VSB TECHNICAL UNIVERSITY OF OSTRAVA

SEMINAR ON NUMERICAL ANALYSIS

*Modelling and Simulation*
*of Challenging Engineering Problems*

NHR FAU

# Solvers for Extreme Scale Computing

SNA ’25 - Ostrava

## Blaheta Lecture


Ulrich Rüde (ulrich.ruede@fau.de)

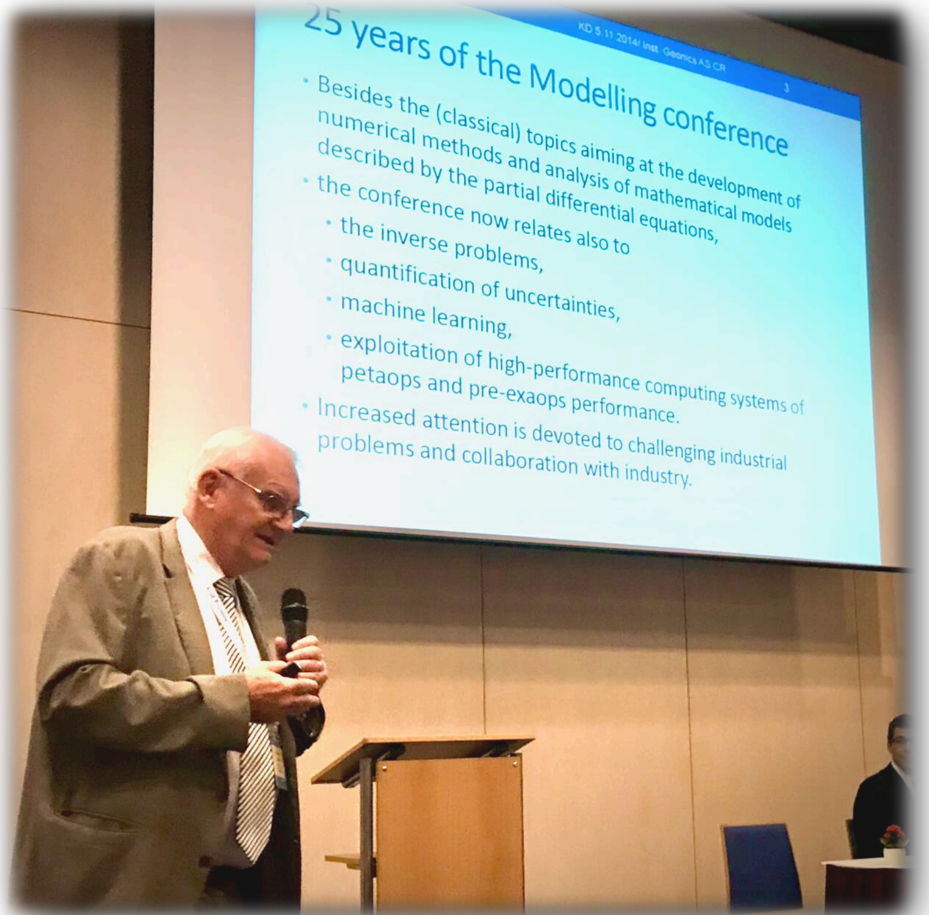

January 30, 2025


LSS
*Lehrstuhl für Simulation*
*Universität Erlangen-Nürnberg*
https://www.cs10.tf.fau.de

*VSB*
*Technical University of Ostrava*

CERFACS
*Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique*
www.cerfacs.fr

# Olomouc September 2019



# Overview

- A preamble
- My guiding example: Earth mantle convection
- Supercomputers
- Scalable Solvers, Multigrid
- Automatic Program Generation
- Scalability, Performance
- Textbook Efficiency

- Lattice Boltzmann for Complex Flows (time permitting)

My talk presents results of my teams
and years of collaboration with colleagues

# Preamble:

## What is the fastest solver for Poisson's equation?

# The context:

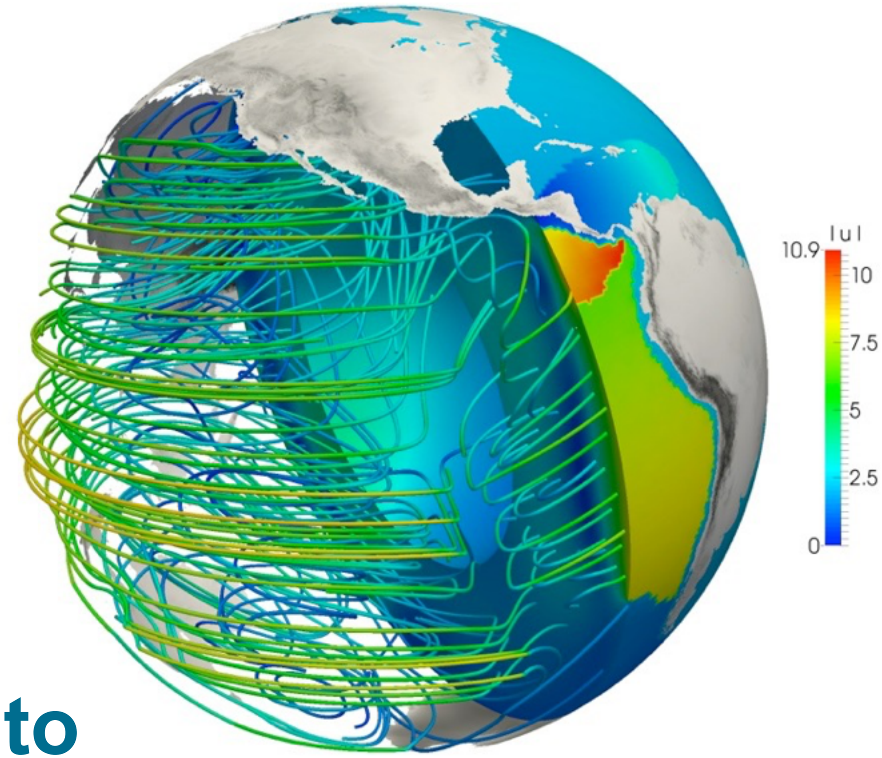
- Scientific Computing is about efficient methods
- Numerical algorithms require a tradeoff between accuracy and cost
    - If accuracy is irrelevant, cheap algorithms are trivial to find
    - If cost is irrelevant, accuracy is trivial to achieve

# Setting accuracy in relation to cost:

- We need metrics for
  - cost (algorithmic complexity)
  - accuracy (magnitude of error)
- Both are surprisingly unclear
  - Cost: counting #unknowns, counting #FLOPS, memory consumption, run time, energy consumption, ….
  - Accuracy: Residual vs. error? Which norm?
    Often not the solution is needed, but a functional thereof, …
- All this makes a difference in what is needed
- The new kid on the block:
  - **Deep Learning (for PDE)**
    When your natural intelligence fails, use an artificial one!

# Making the question more specific:

- When teaching linear algebra we insist that students learn:
  - Gaussian elimination costs $\sim \frac{2}{3}n^3$ FLOPS
- But for PDE? Let's restrict to
  - Poisson's equation in the unit square with
  - 5-point discretization of the Laplace operator
    (at this stage we thus avoid the discussion of accuracy)
  - Complexity metric: FLOPS
- With this: What is the cost of solving the discretized Poisson equation on a grid with $n = n_x \times n_y$ unknowns?
  - … what is the best algorithm known today?
  - … what is the answer for 3D? … or more general equations?
    … more advanced discretization techniques?
- In any case: I insist on the constant, multiplying the dominating term
- When the complexity is (almost) linear, the constant is the critical quantity



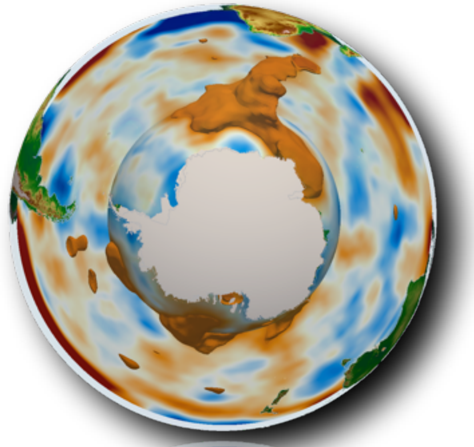

# Part I - An introductory excursion to Earth Mantle Convection

# Simple Earth Mantle convection models:
## Stokes equation coupled with energy transport

$$-\nabla \cdot (2\eta\epsilon(\mathbf{u})) + \nabla p = \rho(T)\mathbf{g},$$
$$\nabla \cdot \mathbf{u} = 0,$$
$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T - \nabla \cdot (\kappa \nabla T) = \gamma.$$

| | |
|---|---|
| $\mathbf{u}$ | velocity |
| $p$ | dynamic pressure |
| $T$ | temperature |
| $\nu$ | viscosity of the material |
| $\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ | strain rate tensor |
| $\rho$ | density |
| $\kappa, \gamma, \mathbf{g}$ | thermal conductivity, heat sources, gravity vector |

Gmeiner, Waluga, Stengel, Wohlmuth, UR: Performance and Scalability of Hierarchical Hybrid Multigrid Solvers for Stokes Systems, SIAM J. Scientific Comp., 2015.



Stokes equation: $$-\mathrm{div}(\nabla \mathbf{u} - p\mathbf{I}) = \mathbf{f}, \quad \mathrm{div}\,\mathbf{u} = 0$$

FEM Discretization:

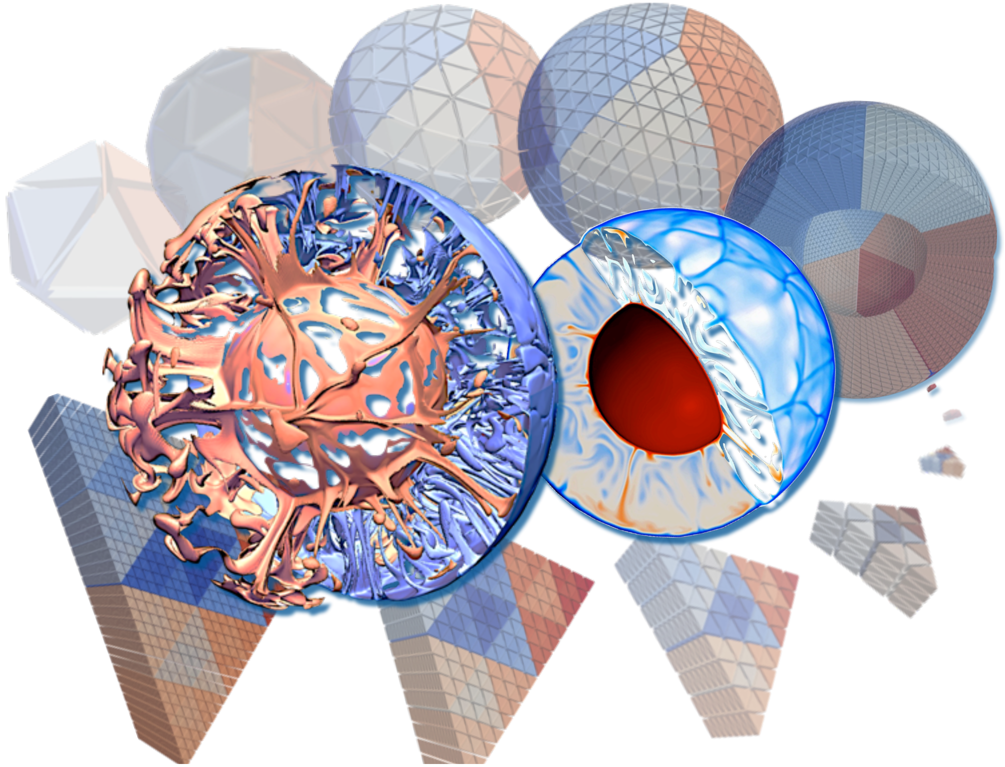
$$\mathbf{a}(\mathbf{u}_l, \mathbf{v}_l) + \mathbf{b}(\mathbf{v}_l, p_l) = \mathbf{L}(\mathbf{v}_l) \qquad \forall\, \mathbf{v}_l \in \mathbf{V}_l,$$
$$\mathbf{b}(\mathbf{u}_l, q_l) - \mathbf{c}(p_l, q_l) = 0 \qquad \forall\, q_l \in Q_l,$$

with: $\mathbf{a}(\mathbf{u},\mathbf{v}) := \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v}\, dx, \quad \mathbf{b}(\mathbf{u},q) := -\int_\Omega \mathrm{div}\,\mathbf{u} \cdot q\, dx$

Schur-complement formulation:

$$\begin{bmatrix} \mathbf{A}_l & \mathbf{B}_l^\top \\ \mathbf{0} & \mathbf{C}_l + \mathbf{B}_l \mathbf{A}_l^{-1} \mathbf{B}_l^\top \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}_l \\ \underline{p}_l \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}}_l \\ \mathbf{B}_l \mathbf{A}_l^{-1} \underline{\mathbf{f}}_l \end{bmatrix}$$

# Mantle Convection



## Why Mantle Convection?

- driving force for plate tectonics
- mountain building and earthquakes

## Why Exascale?

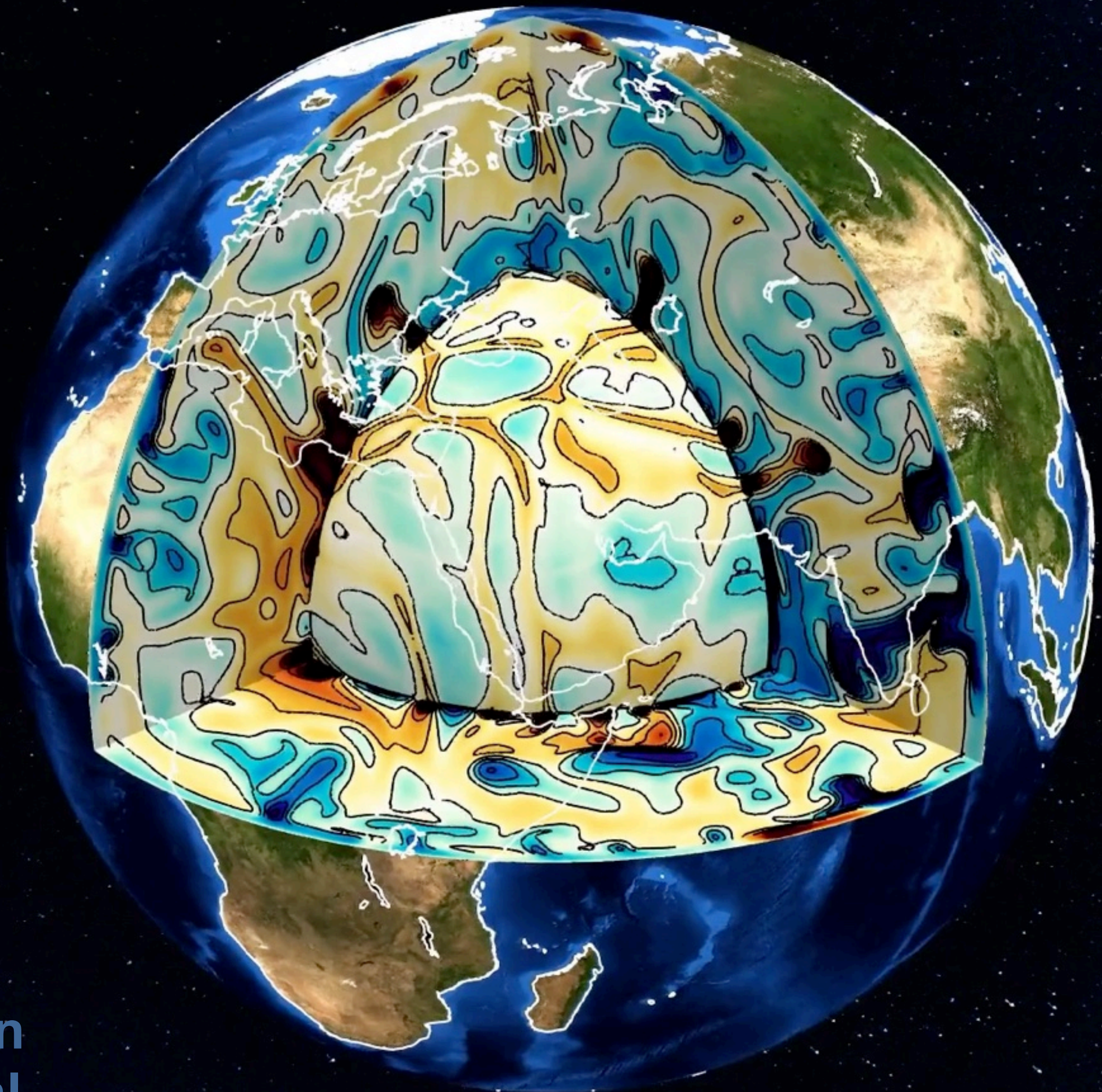
- mantle has $10^{12}$ km$^3$
- inversion and UQ blow up cost

## Why TERRA NEO

- implementation based on HYTEG
- scalable and fast
- sustainable framework

## Challenges

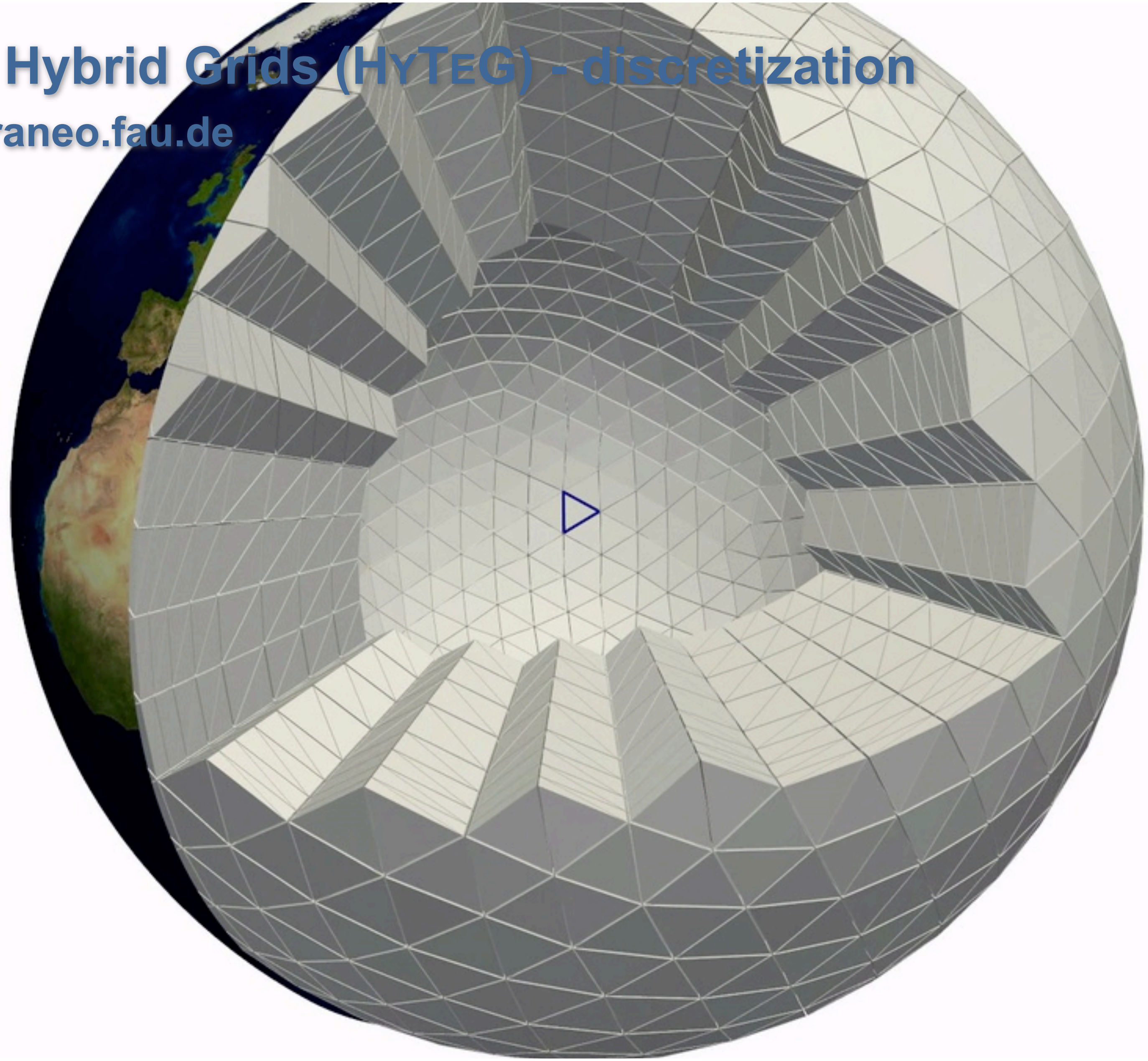
- computer sciences: software design for exascale systems
- mathematics: HPC performance oriented metrics
- geophysics: model complexity and uncertainty
- bridging disciplines: integrated co-design



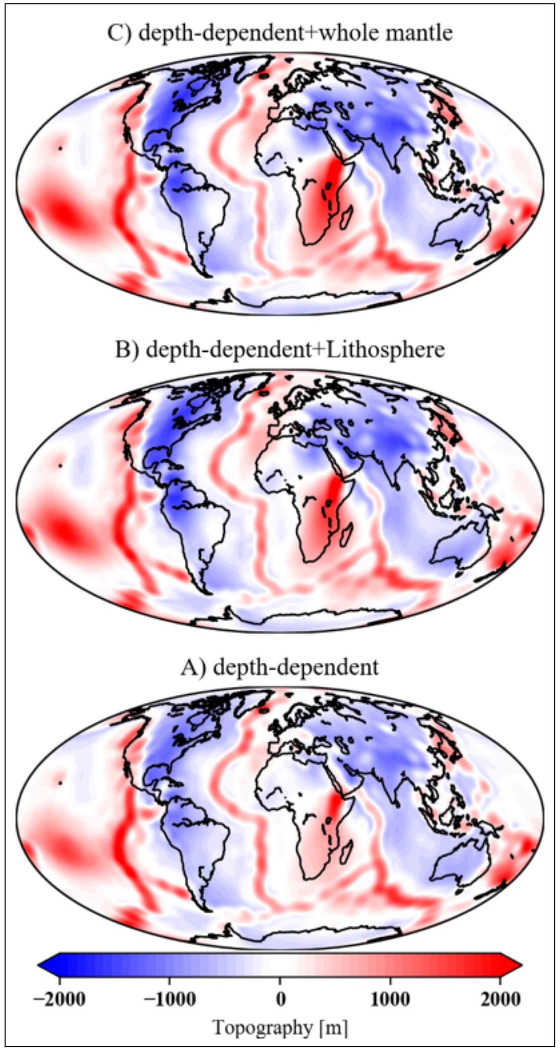
**HyTeG - application**
**Dissertation N. Kohl**

