

PARALLEL HP-ADAPTIVE METHODS FOR BUOYANCY-DRIVEN FLOWS

November 27, 2018 | Marc Fehling | Institute for Advanced Simulation (IAS)

Table of contents

Introduction

- Adaptive methods

- Parallelization

- Parallel generic adaptive methods in FEM

Enumeration of degrees of freedom

Data transfer across subdomains

Load balancing

Summary & Outlook

Efficient use of computational resources

Topic of interest:

- Current scope of computational resources allows solving problems of enormous size
- Combination with efficient algorithms offers massive potential to increase the accuracy of solution on large number of unknowns

Approaches presented in this talk:

- Dynamic resolution with **adaptive methods**
 - Focus computational resources on areas of interest
- Multi-core architecture suggests **parallelization**
 - Use multiple processors at once to solve problems

Adaptive methods

- Focus computational resources on areas of interest
- Align simulation resolution with complexity of current solution
- Finite Element Method (FEM) provides two different possibilities:
 - h-refinement:** dynamic cell sizes good for irregular solutions
 - p-refinement:** dynamic function spaces good for smooth solutions
- Combination of both possible

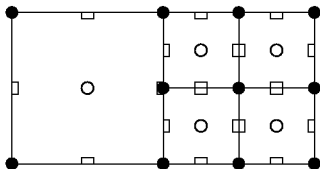


Figure: **h**-adaptive methods

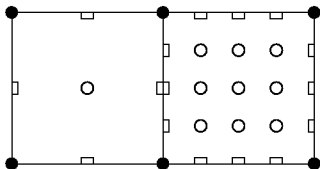


Figure: **p**-adaptive methods

Adaptive methods in fire simulation

- Current open-source software features static, structured meshes
 - All cells have same topology
 - Resolution is prescribed from the beginning
- Unstructured, adaptive meshes feature more flexibility
 - Cell sizes differ
 - Dynamic resolution
- Beneficial for multi-scale simulation domains
 - Narrow openings
 - HVAC systems
 - Sensors
 - Dynamic fire propagation
 - ...

Adaptive methods: Example

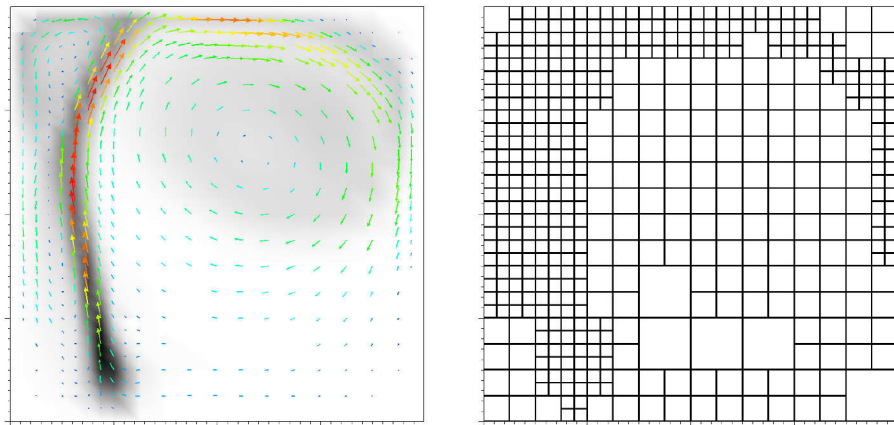


Figure: Exemplary simulation of a buoyancy driven flow with a Gaussian heat source.

Adaptive methods: Benefits

- Comparison of runtime and accuracy between uniform and adaptively refined meshes with the moving vortex example

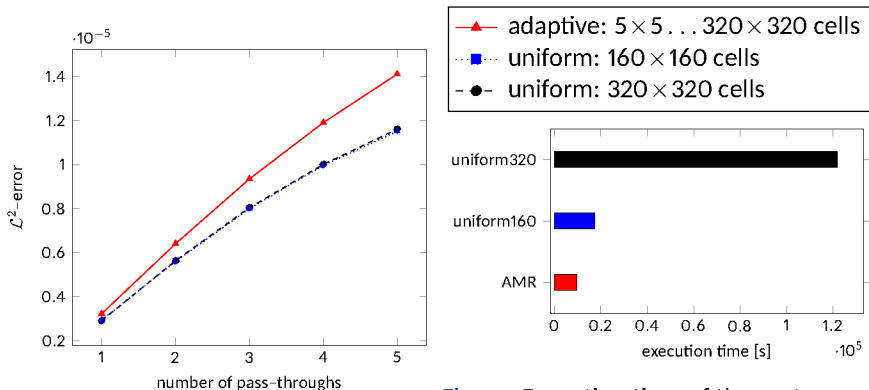


Figure: Global $\mathcal{L}^2$ -errors at each periodic simulation, run in serial on a common pass-through of the vortex.

