MATH 676

–

Finite element methods in scientific computing

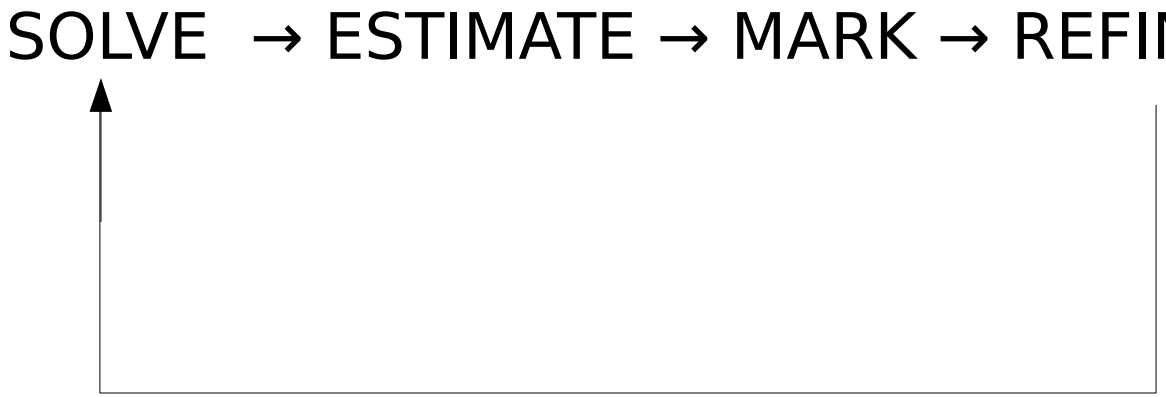
Wolfgang Bangerth, Texas A&M University

Lecture 17.5:

Generating adaptively refined meshes: Which cells to refine

Adaptive mesh refinement (AMR)

Adaptive mesh refinement happens in a loop:



- SOLVE: Assemble linear system, solve it
- ESTIMATE: Compute a refinement indicator for each cell
- MARK: Determine which cells should be refined
- REFINE: Refine these cells (and possibly others)

The MARK phase

Precondition:

In the ESTIMATE phase, we have computed a refinement indicator for each cell:

$$\eta = \{\eta_{K_1}, \eta_{K_2}, \eta_{K_3}, \ldots, \eta_{K_N}\}$$

Goal:

Determine *which* cells need to be refined to obtain the next mesh!

The MARK phase

Strategy 1 (“global refinement”):

Mark *all* cells for refinement.

Advantages:

- Convergence is guaranteed
- Don't even need to compute the refinement indicators

Disadvantages:

- Not an optimal strategy: requires more cells than necessary

The MARK phase

Strategy 2 (“bulk refinement”, “fixed fraction”):

Mark those cells for refinement that (i) have the largest error indicators, (ii) together account for a certain fraction of the error (e.g., 90%).

Advantages (at least for some equations):

- Convergence can be guaranteed theoretically
- Can be shown to lead to optimal meshes

Disadvantages:

- May sometimes refine very few cells (at “singularities”)
- Can be expensive in these cases

The MARK phase

Strategy 3 (“fixed number”):

Mark a fixed fraction of the cells for refinement that have the largest error indicators.
(For example, refine 1/3 of cells in 2d, 1/7 of cells in 3d.)

Advantages:

- Number of cells is guaranteed to grow at a reasonable pace

Disadvantages:

- May not lead to optimal meshes
- May refine too many cells

The MARK phase

Observation:

We typically mark cells based on

- Heuristically derived error indicators
- Error estimators that *overestimate* the true error

Consequence: We sometimes refine the *wrong* cells!

Solution: In each refinement step, also *coarsen* a small number of cells (e.g., 5% of cells) to undo earlier mistakes.

