Combined Adaptive Multimesh hp-FEM/hp-DG for Multiphysics Coupled Problems of Compressible Flow

Lukas Korous, Milan Hanus, Pavel Solin

University of Nevada, Reno

July 14, 2011

We acknowledge the financial support by Subcontract No. 00089911 of Battelle Energy Alliance (U.S. Department of Energy intermediary)

Motivation

In multiphysics problems, physical field exhibit qualitative differences:

- singularities
- boundary layers
- smooth areas

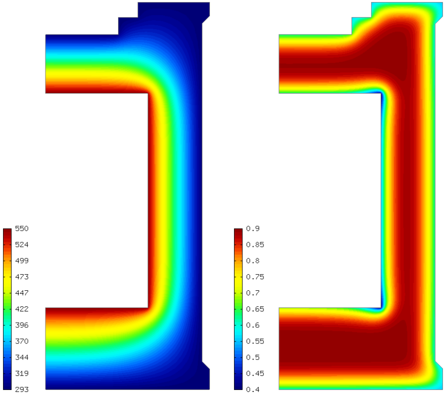
These phenomena require specific treatment.

Our goal: approximate every physical quantity in an optimal way.

- appropriate mesh (for each component of the system)
- appropriate Sobolev space ($H^1, L^2, H_{curl}, H_{div}$, ..)
- appropriate method (continuous FEM, DG method) of higher order

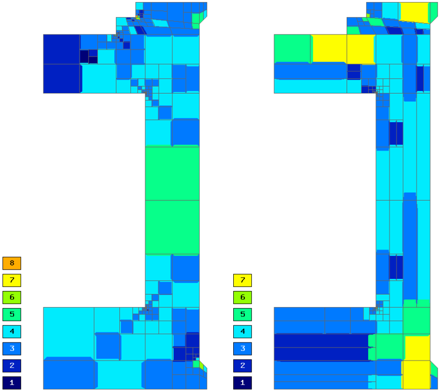
Component specific meshes example

Coupled system of linear second-order PDE
Heat and moisture transfer in concrete

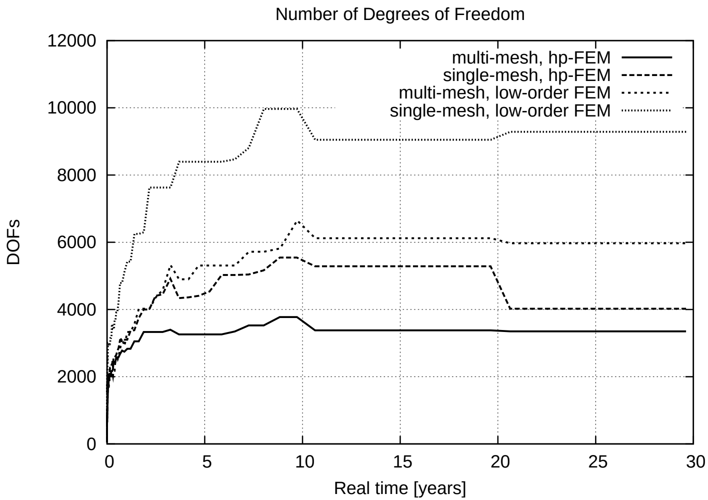


Component specific meshes example

Coupled system of linear second-order PDE
Heat and moisture transfer in concrete.



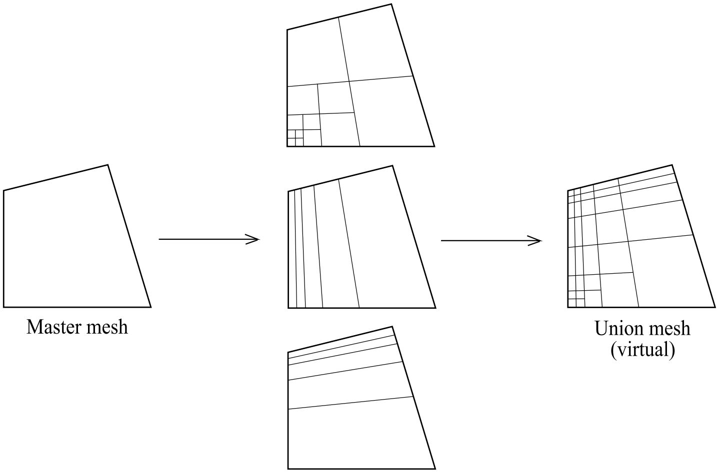
Component specific meshes example - DOFs



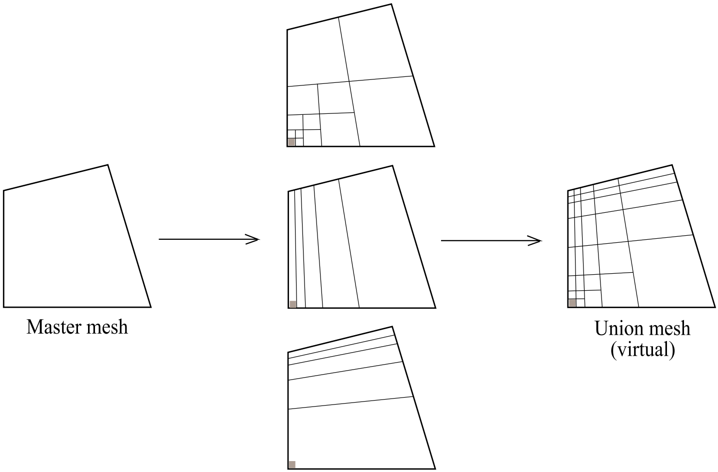
Monolithic Multimesh Adaptive FEM

- meshes come from one **common coarse mesh** (can be one element) that we call master mesh
- every mesh in the system is obtained from this one via an **arbitrary sequence of** elementary **refinements**
- adaptivity puts all elements of all meshes into one array, error estimate as a difference between the coarse and reference mesh approximations
- elements sorted according to the error, elements with the largest errors are **refined as in the standard hp-FEM**
  - well suited for time multiscale transient problems