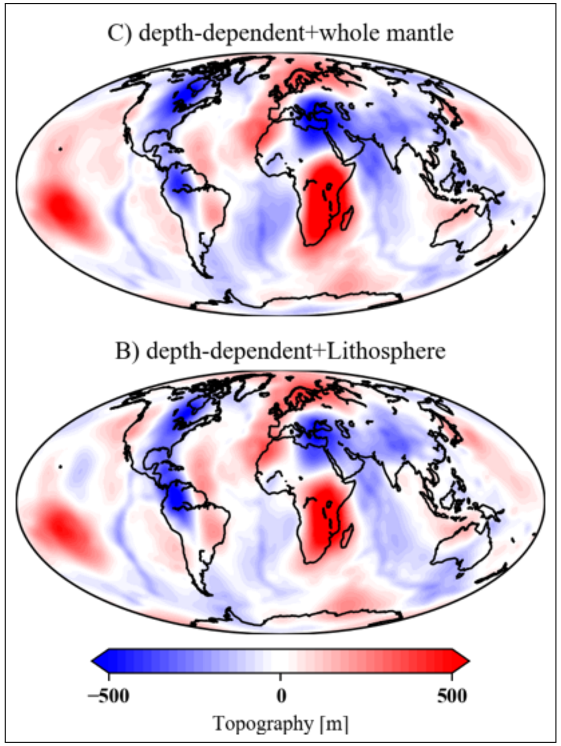
# Hierarchical Hybrid Grids (HyTeG) - discretization

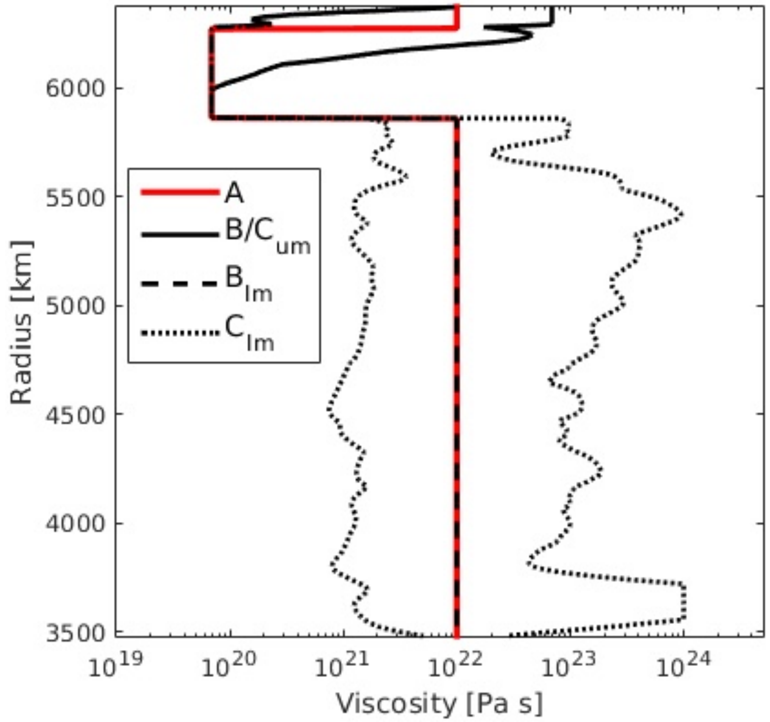
full video on **terraneo.fau.de**



# Geophysical in-silico experiment: dynamical topography

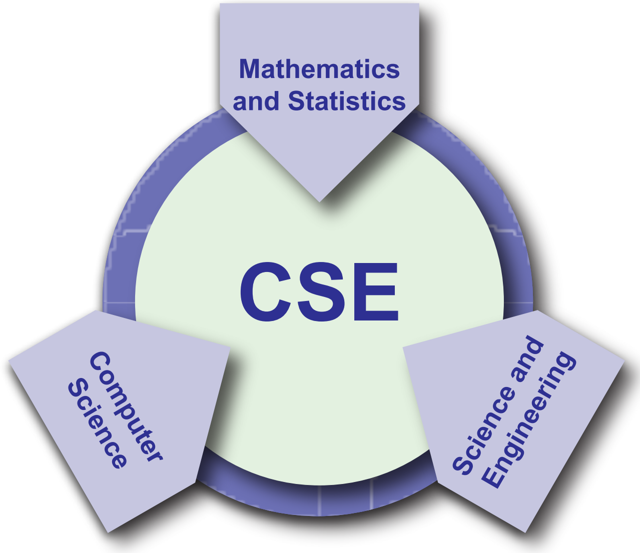


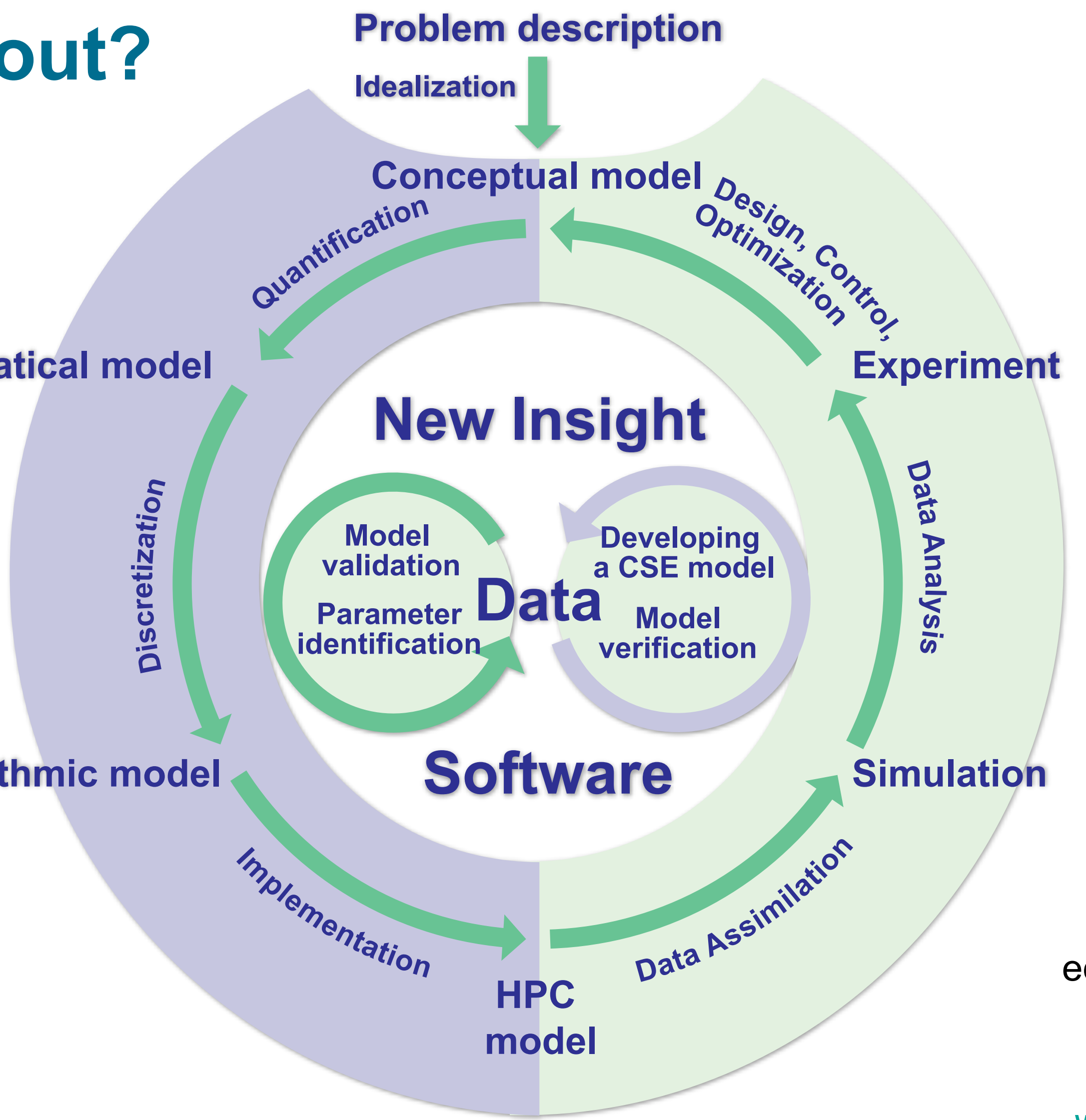

- Global dynamical topography depending on different assumptions
- radial viscosity variations

Weismüller, J., Gmeiner, B., Ghelichkhan, S., Huber, M., John, L., Wohlmuth, B., ... & Bunge, H. P. (2015). Fast asthenosphere motion in high-resolution global mantle flow models. *Geophysical Research Letters*, *42*(18), 7429-7435.




# Part II: The essence of Computational Science and Engineering (CSE)

# What is it about?




UR, Willcox, K., McInnes, L. C., & Sterck, H. D. (2018). Research and education in **computational science and engineering**. *Siam Review*, *60*(3), 707-754.



VSB TECHNICAL UNIVERSITY OF OSTRAVA

# Part III: Supercomputers


SuperMUC-NG: Leibniz Supercomputing Center Garching/Munich


Frontier: Oak Ridge National Laboratory, USA

# A personal review of computer evolution

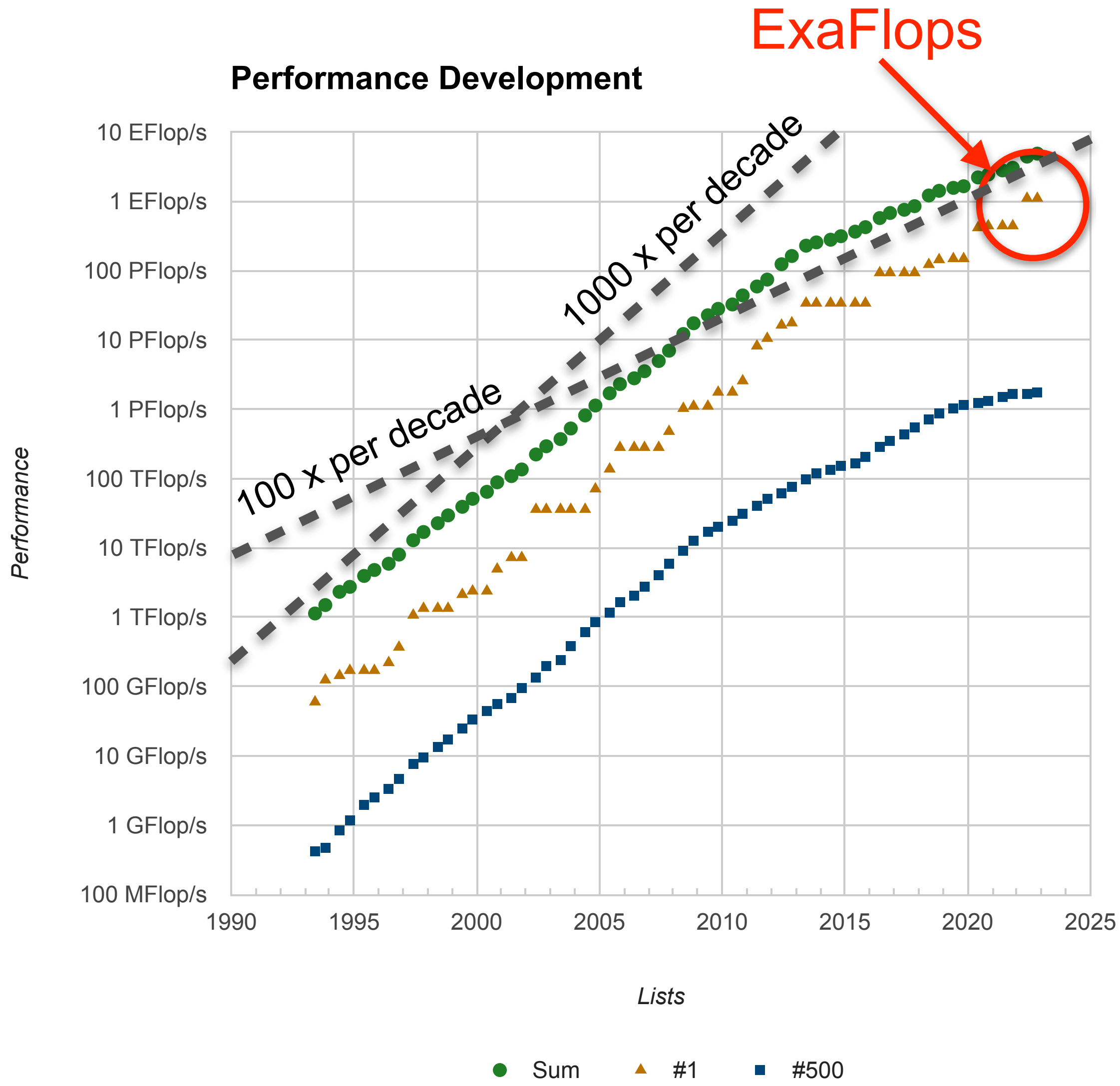
Let's first take a Look Back!


Die PERM im Zustand von 1956 — bis Sommersemester 1974

WE HAVE COME A LONG WAY!

# ExaFLOPS: TOP 500 List

- The fastest computers today (Frontier, ElCapitan) deliver >1 ExaFLOPS= $10^{18}$ FLOPS
- Will the deflation of computational cost continue?

# The Top 5 Supercomputers

| Rank | System | Cores | Rmax (PFlop/s) | Rpeak (PFlop/s) | Power (kW) |
|---|---|---|---|---|---|
| 1 | **El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE<br>DOE/NNSA/LLNL<br>United States | 11,039,616 | 1,742.00 | 2,746.38 | 29,581 |
| 2 | **Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE<br>DOE/SC/Oak Ridge National Laboratory<br>United States | 9,066,176 | 1,353.00 | 2,055.72 | 24,607 |
| 3 | **Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel<br>DOE/SC/Argonne National Laboratory<br>United States | 9,264,128 | 1,012.00 | 1,980.01 | 38,698 |
| 4 | **Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure<br>Microsoft Azure<br>United States | 2,073,600 | 561.20 | 846.84 | |
| 5 | **HPC6** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.8, HPE<br>Italy | 3,143,520 | 477.90 | 606.97 | 8,461 |

Measured Performance

Nominal Performance

Power consumption

# cores:
degree of parallelism

Fastest Computer
in Europe currently
#5, HPC6 (Italy),
477 PFlop/s

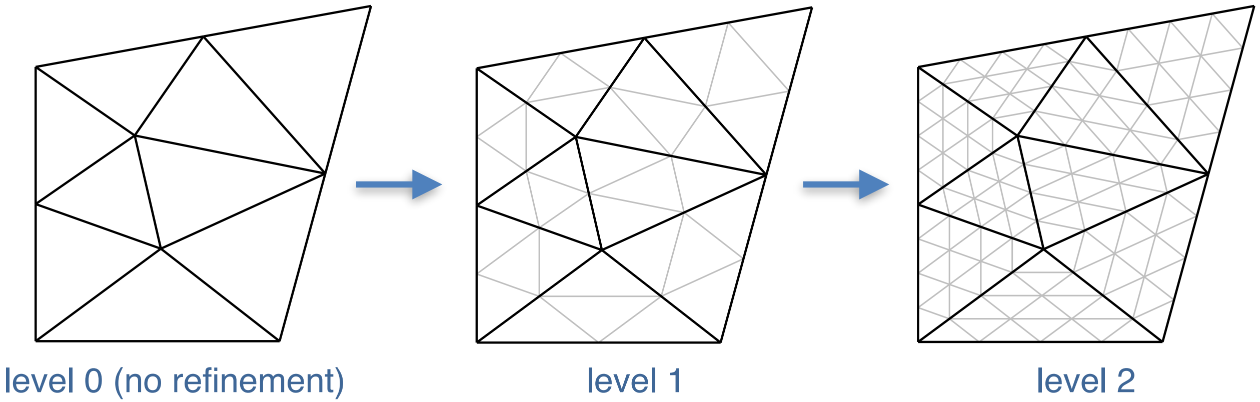
# Moore's Law: We are children of the golden age of computing!

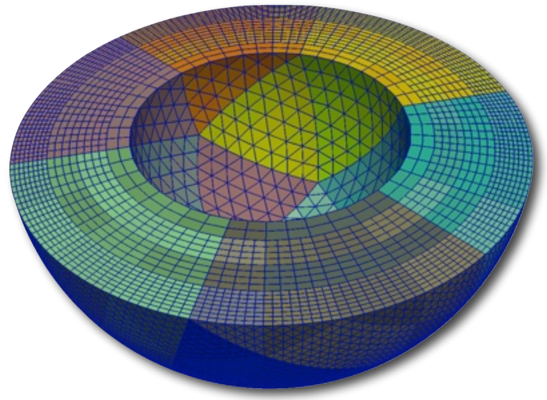
- We are used to 1000x improvement per decade
  It slows down, but for now seems to continue with 100x improvement per decade
- We have seen improvement of speed by factors of
  - $10^7$ since 1993 and
  - $10^{14}$ since 1963

must ignore Einstein's speed limit at $10^9$ km/h

## Moore's Law for the Hitchhikers of the Galaxy

- If my car had seen similar improvements since 1993, it would drive instead of $10^2$ km/h with $10^9$ km/h
  - since the solar system has a diameter of $10^{10}$ km, we could reach Neptune for a summer holiday within approximately 5 hours.
- If my car had sped up by $10^{14}$ since 1963 it would drive at $10^{16}$ km/h
  - since our home galaxy has a diameter of $10^{18}$ km we could tour the galaxy by driving some 100 hours



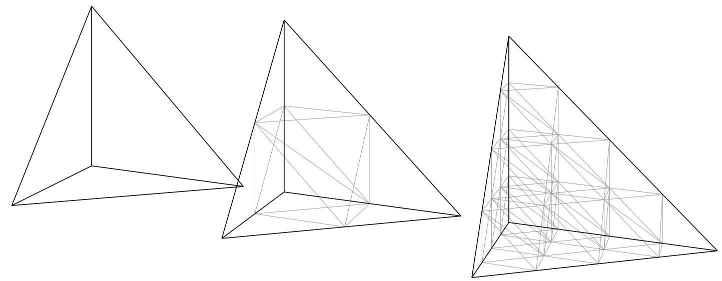
# Part IVa: Hierarchical Hybrid Tetrahedral Grids for Finite Elements


Bergen, B. K., & Hülsemann, F. (2004). **Hierarchical hybrid grids**: data structures and core algorithms for multigrid. *Numerical linear algebra with applications*, *11*(2-3), 279-291.

Bergen, B., Gradl, T., Hulsemann, F., & UR (2006). A massively **parallel multigrid** method for finite elements. *Computing in science & engineering*, *8*(6), 56-62.

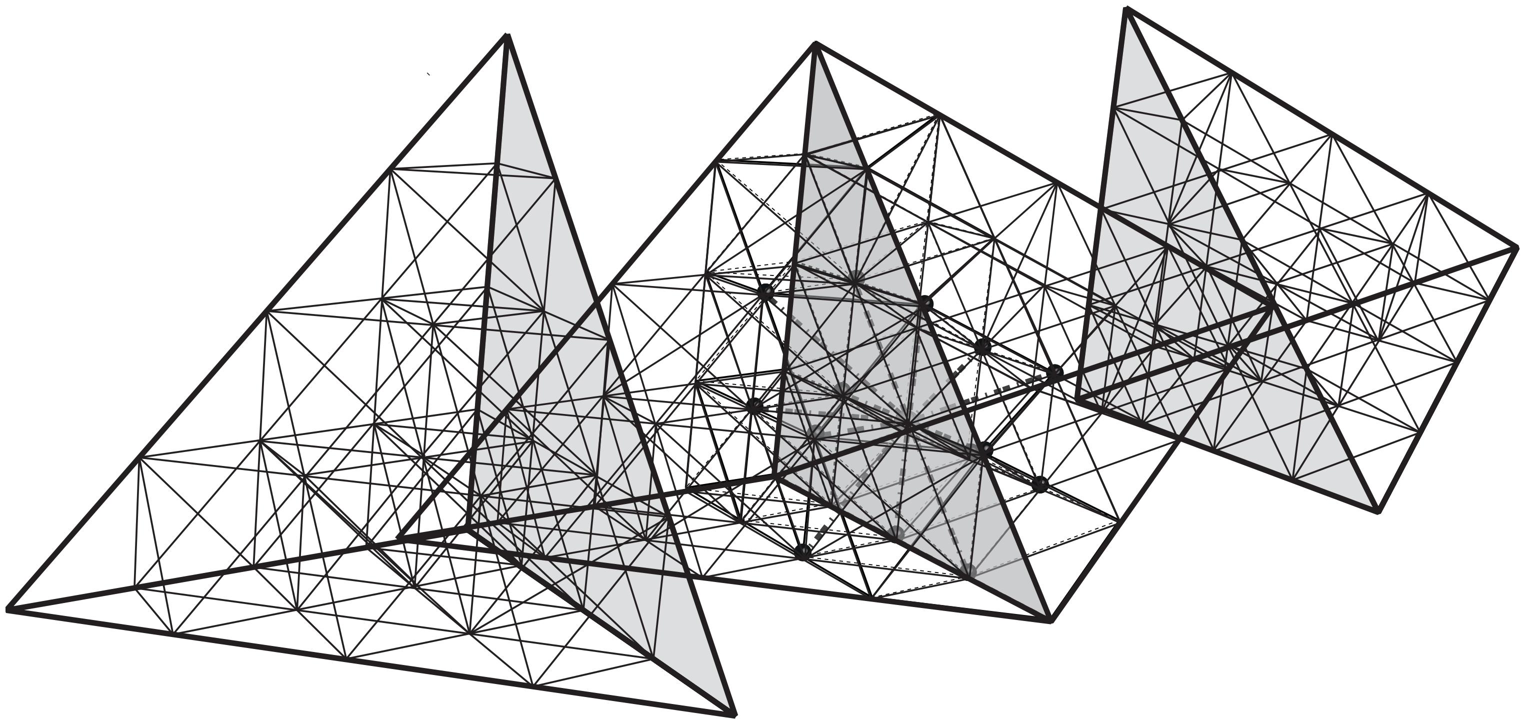
Kohl, N., Thönnes, D., Drzisga, D., Bartuschat, D., UR (2019). The **HyTeG** finite-element software framework for scalable multigrid solvers. *International Journal of Parallel, Emergent and Distributed Systems*, *34*(5), 477-496.

