Computing Maxwell Eigenvalues in 3D using H^1 conforming hp FEM

Philipp Frauenfelder

Kersten Schmidt

Christoph Schwab

pfrauenf@math.ethz.ch

Seminar for Applied Mathematics

Federal Institute of Technology, ETH Zürich

ETH

Eidgenössische Technische Hochschule Zürich

Swiss Federal Institute of Technology Zurich

Overview

- Software
- Assembling
- Scalar Results
- Maxwell Eigevalue Problems:
  Weighted Regularization
- Results of Maxwell EVP
- 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, p only)
- Devloo
- Szabó since 1995: STRESSCHECK (p only)
- Heuveline et al.: HiFlow
- In development: deal.II (Kanschat & Bangerth), ngsolve (Schöberl et al.)

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 [1].
- Currently two parts: hp-FEM, BEM (wavelet and multipole methods).
- C++ class library for general elliptic PDEs.

[1] P. F. and Ch. Lage, "Concepts—An Object Oriented Software Package for Partial Differential Equations", *Mathematical Modelling and Numerical Analysis* 36 (5), pp. 937–951 (2002).

Overview

- Software
- Assembling
- Scalar Results
- Maxwell Eigevalue Problems:
  Weighted Regularization
- Results of Maxwell EVP
- 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

Stiffness matrix: $A_{ij} = a(\phi_i, \phi_j)$, load vector: $l_i = l(\phi_i)$.
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

Stiffness matrix: $A_{ij} = a(\phi_i, \phi_j)$, load vector: $l_i = l(\phi_i)$.
Assembling:

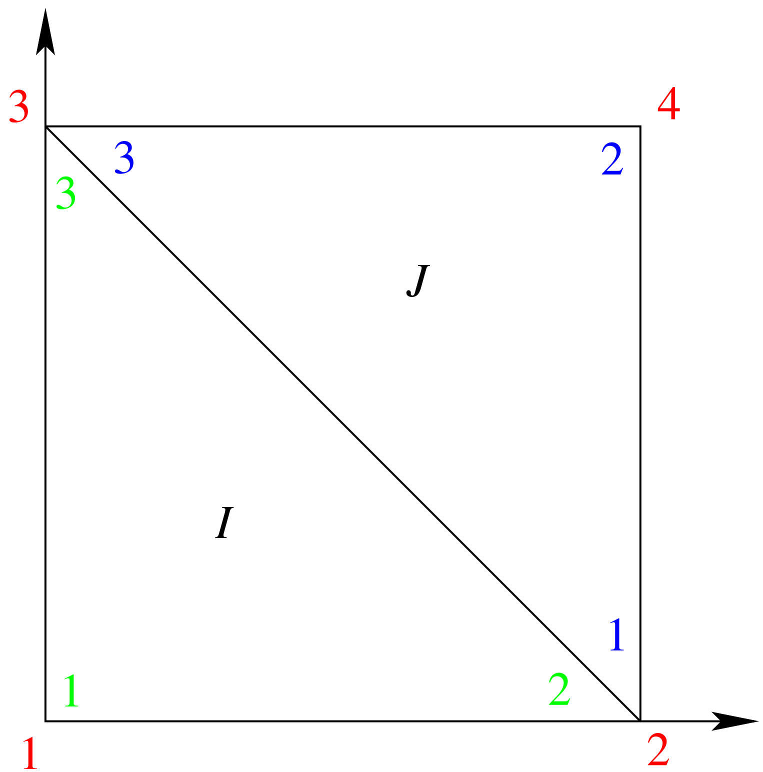
$$\underline{l} = l(\underline{\Phi}) = l\Big(\sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{\phi}^{\tilde{K}}\Big) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} l(\underline{\phi}^{\tilde{K}}) = \sum_{\tilde{K}} \boldsymbol{T}_{\tilde{K}}^{\top} \underline{l}_{\tilde{K}}$$

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

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

Example 1: Regular Mesh

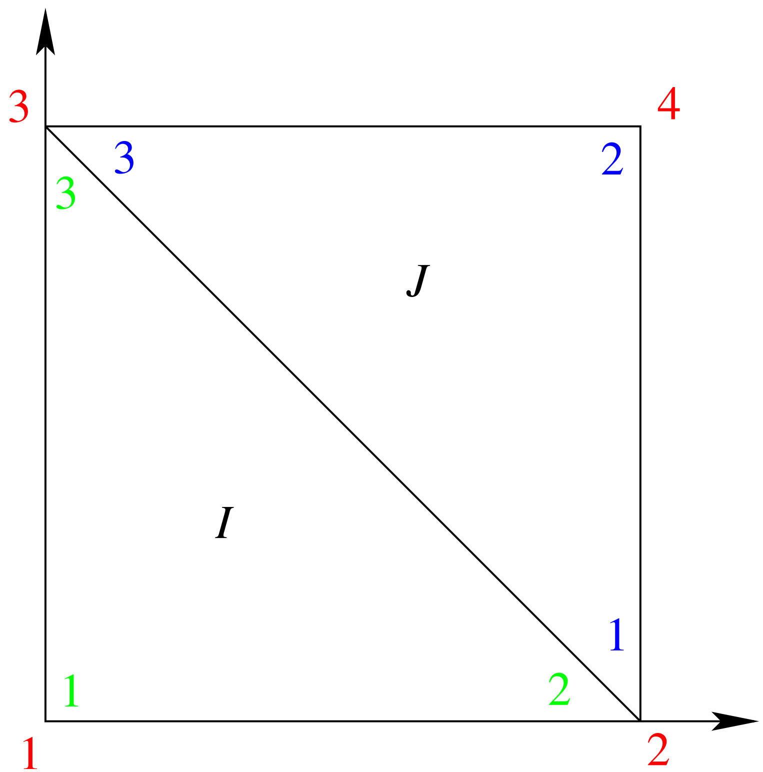
Two elements with three local shape functions each and four global basis functions.



