dealii-X: Overall project plans

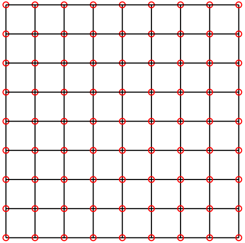
July 3, 2025

Technical aspects at core for RUB and LRZ teams

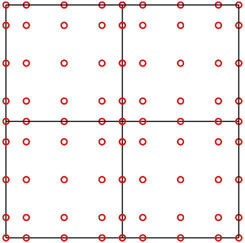
- Develop efficient tools for a range of partial differential equations
  - Basis for several applications in applied groups
- Core algorithm for RUB team: **Matrix-free implementations** for high-order finite-element & discontinuous Galerkin methods
  - Optimizations for GPUs and suitable abstractions
- **Node-level performance tuning**
  - Integration via hardware abstraction layers for matrix-free evaluation
- Work towards integrating selected **deal.II kernels** into benchmarking suites on European clusters
  - Ideally, expand usability such that also other European efforts can use specialized algorithms provided by us, make us generally applicable facility
- **Robust iterative solvers** for large-scale problems with implicit time stepping
  - Multigrid and mixed precision
  - Combination with matrix-based solvers of PSCToolkit (UNITOV) and MUMPS (Toulouse)

Discretizations with FEM

- **Methods of higher order of accuracy**: Finite element methods and discontinuous Galerkin
  - Higher order = better for fine scales due to better dispersion and dissipation behavior
  - Higher order methods can solve problems with fewer global unknowns
  - When implemented with matrix-free algorithms, the additional computations for high order need to be done on cached data $\rightarrow$ fits well with modern hardware
- **Optimized implementations for mixed meshes** (hex, tet, pyramids, wedges)
  - Hexahedra offer best performance, but other element types better for complex geometries, important for several of our partners



pol. degree 1, fine mesh



pol. degree 4, coarse mesh

Central algorithm: matrix-free operator evaluation

RUHR UNIVERSITÄT BOCHUM RUB

Matrix-vector product

matrix-based:

$$\begin{cases} A = \sum_{e=1}^{N_{\text{el}}} P_e^{\mathrm{T}} A_e P_e & \text{(assembly)} \\ v = Au & \text{(matrix-vector product within iterative solver)} \end{cases}$$

$\downarrow$

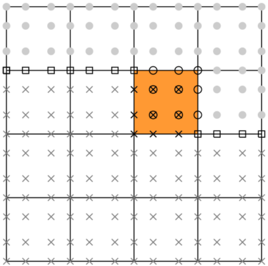
matrix-free:

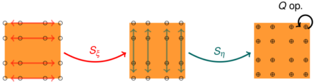
$$v = \sum_{e=1}^{N_{\text{el}}} P_e^{\mathrm{T}} A_e (P_e u)$$

implication: FEM assembly within iterative solvers

Matrix-free evaluation of FEM operator

- $v = 0$
- loop over elements $e = 1, \ldots, N_{\text{el}}$
  - (i) Extract local vector values: $u_e = P_e u$
  - (ii) Apply operation locally by integration: $v_e = A_e u_e$, **do not form** A_e, compute its action by FEM integrals
  - (iii) Sum results from (ii) into the global solution vector: $v = v + P_e^{\mathrm{T}} v_e$





$$A_e u_e = S^{\mathrm{T}} Q_e S u_e$$

M. Kronbichler, K. Kormann, A generic interface for parallel finite element operator application. *Comput. Fluids* 63:135–147, 2012

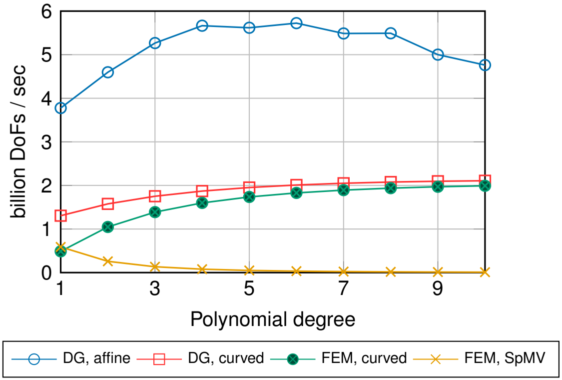
M. Kronbichler, K. Kormann, Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM TOMS* 45(3), 29, 2019

