Simulating Complex Flows on 300,000 Compute Cores

U. Rüde (LSS Erlangen, ruede@cs.fau.de)

joint work with

J. Götz, M. Stürmer, K. Iglberger, S. Donath, C. Feichtinger, T. Preclik,
T. Gradl, C. Freundl, H. Köstler, T. Pohl, D. Ritter, D. Bartuschat, P. Neumann,
G. Wellein, G. Hager, T. Zeiser, J. Habich (RRZE)
N. Thürey (ETH Zürich)

Lehrstuhl für Informatik 10 (Systemsimulation)
Universität Erlangen-Nürnberg
www10.informatik.uni-erlangen.de

May 14, 2010
LSEC, Beijing
Overview

- Rigid Body Dynamics
  - Parallel Computation of Granular Flows
- Flow Simulation with Lattice Boltzmann Methods
  - The LBM
  - Extensions and Applications
- Fluid-Structure Interaction with Moving Objects
- Fluctuating LBM for FSI with nano particles
- Bubbly Flows and Foams
- Animations
- Conclusions
Rigid Multibody Dynamics
What is rigid body dynamics?
Modelling Rigid Body Dynamics

- Generalized positions: $q(t)$
- Generalized mass matrix: $M(q)$
- Newton's law: $f(t, q, v) = M(q)\dot{v}, \quad \dot{q} = v$
- Generalized forces: $f(t, q, v)$
- Semi-Implicit Euler Discretization
Contacts

- Gap functions
  \[ \Phi_n(q(t)) \geq \vec{0} \]

- Complementarity conditions (for impulses \( x_n \))
  \[ \Phi_n(q(t + \Delta t)) \geq \vec{0} \perp \vec{x}_n \geq \vec{0} \]

- Coulomb dry friction (with simplifications)
  - tangential contact forces bounded by normal forces times friction coefficient
Rigid body dynamics with friction and objects of more general shape
(T. Preclik, K. Iglberger)

Solve linear complimentarity problem in each time step
Collisions & Contacts between Rigid Objects
Conditions

- Überlappung vermeiden
  \[\ddot{d} \geq 0\]

- Kontaktkräfte dürfen nicht "festhalten"
  \[f \geq 0\]

- Keine Kraft falls der Kontakt abbricht
  
  entweder \(\ddot{d} = 0\) und \(f \geq 0\)
  
  oder \(\ddot{d} > 0\) und \(f = 0\)

\[f \cdot \ddot{d} = 0\]
Multiple Contacts

- $f_2 \hat{n}_2$ has influence on the acceleration of $p_1$ and consequently on $\ddot{d}_1$
- Isolated consideration of contacts is not possible!
- $\ddot{d}_1$ in dependence of all contact forces to be represented
Force balance at multiple contacts

\[ \ddot{d}_1 = a_{11} f_1 + a_{12} f_2 + a_{13} f_3 + b_1 \]
\[ \ddot{d}_2 = a_{21} f_1 + a_{22} f_2 + a_{23} f_3 + b_2 \]
\[ \ddot{d}_3 = a_{31} f_1 + a_{32} f_2 + a_{33} f_3 + b_2 \]
Composing More Complicated Objects

- **The physics engine pe**
  - Framework for accurate and games rigid body simulation
  - Both primitive and compound geometries
  - Calculation of contact forces and torque
  - Accurate friction calculation during collision
  - Coupling to different simulation frameworks
Dynamics of many objects (rigid body dynamics)
Dynamics of many objects (now with friction)
Dynamics of many objects (composed objects)
Newton’s Cradle
Parallel Rigid Body Dynamics for Simulating Granular Flows
Parallel rigid body dynamics

720,484 spherical particles
12,000 time steps of 0.01s
6.18 h on 32 Xeon 5160 Woodcrest cores 3.0 GHz.
Parallel Rigid Body Dynamics

- Algorithm is based on
  - This algorithm avoids the (global) LCP and treats collisions locally
  - (Modelling) errors can occur when more than two particles are in contact