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

Example 1: Regular Mesh

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

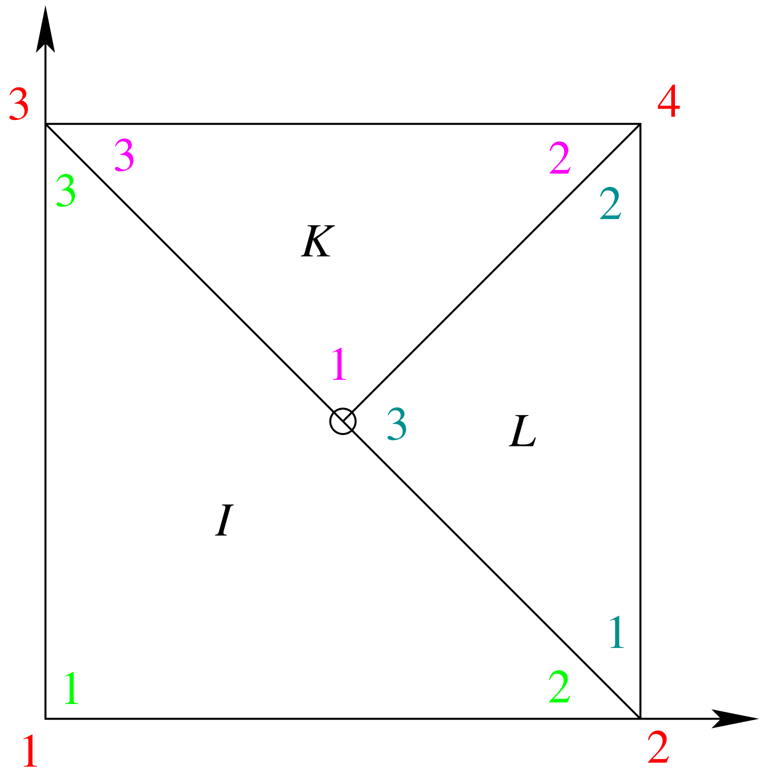


$$T_I = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \end{pmatrix}$$

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

Example 2: Irregular Mesh

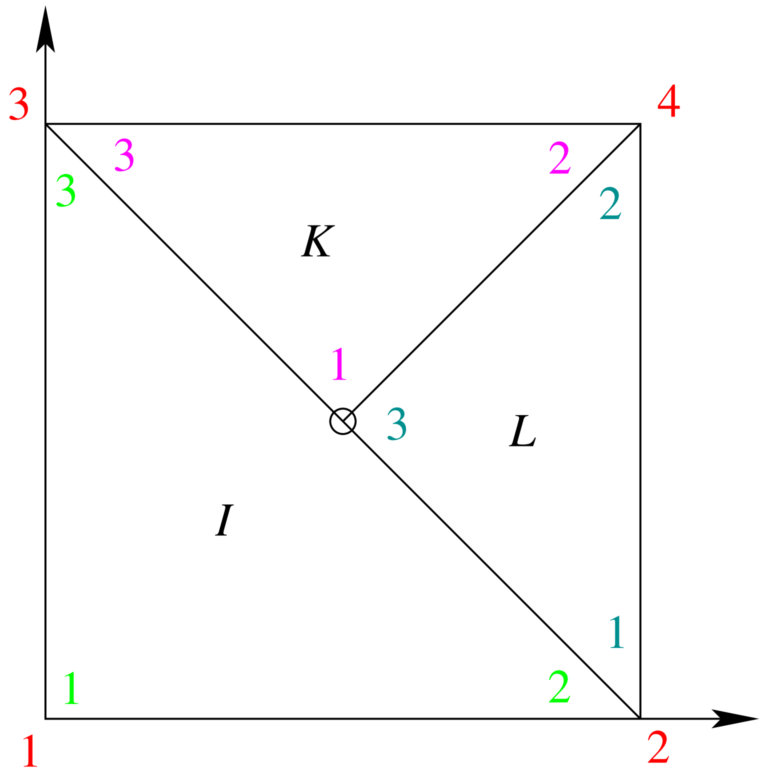
Three elements with three local shape functions each and four global basis functions. The hanging node is marked with $\circ$.



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

Example 2: Irregular Mesh

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



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

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

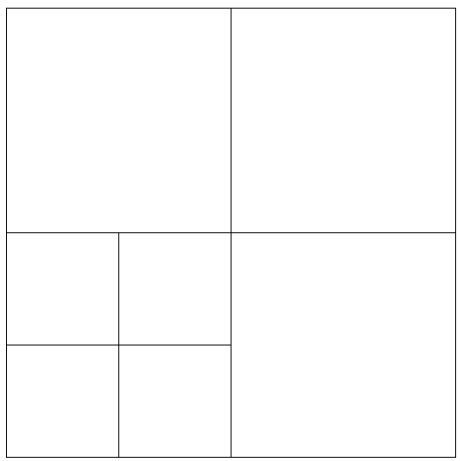
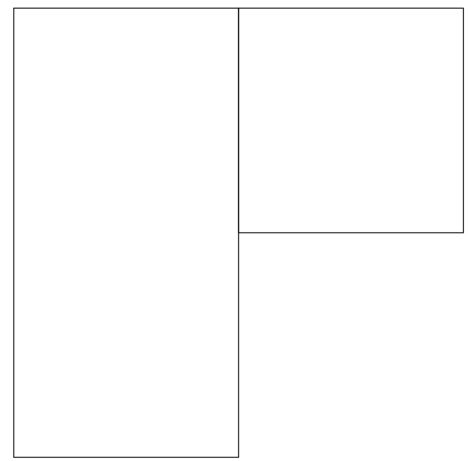
$\Rightarrow$ continuous basis functions.

Generation of T Matrices

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

Generation of T Matrices

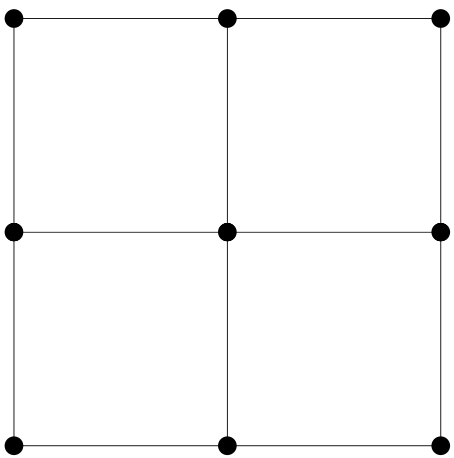
- **Regular Mesh:** Counting and assigning indices with respect to topological entities such as vertices, edges and faces.
- **Irregular Mesh:** Irregularity due to a refinement of an initially regular mesh.

OK:  , not OK: 

Explanation follows.

T Matrices for Irregular Meshes

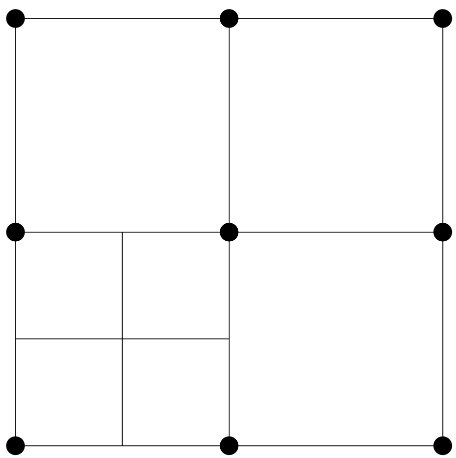
Irregularity due to a refinement of an initially regular mesh.