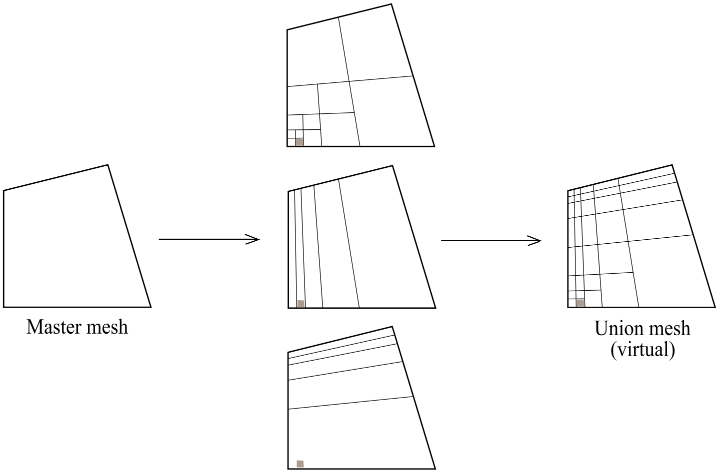
Monolithic Adaptive Multimesh FEM



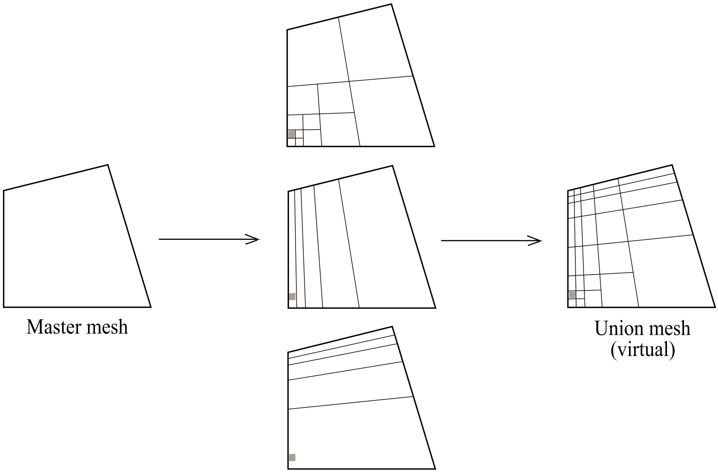
Monolithic Adaptive Multimesh FEM



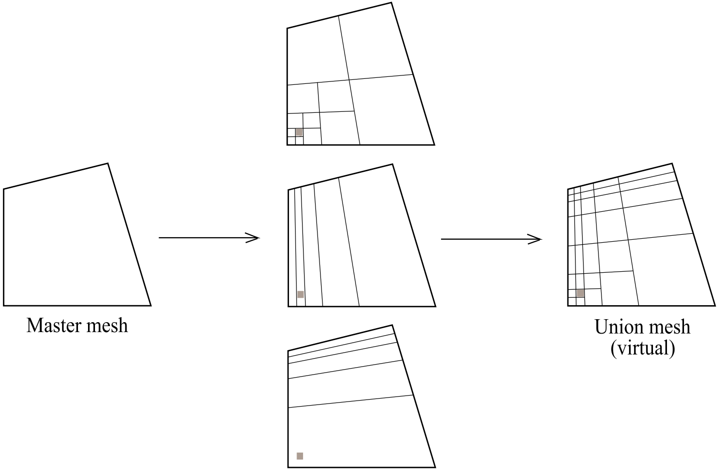
Monolithic Adaptive Multimesh FEM



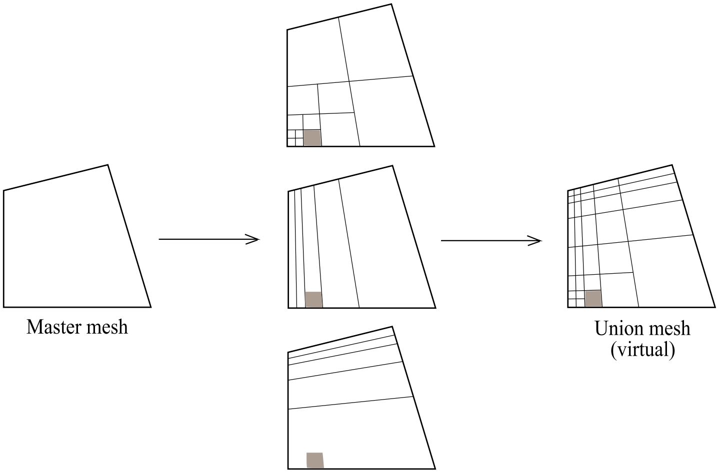
Monolithic Adaptive Multimesh FEM



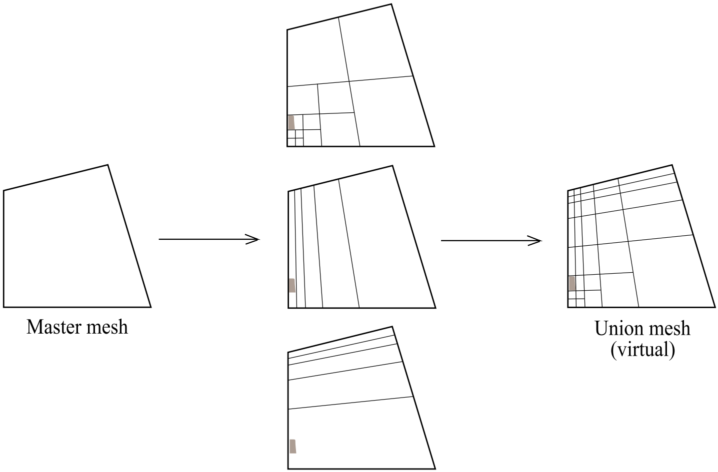
Monolithic Adaptive Multimesh FEM



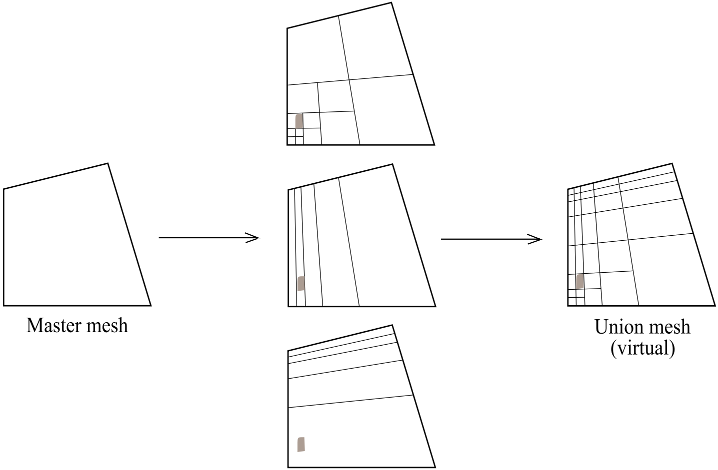
Monolithic Adaptive Multimesh FEM



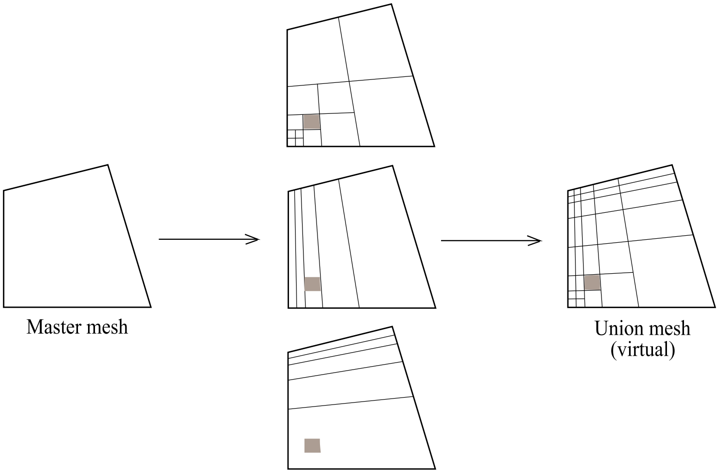
Monolithic Adaptive Multimesh FEM



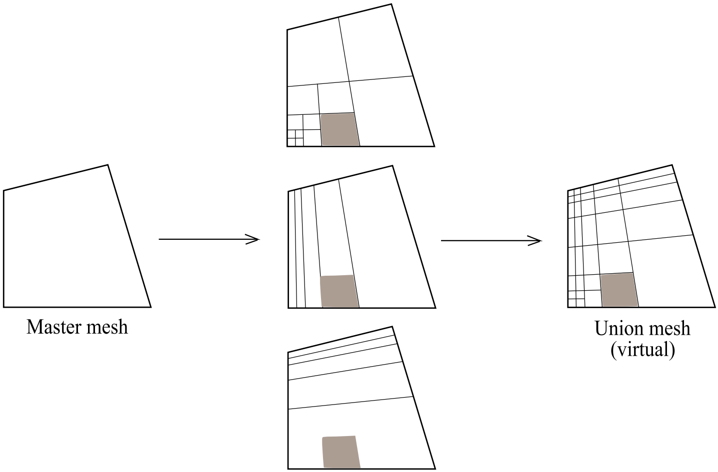
Monolithic Adaptive Multimesh FEM



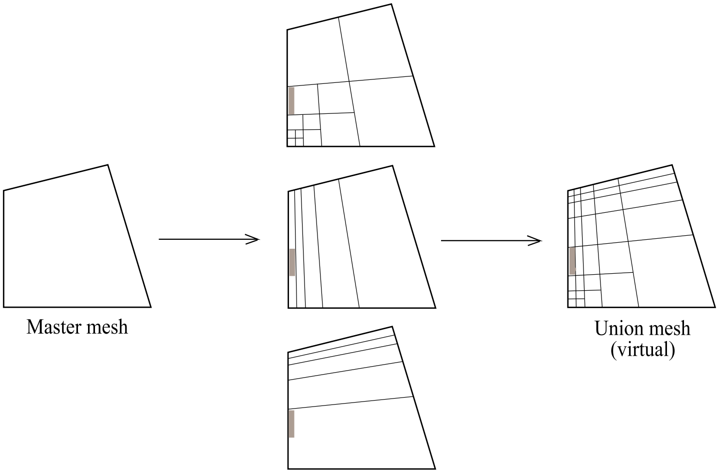
Monolithic Adaptive Multimesh FEM



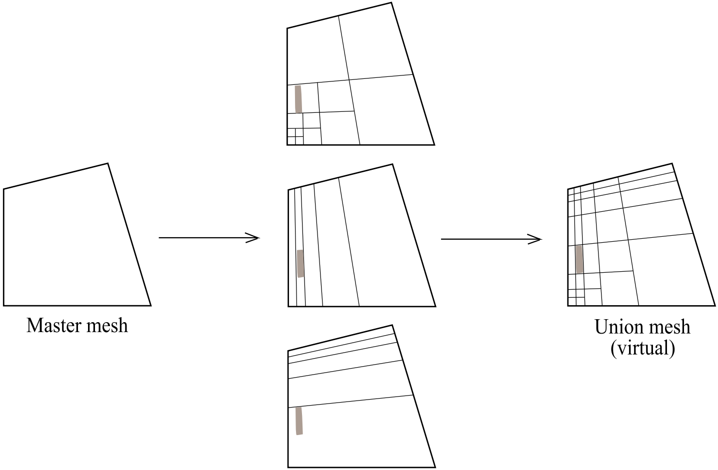
Monolithic Adaptive Multimesh FEM



