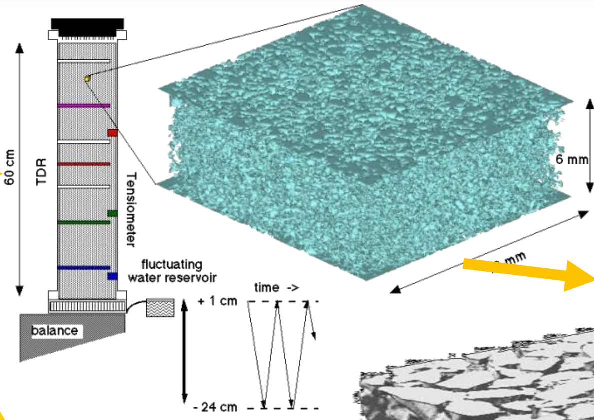
# FIMOTUM - FIrst Principle Based MOdelling of Transport in Unsaturated Media

**Co-operation with:**
-Institut für terrestrische Ökologie, Paul Scherrer Insitut (ETH-Zürich)
-Institut für Wasserbau (Univ. Stuttgart)
-Institut für Geoökologie (TU-BS)

**goal:**
Prediction of  transport processes in unsaturated soils

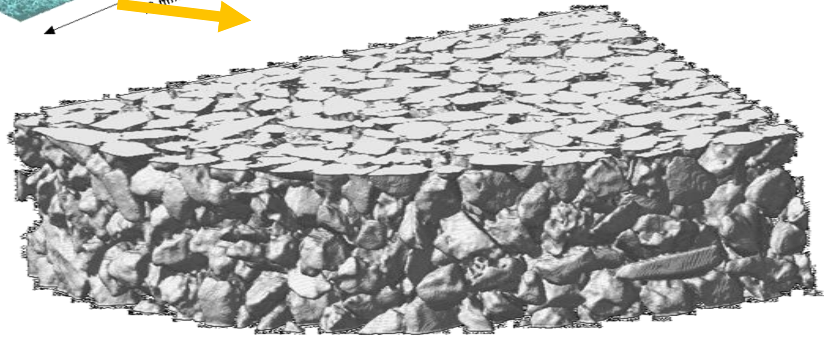




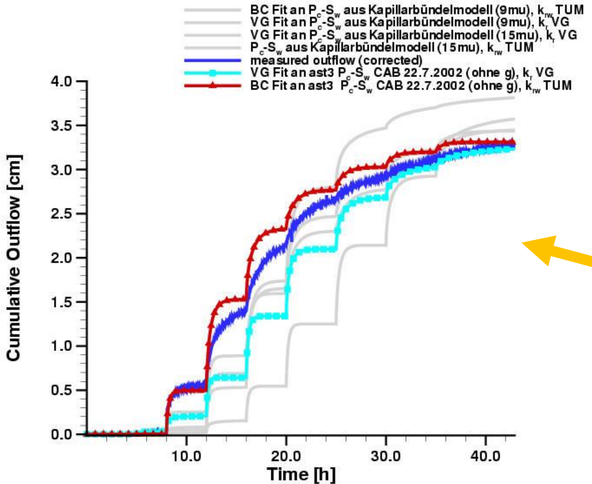



computer tomography

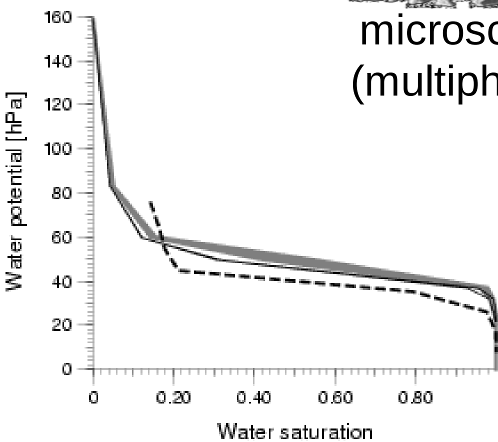
reconstruction

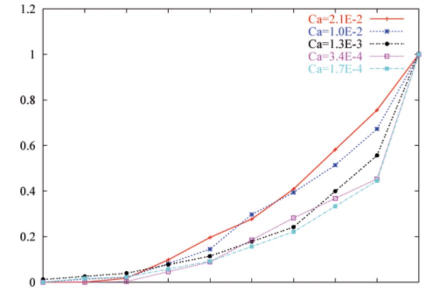


comparison with experiment




microscopic LB-simulation (multiphase Navier-Stokes)



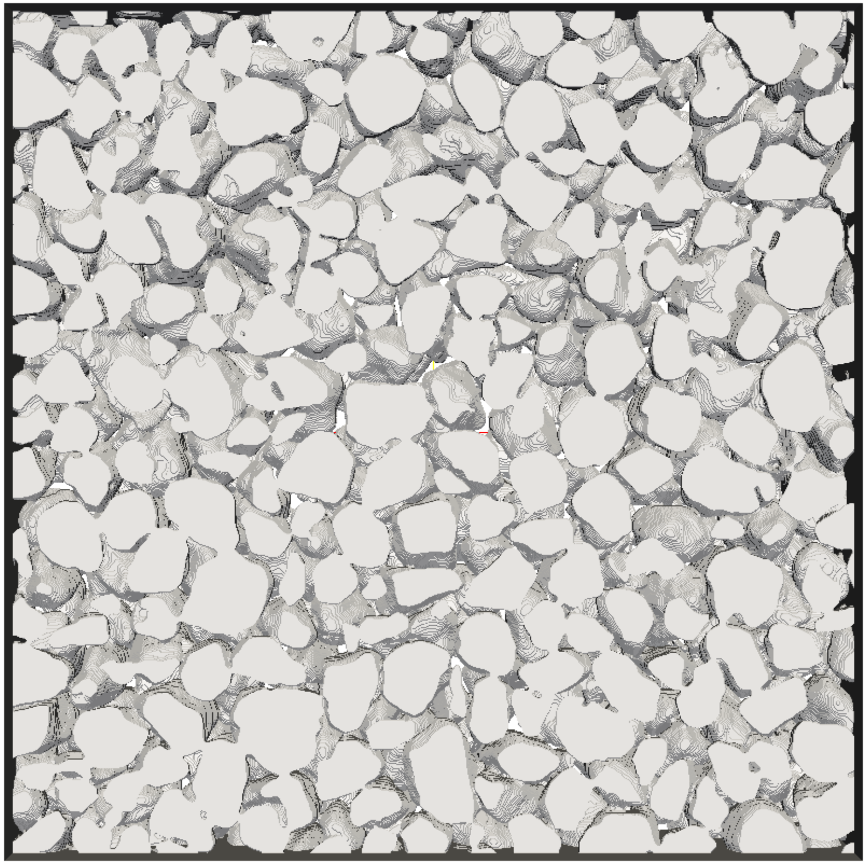
macroscopic FE-simulation (macroscopic multi-phase equations)

material functions:

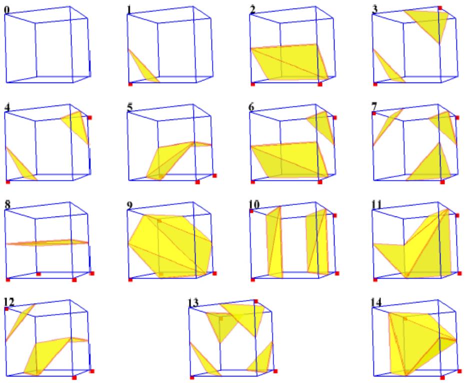
- relative permeabilities
- capillary pressure

