Massively Parallel Algorithms for Fluid-Structure Interaction with Moving Objects


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

in collaboration with many more

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

Herrsching, October 13, 2009

Workshop on Computational Engineering

Fluid-Structure Interaction:
Modelling, Simulation, Optimisation
Overview

- Complex Flow Simulation with Lattice Boltzmann Methods
  - The LBM
  - Multibody Dynamics
  - Fluid-Structure Interaction with Moving Objects
  - Parallel Performance Results
- Conclusions
Computational Fluid Dynamics

Lattice Boltzmann Method
The Lattice-Boltzmann-Method

- Discretization in 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

distribution functions after streaming

velocity

distribution functions after collision

local equilibrium distribution functions

tau weighting with $\tau=1.5$
The interface between liquid and gas

- Volume-of-Fluids like approach
- Flag field: Compute only in fluid
- Special “free surface” conditions in interface cells
- Reconstruct average curvature for surface tension
- Contact angles
Large-Scale Application: 1000 Bubbles

Simulation
- 1000 Bubbles
- \(510 \times 510 \times 530 = 1.4 \times 10^8\) lattice cells
- 70,000 time steps
- 77 GB
- 64 processes
- 72 hours
- 4,608 core hours

Visualization
- 360 images
- 8 processes for
- 4 hours per image
- 11,520 core hours

Best Paper Award for Stefan Donath at ParCFD, May 2009 (Moffett Field, USA)
Rigid Multi Body Dynamics
Rigid body dynamics with friction and objects of more general shape
(T. Preclik, K. Iglberger)
Collisions & Contacts between Rigid Objects
## Scale-Up for Parallel Rigid Body Dynamics

<table>
<thead>
<tr>
<th># Cores</th>
<th># Spheres</th>
<th>Partitioning</th>
<th>Runtime [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>1 000 000</td>
<td>64 x 1 x 1</td>
<td>267.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 8 x 1</td>
<td>394.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 x 4 x 4</td>
<td>547.4</td>
</tr>
<tr>
<td>128</td>
<td>2 000 000</td>
<td>128 x 1 x 1</td>
<td>266.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 8 x 1</td>
<td>392.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 4 x 4</td>
<td>545.3</td>
</tr>
<tr>
<td>256</td>
<td>4 000 000</td>
<td>256 x 1 x 1</td>
<td>269.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 16 x 1</td>
<td>393.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 8 x 4</td>
<td>547.5</td>
</tr>
<tr>
<td>448</td>
<td>7 000 000</td>
<td>448 x 1 x 1</td>
<td>265.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28 x 16 x 1</td>
<td>391.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x 8 x 7</td>
<td>565.8</td>
</tr>
<tr>
<td>960</td>
<td>15 000 000</td>
<td>960 x 1 x 1</td>
<td>273.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 x 30 x 1</td>
<td>402.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 x 8 x 8</td>
<td>560.4</td>
</tr>
<tr>
<td>1920</td>
<td>30 000 000</td>
<td>1920 x 1 x 1</td>
<td>268.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60 x 32 x 1</td>
<td>415.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 15 x 8</td>
<td>585.5</td>
</tr>
<tr>
<td>3840</td>
<td>60 000 000</td>
<td>3840 x 1 x 1</td>
<td>267.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64 x 60 x 1</td>
<td>432.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 x 16 x 15</td>
<td>611.5</td>
</tr>
<tr>
<td>7680</td>
<td>120 000 000</td>
<td>7680 x 1 x 1</td>
<td>263.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>120 x 64 x 1</td>
<td>432.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 x 16 x 15</td>
<td>599.0</td>
</tr>
<tr>
<td>9120</td>
<td>142 500 000</td>
<td>9120 x 1 x 1</td>
<td>264.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96 x 95 x 1</td>
<td>432.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24 x 20 x 19</td>
<td>600.3</td>
</tr>
</tbody>
</table>

Table 2: Weak scaling experiment of 1 000 time steps with up to 142 500 000 spheres. Each process initially owns a block of 25 \( \times \) 25 \( \times \) 25 spheres. The total volume of the spheres occupies approximately 1% of the domain volume.
Massively Parallel Particulate Flows
Fluidized Beds
Mapping Moving Obstacles into the LBM Fluid Grid

(a) Initial setup: The velocities $\mathbf{u}$ of the object cells $\mathbf{x}_b$ are set to the velocity $\mathbf{u}_w(\mathbf{x}_b)$ of the object. In this example the object only has a translational velocity component. Fluid cells are marked with $\mathbf{x}_f$.

(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.
Parallelization of Flow with many Particles

<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>
Coupling the physics engine to the Lattice Boltzmann solver

Example

Timeloop
1. Set boundary conditions
2. LBM stream collide
3. Add forces from fluid to obstacles
4. Move and collide obstacles
   - send to instance
   else
   - move and collide locally
5. Move and collide obstacles on instance
6. Send values back
7. Receive and update values from instance
8. If obstacle getting near border, then send to neighbor
Parallel Performance of Fluid-Structure Interaction with Multibody Dynamics

Largest simulation to date: 622 Billion unknowns per time step (LBM alone) 12 TByte

**Figure 4:** Performance of simulations with rigid bodies incorporated in the fluid. The number of lattice cells per processor core is 4 million, spheres are used with a diameter of 6 lattice cells and a average spacing of 12 lattice cells between the spheres. For 8,192 cores more than 37 million objects are simulated. Due to the architecture of the HLRB-2, 4 cores per memory channel are used instead of 2 when running on more than 7,900 cores, which reduces the performance.
waLBerla

Widely applicable lattice Boltzmann from Erlangen

- CFD project based on lattice Boltzmann method
- Modular software concept
  - Supports various applications, currently planned:
    - 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
  - Combining Functionality
  - Parallelization
  - Load Balancing
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 (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
- 3 Indian student interns

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

Funding by KONWIHR, DFG, BMBF, EU, Elitenetzwerk Bayern
Conclusions

Parallel Supercomputer Performance is Easy!

- If parallel efficiency is bad, choose a slower serial algorithm
  - it is probably easier to parallelize
  - and will make your speedups look much more impressive

- Introduce the “CrunchMe” variable for getting high Flops rates
  - advanced method: disguise CrunchMe by using an inefficient (but compute-intensive!) algorithm from the start

- Introduce the “HitMe” variable to get good cache hit rates
  - advanced version: Derive HitMe from the “Transgalactic Hash-Brown Lookaside Table in the Multi-Threatened Cash-Filling Spock-Tree” Class
  - ... impressing yourself and others

- Never cite “time-to-solution”
  - who cares whether you solve a real-life problem anyway
  - it is the MachoFlops that interest the people who pay for your research

- Never waste your time by trying to use a complicated algorithm in parallel
  - Use Primitive Algorithm => Easy to Maximize your MachoFlops
  - A few million CPU hours can easily save you days of reading in boring math books
Thanks for your attention!

Questions?

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