Anisotropic h and p refinement for conforming FEM in 3D

Philipp Frauenfelder

Christian Lage

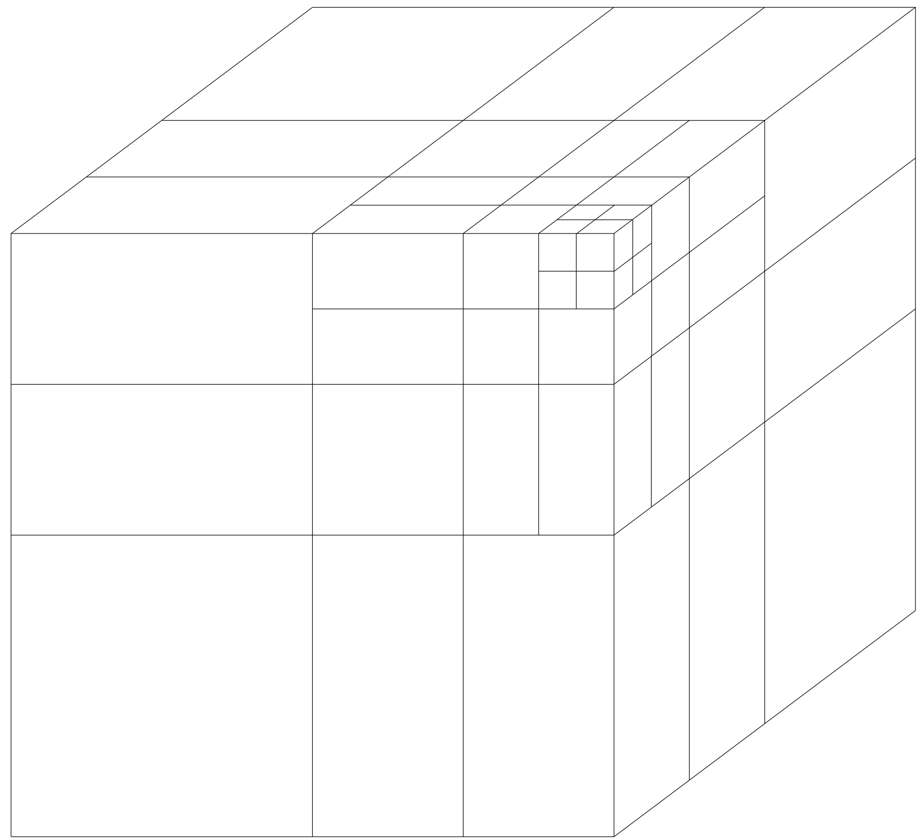
Christoph Schwab

pfrauenf@math.ethz.ch

Seminar for Applied Mathematics

Federal Institute of Technology, ETH Zürich

Goal



- Hierarchy of hanging nodes
- Anisotropic refinements

Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports

Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- Anisotropic p refinements

Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- Anisotropic p refinements
- hp Meshes

Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- Anisotropic p refinements
- hp Meshes
- Perspectives

Previous hp Software

- Szabó 1985: PROBE (p only)
- Demkowicz, Oden, Rachowicz et al. 1989: PHLEX, hp90
- Anderson: STRIPE (p only on a-priori generated meshes)
- Flaherty, Shephard: Tetrahedra only (3D anisotropy?)
- Karniadakis, Sherwin: NEKTAR (regular meshes only, tetrahedra, hexahedra, prisms)
- Devloo
- Szabó since 1995: STRESSCHECK
- Heuveline et al.: HiFlow

FE Method

- Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ (dimension independent design)
- Find $u \in V$ such that

$$a(u, v) = l(v) \quad \forall v \in V,$$

  V a FE space, $a(.\,,.)$ a bilinear form and $l(.)$ a linear form.
- Standard FE: $V \subset H^1(\Omega)$

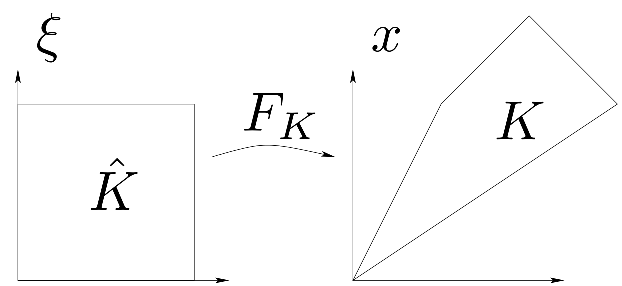
$$\begin{aligned} V &= S^{1,\underline{p}}(\Omega, \mathcal{T}) \\ &= \left\{ u \in H^1(\Omega) : u|_K \circ F_K \in \mathcal{Q}_p \; \forall K \in \mathcal{T} \right\} \end{aligned}$$

  $\Rightarrow u \in V$ is continuous, ie. $\mathcal{C}^0(\bar{\Omega})$.
- Vector valued problems are possible

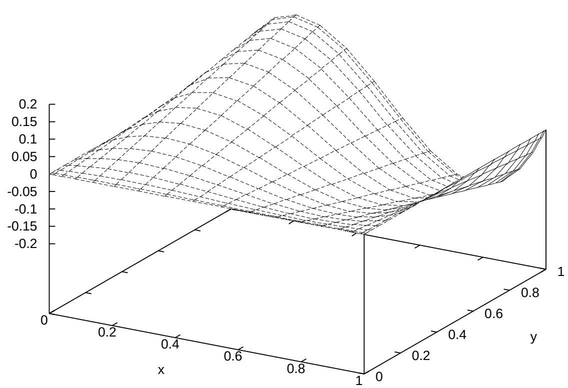
FE Space: Generalities

- Basis $\{\Phi_i\}_{i=1}^{N}$ constructed from element shape functions ϕ_j^K on elements $K \in \mathcal{T}$.
- Reference element shape functions: N_j, element map: $F_K : \hat{K} \to K$

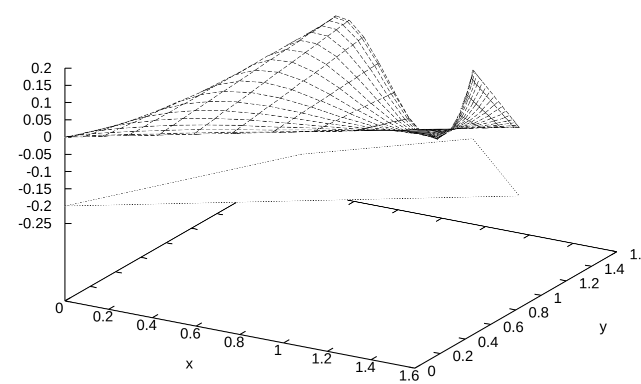
$$\Rightarrow \phi_j^K \circ F_K = N_j .$$



N_j

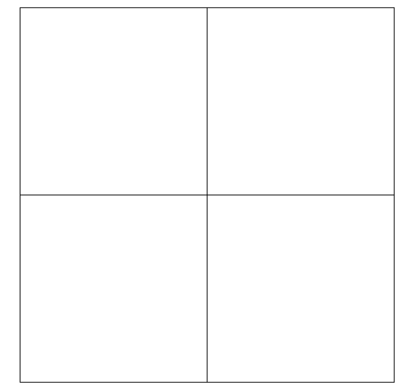


Φ_j



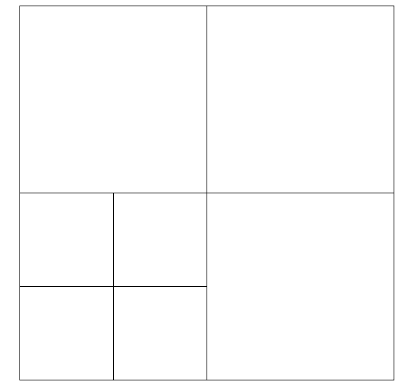
FE Meshes

Local refinements as mean to improve approximation of exact solution by FE solution



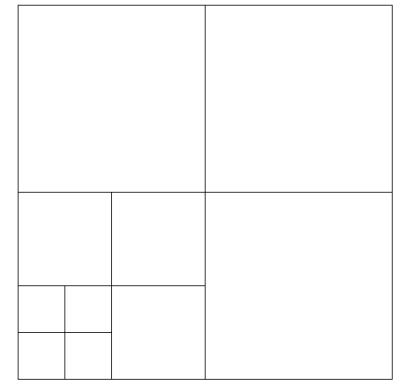
FE Meshes

Local refinements as mean to improve approximation of exact solution by FE solution



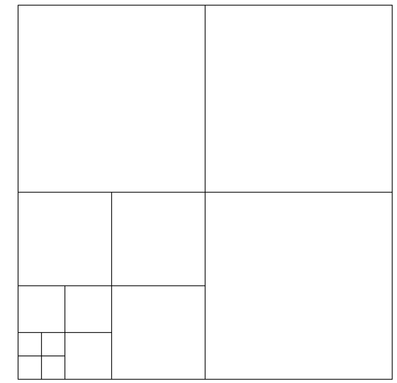
FE Meshes

Local refinements as mean to improve approximation of exact solution by FE solution

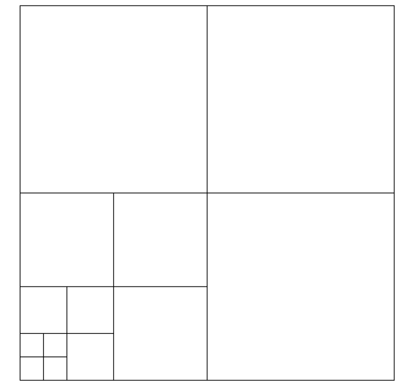


FE Meshes

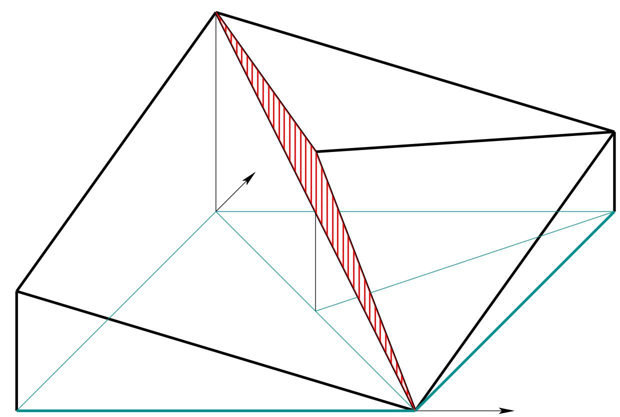
Local refinements as mean to improve approximation of exact solution by FE solution



FE Meshes

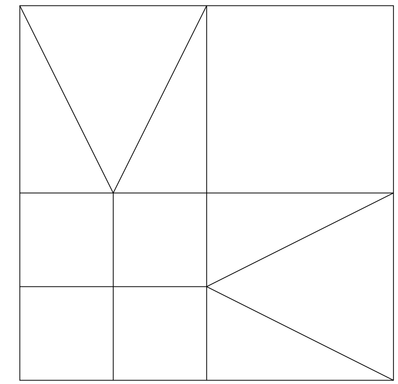


Local refinements as mean to improve approximation of exact solution by FE solution



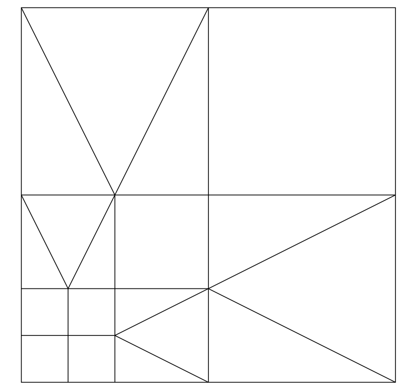
But standard FE forbids locally refined grids: discontinuities are possible.

Mortar vs. Enforcing Continuity



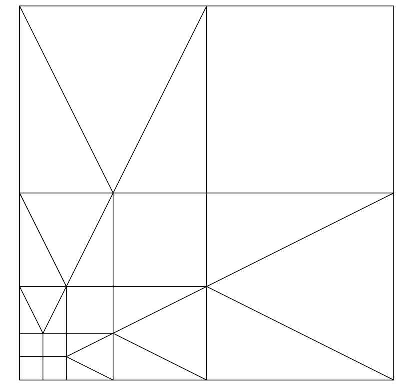
- Topolocigal closure

Mortar vs. Enforcing Continuity

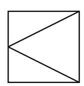


- Topolocigal closure

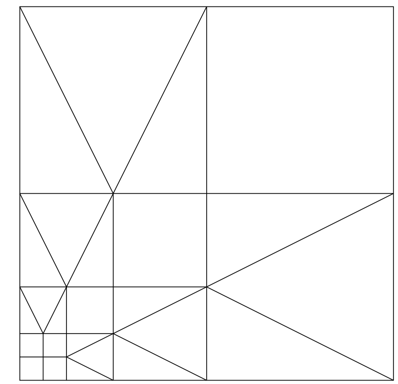
Mortar vs. Enforcing Continuity



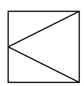
- Topolocigal closure

  Drawbacks: more elements, more element types, what about refining a  ?

Mortar vs. Enforcing Continuity



- Topolocigal closure

  Drawbacks: more elements, more element types, what about refining a  ?