T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

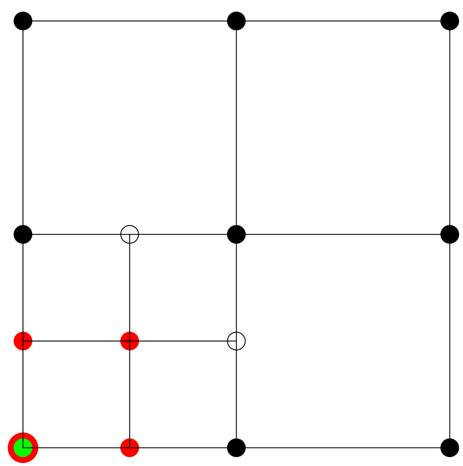
Mesh	$\mathcal{M}$	refine	$\mathcal{M}'$
Basis fcts.	$B = B_{\text{repl}} \cup B_{\text{keep}}$	$\longrightarrow$	$B' = B_{\text{ins}} \cup B_{\text{keep}}$



T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

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



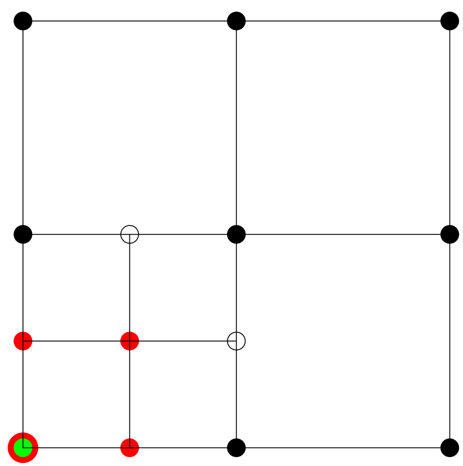
B_{repl}: basis fcts. which can be solely described by elements of $\mathcal{M}' \setminus \mathcal{M}$

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

T Matrices for Irregular Meshes

Irregularity due to a refinement of an initially regular mesh.

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



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

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.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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*

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

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

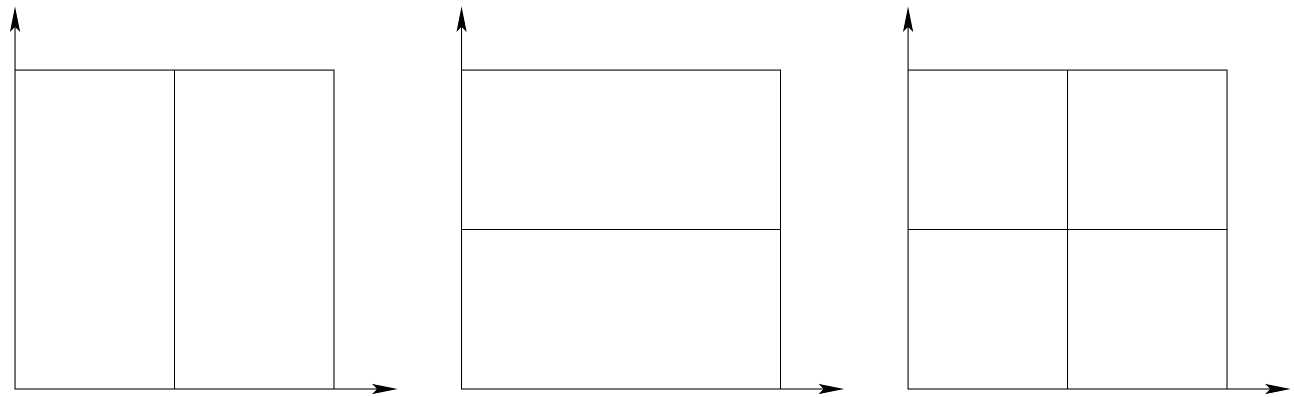
Proposition 2. *Let $\hat{K}' \subset \hat{K}$ be the result of a refinement of the reference element $\hat{K}$ with $H : \hat{K} \to \hat{K}'$ the subdivision map. The element maps are*

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

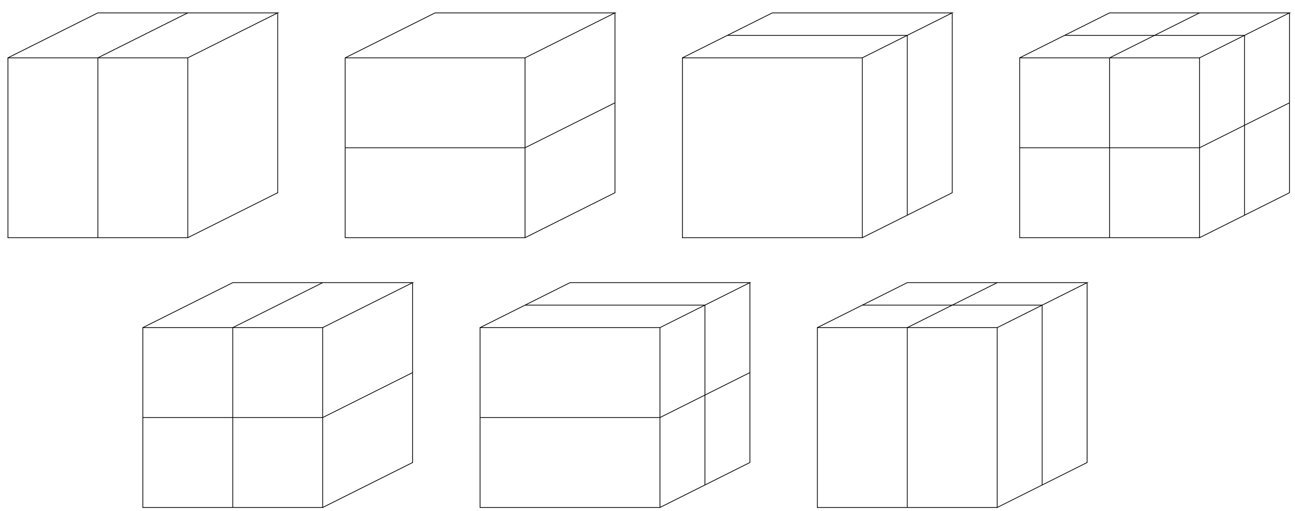
and $F_{K'} \circ H^{-1} = F_K$ holds. Then, $\boldsymbol{S}_{\hat{K}'\hat{K}} = \boldsymbol{S}_{K'K}$.

Subdivisions

Subdivisions of a quadrilateral in 2D:



Subdivisions of a hexahedron in 3D:



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

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

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

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

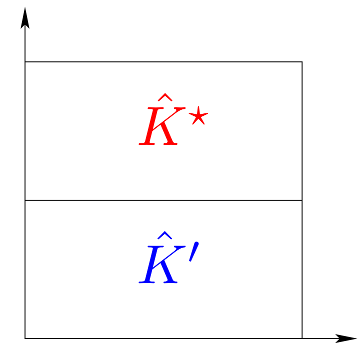
Hierarchic shape functions $\Rightarrow$ hierarchic S matrices.