# Reconstruction of geometry

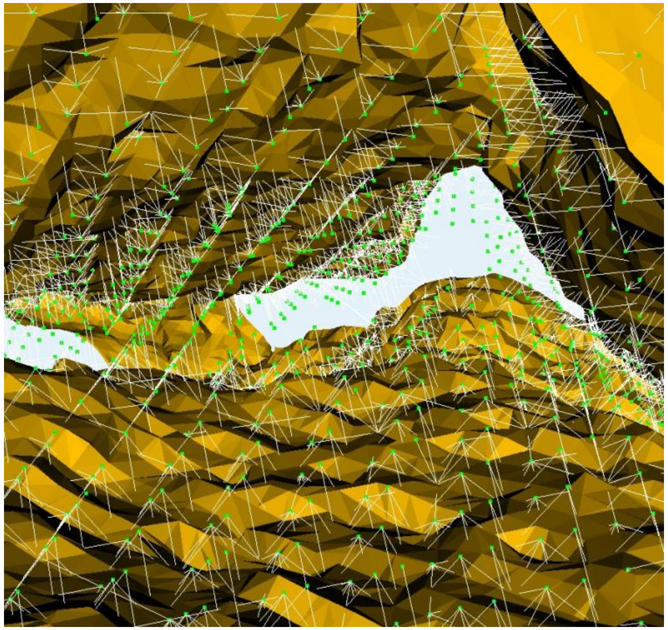
voxel geometry
(Scans von Kästner and Lehmann)



Iso-contur algorithm:
Marching cube

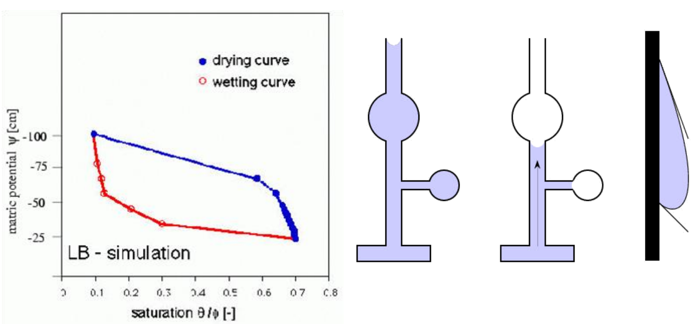


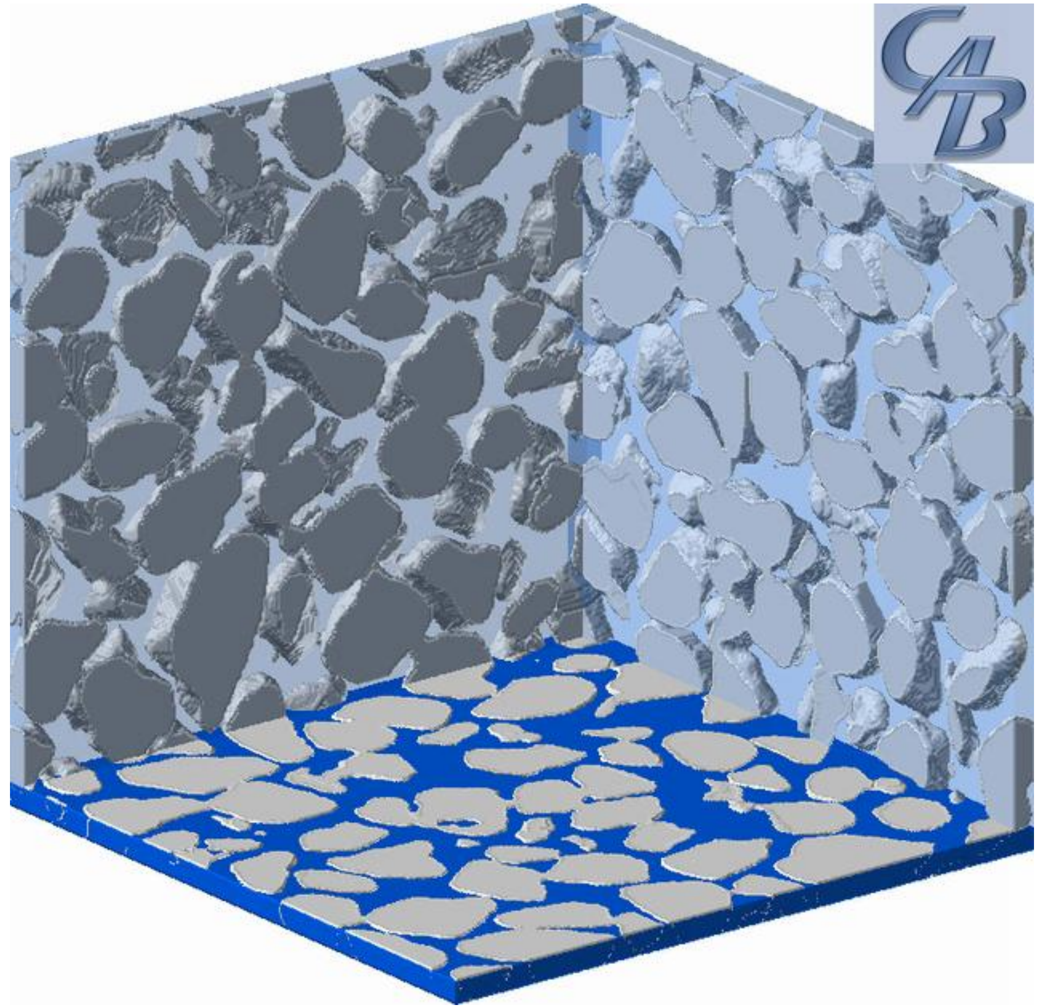
800^3 voxel set:
110 Mio triangles



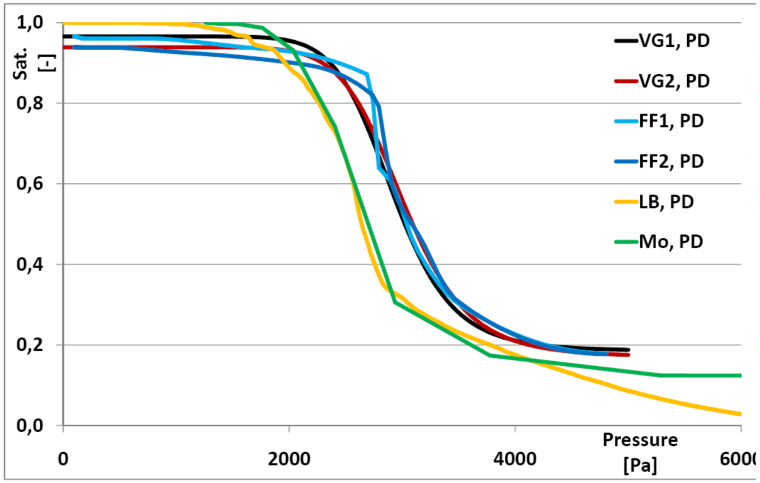
# Numerical Experiments: Drainage-Imbibition

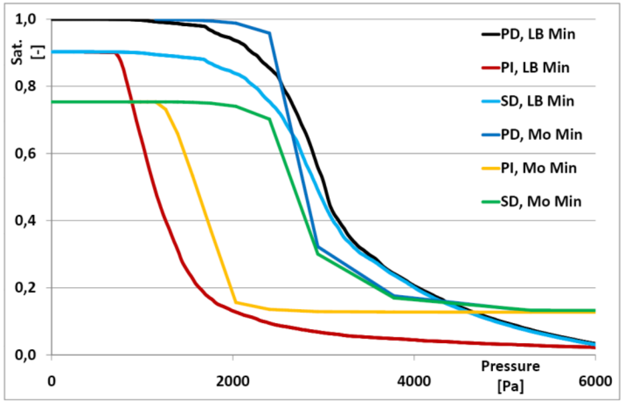
- sand, 100-500 micrometer
- resolution: ca. 5 micron
- marching cube/triangles
- Parallel
- multiphase-model
- up to 500 Mio. grid nodes