- Our philosophy: hexahedral meshes only (tensorized interpolants, spectral quadrature techniques)

- Our solution: Treating the constraints induced by the hanging nodes
  Why conforming? $a(u, v) = a(v, u)$ and $a(u, u) \geq \alpha \|u\|_V^2 \Rightarrow \boldsymbol{A}$ SPD, pccg . . .

Our Software: Concepts

- Started by Christian Lage during his Ph.D. studies (1995).
- Used and improved by Frauenfelder, Matache, Schmidlin, Schmidt and several students.
- Concept Oriented Design using mathematical principles.
- Currently two parts: hp-FEM, BEM (wavelet and multipole methods).
- C++

Overview

- Introduction
- Anisotropic h refinements
  - **S and T matrices**
  - Assembly of Supports
- Anisotropic p refinements
- hp Meshes
- Perspectives

T Matrix

Definition 1 (T Matrix). *Element shape functions* $\{\phi_j^K\}_{j=1}^{m_K}$ *on element* K, *global basis functions* $\{\Phi_i\}_{i=1}^{N}$.
The T matrix $\boldsymbol{T}_K \in \mathbb{R}^{m_K \times N}$ *of element* K *is implicitly defined by*

$$\Phi_i|_K = \sum_{j=1}^{m_K} [\boldsymbol{T}_K]_{ji}\, \phi_j^K$$

as vectors:

$$\underline{\Phi}|_K = \boldsymbol{T}_K^\top \underline{\phi}^K.$$

Assembly using T Matrices

Assembling:

$$\underline{l} = l(\underline{\Phi}) = l\Big(\sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{\phi}^{\tilde{K}}\Big) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} l(\underline{\phi}^{\tilde{K}}) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{l}_{\tilde{K}}$$

Assembly using T Matrices

Assembling:

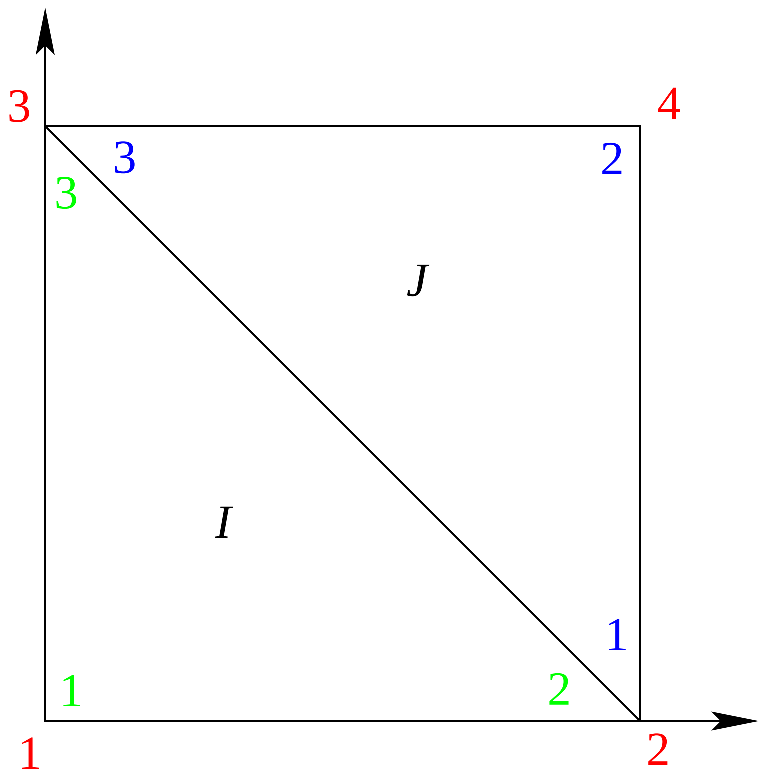
$$\underline{l} = l(\underline{\Phi}) = l\Big(\sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{\phi}^{\tilde{K}}\Big) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} l(\underline{\phi}^{\tilde{K}}) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{l}_{\tilde{K}}$$

$$\boldsymbol{A} = a(\underline{\Phi}, \underline{\Phi}) = \sum_{K,\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} a(\underline{\phi}^{K}, \underline{\phi}^{\tilde{K}}) \boldsymbol{T}_K = \sum_{K,\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \boldsymbol{A}_{\tilde{K}K} \boldsymbol{T}_K$$

Note: $\boldsymbol{A}_{\tilde{K}K} = 0$ in standard FEM for $\tilde{K} \neq K$.

Example: Regular Mesh

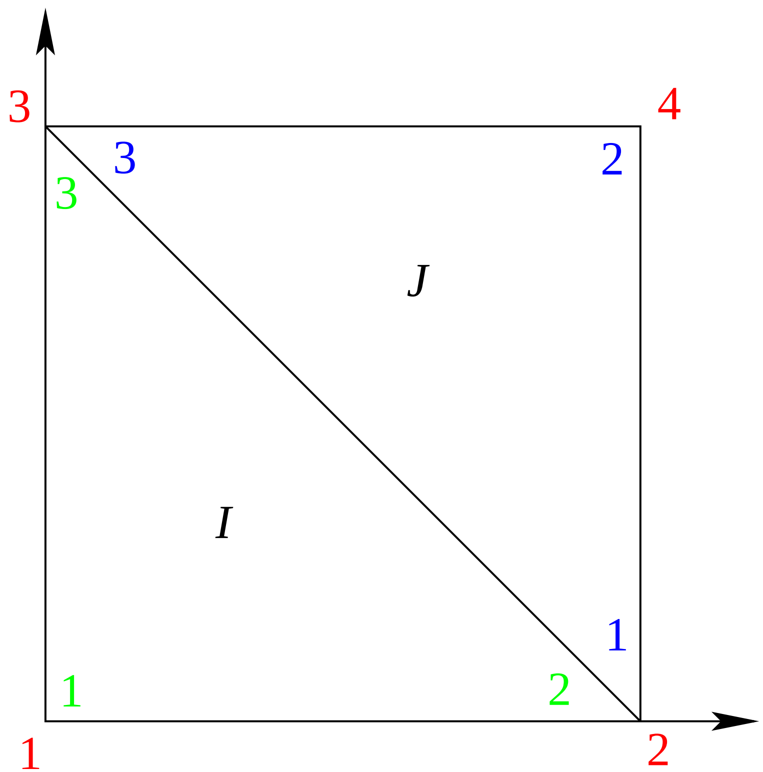
Two elements with three local shape functions each and four global basis functions.



$$T_I = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Example: Regular Mesh

Two elements with three local shape functions each and four global basis functions.

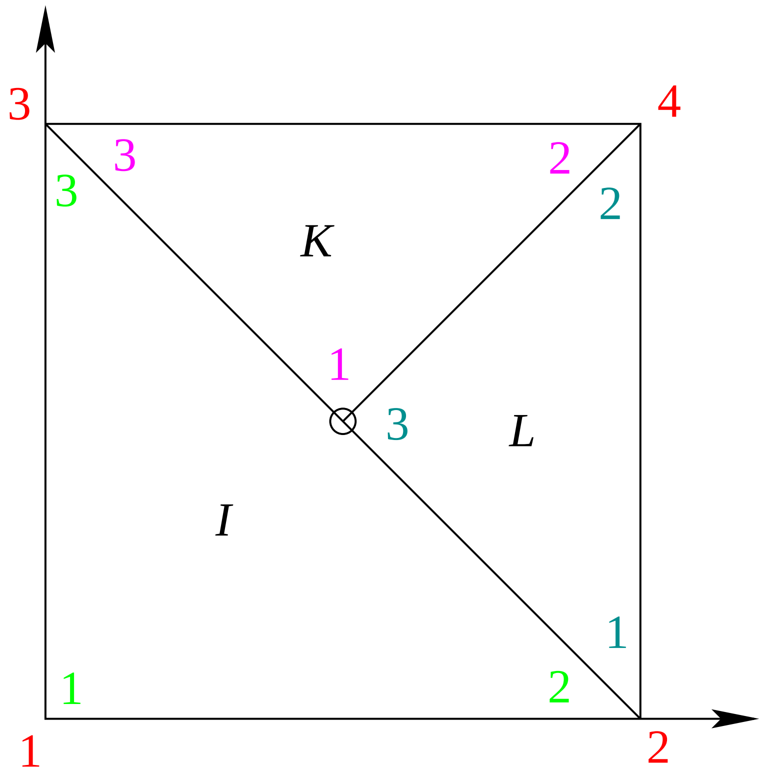


$$\boldsymbol{T}_I = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\boldsymbol{T}_J = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Example: Irregular Mesh

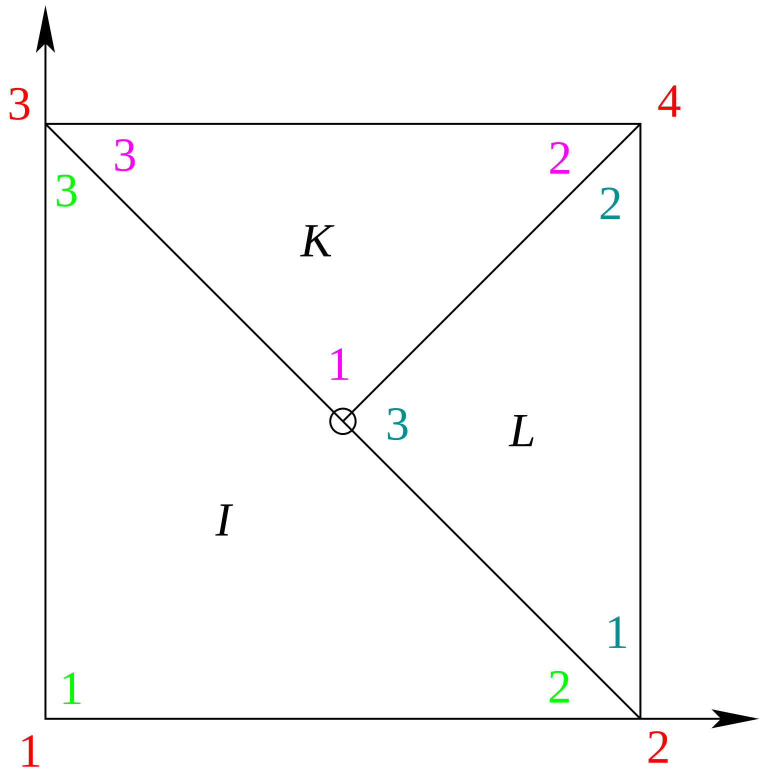
Three elements with three local shape functions each and four global basis functions. The hanging node is marked with $\circ$.



$$\boldsymbol{T}_L = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

Example: Irregular Mesh

Three elements with three local shape functions each and four global basis functions. The hanging node is marked with $\circ$.



$$
\boldsymbol{T}_L = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}
$$

$$
\boldsymbol{T}_K = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1/2 & 1/2 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 & 0 \end{pmatrix}
$$

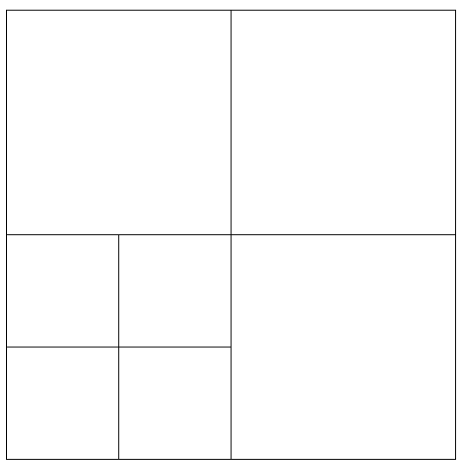
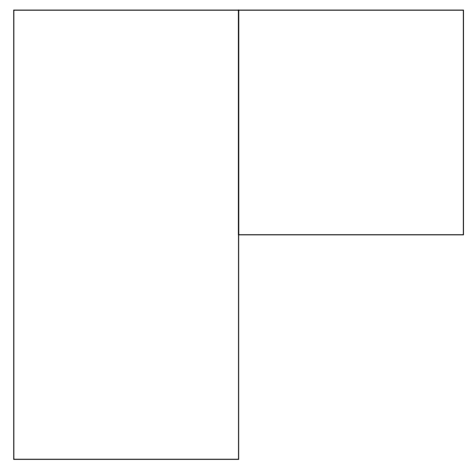
$\Rightarrow$ continuous basis functions.

Generation of T Matrices

- **Regular Mesh:** Counting and assigning indices with respect to topological entities such as vertices, edges and faces. Explained in detail later.

Generation of T Matrices

- **Regular Mesh:** Counting and assigning indices with respect to topological entities such as vertices, edges and faces. Explained in detail later.
- **Irregular Mesh:** Irregularity due to a refinement of an initially regular mesh.

  OK: , not OK: 

  Explanation follows.