S Matrices: Tensor Product in 2D

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 $\hat{K}^\star$.

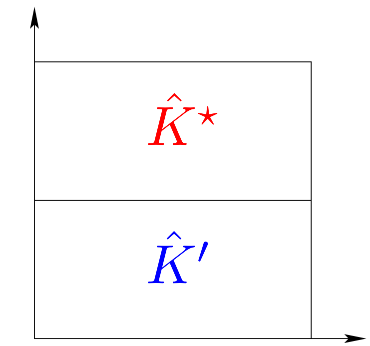


S Matrices: Tensor Product in 2D

Horizontal subdivision:

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

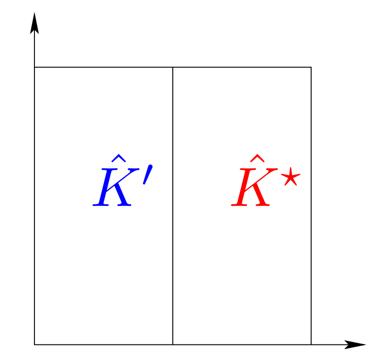
$$\boldsymbol{S}_{\hat{K}^\star\hat{K}} = \boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star\hat{J}}$$ for the top quad $\hat{K}^\star$.



Vertical subdivision:

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

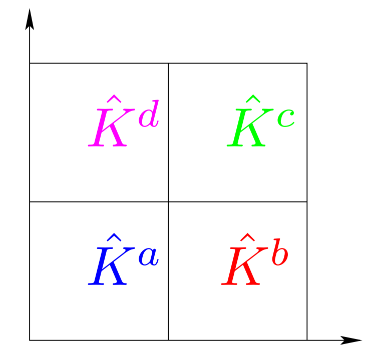
$$\boldsymbol{S}_{\hat{K}^\star\hat{K}} = \boldsymbol{S}_{\hat{J}^\star\hat{J}} \otimes \boldsymbol{E}$$ for the right quad $\hat{K}^\star$.



S Matrices: Tensor Product in 2D & 3D

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.



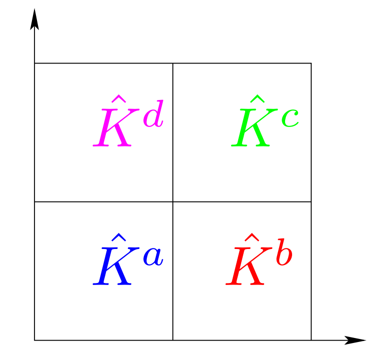
$$\boldsymbol{S}_{\hat{K}^d\hat{K}} = \left(\boldsymbol{S}_{\hat{J}',\hat{J}} \otimes \boldsymbol{E}\right) \cdot \left(\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star,\hat{J}}\right) \qquad \boldsymbol{S}_{\hat{K}^c\hat{K}} = \left(\boldsymbol{S}_{\hat{J}^\star,\hat{J}} \otimes \boldsymbol{E}\right) \cdot \left(\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star,\hat{J}}\right)$$

$$\boldsymbol{S}_{\hat{K}^a\hat{K}} = \left(\boldsymbol{S}_{\hat{J}',\hat{J}} \otimes \boldsymbol{E}\right) \cdot \left(\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}',\hat{J}}\right) \qquad \boldsymbol{S}_{\hat{K}^b\hat{K}} = \left(\boldsymbol{S}_{\hat{J}^\star,\hat{J}} \otimes \boldsymbol{E}\right) \cdot \left(\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}',\hat{J}}\right)$$

S Matrices: Tensor Product in 2D & 3D

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.



$$\boldsymbol{S}_{\hat{K}^d\hat{K}} = (\boldsymbol{S}_{\hat{J}'\hat{J}} \otimes \boldsymbol{E}) \cdot (\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star\hat{J}}) \quad \boldsymbol{S}_{\hat{K}^c\hat{K}} = (\boldsymbol{S}_{\hat{J}^\star\hat{J}} \otimes \boldsymbol{E}) \cdot (\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}^\star\hat{J}})$$

$$\boldsymbol{S}_{\hat{K}^a\hat{K}} = (\boldsymbol{S}_{\hat{J}'\hat{J}} \otimes \boldsymbol{E}) \cdot (\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}'\hat{J}}) \quad \boldsymbol{S}_{\hat{K}^b\hat{K}} = (\boldsymbol{S}_{\hat{J}^\star\hat{J}} \otimes \boldsymbol{E}) \cdot (\boldsymbol{E} \otimes \boldsymbol{S}_{\hat{J}'\hat{J}})$$

3D: Same idea as in 2D, just of this form:

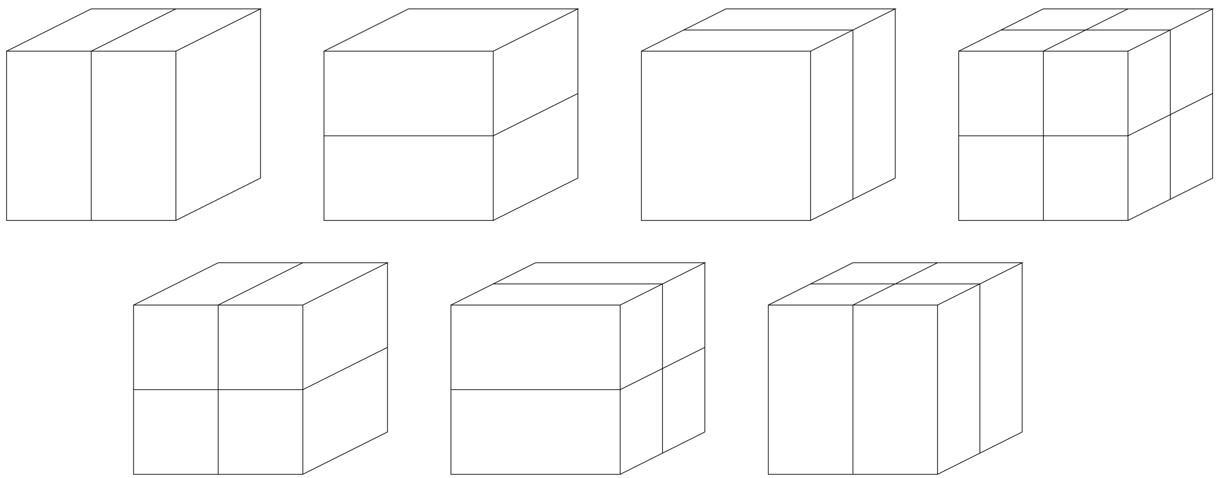
$$\boldsymbol{S}_{\hat{K}'\hat{K}} = \prod (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C})$$

in each of the factors, one of $\boldsymbol{A}$, $\boldsymbol{B}$ or $\boldsymbol{C}$ is an 1D S matrix.