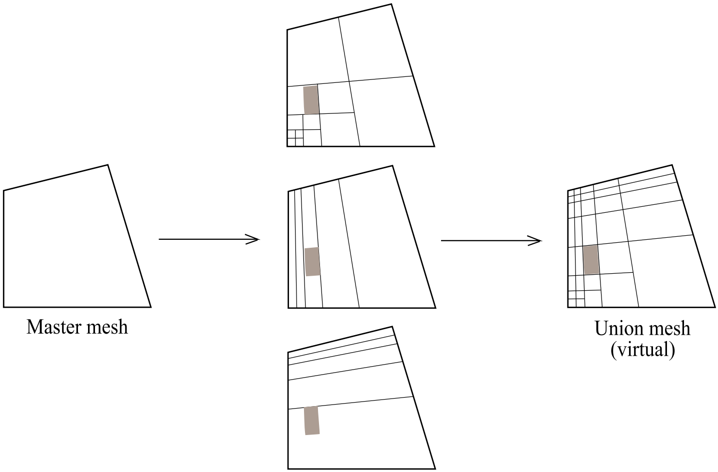
Monolithic Adaptive Multimesh FEM



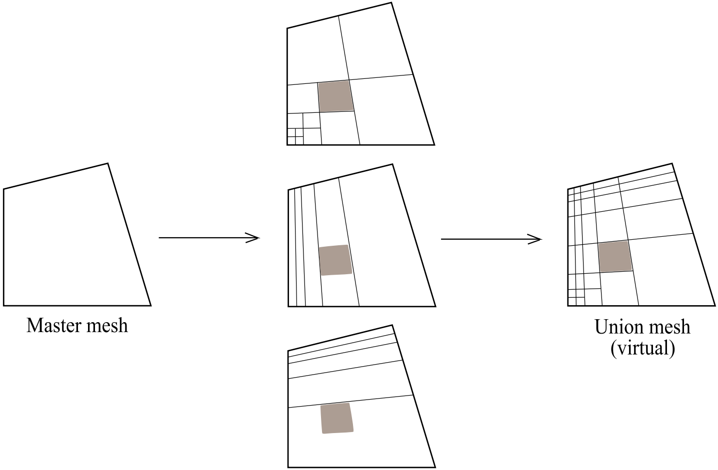
Monolithic Adaptive Multimesh FEM



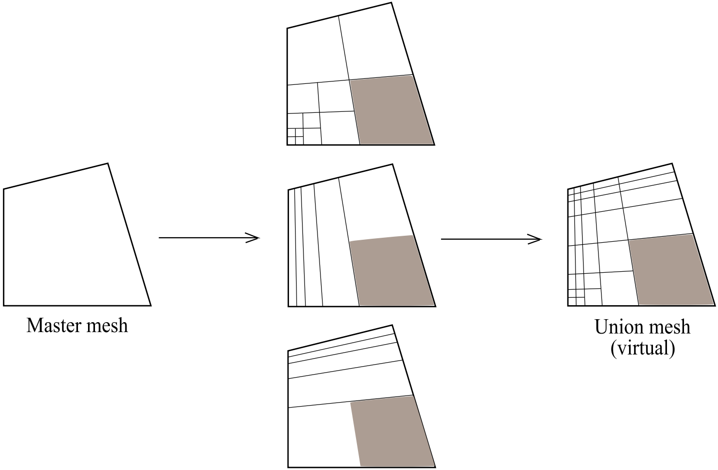
Monolithic Adaptive Multimesh FEM



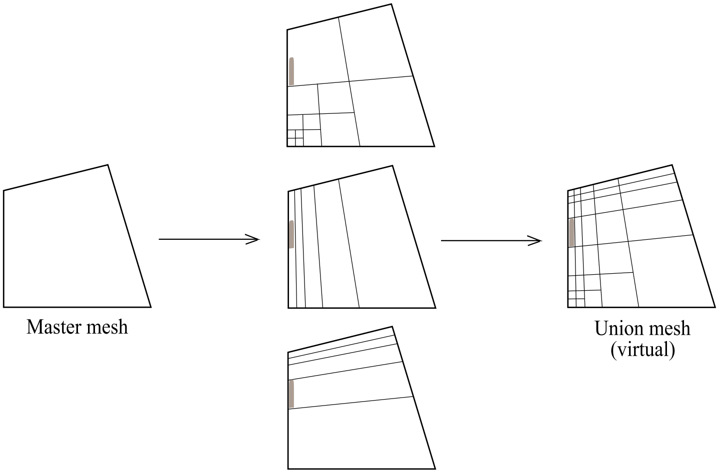
Monolithic Adaptive Multimesh FEM



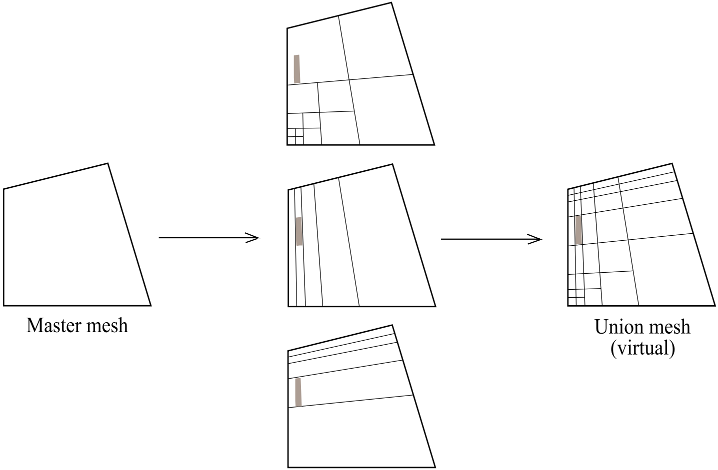
Monolithic Adaptive Multimesh FEM



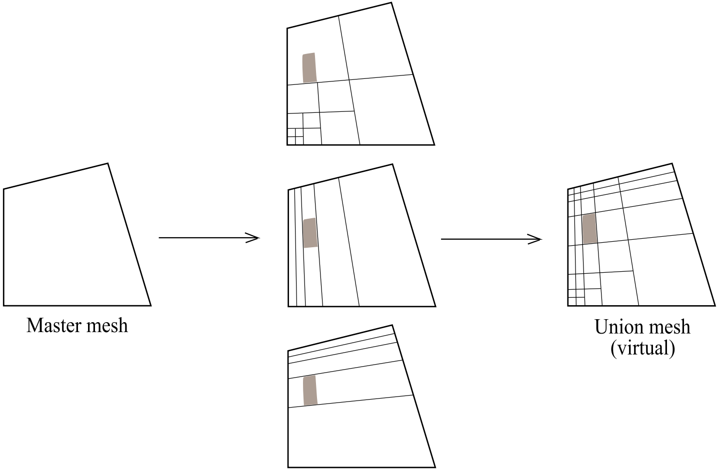
Monolithic Adaptive Multimesh FEM



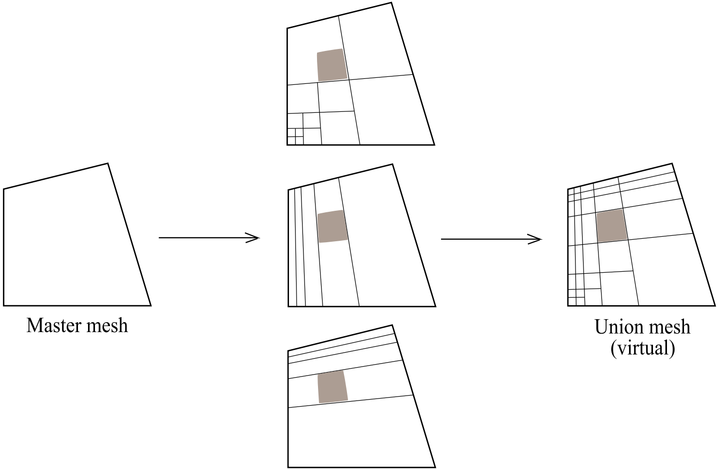
Monolithic Adaptive Multimesh FEM



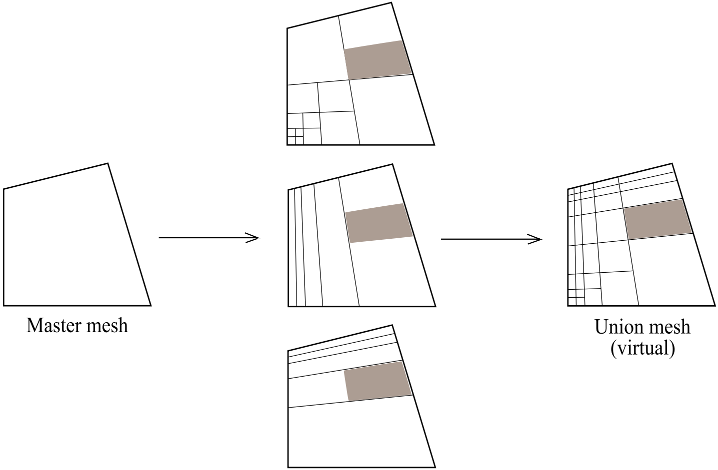
Monolithic Adaptive Multimesh FEM



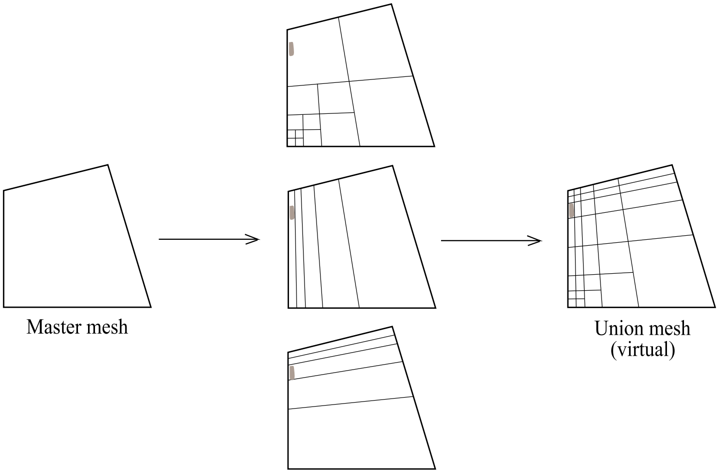
Monolithic Adaptive Multimesh FEM



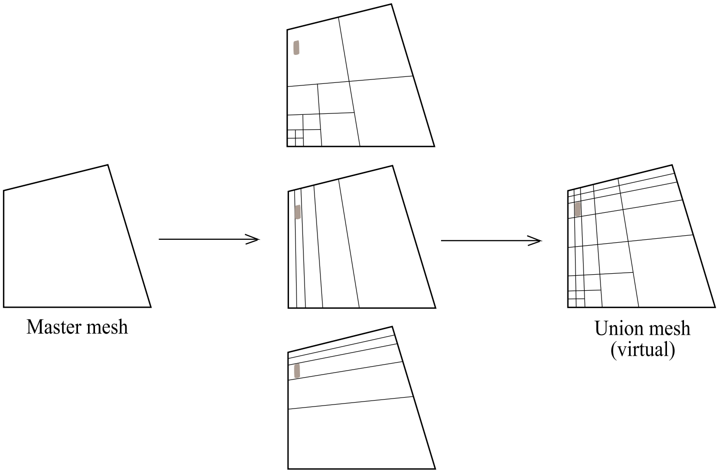
Monolithic Adaptive Multimesh FEM