The MARK phase

Implementation: The 3 strategies for refinement are implemented in the following functions:

- Triangulation::refine_global()
- GridRefinement::refine_and_coarsen_fixed_fraction()
- GridRefinement::refine_and_coarsen_fixed_number()

Each of these increases the number of cells.

If the refinement indicators are good, each will eventually yield convergence of the error to zero.

Time dependent problems

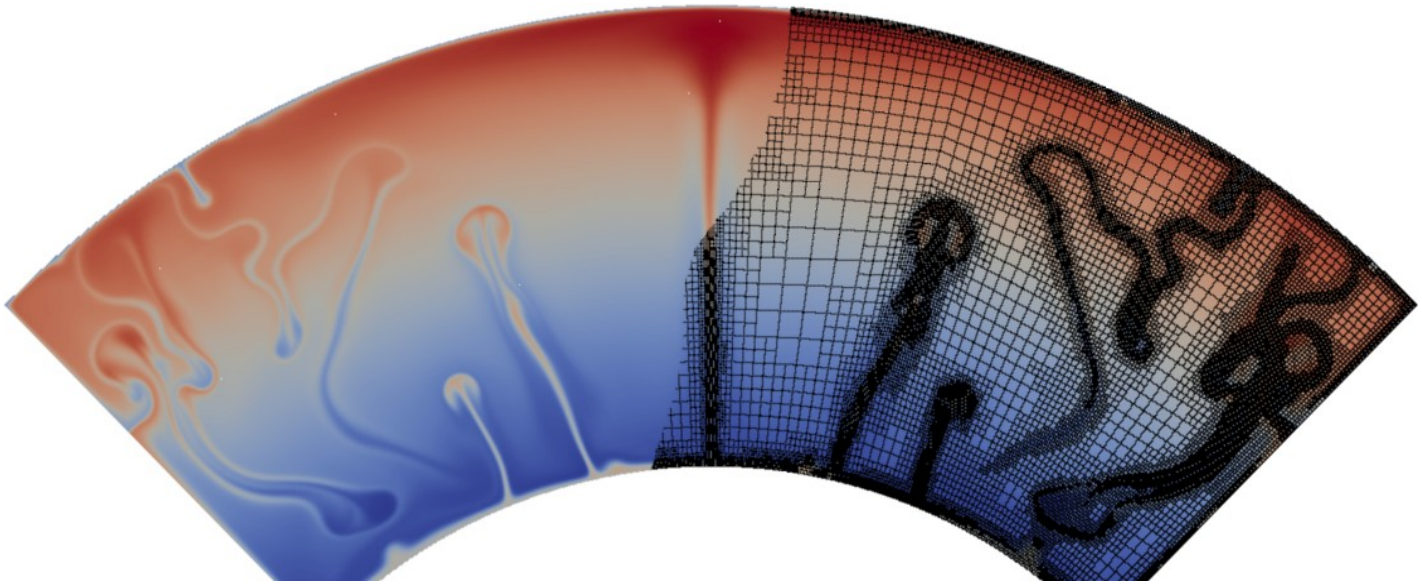
Time-dependent equations (see, for example, step-26, lecture 30):

- Start with a coarse mesh in time step 1
- Refine it a number of times to resolve the solution
- Do time iteration
- Every few time steps, adjust the mesh:
  – start with the mesh from the previous time step
  – coarsen and refine some cells
  – roughly keep number of cells constant

Requires "fixed number" strategy: Mark proportional numbers of cells for refinement and coarsening.

