Space and Time Adaptive Two-Mesh *hp*-FEM for the Coupled Problem of Heat and Moisture Transport in Concrete

J. Červený, P. Šolín, J. Kruis

Academy of Sciences of the Czech Republic, Prague
University of Nevada, Reno, USA
Czech Technical University, Prague, Czech Republic

January 6, 2009

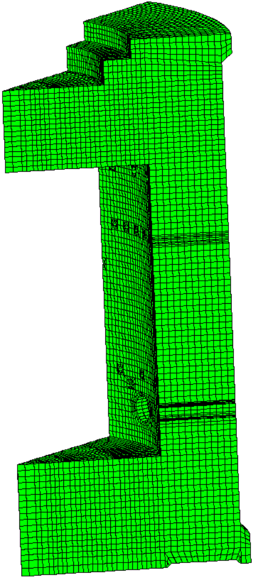
Outline

- Problem: heat and moisture transfer in a reactor vessel
- Automatic hp-adaptivity
- Multi-mesh assembling
- Dynamic meshes

- Results

- Nonlinear problem
- Conclusion

Heat and moisture transfer in a reactor vessel



- Nuclear power plant reactor vessel
- Massive structure made of prestressed concrete
- Height $\approx$ 36 m
- Thickness $\approx$ 5 m
- Will consider simplified axisymmetric model (2D)

Heat and moisture transfer in a reactor vessel

- Unknowns:
  - T – temperature
  - w – relative humidity

- System of linear parabolic equations:

$$c_{TT}\frac{\partial T}{\partial t} - d_{TT}\Delta T - d_{Tw}\Delta w = 0$$

$$c_{ww}\frac{\partial T}{\partial t} - d_{wT}\Delta T - d_{ww}\Delta w = 0$$

- $c_{??}, d_{??}$ – constant coefficients
  (more realistic model: functions of T, w)

The standard approach

- Fixed mesh, as fine as possible
- Standard linear FEM
- No adaptivity
- No error control
- Fixed time step

Our approach

- Dynamic meshes
  - Capture moving fronts more efficiently
- Different mesh for each component
- Higher-order elements
- Full *hp*-adaptivity in space
- Adaptive time step
- Error control both in space and time

hp-FEM

What is *hp*-FEM?

- FEM with variable h and p
- Exponential convergence
- More complicated than standard FEM

hp-FEM

What is *hp*-FEM?

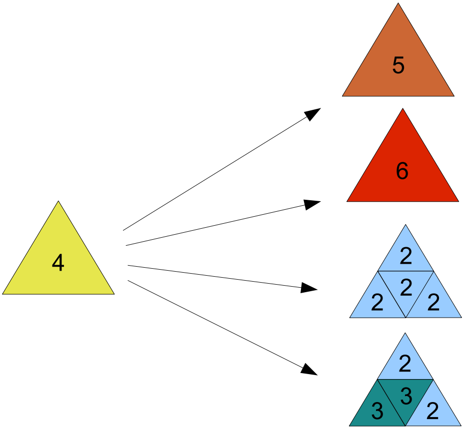
- FEM with variable h and p
- Exponential convergence
- More complicated than standard FEM

hp-adaptivity

- Substantially different from *h*-adaptivity
- Standard error estimates not sufficient
    - Many refinement candidates per element
- More information to handle (complex data structures)

Automatic *hp*-adaptivity

- Many refinement options



and more...

Automatic *hp*-adaptivity

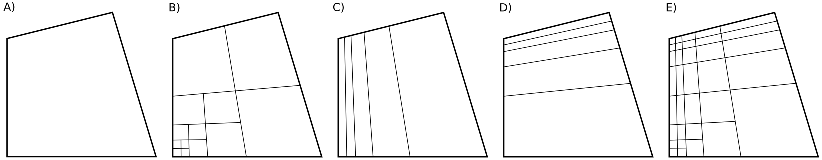
- Error estimation: reference solution
- Hanging nodes (arbitrary level)

- Algorithm:
  - Calculate coarse solution
  - Calculate reference solution
  - Refine elements whose error is too high
    - Select from about 100 refinement candidates
      $\rightarrow$ true *hp*-adaptivity, exponential convergence
  - Repeat