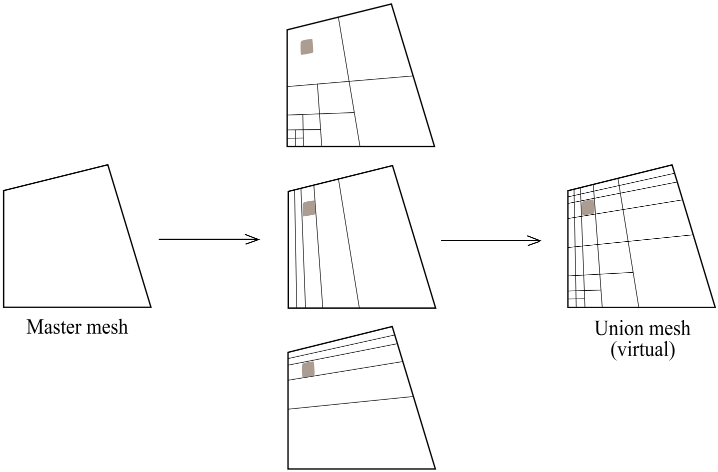
Monolithic Adaptive Multimesh FEM



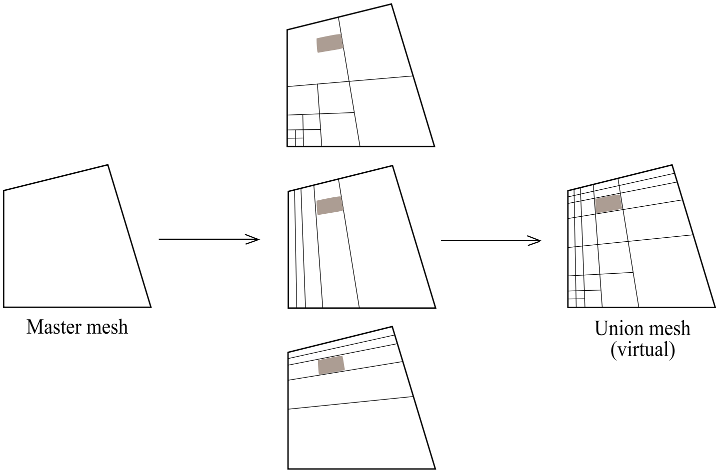
Monolithic Adaptive Multimesh FEM



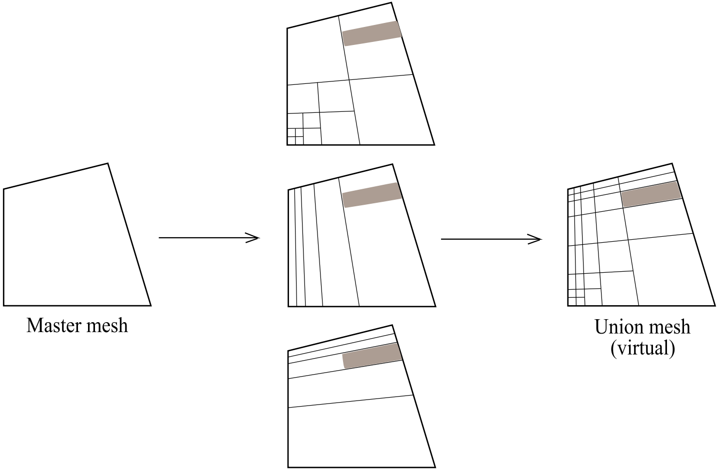
Monolithic Adaptive Multimesh FEM



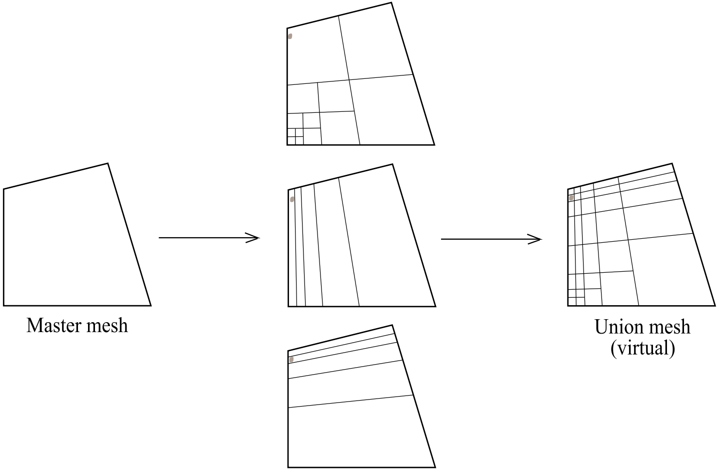
Monolithic Adaptive Multimesh FEM



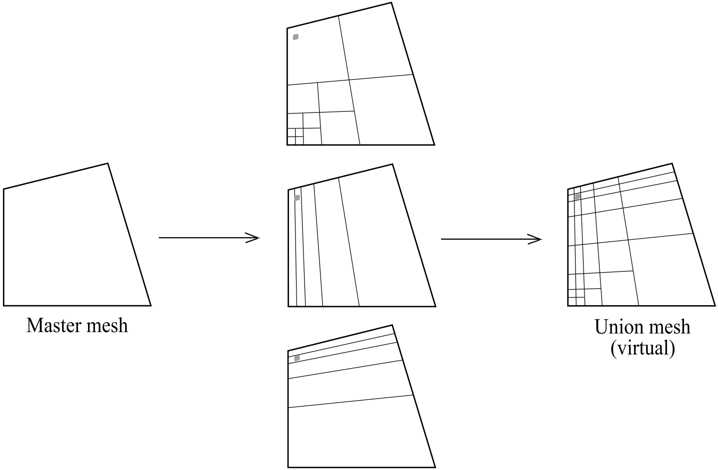
Monolithic Adaptive Multimesh FEM



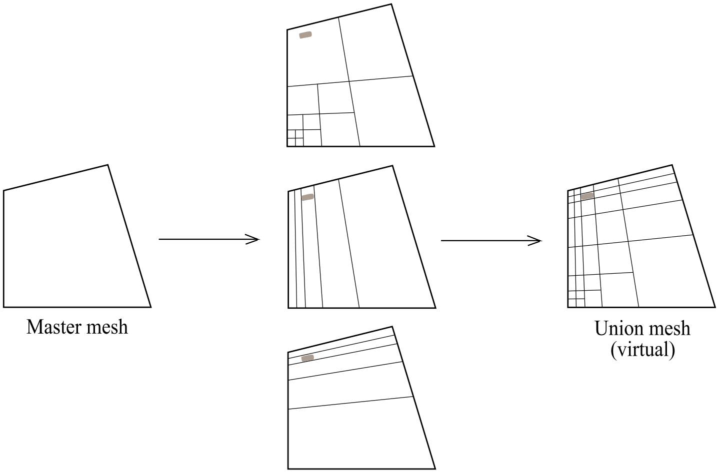
Monolithic Adaptive Multimesh FEM



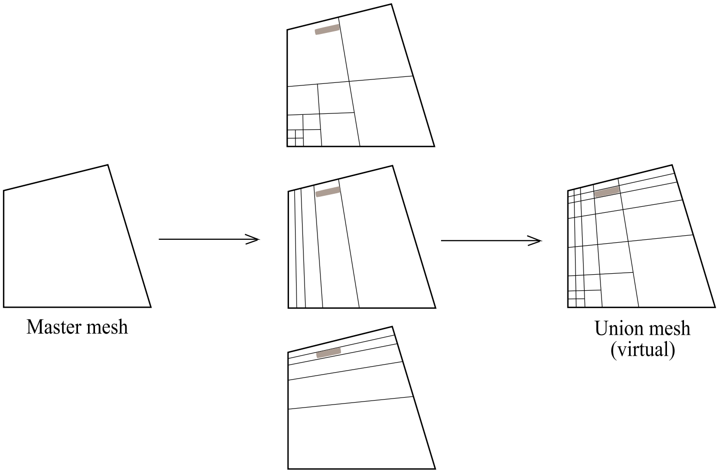
Monolithic Adaptive Multimesh FEM



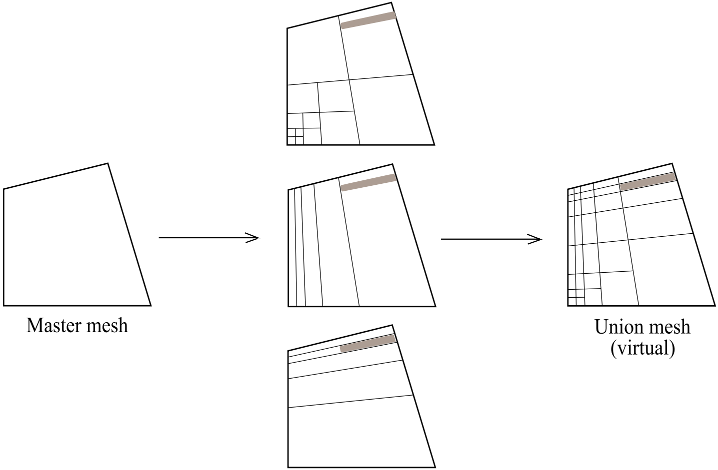
Monolithic Adaptive Multimesh FEM



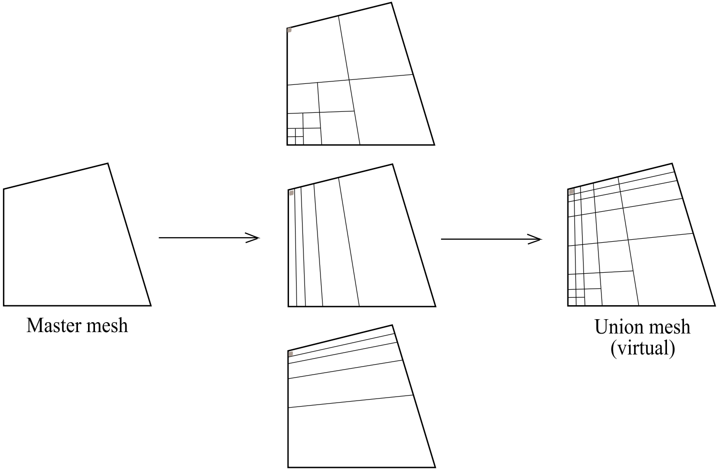
Monolithic Adaptive Multimesh FEM



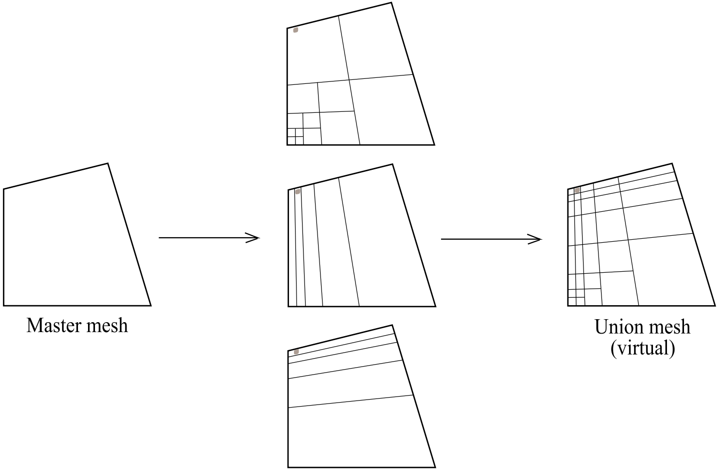