## HyTeG: A matrix-free architecture for FE



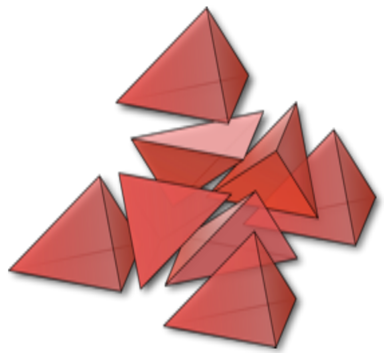
Structured refinement of an unstructured base mesh

Geometrical Hierarchy: Volume, Face, Edge, Vertex

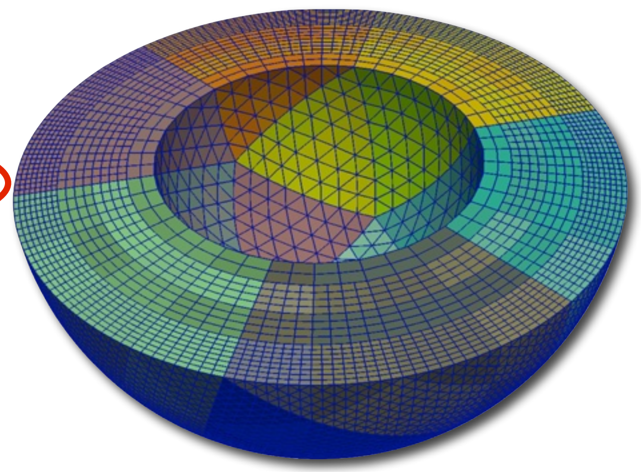


# Hierarchical Hybrid Grids (HHG) and Multigrid (HyTeG)

- Parallelize multigrid for tetrahedral finite elements
  - partition domain
  - parallelize all operations on all grids
  - use clever data structures
  - matrix free implementation
- Coarse grids
  - agglomeration?
  - sequential dependency in grid hierarchy
- Elliptic problems always require global communication and thus coarser grids for the global data transport



Bey's Tetrahedral Refinement

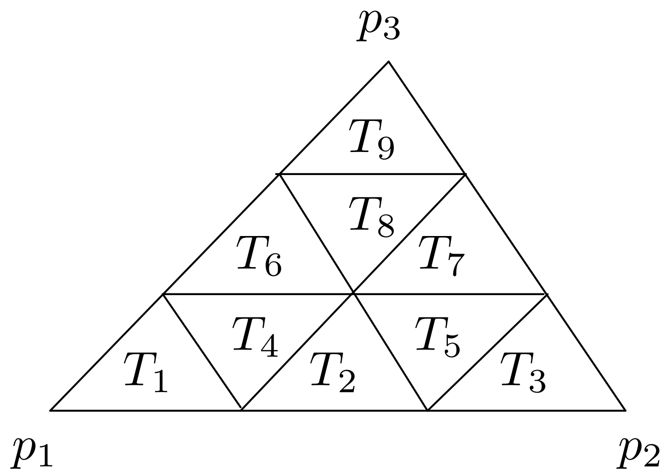


B. Bergen, F. Hülsemann, UR, G. Wellein: „Is 1.7× $10^{10}$ unknowns the largest finite element system that can be solved today?", SuperComputing, 2005.

Gmeiner, UR, Stengel, Waluga, Wohlmuth: Towards Textbook Efficiency for Parallel Multigrid, Journal of Numerical Mathematics: Theory, Methods and Applications, 2015

# Remark about R. Blaheta's work (1)

$$h = H/m.$$




- An essential result is the proof of the strengthened Cauchy-Bunyakowski-Schwarz inequality for linear triangular elements in 2D on HyTeG-like grids

Axelsson, O., & Blaheta, R. (2004). Two simple derivations of universal bounds for the CBS inequality constant. *Applications of Mathematics*, *49*, 57-72.

**Theorem 3.1.** *Consider the bilinear forms* (2.14) *and* (2.15) *corresponding respectively to a general* 2D *anisotropic Laplacian or a general* 2D *anisotropic elasticity operator on* $\Omega$. *Further, let* $\mathcal{T}_H$ *be a triangulation of* $\Omega$ *and assume that the problem coefficients are constant on the coarse elements* $E \in \mathcal{T}_H$. *Assume also that each element* $E \in \mathcal{T}_H$ *is divided into* $m^2$ *smaller congruent triangles in the described way. Then*

$$\gamma_{E,1} \leqslant \sqrt{\frac{m^2-1}{m^2}}. \tag{3.1}$$

# Remark about R. Blaheta's work (2)

Axelsson, O., & Blaheta, R. (2004). Two simple derivations of universal bounds for the CBS inequality constant. *Applications of Mathematics*, *49*, 57-72.

Remark 4.1. The relation

$$A_H^{(2)} = \frac{4}{3} A_{H/2}^{(1)} - \frac{1}{3} \begin{bmatrix} 0 & 0 \\ 0 & A_H^{(1)} \end{bmatrix}$$

quadratic elements on coarse grid

linear elements on fine grid

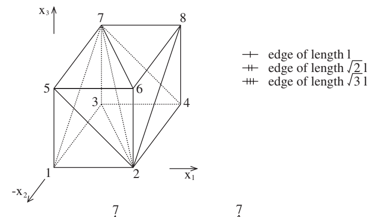
linear elements on coarse grid

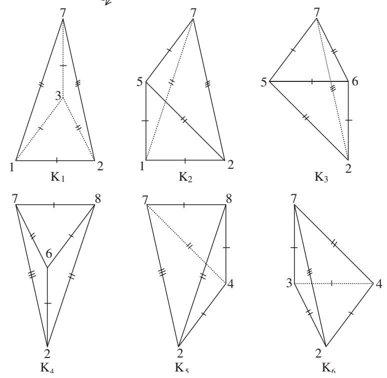
- This relates the matrices for quadratic and linear elements
- Axelsson and Blaheta use this to derive a strengthened Cauchy-Bunyakowski-Schwarz inequality also for quadratic elements
- The same relation can also be used to prove the so-called tau-extrapolation within a multigrid/multilevel method, as, e.g. in

Rüde, U. (1992). The hierarchical basis extrapolation method. *SISC 13*(1), 307-318.
Jung, M., & Rüde, U. (1998). Implicit extrapolation methods for variable coefficient problems. SISC *19*(4), 1109-1124.

# Remark about R. Blaheta’s work (3)

Blaheta, R. (2003). Nested tetrahedral grids and strengthened CBS inequality. *Numerical linear algebra with applications*, *10*(7), 619-637.

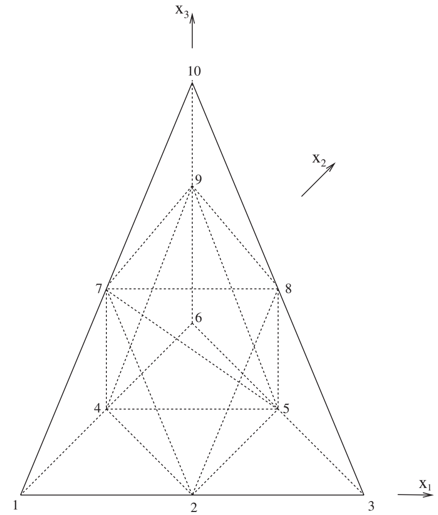


- The article contains a concise description of the 3D refinement procedure (based e.g. on work of Kuhn and Bey)
- and proves the strengthened Cauchy-Schwarz inequalities in 3D
- This is essential to show a fast convergence of multigrid methods

*Theorem 3.1*
Let $\mathscr{T}_H, \mathscr{T}_h$ be two divisions of the domain $\Omega$ into tetrahedra, which are constructed in the way described in Section 2. Let any tetrahedra $T \in \mathscr{T}_H$ is decomposed into $m^3$ tetrahedra from $\mathscr{T}_h$, $m = 2, 3, 4, 5$. Assume also that $D$ is constant on each tetrahedron $T \in \mathscr{T}_H$. Then for any $D$ and any shape of tetrahedra from $\mathscr{T}_H$, the C.B.S. constant $\gamma$ from (22) can be bounded by $\bar{\gamma}$,

$$\gamma \leqslant \bar{\gamma} = \sqrt{\frac{(m^2-1)(m^2+2)}{m^2(m^2+1)}} \tag{23}$$



# Algorithms Matter!

- Solution of Laplace equation in 3D wit $N=n^3$ unkowns
- Direct methods:
  - banded: $\sim n^7 = N^{2.33}$
  - nested dissection: $\sim n^6 = N^2$
- Iterative Methods:
  - Jacobi: $\sim 50\ n^5 = 50\ N^{1.66}$
  - CG: $\sim 100\ n^4 = 100\ N^{1.33}$

[redacted]

| Energy per FLOP: 1nJ | | | | |
|---|---|---|---|---|
| Computer Generation | gigascale: $10^9$ | terascale: $10^{12}$ | petascale: $10^{15}$ | exascale: $10^{18}$ |
| problem size: DoF=N | $10^6$ | $10^9$ | $10^{12}$ | $10^{15}$ |
| Direct method: $1*N^2$ | 0.278 Wh | 278 kWh | 278 GWh | 278 PWh |
| Krylov method: $100*N^{1.33}$ | 10 Ws | 28 Wh | 278 kWh | 2.77 GWh |
| [redacted] | | | | |
| TerraNeo prototype (est. for Juqueen) | 0.13 Wh | 30 Wh | 27 kWh | ? |

# Exploring the limits

Gmeiner et al. 2016, A quantitative **performance study for Stokes** solvers at the extreme scale, Journal of Computational Science.

- matrix-free multigrid with Uzawa smoother
- optimized for minimal memory consumption

- $10^{13}$ Unknowns correspond to 80 TByte for the solution vector
- Juqueen had ~450 TByte memory
- matrix free implementation essential

| nodes | threads | DoFs | iter | time | time w.c.g. | time c.g. in % |
|---|---|---|---|---|---|---|
| 5 | 80 | $2.7 \cdot 10^{9}$ | 10 | 685.88 | 678.77 | 1.04 |
| 40 | 640 | $2.1 \cdot 10^{10}$ | 10 | 703.69 | 686.24 | 2.48 |
| 320 | 5 120 | $1.2 \cdot 10^{11}$ | 10 | 741.86 | 709.88 | 4.31 |
| 2 560 | 40 960 | $1.7 \cdot 10^{12}$ | 9 | 720.24 | 671.63 | 6.75 |
| 20 480 | 327 680 | $1.1 \cdot 10^{13}$ | 9 | 776.09 | 681.91 | 12.14 |

# Geometric Multigrid: to $10^{12}$ unknowns and beyond

Goal: solve $A^h u^h = f^h$ using a hierarchy of grids

Relax on $A^h u^h = f^h$

Residual $r^h = f^h - A^h u^h$

Restrict $r^H = I_h^H r^h$

Correct $u^h \leftarrow u^h + e^h$

Interpolate $e^h = I_H^h e^H$

Multigrid uses coarse grids
to accomplish the inevitable global data
exchange in the most efficient possible way

...

...

# Algorithms for saddle point systems

Benzi, M., Golub, G. H., & Liesen, J. (2005). Numerical solution of saddle point problems. *Acta numerica*, *14*, 1-137.
Rozložník, M. (2018). *Saddle-point problems and their iterative solution*. Basel: Birkhäuser.

- Monolithic multigrid

Gmeiner, B., Rüde, U., Stengel, H., Waluga, C., & Wohlmuth, B. (2015). Towards textbook efficiency for parallel multigrid. *Numerical Mathematics: Theory, Methods and Applications*, *8*(1), 22-46.

Drzisga, D., John, L., Rude, U., Wohlmuth, B., & Zulehner, W. (2018). On the analysis of block smoothers for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, *39*(2), 932-960.

Kohl, N., & Rüde, U. (2022). Textbook efficiency: massively parallel matrix-free multigrid for the Stokes system. *SIAM Journal on Scientific Computing*, *44*(2), C124-C155.

- Exploiting block structure and/or Schur complement formulation

Axelsson, O., Blaheta, R., Byczanski, P., Karátson, J., & Ahmad, B. (2015). Preconditioners for regularized saddle point problems with an application for heterogeneous Darcy flow problems. *Journal of Computational and Applied Mathematics*, *280*, 141-157.

Blaheta, R., Luber, T., & Kružík, J. (2018). Schur Complement-Schwarz DD Preconditioners for Non-stationary Darcy Flow Problems. In *High Performance Computing in Science and Engineering: Third International Conference, HPCSE 2017, Karolinka, Czech Republic, May 22–25, 2017, Revised Selected Papers 3* (pp. 59-72). Springer.

Darrigrand, V., Dumitrasc, A., Kruse, C., & Rüde, U. (2023). Inexact inner–outer Golub–Kahan bidiagonalization method: A relaxation strategy. *Numerical Linear Algebra with Applications*, *30*(5), e2484.

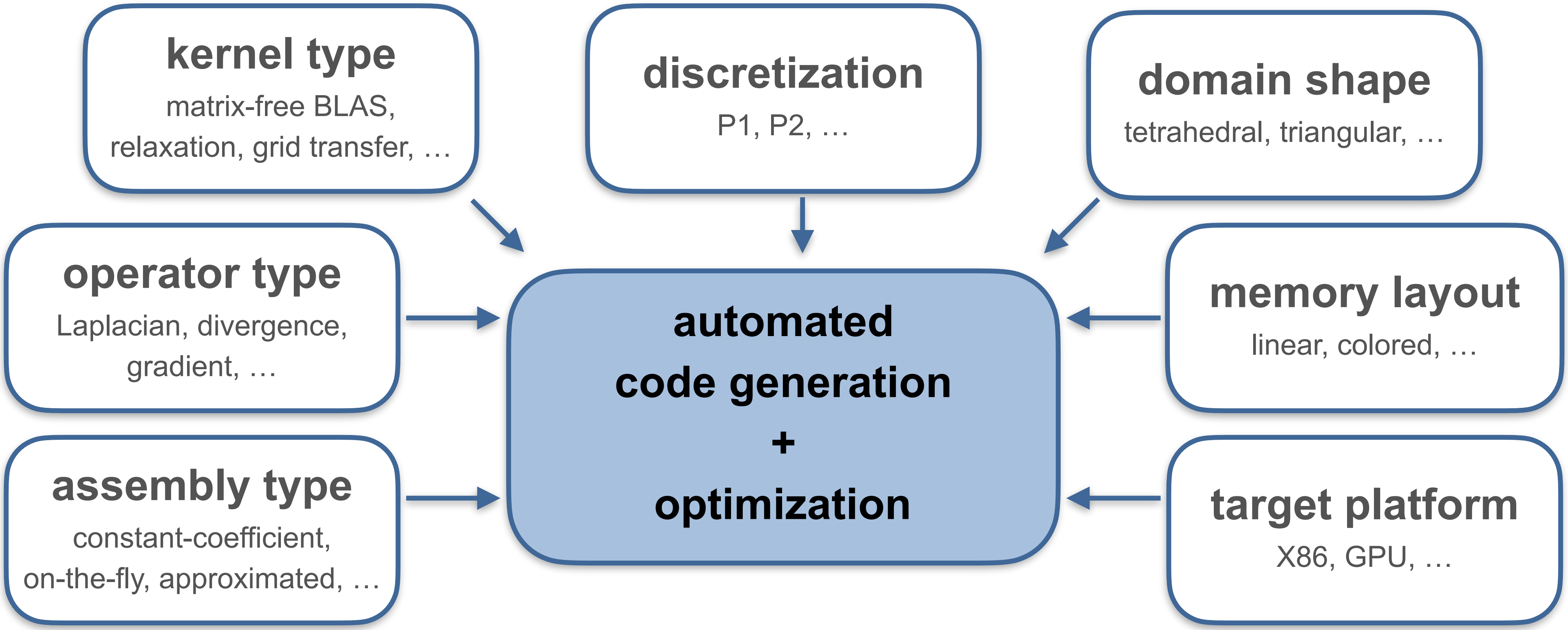
HYTEG
Hybrid Tetrahedral Grids

# Part IVb:

# Automatic Code Generation
# Metaprogramming

# The HyTeG framework - code generation

Combinatorial explosion leads to many different kernels and would require **an enormous manual implementation and optimization effort!**



# Performance Analysis and Code Optimization

### Measurements

- Fritz Supercomputer at NHR@FAU
- Matrix-vector multiplication (without communication)
- Single socket: Intel Xeon Platinum 8360Y ("Ice Lake")
- 36 cores per socket
- LIKWID performance monitoring and benchmarking suite

### Generated optimizations

- Symmetry (S)
- Inter-element vectorization (V)
- Loop invariants (I)
- Cubes loop strategy (C)
- Under-integration (U)
- Fused quadrature loops (fQ)
- Tabulation (T)

→

### Optimization search

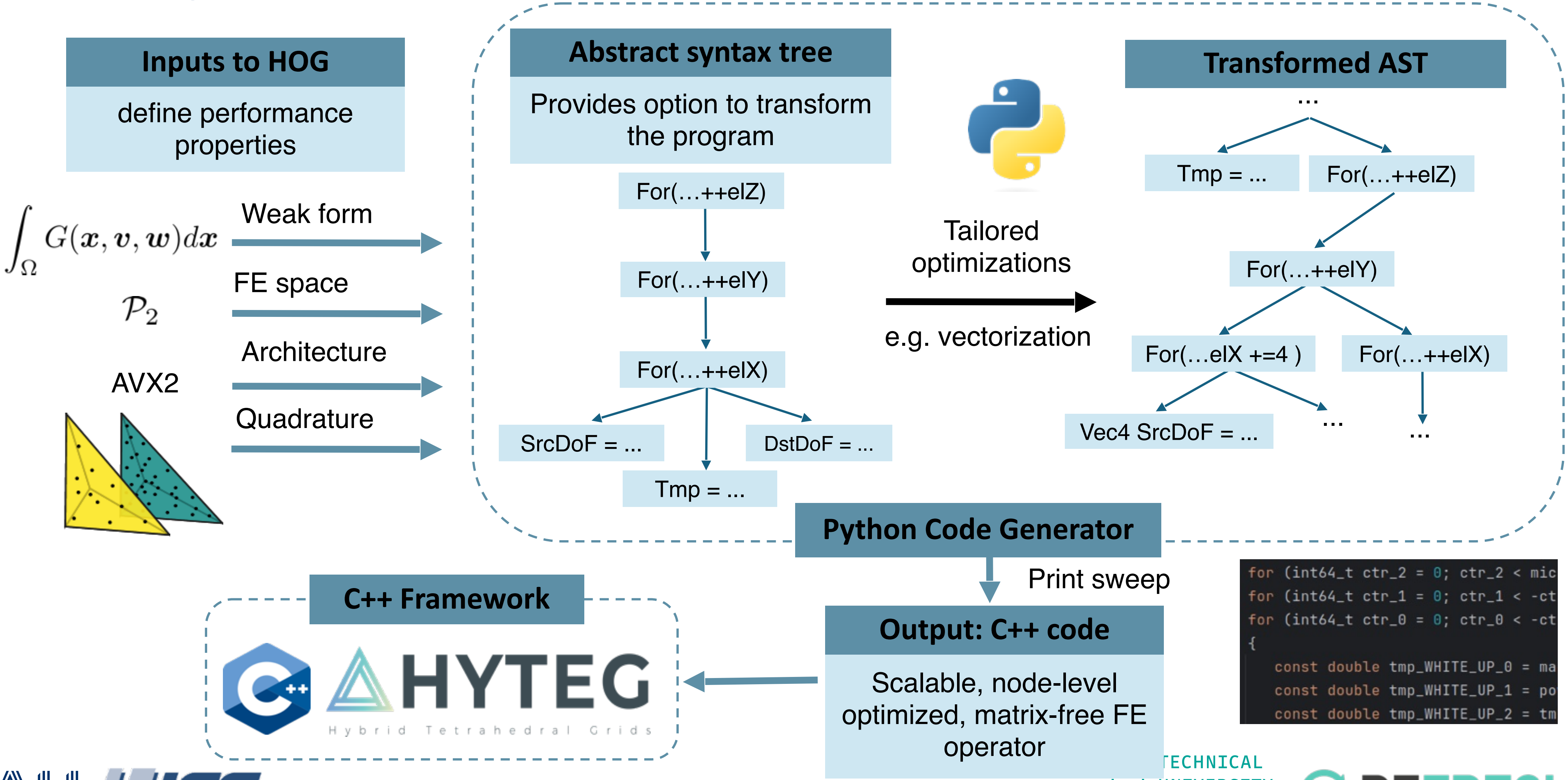
- Generate all combinations
- Determine the **set of most effective optimizations**

→

### Optimization path

For the fastest operator:
Roofline analysis of
optimization path

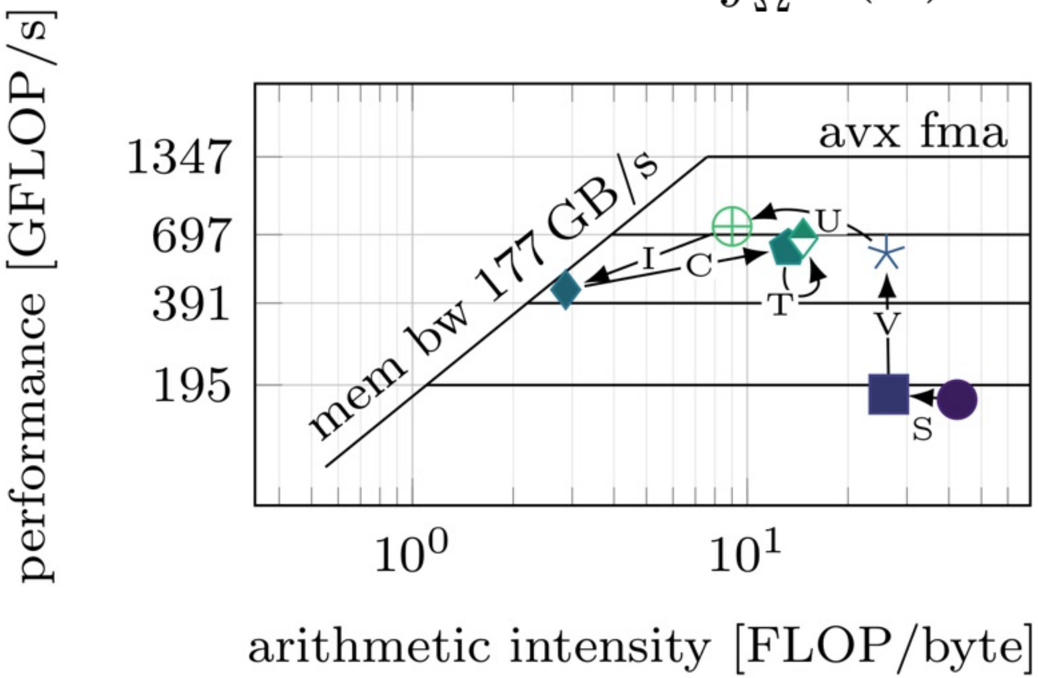
# HyTeG Operator Generator (HOG)