T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

Mesh	$\mathcal{M}$	refine	$\mathcal{M}'$
Basis fcts.	$B = B_{\mathrm{repl}} \cup B_{\mathrm{keep}}$	$\longrightarrow$	$B' = B_{\mathrm{ins}} \cup B_{\mathrm{keep}}$

T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

Mesh	$\mathcal{M}$	refine	$\mathcal{M}'$
Basis fcts.	$B = B_{\mathrm{repl}} \cup B_{\mathrm{keep}}$	$\longrightarrow$	$B' = B_{\mathrm{ins}} \cup B_{\mathrm{keep}}$

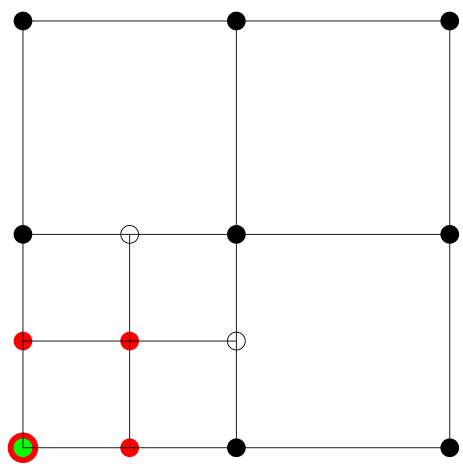
B_{repl}: basis fcts. which can be solely described by elements of $\mathcal{M}' \backslash \mathcal{M}$

B_{ins}: basis fcts. generated by regular parts of $\mathcal{M}' \backslash \mathcal{M}$

T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

Mesh	$\mathcal{M}$	refine	$\mathcal{M}'$
Basis fcts.	$B = B_{\text{repl}} \cup B_{\text{keep}}$	$\longrightarrow$	$B' = B_{\text{ins}} \cup B_{\text{keep}}$



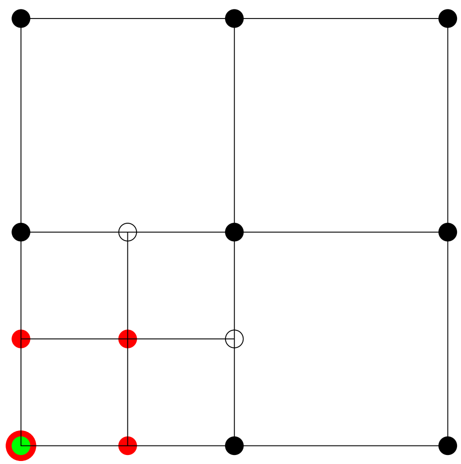
B_{repl}: basis fcts. which can be solely described by elements of $\mathcal{M}' \backslash \mathcal{M}$

B_{ins}: basis fcts. generated by regular parts of $\mathcal{M}' \backslash \mathcal{M}$

T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

Mesh	$\mathcal{M}$	refine	$\mathcal{M}'$
Basis fcts.	$B = B_{\text{repl}} \cup B_{\text{keep}}$	$\longrightarrow$	$B' = B_{\text{ins}} \cup B_{\text{keep}}$



B_{repl}: basis fcts. which can be solely described by elements of $\mathcal{M}' \setminus \mathcal{M}$

B_{ins}: basis fcts. generated by regular parts of $\mathcal{M}' \setminus \mathcal{M}$

Every element of B has a column in the T matrix. Generation is

- easy for B_{ins} (like regular mesh),
- simple for B_{keep}: modify column from $\mathcal{M}$ by S matrix.

S Matrix

Definition 2 (S Matrix). *Let $K' \subset K$ be the result of a refinement of element K. The S matrix $\boldsymbol{S}_{K'K} \in \mathbb{R}^{m_{K'} \times m_K}$ is defined by*

$$\phi_j^K\Big|_{K'} = \sum_{l=1}^{m_{K'}} [\boldsymbol{S}_{K'K}]_{lj}\, \phi_l^{K'}$$

as vectors:

$$\underline{\phi}^K\Big|_{K'} = \boldsymbol{S}_{K'K}^\top \underline{\phi}^{K'}$$

$\phi_j^K\big|_{K'}$ is represented as a linear combination of the shape functions $\left\{\phi_l^{K'}\right\}_{l=1}^{m_{K'}}$ of K'.

Application of S Matrix

Proposition 1. *Let $K' \subset K$ be the result of a refinement of an element K. Then, the T matrix of K' can be computed as*

$$\boldsymbol{T}_{K'} = \boldsymbol{S}_{K'K} \boldsymbol{T}_K^{\mathrm{keep}} + \boldsymbol{T}_{K'}^{\mathrm{ins}}$$

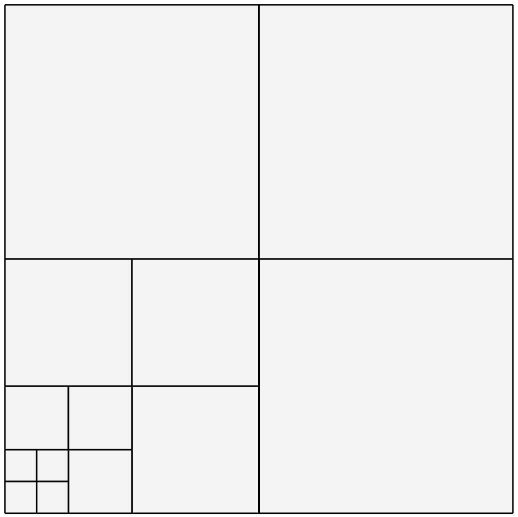
where $\boldsymbol{T}_K^{\mathrm{keep}}$ denotes the T matrix of element K (with columns not related to functions in B_{keep} set to zero) and $\boldsymbol{T}_{K'}^{\mathrm{ins}}$ the T matrix for functions in B_{ins} with respect to K'.

Proposition 2. *Let $\hat{K}' \subset \hat{K}$ be the result of a refinement of the reference element $\hat{K}$ with $H : \hat{K} \to \hat{K}'$ the subdivision map. The element maps are*

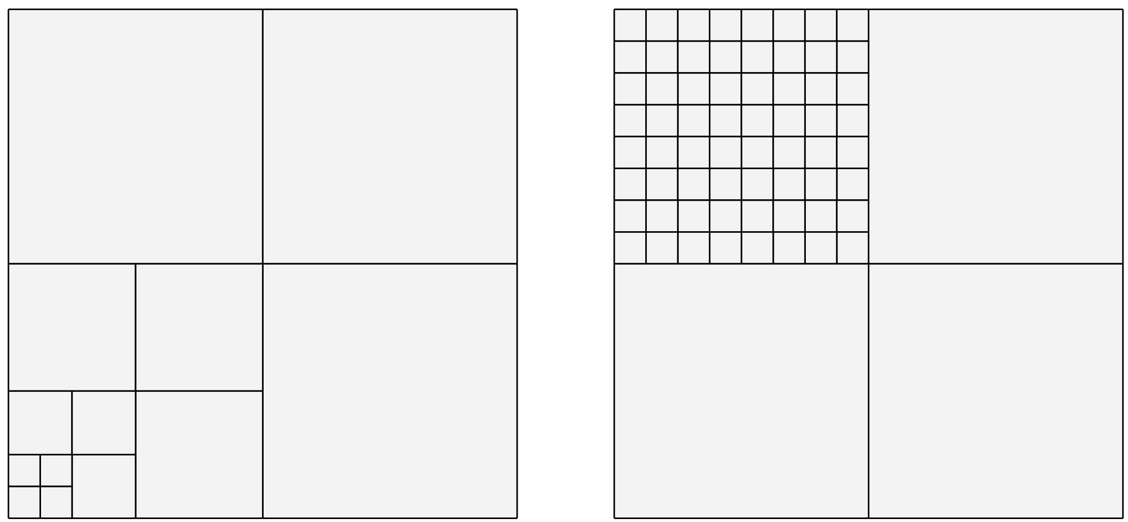
$$F_K : \hat{K} \to K \text{ and } F_{K'} : \hat{K} \to K'$$

and $F_{K'} \circ H^{-1} = F_K$ holds. Then, $\boldsymbol{S}_{\hat{K}'\hat{K}} = \boldsymbol{S}_{K'K}$.

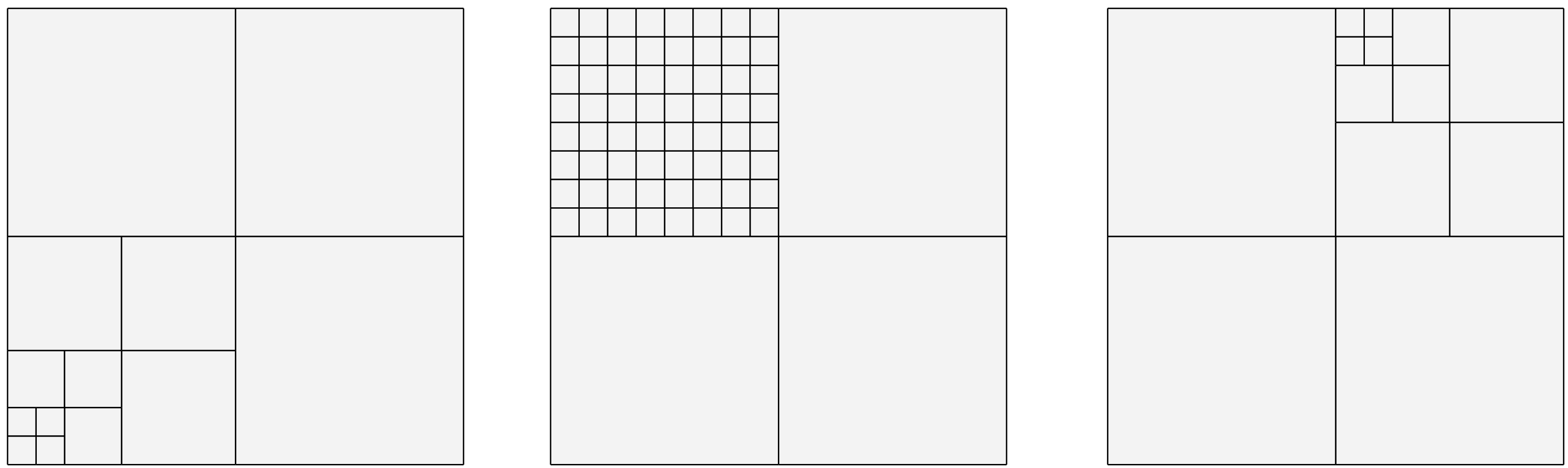
Meshes



Meshes



Meshes



S Matrix in Dimension $d = 1$

Subdividing $\hat{J} = (0,1)$ in $\hat{J}' = (0, 1/2)$ and $\hat{J}^\star = (1/2, 1)$ with the reference element shape functions

$$N_j(\xi) = \begin{cases} 1-\xi & j=1 \\ \xi & j=2 \\ \xi(1-\xi)P^{1,1}_{j-3}(2\xi-1) & j=3,\ldots,J \end{cases}$$

yields (solving a linear system) for $J = 4$:

$$\boldsymbol{S}_{\hat{J}',\hat{J}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/4 & 0 \\ 0 & 0 & 1/4 & -3/4 \\ 0 & 0 & 0 & 1/8 \end{pmatrix} \text{ and } \boldsymbol{S}_{\hat{J}^\star,\hat{J}} = \begin{pmatrix} 1/2 & 1/2 & 1/4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/4 & 3/4 \\ 0 & 0 & 0 & 1/8 \end{pmatrix}.$$

Hierarchic shape functions $\Rightarrow$ hierarchic S matrices.

S Matrices: Tensor Product in 2D

- $d > 1$ with hexahedral meshes $\Rightarrow$ S matrices are built from tensor products of 1D S matrices.

