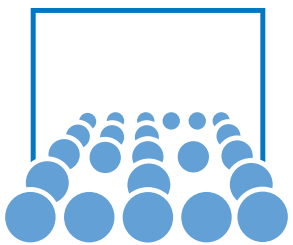
Parallel Particle Simulation in Multiscale Fluid Representations

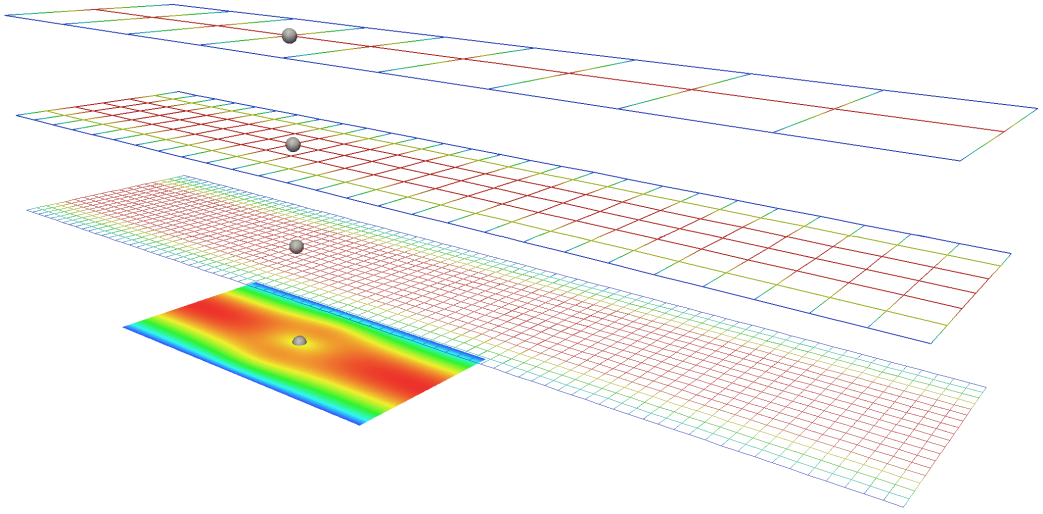
SuperMUC Review Workshop

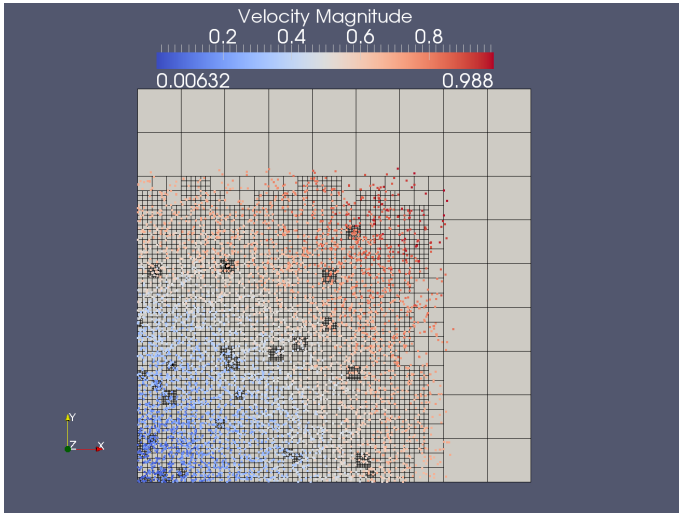
Philipp Neumann, Tobias Weinzierl, Denis Jarema, Bart Verleye

8 July 2014,
neumanph@in.tum.de

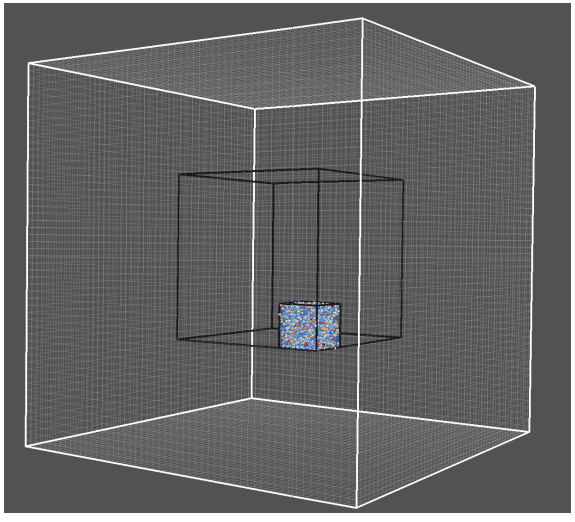


Motivation





1. Multi-level fluid-particle treatment
2. Many-particle systems: Spacetree algorithms
3. Outlook: Walking down the scales