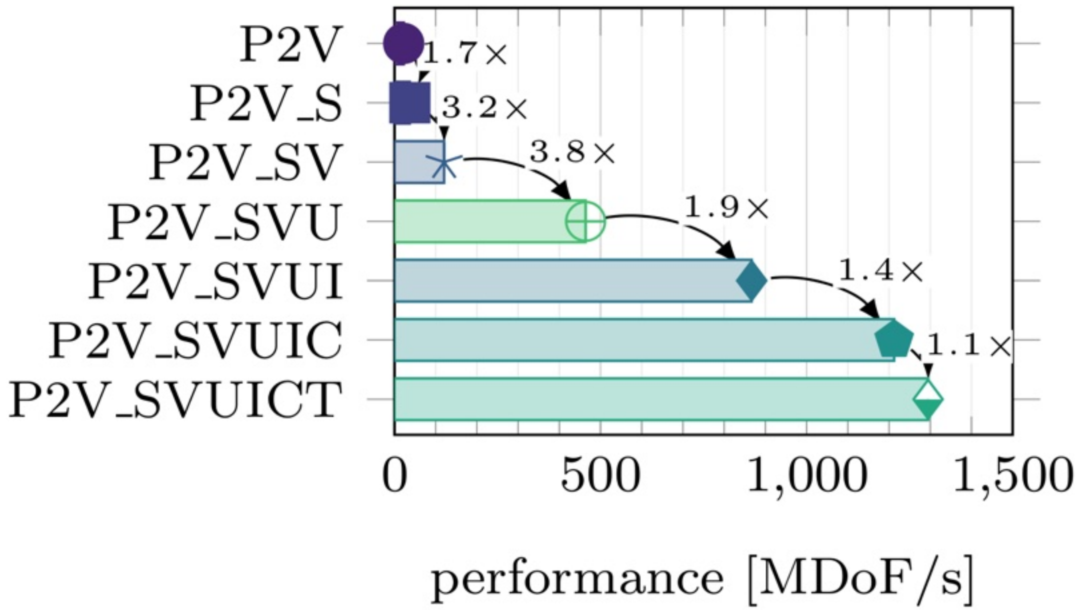


# Optimization Path: P2V

Operator P2V: $\int_\Omega k(\boldsymbol{x}) \nabla v \cdot \nabla w, \mathcal{P}_2$

- Symmetry (S)
- Inter-element vectorization (V)
- Loop invariants (I)
- Cubes loop strategy (C)
- Under-integration (U)
- Fused quadrature loops (fQ)
- Tabulation (T)

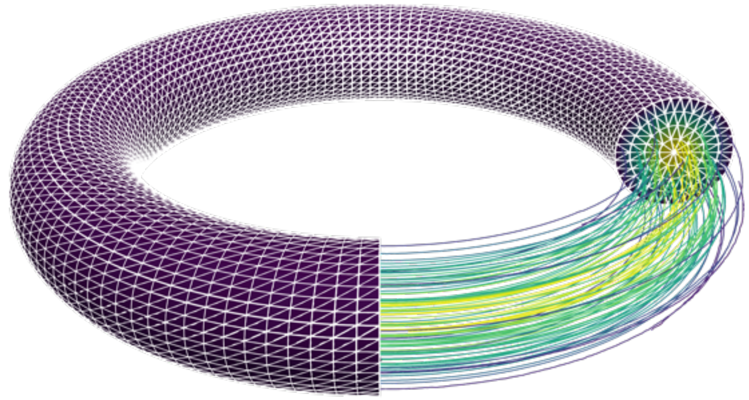


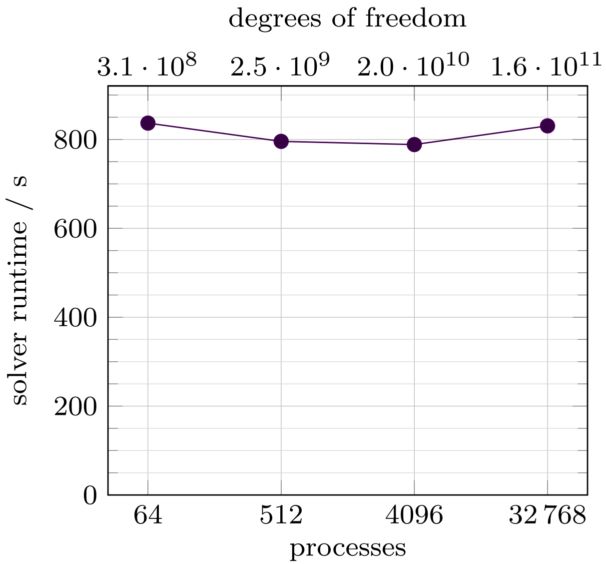

- Starting point: already compute-bound
- Series of opts reducing arithmetic intensity
- Compute-intense **P2V becomes memory-bound** with P2V_SVUI
- Cubes loop applicable -> more speed-up
- **58x accumulated speed-up, 50% peak, 1.4 GDoF/s**

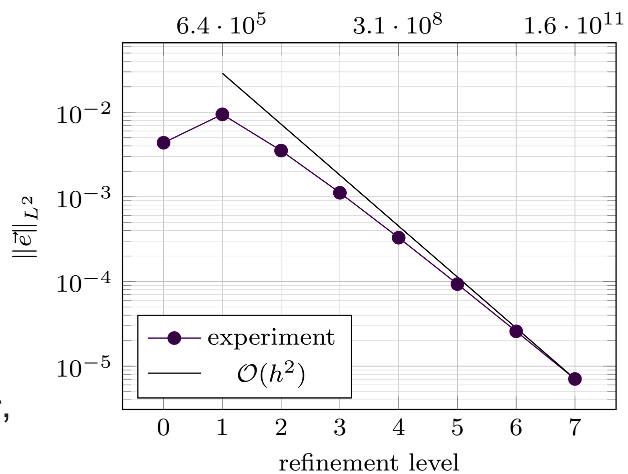
# HyTeG: A for the curl-curl problem

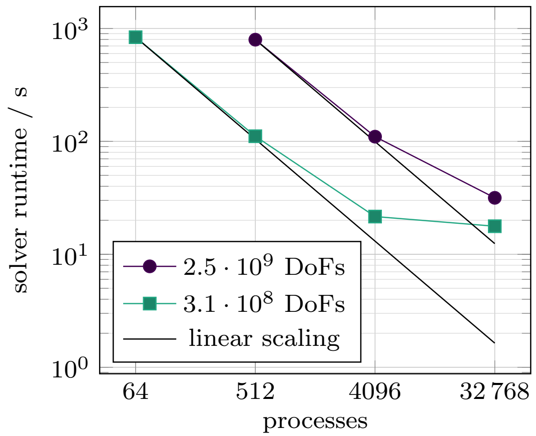
$$\alpha\,\mathbf{curl}\,\mathbf{curl}\,\vec{u} + \beta\vec{u} = \vec{f} \quad \text{in } \Omega,$$
$$\vec{u} \times \vec{n} = 0 \quad \text{on } \partial\Omega,$$



- linear Nédélec elements of the first kind
- 65 280 curvilinear macro-tetrahedra
- total number of DoFs 1.6 x $10^{11}$



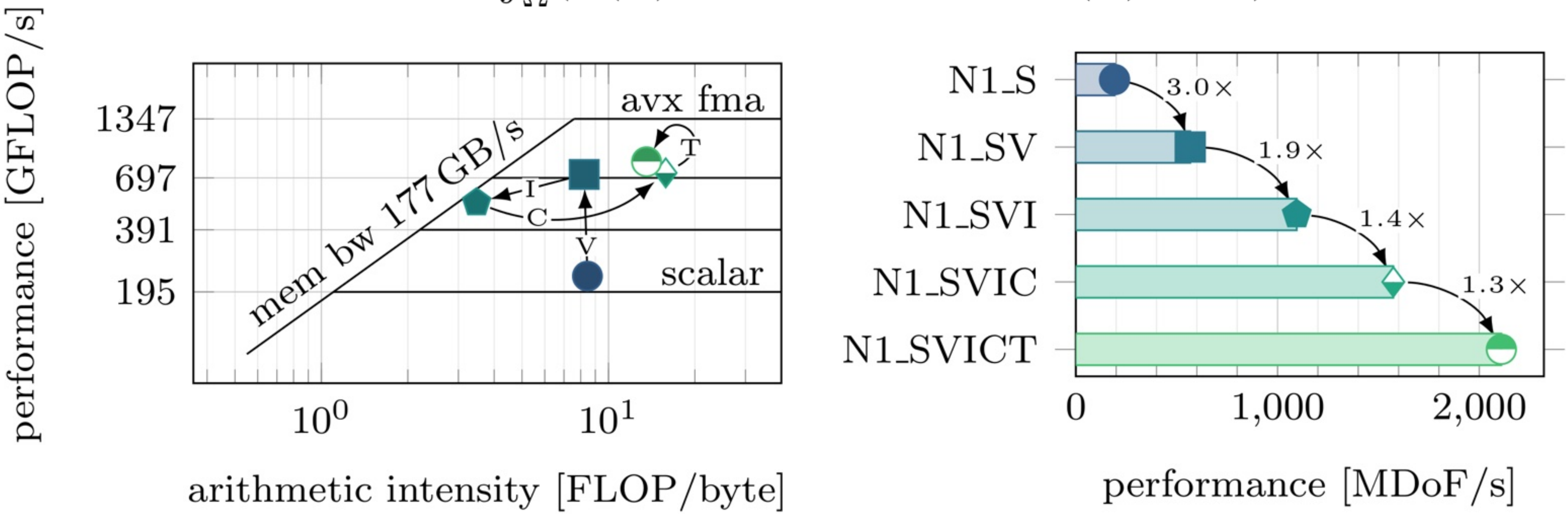





Kohl, N., Bauer, D., Böhm, F., & Rüde, U. (2024). Fundamental data structures for **matrix-free finite elements** on hybrid tetrahedral grids. *International Journal of Parallel, Emergent and Distributed Systems*, *39*(1), 51-74.

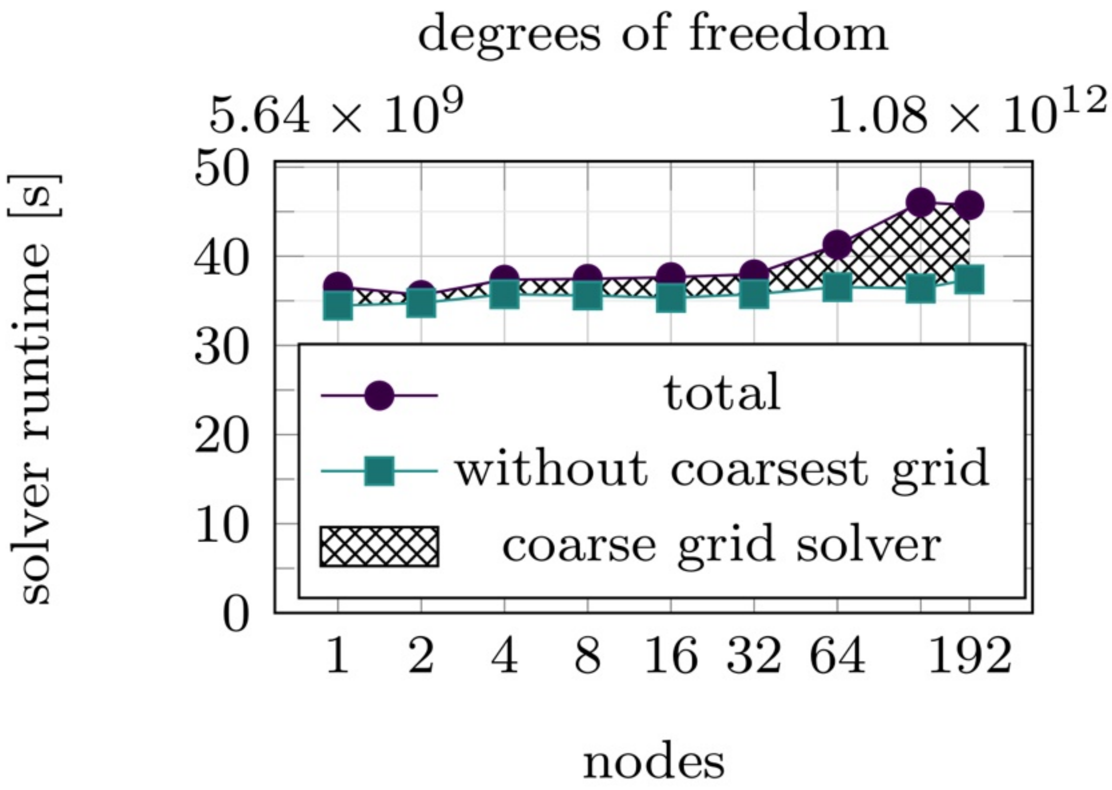
# Optimization Path: N1

Operator N1: $\int_\Omega (\alpha(\boldsymbol{x}) \nabla \times \boldsymbol{v} \cdot \nabla \times \boldsymbol{w} + \beta(\boldsymbol{x}) \boldsymbol{v} \cdot \boldsymbol{w}), \mathcal{ND}_1$




- Vectorization: steep **performance boost**
- Loop invariants: **arithmetic intensity decreases**
- Cubes loop: cache-locality improves**, arithmetic intensity increases**
- **11x accumulated speed-up, 62% peak, >2GDoF/s**

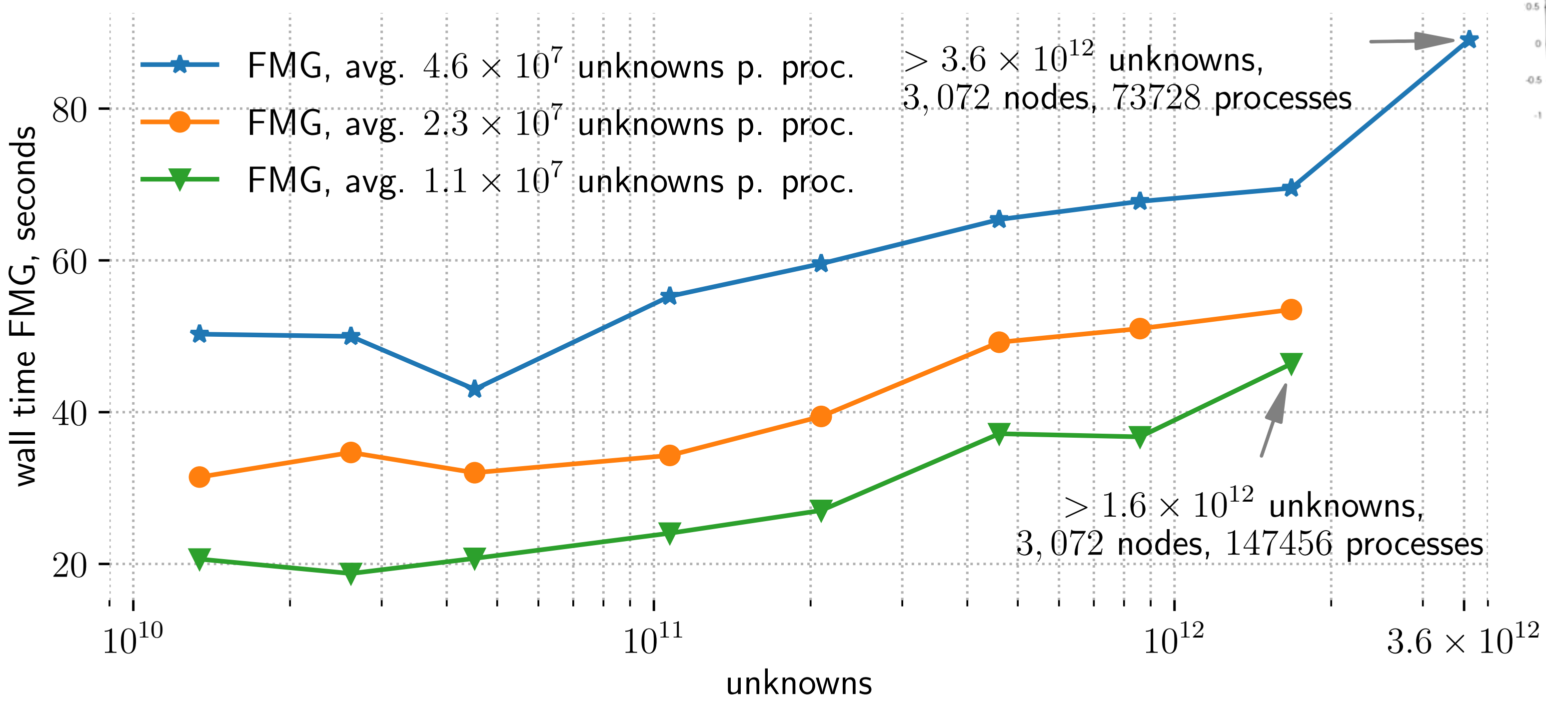
# Weak Scaling N1 to a Trillion DoFs

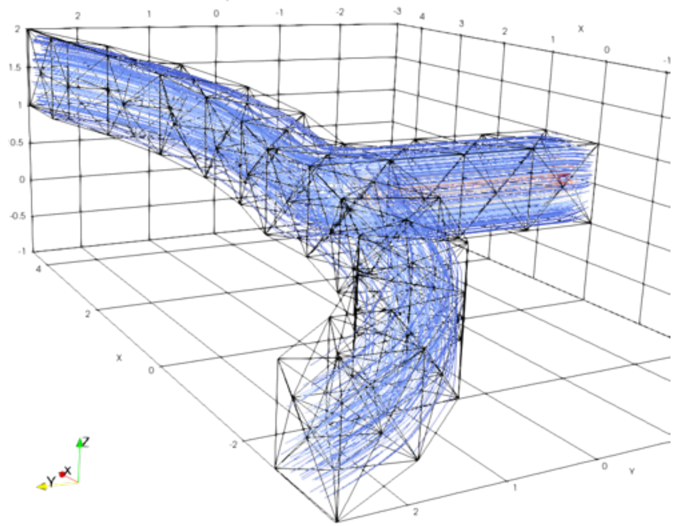



- Full Multigrid with Hiptmair's hybrid smoother
- SuperMUC NG Phase 2
- 21 504 processes
- **Solve 10^12 DoFs linear system in 50 seconds**

Böhm, F., Bauer, D., Kohl, N., Alappat, C., Thönnes, D., Mohr, M., ... & Rüde, U. (2024). Code Generation and Performance Engineering for Matrix-Free Finite Element Methods on Hybrid Tetrahedral Grids. *arXiv preprint arXiv:2404.08371.* (to appear in SISC 225, vol 47, pp. B131-B159)

# HyTeG: Scaling for the Stokes Problem



Kohl, N., & Rüde, U. (2022). **Textbook efficiency**: massively parallel matrix-free multigrid for the Stokes system. *SIAM Journal on Scientific Computing*, *44*(2), C124-C155.

Kohl, N., Mohr, M., Eibl, S., & Rüde, U. (2022). A Massively Parallel Eulerian-Lagrangian Method for Advection-Dominated Transport in Viscous Fluids. *SIAM Journal on Scientific Computing*, *44*(3), C260-C285.

$$\frac{\mathfrak{W}(\mathrm{MG})}{\mathfrak{W}(A)} < 10$$

# Part IVc

# Textbook Multigrid Efficiency

# Textbook Multigrid Efficiency (TME)

„Textbook multigrid efficiency means solving a discrete PDE problem with a computational effort that is only a small (less than 10) multiple of the operation count associated with the discretized equations itself.“

[Brandt, 98]

This is a programmatic claim - not a theorem.

For which types of PDE is it achievable?