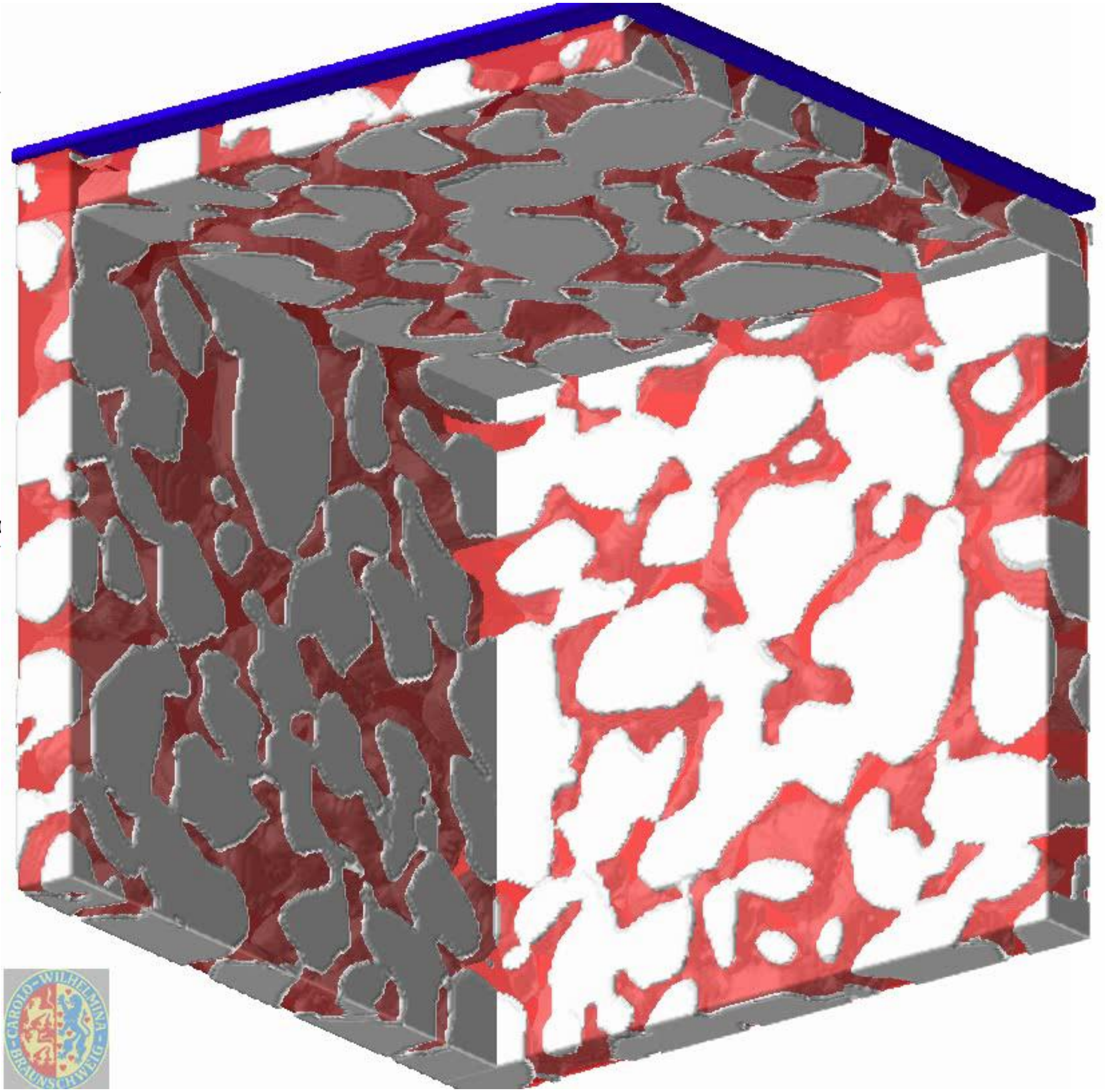


# Hysteresis

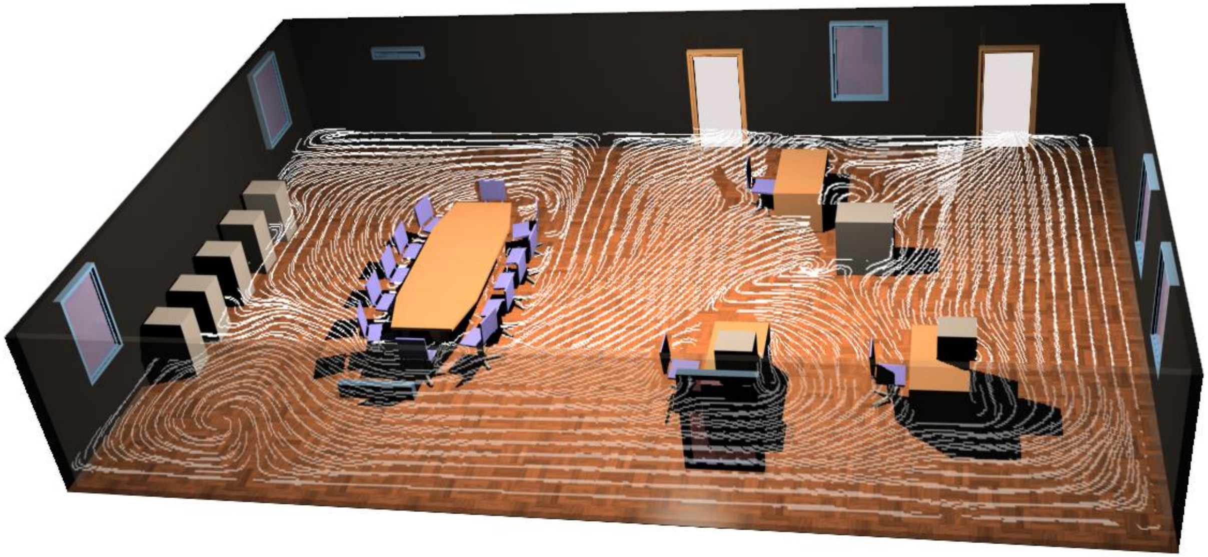

1,0
0,8
0,6
0,4
0,2
0,0
Sat. [-]
0
2000
4000
6000
Pressure [Pa]
VG1, PD
VG2, PD
FF1, PD
FF2, PD
LB, PD
Mo, PD


1,0
0,8
0,6
0,4
0,2
0,0
Sat. [-]
0
2000
4000
6000
Pressure [Pa]
PD, LB Min
PI, LB Min
SD, LB Min
PD, Mo Min
PI, Mo Min
SD, Mo Min



# HVAC comfort optimization

DFG-SPP 1103
"Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau"



# Optimization of HVAC layout

- virtual design space
- integration of different disciplines in one product model
- efficient methods for model transfer
- interactive optimization (Computational-Steering)
- visualization of target function immediately

# HVAC Simulation: Predicted Mean Vote Index

**„Multi-Physics“ Problem**

temperature (LB)

air velocity (LB)

radiation (Radiosity)

PMV

humidity (LB)

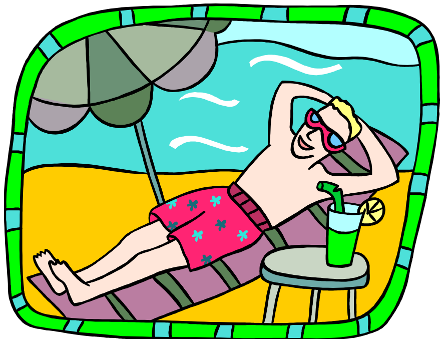
metabolic rate (MET-Index)

clothing (Clo-Index)