S Matrices: Tensor Product in 2D

- $d > 1$ with hexahedral meshes $\Rightarrow$ S matrices are built from tensor products of 1D S matrices.
- In 2D: $N_{i,j} = N_i \otimes N_j$, the four bilinear shape functions are:

$$N_{1,2}(\underline{\xi}) = N_1(\xi_1) \cdot N_2(\xi_2) \qquad N_{2,2}(\underline{\xi}) = N_2(\xi_1) \cdot N_2(\xi_2)$$

$$N_{1,1}(\underline{\xi}) = N_1(\xi_1) \cdot N_1(\xi_2) \qquad N_{2,1}(\underline{\xi}) = N_2(\xi_1) \cdot N_1(\xi_2)$$

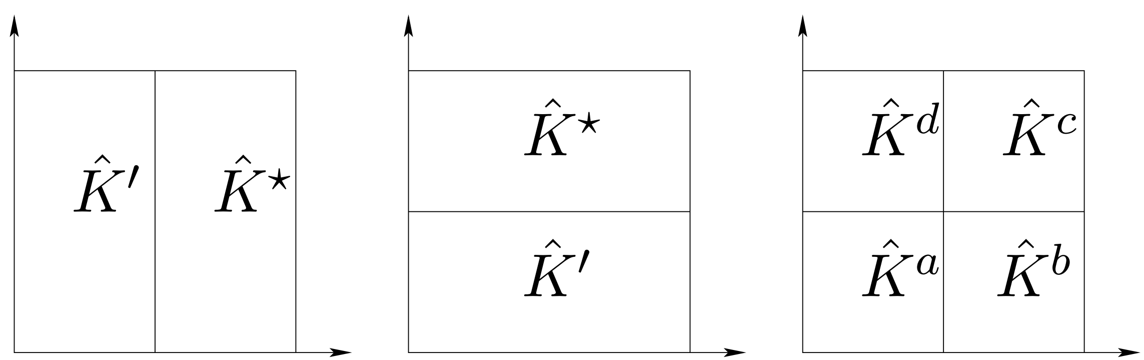
S Matrices: Tensor Product in 2D

- $d > 1$ with hexahedral meshes $\Rightarrow$ S matrices are built from tensor products of 1D S matrices.
- In 2D: $N_{i,j} = N_i \otimes N_j$, the four bilinear shape functions are:

$$N_{1,2}(\underline{\xi}) = N_1(\xi_1) \cdot N_2(\xi_2) \qquad N_{2,2}(\underline{\xi}) = N_2(\xi_1) \cdot N_2(\xi_2)$$

$$N_{1,1}(\underline{\xi}) = N_1(\xi_1) \cdot N_1(\xi_2) \qquad N_{2,1}(\underline{\xi}) = N_2(\xi_1) \cdot N_1(\xi_2)$$

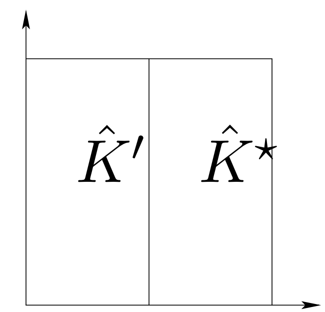
- Consider the subdivisions:



S Matrices: Tensor Product in 2D II

Subdivision map of left variant: $H : \hat{K} \to \hat{K}', \underline{\xi} \mapsto \begin{pmatrix} \xi_1/2 \\ \xi_2 \end{pmatrix}$. S matrix $\boldsymbol{S}_{\hat{K}'\hat{K}}$ is defined by:

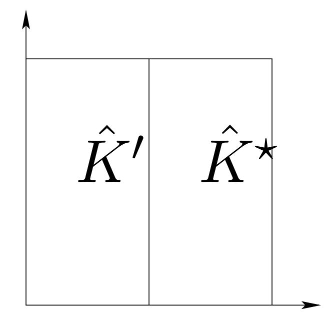
$$N_{i,j}|_{\hat{K}'} = \sum_{k,l} \left[\boldsymbol{S}_{\hat{K}'\hat{K}}\right]_{(k,l),(i,j)} N_{k,l} \circ H^{-1}.$$



S Matrices: Tensor Product in 2D II

Subdivision map of left variant: $H : \hat{K} \to \hat{K}', \underline{\xi} \mapsto \begin{pmatrix} \xi_1/2 \\ \xi_2 \end{pmatrix}$. S matrix $\boldsymbol{S}_{\hat{K}'\hat{K}}$ is defined by:

$$N_{i,j}|_{\hat{K}'} = \sum_{k,l} \left[\boldsymbol{S}_{\hat{K}'\hat{K}}\right]_{(k,l),(i,j)} N_{k,l} \circ H^{-1}.$$



Tensor product shape functions:

$$(N_i \otimes N_j)|_{\hat{K}'} = \sum_{k,l} \left[\boldsymbol{S}_{\hat{K}'\hat{K}}\right]_{(k,l),(i,j)} (N_k \otimes N_l) \circ H^{-1}. \quad (1)$$

S Matrices: Tensor Product in 2D III

S matrices for 1D reference element shape fcts. used in (1):

$$N_i|_{\hat{J}'} = \sum_m [\boldsymbol{S}_{\hat{J}',\hat{J}}]_{mi} N_m \circ G^{-1}$$ for the ξ_1 part and

$$N_j = \sum_n [\boldsymbol{E}]_{nj} N_n$$ for the ξ_2 part,

where $G : \xi \mapsto \xi/2$.

S Matrices: Tensor Product in 2D III

S matrices for 1D reference element shape fcts. used in (1):

$$N_i|_{\hat{J}'} = \sum_m \left[\boldsymbol{S}_{\hat{J}',\hat{J}}\right]_{mi} N_m \circ G^{-1}$$ for the ξ_1 part and

$$N_j = \sum_n [\boldsymbol{E}]_{nj} N_n$$ for the ξ_2 part,

where $G : \xi \mapsto \xi/2$. Plugging into the left hand side of (1) yields:

$$\begin{aligned}(N_i \otimes N_j)|_{\hat{K}'} = N_i|_{\hat{J}'} \otimes N_j &= \sum_{m,n} \left(\left[\boldsymbol{S}_{\hat{J}',\hat{J}}\right]_{mi} N_m \circ G^{-1}\right) \otimes \Big([\boldsymbol{E}]_{nj} N_n\Big) \\ &= \sum_{m,n} \left[\boldsymbol{S}_{\hat{J}',\hat{J}}\right]_{mi} \cdot [\boldsymbol{E}]_{nj} N_m \circ G^{-1} \otimes N_n.\end{aligned}$$

S Matrices: Tensor Product in 2D IV

Comparing with the right hand side of (1):

$$\sum_{m,n} \left[\boldsymbol{S}_{\hat{J}',\hat{J}}\right]_{mi} \cdot [\boldsymbol{E}]_{nj}\, N_m \circ G^{-1} \otimes N_n$$

$$= \sum_{k,l} \left[\boldsymbol{S}_{\hat{K}'\hat{K}}\right]_{(k,l),(i,j)} N_k \circ G^{-1} \otimes N_l .$$

S Matrices: Tensor Product in 2D IV

Comparing with the right hand side of (1):

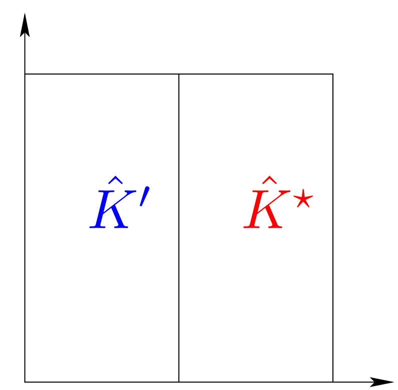
$$\sum_{m,n} \left[\boldsymbol{S}_{\hat{J}'\hat{J}}\right]_{mi} \cdot [\boldsymbol{E}]_{nj}\, N_m \circ G^{-1} \otimes N_n$$

$$= \sum_{k,l} \left[\boldsymbol{S}_{\hat{K}'\hat{K}}\right]_{(k,l),(i,j)} N_k \circ G^{-1} \otimes N_l.$$

Therefore for the vertical subdivision:

$\boldsymbol{S}_{\hat{K}'\hat{K}} = \boldsymbol{S}_{\hat{J}'\hat{J}} \otimes \boldsymbol{E}$ for the left quad $\hat{K}'$,

$\boldsymbol{S}_{\hat{K}^\star\hat{K}} = \boldsymbol{S}_{\hat{J}^\star\hat{J}} \otimes \boldsymbol{E}$ for the right quad $\hat{K}^\star$.

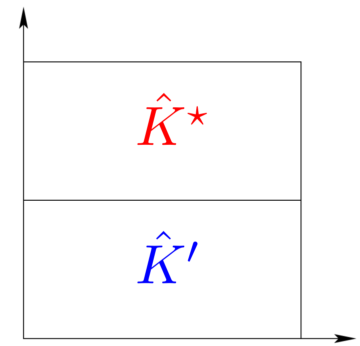


S Matrices: Tensor Product in 2D V

Horizontal subdivision:

$\boldsymbol{S}_{\hat{K}'\hat{K}} = \boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}'\hat{J}}$ for the bottom quad $\hat{K}'$,

$\boldsymbol{S}_{\hat{K}^\star\hat{K}} = \boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star\hat{J}}$ for the top quad $\hat{K}^\star$.

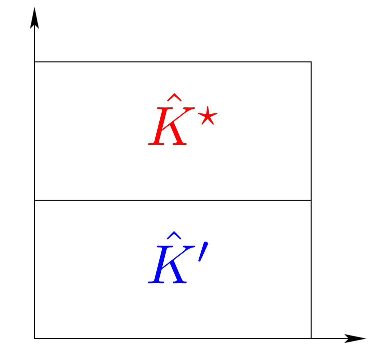


S Matrices: Tensor Product in 2D V

Horizontal subdivision:

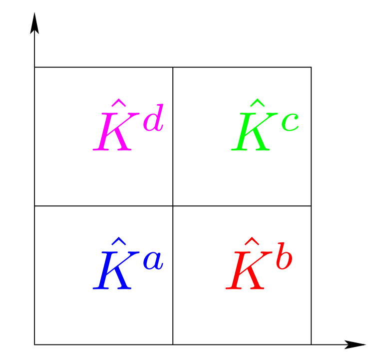
$$S_{\hat{K}'\hat{K}} = E \otimes S_{\hat{J}'\hat{J}} \quad \text{for the bottom quad } \hat{K}',$$

$$S_{\hat{K}^\star\hat{K}} = E \otimes S_{\hat{J}^\star\hat{J}} \quad \text{for the top quad } \hat{K}^\star.$$



Subdivision into four quads:

- subdivide $\hat{K}$ horizontally into two children
- subdivide upper and lower child vertically into $\hat{K}^d$ and $\hat{K}^c$ and $\hat{K}^a$ and $\hat{K}^b$ resp.



$$S_{\hat{K}^d\hat{K}} = \left(S_{\hat{J}'\hat{J}} \otimes E\right) \cdot \left(E \otimes S_{\hat{J}^\star\hat{J}}\right) \quad S_{\hat{K}^c\hat{K}} = \left(S_{\hat{J}^\star\hat{J}} \otimes E\right) \cdot \left(E \otimes S_{\hat{J}^\star\hat{J}}\right)$$

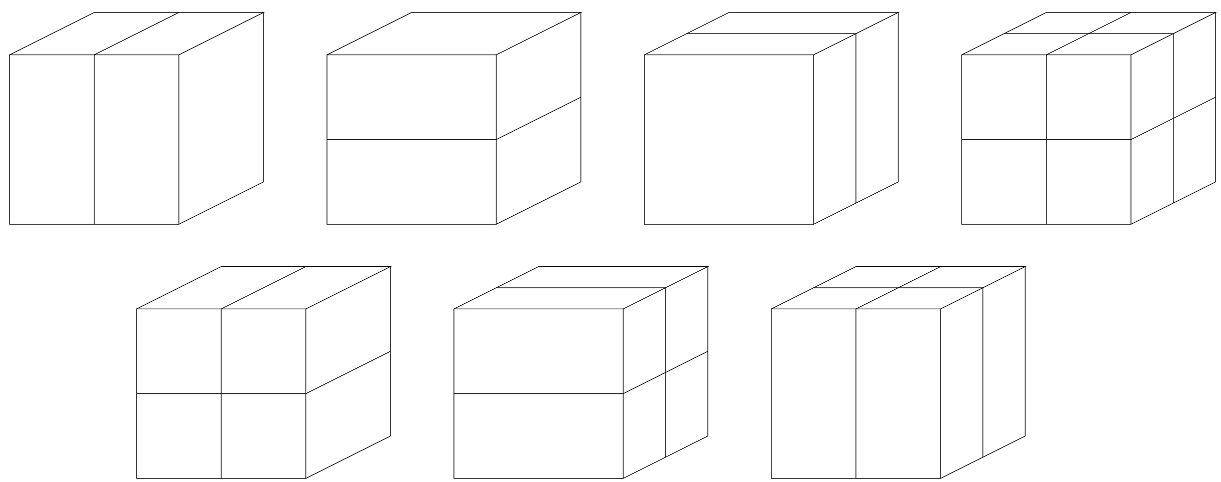
$$S_{\hat{K}^a\hat{K}} = \left(S_{\hat{J}'\hat{J}} \otimes E\right) \cdot \left(E \otimes S_{\hat{J}'\hat{J}}\right) \quad S_{\hat{K}^b\hat{K}} = \left(S_{\hat{J}^\star\hat{J}} \otimes E\right) \cdot \left(E \otimes S_{\hat{J}'\hat{J}}\right)$$

S Matrices: Tensor-Product in 3D

Same idea as in 2D, just of this form:

$$\boldsymbol{S}_{\hat{K}'\hat{K}} = \prod \left(\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C} \right)$$

in each of the factors, one of $\boldsymbol{A}$, $\boldsymbol{B}$ or $\boldsymbol{C}$ is an 1D S matrix.
Depending on the factors, 7 subdivisions are possible:



Concepts: allow arbitrary number and combination of these 7 subdivisions

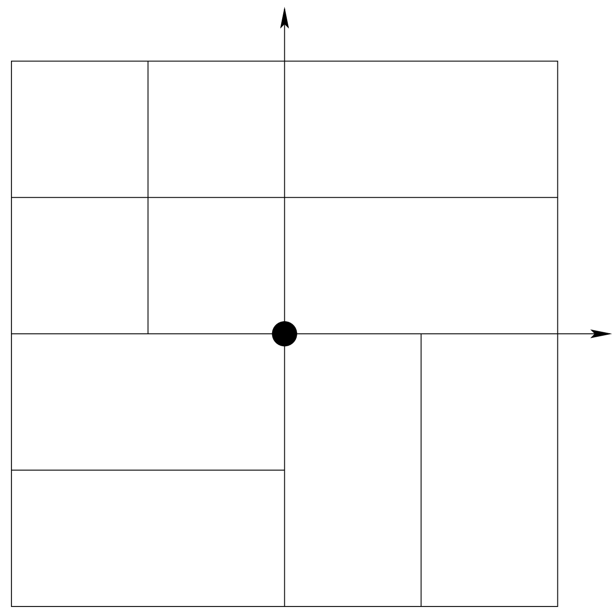
in 3D.

Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- Anisotropic p refinements
- hp Meshes
- Perspectives

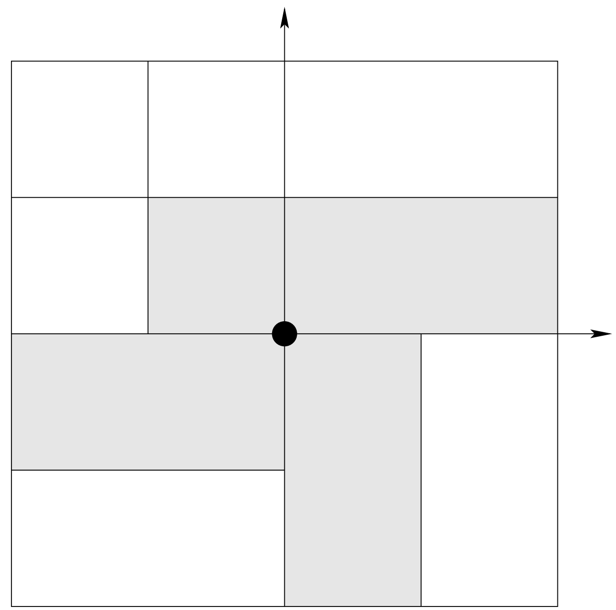
Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.



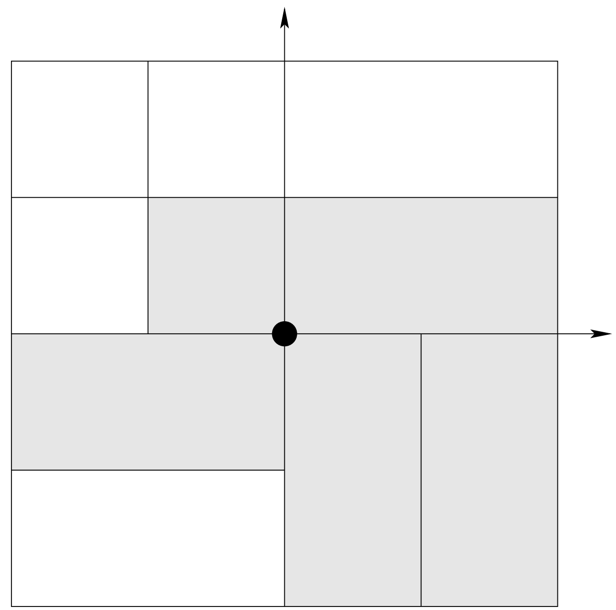
Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.



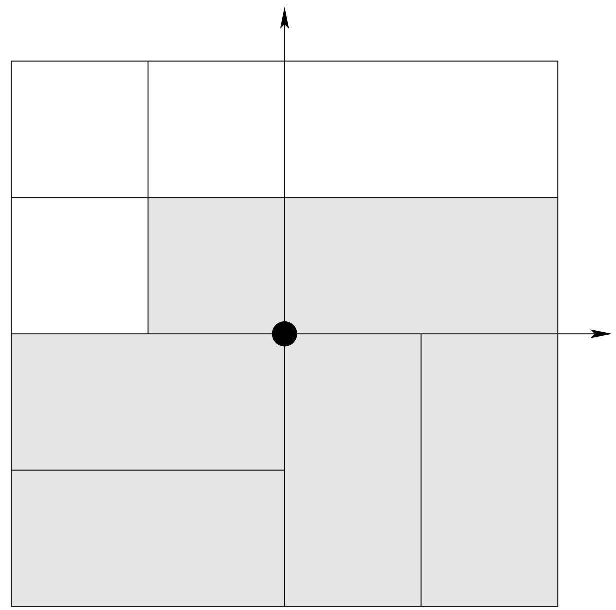
Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.



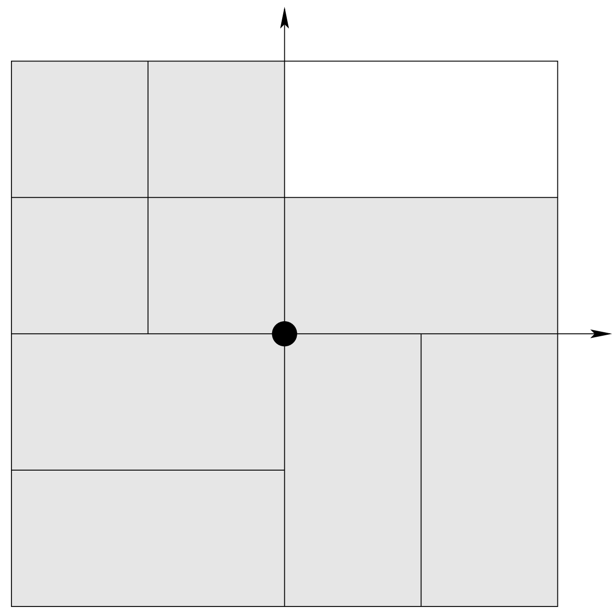
Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.



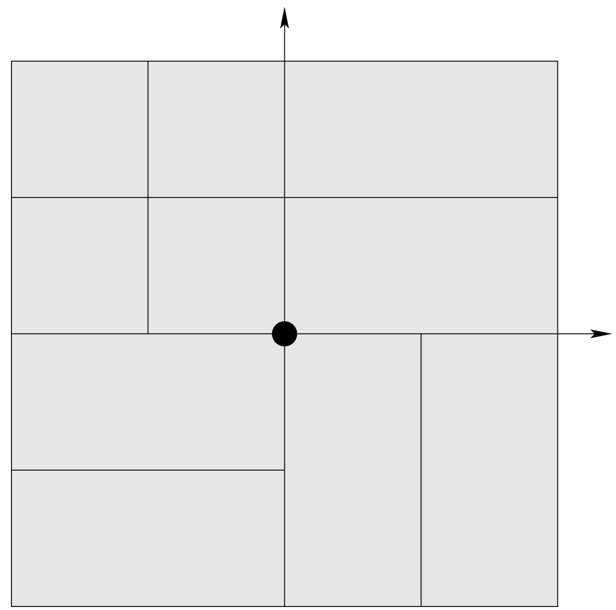
Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.

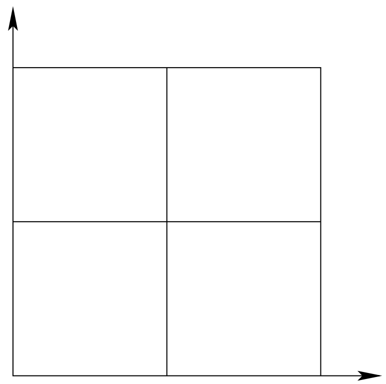


Anisotropic and Conforming

Main point: find the cells (either coarse or fine) which belong to the support of a certain basis function.

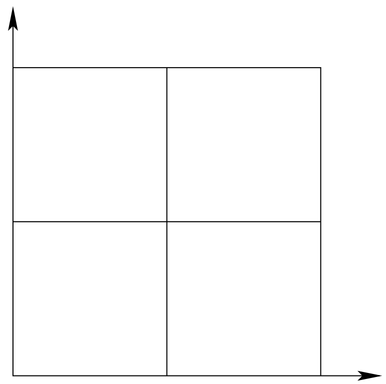


Anisotropic and Conforming II



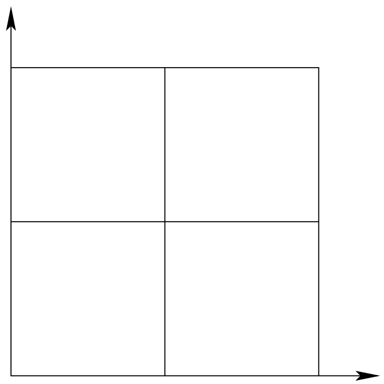
- Can easily be treated since all edges are broken

Anisotropic and Conforming II

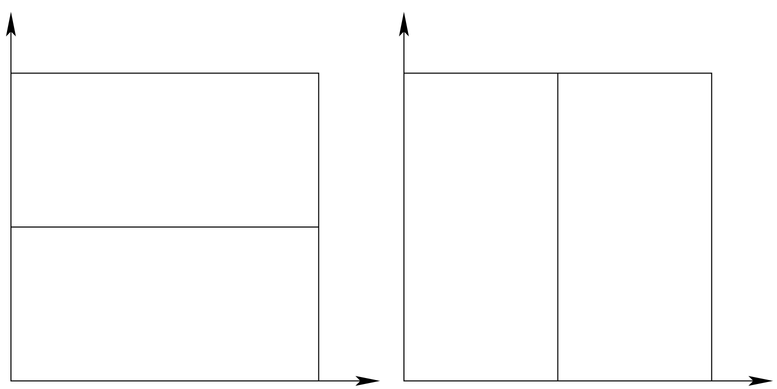


- Can easily be treated since all edges are broken
- “Level of refinement” on each cell is enough to handle hanging nodes

Anisotropic and Conforming II

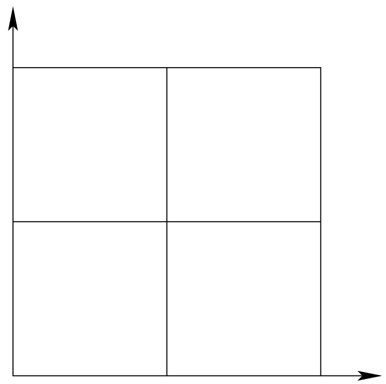


- Can easily be treated since all edges are broken
- “Level of refinement” on each cell is enough to handle hanging nodes

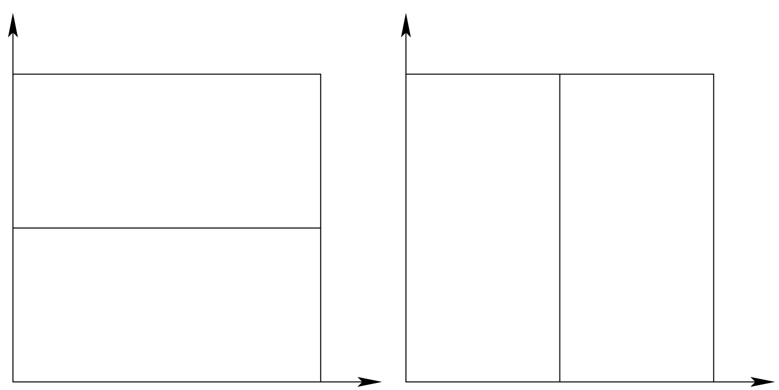


- More complicated as not all edges are broken

Anisotropic and Conforming II

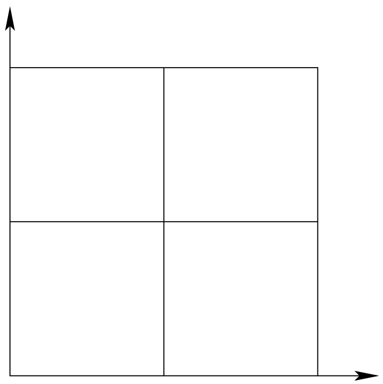


- Can easily be treated since all edges are broken
- “Level of refinement” on each cell is enough to handle hanging nodes

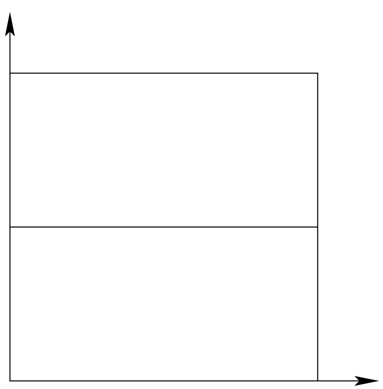


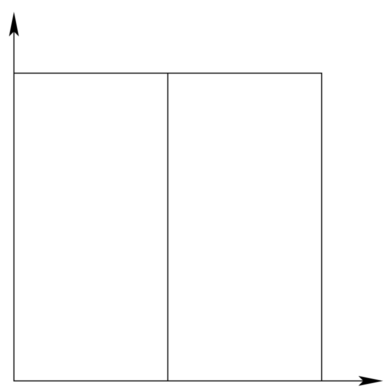
- More complicated as not all edges are broken
- “Level of refinement” (also a vector valued level) is not enough

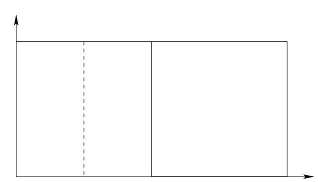
Anisotropic and Conforming II



- Can easily be treated since all edges are broken
- “Level of refinement” on each cell is enough to handle hanging nodes





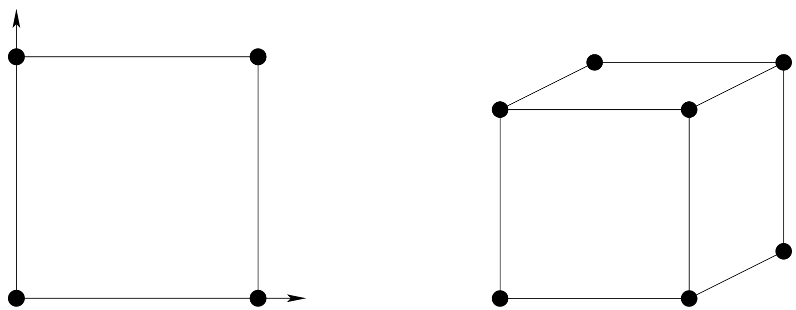
- More complicated as not all edges are broken
- “Level of refinement” (also a vector valued level) is not enough
-  should be seen as conforming

Condition for Continuity

- In order to have continuous global basis functions Φ_i, the unisolvent sets on the interfaces in the support of Φ_i have to match.