Monolithic Adaptive Multimesh FEM



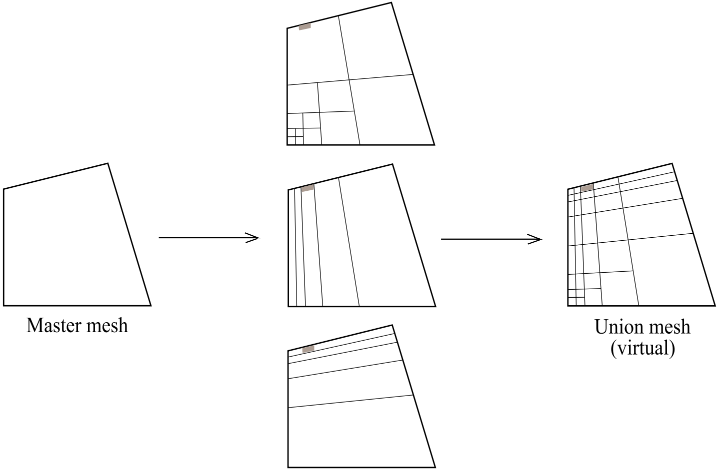
Monolithic Adaptive Multimesh FEM



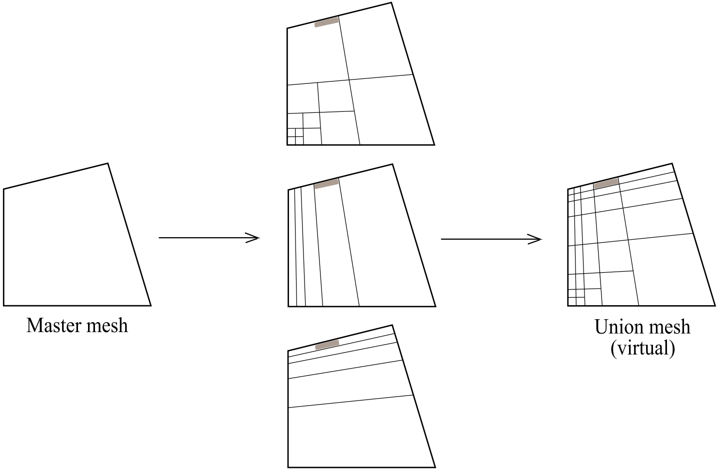
Monolithic Adaptive Multimesh FEM



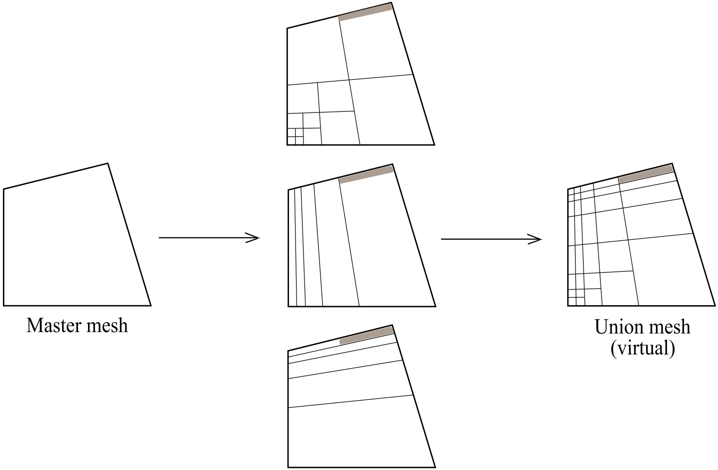
Monolithic Adaptive Multimesh FEM



Monolithic Adaptive Multimesh FEM



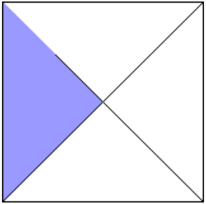
Monolithic Adaptive Multimesh FEM



Hanging nodes

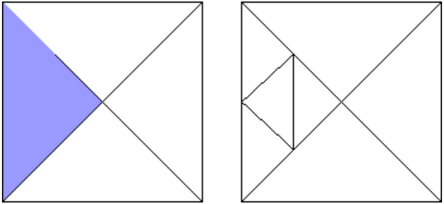
Hanging nodes

- Regular mesh



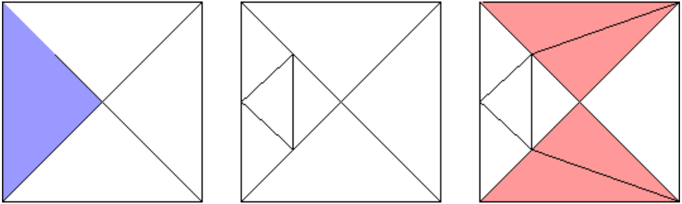
Hanging nodes

- Regular mesh



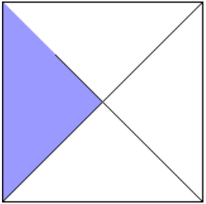
Hanging nodes

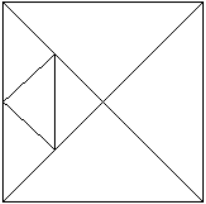
- Regular mesh

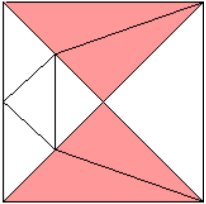


Hanging nodes

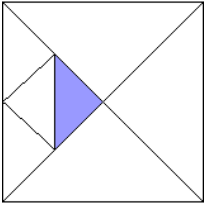
- Regular mesh