Lattice Boltzmann–Navier-Stokes: Models

Navier-Stokes

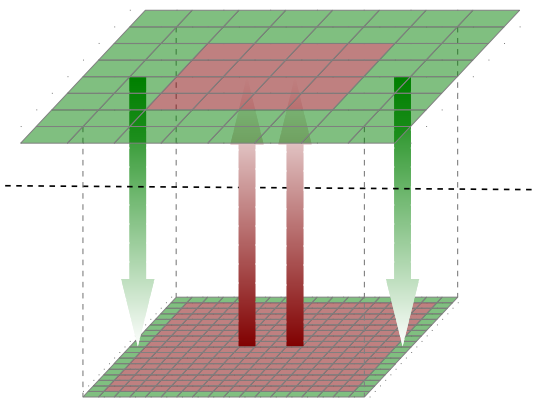
Mesh-based

Fluid in equilibrium

DoF: velocities $\mathbf{u}$, pressure p

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{1}{\mathrm{Re}} \Delta \mathbf{u} \end{aligned}$$

Navier-Stokes



Lattice Boltzmann

Lattice Boltzmann

Mesh-based

Fluid close to equilibrium

DoF: distribution functions f_i

$$f_i(\mathbf{x} + \mathbf{c}_i dt, t + dt) = f_i(\mathbf{x}, t) + \Delta_i(f - f^{eq})$$

$$\begin{aligned} \rho(\mathbf{x}, t) &= \sum_{i=1}^{Q} f_i \\ p(\mathbf{x}, t) &= \rho(\mathbf{x}, t)/c_s^2 \\ \mathbf{u}(\mathbf{x}, t) &= \frac{1}{\rho(\mathbf{x},t)} \sum_{i=1}^{Q} f_i \mathbf{c}_i \end{aligned}$$

From Navier-Stokes to Lattice Boltzmann

- Given: ρ, $\mathbf{u}$, $\partial_{x_\beta} \mathbf{u}_\alpha + \partial_{x_\alpha} \mathbf{u}_\beta$
- Generally, it holds

$$Q > \underbrace{1}_{\text{mass}} + \underbrace{D}_{\text{momentum}} + \underbrace{D(D+1)/2}_{\text{stresses}}$$

From Navier-Stokes to Lattice Boltzmann

- Given: ρ, $\mathbf{u}$, $\partial_{x_\beta}\mathbf{u}_\alpha + \partial_{x_\alpha}\mathbf{u}_\beta$
- Generally, it holds

$$Q > \underbrace{1}_{\text{mass}} + \underbrace{D}_{\text{momentum}} + \underbrace{D(D+1)/2}_{\text{stresses}}$$

- Split $f_i = f_i^{eq}(\rho, \mathbf{u}) + f_i^{neq}$
- Ansatz[1]: continuum $\leftrightarrow$ non-equ. parts as small as possible

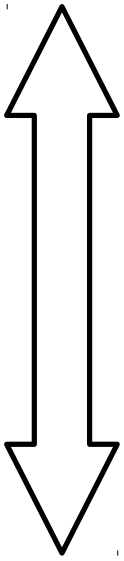
$$\min_{f^{neq} \in \mathbb{R}^Q} g(f^{neq}) \quad \text{such that}$$

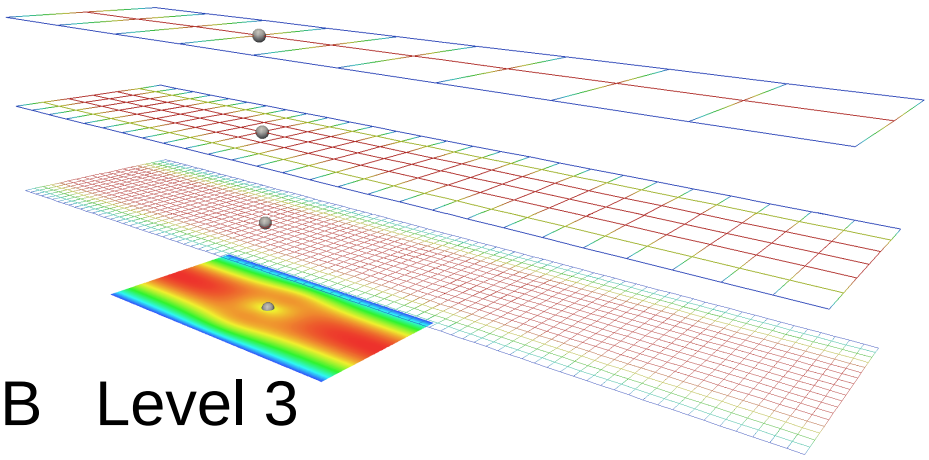
$$\begin{aligned}
\textstyle\sum_i f_i^{neq} &= 0 && \text{mass} \\
\textstyle\sum_i f_i^{neq} \mathbf{c}_{i_\alpha} &= 0 && \text{momentum} \\
\textstyle\sum_i f_i^{neq} \mathbf{c}_{i_\alpha} \mathbf{c}_{i_\beta} &= -c_s^2 \tau \left(\partial_{x_\beta}\mathbf{u}_\alpha + \partial_{x_\alpha}\mathbf{u}_\beta\right) && \text{stresses}
\end{aligned}$$

[1] P. Neumann, H.-J. Bungartz, M. Mehl, T. Neckel, and T. Weinzierl. Commun. Comput. Phys. 12(1):65–84, 2012.

Application: Hierarchical Particle Transport

Coarse-scale model: NS (fluid) + Faxen[2] (particle)





NS Level 0

NS Level 1

LB Level 3

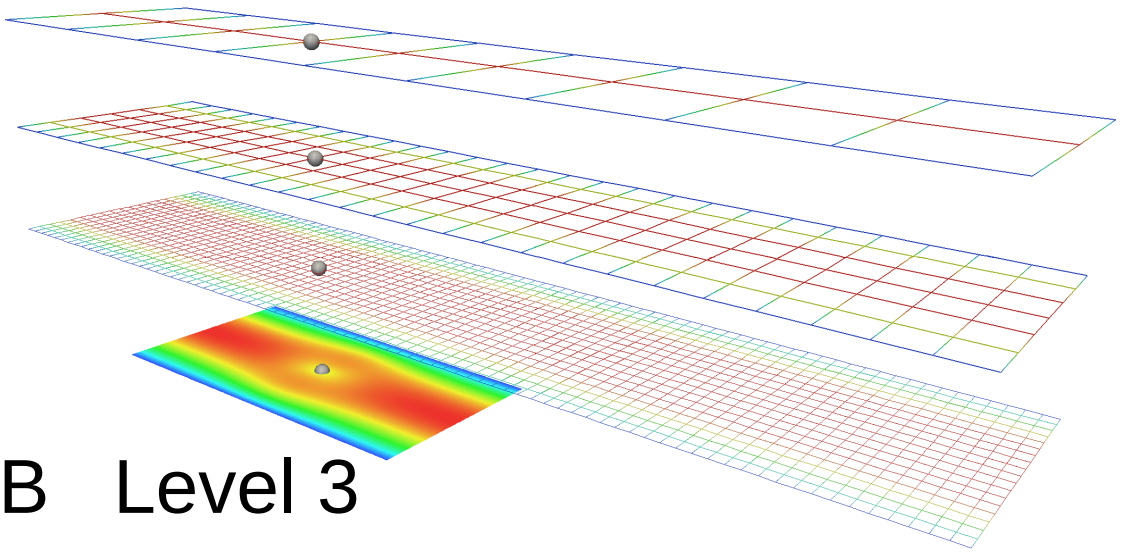
NS Level 2

Fine-scale model: LB + Two-way coupled particle solver[3]

[2] H. Faxen. Appelberg, 1921.

[3] A.J.C. Ladd. J. Fluid Mech., 271:285–339, 1994.

Particle Displacement in Channel Flow (1)

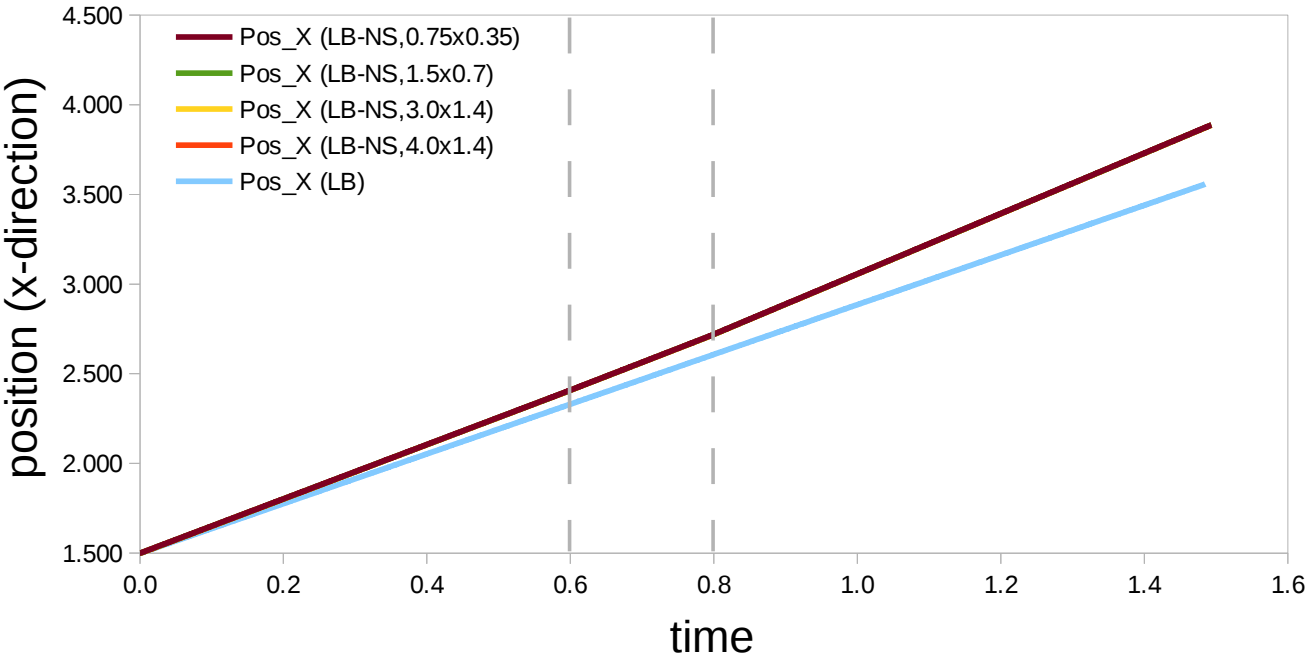


- Start with LB, Level 3
  Switch to
  - NS-Faxen, Level 2 at $t = 0.01$
  - NS-Faxen, Level 1 at $t = 0.60$
  - NS-Faxen, Level 0 at $t = 0.80$
- Compare to dynamically adaptive LB solution[4]

LB	LB–NS
$dx_F = 3.1\text{e-}3$ $dx_C = 2.8\text{e-}2$	$dx_F = 3.1\text{e-}3$ $dx_C = 5.0\text{e-}1$
$dt_{LB} = 4.8\text{e-}7$	$dt_{LB} = 3.2\text{e-}7$ $dt_{NS} = 3.0\text{e-}5$

[4] P. Neumann and T. Neckel. Computational Mechanics, 51(2):237–253, 2013.

Particle Displacement in Channel Flow (2)

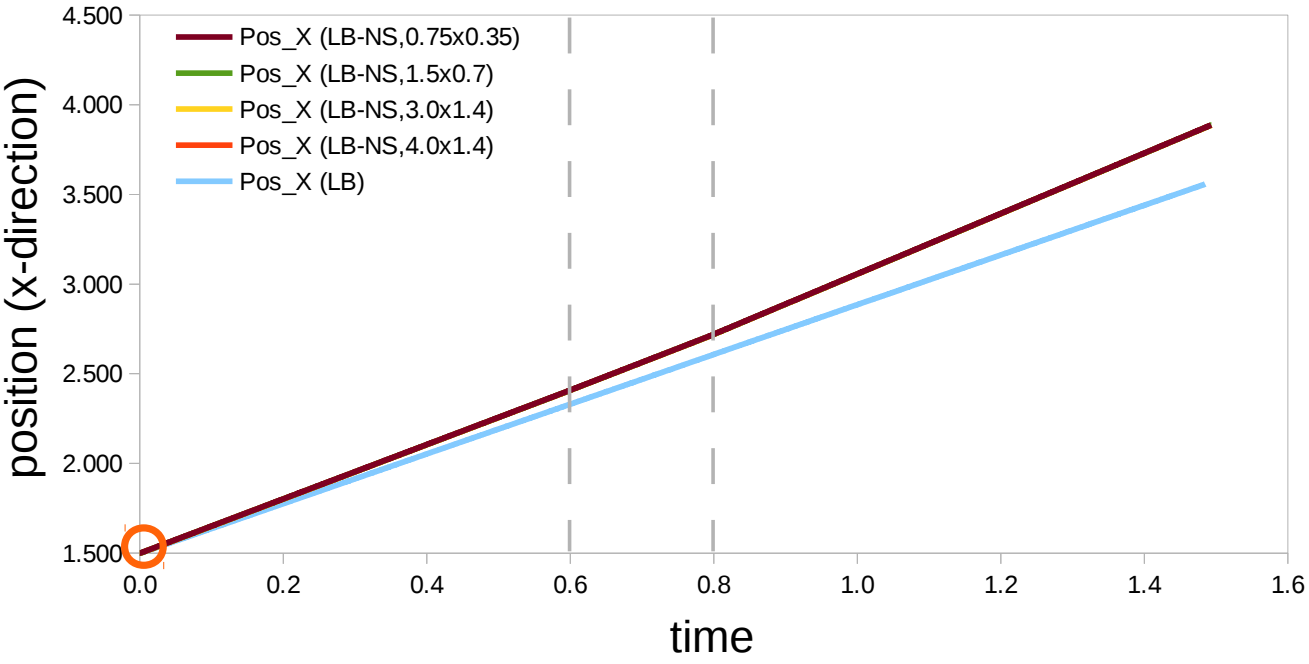


- Start with LB, Level 3
  Switch to
  - NS-Faxen, Level 2 at $t = 0.01$
  - NS-Faxen, Level 1 at $t = 0.60$
  - NS-Faxen, Level 0 at $t = 0.80$
- Compare to dynamically adaptive LB solution[4]

LB	LB–NS
$dx_F = 3.1\text{e-}3$ $dx_C = 2.8\text{e-}2$	$dx_F = 3.1\text{e-}3$ $dx_C = 5.0\text{e-}1$
$dt_{LB} = 4.8\text{e-}7$	$dt_{LB} = 3.2\text{e-}7$ $dt_{NS} = 3.0\text{e-}5$

[4] P. Neumann and T. Neckel. Computational Mechanics, 51(2):237–253, 2013.

Particle Displacement in Channel Flow (2)

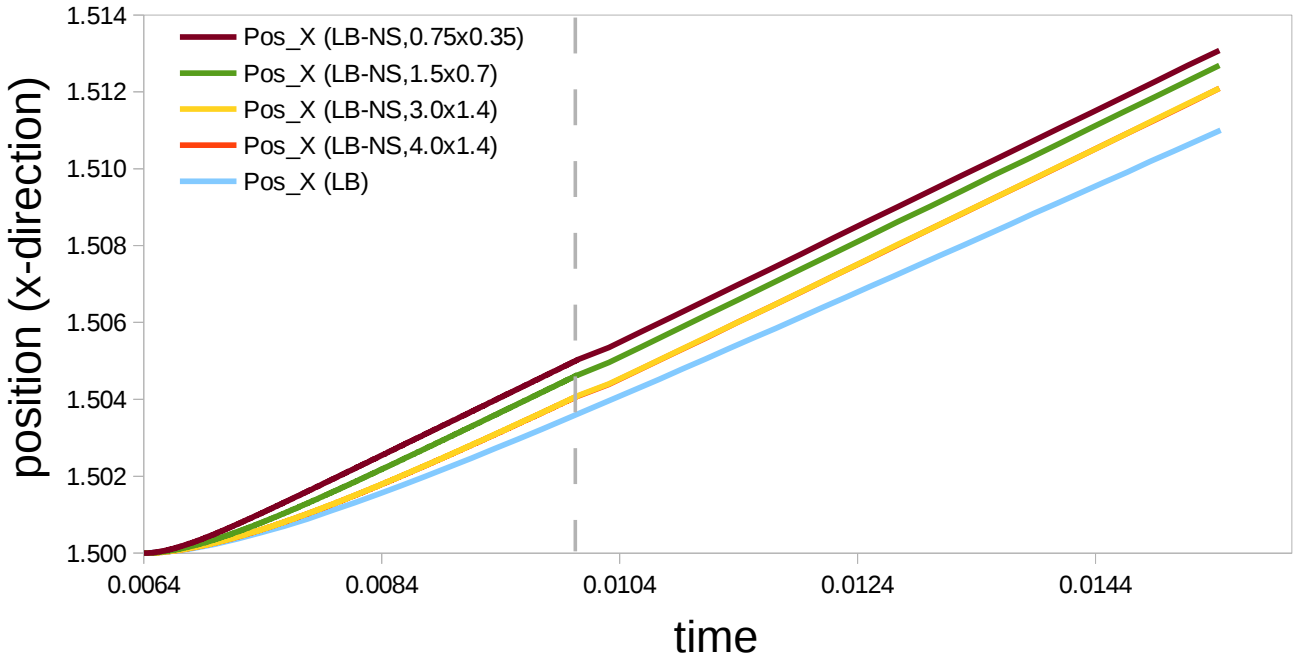


- Start with LB, Level 3
  Switch to
  - NS-Faxen, Level 2 at $t = 0.01$
  - NS-Faxen, Level 1 at $t = 0.60$
  - NS-Faxen, Level 0 at $t = 0.80$
- Compare to dynamically adaptive LB solution[4]

LB	LB–NS
$dx_F = 3.1\text{e-}3$ $dx_C = 2.8\text{e-}2$	$dx_F = 3.1\text{e-}3$ $dx_C = 5.0\text{e-}1$
$dt_{LB} = 4.8\text{e-}7$	$dt_{LB} = 3.2\text{e-}7$ $dt_{NS} = 3.0\text{e-}5$

[4] P. Neumann and T. Neckel. Computational Mechanics, 51(2):237–253, 2013.

Particle Displacement in Channel Flow (2)



- Start with LB, Level 3
  Switch to
  - NS-Faxen, Level 2 at $t = 0.01$
  - NS-Faxen, Level 1 at $t = 0.60$
  - NS-Faxen, Level 0 at $t = 0.80$
- Compare to dynamically adaptive LB solution[4]

LB	LB–NS
$dx_F = 3.1\text{e-}3$ $dx_C = 2.8\text{e-}2$	$dx_F = 3.1\text{e-}3$ $dx_C = 5.0\text{e-}1$
$dt_{LB} = 4.8\text{e-}7$	$dt_{LB} = 3.2\text{e-}7$ $dt_{NS} = 3.0\text{e-}5$

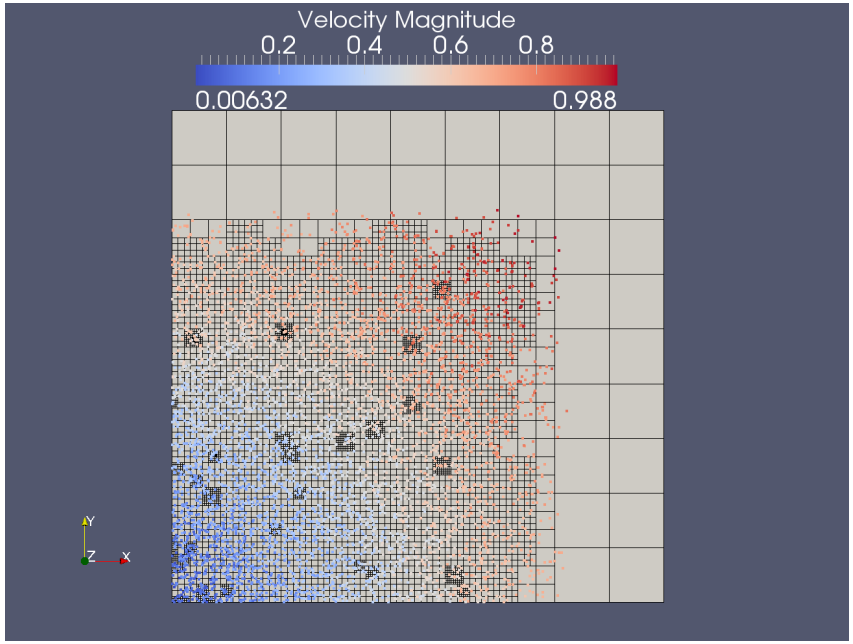
[4] P. Neumann and T. Neckel. Computational Mechanics, 51(2):237–253, 2013.

Algorithmic Requirement Analysis

- (Transient) **Mapping from particles to grid** and vice versa
- Linked lists/ cells: inappropriate
  $\rightarrow$ particles may move more than one cell
- For few particles: global scatter/gather of particle position
  $\rightarrow$ each rank holds all particle information
- For **many particles**: localised approach
  $\rightarrow$ only domains which hold particle maintain particle data

Particle-in-Tree: Algorithm

- Make each cell of spacetree hold array of particles (dynamic, most of time empty)
- Make physics move particle, i.e. update particle positions (local computation within cell)
- Before cell is left: check whether (moved) particle still is within cell. Otherwise *lift* it to next coarser level

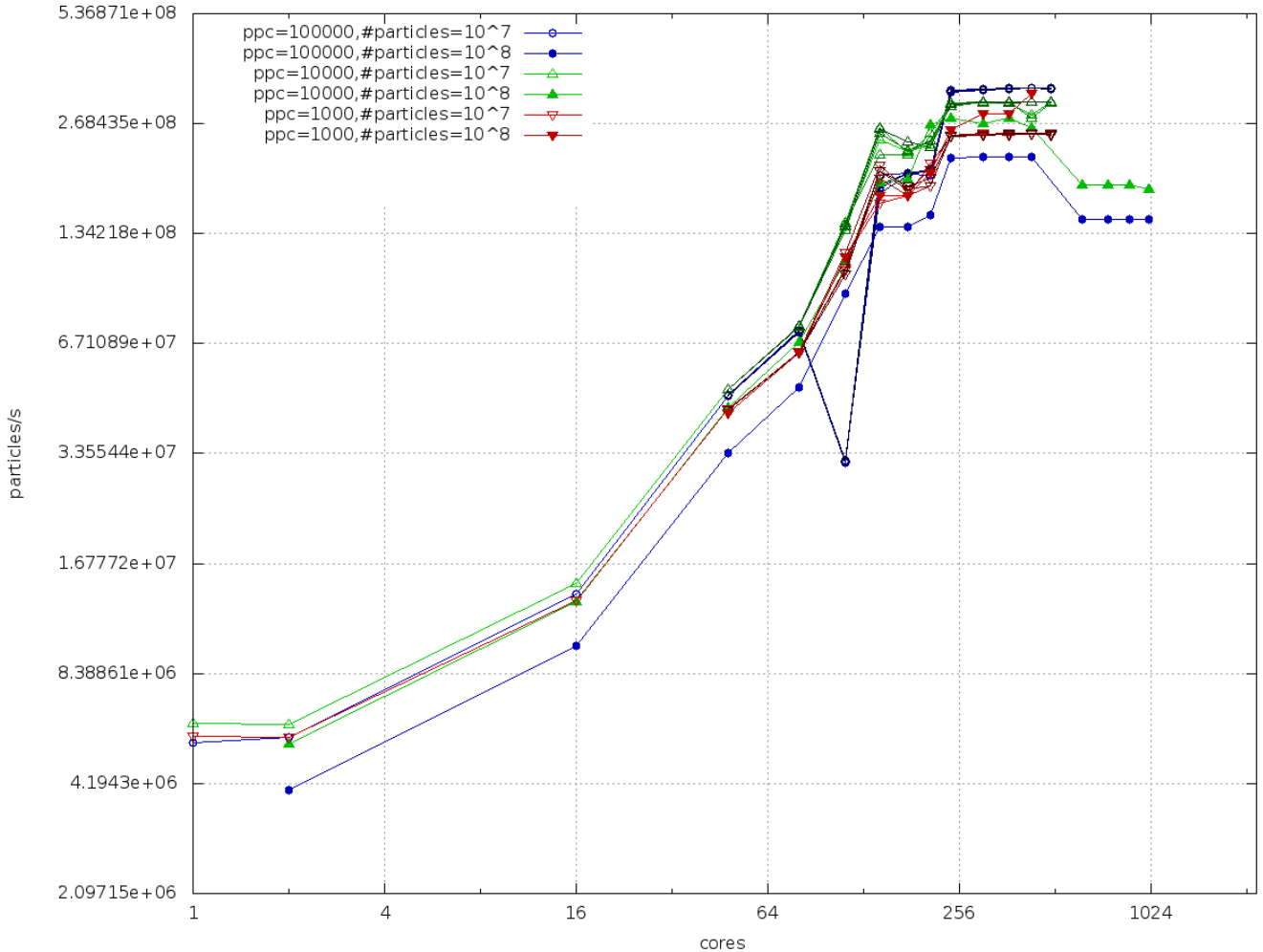


- Apply lift checks recursively
- Before algorithm descends in spacetree: Check whether refined cell holds particles. If so, sieve them into the children (*drop*)
- Apply drops recursively

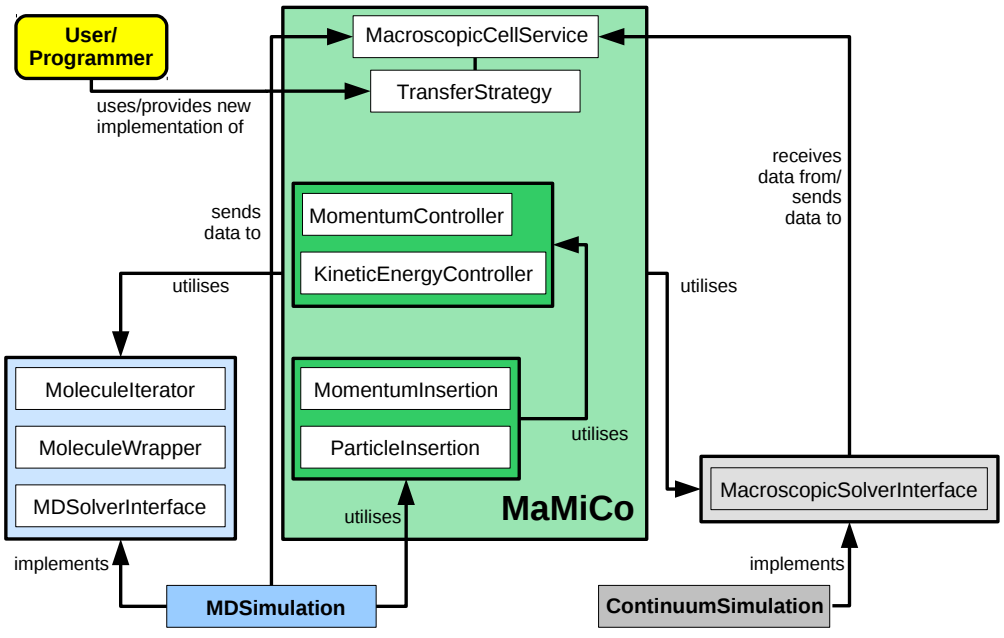
Remarks:

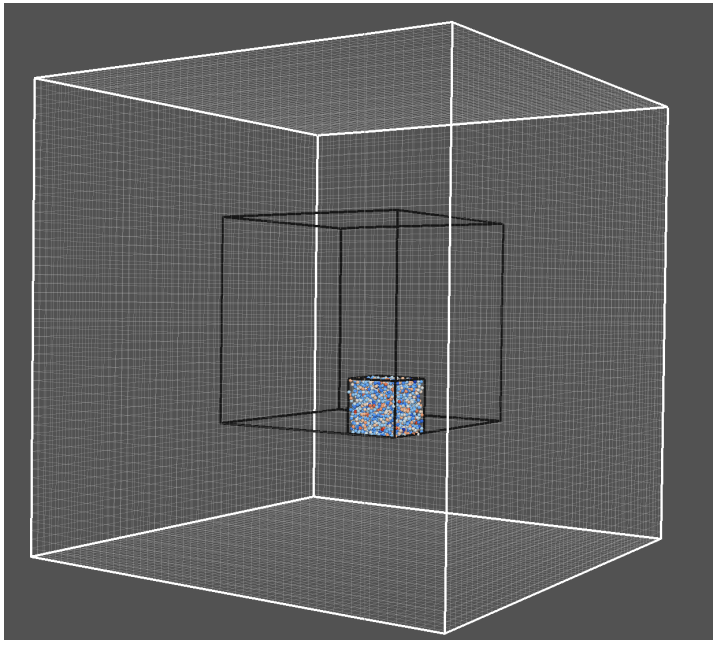
- Particles belonging to level ℓ are never dropped deeper than level ℓ (drop constraint)
- Benchmarking difficulties: Density, speed, mesh width...

Particle-in-Tree: Scaling



Outlook: Lattice Boltzmann–Molecular Dynamics





Goal:

- Resolve regions of interest by molecular dynamics
- Resolve other regions by Lattice Boltzmann
- Partitioned approach: Macro-Micro-Coupling Tool[5]
  $\rightarrow$ Separation: molecular dynamics $\leftrightarrow$ Mesh-based solver

[5] P. Neumann and N. Tchipev. Proceedings of ISPDC, 2012.

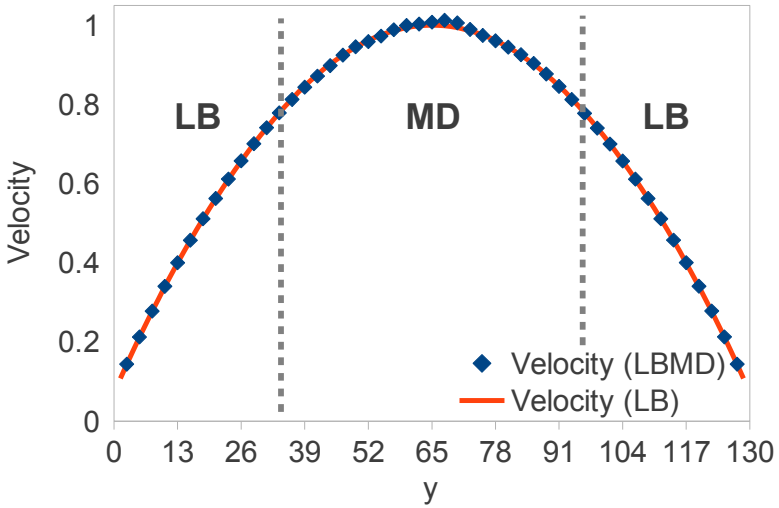
LB–MD: Channel Flow

LB codes:

- Peano framework[6]
- waLBerla[7]
- OpenLB (in preparation)

MD codes:

- Simple MD
- LAMMPS[8]
- ESPResSo[9] (in preparation)
- MarDyn (planned)



- 3D steady-state coupling
  $\rightarrow$ velocity relaxation
- LB: $54 \times 54 \times 54$ cells, $dx = 2.5$
- MD: 10^5 LJ atoms
- MD: periodic boundaries
  + buffer regions

[6] www.peano-framework.org [7] www.walberla.net [8] lammps.sandia.gov [9] espressomd.org

Thank You...

... for your attention!