Condition for Continuity

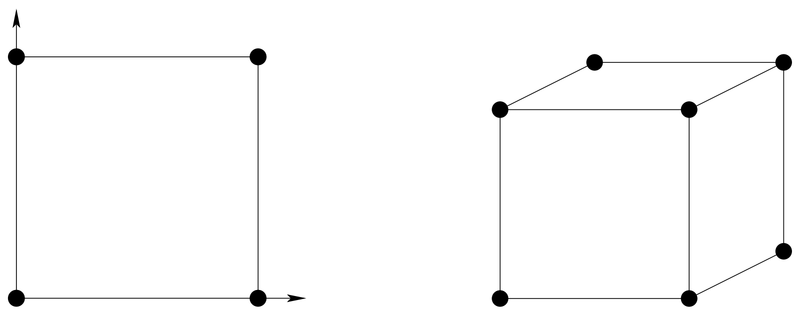
- In order to have continuous global basis functions Φ_i, the unisolvent sets on the interfaces in the support of Φ_i have to match.
- Unisolvent set for $\mathcal{Q}_1$ in a quad / hex are the corners:



$\Rightarrow$ matching edges which coincide in a vertex is sufficient

Condition for Continuity

- In order to have continuous global basis functions Φ_i, the unisolvent sets on the interfaces in the support of Φ_i have to match.
- Unisolvent set for $\mathcal{Q}_1$ in a quad / hex are the corners:



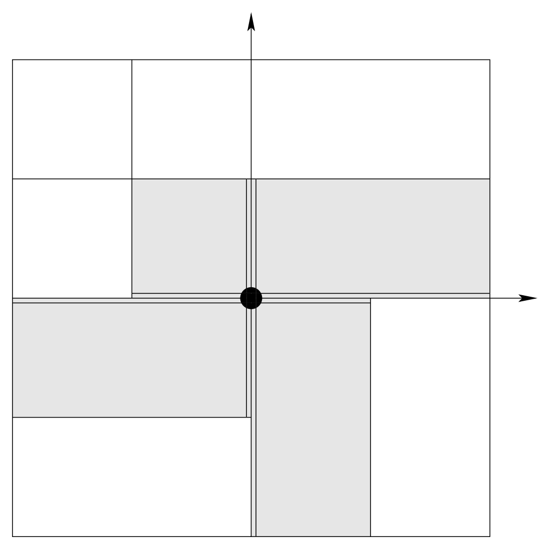
$\Rightarrow$ matching edges which coincide in a vertex is sufficient

- Unisolvent set for $\mathcal{Q}_p$ are
  - additional $p-1$ points on every edge
  - additional $(p-1)^2$ points on every face
  - additional $(p-1)^3$ points in the interior

$\Rightarrow$ matching faces which coincide in an edge is sufficient for continuous edge modes

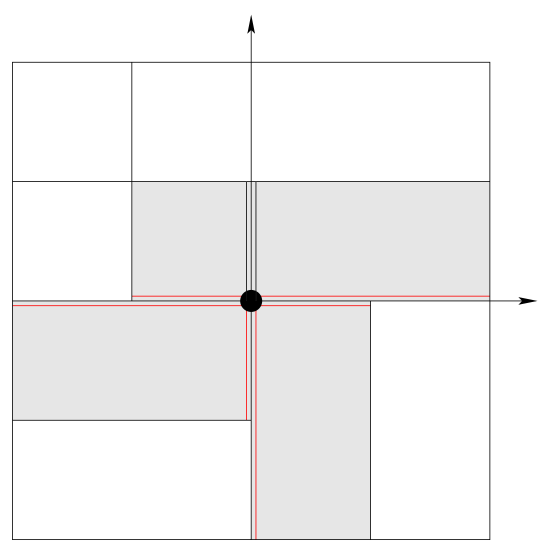
Algorithm for Continuity (Vertex)

- In every cell of the finest mesh, register all edges and cells in their vertices.



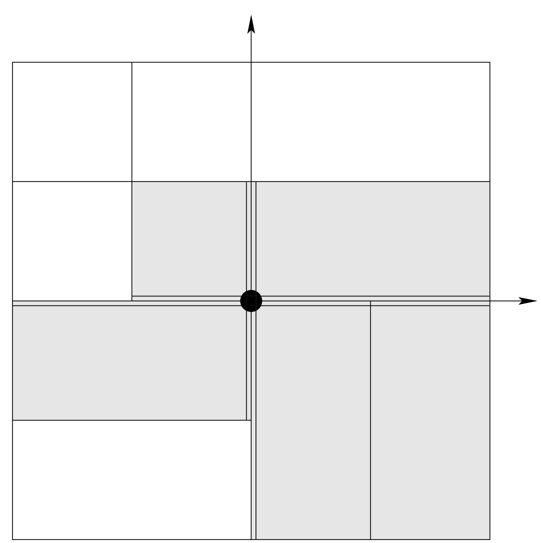
Algorithm for Continuity (Vertex)

- In every cell of the finest mesh, register all edges and cells in their vertices.
- For every vertex, while something changed in the last loop:
  - Check if some of the edges of the vertex have a relationship (ancestor / descendant).



Algorithm for Continuity (Vertex)

- In every cell of the finest mesh, register all edges and cells in their vertices.
- For every vertex, while something changed in the last loop:
  - Check if some of the edges of the vertex have a relationship (ancestor / descendant).
  - If two edges are related, exchange the smaller cell in the list of the vertex by the cell matching the larger cell.
  - Delete the list of edges and rebuild it from the list of cells.



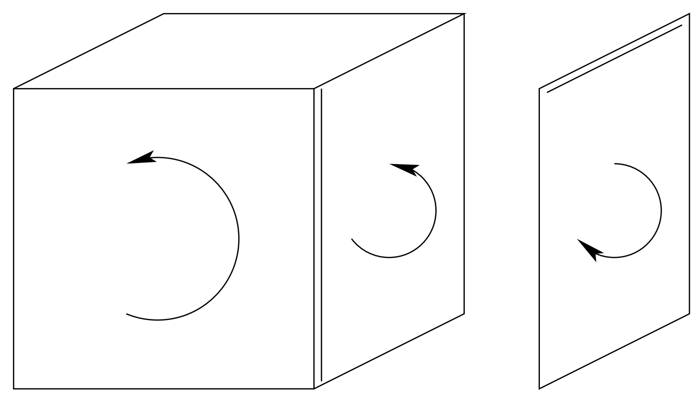
Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- **Anisotropic p refinements**
- hp Meshes
- Perspectives

Anisotropic p

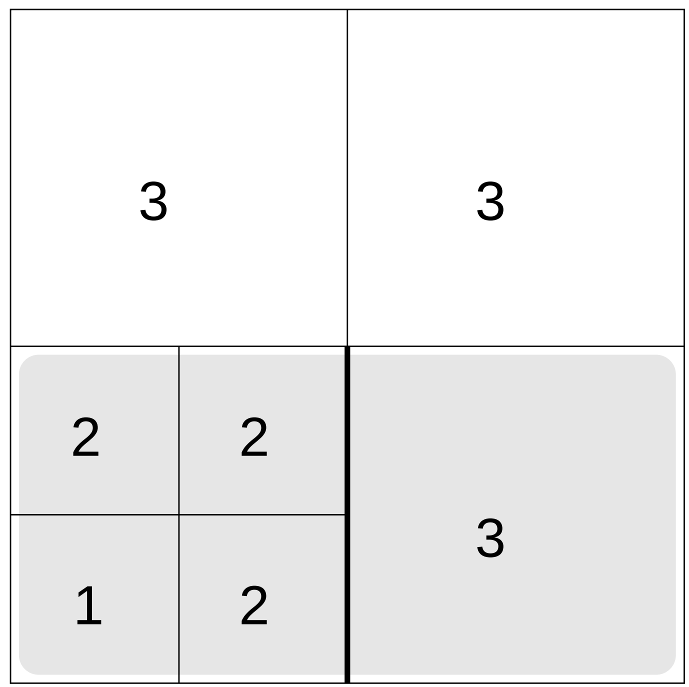
Why anisotropic p? Necessary for thin plates, shells, films.

- Every edge has p, every face has $\underline{p} = (p_0, p_1)$, every cell has $\underline{p} = (p_0, p_1, p_2)$. They can differ in a cell!
- Every face exists only once. It has a distinguished edge and an orientation. The orientation of the face in the cell and the position of the marked edge are stored: $\rho = 1$, $\tau = 3$.



p Enrichment on Edges

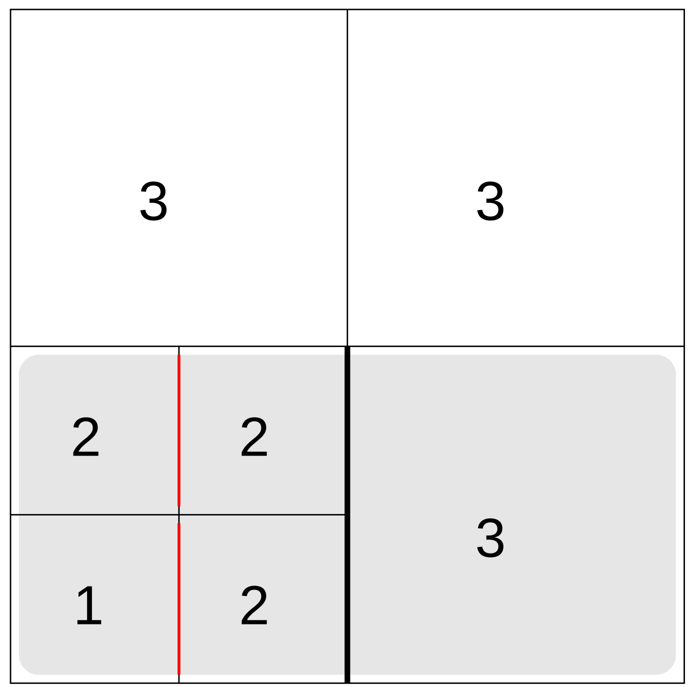
- $p^\star \geq 2$ for the basis functions Φ_i on the marked edge must be possible to achieve exponential convergence.



Analogly for edge and faces in 3D.

p Enrichment on Edges

- $p^\star \geq 2$ for the basis functions Φ_i on the marked edge must be possible to achieve exponential convergence.
- The basis functions on the red edges contribute to Φ_i $\Rightarrow p \geq p^\star$ must be possible and enforced.

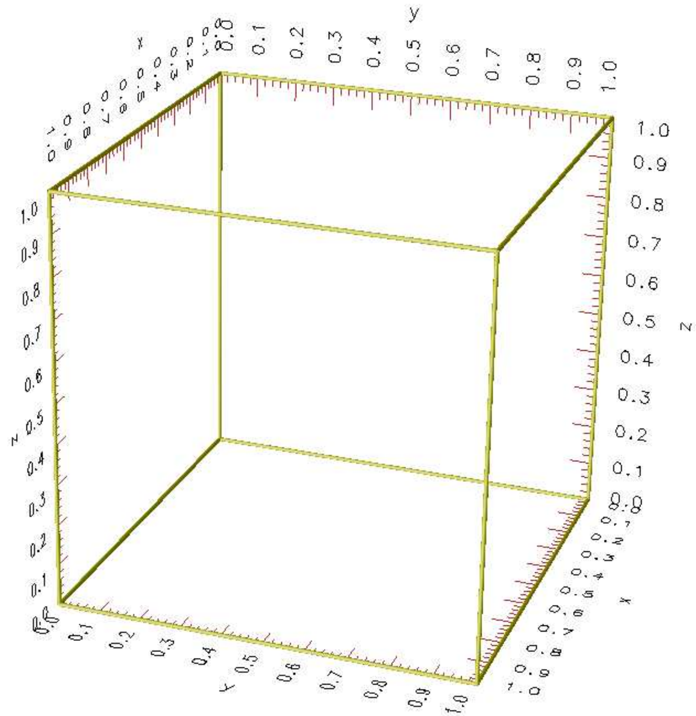


Analogly for edge and faces in 3D.

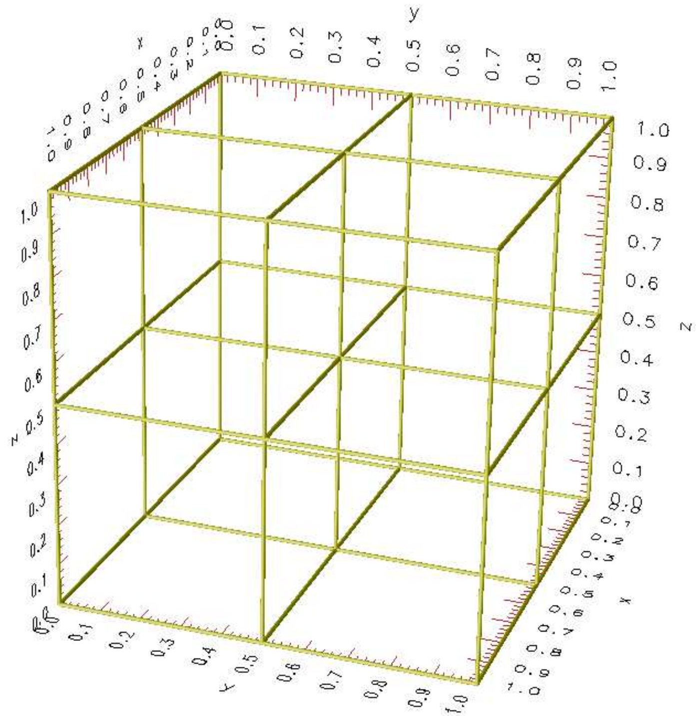
Overview

- Introduction
- Anisotropic h refinements
  - S and T matrices
  - Assembly of Supports
- Anisotropic p refinements
- hp Meshes
- Perspectives

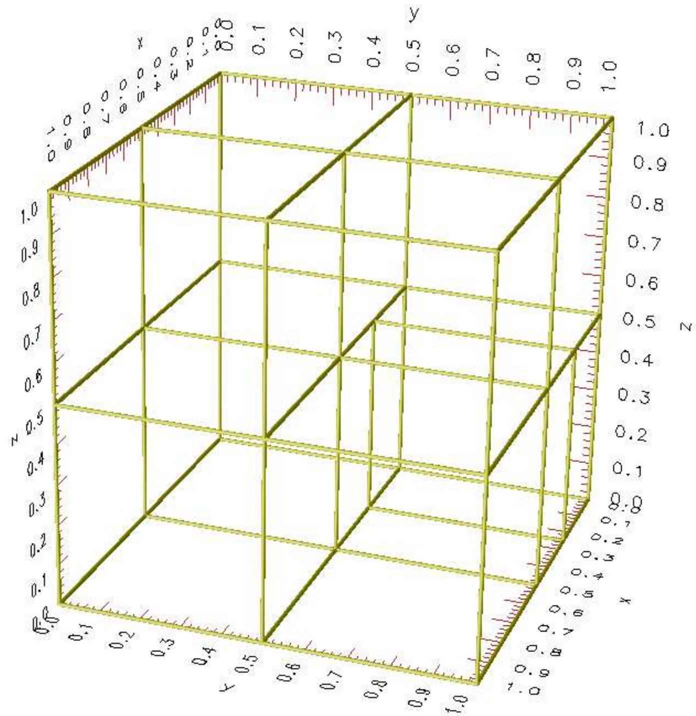
Some Basis Functions



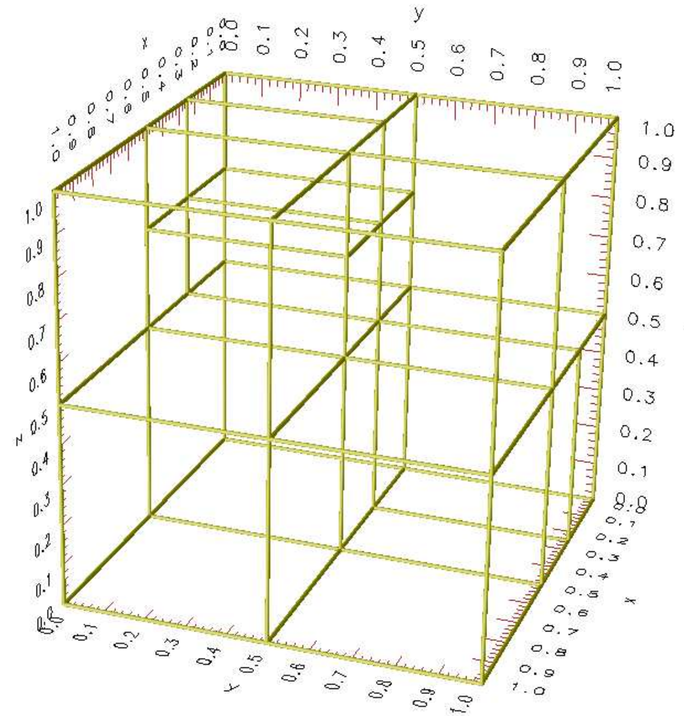
Some Basis Functions



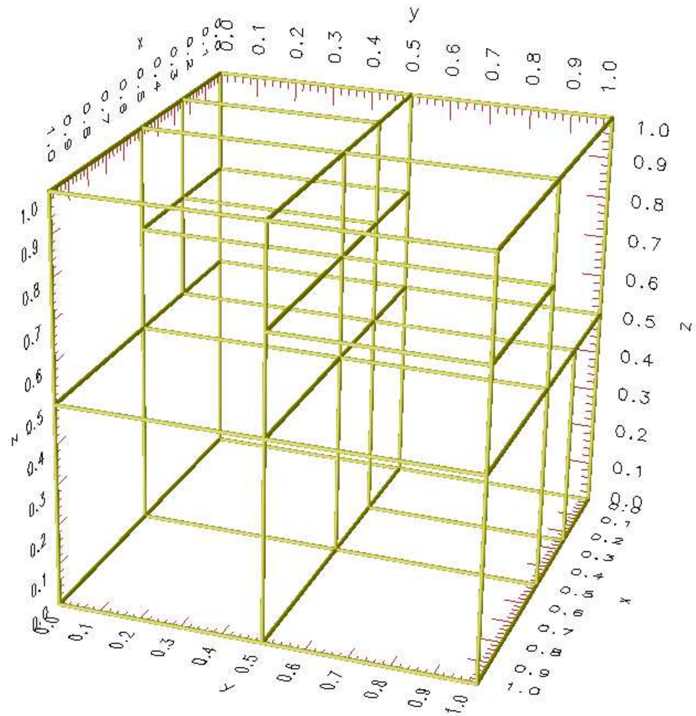
Some Basis Functions

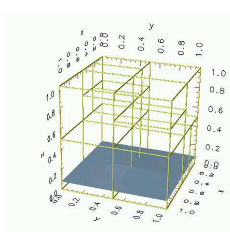


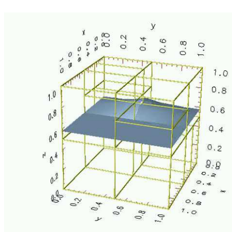
Some Basis Functions

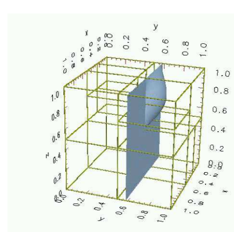


Some Basis Functions



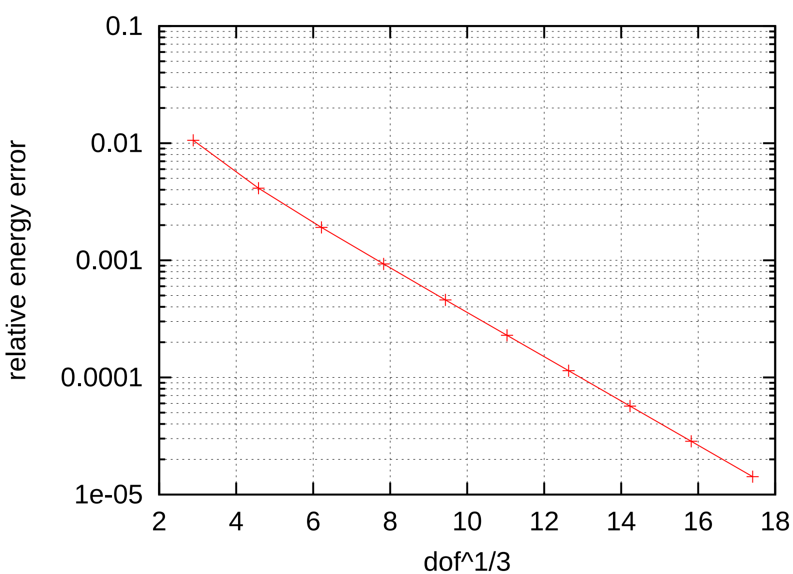


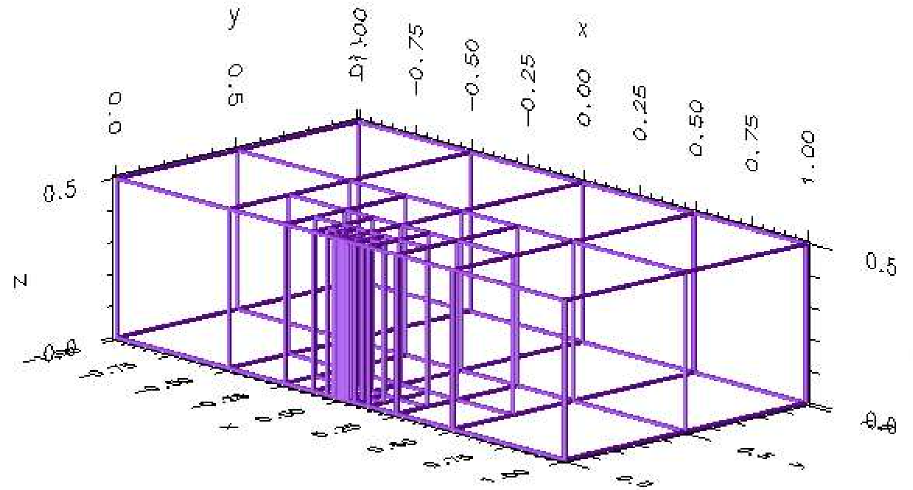




Exponential Convergence in Pseudo-3D

Edge type singularity.





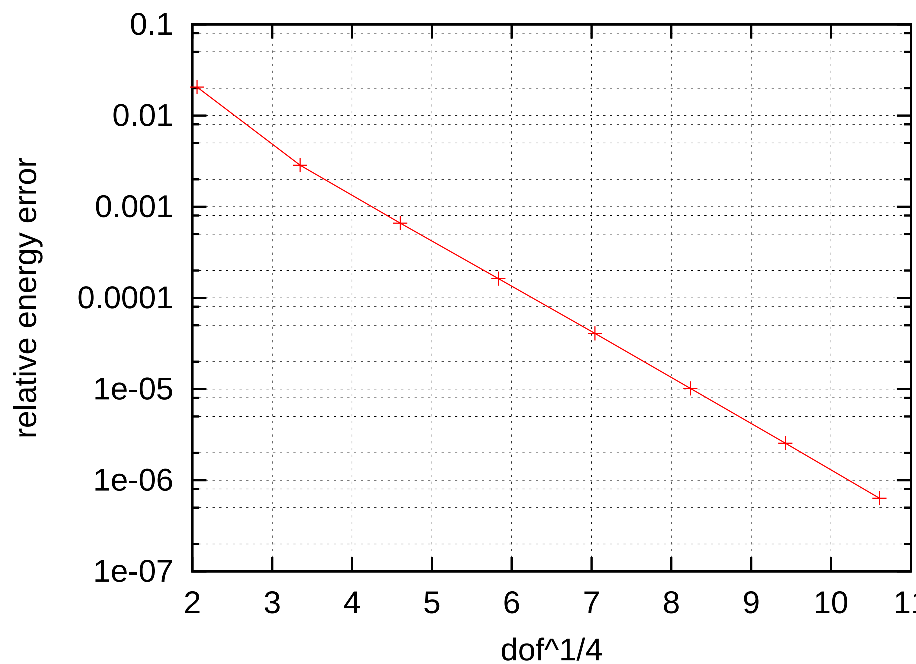
$$-\Delta u + u = f \text{ in } \Omega = (-1,1)\times(0,1)\times(0,{}^1\!/_2)$$

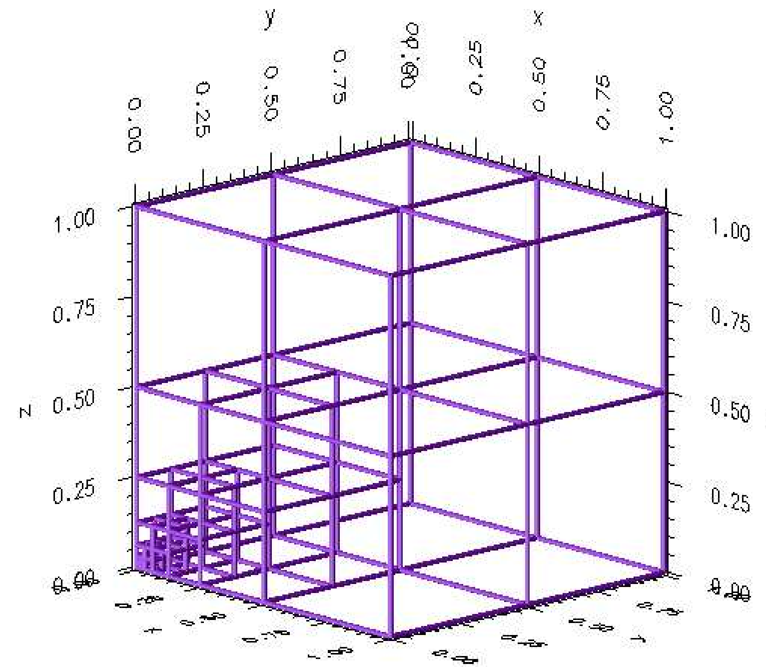
$$u(r,\phi,z) = \sqrt{r}\sin({}^{\phi}\!/_2)z(1-z) \qquad \text{in } \Omega$$