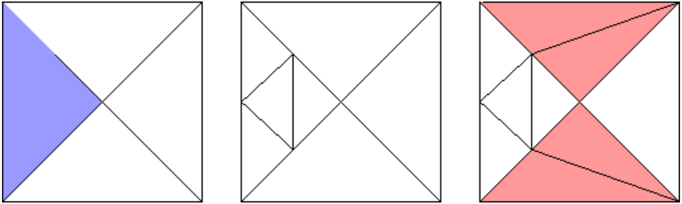


- One-level hanging nodes (1-irregular mesh)

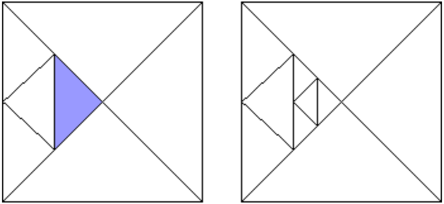


Hanging nodes

- Regular mesh

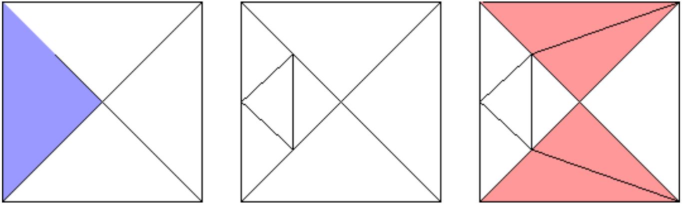


- One-level hanging nodes (1-irregular mesh)

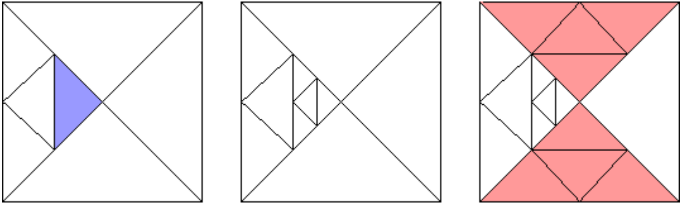


Hanging nodes

- Regular mesh

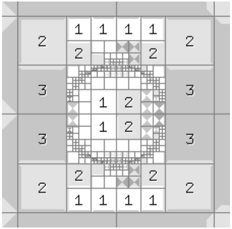


- One-level hanging nodes (1-irregular mesh)



Hanging nodes

- Forced refinements
  - introduce unnecessary DOF
  - spoil element shapes
  - have recursive nature
  - cause incompatible refinements in the multi-mesh *hp*-FEM
- Arbitrary-level hanging nodes:



P.S., J. Cerveny, I. Dolezel: Arbitrary-Level Hanging Nodes and Automatic Adaptivity in the hp-FEM, Math. Comput. Simul. 77 (2008), 117 - 132.