Figure: Execution time of the vortex simulation, run in serial on a common desktop computer.

Parallelization

- Current computer architectures provide multi-core processors
 - Supercomputers arrange those on distributed nodes
- Using all resources efficiently requires parallelization
 - Distribution of workload and memory demand
- Our approach: Distribution of domain on several processes
 - Each subdomain needs relevant part of the global solution
 - Requires a layer of so called ghost cells
 - Involves communication between processors

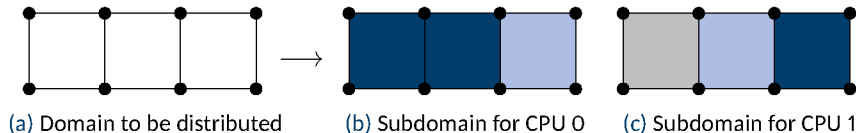


Figure: Illustration of **locally owned**, **ghost**, and **artificial** cells

Parallel generic adaptive methods in FEM

- Combine both approaches to get parallel hp-adaptive methods
- Develop generic algorithm, applicable for any FEM software
- The non-trivial parts are:
 - 1 Enumeration of degrees of freedom (DoFs)
 - 2 Data transfer across subdomains
 - 3 Load balancing
- Reference implementation in deal.II library [1]
 - See pull request #3511 for development log



Enumeration of degrees of freedom

- Numbering of DoFs necessary to build linear equation
 - Parallelization and p-adaptive methods require different algorithms
 - Combination of both **not trivial**
-
- Why not number DoFs locally and handle relations with constraints?
 - Number of DoFs would vary with number of processors
 - Larger matrices & vectors, thus higher memory demand

Distribution algorithm

- Need for algorithm independent of number of processors
 - Results independent of number of processors
 - Proof that solvers work
 - Simplifies debugging

- Developed 6-phase algorithm
 - 1 Local enumeration of DoFs
 - 2 Invalidate DoFs on ghost interfaces to processors with lower rank
 - 3 Unification of DoFs on local domain **and** ghost interfaces
 - Ownership of DoFs clarified
 - 4 Global re-enumeration of DoFs
 - Local DoF indices set
 - 5 Exchange of locally owned DoFs
 - 6 Merge DoFs on ghost interfaces
 - Global DoF indices set

Data transfer across subdomains

- On distributed triangulations, each subdomain needs access to relevant fraction of global quantities
- Communication between processors required to transfer:
 - Active finite element indices
 - Solution
 - ...
- With p-adaptive methods, per cell data sizes may differ
- Algorithm should be generic, i.e. independent of chosen adaptive method, number of callback functions, ...
 - 1 Creation of memory buffers should be generic as well
 - 2 Use non-blocking MPI communication to transfer data of **fixed** and **variable** size

Structure of data chunk

■ On each cell:

- 1 Pack data for each registered callback individually
- 2 Store memory sizes of each callback's data pack
- 3 Store memory size of cell's complete data pack

■ After each cell is processed:

- 4 Move each cell's packed data into contiguous buffer



Figure: Division of contiguous memory chunk.

Load balancing

- p-adaptivity yields differing workload between cells
- **Weighted repartitioning** achieves balanced load per processor
- Factors that determine workload:
 - Cell construction
 - Matrix & right-hand-side assembly
 - Type of solver
- Correlation to **number of DoFs, quadrature formula, ...**
- How to find a suitable estimate for a cell's workload?
 - Open question

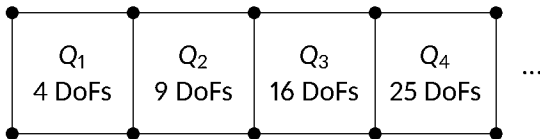


Figure: Different finite elements and their number of degrees of freedom in 2D.

Summary & Outlook

- New algorithm for massively parallel hp-adaptive methods, generally applicable for any FEM software
- Reference implementation in deal.II involves:
 - Enumeration of degrees of freedom, independent of number of subdomains
 - Consignment of contiguous memory chunks for data transfer
 - Weighted repartitioning for load balancing
- Future steps:
 - Heuristic analysis on reasonable cell weights
 - Simplify interface for hp-methods in deal.II
 - Provide tutorials in deal.II as a manual for a broader audience

Thank you for your kind attention!

Bibliography I



G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells.

The deal.II Library, Version 9.0.

Journal of Numerical Mathematics, 2018, accepted.



C. Burstedde.

Parallel tree algorithms for AMR and non-standard data access.

CoRR, abs/1803.08432, 2018.