S Matrices: Tensor-Product in 3D

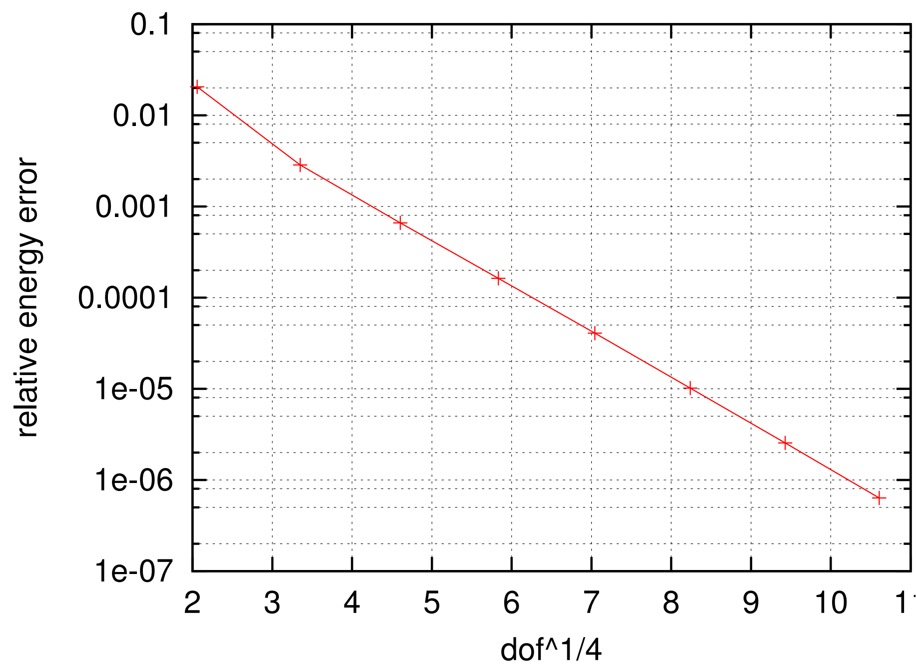
$$S_{\hat{K}'\hat{K}} = \prod (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C})$$

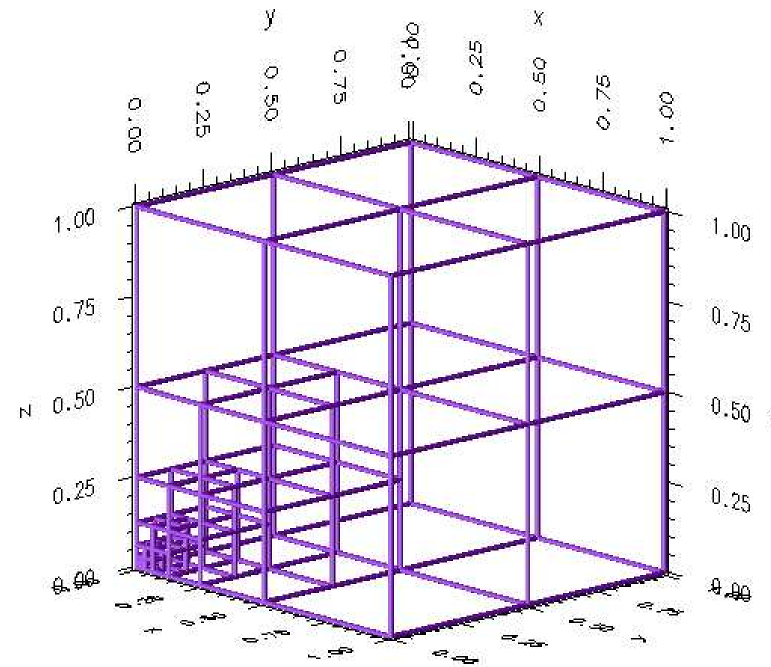
in each of the factors, one of $\boldsymbol{A}$, $\boldsymbol{B}$ or $\boldsymbol{C}$ is an 1D S matrix.
Depending on the factors, 7 subdivisions are possible:

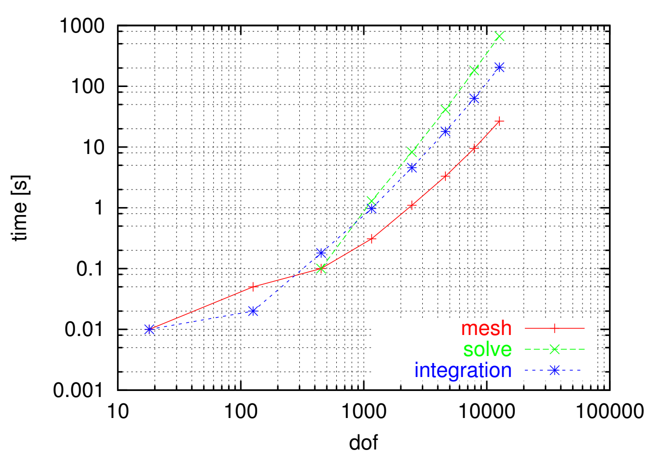


Concepts: arbitrary number and combination of these 7 subdivisions in 3D.

Scalar Computations: Vertex Singularity







Vertex type singularity.

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

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

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

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Overview

- Software
- Assembling
- Scalar Results
- Maxwell Eigevalue Problems:
  Weighted Regularization
- Results of Maxwell EVP
- Perspectives

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Electric Eigenvalue Problem

Find $\omega > 0$ sucht that $\exists \underline{E} \in X_n \setminus \{0\}$ with

$$\int_\Omega \mu^{-1} \operatorname{curl} \underline{E} \cdot \operatorname{curl} \underline{F} + \int_\Omega \operatorname{div} \underline{E} \operatorname{div} \underline{F} = \omega^2 \int_\Omega \varepsilon \underline{E} \cdot \underline{F} \quad \forall \underline{F} \in X_n$$

$H_n := \{\underline{u} \in H^1(\Omega)^3 : \underline{u} \wedge \underline{n} = 0 \text{ on } \partial\Omega\}$

- X_n is curl and div conforming, hence continuous across interfaces $\Rightarrow H_n = X_n$
- H_n is easy to discretise and implement: Cartesian product of scalar discretisation $S^{1,\underline{p}}(\Omega, \mathcal{T})$ of $H^1(\Omega)$
- Converges to wrong solutions if Ω has reentrant corners:
    - $H_n \neq X_n$
    - $\operatorname{codim}_{X_n} H_n = \infty$
    - H_n closed in X_n ie., sequences in H_n have their limits in H_n.