Component specific function spaces

Component specific function spaces

Mathematically, solutions of different equations lie in different function spaces, e.g.

- Temperature, displacement, etc. - H^1 $(H^2, ..)$.
- Electromagnetics - H_{curl} (discontinuous vector fields with continuous tangential components along all mesh edges $\sim$ gradients of H^1 functions).
- Magnetism - H_{div} (discontinuous vector fields with continuous normal components along all mesh edges $\sim$ divergences of H^1 functions).
- Pressure (in Navier-Stokes equations) - L^2.

What about compressible flow? H^1 too small, L^2 too big.

Discontinuous Galerkin method

- discontinuous piecewise polynomial approximation
- local character
- well suited for conservation laws
- the non-uniqueness of the solution on mesh edges is handled as in FVM by introducing a suitable numerical flux

DG methods

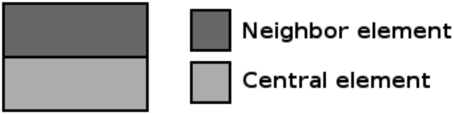
- Low order ($p = 0$) $\sim$ Finite Volume Method
  - easy to implement
  - robust
  - need for higher order approximation
- Higher order ($p > 0$, hp-adaptivity)
  - good properties from Finite Volume Method
  - possible need of handling oscillations (undershoots/overshoots)

Arbitrary-level hanging nodes with DG

- values from both sides of the edge
- correct pairing of integration points

Arbitrary-level hanging nodes with DG

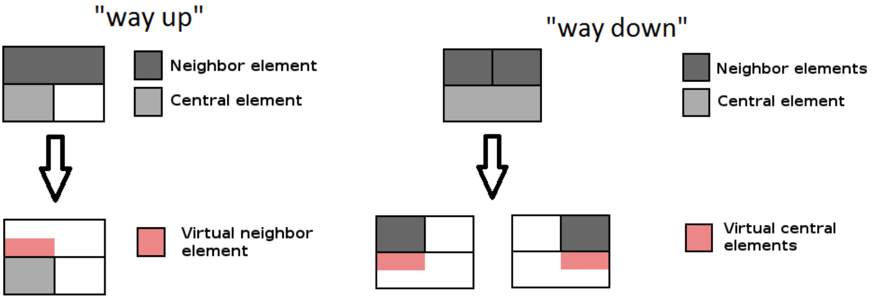
Simple case



- equal integration order on the edge from both sides $\Rightarrow$ identical integration points

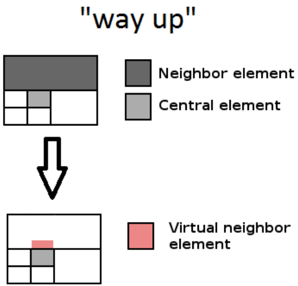
Arbitrary-level hanging nodes with DG

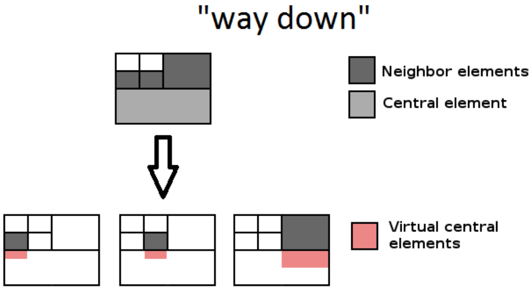
1-level hanging nodes
use of virtual sub-elements



Arbitrary-level hanging nodes with DG

2-level hanging nodes
use of virtual sub-elements





Model problem

Compressible Euler equations and the advection-diffusion equation (concentration)

- compressible Euler equations
  - first order system of hyperbolic laws
- linear advection-diffusion equation
  - second order equation

Euler equations

Non-conservative form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1)$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}^T) + \nabla p = 0 \quad (2)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u}(E + p)) = 0, \quad (3)$$

where $E = \rho e + \frac{1}{2}\rho u^2$.

Conservative form for DG discretization

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}_x}{\partial x} + \frac{\partial \mathbf{f}_y}{\partial y} = 0, \tag{4}$$

where

$$\mathbf{w} = \begin{pmatrix} \varrho \\ \rho u_1 \\ \rho u_2 \\ E \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} \mathbf{f}_x = \begin{pmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(E+p) \end{pmatrix} = \begin{pmatrix} w_1 \\ \frac{w_1^2}{w_0} + p \\ \frac{w_1 w_2}{w_0} \\ \frac{w_1}{w_0}(w_3 + p) \end{pmatrix}$$

$$\mathbf{f}_y = \begin{pmatrix} \rho u_2 \\ \rho u_2 u_1 \\ \rho u_2^2 + p \\ u_2(E+p) \end{pmatrix} = \begin{pmatrix} w_2 \\ \frac{w_2 w_1}{w_0} \\ \frac{w_2^2}{w_0} + p \\ \frac{w_2}{w_0}(w_3 + p) \end{pmatrix} \tag{5}$$

$$p = \frac{R}{c_v}\left(E - \frac{1}{2}\rho\left(u_1^2 + u_2^2\right)\right) = \frac{R}{c_v}\left(w_3 - \frac{w_1^2 + w_2^2}{2 w_0}\right). \tag{6}$$

Advection-diffusion equation

$$\frac{\partial c}{\partial t} + \epsilon \Delta c + (\mathbf{u} \cdot \nabla) c = 0 \tag{7}$$

$$\frac{\partial c}{\partial t} + \epsilon \Delta c + \left(\frac{w_1}{w_0} \frac{\partial c}{\partial x} + \frac{w_2}{w_0} \frac{\partial c}{\partial y} \right) = 0. \tag{8}$$

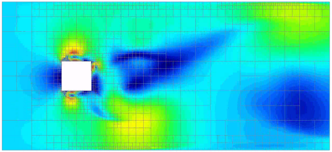
Optional variational multiscale stabilization (adding a stabilization form)

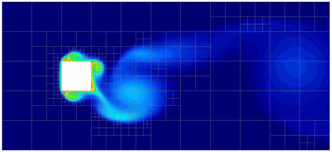
$$S(u, v) = \int_K \beta \left(-\frac{w_1}{w_0} \frac{\partial v}{\partial x} - \frac{w_2}{w_0} \frac{\partial v}{\partial y} + \epsilon \Delta v \right) * \left(-\frac{w_1}{w_0} \frac{\partial u}{\partial x} - \frac{w_2}{w_0} \frac{\partial u}{\partial y} + \epsilon \Delta u \right), \tag{9}$$

where $\beta = \frac{1}{\sqrt{9\left(\frac{4\epsilon}{|K|^2}\right)^2 + 4\left(\frac{|u|}{|K|}\right)^2}}$ and ϵ is diffusivity.

