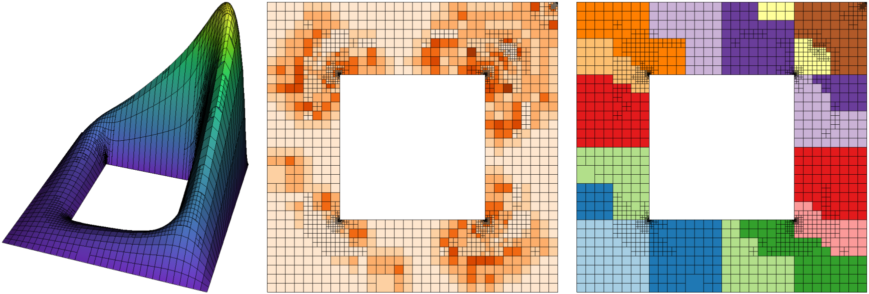


MASSIVELY PARALLEL HP-ADAPTIVE FINITE ELEMENT METHODS

November 29, 2019 | Marc Fehling | m.fehling@fz-juelich.de | Institute for Advanced Simulation (IAS)

JÜLICH Forschungszentrum

Table of contents

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-adaptation:	dynamic cell sizes	good for irregular solutions
p-adaptation:	dynamic function spaces	good for smooth solutions

- Combination of both possible

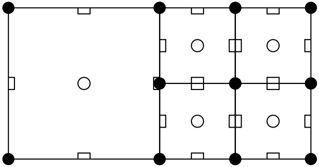


Figure: **h**–adaptive methods

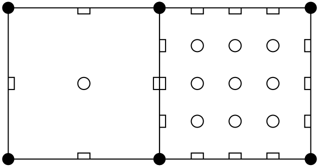
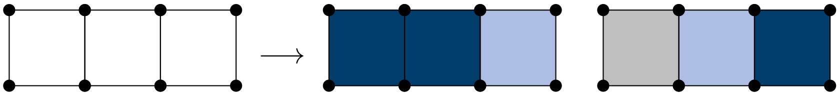


Figure: **p**–adaptive methods

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



(a) Domain to be distributed (b) Subdomain for CPU 0 (c) Subdomain for CPU 1

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 issue #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
  - See parallel [2] and hp [3] papers for details

Combination of both algorithms **not trivial**

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

Enumeration algorithm on paper

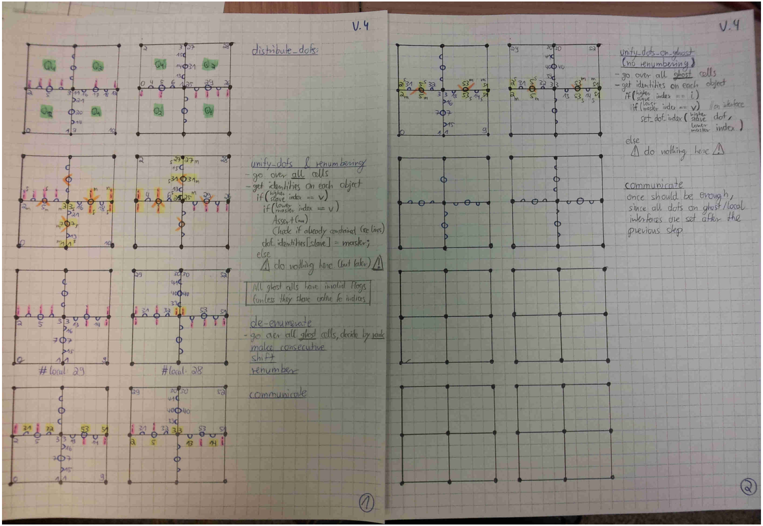


Figure: Final version of the enumeration algorithm

Data transfer across subdomains

- On distributed triangulations, each subdomain needs access to relevant fraction of global quantities
- Changes on cell ownership requires transfer of these quantities
- With p-adaptive methods, per cell data sizes may differ

Communication between involved processors required

- Creation of **memory buffers** for **fixed and variable** size data

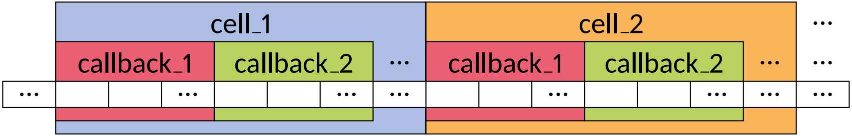


Figure: Division of contiguous memory chunk

Data transfer

- Treat fixed and variable size data separately
  - Each transfer algorithm optimized for their specific task
  - Potentially slower variable size transfer will only be used when necessary
  - Compression possible with variable size transfer