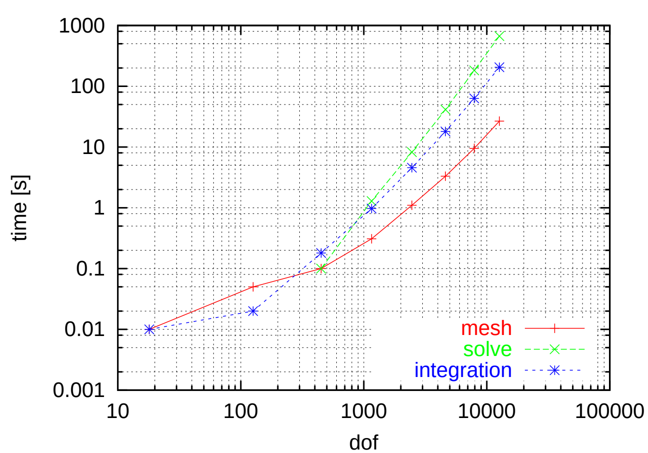
$$u = 0 \qquad \text{on } \{z=0\}\subset\partial\Omega$$

$$\text{and on } \{y=0\}\cap\{x\geq 0\}\subset\partial\Omega$$

Exponential Convergence in 3D







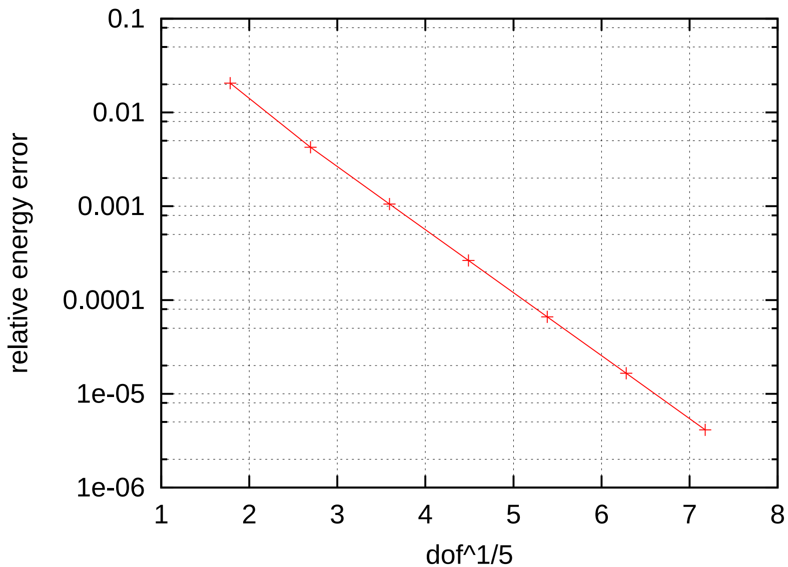
Vertex type singularity.

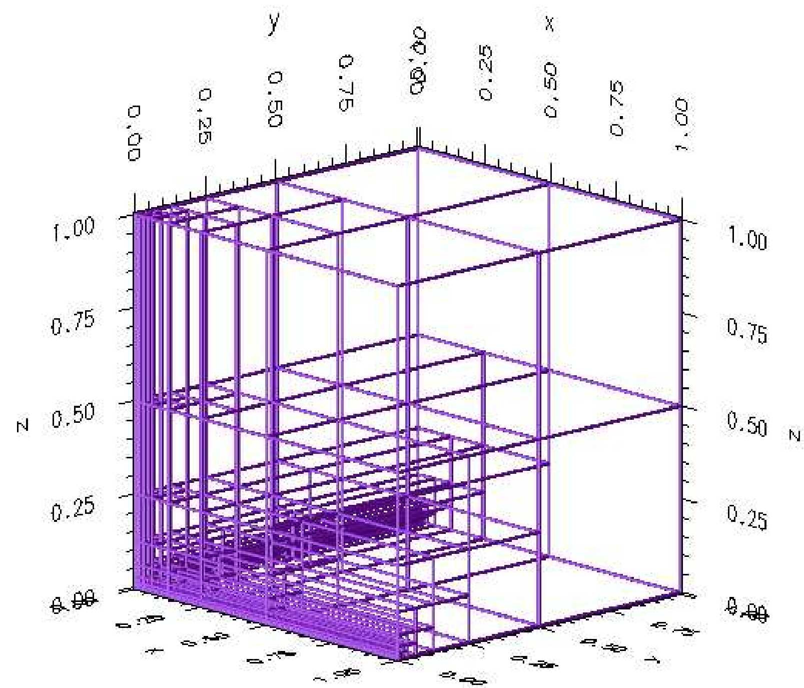
$$-\Delta u + u = f \text{ in } \Omega = (0,1)^3$$

$$u(r,\theta,\phi) = \sqrt{r}\sin\theta\sin\phi \qquad \text{in } \Omega$$

$$u = 0 \qquad \text{on } \{y=0\} \subset \partial\Omega$$

Exp. Conv. in 3D, Edge Mesh





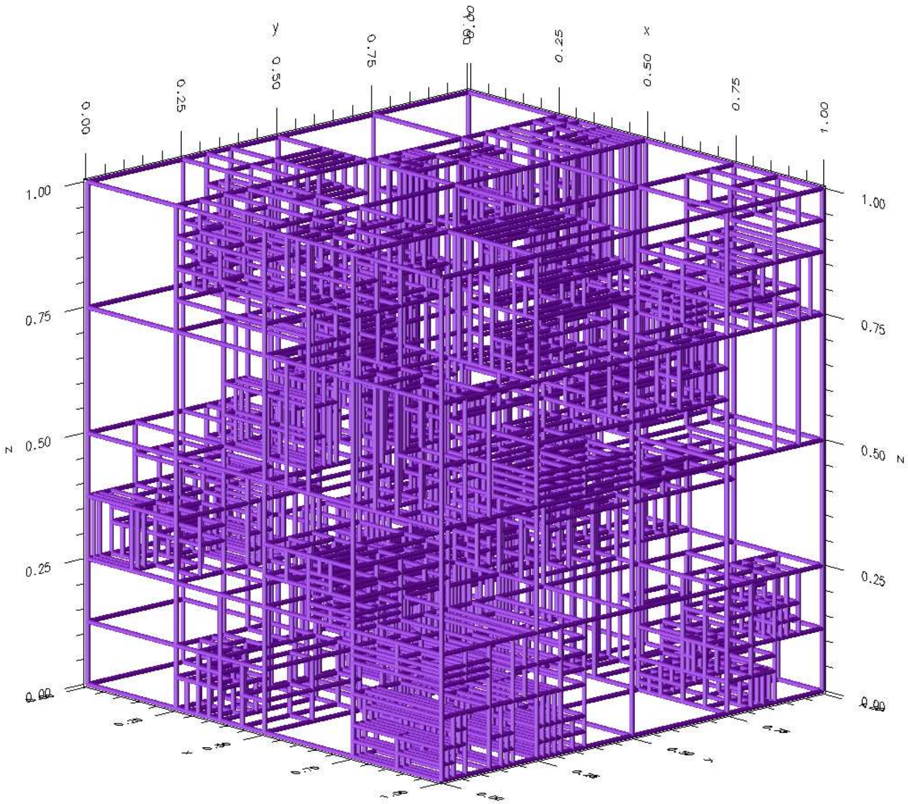
Vertex type singularity.

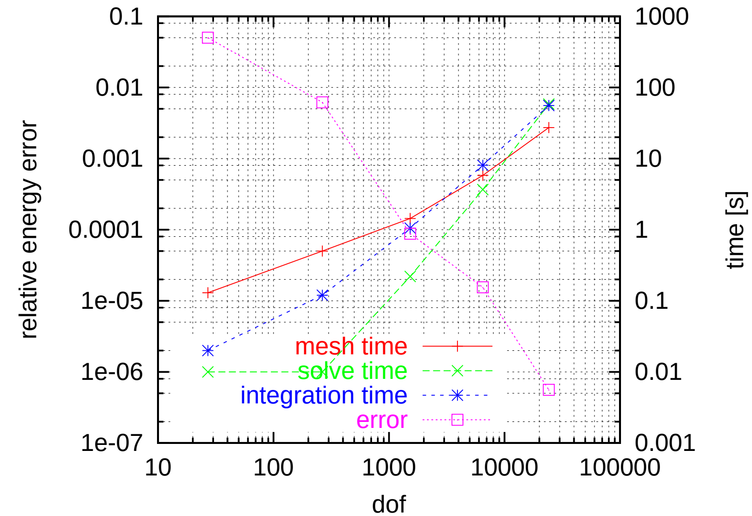
$$-\Delta u + u = f \text{ in } \Omega = (0,1)^3$$

$$u(r,\theta,\phi) = \sqrt{r}\sin\theta\sin\phi \qquad \text{in } \Omega$$

$$u = 0 \qquad \text{on } \{y=0\} \subset \partial\Omega$$

Test: Randomized Refinement





Refinements which are not allowed were dropped.

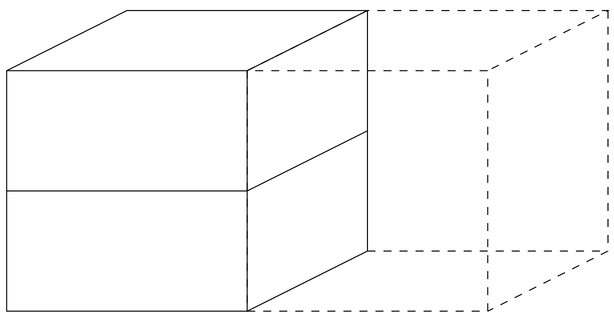
$$-\Delta u + u = f \text{ in } \Omega = (0,1)^3$$

$$u(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z) \quad \text{in } \Omega$$

$$u = 0 \quad \text{on } \partial\Omega$$

Perspectives

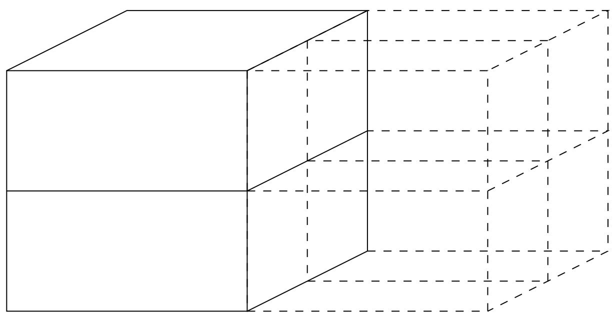
- Applications: Maxwell with weighted regularization (Costabel, Dauge)
- Anisotropic error estimation,
  anistropic regularity estimation
- Improved mesh handling



- Iterative multilevel domain decompositioning solvers:
  Toselli (Zürich), Schöberl (Linz)

Perspectives

- Applications: Maxwell with weighted regularization (Costabel, Dauge)
- Anisotropic error estimation,
  anistropic regularity estimation
- Improved mesh handling



- Iterative multilevel domain decompositioning solvers:
  Toselli (Zürich), Schöberl (Linz)

Hanging Nodes in Isotropic Meshes

- Traverse all cells on locally finest level: mark every vertex / edge / face being used.
- On next (hierarchical) traversal of the mesh:
  - Add dofs which are marked to be on the current level to the list L of local dofs. Mark dof as registered.
  - If cell is on finest level $L \rightarrow$ T matrix
  - Otherwise $S \cdot L$ is added to L of child (next deeper level)

Mortar

- Give up $\mathcal{C}^0$, introduce Lagrange multiplier (the mortar)
- $-\Delta u = f$ in Ω with hom. Dirichlet bc. using mortar method leads to $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{\Lambda} \\ \boldsymbol{\Lambda}^\top & 0 \end{pmatrix} \cdot \begin{pmatrix} \underline{u} \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ \underline{0} \end{pmatrix}$, ie.

$$\begin{aligned} \text{SPD PDE} \;&\not\Rightarrow\; \text{SPD matrix} \\ &\Rightarrow\; \text{conjugate gradients not applicable} \\ &\Rightarrow\; \text{no standard domain decompositioning solvers} \\ &\Rightarrow\; \text{inf-sup condition needed} \end{aligned}$$

- The inf-sup cond. is OK in 2D, 3D for shape regular meshes.
  Not OK for hp FEM, existing proofs only for uniform meshes.
- Analogly for Discontinous Galerkin in 3D: Stability of hp DG on geometric meshes is not clear. First results by Schwab, Toselli, Schötzau for Stokes (not Mortar).