1.2 CLO

0.15 CLO

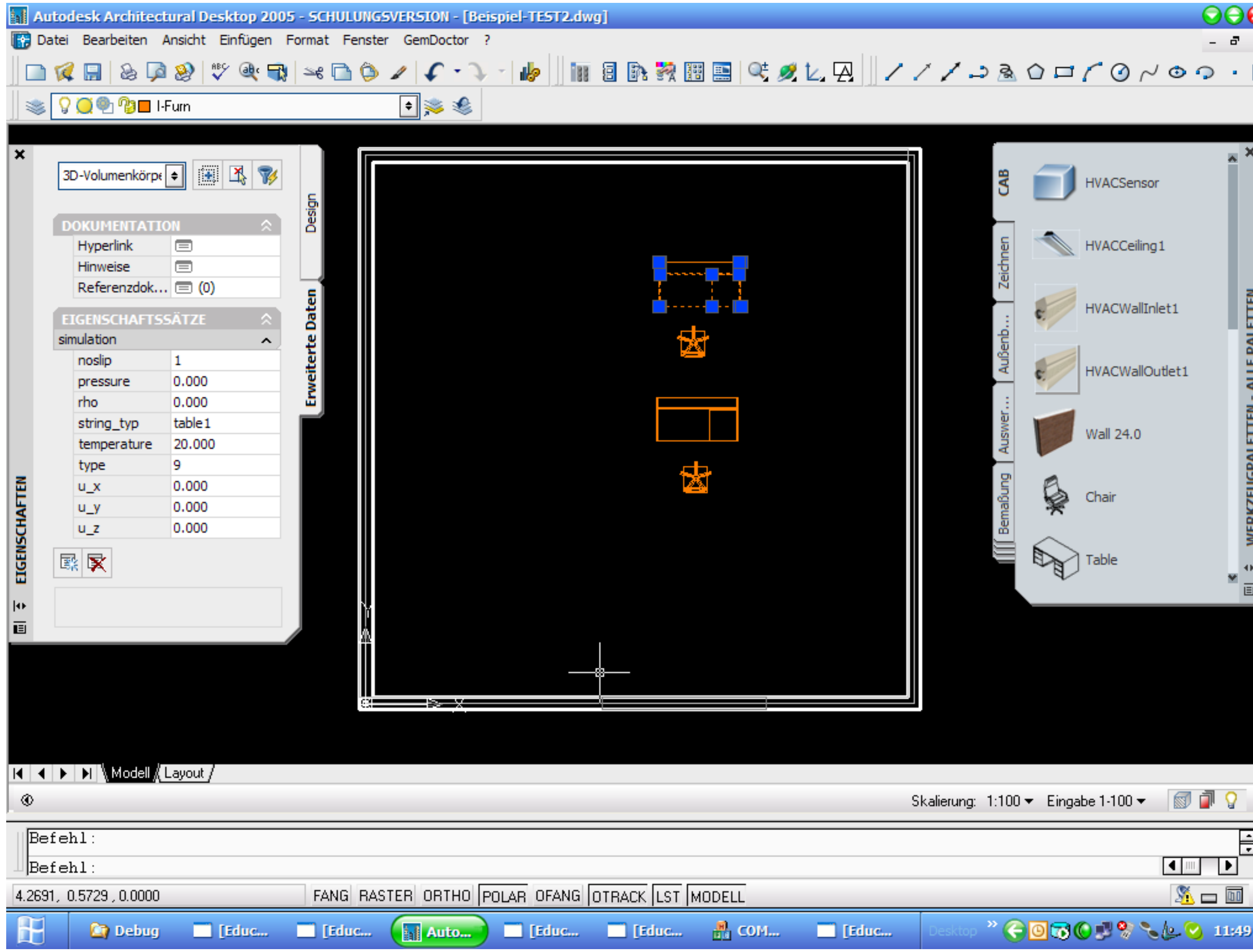
8 MET



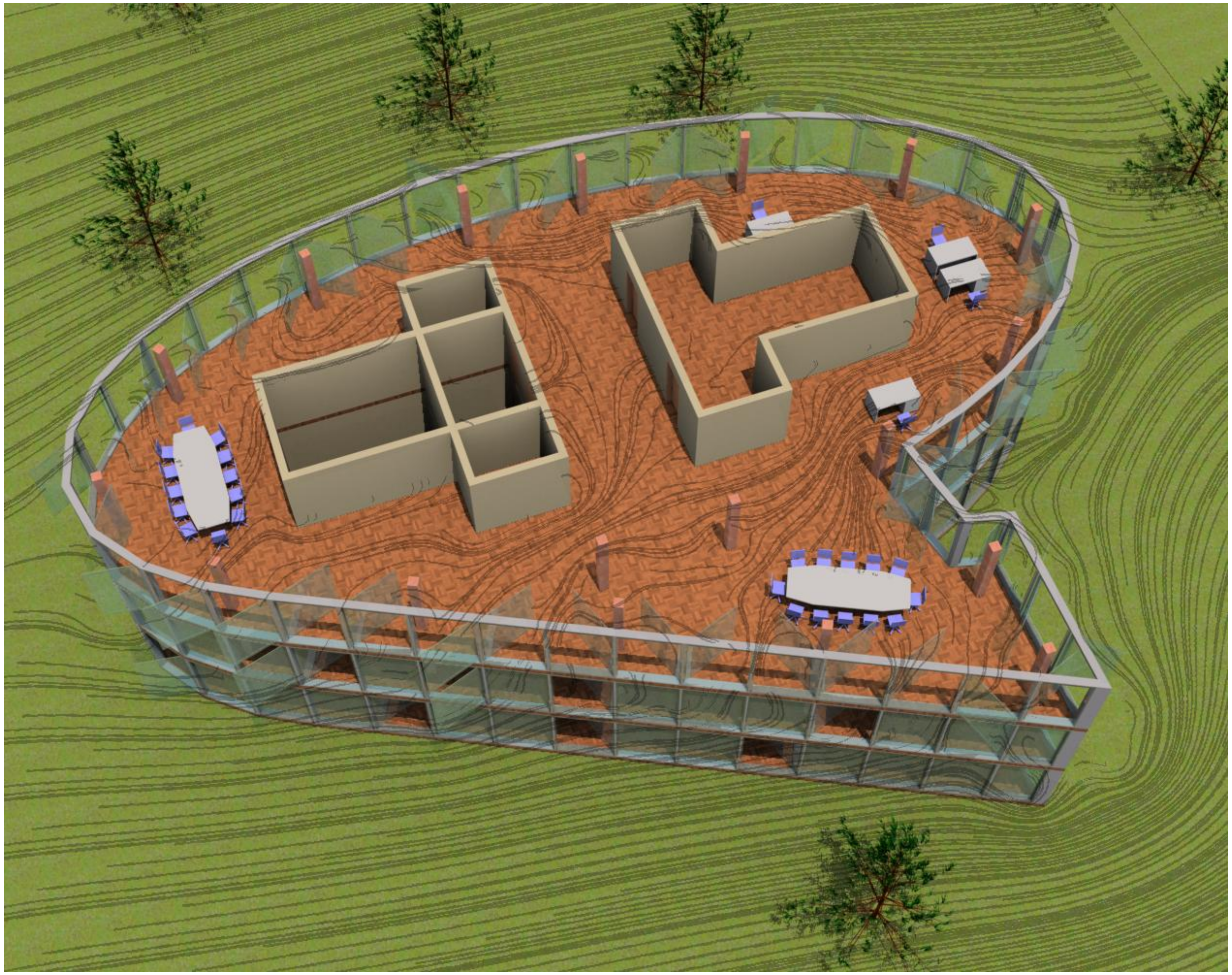
0.8 MET

# Modeler

- *Autodesk Architectural Desktop* (ADT)
- *Object Modeling Framework* (OMF)