- We have contiguous memory chunks for data transfer during repartitioning, refinement/coarsening, serialization
  - Program may be resumed with a different number of processors

- Data consignment **independent** of transfer algorithms used for repartitioning, refinement/coarsening, serialization
  - Use non-blocking `MPI` communication for all operations
  - `deal.II` utilizes interface to `p4est` [4]

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

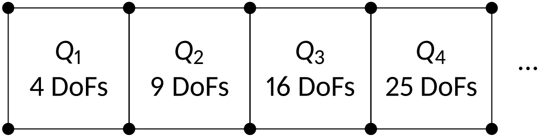


Figure: Different finite elements and their number of degrees of freedom in 2D

Dynamic hp-adaptive methods

- At this stage, parallel hp-FEM on static meshes possible
- **Dynamic strategies** required for successive adaptation
- Variety of hp-adaptive strategies reviewed by Mitchell [5]
  1. Refinement history
  2. Smoothness estimation
- Selection implemented in `deal.II` library [1]
  - See issue #7515 for development log

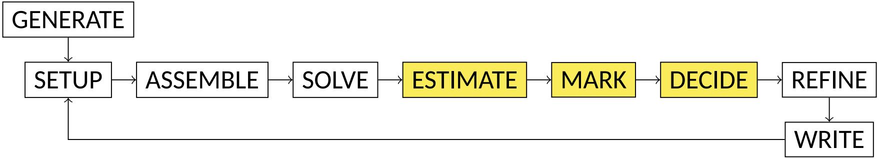


Figure: Enhanced SOLVE-ESTIMATE-MARK-REFINE cycle

Mark cells for adaptation

- Determine sensitive areas of solution where resolution shall be adapted
- For this, assess *a posteriori* error estimates as indicators for adaptation

1. **Estimate** errors on all cells
   - Estimator by Kelly et al. [6] for Laplace equation: $-\nabla \cdot (a \nabla u) = f$
   - Proved as reasonable indicator for other scenarios as well
   - Implemented in `deal.II` as `KellyErrorEstimator`

$$\|\nabla(u - u_{\mathrm{h}})\|^2_{H^1(\Omega)} \leq C \sum_K \eta_K^2, \quad \eta_K^2 = \sum_{F \in \partial K} c_F \int_{\partial K_F} \left[a \frac{\partial u_h}{\partial n} \right]^2 do, \quad c_F = \frac{h_F}{2 p_F}$$

2. **Mark** cells for adaptation
   - Most prominent strategies

     fixed number: controls growth of mesh size

     fixed fraction: controls reduction of error estimates

A priori error prediction

- Error behavior for hp-FEM is well understood [5]
- Algebraic convergence rate with h-adaptation

$$\|\nabla (u - u_{\mathrm{hp}})\|_{H^1(\Omega)} \leq C \, \frac{h^{\mu}}{p^{m-1}} \, \|u\|_{H^m(\Omega)}, \; \mu = \min(p, m-1), \; C \text{ dependent on } m$$

- Exponential convergence rate with p- or hp-adaptation
  - Requires solution to be sufficiently regular

$$\|\nabla (u - u_{\mathrm{hp}})\|_{H^1(\Omega)} \leq C \, \exp\left(-b \, N_{\mathrm{dofs}}^{1/3}\right), \; C, \; b > 0 \text{ independent of } N_{\mathrm{dofs}}$$

Refinement history

Predict and **verify** error of solution during hp-adaptation process

Refinement history

- Verify prediction's accuracy
- Decide on either h- or p-adaptation on marked cells

  Keep **h** fine: $\eta_K > \eta_{K,\text{pred}}$

  Keep **p** large: $\eta_K \leq \eta_{K,\text{pred}}$

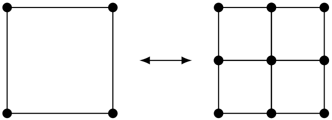


Figure: h-adaptation

Figure: Error prediction algorithm based on Melenk and Wohlmuth [7]

Adaptation type	Prediction formula	
no adaptation	$\eta_{K,\text{pred}} = \eta_K \, \gamma_\text{n}$	$\gamma_\text{n} \in (0, \infty)$
p-adaptation	$\eta_{K,\text{pred}} = \eta_K \, \gamma_\text{p}^{(p_{K,\text{future}} - p_K)}$	$\gamma_\text{p} \in (0, 1)$
h-refinement	$\eta_{K_c,\text{pred}} = \eta_K \, \gamma_\text{h} \, 0.5^{p_K} \, 0.5^{\text{dim}}$	$\gamma_\text{h} \in (0, \infty)$
h-coarsening	$\eta_{K,\text{pred}} = \sum_{K_c} \eta_{K_c} / (\gamma_\text{h} \, 0.5^{p_{K_c}})$	$\forall K_c$ children of K