Weighted Regularization

Find the frequencies $\omega > 0$ such that $\exists \underline{E} \in H_n \setminus \{0\}$ with

$$\int_\Omega \operatorname{curl} \underline{E} \cdot \operatorname{curl} \underline{F} + s\langle \underline{E}, \underline{F} \rangle_Y = \omega^2 \int_\Omega \underline{E} \cdot \underline{F} \qquad \forall \underline{F} \in H_n$$

$$\langle \underline{E}, \underline{F} \rangle_Y = \int_\Omega \rho(\underline{x}) \operatorname{div} \underline{E} \operatorname{div} \underline{F}$$

Properly chosen weight $\rho(\underline{x})$ and $s \in \mathbb{R}_+$.

Weighted Regularization

Find the frequencies $\omega > 0$ such that $\exists \underline{E} \in H_n \setminus \{0\}$ with

$$\int_\Omega \operatorname{curl} \underline{E} \cdot \operatorname{curl} \underline{F} + s\langle \underline{E}, \underline{F} \rangle_Y = \omega^2 \int_\Omega \underline{E} \cdot \underline{F} \qquad \forall \underline{F} \in H_n$$

$$\langle \underline{E}, \underline{F} \rangle_Y = \int_\Omega \rho(\underline{x}) \operatorname{div} \underline{E} \operatorname{div} \underline{F}$$

Properly chosen weight $\rho(\underline{x})$ and $s \in \mathbb{R}_+$.

Idea: use spaces

$$X_n[Y] := \{\underline{u} \in H_0(\operatorname{curl}, \Omega) : \operatorname{div} \underline{u} \in Y\} \supset H_n \text{ dense}$$

and the solutions of Maxwell equations $\in X_n[Y]$.

[2] Martin Costabel and Monique Dauge, "Weighted regularization of Maxwell equations in polyhedral domains", *Numer. Math.* 93 (2), pp. 239–277 (2002).

Chosing the Weight and s

$$s\langle \underline{E}, \underline{F} \rangle_Y = s \int_\Omega \rho(\underline{x}) \operatorname{div} \underline{E} \operatorname{div} \underline{F}$$

2D:
$\rho(\underline{x}) = r^\alpha$ where r is the distance to a reentrant corner and $\alpha \in [0, 2]$ depending on the angle of the reentrant corner: $\alpha \in (2 - 2^\pi/\omega_c, 2]$

s scales the $\langle . , . \rangle_Y$ form. Spurious Eigenvalues get scaled too, real Eigenvalues not. Sensible range: $(0, 30)$. $s = 0$ gives a large kernel since $\operatorname{div} \underline{E} = 0$ is not enforced at all.
$\alpha = 2$ is the limiting case, nice to implement since r^2 is polynomial.

Chosing the Weight and s

$$s\langle \underline{E}, \underline{F}\rangle_Y = s \int_\Omega \rho(\underline{x}) \operatorname{div} \underline{E} \operatorname{div} \underline{F}$$

2D:
$\rho(\underline{x}) = r^\alpha$ where r is the distance to a reentrant corner and $\alpha \in [0, 2]$ depending on the angle of the reentrant corner: $\alpha \in (2 - 2^{\pi}/\omega_c, 2]$

3D:

$$\rho(\underline{x}) = \operatorname{dist}(\underline{x}, \mathcal{C} \cup \mathcal{E})^\alpha$$

where $\alpha \in [0, 2]$ (depending on angle of edge and cone of corner).

s scales the $\langle . , . \rangle_Y$ form. Spurious Eigenvalues get scaled too, real Eigenvalues not. Sensible range: $(0, 30)$. $s = 0$ gives a large kernel since $\operatorname{div} \underline{E} = 0$ is not enforced at all.
$\alpha = 2$ is the limiting case, nice to implement since r^2 is polynomial.

Convergence of Eigenvalues

Eigenvectors:

$$\|E_m - E_{m,N}\|_{X_n} \leq C \sup_{F \in W_m} \inf_{F_N \in V_N} \|F - F_N\|_{X_n}$$

Simple Eigenvalues:

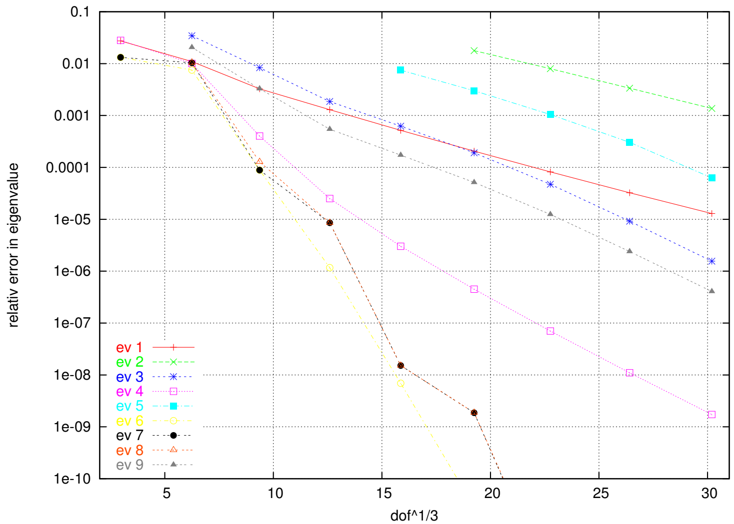
$$|\lambda_m - \lambda_{m,N}| \leq C \sup_{F \in W_m} \inf_{F_N \in V_N} \|F - F_N\|^2_{X_n}$$

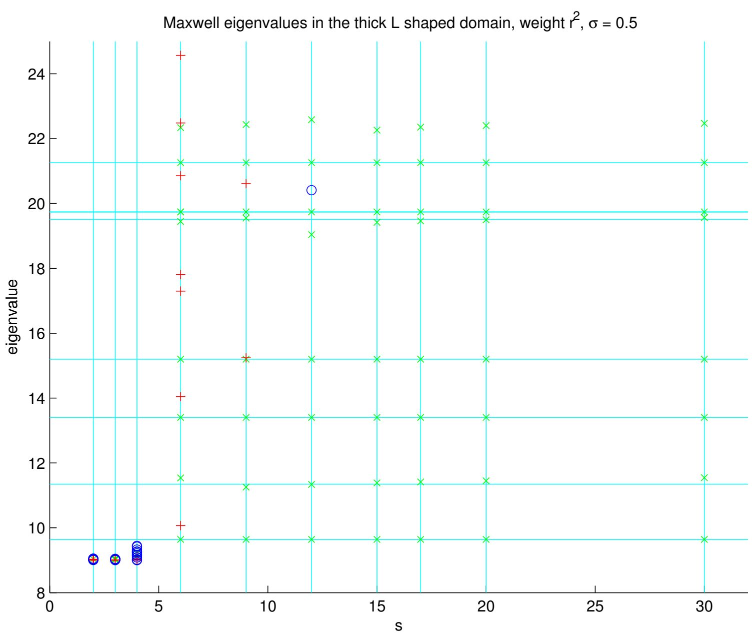
For $\|F - F_N\|_{X_n}$, exponential convergence possible:
$\mathbb{R}^2$: Proof by Costabel, Dauge, Schwab
$\mathbb{R}^3$: experimental evidence, proof in preparation