# From Model to Simulation

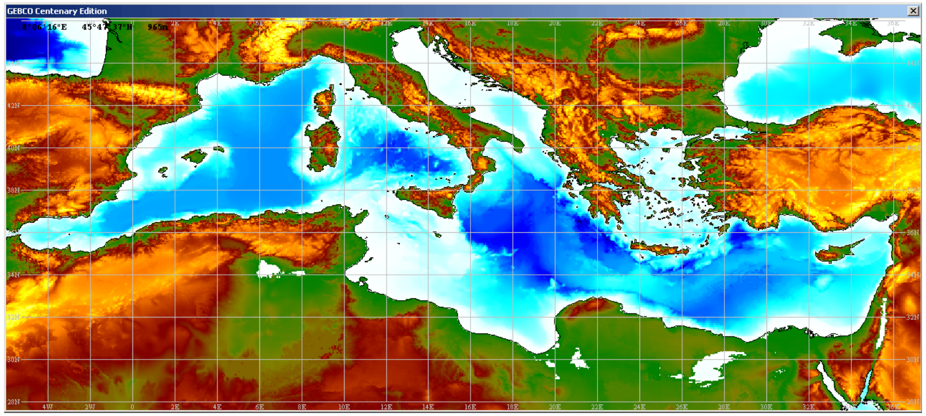


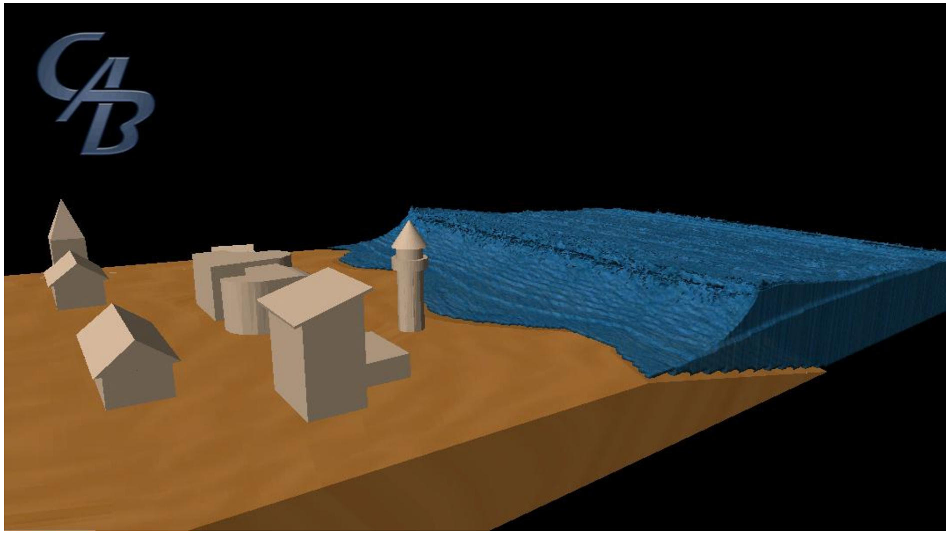
- IFC product model
- modeler / CAD
- trangulation
- octree
- CFD Simulation
- visualization/ post-processing

# Simulation of spring events and tsunamis

- „automatic“ acquisition of GIS data
- fast mesh generation
- adaptive mesh refinement / coarsening using blocks
- coupling 3D free surface + 2D shallow water
- faster than real-time

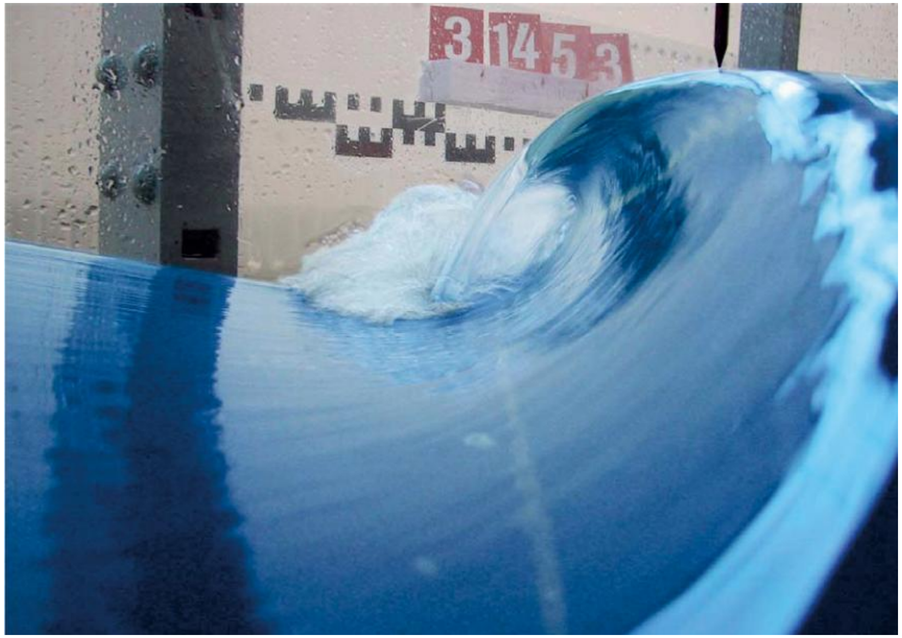
GIS-Data from GEBCO Digital Atlas, British Oceanographic Data Centre