A different perspective

Consider this mesh:

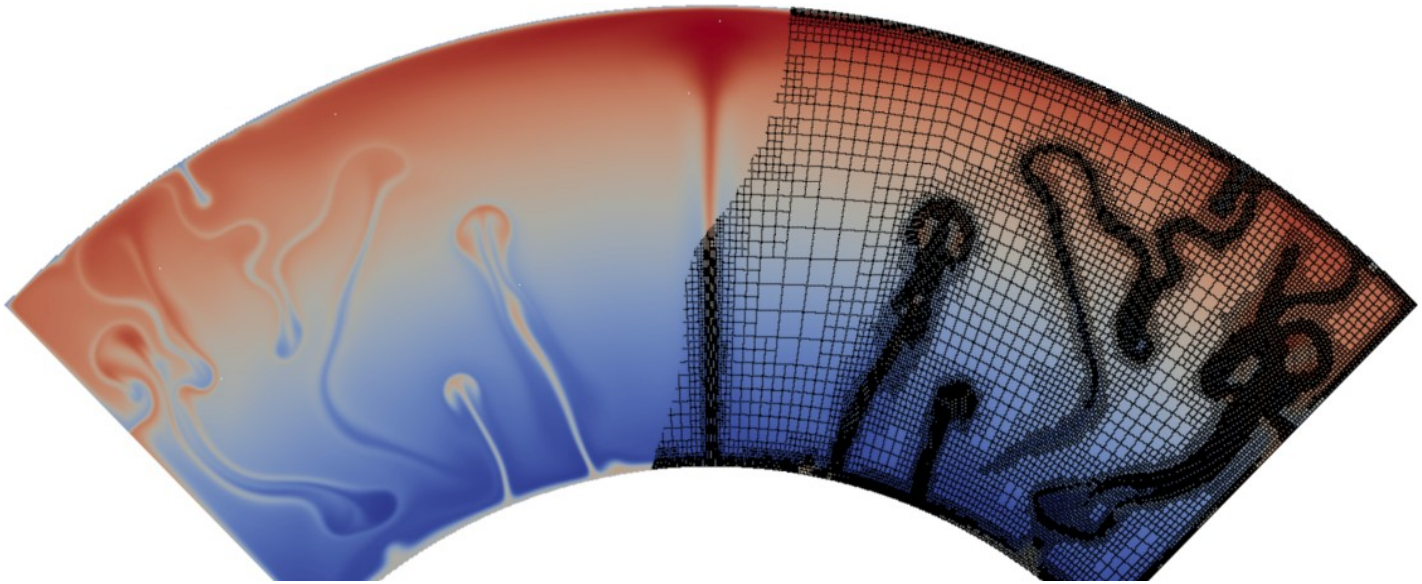


We can say:

Compared to a uniform mesh, we selectively refined to make the solution far more accurate!

A different perspective

Consider this mesh:

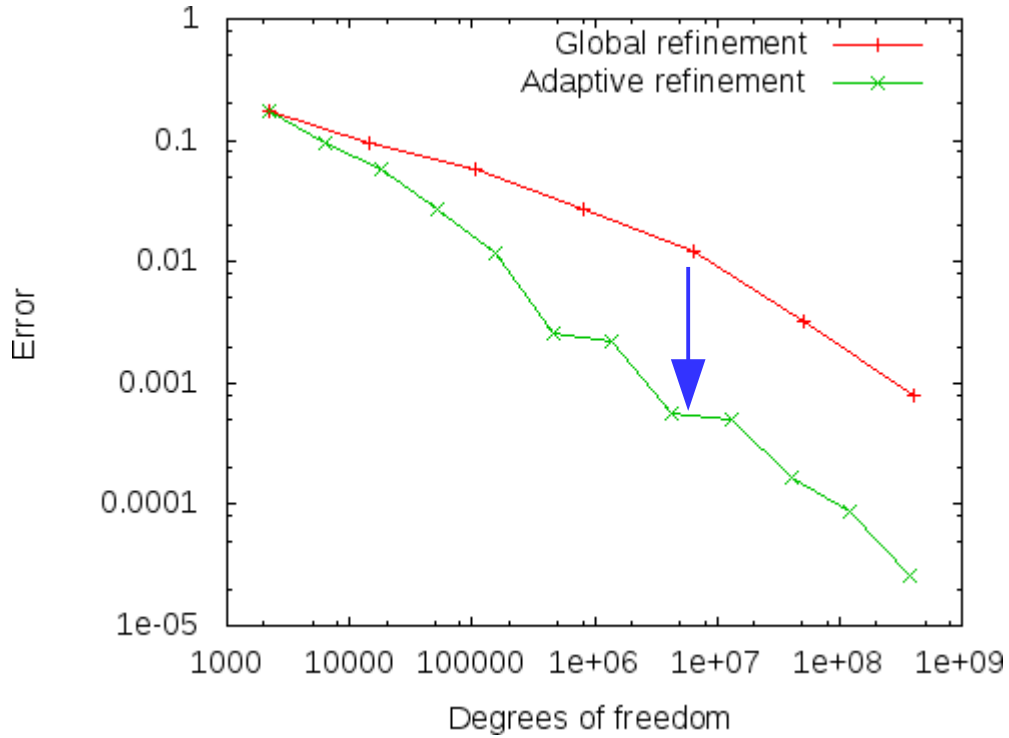


Or we can say:

Compared to a uniform mesh, we selectively coarsened to make the computation far cheaper!

A different perspective

An illustration in a graph:

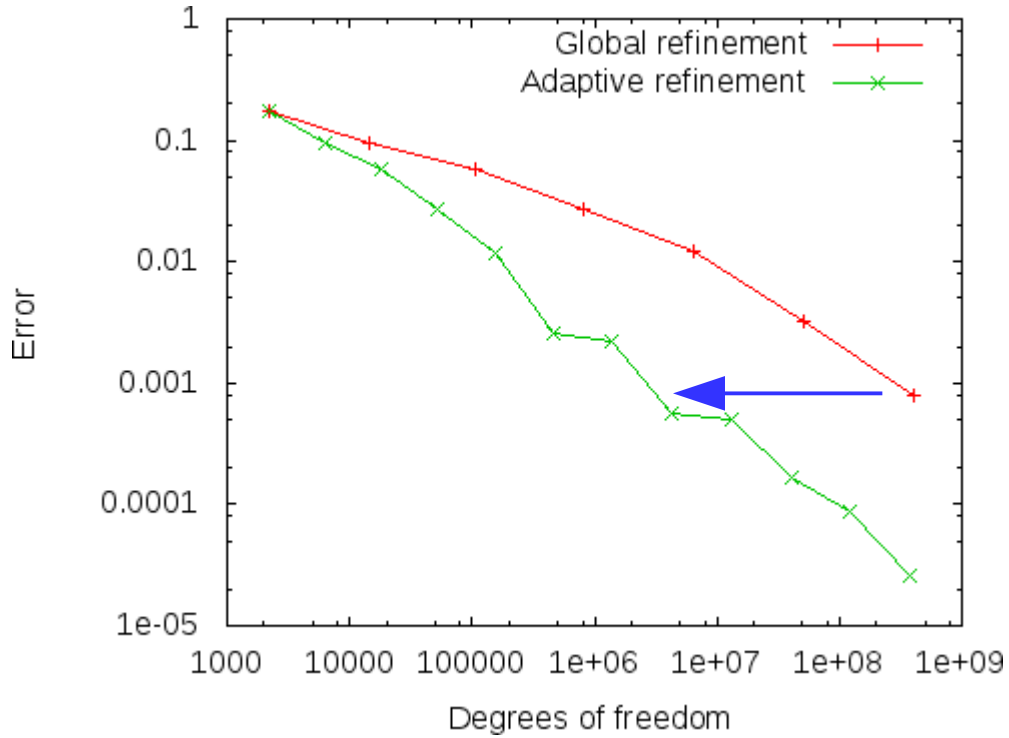


We can say:

Compared to a uniform mesh, we selectively refined to make the solution far more accurate!

A different perspective

An illustration in a graph:



Or we can say:

Compared to a uniform mesh, we selectively coarsened to make the computation far cheaper!

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University