Overview

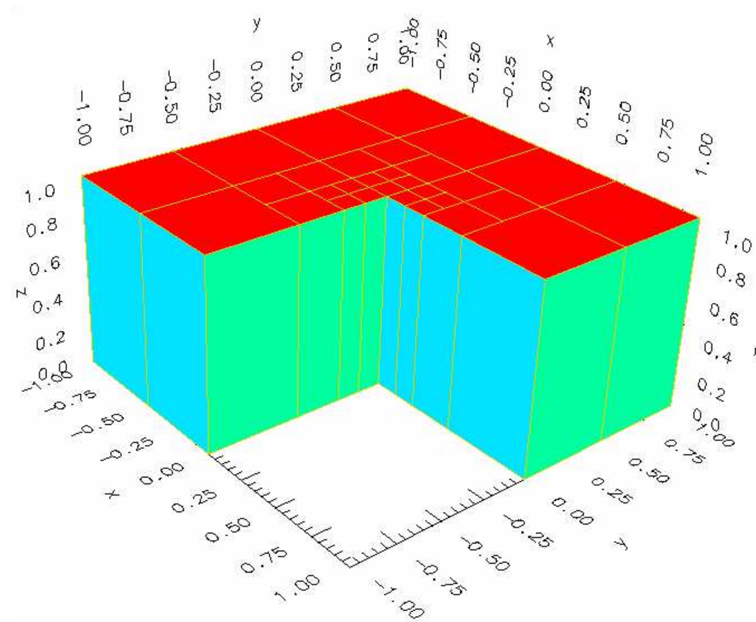
- Software
- Assembling
- Scalar Results
- Maxwell Eigevalue Problems:
  Weighted Regularization
- Results of Maxwell EVP
- Perspectives