- Parallel algorithm described in:
  - K. Iglberger, UR: Massively parallel rigid body dynamics simulations, Computer Science - Research and Development (23) 2009, pp. 159-167
  - K. Iglberger, UR: Massively parallel granular flow simulations with non-spherical particles, 2010 (to be published) - PRACE Award 2010
Parallel Rigid Body Dynamics

- No point masses, but volumetric, geometrically defined objects
  - objects may (geometrically) extend across several processors
- Objects overlapping with process boundaries must be synchronized
- Objects are assigned logically to exactly one process
- Unique identifier from rank of the process and local counter
Parallel rigid body dynamics

344,960 spherical particles
60,300 time steps
time step size 0.001s
3.6 hours on 64 Xeon 5160 Woodcrest @3.0GHz
single time step: 0.22s
Each process: 5390 particles
Granular Flows with Non-Spherical Particles and Frictional Elastic Collisions

64 Processes, 62658 particles, each composed of 2-5 overlapping spheres, approx. 13 hours runtime
Weak Scaling

up to 9120 processor cores
more than one billion geometric objects

HLRB-II: SGI Altix
Leibniz Computing Center Garching
Itanium based
63 TFlop Peak
40 TByte memory

<table>
<thead>
<tr>
<th># Cores</th>
<th># Spheres</th>
<th>Partitioning</th>
<th>Runtime [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>16 000 000</td>
<td>128 x 1 x 1</td>
<td>1059.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 8 x 1</td>
<td>1139.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 4 x 4</td>
<td>1400.87</td>
</tr>
<tr>
<td>256</td>
<td>32 000 000</td>
<td>256 x 1 x 1</td>
<td>1056.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 16 x 1</td>
<td>1145.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 8 x 4</td>
<td>1406.18</td>
</tr>
<tr>
<td>448</td>
<td>56 000 000</td>
<td>448 x 1 x 1</td>
<td>1049.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28 x 16 x 1</td>
<td>1141.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 8 x 7</td>
<td>1405.81</td>
</tr>
<tr>
<td>960</td>
<td>120 000 000</td>
<td>960 x 1 x 1</td>
<td>1051.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 x 30 x 1</td>
<td>1148.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 x 8 x 8</td>
<td>1408.58</td>
</tr>
<tr>
<td>1920</td>
<td>240 000 000</td>
<td>1920 x 1 x 1</td>
<td>1065.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60 x 32 x 1</td>
<td>1157.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 15 x 8</td>
<td>1414.46</td>
</tr>
<tr>
<td>3840</td>
<td>480 000 000</td>
<td>3840 x 1 x 1</td>
<td>1055.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64 x 60 x 1</td>
<td>1157.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 16 x 15</td>
<td>1418.86</td>
</tr>
<tr>
<td>9120</td>
<td>1 140 000 000</td>
<td>9120 x 1 x 1</td>
<td>1066.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96 x 95 x 1</td>
<td>1359.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24 x 20 x 19</td>
<td>1613.73</td>
</tr>
</tbody>
</table>
Computational Fluid Dynamics with the Lattice Boltzmann Method

Falling Drop with Turbulence Model (slow motion)
LBM in Action
(with free surface)
Computational Fluid Dynamics

The classical method is based on the Navier-Stokes equations (here incompressible form)

\[
\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \nu \Delta \mathbf{v} + \mathbf{f},
\]

\[\nabla \cdot \mathbf{v} = 0,\]

More recently, the Lattice Boltzmann method (LBM) has been developed as an alternative

\[
\partial_t f(\bar{x}, \bar{v}, t) + \bar{v} \nabla_x f(\bar{x}, \bar{v}, t) + \frac{\vec{F}}{m} \nabla_v f(\bar{x}, \bar{v}, t) = I_c(f),
\]
The Lattice-Boltzmann Method (1)

- Based on cellular automata
  - Introduced by von Neumann around 1940
  - Famous: Conway’s Game of Life
  - Lattice Gas Cellular Automata were used to simulate gases
- Complex system with simple rules
  - Regular grid
  - Local rules specifying time evolution
- Interpretation: Discretization of the Boltzmann equation on uniform grid
- Explicit time stepping,
- Intrinsically parallel
The Lattice-Boltzmann-Method