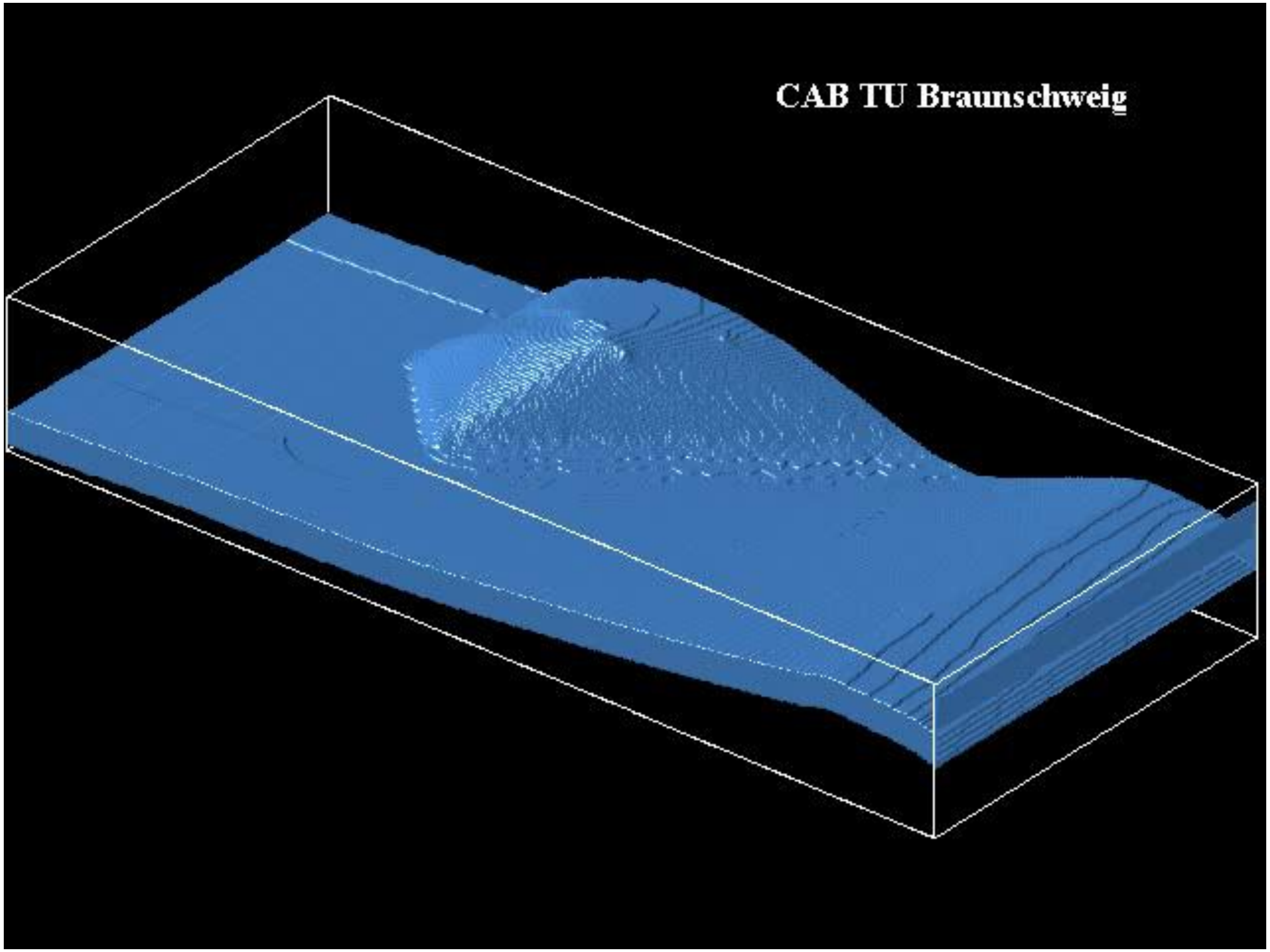



# Simulation of a surf wave

**TU München, Oskar von Miller – Institut, Versuchsanstalt für Wasserbau und Wasserwirtschaft**



# Simulation of a surf wave



# Performance of LB kernels

simple LB D2Q9 kernel, propagation optimized data layout (Wellein 2006):

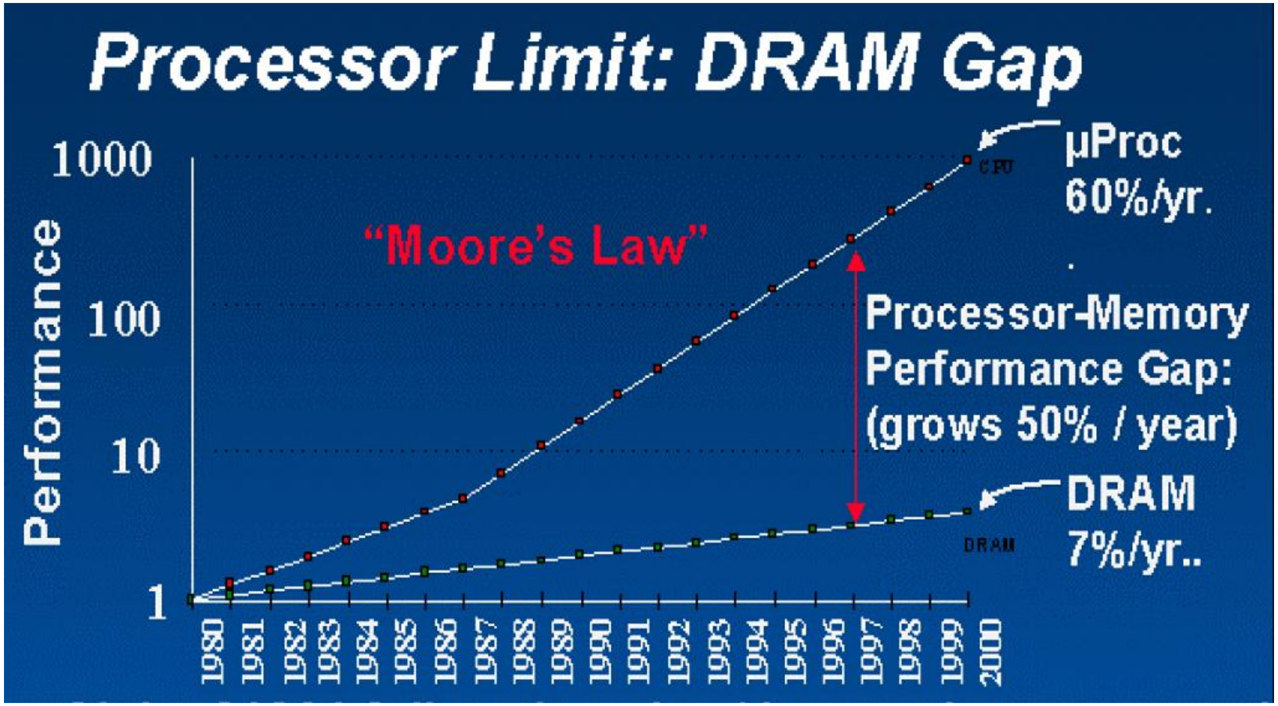
```
float F(nx,ny,8,2)
...
for(y=0 ; y<ny ; y++){
   for(x=0 ; x<nx ; x++){
        load F(x,y,0,told)
        load F(x,y,1,told)
        ...
        load F(x,y,8,told)

        // collision
        ...(complex computations)

        // propagation
        save F(x,y,0,tnew)
        save F(x+1,y,1,tnew)
        save F(x,y+1,2,tnew)
            ...
        save F(x+1,y-1,8,tnew)
   }
}
```

# DRAM GAP

contiguous memory access is mandatory!



# Performance of LB-Kernels

- LUPS: Lattice updates per second
- Limitation by Memory Bandwidth: d2q9
  max LUPS = theoretical BW / [ (1(read)+2(write)) * 4 byte * 9 particles ]

  Example:
  BW=6GB/s
  max LUPS = 6.0E9 GB/s / (3*4*9 Byte/lattice node) = 55E6 LUPS
- Limitation by FLOPS: d2q9
  max LUPS = theortical FLOPS / (NCOLL Bytes/lattice node)

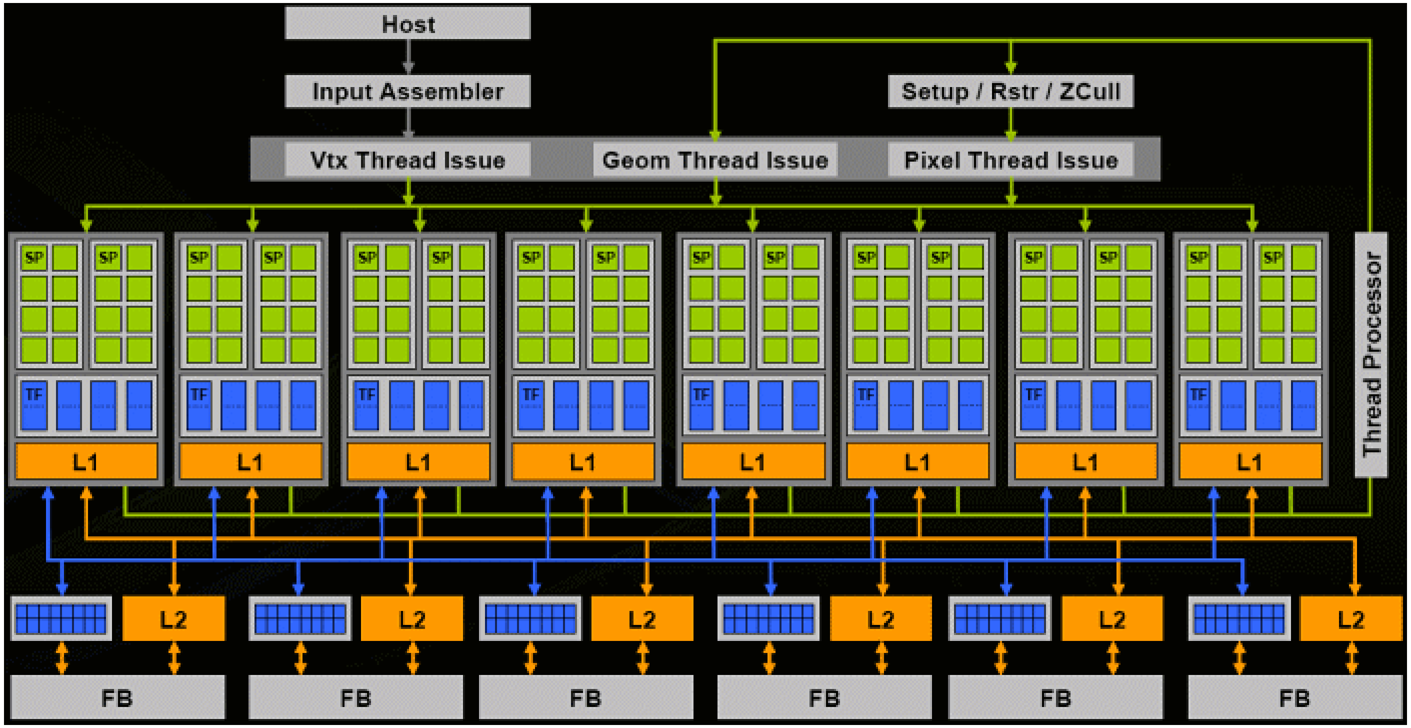
  Example:
  8 GFLOPS=8E9 FLOPS
  NCOLL = 200 FLOP (d2q9, MRT)
  max LUPS = 8.0E9 FLOPS / (200 FLOP/lattice node) = 40E6 LUPS