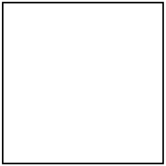
Multiple meshes

Main idea

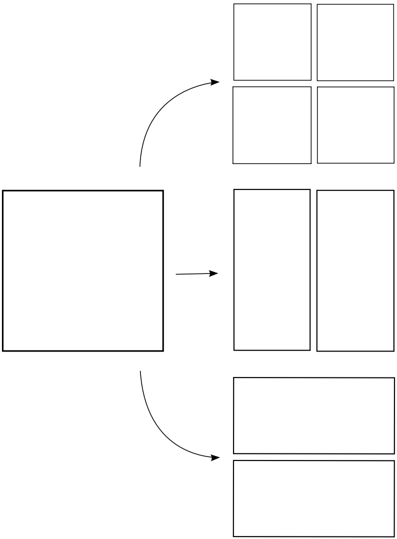
- Very coarse master mesh shared by all components
- Independent refinement of the master mesh at each component



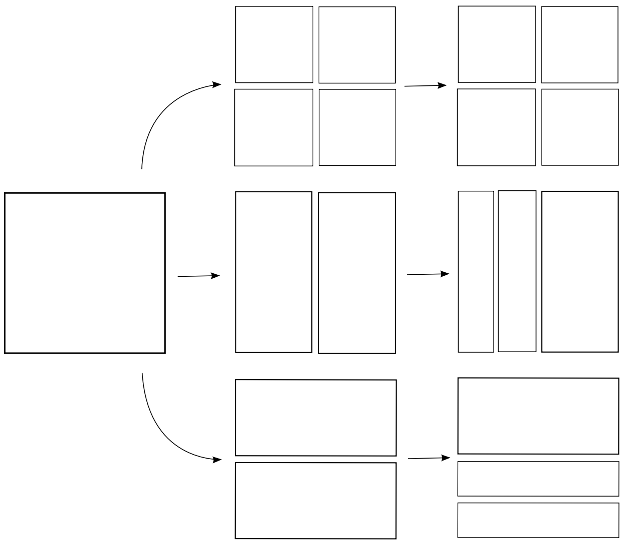
Multiple meshes – Assembling



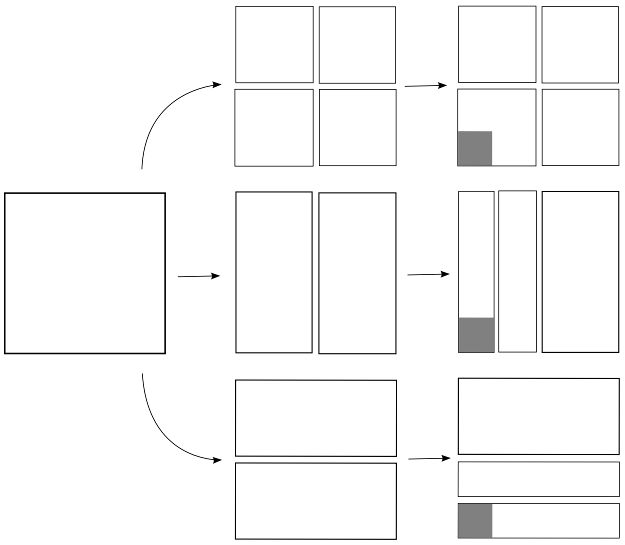
Multiple meshes – Assembling



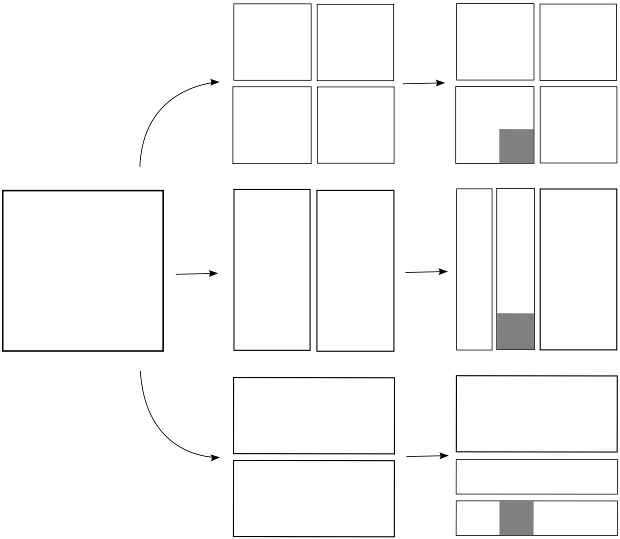
Multiple meshes – Assembling



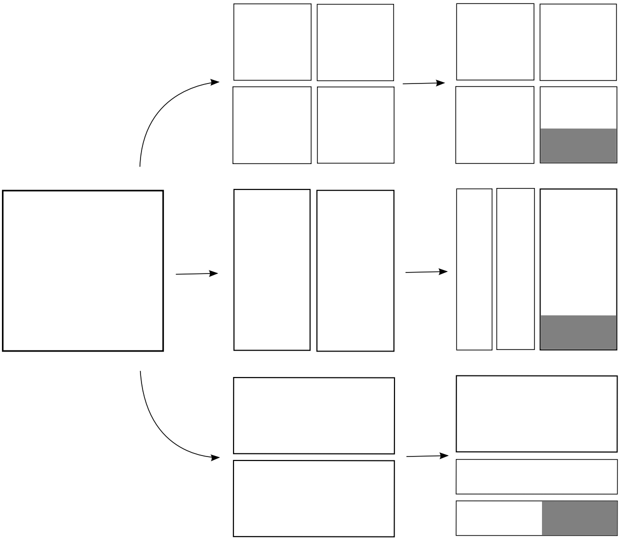
Multiple meshes – Assembling



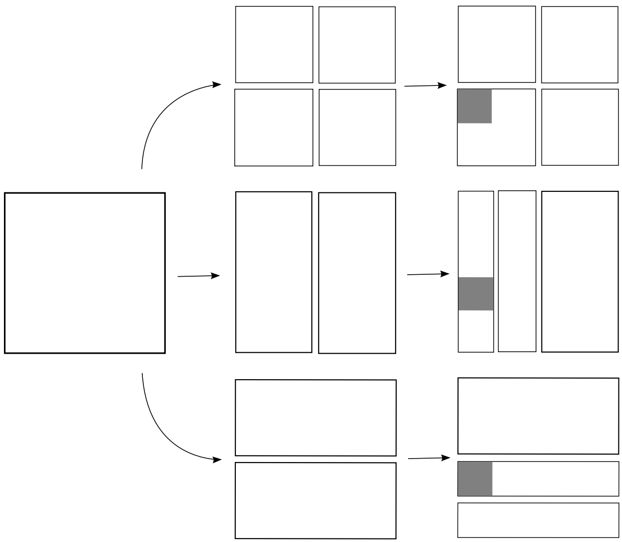
Multiple meshes – Assembling



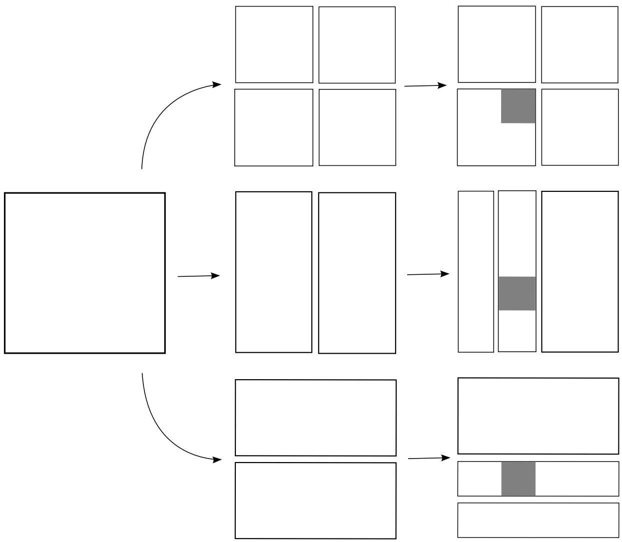
Multiple meshes – Assembling



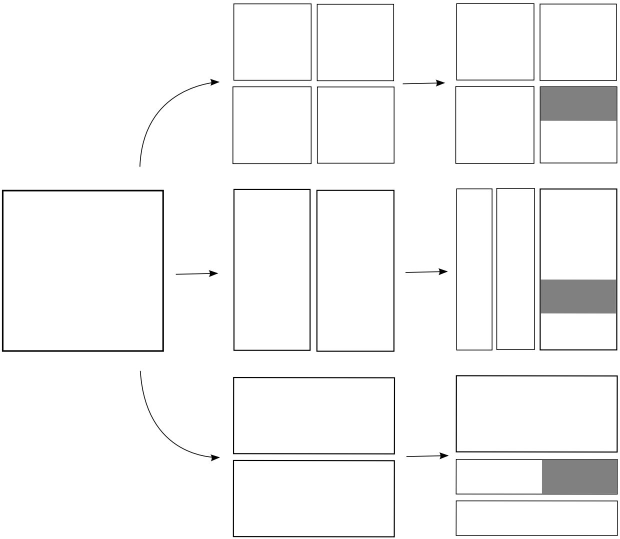
Multiple meshes – Assembling



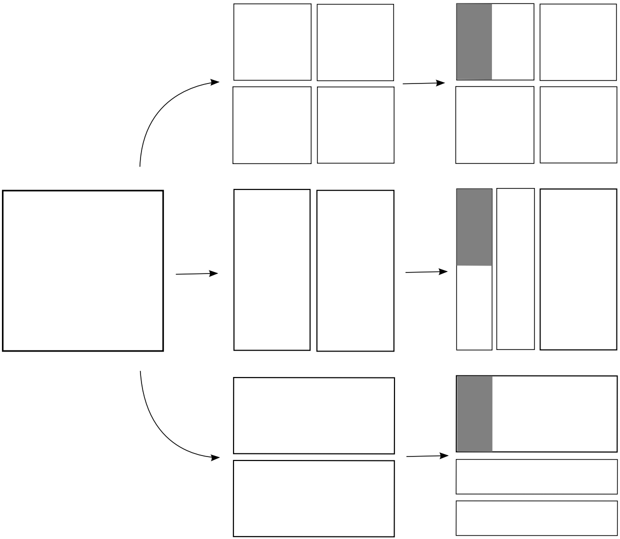
Multiple meshes – Assembling



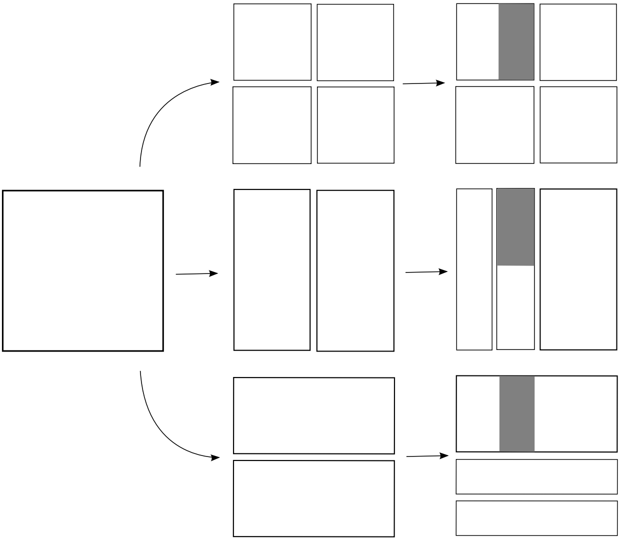
Multiple meshes – Assembling



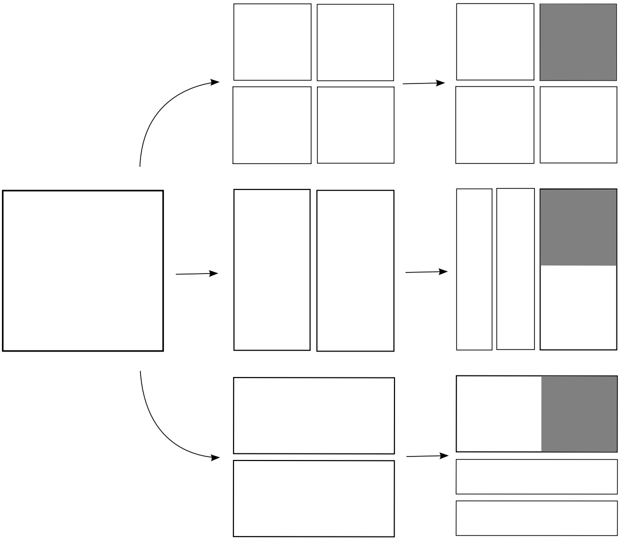
Multiple meshes – Assembling



Multiple meshes – Assembling



Multiple meshes – Assembling



Multiple meshes – Assembling

Adaptive time step

- Simple PID controller from Control Theory
- The controller regulates Δt with the goal to keep

$$e_n = \frac{||u_{n+1} - u_n||}{||u_{n+1}||}$$

under the prescribed tolerance.

- If the tolerance is not met the solution u_{n+1} is thrown away and recalculated with a smaller Δt.
- Otherwise the time step is increased smoothly towards the limit.
- Very cheap, in the future more rigorous adaptivity by comparing first and second order Euler.

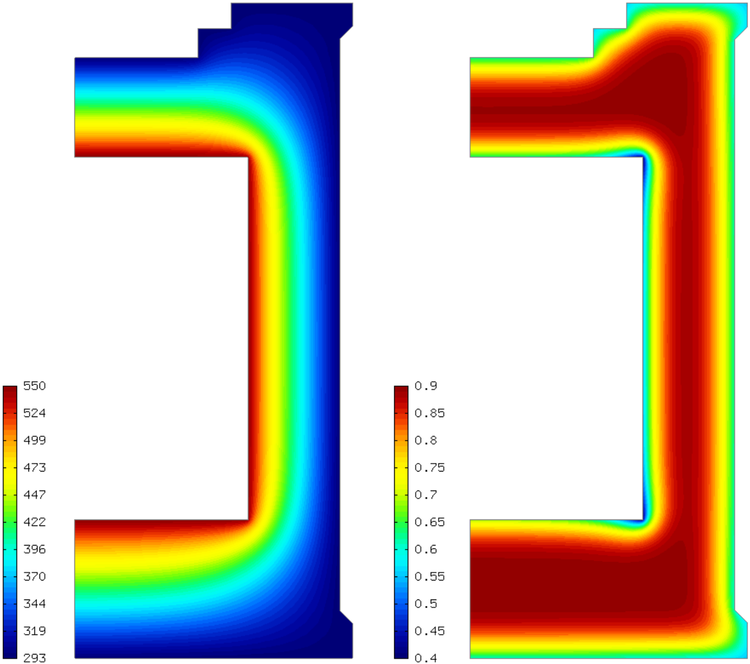
dynamic meshes

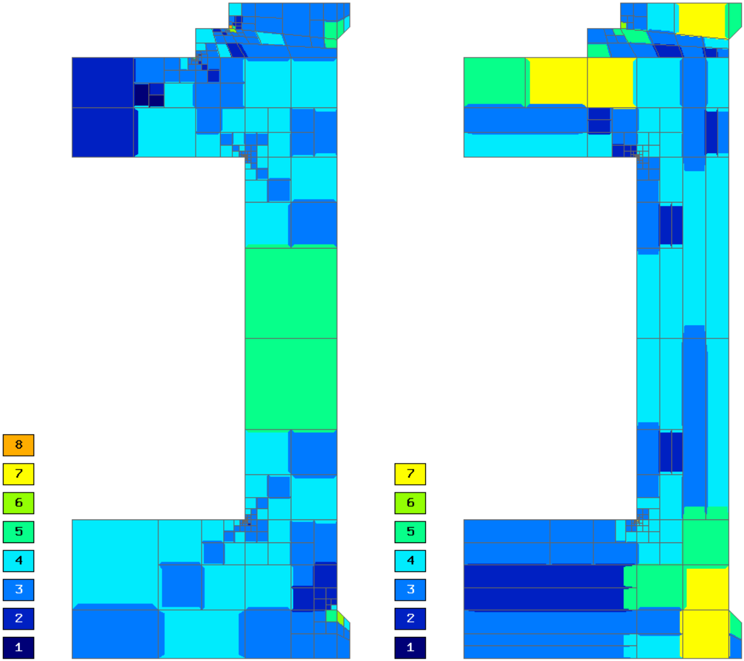
- Semi-discretization in time by Rothe method
  $\rightarrow$ series of time-independent problems
  $\rightarrow$ apply separate hp-adaptivity for each time level

- Successive time levels can have different meshes thanks to multi-mesh assembling

- Speedup: reuse meshes from previous time step

- Only start from the coarse mesh every 10 time steps

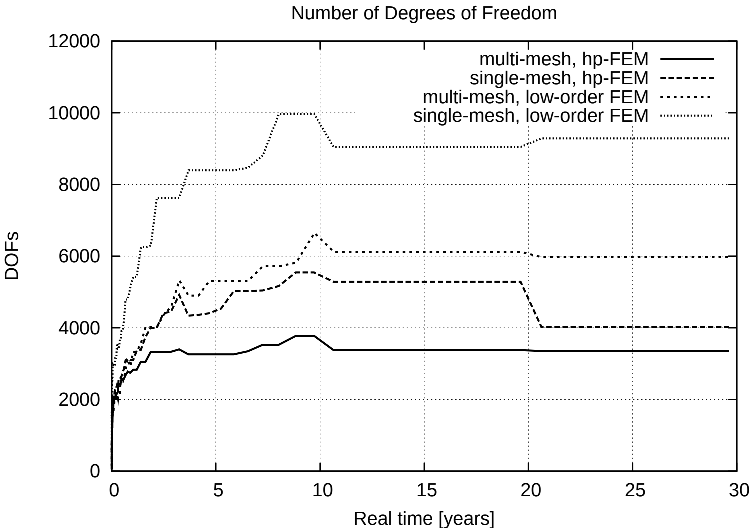

550
524
499
473
447
422
396
370
344
319
293
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4


8
7
6
5
4
3
2
1
7
6
5
4
3
2
1

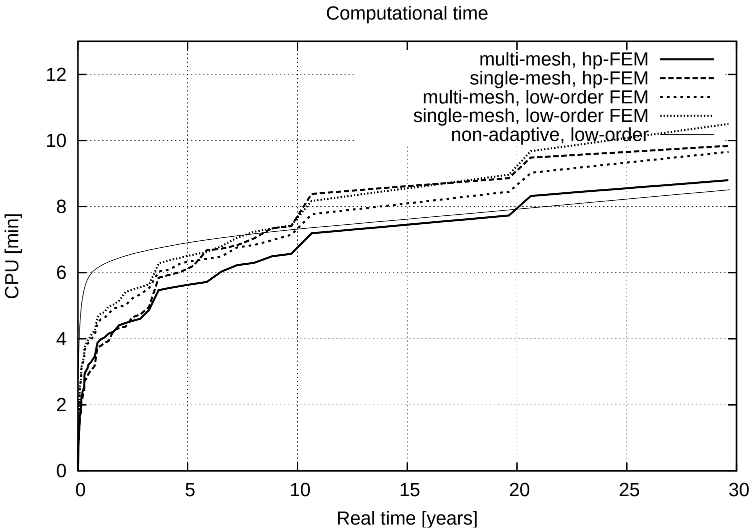
Comparisons

1. Multi-mesh approach on *hp*-meshes dynamicly changing
2. Both components share the *hp*-mesh (single mesh), dynamic mesh
3. Multi-mesh approach on quadratic elements using *h*-adaptivity, dynamic meshes
4. Both components share the mesh with quadratic elements, dynamic mesh
5. Computation on fixed mesh for both components and for all time steps, no adaptivity in space at all (sufficiently fine mesh suitable for all steps)