- Discretization in squares or cubes (cells)
- 9 numbers per cell (or 19 in 3D)
  - = number of particles traveling towards neighboring cells
- Repeat (many times)
  - stream
  - collide
The stream step

Move particle (numbers) into neighboring cells
The collide step

Compute new particle numbers according to the collisions
Advantages/Disadvantages of the (LBM)

- Simple algorithm
- Easy to parallelize
- Easy to adapt, e.g. for
  - complicated geometries
  - time-varying geometries
  - free surfaces
  - additional physical or chemical effects
- Compute-Intensive
  - Some extensions (stretched grids) not available
  - Adaptivity difficult (must be done in space and time)
  - Low Mach number (<0.1)
  - Few commercial software available (yet)
Algorithm proceeds in two steps:

- **Stream**: advect fluid elements (copy DFs to neighbors)
- **Collide**: compute collisions of fluid molecules
Fluid Cell Treatment

Algorithm proceeds in two steps:

- Stream: advect fluid elements (copy DFs to neighbors)
- Collide: compute collisions of fluid molecules
Fluid Cell Treatment

Algorithm proceeds in two steps:

- Stream: advect fluid elements (copy DFs to neighbors)
- Collide: compute collisions of fluid molecules
Fluid Cell Treatment

Algorithm proceeds in two steps:

- **Stream**: advect fluid elements (copy DFs to neighbors)
- **Collide**: compute collisions of fluid molecules
Fluid Cell Treatment

Algorithm proceeds in two steps:

- Stream: advect fluid elements (copy DFs to neighbors)
- Collide: compute collisions of fluid molecules
From Lattice Gas to \textit{Lattice Boltzmann} I

Both Lattice gas and Lattice Boltzmann use models with

- a discrete time step $\Delta t$
- a discrete lattice of points
- a fixed number of directions/ lattice velocities
What makes Lattice Boltzmann different from Lattice Gas

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 travelling at a lattice point in any one of the lattice directions. These numbers are called *distribution functions*. Mathematically they are *expected values* of the statistical distribution functions for the lattice point.
Lattice Boltzmann Models

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

Density: \( \rho(x, t) = \sum_{i=0}^{8} f_i(x, t) \)

Momentum density: \( j(x, t) = \sum_{i=0}^{8} c_i f_i(x, t) \)
LBM Demonstration
(Java applet)

start-applet
How to simulate collisions

Collisions with boundary: One can (for example) use the bounce back condition, where the velocity of particles which hit the wall are simply reversed. This represents the no-slip boundary conditions in fluid dynamics.

A detailed model of particle-particle collisions would be very complicated. Practical Lattice Boltzmann methods often use the so-called single time relaxation approximation.
Single Time Relaxation

Assume that:

\( f_i^{eq}(x, t) \) is the equilibrium distribution for the particles at node \((x, t)\).

\( f_i^{eq}(x, t) \) is a local equilibrium: The number of particles entering from each direction is exactly equal to the number of particles leaving in this direction.

The equation for updating the particle distributions now becomes

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

The second term is the simplified collision operator.
Relaxation Constants

Here $1/\tau$ is the *relaxation constant*, the rate of change toward the equilibrium.

$\tau$: relaxation time.

$\tau = 1$: The distribution functions are exactly set to the equilibrium distribution.

$\tau = 2$: The distribution functions are midway between incoming distribution and the equilibrium distribution.

etc.

In realistic simulations, outside forces must still be included (e.g. gravity).
Lattice Boltzmann Algorithm

1. Initialize $f_i(x, t) := f_{i}^{eq}(x, t)$ from given $\rho(x, t)$ and $j(x, t)$.

2. Apply single time relaxation: $t \rightarrow t + 1$.

3. Compute new $\rho(x, t)$ and $j(x, t)$.

4. Compute new $f_{i}^{eq}(x, t)$ and continue with step 2.
Boltzmann Equation

\[ \frac{\partial f}{\partial t} + \langle u, \nabla f \rangle = Q \]

collision operator \( Q \)

\[ Q = -\frac{1}{\tau} \left( f(x, t) - f^{(0)}(x, t) \right) \]

\( \tau \) is the relaxation time

\( f^{(0)}(x, t) \) equilibrium distribution function
Discretizing the Boltzmann Equation

$$\frac{\partial f}{\partial t} + \langle u, \nabla f \rangle = -\frac{1}{\tau} \left( f - f^{(0)} \right)$$

finite set \( \{v_i\}, 0 \leq i \leq n, \) of velocities

$$f_i(x, t) = f(x, v_i, t)$$

**discrete Boltzmann equation**

$$\frac{\partial f_i}{\partial t} + \langle v_i, \nabla f_i \rangle = -\frac{1}{\tau} \left( f_i - f_i^{(0)} \right)$$

$$F_i(x + c_i \Delta t, t + \Delta t) - F_i(x, t) = -\frac{1}{\tau} \left( F_i(x, t) - F_i^{(0)}(x, t) \right)$$
Macroscopic quantities

mass density \( \rho(x, t) \)

\[
\rho(x, t) = \sum_i F_i(x, t)
\]

momentum density \( j(x, t) \)

\[
j(x, t) = \rho(x, t) u(x, t) = \sum_i c_i F_i(x, t)
\]
LBM equilibrium distribution functions (2D)

Stream/Collide:

\[ F_i(x + c_i \Delta t, t + \Delta t) - F_i(x, t) = -\frac{1}{\tau} \left( F_i(x, t) - F_i^{(0)}(x, t) \right) \]

Equilibrium DF:

\[ F_i^{(0)}(x, t) = \frac{4}{9} \rho(x, t) \left( 1 - \frac{3}{2} \frac{\langle u(x, t), u(x, t) \rangle}{c^2} \right) \]

for \( i = C \)

\[ F_i^{(0)}(x, t) = \frac{1}{9} \rho(x, t) \left( 1 + 3 \frac{\langle c_i, u(x, t) \rangle}{c^2} + \frac{9}{2} \frac{\langle c_i, u(x, t) \rangle^2}{c^4} - \frac{3}{2} \frac{\langle u(x, t), u(x, t) \rangle}{c^2} \right) \]

for \( i \in \{N, E, S, W\} \)

\[ F_i^{(0)}(x, t) = \frac{1}{36} \rho(x, t) \left( 1 + 3 \frac{\langle c_i, u(x, t) \rangle}{c^2} + \frac{9}{2} \frac{\langle c_i, u(x, t) \rangle^2}{c^4} - \frac{3}{2} \frac{\langle u(x, t), u(x, t) \rangle}{c^2} \right) \]

for \( i \in \{NE, SE, NW, SW\} \)
LBM in Equations (3D)

Stream/Collide:

\[
F_i(x + c_i \Delta t, t + \Delta t) - F_i(x, t) = -\frac{1}{\tau} \left( F_i(x, t) - F_i^{(0)}(x, t) \right)
\]

Equilibrium DF:

\[
F_i^{(0)}(x, t) = \frac{1}{3} \rho(x, t) \left( 1 - \frac{3}{2} \frac{\langle \mathbf{u}(x, t), \mathbf{u}(x, t) \rangle}{c^2} \right)
\]

for \( i = \text{C}, \text{N}, \text{E}, \text{S}, \text{W}, \text{T}, \text{B} \)

\[
F_i^{(0)}(x, t) = \frac{1}{18} \rho(x, t) \left( 1 + 3 \frac{\langle c_i, \mathbf{u}(x, t) \rangle}{c^2} + \frac{9}{2} \frac{\langle c_i, \mathbf{u}(x, t) \rangle^2}{c^4} - \frac{3}{2} \frac{\langle \mathbf{u}(x, t), \mathbf{u}(x, t) \rangle}{c^2} \right)
\]

for \( i \in \{\text{TN}, \text{TS}, \text{BN}, \text{BS}, \text{TE}, \text{TW}, \text{BE}, \text{BW}, \text{NE}, \text{NW}, \text{SE}, \text{SW}\} \)
WaLBerla
LBM Software Framework
waLBerla

Widely applicable lattice Boltzmann from Erlangen

- CFD SW framework based on lattice Boltzmann method
- Modular software concept
  - Supports various applications:
    - Blood flow in aneurysms
    - Moving particles and agglomerates
    - Free surfaces to simulate foams, fuel cells, a.m.m.
    - Charged colloids
    - Arbitrary combinations of above
  - Integration in efficient massively-parallel environment