## Work unit (WU)

- Linear system $Ax = b$
- Work unit (WU) to apply operator: $1\mathrm{WU} := \mathfrak{W}(A)$
  - or perform one sweep of relaxation
- TME achieved, if work for MG solver(!) less than 10 WU:

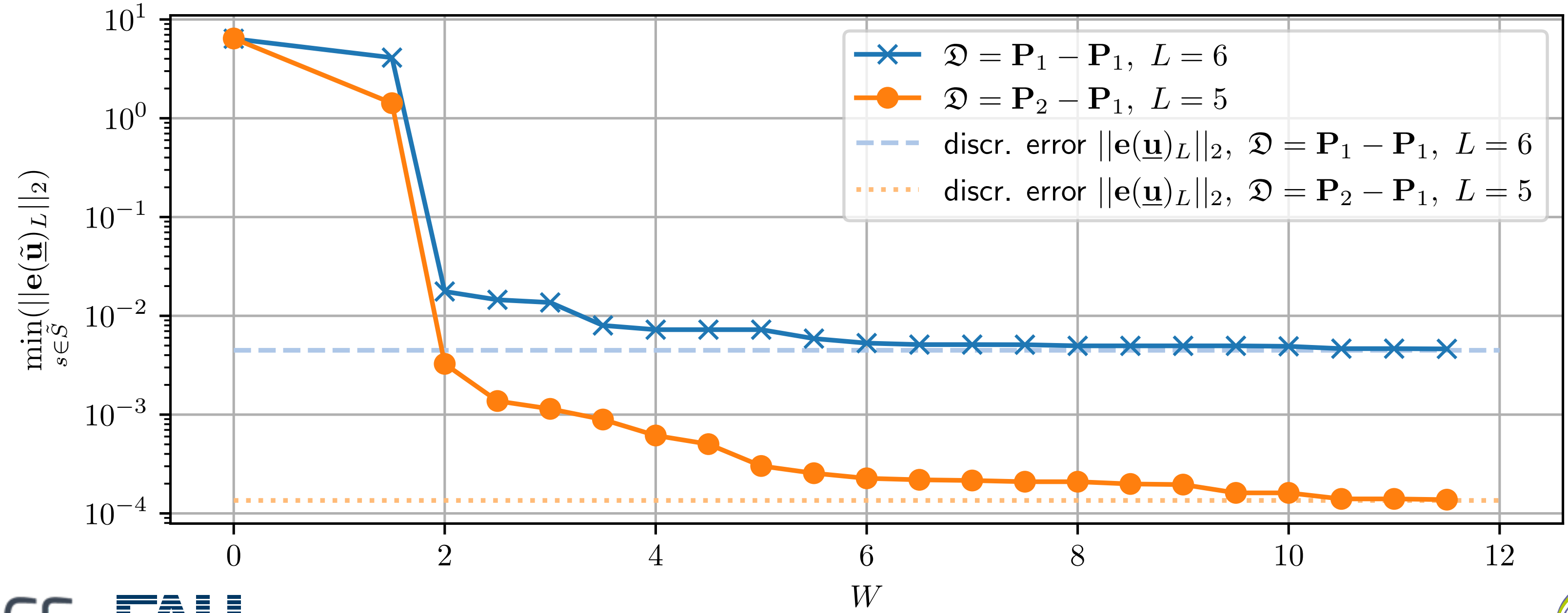
$$\frac{\mathfrak{W}(\mathrm{MG})}{\mathfrak{W}(A)} < 10$$

- TME defined wrt. to underlying differential equation
- TME is (much!) more ambitious than asymptotic optimality or mesh independent convergence of an iterative solver
- TME requires to quantify the constant
  - Hard to assess theoretically
  - But systematic numerical studies possible

# Cost comparison for Stokes with stabilized P1-P1 vs. P2-P1

$$\lim_{\ell\to\infty} \frac{\mathfrak{W}(\mathbf{A}_\ell^{\mathbf{P}_2-\mathbf{P}_1})}{\mathfrak{W}(\mathbf{A}_{\ell+1}^{\mathbf{P}_1-\mathbf{P}_1})} = \frac{23}{12}, \qquad \lim_{\ell\to\infty} \frac{\mathfrak{W}(\mathbf{B}_\ell^{\mathbf{P}_2-\mathbf{P}_1})}{\mathfrak{W}(\mathbf{B}_{\ell+1}^{\mathbf{P}_1-\mathbf{P}_1})} = \frac{13}{24} \qquad \lim_{\ell\to\infty} \frac{\mathfrak{W}(\mathcal{A}_\ell^{\mathbf{P}_2-\mathbf{P}_1})}{\mathfrak{W}(\mathcal{A}_{\ell+1}^{\mathbf{P}_1-\mathbf{P}_1})} = \frac{9}{10}$$

- A WU for P2-P1 and for P1-P1 are roughly equivalent
- Velocity error after an FMG iteration with parameterization chosen to achieve minimal error



# With this let's come back to:

## What is the fastest solver for Poisson's equation?

# Algorithms Matter!

- Solution of Laplace equation in 3D wit $N=n^3$ unkowns
- Direct methods:
  - banded: $\sim n^7 = N^{2.33}$
  - nested dissection: $\sim n^6 = N^2$
- Iterative Methods:
  - Jacobi: $\sim 50\ n^5 = 50\ N^{1.66}$
  - CG: $\sim 100\ n^4 = 100\ N^{1.33}$

[redacted]

| Energy per FLOP: 1nJ | | | | |
|---|---|---|---|---|
| Computer Generation | gigascale: $10^9$ | terascale: $10^{12}$ | petascale: $10^{15}$ | exascale: $10^{18}$ |
| problem size: DoF=N | $10^6$ | $10^9$ | $10^{12}$ | $10^{15}$ |
| Direct method: $1*N^2$ | 0.278 Wh | 278 kWh | 278 GWh | 278 PWh |
| Krylov method: $100*N^{1.33}$ | 10 Ws | 28 Wh | 278 kWh | 2.77 GWh |
| [redacted] | | | | |
| TerraNeo prototype (est. for Juqueen) | 0.13 Wh | 30 Wh | 27 kWh | ? |

# References from the stone age of multigrid research

[ST] Stüben, K., & Trottenberg, U. Multigrid methods: Fundamental algorithms, model problem analysis and applications, in vol. 960 of Lecture Notes in Mathematics. Springer Verlag, 1982

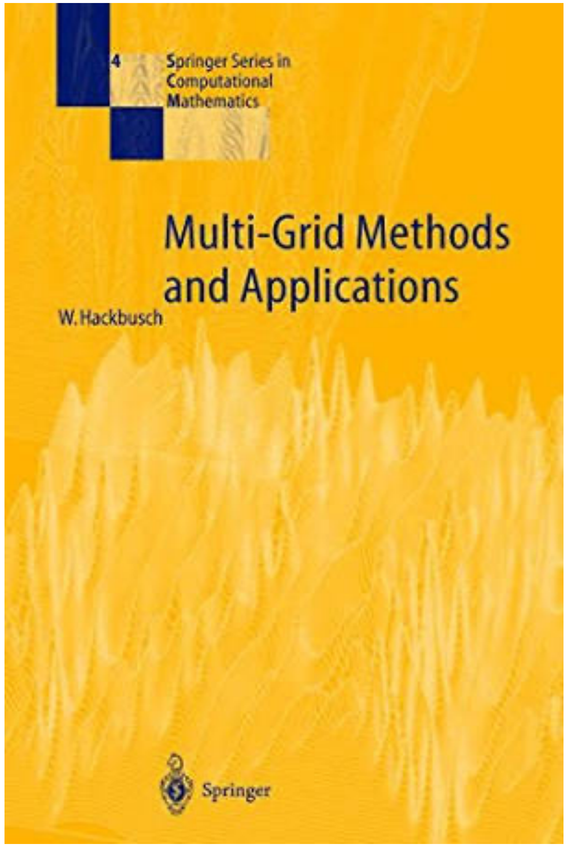
This is in the proceedings of the 1st European conf on multigrid methods that was held in Köln in 1981.
This volume also contains Brandt's original „Multigrid Guide".

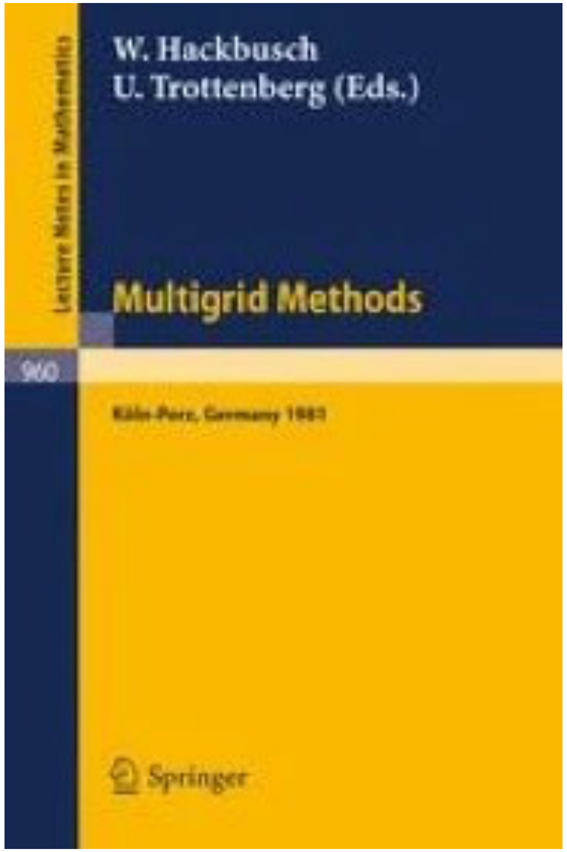
[Hac] W. Hackbusch: Multi-grid methods and applications, 1985, Springer Berlin, ISBN 3-540-12761-5

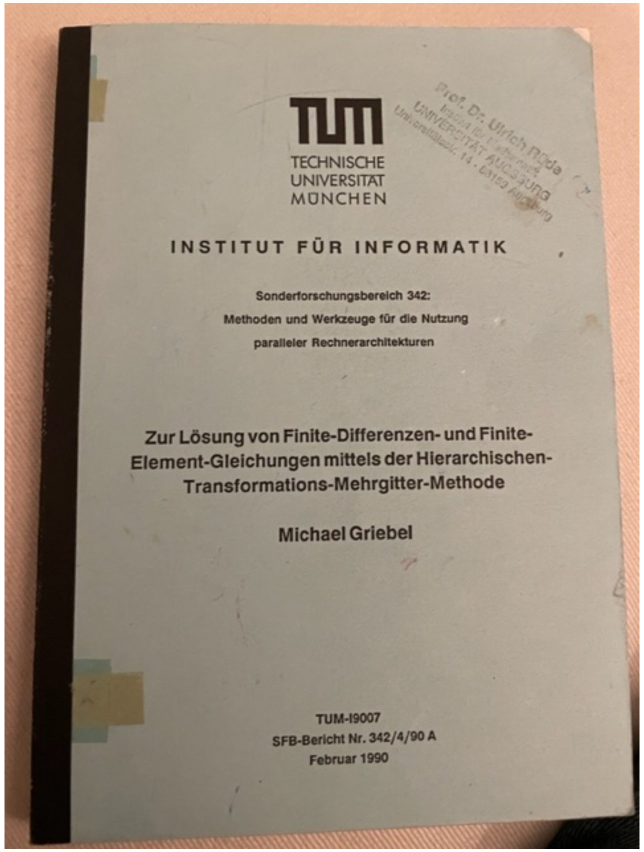
[Gri] M. Griebel. Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen Transformations-Mehrgitter-Methode. Technical Report, SFB Bericht 342/4/90 A, Institut für Informatik, TU München, 1990

This is the dissertation of the author, submitted and defended in 1989

# The three references




- [ST] and [Hac] are easily accessible online, [Gri] I have as physical copy

# Work estimates from [ST] for 5-pt discretization of Poisson's eq 2-grid-method with red-black Gauss-Seidel smoothers

| | | $I_h^{2h}$ : FW | | | $I_h^{2h}$ : HW | | |
|---|---|---|---|---|---|---|---|
| $\nu$ | $(\mu^*)^\nu$ | $\rho^*$ | # Add | # Mult | $\rho^*$ | # Add | # Mult |
| 1 | 0.250 | 0.250 | 6.75 | 2.25 | 0.500 | 5.5 | 1.75 |
| 2 | 0.063 | $\boxed{0.074}$ | $\boxed{9.75}$ | $\boxed{3.25}$ | 0.125 | 8.5 | 2.75 |
| 3 | 0.034 | 0.053 | 12.75 | 4.25 | $\boxed{0.034}$ | $\boxed{11.5}$ | $\boxed{3.75}$ |
| 4 | 0.025 | 0.041 | 15.75 | 5.25 | 0.025 | 14.5 | 4.75 |

Table 8.1a: $\mu^*$, $\rho^*$ and computational work $W_h^{2h}/\mathcal{N}_h$ in case of smoothing by RB relaxation (for 5-point Laplace discretization)

Once the dust has been wiped off, this is still healthy, good, solid numerics

$\mu^*(\nu)$ smoothing factor

$\rho^*(\nu)$ Asymptotic 2-grid convergence factor

Half-weighting restriction (HW) $I_h^{2h} \hat{=} \frac{1}{8}\begin{bmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{bmatrix}_h^{2h}$

# And what is achieved by this

All following quantitative results refer to Poisson's equation and the MGØ1 version described above with

$$\nu_1 = 2, \; \nu_2 = 1. \qquad (10.2)$$

If $\mathcal{N}$ denotes the number of grid points of $\Omega_h$, the total computational work for one iteration step of the corresponding method is less than

$$\boxed{\begin{array}{lll} 15\mathcal{N} \text{ additions}, & 5\,\mathcal{N} \text{ multiplications} & \text{(for V-cycles)}, \\ 23\mathcal{N} \text{ additions}, & 7.5\mathcal{N} \text{ multiplications} & \text{(for W-cycles)}, \end{array}} \qquad (10.3)$$

neglecting lower order terms. These numbers are independent of the shape of the domain.

V(2,1)-cycle: 20 Flops/unknown
W(2,1)-cycle 30.5 Flops/unknown

# … and if we use full multigrid (FMG)?

The total computational work of MGØ1 in the FMG version (r=1) is less than

$$\boxed{\begin{array}{ll} 22\mathcal{N} \text{ additions}, \quad 8\mathcal{N} \text{ multiplications} & \text{(if V-cycles are used)}, \\ 32.5\mathcal{N} \text{ additions}, \quad 11.5\mathcal{N} \text{ multiplications} & \text{(if W-cycles are used)} \end{array}} \qquad (10.5)$$

- Summarizing: We should be solving the 2D Poisson equation
  - to discretization error accuracy
  - **with 30 Flops per unknown!**
  - in the model case, FMG-V(2,1) cycles are enough to achieve asymptotic optimality

# So, what is the cost of solving the discrete Poisson equation?

- What is the best constant published?
  - For Poisson 2D, second order:
    #Flops ~ **30 n** (Stüben, 1982)
- assume computer with 1 PetaFLOPS, $n=10^8$
  - expected time to solution: Poisson 2D
    **$3*10^{-6}$ sec** (microseconds!)
- assume computer with 1 ExaFLOPS, $n=10^{12}$
  - expected time to solution: Poisson 2D
    **$30*10^{-6}$ sec** (30 micro-seconds!)
- standard computational practice in 2024 misses this by **several orders of magnitude**!
- Why this huge **gap** between theory and practice? Do we need a failure analysis?
- Related questions:
  - Cost of complex discretizations?
  - Has the deflation of computational cost lured us into mis-developments?

## Intermediate Conclusion and Outlook

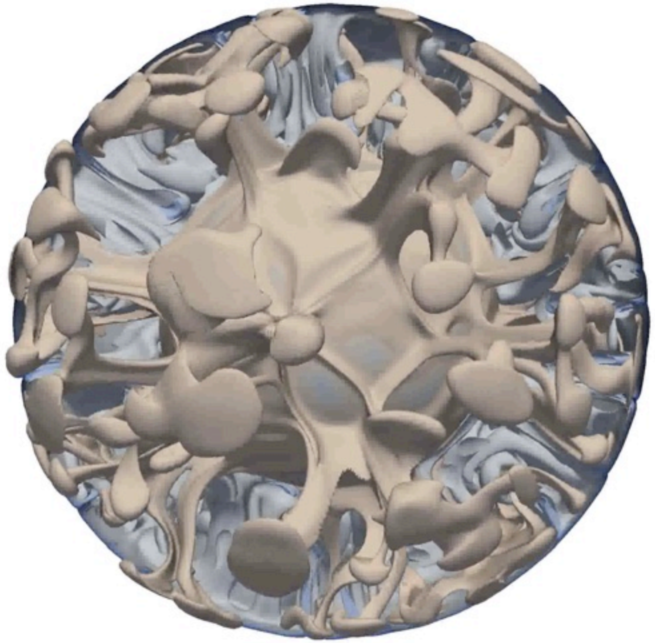
### Multigrid scales!

**HHG** (since 2000):

- prototype implementation, reaching $10^{13}$ DOF
- concepts are suitable and efficient
- limited to linear elements

**HyTeG** (since 2018):

- sustainable, flexible software architecture
- implements core concepts of HHG
- advanced discretizations
- What about GPUs?

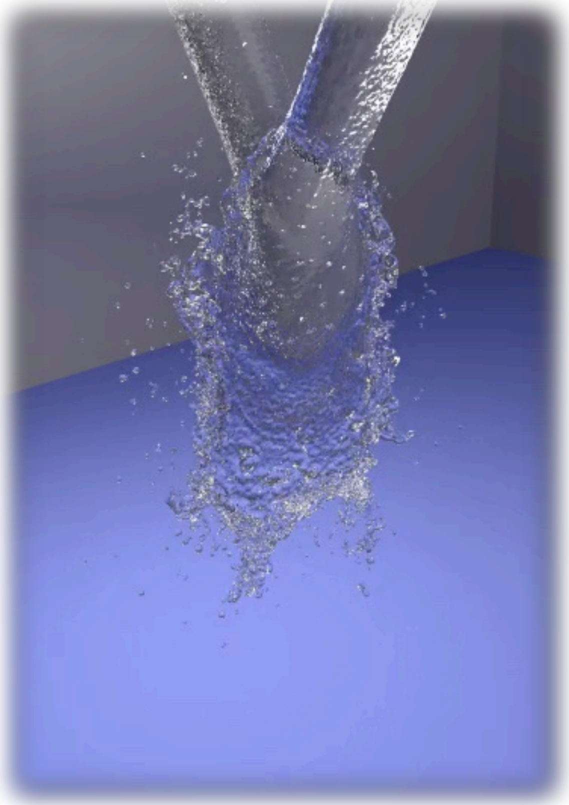


6.5 billion unknowns
10000 time steps
compute time: 7 days @ 288 cores

**Links:**

- terraneo.fau.de
- pypi.org/project/pystencils

## Part V: waLberla

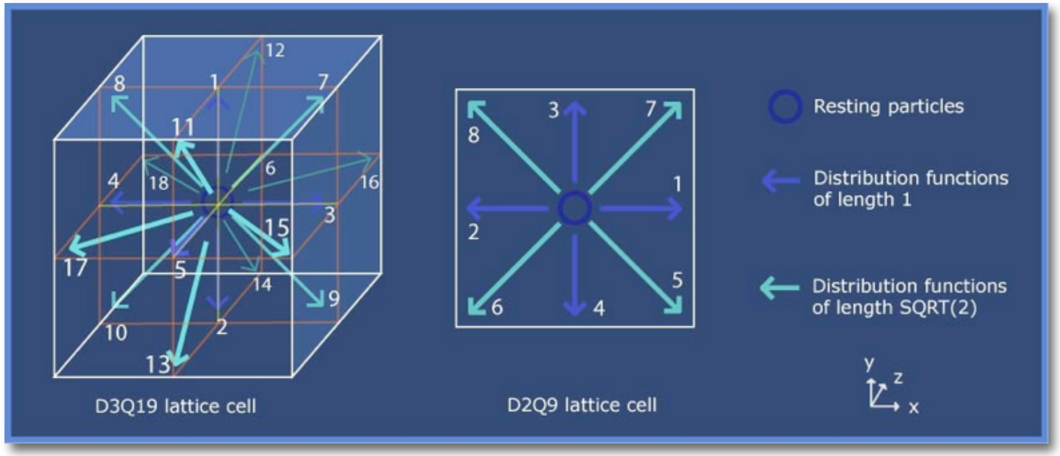


Feichtinger, C., Donath, S., Köstler, H., Götz, J., & Rüde, U. (2011). WaLBerla: HPC software design for computational engineering simulations. *Journal of Computational Science*, *2*(2), 105-112.

Bauer, M., Eibl, S., Godenschwager, C., Kohl, N., Kuron, M., Rettinger, C., ... & Rüde, U. (2021). waLBerla: A block-structured high-performance framework for multiphysics simulations. *Computers & Mathematics with Applications*, *81*, 478-501.

# Eulerian: Lattice-Boltzmann-Method

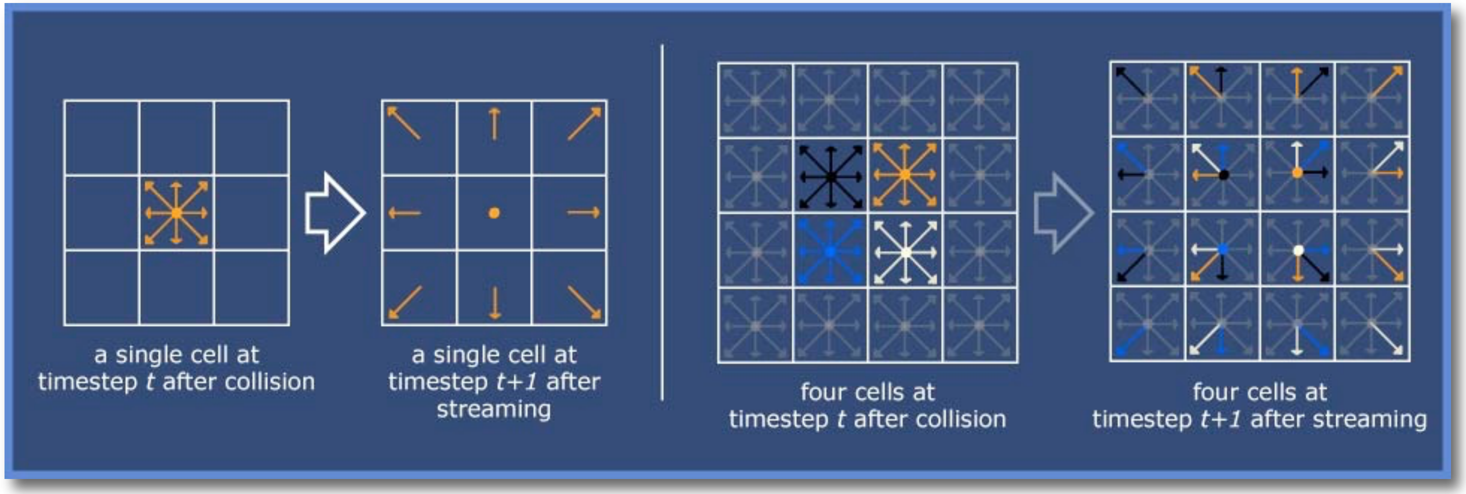
- Discretization in squares or cubes (cells)
- Common examples for particle distribution functions (PDF)
  - in 2D: 9 numbers (2DQ9)
  - in 3D: D3Q19 (alternatives D3Q27, etc)




$$\vec{f} = (f_0, f_1, ..., f_{N-1})$$

# The stream step

Move PDFs
into neighboring cells




$$\underbrace{f_i(\vec{x}+\vec{e_i},t+1)-f_i(\vec{x},t)}_{\substack{\textit{Non-local}\text{ part,}\\ \textit{Linear}\text{ propagation to neighbors}\\ \textit{(stream step)}}} = \underbrace{C_i(\vec{f}(\vec{x},t))}_{\substack{\textit{Local}\text{ part,}\\ \textit{Non-linear}\text{ operator,}\\ \textit{(collide step)}}}$$

# The collide step

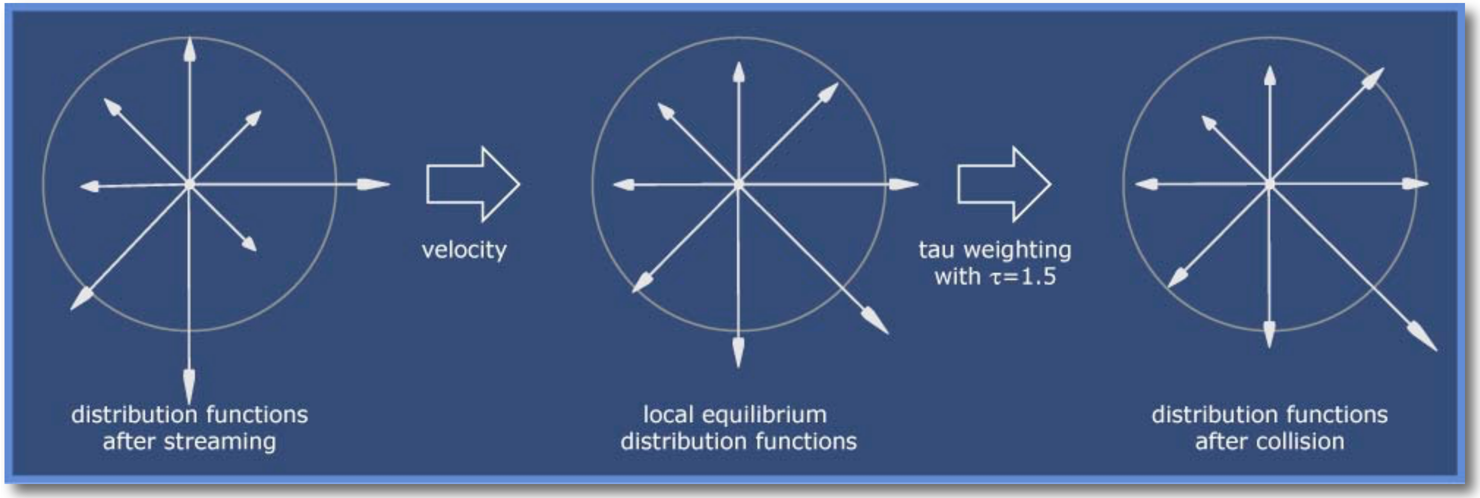
Compute new PDFs modeling molecular collisions

Most collision operators can be expressed as

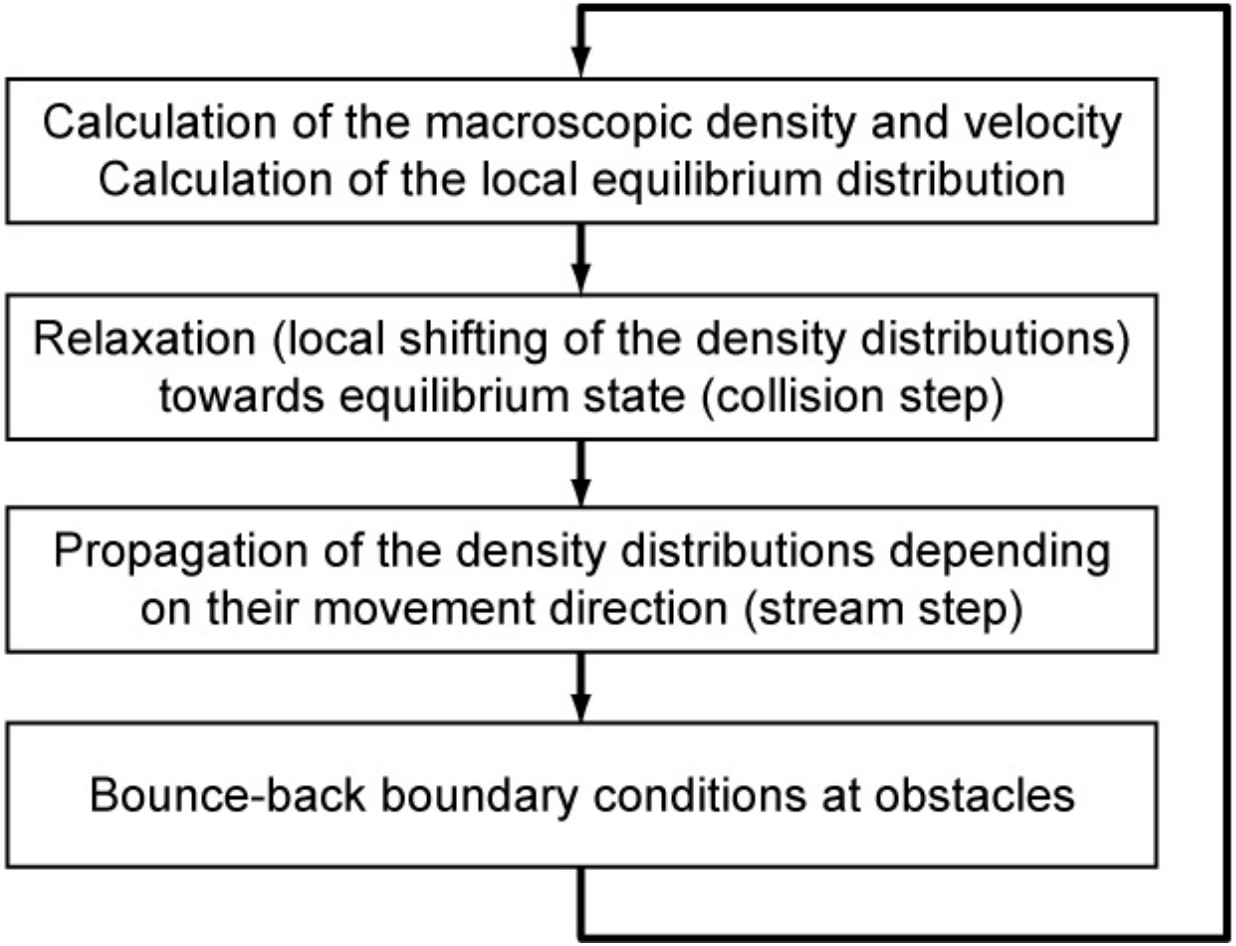
$$C_i = A_{ij}[f_j(\vec{x},t) - f_j^{eq}(\vec{x},t)].$$

Equilibrium function: non-linear,
depending on the conserved moments $\rho$, $\vec{u}$, and $\vec{f}$.

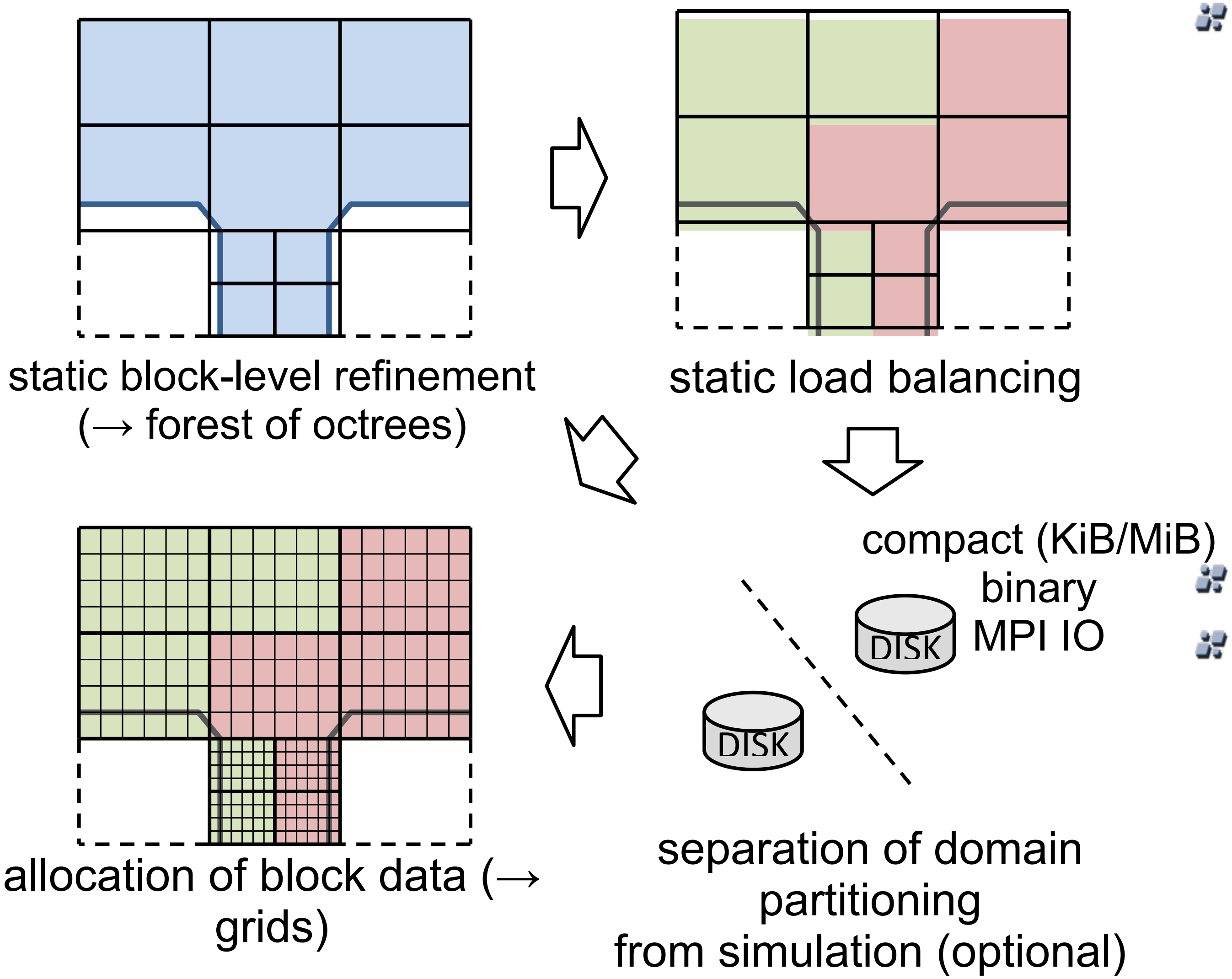
$$\vec{f}^{eq}(\vec{x},t) = \vec{f}^{eq}(\rho(\vec{x},t), \vec{u}(\vec{x},t))$$



# The Lattice Boltzmann Algorithm



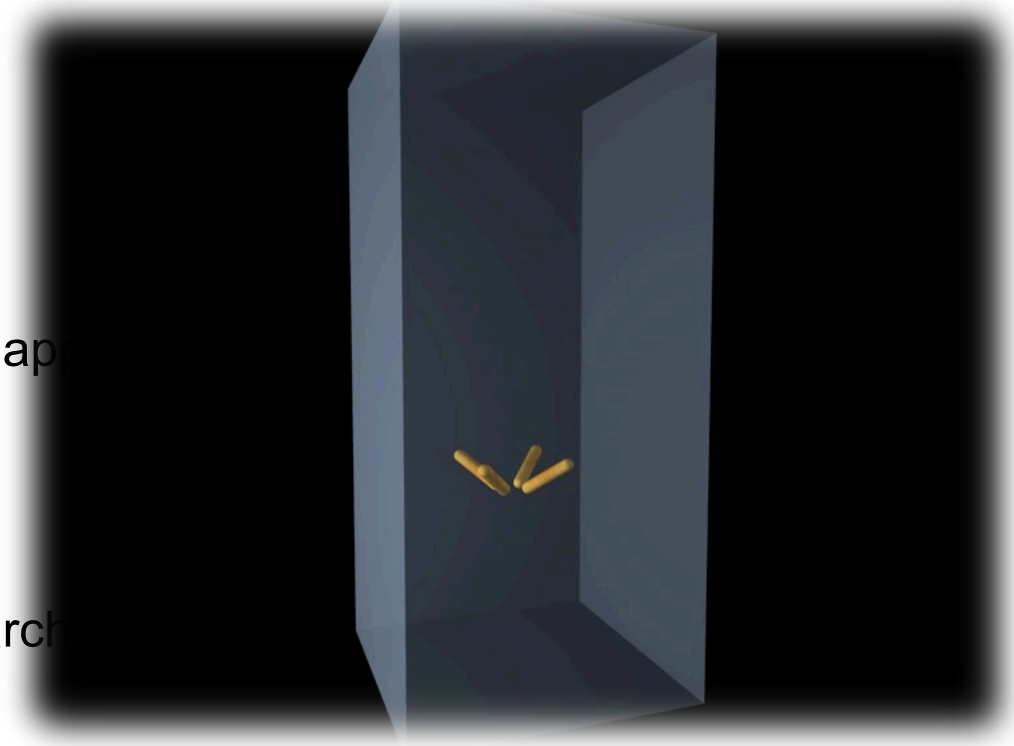
# waLBerla exascale SW structure



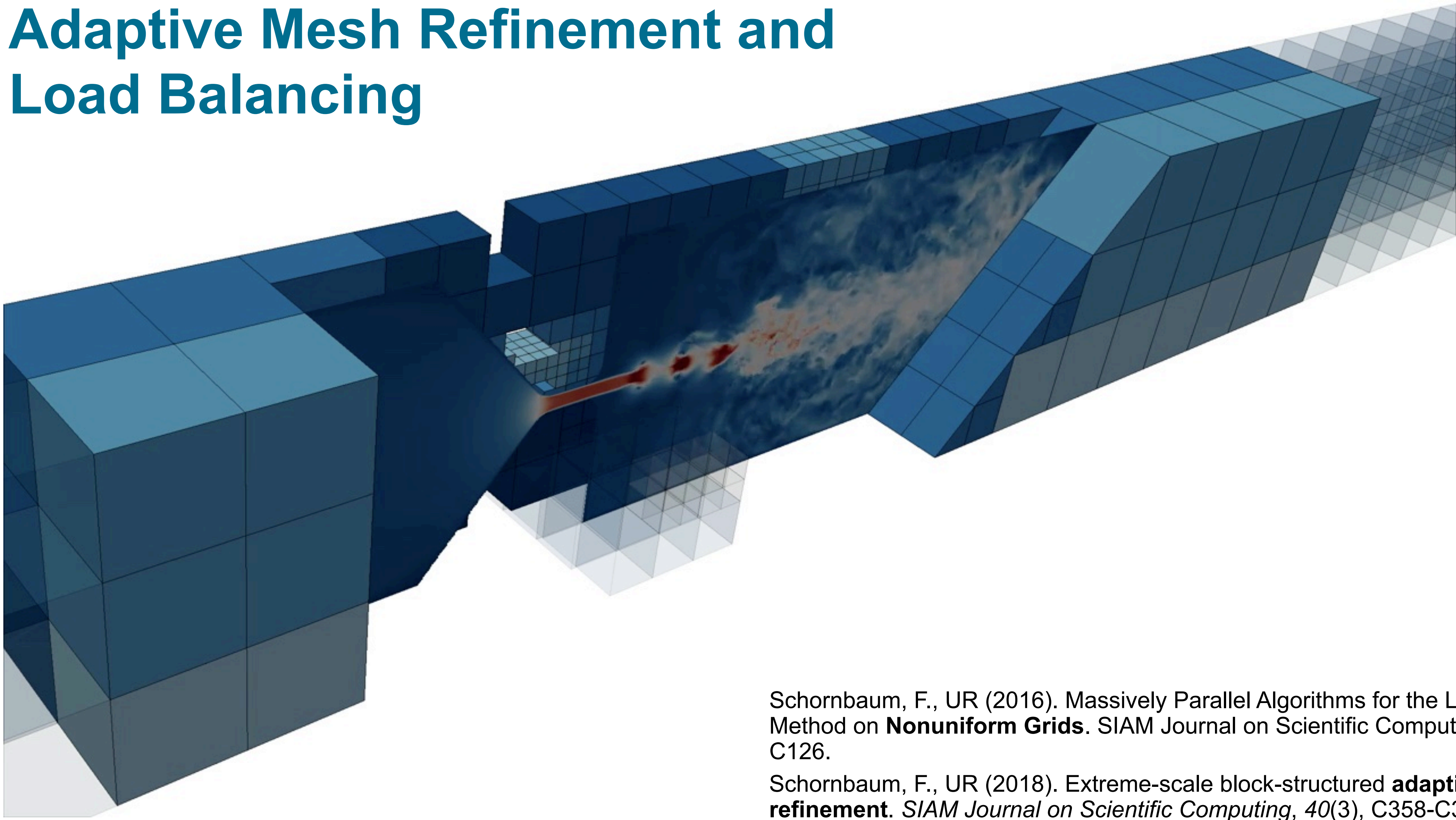

- Framework based on domain partitioning
  - Block structured grids (good for vectorization, GPUs)
  - Lagrangian co-simulation (fluid-structure interaction)
  - Forest of octrees for load balancing
- Hybrid parallelization
- Automatic program generation
  - DSL Pystencils for stencil codes
  - DSL LBMpy for Lattice Boltzmann

## waLBerla concepts

- LBM based multiphysics **framework**
  - not an application software
  - not a library
  - But: a toolbox to support the construction of ap
- Design goals
  - scalability
  - node level efficiency
  - performance portability for a wide range of arch
- Block structured meshes
- Coupling functionality
  - Eulerian: LBM (FV, FE)
  - Lagrangian: Particles



# Adaptive Mesh Refinement and Load Balancing

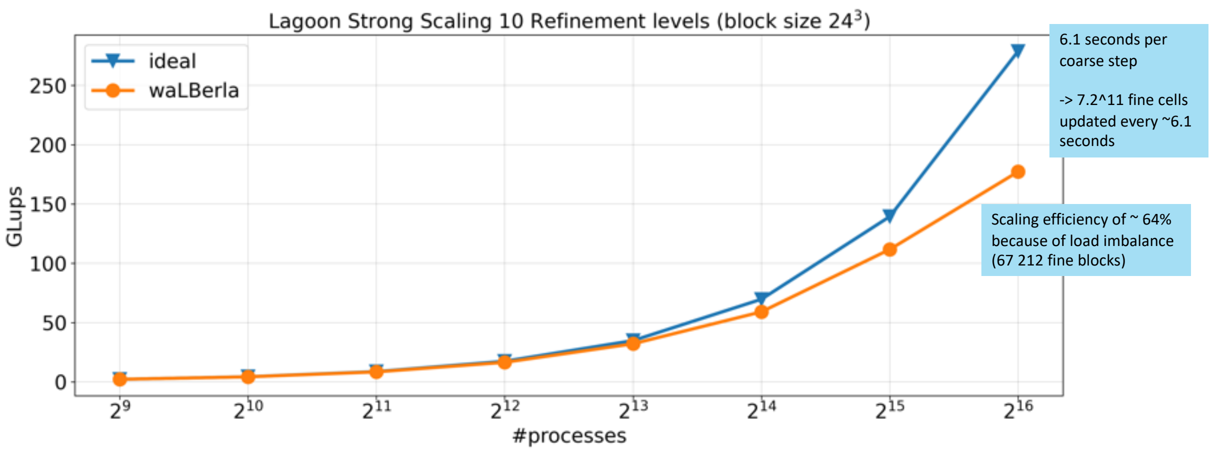


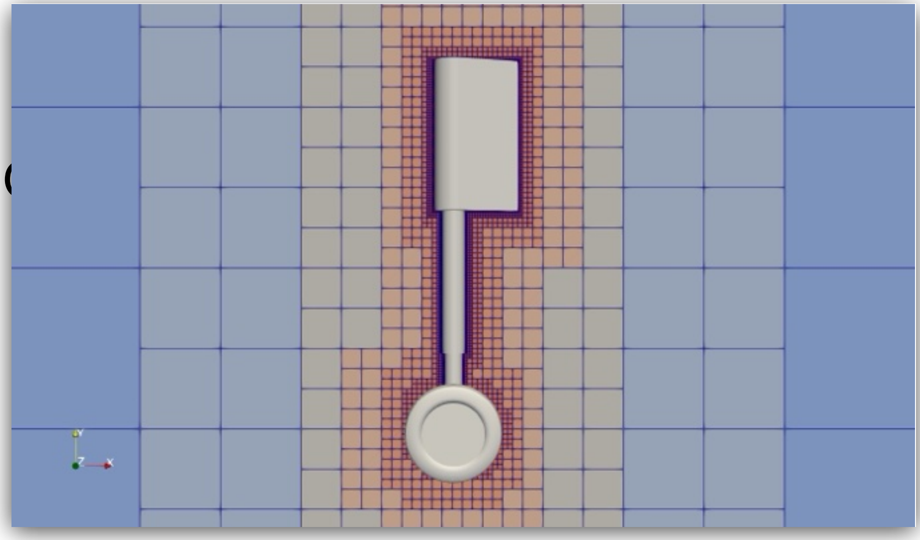
Schornbaum, F., UR (2016). Massively Parallel Algorithms for the Lattice Boltzmann Method on **Nonuniform Grids**. SIAM Journal on Scientific Computing, 38(2), C96-C126.

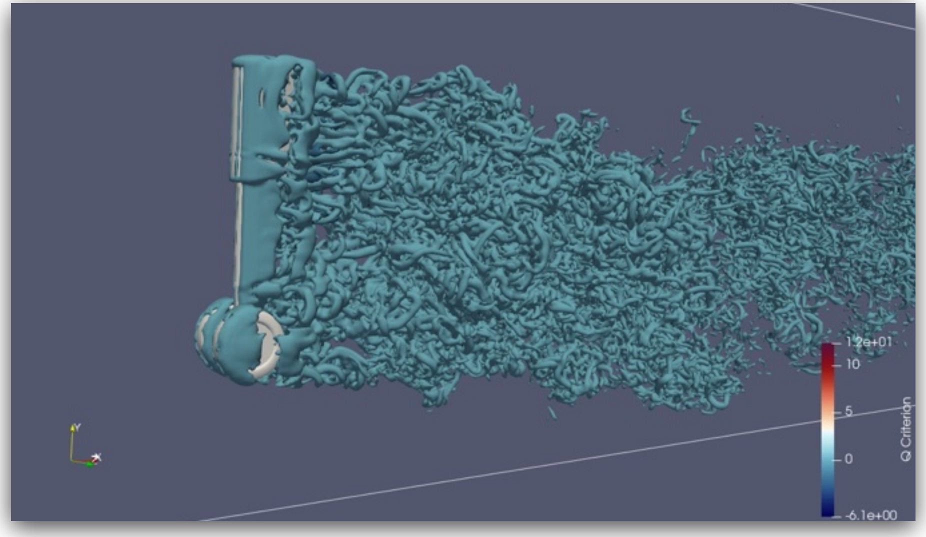
Schornbaum, F., UR (2018). Extreme-scale block-structured **adaptive mesh refinement**. *SIAM Journal on Scientific Computing*, *40*(3), C358-C387.