Discretization of time dependent problems

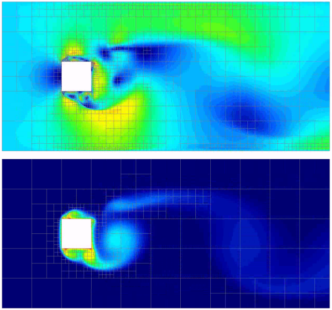
- Rothe's method
  - discretization in time first
  - discretized problem at each time level solved by adaptive hp-FEM
- dynamical meshes
  - meshes evolve in time
  - independent of each other





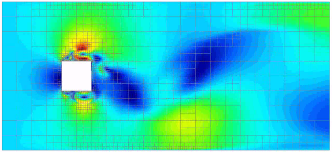
Discretization of time dependent problems

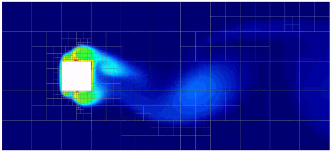
- Rothe's method
  - discretization in time first
  - discretized problem at each time level solved by adaptive hp-FEM
- dynamical meshes
  - meshes evolve in time
  - independent of each other



Discretization of time dependent problems

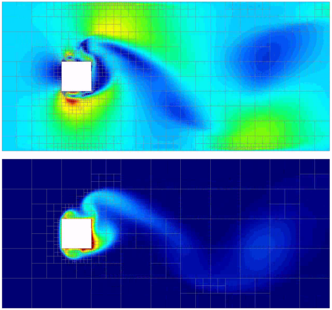
- Rothe's method
  - discretization in time first
  - discretized problem at each time level solved by adaptive hp-FEM
- dynamical meshes
  - meshes evolve in time
  - independent of each other





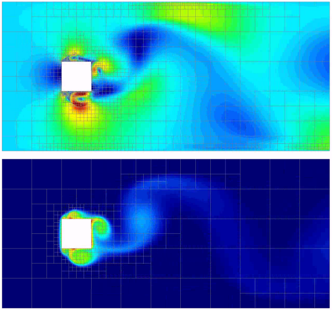
Discretization of time dependent problems

- Rothe's method
  - discretization in time first
  - discretized problem at each time level solved by adaptive hp-FEM
- dynamical meshes
  - meshes evolve in time
  - independent of each other



Discretization of time dependent problems

- Rothe's method
  - discretization in time first
  - discretized problem at each time level solved by adaptive hp-FEM
- dynamical meshes
  - meshes evolve in time
  - independent of each other



Discretization in time

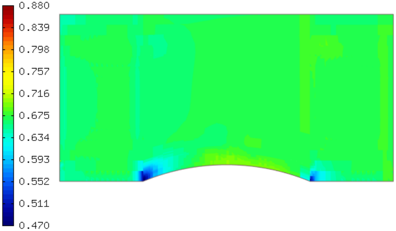
Semi-implicit discretization

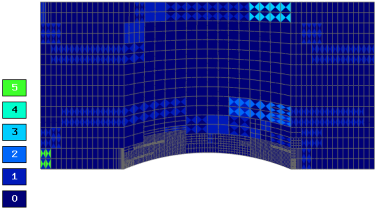
- linear terms discretized implicitly
- nonlinear terms explicitly
- possible increase of time step

Discretization in space

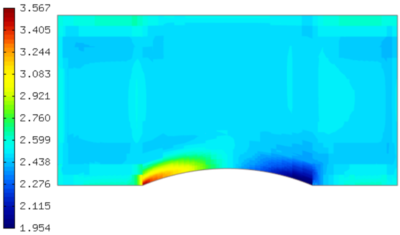
- adaptive hp-DG for the Euler equations
  - Steger-Warming numerical flux (easy linearization)
  - polynomial orders up to 10 (orders used in practice are lower)
  - Shock capturing
  - Flux limiting
- adaptive hp-FEM for the advection-diffusion equation
  - polynomial orders up to 10

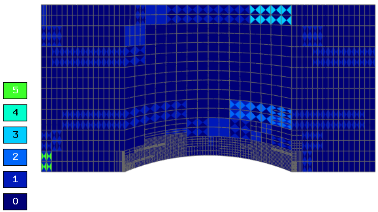
Evolution of Mach number, $\epsilon = 0.01$



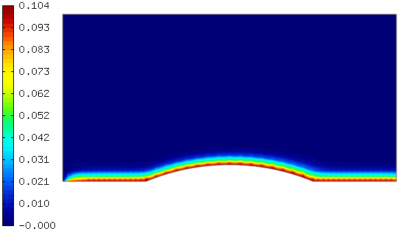


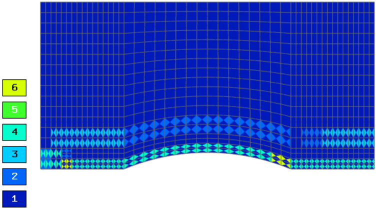
Evolution of pressure, $\epsilon = 0.01$





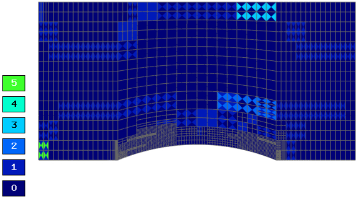
Evolution of Concentration, $\epsilon = 0.01$

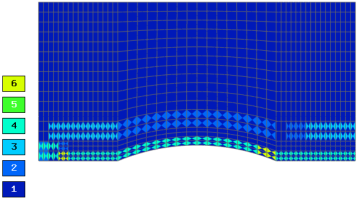




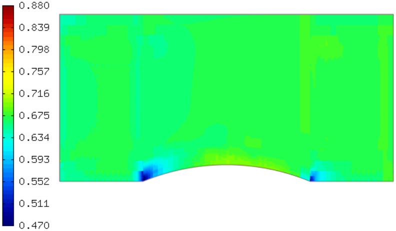
Comparison of meshes, $\epsilon = 0.01$

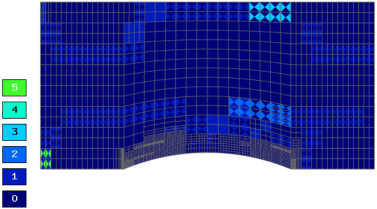
Mesh for the flow (top), mesh for the concentration (bottom)



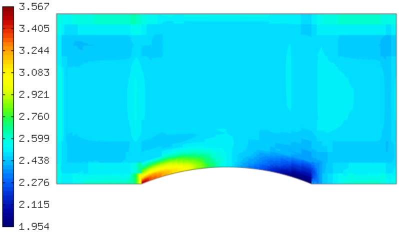


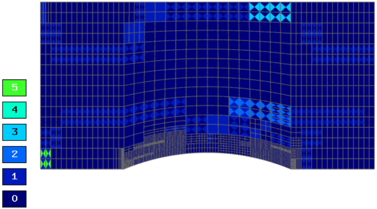
Evolution of Mach number, $\epsilon = 0.001$



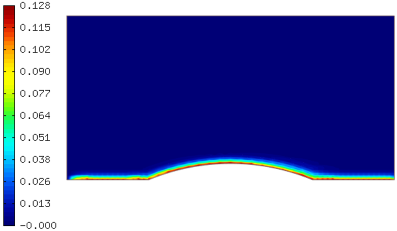


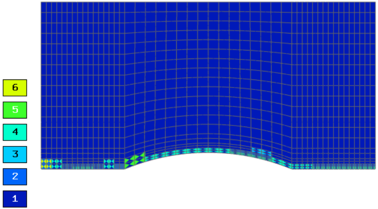
Evolution of pressure, $\epsilon = 0.001$





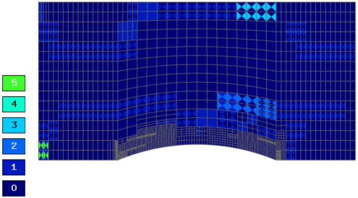
Evolution of Concentration, $\epsilon = 0.001$

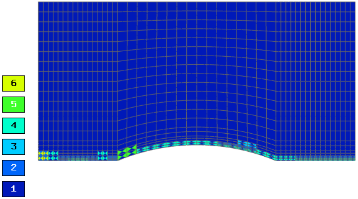




Comparison of meshes, $\epsilon = 0.001$

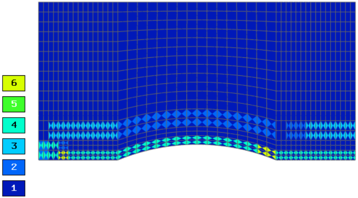
Mesh for the flow (top), mesh for the concentration (bottom)

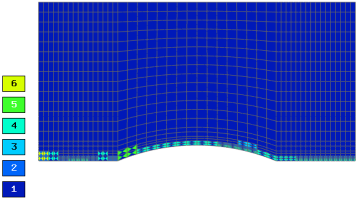




Comparison of meshes

Concentration meshes for $\epsilon = 0.01$ (top) and $\epsilon = 0.001$ (bottom)





Outlook

- \- Use the fully implicit scheme
- \- Use JFNK method (NOX package from the Trilinos library)
- \- Use the "full" Newton's method

Thank you for your attention.