Results - DOFs

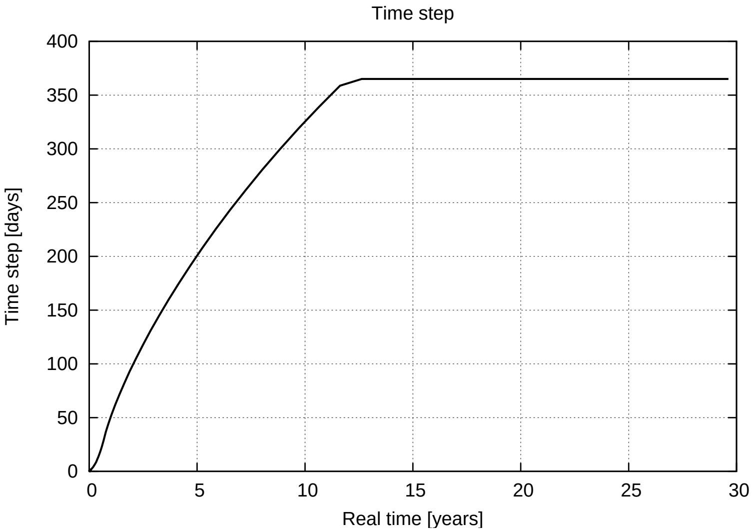


Results - CPU



Computational time
multi-mesh, hp-FEM
single-mesh, hp-FEM
multi-mesh, low-order FEM
single-mesh, low-order FEM
non-adaptive, low-order
CPU [min]
0
2
4
6
8
10
12
0
5
10
15
20
25
30
Real time [years]

Results - length of time step



Example with a moving front

- Nonlinear example – combustion problem:

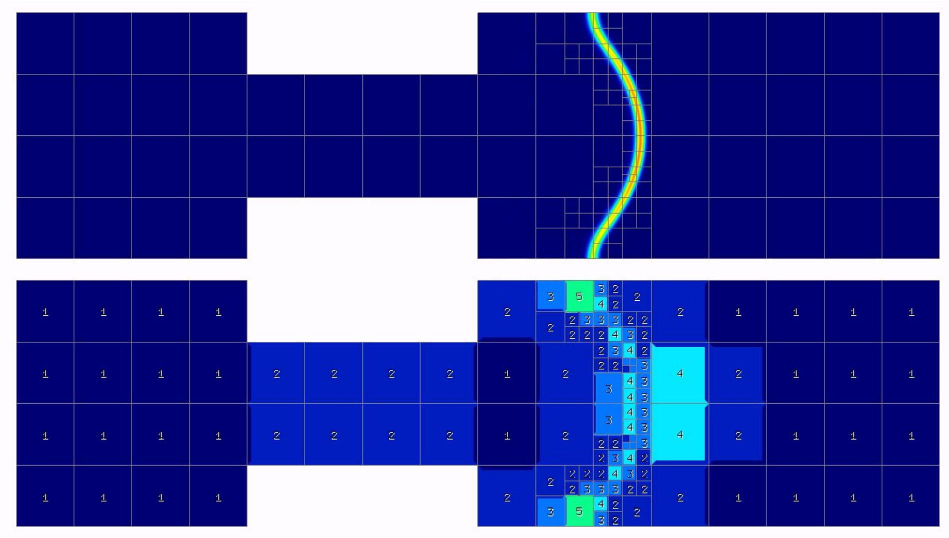
$$\begin{aligned} \frac{\partial \theta}{\partial t} - \Delta\theta &= \omega(\theta, Y) \\ \frac{\partial Y}{\partial t} - \Delta Y &= -\omega(\theta, Y) \end{aligned}$$

- θ – temperature (rises from 0 to 1)
- Y – fuel concentration (decreases from 1 to 0)
- $\omega(\theta, Y)$ – reaction rate:

$$\omega(\theta, Y) = \frac{\beta^2}{2} Y \cdot \exp \frac{\beta(\theta - 1)}{1 + \alpha(\theta - 1)}$$

- α, β – physical constants

Example with a moving front



Conclusion and Outlook

- Benefits of multi-mesh assembling:
  - Allows different meshes for different components.
  - Allows dynamic meshes in time-dependent problems.
  - Saves degrees of freedom $\rightarrow$ faster solution of the linear system.
- Costs: slower assembling.
- In the future:
  - More advanced multi-physics coupled problems
  - Magneto-hydro-dynamics...

Hermes2D

- Free software (GNU GPL)
- Modular C++/Python hp-FEM library
- From stationary linear equations to nonlinear time-dependent multi-physics PDE systems
- Several types of finite elements – H^1-conforming, H(curl)-conforming or L^2-conforming elements
- Automatic adaptivity algorithms on meshes with hanging nodes
- Multi-mesh technology
- `http://spilka.math.unr.edu/projects/hermes2d`