# SCALABLE

- EuroHPC consortium including Airbus, FAU, IT4I (
  CS Group, and CERFACS
- LBM for aerodynamics/aeroacoustics
- Up to 10 refinement levels
- Scaling up to 65000 cores
- 1.3 x $10^9$ cells
- Resolution of 0.25 mm around landing gear

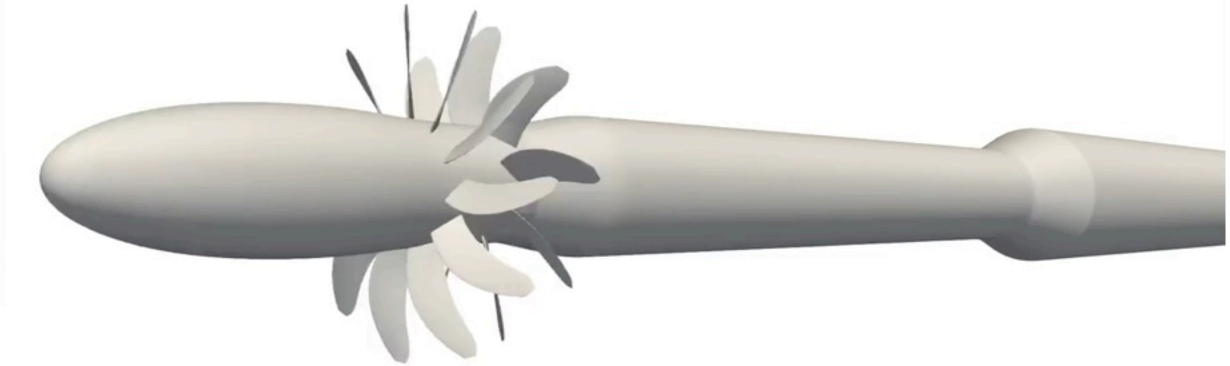




# SCALABLE

- Test case of counter rotating rotor
- Partially saturated cells (PSM) method for moving boundaries/geometry
- realistic Re-number still problematic
- Adaptive refinement in parallel under development (refine/coarsen)



Visualization/ performance optimization by P. Suffa, presented at DSFD conference. Best paper award

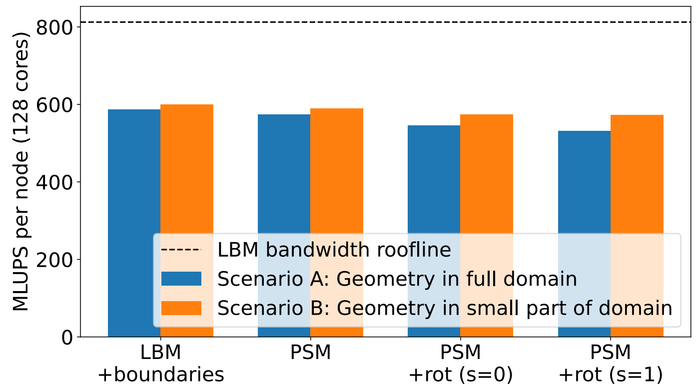



FIG. 10: Node level performance of PSM vs LBM on 2x AMD EPYC 7763 CPUs on LUMI-C with 128 blocks and $64^3$ cells per block. Scenario A and B are visualized in Figure 9.

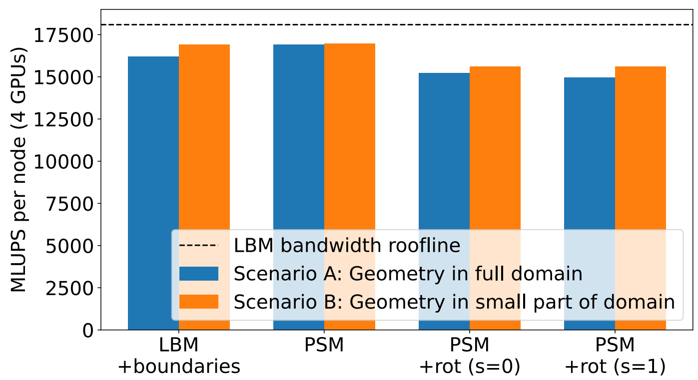



FIG. 11: Node level performance of PSM vs LBM on 4 NVIDIA A100 GPUs on JUWELS-Booster with 1 block per GPU and $256^3$ cells per block. Scenario A and B are visualized in Figure 9.

- Scalability tests with CROR benchmark: CPU cluster and GPU cluster
- Small test: $10^8$ cells, Large test: $4 \times 10^9$ cells

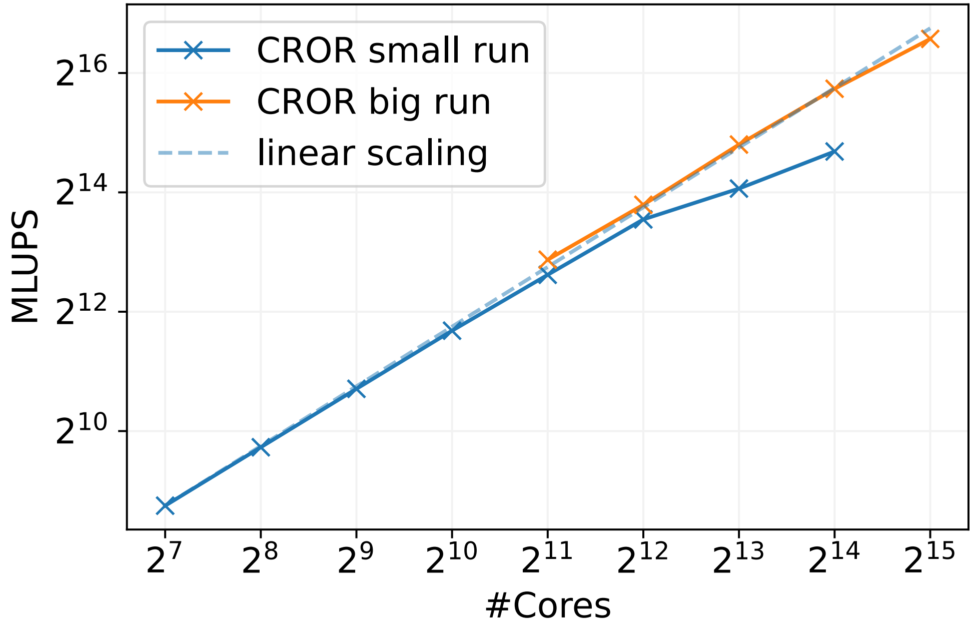



FIG. 12: Strong scaling of the CROR simulation on the CPU partition of LUMI (AMD EPYC 7763 CPU) for the two problem sizes in Table IV.

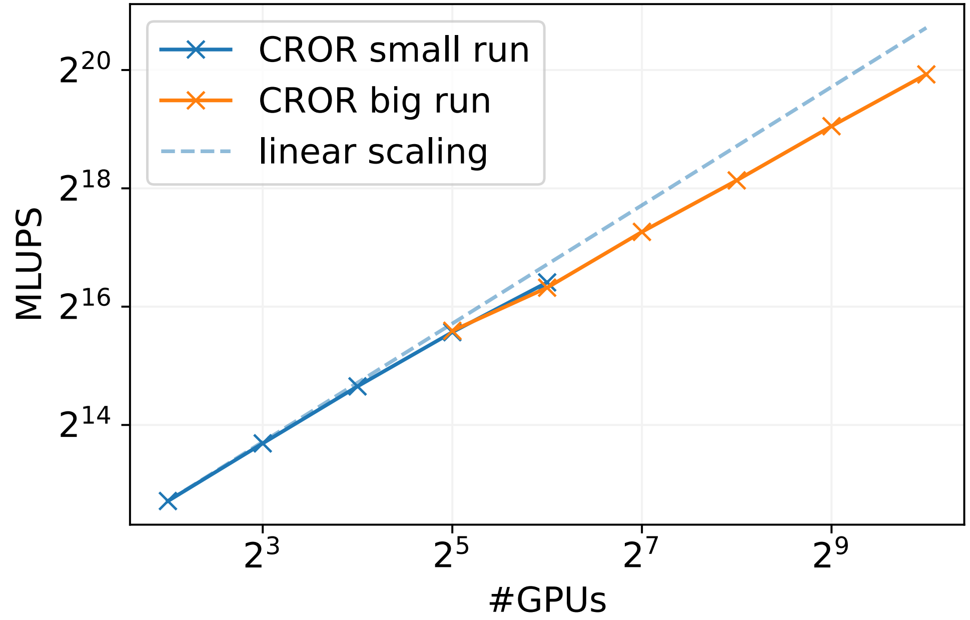



FIG. 13: Strong scaling of the CROR simulation on the CPU partition of JUWELS (NVIDIA A100 GPUs) for the two problem sizes in Table IV.

# Part VI: waLBerla for wind energy

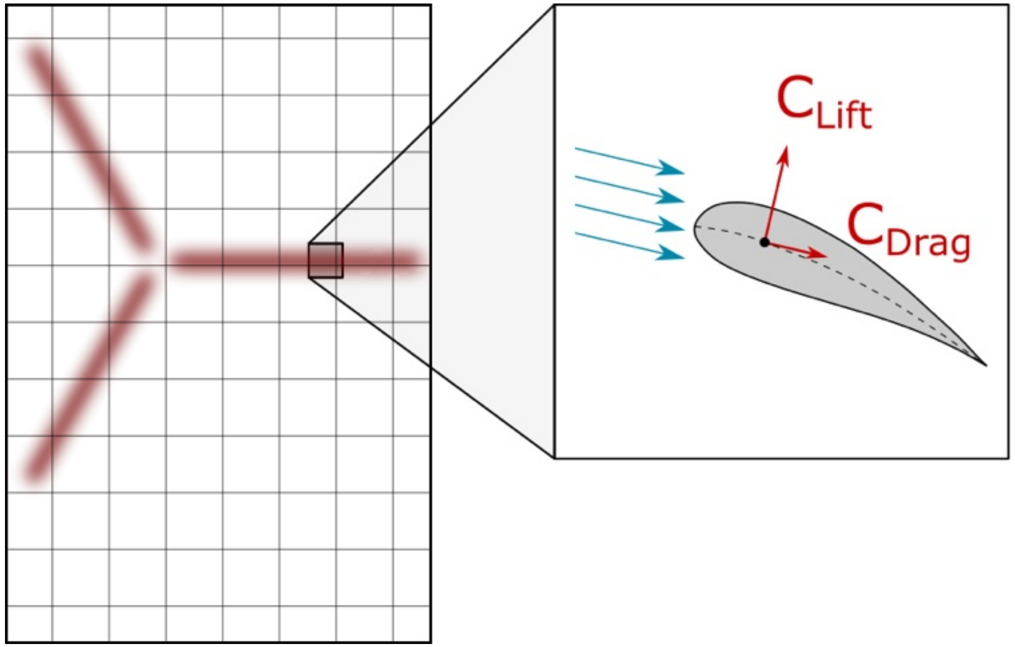
@FAU & IFPEN: PhD student Helen Schottenhamml
@CERFACS: PhD student Markus Holzer
@IFPEN: Ani Anciaux Sedrakian  and Frédéric Blondel

# The actuator line model

- Turbines are not geometrically resolved but represented by the forces they apply to the flow

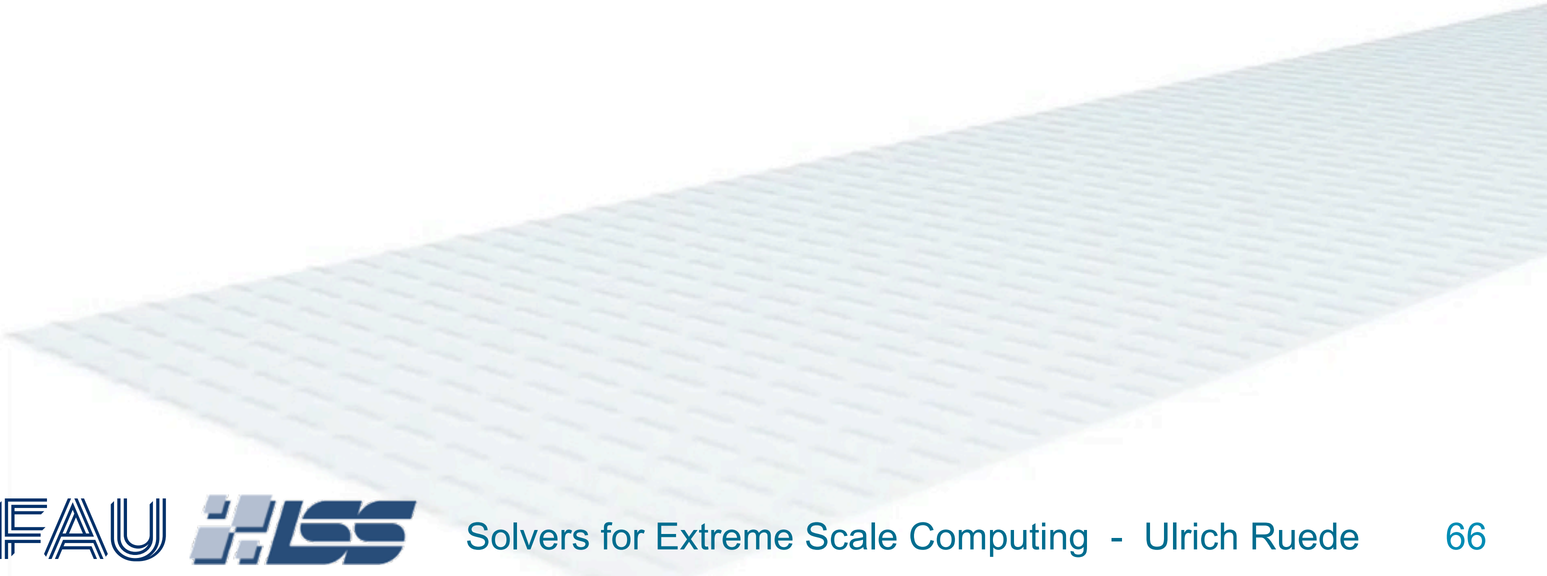
$$F = \frac{1}{2}\rho u_{rel}^{2} w l \left( C_L e_L + C_D e_D \right)$$

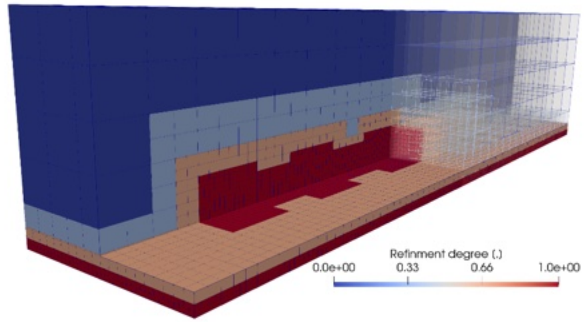
- Lift and drag coefficients are interpolated from airfoil data
- Projection onto the grid via, e.g., Gaussian kernel



# Wind turbine simulations - mesh refinement

- Static refinement
  - Automatic refinement around wind turbines
  - Optionally: refinement at boundaries, user-defined boxes
  - Gradient-based vs. Vorticity-based
- Dynamic refinement
  - Refinement criteria based on flow field
  - AMR-GPU version available, continuing development

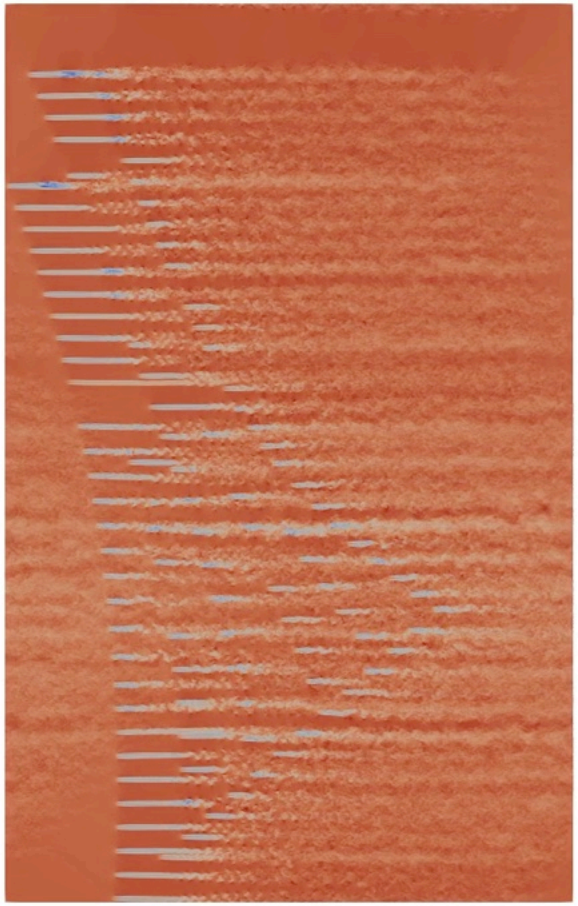


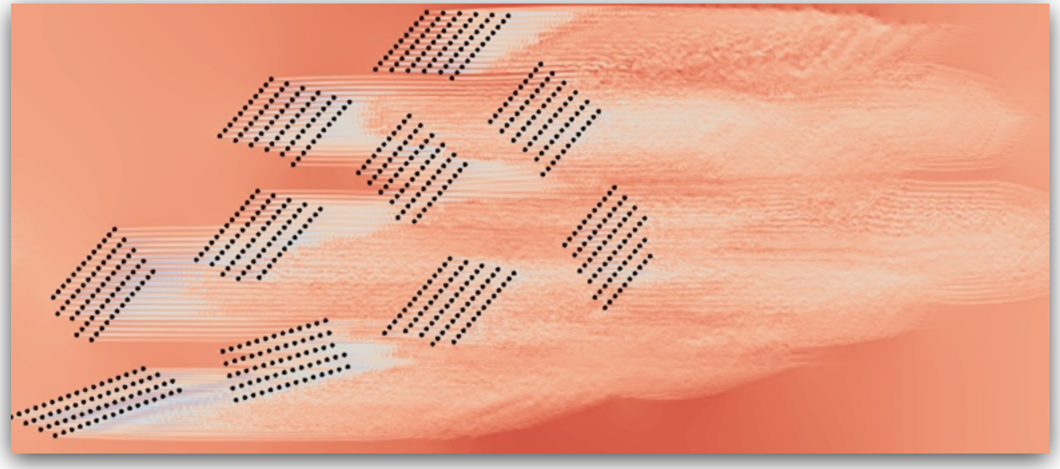


# Towards the simulation of wind farms

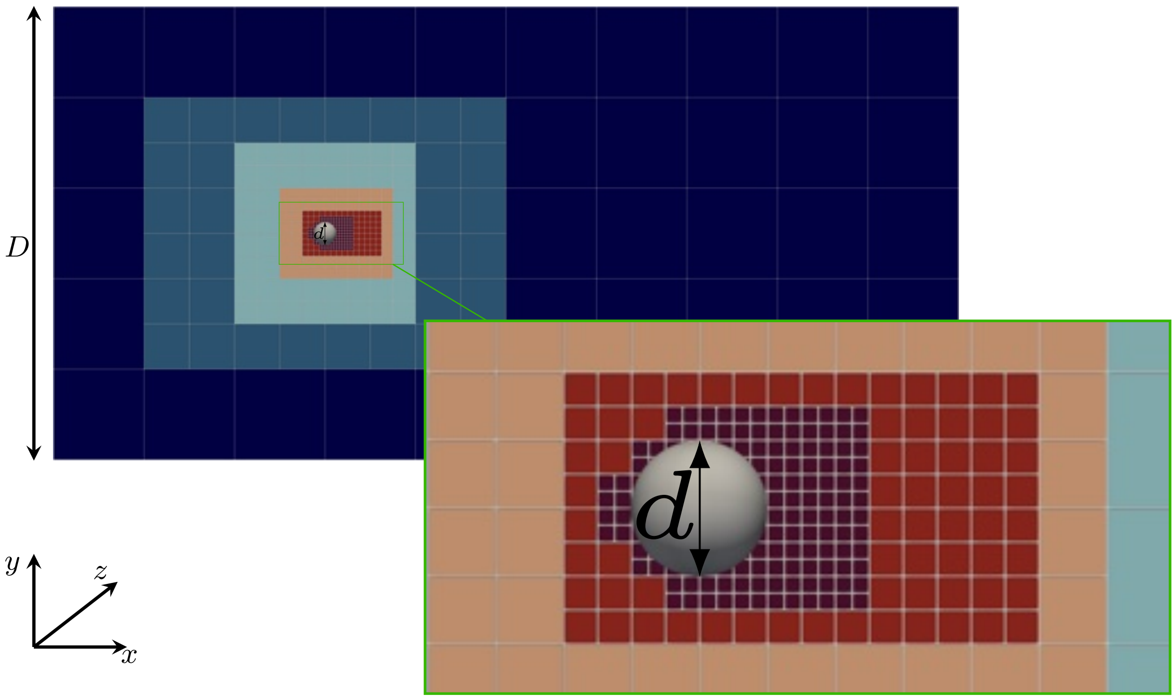
- Anhalt wind farm (off Denmark)
- Actuator disc model
- domain: 2496 x 3920 x 40
- 20 cells / diameter
- 3840 Prozesse = 30 nodes à 128 cores on Topaze Cluster (https://www-ccrt.cea.fr/fr/moyen_de_calcul/index.htm)

- 128 008 timesteps à 0.04s
- 3662 MLUPS
- 13679s run time
- 9.4 time steps / second
- each cell has a size of $(6.3\text{m})^3$
- cumulant LBM with D3Q27 stencil
- periodic boundary conditions



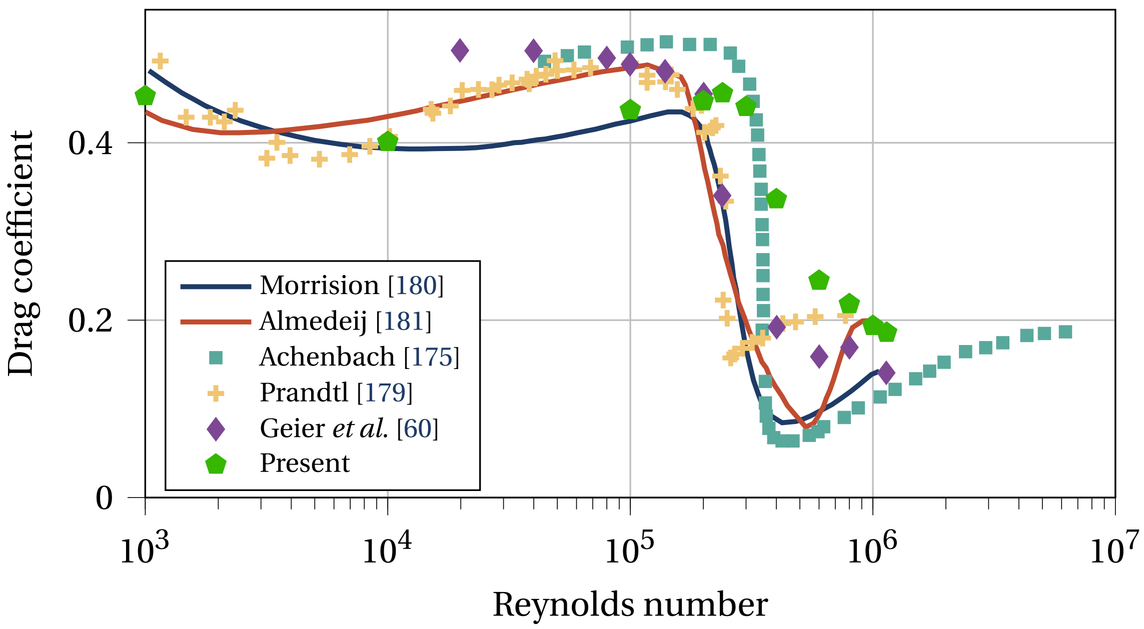


## LBM for High-Re Flows: Drag crisis for flow past a sphere

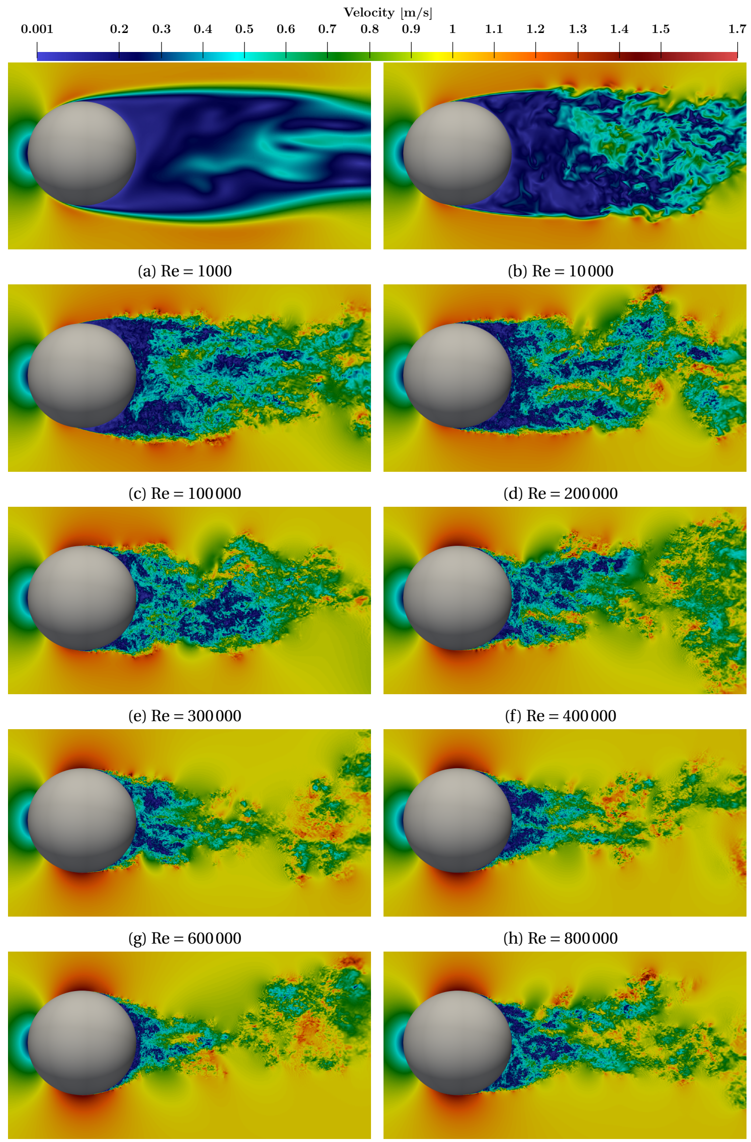



- Spherical obstacle in a simulated wind tunnel
- 6 layers of refined meshes
- Reynolds numbers beyond $10^6$
- Advanced Cumulant LBM with limiter (M. Geier)

# LBM at high Re

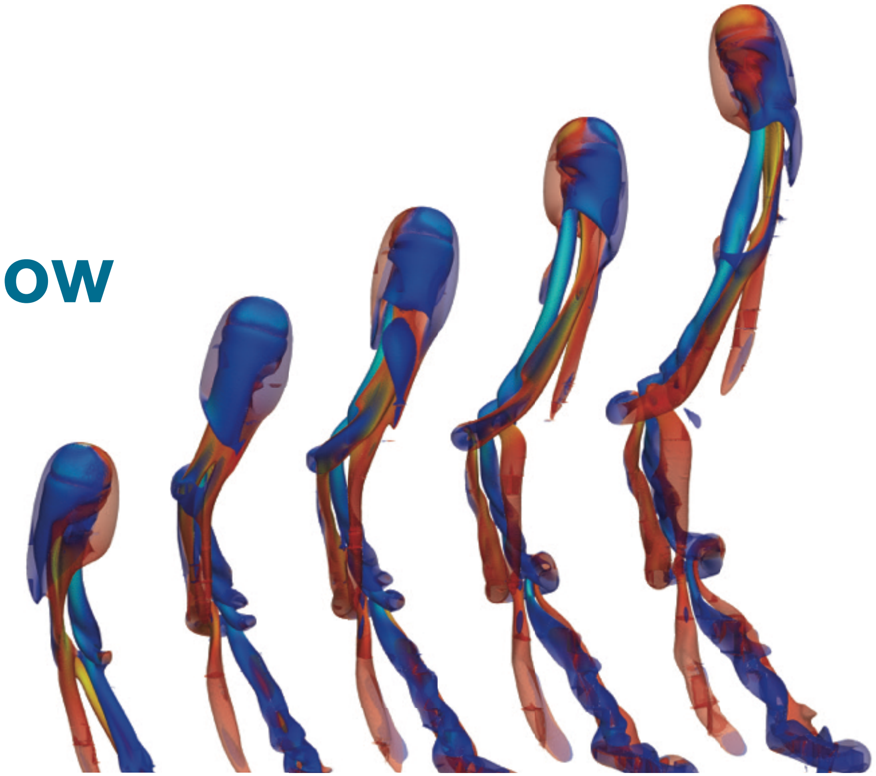



- $9.76 \cdot 10^8$ simulation cells
- 60000 coarse timesteps
- 500 000 fine time steps
- <10 hours using 32 AMD MI250X GPGPUs




(a) Re = 1000
(b) Re = 10000
(c) Re = 100000
(d) Re = 200000
(e) Re = 300000
(f) Re = 400000
(g) Re = 600000
(h) Re = 800000
(i) Re = 1000000
(j) Re = 1140000

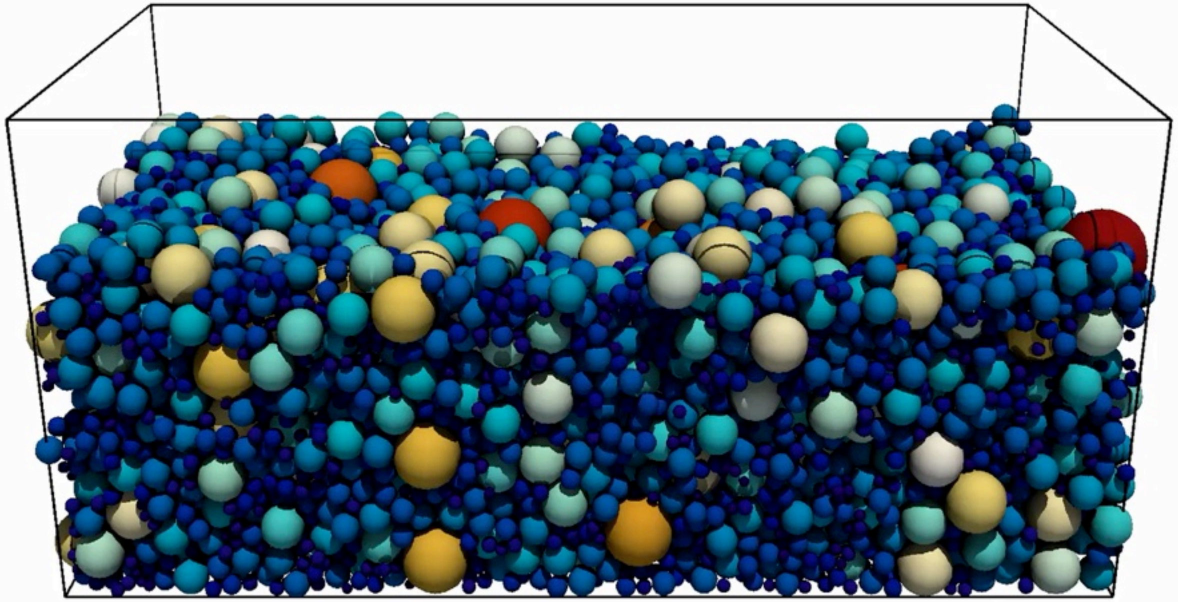
# Part VII:
# Fully Resolved Particle Laden Flow with waLBerla



Vortices behind a rising sphere at Ga=500, density ratio 0.05

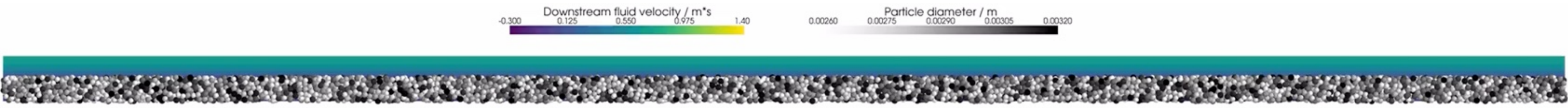
Werner, L., Rettinger, C., UR (2021). Coupling **fully resolved** light particles with the Lattice Boltzmann method on **adaptively refined grids.** *Numerical Methods in Fluid Mechanics, vol. 93, pp. 3280-3303*

# waLBerla - multi physics simulation
# Geometrically resolved particles/LBM hydrodynamics
# sediment transport

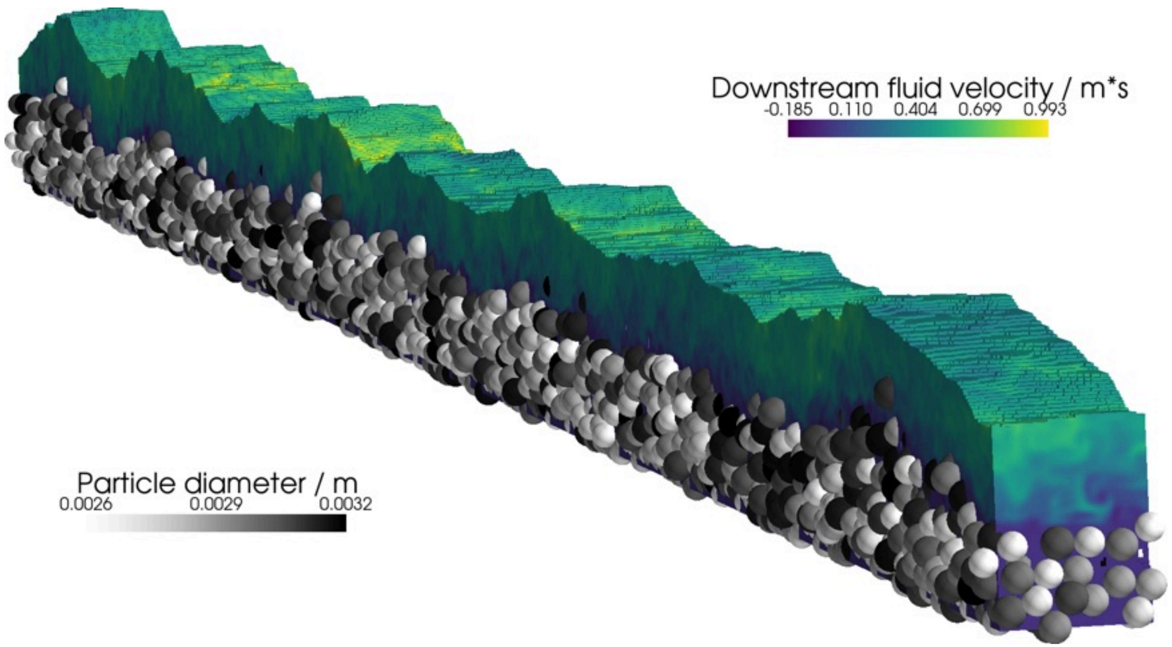


Rettinger, C., Eibl, S., Rüde, U., & Vowinckel, B. (2022). Rheology of mobile sediment beds in laminar shear flow: effects of creep and polydispersity. *Journal of Fluid Mechanics*, *932*.

- Flow domain:
  - 1024 x 512 x 480 = 2.5e8 cells
  - D3Q19 TRT
- 14500 particles
  - diam = 10-100 LBM cells
  - log-normal distribution
  - momentum exchange
  - lubrication forces
- 7e6 LBM time steps
  - each 10 DEM substeps
- Supermuc-NG
  - 160 Nodes = 7680 processes
  - 48h run time

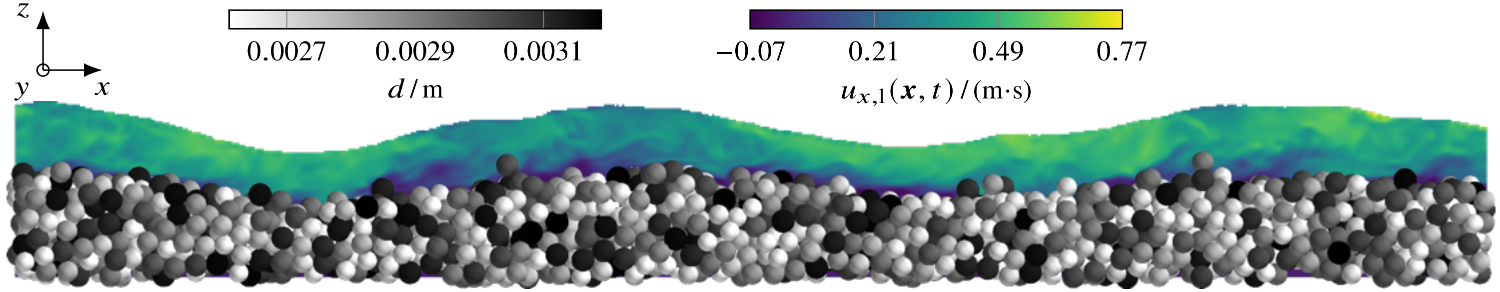


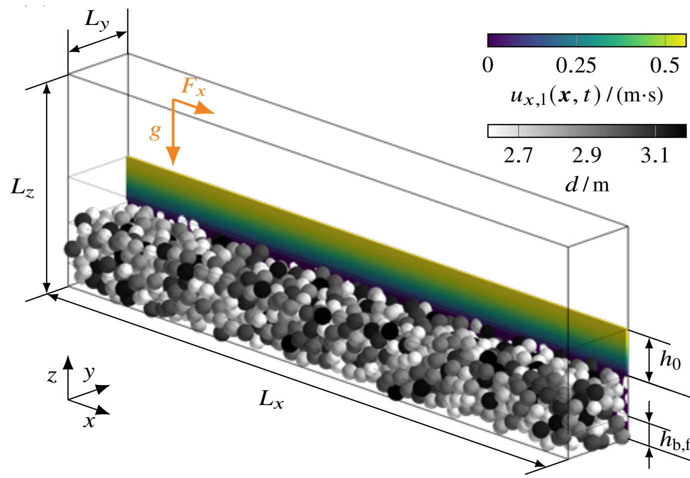

# waLBerla - simulation of anti-dunes

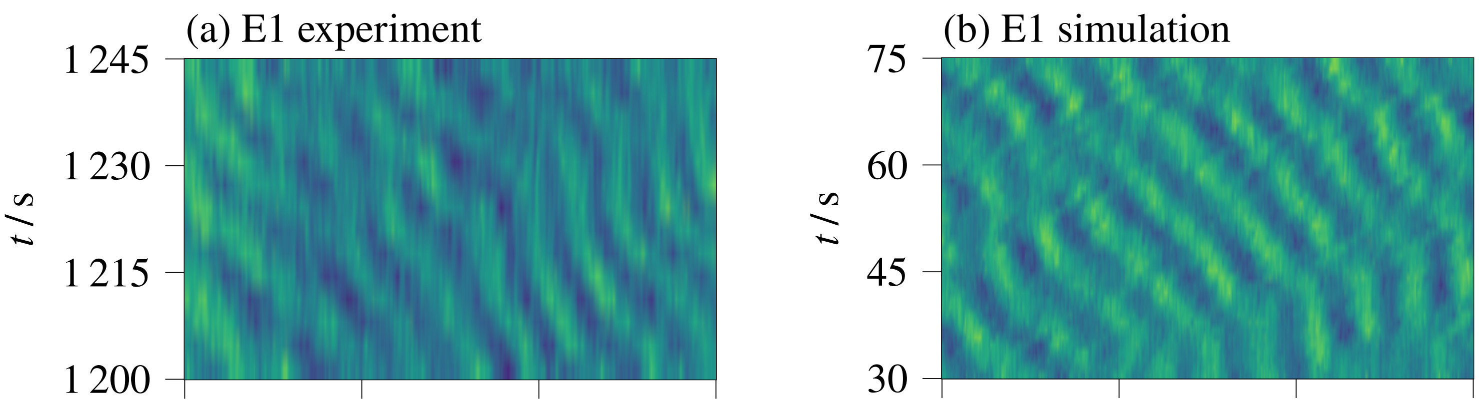


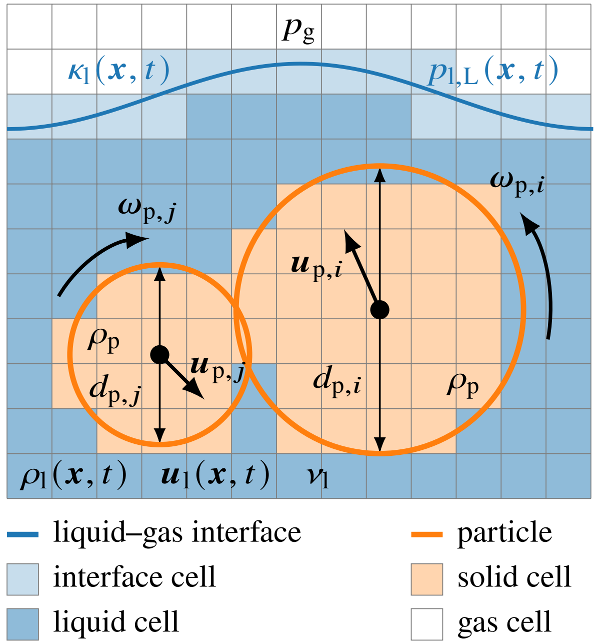

- Modeling particles transported below below a free surface flow
- The transport of particles downstream leads to dune formations that are effectively traveling upstream
- First simulations that model this effect from first principles based on an interaction between
  - Free surface flow
  - Wave formation
  - Transport of fully resolved particles

# waLBerla - simulation of anti-dunes



# Part VII: Conclusions

# Welcome to the **Era of Prophecy Machines**

- Algorithms can make science **predictive**
  for example in
  - climate simulation, spread of diseases
  - all fields of engineering
  - geophysical research
  - …
- An ancient dream of humans becomes reality …
  (especially of politicians)
- Universal:
  - technology
  - economy
  - society
  - politics
- **Predictive capability** can be the basis for far-reaching decisions
- Impact is yet little understood

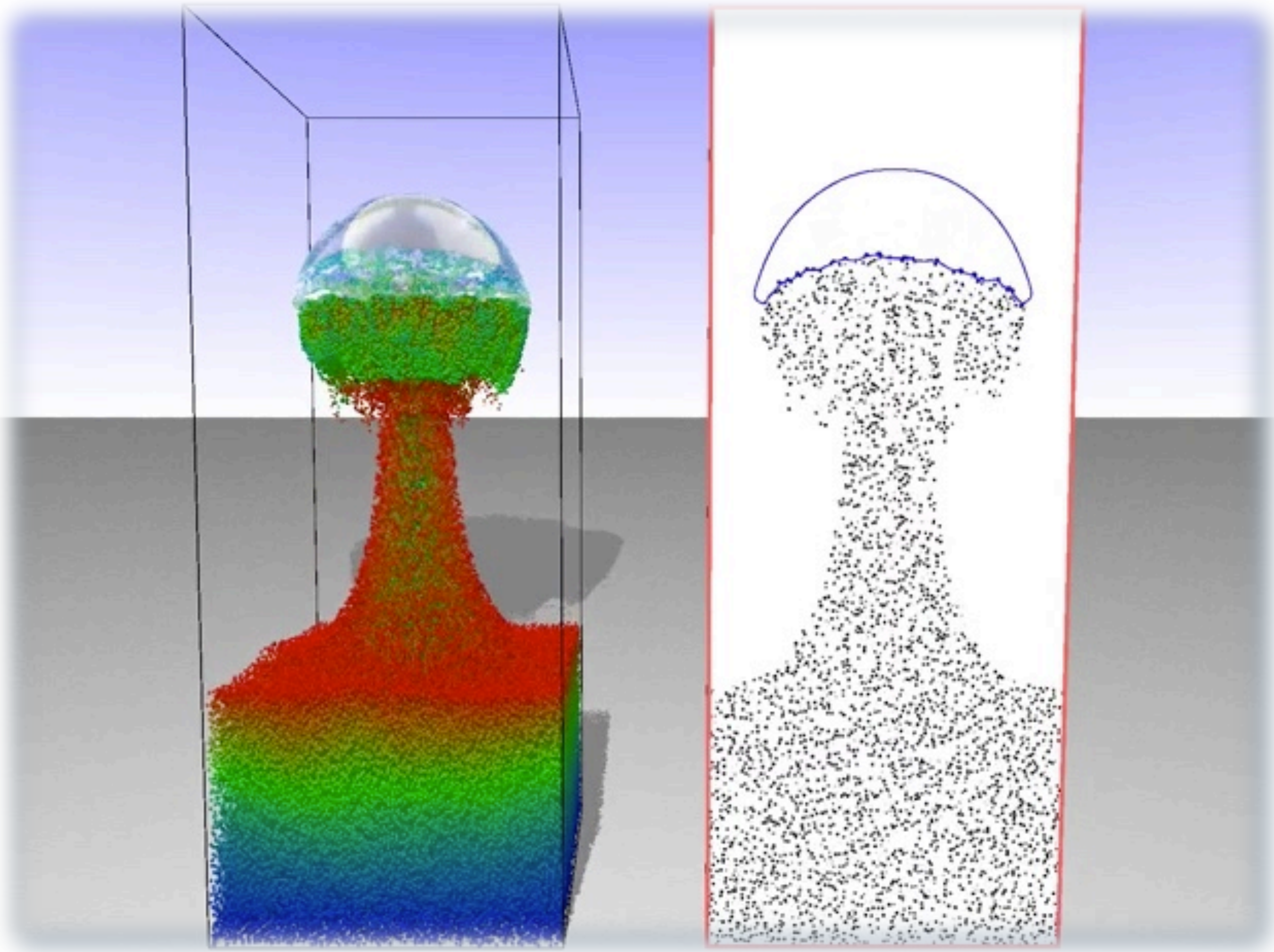
**Clarke's Third Law**:
Any sufficiently advanced technology is indistinguishable from magic.

# Can we predict


?

What does prediction mean?

# Thank you for your attention!



Bogner, S., & UR. (2013). Simulation of **floating bodies** with the lattice Boltzmann method. *Computers & Mathematics with Applications*, *65*(6), 901-913.

Anderl, D., Bogner, S., Rauh, C., UR, & Delgado, A. (2014). Free surface lattice Boltzmann with enhanced **bubble model**. *Computers & Mathematics with Applications*, *67*(2), 331-339.

Bogner, S., Harting, J., & Rüde, U. (2017). Direct simulation of **liquid–gas–solid flow** with a free surface lattice Boltzmann method. *International Journal of Computational Fluid Dynamics*, *31*(10), 463-475.