Included in `deal.II` finite element library, `www.dealii.org`

Matrix-free vs. matrix-based methods

- Performance of matrix-vector product essential for iterative solvers
- Sparse matrices unsuitable for higher orders $p \geq 2$ on modern hardware due to **memory-bandwidth** limit
- Matrix-free algorithm successful in trading computations for less memory transfer
  - Software: Specify operation at quadrature points
  - Combine with reference cell interpolation matrices
  - Indirect access into vector entries for continuous FEM

Throughput of matrix-vector product (unknowns processed per second) of 3D Laplacian



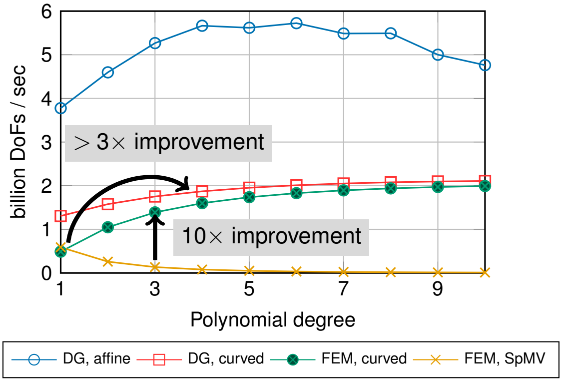
System: 1 node of 2×24 cores of SuperMUC-NG (Intel Xeon Skylake)
Memory bw: 205 GB/s, arithmetic peak 3.5 TFlop/s

Kronbichler, Kormann: A generic interface for parallel cell-based finite element operator application. *Comput Fluids* 63:135–147, 2012
Kronbichler, Wall: A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers. *SISC* 40(5):A3423–48, 2018
Kronbichler, Kormann: Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM TOMS* 45(3):29/1–40, 2019

Matrix-free vs. matrix-based methods

- Performance of matrix-vector product essential for iterative solvers
- Sparse matrices unsuitable for higher orders $p \geq 2$ on modern hardware due to **memory-bandwidth** limit
- Matrix-free algorithm successful in trading computations for less memory transfer
  - Software: Specify operation at quadrature points
  - Combine with reference cell interpolation matrices
  - Indirect access into vector entries for continuous FEM

Throughput of matrix-vector product (unknowns processed per second) of 3D Laplacian

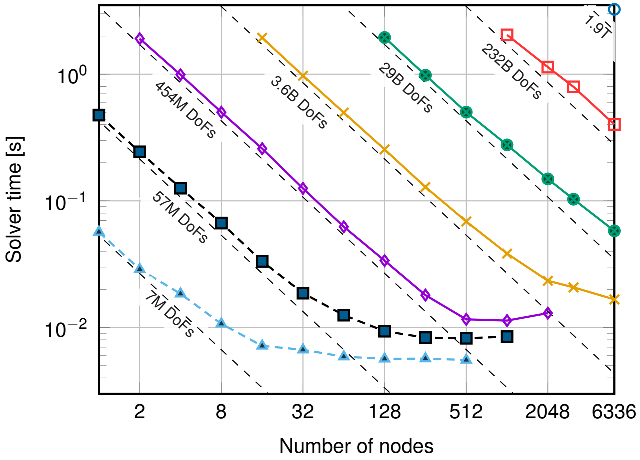


System: 1 node of 2×24 cores of SuperMUC-NG (Intel Xeon Skylake)
Memory bw: 205 GB/s, arithmetic peak 3.5 TFlop/s

Kronbichler, Kormann: A generic interface for parallel cell-based finite element operator application. *Comput Fluids* 63:135–147, 2012
Kronbichler, Wall: A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers. *SISC* 40(5):A3423–48, 2018
Kronbichler, Kormann: Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM TOMS* 45(3):29/1–40, 2019

Efficient solver: multigrid

- Work on hierarchy of coarser problems, local smoother operations on each level
  - All but the finest 2 or 3 levels are communication-limited
  - With Ivan Prusak, investigate new algorithms to reduce communication frequency
- Discontinuous elements, $p = 5$, affine mesh
- 2 CG iterations (tolerance 10^{-3})
- Serial performance very high: **2.5 million DoFs / core / sec**
- Intel Xeon Platinum 8174, 48 cores per node, up to 6336 nodes (full SuperMUC-NG)



Arithmetic performance 1.9 trillion DoFs:
5.9 PFlop/s (5.7 PFlop/s in SP, 0.27 PFlop/s in DP)
180 GB/s per node (STREAM: 205 GB/s)