Smoothness estimation

- Decay of series expansion coefficients for indicating smoothness

- Legendre series expansion as presented by Mavriplis [8]
  - Evaluate exponential decay of Legendre coefficients $a_i \sim C\, e^{-\sigma\, i}$
  - Convergence rates $\sigma > 1$ indicate good grid resolution (keep **h** fine)

- Fourier series expansion as presented in `step-27`
  - Mapped solution $\hat{u}(\hat{\mathbf{x}})$ in $\mathcal{H}^{\mu - dim/2}$ when Fourier coefficients $\hat{U}_{\mathbf{k}}$ decay as

$$\left|\hat{U}_{\mathbf{k}}\right| = \left| \int_{\hat{K}} \exp\left(i\, \mathbf{k} \cdot \hat{\mathbf{x}}\right)\, \hat{u}(\hat{\mathbf{x}})\, d\hat{\mathbf{x}} \right| = \mathcal{O}\left(|\mathbf{k}|^{-\mu-\epsilon}\right)$$

  - Use convergence rates μ as smoothness indicators and compare them to relative thresholds (`step-27`) ...

Example: Reentrant corner

- Domain with reentrant corner

$$\Omega = \left\{(r,\varphi) : 0 \le r \wedge 0 \le \varphi \le \frac{\pi}{\alpha}\right\}$$

- Laplace problem has a solution

$$-\nabla^2 u = 0$$
$$\bar{u} = r^{\alpha} \sin(\alpha\varphi)$$

with singularity for $\alpha \in \left(\frac{1}{2}, 1\right)$

- We pick $\alpha = \frac{2}{3}$ and solve on

$$\Omega = [-1,1]^2 \setminus ([0,1] \times [-1,0])$$

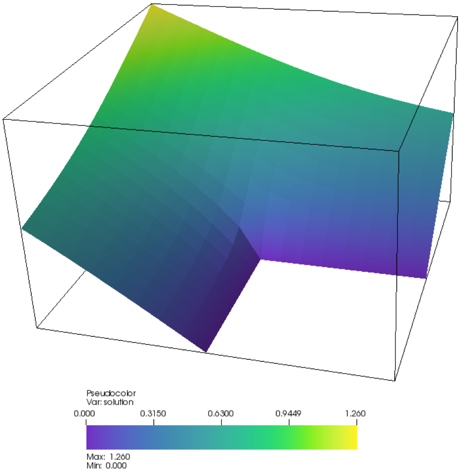
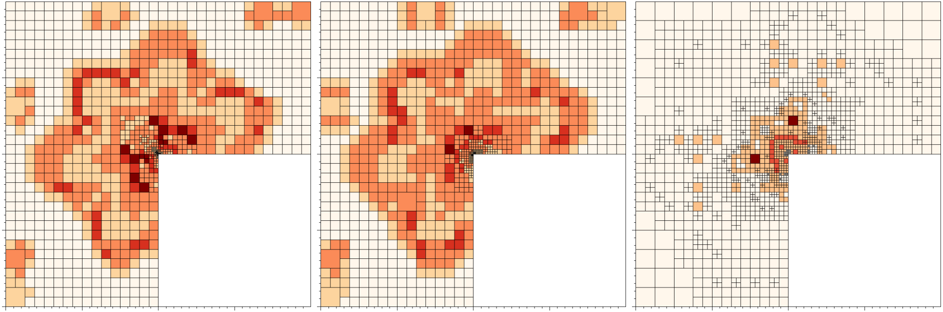


Figure: L-shaped domain

Example: Comparison of decision strategies



(a) Fourier coefficient decay (b) Legendre coefficient decay (c) Refinement history

Figure: Mesh and polynomial degrees of finite elements after 4 consecutive hp-adaptations.