EVP in the Thick L Shaped Domain





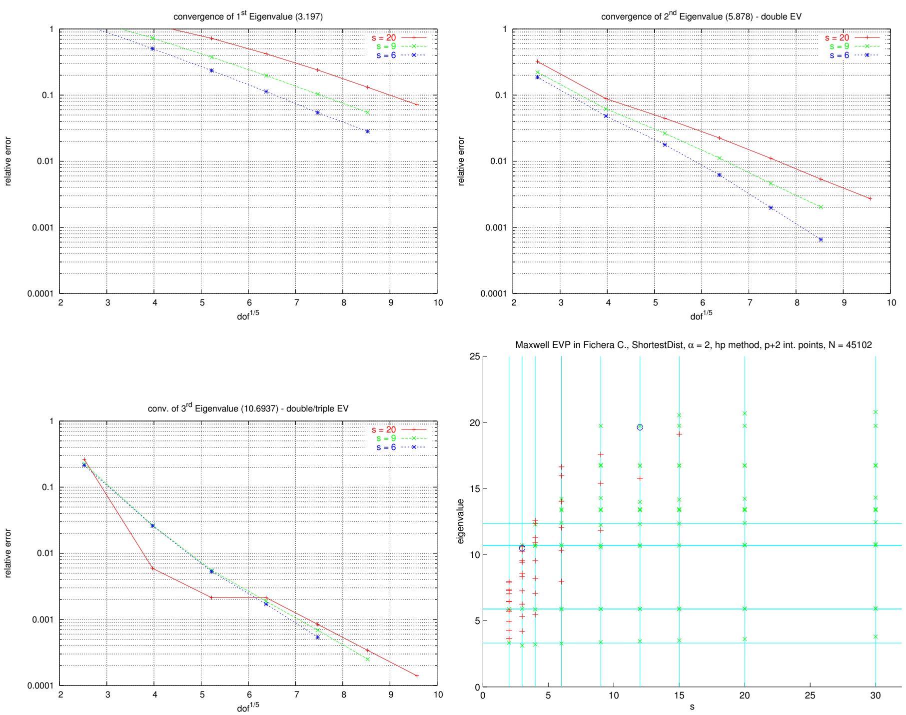
7054 dof



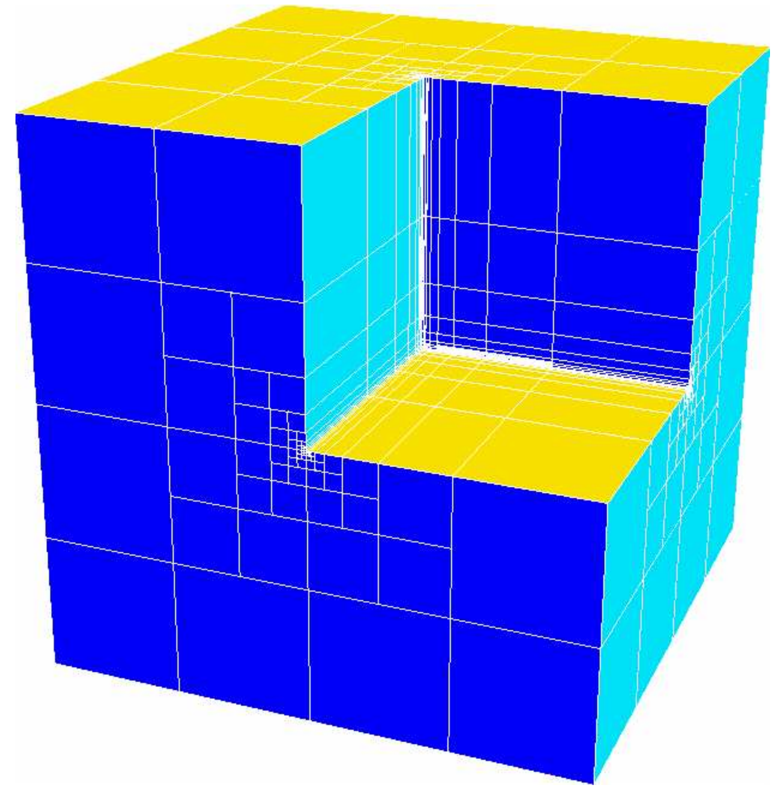
ShortestDist

$\alpha = 2$

EVP in the Fichera Corner



45102 dof

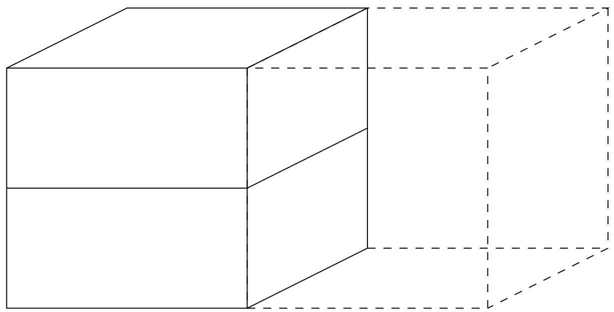


ShortestDist

$\alpha = 2$

Perspectives

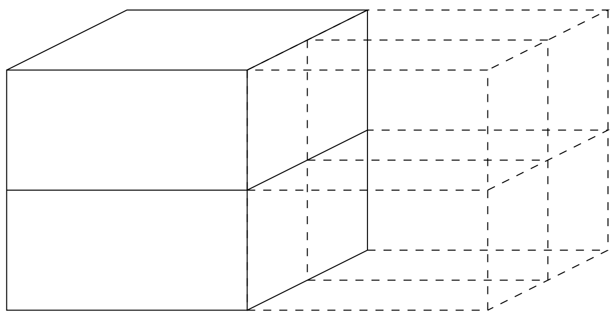
- Maxwell source problems
- A posteriori error estimation, anisotropic regularity estimation
- Improved mesh handling



- Iterative multilevel domain decompositioning solvers:
  Toselli (Zürich), Schöberl (Linz)
- Open Source version of Concepts. Contact:
  `pfrauenf@math.ethz.ch`

Perspectives

- Maxwell source problems
- A posteriori error estimation, anisotropic regularity estimation
- Improved mesh handling



- Iterative multilevel domain decompositioning solvers:
  Toselli (Zürich), Schöberl (Linz)
- Open Source version of Concepts. Contact:
  `pfrauenf@math.ethz.ch`

Shape Functions

The reference element shape functions on $(-1,1)$ of order p [3]:

$$N_i(\xi) = \begin{cases} \frac{1-\xi}{2} & i = 0 \\ \frac{1-\xi}{2}\frac{1+\xi}{2}P_{i-1}^{1,1}(\xi) & 1 \le i \le p-1 \\ \frac{1+\xi}{2} & i = p \end{cases}$$

$P_{i-1}^{1,1}(\xi)$ are integrated Legendre Polynomials: $L_i(\xi) = P_i^{0,0}(\xi)$ and

$$\int_{-1}^{\xi}(1-x)^{\alpha}(1+x)^{\beta}P_i^{\alpha,\beta}(x)\,dx = \frac{-1}{2i}(1-\xi)^{\alpha+1}(1+\xi)^{\beta+1}P_{i-1}^{\alpha+1,\beta+1}(\xi)$$

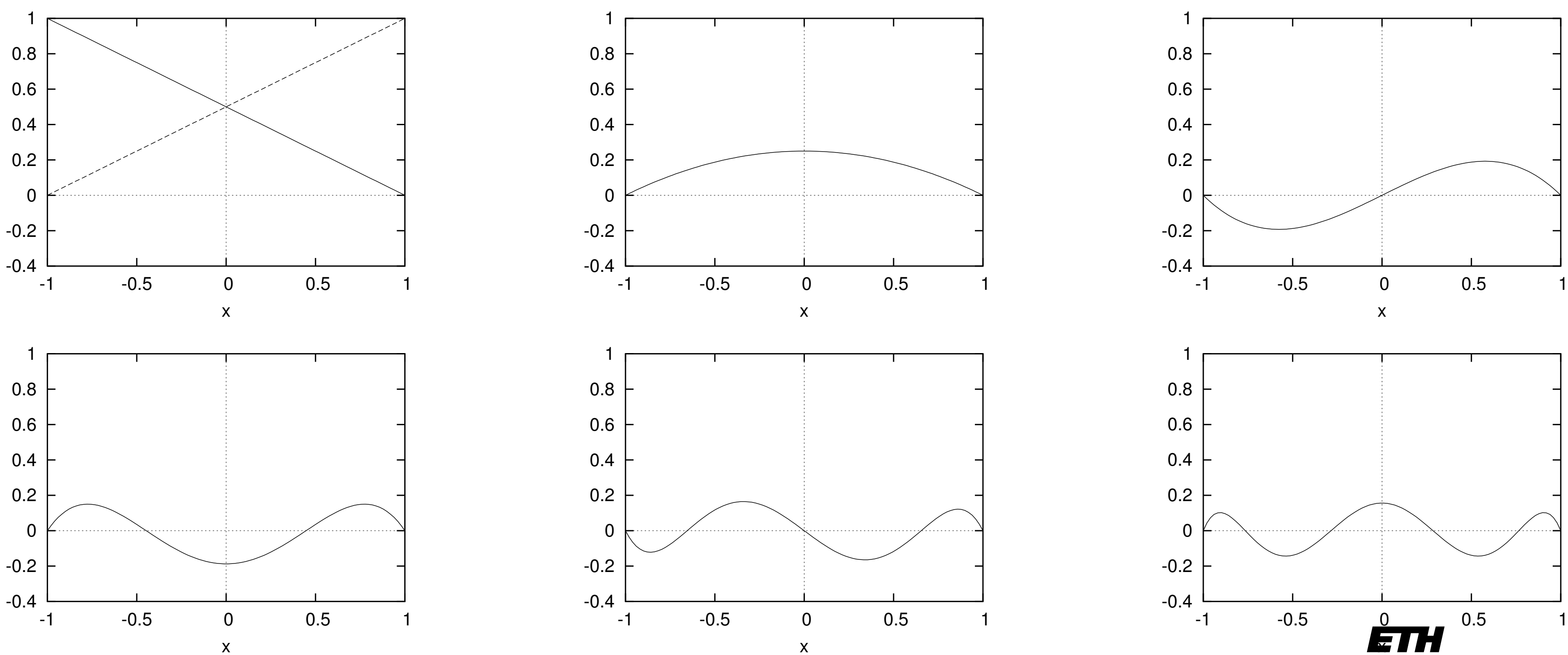
$$\Rightarrow \int_{-1}^{\xi}P_i^{0,0}(x)\,dx = \frac{-1}{2i}(1-\xi)(1+\xi)P_{i-1}^{1,1}(\xi)$$

[3] Karniadakis and Sherwin, "Spectral/hp Element Methods for CFD", Oxford University Press, 1999.

Shape Functions

The reference element shape functions on $(-1, 1)$ of order p [3]:

$$N_i(\xi) = \begin{cases} \frac{1-\xi}{2} & i = 0 \\ \frac{1-\xi}{2}\frac{1+\xi}{2}P_{i-1}^{1,1}(\xi) & 1 \leq i \leq p-1 \\ \frac{1+\xi}{2} & i = p \end{cases}$$



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich