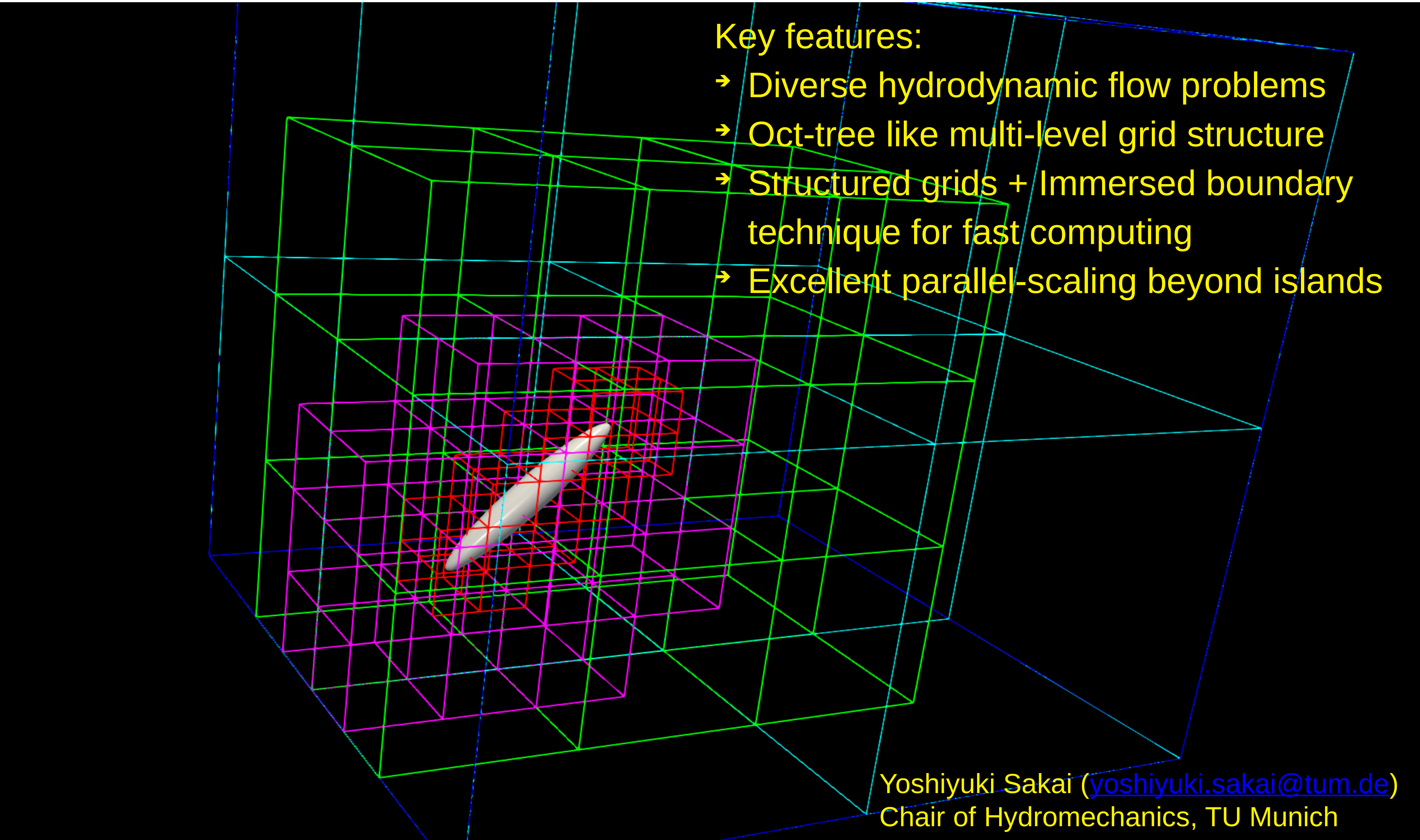
Performance Optimisation of our in-house CFD code MGLET

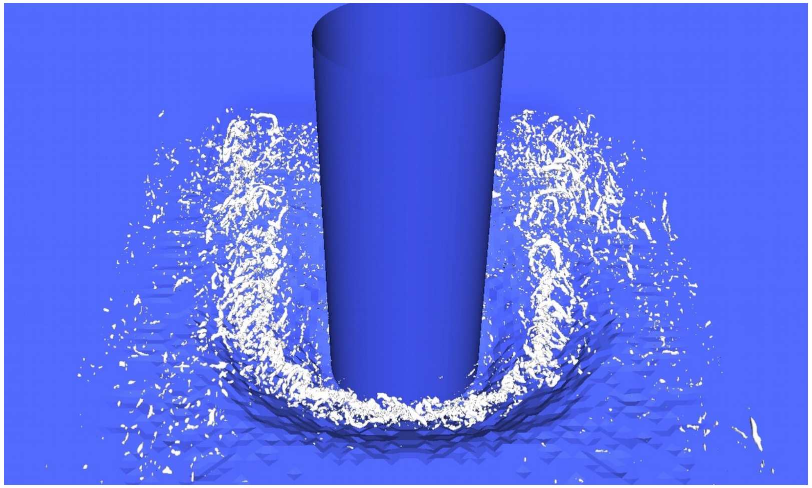


Key features:

- Diverse hydrodynamic flow problems
- Oct-tree like multi-level grid structure
- Structured grids + Immersed boundary technique for fast computing
- Excellent parallel-scaling beyond islands

Yoshiyuki Sakai (yoshiyuki.sakai@tum.de)
Chair of Hydromechanics, TU Munich

Our in-house CFD code MGLET

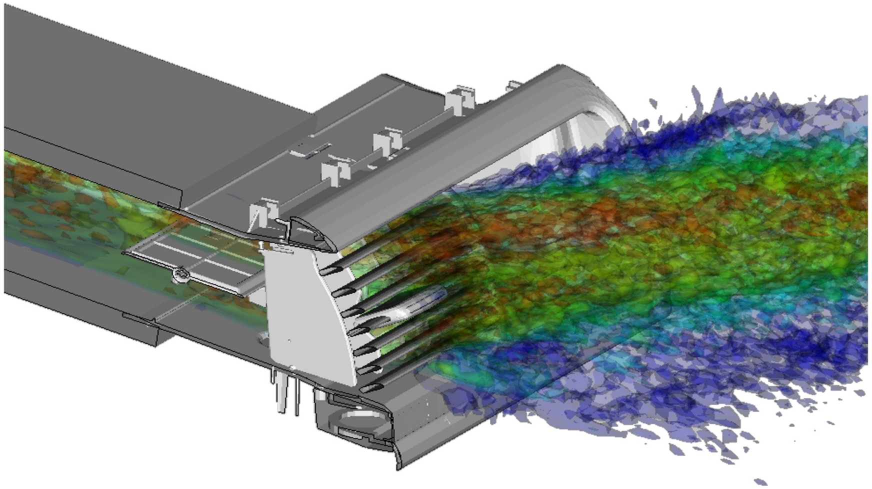


What MGLET can do:

- Turbulent flow in complex domains (DNS/LES)
- Scalar transport, aeroacoustics, non-Newtonian fluid flow
- Adaptive local grid refinement

MGLET in HPC:

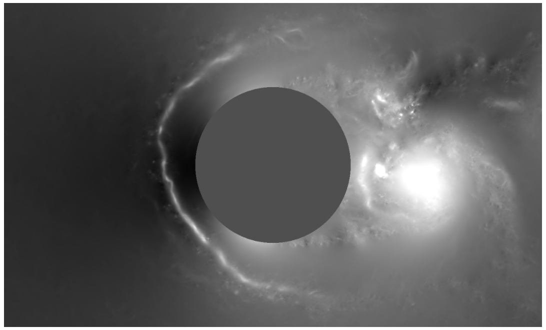
- >50% parallel efficiency up to 33000 cores
- Up to 30 billion grid cells
- Scalable parallel I/O via HDF5 library

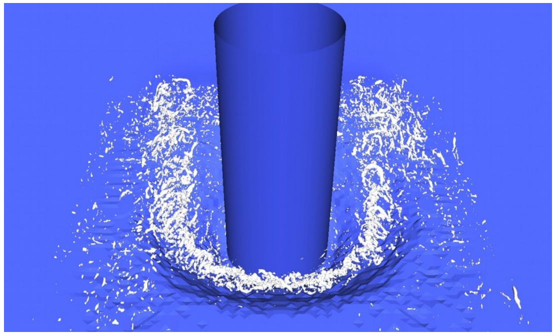


Current projects

1. Turbulent flow around wall-mounted cylinder

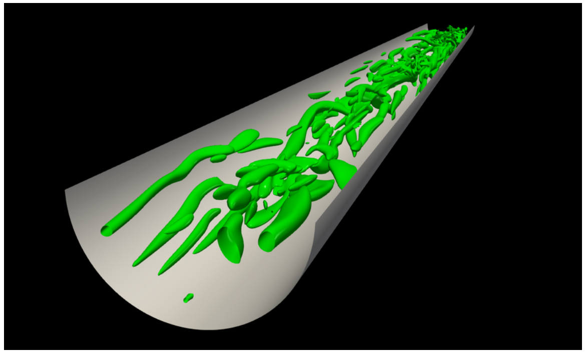
- Highly-resolved LES up to Re=78000
- A study of complex scour hole development process around a bridge pillar
- Investigating three-dimensional turbulence activities inside a real-world scour hole geometry
- Comparative investigations between numerics and experiments





2. Turbulent secondary flow in straight half-filled pipe

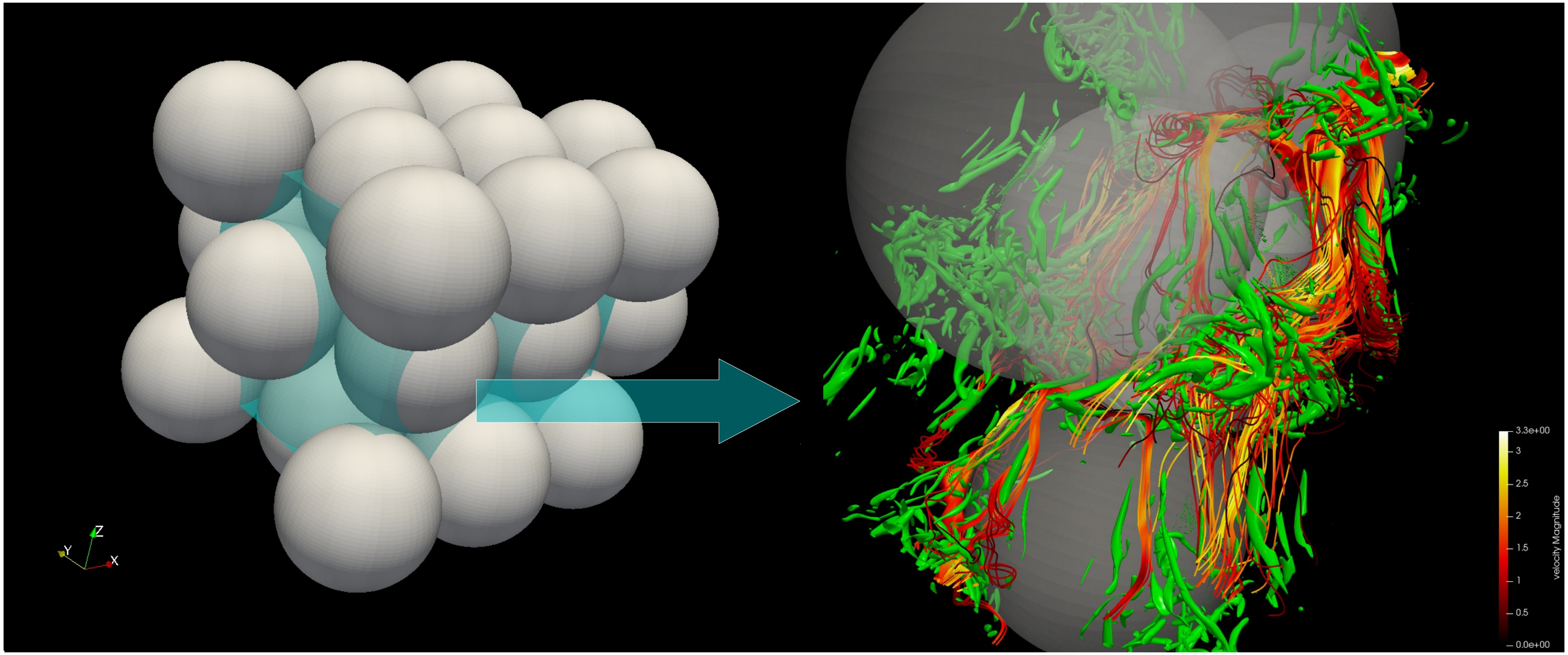
- Fully-resolved DNS
- A study of turbulent secondary flow phenomena in half-filled pipe
- Engineering applications:
  e.g. waste-water channels, urban canals



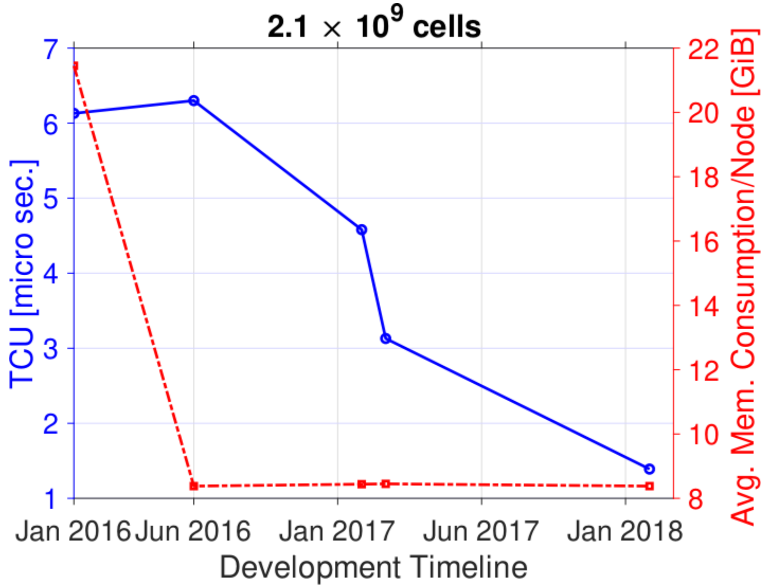
Current projects

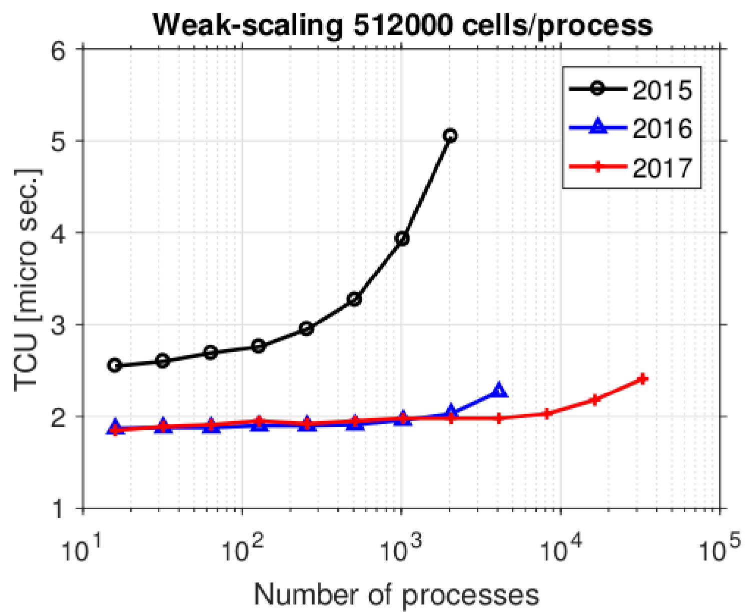
3. Transient porous media flow through close-pack spheres

- Fully-resolved DNS
- Transient flow accelerating from zero velocity
- 3 distinct flow regimes: linear creeping, non-linear unsteady & chaotic turbulent
- Engineering & environmental applications:
  e.g.: coral communities exposed to wave motions, liquid chromatography



Continuous performance optimisation





- Close collaboration with CFDLab* @ LRZ since 2015
- **4x** performance improvement** over 25 months → Ever faster!
- Significant increase in the maximum number of cells → Ever larger!
- Parallel I/O implementation and optimisation resulted **20 – 25x** shorter I/O time*** w.r.t. original serial I/O
- Currently working on node-level (SIMD) optimisation

* https://www.lrz.de/services/compute/labs/cfdlab/

** Based on cell-update time (TCU)

*** Tested on IBM GPFS parallel filesystem