- Patch concept enables
  - Extendability: new functionality
  - Parallelization
  - Load Balancing
  - Accelerators!
A Patch Based Data Structure

- Obstacles are marked in „flag fields“
- Domain partitioning in equally sized blocks, so-called **patches**
- Only allocate patches containing (some) fluid cells
- Memory requirements and the computational time reduced significantly
- Patch size adapted to processor architecture and problem.
  - Typical patch size: $16^3$ or $32^3$.
  - But e.g. for the Cell processor we use patches of size $8x8x8$, fitting into the SPU local memory
LBM Extensions and Applications
Turbulence Modelling and Non-Newtonian Flows

- Smagorinsky Subgrid Model:
  - Similar to approach for NS-Solvers
  - Model subgrid-scale vortices by locally changing the viscosity
  - Suitable for LES

- Implementation for LBM
  - Reynolds stress tensor computed for each cell
  - Changes only in collision operator
  - Ca. 20% additional cost
Non-Newtonian Flows for Computational Hemodynamics
Results

Velocity near the wall in an aneurysm

Oscillatory shear stress near the wall in an aneurysm
Pulsating Blood Flow in an Aneurysm

Collaboration between:
Neuro-Radiology
(Prof. Dörfler, Dr. Richter)
Computer Science
Simulation
Imaging
CFD

Data Set
Simulation of Clotting Processes Using non-Newtonian Blood Models

- Motivation
- Knowledge about the clotting of blood is important for many medical applications:
  - Diagnostics
  - Surgery
Non-Newtonian blood model: The Casson model

\[ \sqrt{\eta |\dot{\gamma}|} = \begin{cases} \sqrt{\sigma} - \sqrt{\sigma_y} & \sigma > \sigma_y \\ 0 & \text{else} \end{cases} \]
Cell LBM Simulations

Goal: demanding (flow) simulations at moderate cost but very fast, e.g. for simulation of blood-flow in an aneurysm for therapy and surgery planning

Available cell systems:
- Roadrunner
- Blades
- Playstation 3
Cell Architecture: 9 cores on a chip

- **Broadband Engine Interface**
  - coherent connection to 2nd CPU
  - IO devices

- **Element Interconnect Bus**
  - 4 rings
  - up to 204.8 GB/s¹

- **Power Processor Unit**
  - simplified PowerPC
  - similar to PPC970
  - 2x SMT, but in order

- **Memory Interface Controller**
  - Rambus eXtreme Data Rate
  - 2 channels
  - 25.6 GB/s¹ bandwidth

- **Synergistic Processor Unit**
  - DMA transfer into and out of LS
  - channel interface
  - "interface controller" of the SPU

- **Synergistic Processor Element**
  - 128 SIMD registers (128 bit)
  - two SIMD pipelines
  - 204.8 GFlop/s¹ max. for 8 SPEs

- **Local Store**
  - "main memory" of SPE
  - 256 kB only
  - 16 B/s bandwidth
  - 128 B/s if all 8 banks accessed concurrently

¹ @3.2GHz
Performance Results

LBM performance on a single core (8x8x8 channel flow)

- Xeon 5160: 10.4
- PPE: 4.8
- SPE*: 2.0

*on Local Store without DMA transfers

straight-forward C code
SIMD-optimized assembly
LBM Optimized for Cell

- memory layout
  - optimized for DMA transfers
  - information propagating between patches is reordered on the SPE and stored sequentially in memory for simple and fast exchange

- code optimization
  - kernels hand-optimized in assembly language
  - SIMD-vectorized streaming and collision
  - branch-free handling of bounce-back boundary conditions
Performance Results

- 1: 42
- 2: 81
- 3: 93
- 4: 94
- 5: 94
- 6: 95
Massively Parallel Particulate Flows

Fluid-Structure Interaction with Moving Objects
Parallelization of Particle-laden Flows

<table>
<thead>
<tr>
<th>Module</th>
<th>% of compute time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object mapping</td>
<td>13.0</td>
</tr>
<tr>
<td>LBM solver</td>
<td>53.2</td>
</tr>
<tr>
<td>Force calculation</td>
<td>15.6</td>
</tr>
<tr>
<td>PE solver</td>
<td>1.1</td>
</tr>
<tr>
<td>LBM communication</td>
<td>3.2</td>
</tr>
<tr>
<td>PE communication</td>
<td>14.0</td>
</tr>
</tbody>
</table>

xsize: 540, ysize: 500, zsize: 500
135 x 10^6 lattice cells
129 processes/cores
2,500 objects
27,000 time steps
12:54h
Mapping Moving Obstacles into the LBM Fluid Grid

(a) Initial setup: The velocities $u$ of the object cells $x_b$ are set to the velocity $u_w(x_b)$ of the object. In this example the object only has a translational velocity component. Fluid cells are marked with $x_f$.

(b) Updated setup: Two fluid cells have to be transformed to object cells and for two object cells the pdfs have to be reconstructed.

Figure 1: 2D mapping example.
Fluid-Structure Interaction

- Collision detection
- (frictional) collision response
- Time integration

Rigid bodies act as obstacles

Fluid results in external forces

- Update of fluid nodes: stream/collide
- Calculation of hydrodynamic forces (momentum exchange)
Algorithm 2 Coupled LBM-PE solver

1: **MPI communicate ghost layer of velocity and density**
2: **for each body B do**
3:   **Map B to lattice grid**
4: **end for**
5: **MPI communicate ghost layer of PDFs**
6: **for each lattice cell do**
7:   **Stream and collide**
8: **end for**
9: **for each surface cell do**
10:   **Add forces from fluid to rigid objects**
11: **end for**
12: **Time step in the rigid body simulation**
Fluidized Beds
Virtual Fluidized Bed

- 512 processors of HLRB2:
- Size of Simulation Domain
  400x400x480 cells of LBM
- number of rigid objects: 25,000
- number of time steps: 252,000
- Run time:
  30h 4min
- corresponds to 15,000 core hours.
Virtual Fluidized Bed

512 Processors
HLRB-II

Simulation Domain
Size: 180x198x360 cells of LBM

900 capsules and 1008 spheres = 1908 objects

Number time steps: 252,000

Run Time:
Segregation simulation of 12,013 objects with two different shapes in different time steps simulated on 2,048 cores in a box. Density values of 0.8 kg/dm$^3$ and 1.2 kg/dm$^3$ are used for the objects in water with density 1 kg/dm$^3$ and a gravitation field. Lighter particles are rising to the top of the box, while heavier particle sink to the bottom.
Weak Scaling

![Graph showing weak scaling efficiency for different numbers of cores.]

**Jugene**

**Blue Gene/P**

**Jülich Supercomputer Center**

- Scaling 64 to 294 912 cores
- 150 994 944 000 lattice cells
- Sparsely packed particles
- 83 804 982 rigid spherical objects

Largest simulation to date:
- 8 Trillion (10^{12}) variables per time step (LBM alone)
- 50 TByte

**Friedrich-Alexander-Universität Erlangen-Nürnberg**

**Lehrstuhl für Informatik 10 (Systemsimulation)**
Lattice Boltzmann: \( f_i(x+\Delta x_i, t+\Delta t) = f_i(x, t) + \Delta_i(f) \)

Nanoscales: High impact of thermal fluctuations:
- Stochastic model for collision operator
\[
\Delta_i(f) = \tilde{\Delta}_i(f) + R_i
\]
  - Regular treatment of (mean) collision behaviour
  - Brownian random structures

Random structures:
- Modeled by Gaussian pseudo-random numbers in momentum space
- Variance dependent on fluid properties (T, \( \nu \))

Method proposed by Dünweg et.al. and Ladd et.al.
Fluctuating Lattice Boltzmann for Microfluidics: Brownian Motion

- cell size: 1nm
- Domain size: $100^3$ nm
- $100^3$ cells
- time step $10^{-13}$ sec
- 100,000 time steps ≈ 1 ns
Lattice Boltzmann Methods

Free Surface Flow Simulation

for foams, fuel cells, food processing, etc.
The interface between liquid and gas

- Volume-of-Fluids like approach
- Flag field: Compute only in fluid
- Special “free surface” conditions in interface cells
Problem:
*Missing distribution functions* at interface cells after streaming!

Reconstruction such that *macroscopic boundary conditions* are satisfied.

Rising Bubbles
Free Surface Treatment

Treatment of Interface Cell
Free Surface Treatment

Treatment of Interface Cell

Calculate Mass Exchange
Free Surface Treatment

- Treatment of Interface Cell
- Calculate Mass Exchange
- Stream from Fluid & Interface Cells
Free Surface Treatment

- Treatment of Interface Cell
- Calculate Mass Exchange
- Stream from Fluid & Interface Cells
- Reconstruct DFs from Empty Cells
Free Surface Treatment

1. Treatment of Interface Cell
2. Calculate Mass Exchange
3. Stream from Fluid & Interface Cells
4. Reconstruct DFs from Empty Cells

Calculate Surface Normal $n$
Free Surface Treatment

1. Treatment of Interface Cell
2. Calculate Mass Exchange
3. Stream from Fluid & Interface Cells
4. Reconstruct DFs from Empty Cells

5. Calculate Surface Normal $n$
6. Reconstruct DFs along Normal
Free Surface Treatment

- Treatment of Interface Cell
- Calculate Mass Exchange
- Stream from Fluid & Interface Cells
- Reconstruct DFs from Empty Cells

- Calculate Surface Normal $n$
- Reconstruct DFs along Normal
- Perform Collision Step
Free Surface Treatment

1. Treatment of Interface Cell
2. Calculate Mass Exchange
3. Stream from Fluid & Interface Cells
4. Reconstruct DFs from Empty Cells

5. Calculate Surface Normal $n$
6. Reconstruct DFs along Normal
7. Perform Collision Step
8. Store DFs and continue with next Cell
Free Surface Cell Conversions

- Emptied interface cell $\rightarrow$ gas
- Filled interface cell $\rightarrow$ fluid
- Guarantee closed layer of interface cells
- Redistribute mass in the neighborhood
Free surface simulations

Algorithmic Overview:

- Before stream step, compute mass exchange across cell boundaries for interface cells
- Calculate bubble volumes and pressure
- Surface curvature for surface tension
- Change topology if interface cells become full or empty – keep layer of interface cells closed
Curvature calculation (version I)

Alternative approaches:
- Integrate normals over surface (weighted triangles)
- Level set methods (track surface as implicit function)
Surface Tension (Vers. 2)

Marching-cube surface triangulation
Compute a curvature for each triangle

\[ \kappa = \frac{1}{2} \frac{\delta A}{\delta V} \]

- Associate with each LBM cell the average curvature of its triangles
- Complicated
- Beats level sets for our applications (mass conservation)
**VOF vs. Level-Set Methods**

**Volume of Fluid**
- Mass conservation
- Efficiency
- Good integration for LBM

**Level Sets**
- Smooth representation
- Accurate & efficient curvature calculation
**Numerical Experiment: Single Rising Bubble**

- Comparison to (rotationally symmetric) 2D level-set volume-of-fluid method and experimental results; T. Pohl, D. Gerlach, F. Durst (Erlangen), G. Biswas (IIT Kanpur)
- Modified parameter: surface tension
Simulation of Metal Foams

Example application:
- Engineering: metal foam simulations

Based on LBM:
- Free surfaces
- Surface tension
- Disjoining pressure to stabilize thin liquid films
- Parallelization with MPI and load balancing

Collaboration with C. Körner (Dept. of Material Sciences, Erlangen)

Other applications:
- Food processing
- Fuel cells
Large-Scale Application: 3000 Bubbles

- Simulation
- 3000 Bubbles
- $910 \times 910 \times 935 = 7.7 \times 10^8$ lattice cells
- 7,000 time steps
- 423 GB
- 512 processes
- 48 hours
- 24,576 core hours

First time step

After 5,000 time steps

Images of small subset of domain

Difficulties: Topology changes due to bubble coalescence (OK) or bubble break up (not implemented)
Efficient parallelization!