# New hardware developements



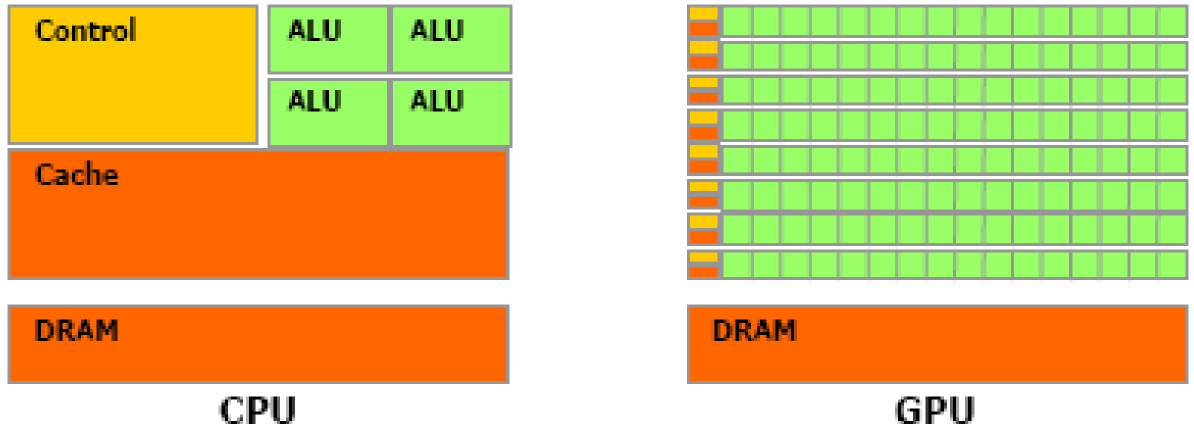
- nVIDIA GTX 8800
- Compute Unified Device Architecture (CUDA)

# nVIDIA - G80: the parallel stream processor



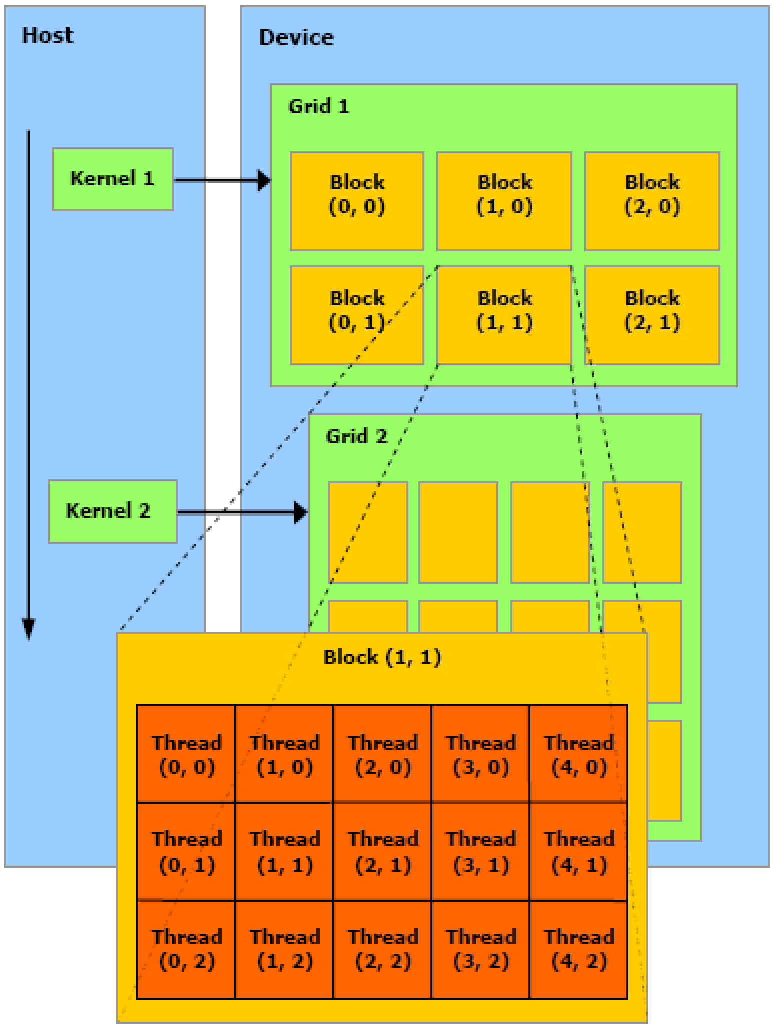
# nVIDIA - G80: the parallel stream processor

- The G80 has eight groups of 16 stream processors, for a total of 128 SPs
- Generalized floating-point processors capable of operating on any manner of data.
- G80's stream processors are scalar — each SP handles one component
- SPs are clocked 1.35GHz, giving the GeForce 8800 a tremendous amount of floating-point processing power: 1.35*2*128 = 345 GFLOPs
- Eight "clusters" of stream processors are connected to six Render Output Unit (ROP) by a crossbar-style switch
- Each ROP partition has an 64-bits wide interface to graphics memory, which is clocked at 900 MHz.
- Memory Bandwidth: 6*64/8*0.9*2 (DDR) GB/s = 86 GB/s

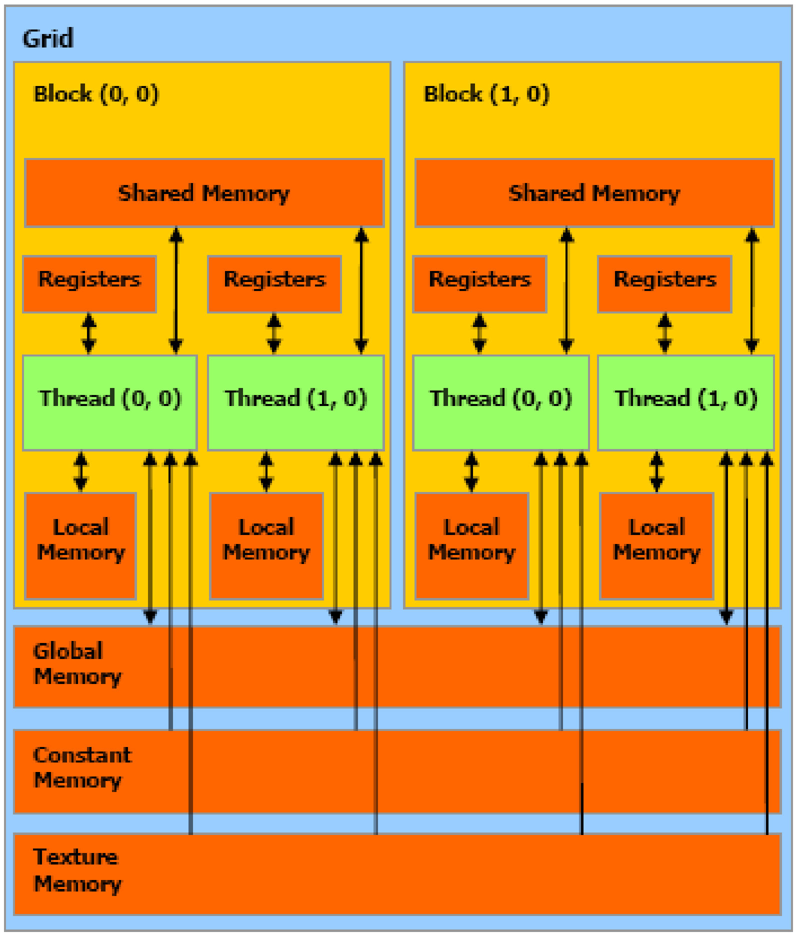



| Platform | Peak[Gflops] | MemBW[GB/s] | price [Euro] |
|---|---:|---:|---:|
| Intel Core 2 Duo (2 GHz) | 16 | 6-10 | 2000 |
| NEC SX-8R A (8 CPUs) | 281 | 563 | Ca. 400 000 |
| nVIDIA 8800 GTX | 345 | 86 | 500 |

Programming model




Memory Model



# Application Programming Interface (API)

- Thread Block
- Grid of Thread Blocks
- Function Type Qualifiers (_device_, _global_, _host_)
- Variable Type Qualifiers (_device_,_shared_)
- Memory management  (cudaMalloc, cudaMemcpy)
- Synchronisation (_syncthreads() )

# Memory Bandwidth

- Effective bandwidth of each memory space depends significantly on the memory access pattern
- simultaneous memory accesses of one thread block can be coalesced into a single contiguous, aligned memory access if:
  - thread number N  access address  BaseAddress + N
  - BaseAddress has to be aligned to 16*sizeof(type) bytes ( otherwise memory bandwidth performance breaks down to about 8 GB/sec )

Simple example: Multiply a matrix with 0.5

The host code:

```
...
// allocation of host memory
float* fH  = (float*) malloc(mem_size_Mat);
// initialize host memory
for(y=0 ; y< ny ; y++){
 for(x=0 ; x< nx ; x++){
  k = nx*y+x;
  fH[k]=1.0;
 }
}
// allocate device memory
cudaMalloc((void**) &f0, mem_size_Mat);
cudaMalloc((void**) &f1, mem_size_Mat);

// copy host memory to device
cudaMemcpy(f0,  fH,  mem_size_Mat,  cudaMemcpyHostToDevice);

// setup execution parameters
dim3 threads(num_threads, 1, 1);
dim3 grid(nx/num_threads, ny);
//Execute the kernel
Kernel<<< grid, threads >>> ( nx, f0,f1);
...
```

The device code (kernel):

```
_global_ void Kernel(int nx,float* f0,float* f1)
{
    // number of threads
    int num_threads = blockDim.x;

    // Thread index
    int tx = threadIdx.x;

    // Block index x
    int bx = blockIdx.x;

    // x-Index
    int x = tx +  bx*num_threads;
    // Block index y = y-Index
    int y =  blockIdx.y;

    // f0[k]:Load data from device memory
    // f1[k]:Write data to device memory
    int k = nx*y + x;
    f1[k]=0.5*f0[k];
}
```

→ 56 GB/s memory throughput : 65% peak perf.

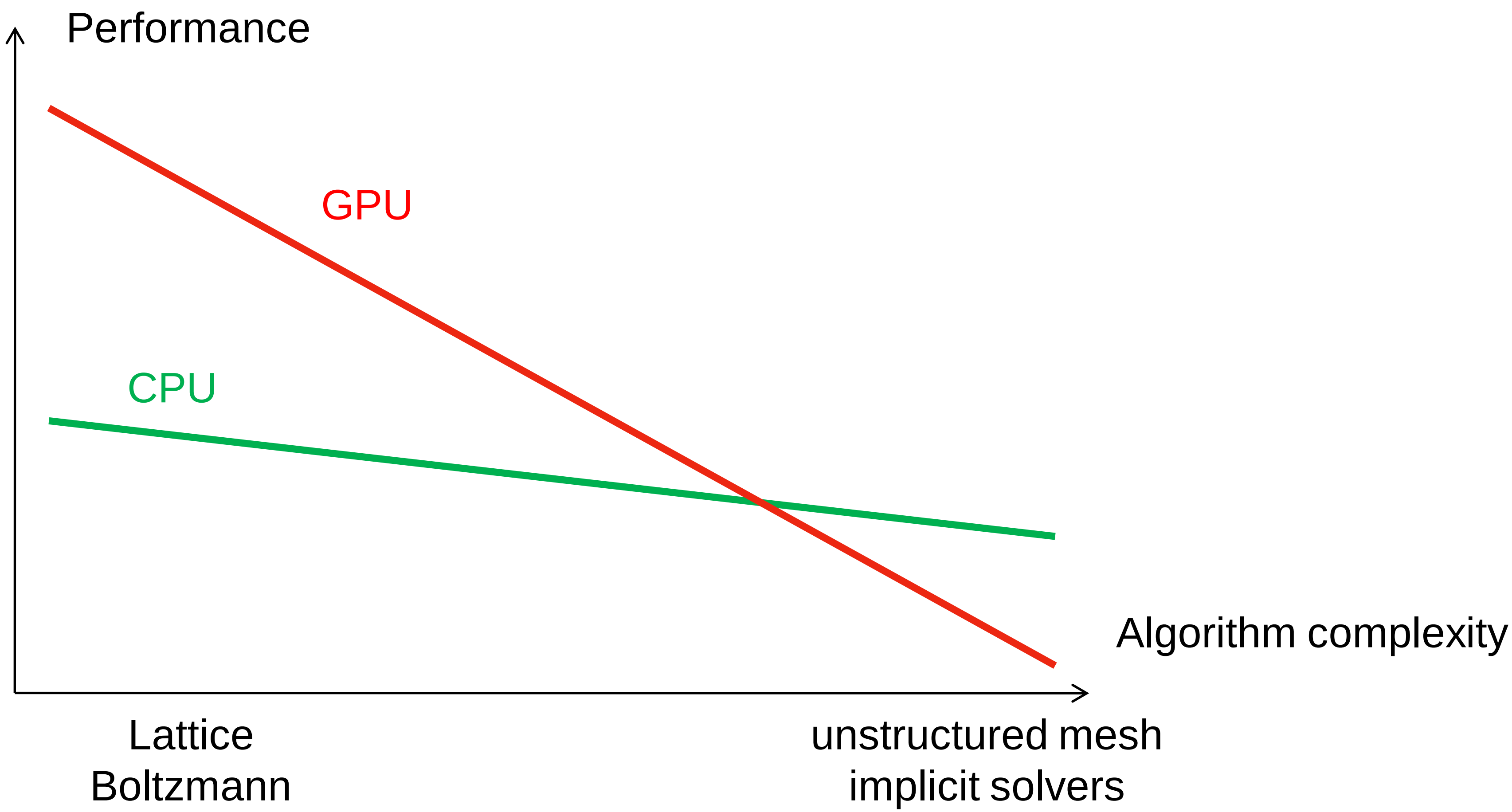
## Nonlocal operations

- Each thread block has shared memory of 16 KB (2 cycles latency)
- Use shared memory for nonlocal operations (LB: Propagation)
- Synchronize
- Write back to device memory
- Synchronize Grid of Thread Blocks over borders

## Results

| Platform | MLUPS | MemBW[GB/s] | GFlops |
|---|---|---|---|
| Intel Core Duo (2 GHz) | 8 | 1.0 (17 %) | 1.6 (20%) |
| Intel Core 2 Duo (2 GHz) | 16 | 2.0 (25 %) | 3.2 (20%) |
| nVIDIA 8800 GTX | 330 | 25.0 (30 %) | 66.0 (19 %) |

# CPU versus GPU – The war is on



Thank you very much for your attendance!