Example: Comparison of refinement types

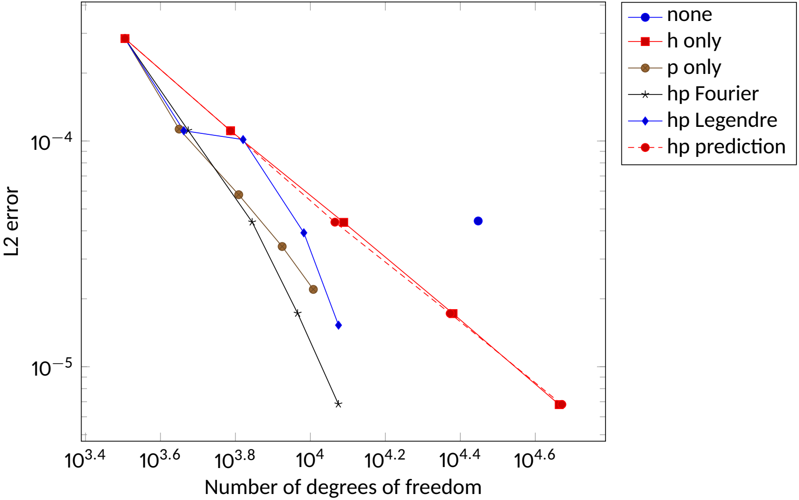


Figure: Error convergence for different strategies (n_glob_refs=4, p_init=2)

Results for `hp Legendre` not satisfactory → investigate!

Example: Scaling

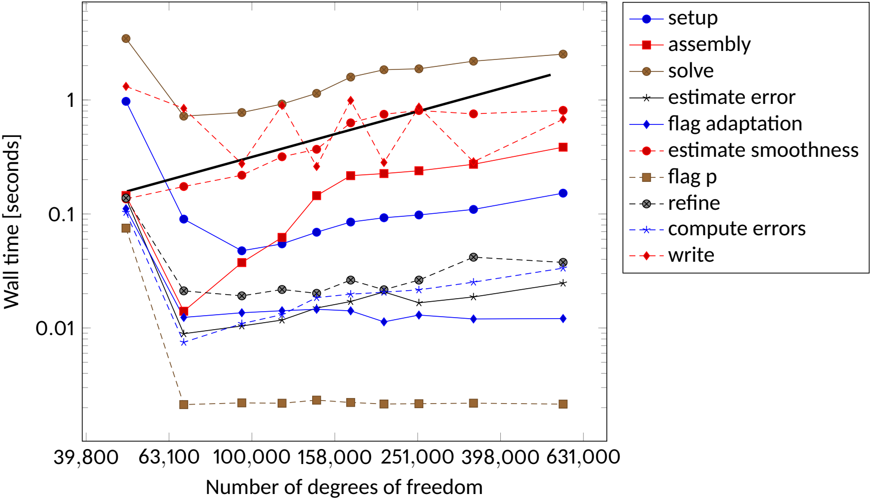


Figure: Weak scaling for Fourier decay strategy (`PETSc`, n_procs=136, p=2,3,4,5,6)

Results for `assembly` not satisfactory $\rightarrow$ investigate!

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
  - Selection of adaptation strategies for hp-FEM

- Future steps:
  - Heuristic analysis on reasonable cell weights
  - Provide tutorials in `deal.II` as a manual for a broader audience

Bibliography I

- D. Arndt, W. Bangerth, T. C. Clevenger, D. Davydov, M. Fehling, D. Garcia-Sanchez, G. Harper, T. Heister, L. Heltai, M. Kronbichler, R. M. Kynch, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells.
  The `deal.II` Library, Version 9.1.
  Journal of Numerical Mathematics, 2019.

- W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler.
  Algorithms and Data Structures for Massively Parallel Generic Adaptive Finite Element Codes.
  ACM Transactions on Mathematical Software, 38(2):14:1–14:28, January 2012.

- W. Bangerth and O. Kayser-Herold.
  Data Structures and Requirements for Hp Finite Element Software.
  ACM Transactions on Mathematical Software, 36(1):4:1–4:31, March 2009.

Bibliography II

C. Burstedde.
Parallel tree algorithms for AMR and non-standard data access.
arXiv e-prints, page arXiv:1803.08432, Mar 2018.

W. F. Mitchell and M. A. McClain.
A Comparison of hp-Adaptive Strategies for Elliptic Partial Differential Equations.
ACM Transactions on Mathematical Software, 41(1):2:1–2:39, October 2014.

D. W. Kelly, J. P. De S. R. Gago, O. C. Zienkiewicz, and I. Babuška.
A posteriori error analysis and adaptive processes in the finite element method: Part I–Error Analysis.
International Journal for Numerical Methods in Engineering, 19(11):1593–1619, Nov 1983.

Bibliography III

J. M. Melenk and B. I. Wohlmuth.
On residual-based a posteriori error estimation in hp-FEM.
Advances in Computational Mathematics, 15(1):311–331, Nov 2001.

C. Mavriplis.
Adaptive mesh strategies for the spectral element method.
Computer Methods in Applied Mechanics and Engineering, 116(1):77–86, 1994.

M. Ainsworth and B. Senior.
An adaptive refinement strategy for hp-finite element computations.
Applied Numerical Mathematics, 26(1):165–178, 1998.

JÜLICH Forschungszentrum