Render time for this image:
192 hours on single CPU
Larger-Scale Computation: 1000 Bubbles

Simulation
1000 Bubbles
510x510x530 = 1.4 \cdot 10^8 lattice cells
70,000 time steps
77 GB
64 processes
72 hours
4,608 core hours

Visualization
770 images
Approx. 12,000 core hours for rendering

Best Paper Award for Stefan Donath (LSS Erlangen) at ParCFD, May 2009 (Moffett Field, USA)
Parallel Efficiency

Woodcrest cluster
moderate domain size per core
(strong: 6.5 GB, weak: 1 GB/core)

HLRB2, SGI Altix 4700
Full domain size (3.5 GB/core)
Largest domain size:
$6.1 \cdot 10^{10}$ cells $\approx 31$ TB Mem
EU-Project DECODE
Fuel Cells

Collaboration with K. Mecke (& Opel, Volvo, DLR,...)
Consequences of a loss of hydrophobicity in gas diffusion layer and reaction layer
- Water management changes with hydrophobicity
- As a consequence, electrochemical performance decreases
- After drying, performance is recovered but then accelerated decrease during operation is observed
Simulation of fuel and water transport in Gas Diffusion Medium of Polymer-Electrolyte Fuel Cells

Geometry: 0.45 · 0.45 · 0.1 mm³
High resolution needed due to high surface tension (water)
Typical simulation: $2 \cdot 10^{11}$ lattice cells ($\approx 10$ TB memory)
Flow Simulation

Visualization and Animation
Parallel “Tsunami”-Simulation
Example Coupled Simulations
Massively parallel simulations with many particles (LSS cluster)

- xsize: 540
- ysize: 240
- zsize: 400
- 52 x 10^6 lattice cells
- 31 processes/cores
- on LSS cluster
- 750 objects
- 10,000 timesteps
- 9:45h
Simulations with Fluid Control
Part IV

Conclusions
The Two Principles of Science

Theory
- Mathematical Models, Differential Equations, Newton

Experiments
- Observation and prototypes
- empirical Sciences

Computational Science
- Simulation, Optimization
- (quantitative) virtual Reality
Multiscale modelling and simulation

Key challenges
- Structure-property functions
- Process-structure functions

Processes & Applications
- FEM, CFD, PBE, LB
- MD, MC
- SD/BD, DEM

Complex Structures

From molecules to functions

Length scale

Time scale

Hierarchical structure design
Acknowledgements

**Collaborators**
- In Erlangen: WTM, LSE, LSTM, LGDV, RRZE, LME, Neurozentrum, Radiologie, etc.
- Especially for foams: C. Körner (WTM)
- International: Utah, Technion, Constanta, Ghent, Boulder, München, Zürich, Delhi, ...

**Dissertationen Projects**
- N. Thürey, T. Pohl, S. Donath, S. Bogner (LBM, free surfaces, 2-phase flows)
- M. Kowarschik, J. Treibig, M. Stürmer, J. Habich (architecture aware algorithms)
- K. Iglberger, T. Preclik, K. Pickel (rigid body dynamics)
- J. Götz, C. Feichtinger (Massively parallel LBM software, suspensions)
- C. Mihoubi, D. Bartuschat (Complex geometries, parallel LBM)

**(Long Term) Guests in summer/fall 2009:**
- Dr. S. Ganguly, IIT Kharagpur (Humboldt) - Electroosmotic Flows
- Prof. V. Buwa, IIT Delhi (Humboldt) - Gas-Fluid-Solid flows
- Felipe Aristizabal, McGill Univ., Canada (LBM with Brownian Motion)
- Prof. Popa, Constanta, Romania (DAAD) Numerical Linear Algebra
- Prof. N. Zakaria, Universiti Petronas, Malaysia
- Prof. Hanke, KTH Stockholm (DAAD), Mathematical Modelling
- several Indian student interns

~25 Diplom- /Master- Thesis, ~30 Bachelor Thesis

Funding by KONWIHR, DFG, BMBF, EU, Elitenetzwerk Bayern
Thanks for your attention!

Questions?

Slides, reports, thesis, animations available for download at:
www10.informatik.uni-erlangen.de