
$\boxed{25}$ Let $\mathcal{T}_h$ a mesh of $(0,1)$ with maximal mesh size h, let V_h be the corresponding finite element space (using the Courant element), and let $I_h : H^1(0,1) \to V_h$ the corresponding interpolation operator. Show the interpolation error estimate

$$|v - I_h v|_{H^1(0,1)} \;\leq\; C_1\, h\, \|v''\|_{L^2(0,1)} \forall v \in C^2(0,1).$$

Hint: Analogous to the proof of the L^2-estimate in the lecture: split into element terms, transform to the reference element, estimate the interpolation error on the reference element, transform back.

$\boxed{26^*}$ (BONUS exercise) Let $V_0 := \{v \in H^1(0,1) : v(0) = 0\}$ and let $a : V_0 \times V_0 \to \mathbb{R}$ and $F \in V_0^*$ fulfill the assumptions of the Lax-Milgram theorem. Let V_h be the Courant FE space as in Exercise $\boxed{25}$ and $V_{0h} = V_h \cap V_0$. Furthermore, let $u \in V_0$ be the solutions of the variational formulation and $u_h \in V_{0h}$ the Galerkin-FE solution. Show that for a sequence of meshes $\{\mathcal{T}_h\}$ where $h \to 0$, we have

$$\|u - u_h\|_{H^1(0,1)} \;\to\; 0,$$

even if $u \notin H^2(0,1)$. *Hint:* Use Céa's lemma and the fact that H^2 is dense in H^1.

$\boxed{27}$ Derive the variational formulation of the d-dimensional model problem: $\Omega \subset \mathbb{R}^d$ Lipschitz domain, $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$, find $u : \overline{\Omega} \to \mathbb{R}$ such that

$$\begin{aligned} -\mathrm{div}\big(A(x)\,\nabla u(x)\big) + \vec{b}(x)\cdot\nabla u(x) + c(x)\,u(x) &= f(x) \qquad \forall x \in \Omega, \\ u(x) &= g_D(x) \qquad \forall x \in \Gamma_D\,, \\ \big(A(x)\,\nabla u(x)\big)\cdot\vec{n}(x) &= g_N(x) \qquad \forall x \in \Gamma_N\,, \end{aligned}$$

where the scalar functions f, g_D, g_N, and the coefficients $A(x) \in \mathbb{R}^{d\times d}$, $\vec{b}(x) \in \mathbb{R}^d$, and $c(x) \in \mathbb{R}$ are given.
Specify V_g, V_0, $a(\cdot,\cdot)$, and $\langle F, \cdot\rangle$.

$\boxed{28}$ Consider the variational formulation of the d-dimensional model problem. Assume that $\mathrm{meas}_{d-1}(\Gamma_D) > 0$, that $f \in L^2(\Omega)$, $g_N \in L^2(\Gamma_N)$, and $g_D \in H^{1/2}(\Gamma_D)$, which means that $g_D \in L^2(\Gamma_D)$ and there exists $g \in H^1(\Omega) : \gamma_D g = g_D$. Show that $a(\cdot,\cdot)$ and $\langle F,\cdot\rangle$ are $H^1(\Omega)$-bounded and that $a(\cdot,\cdot)$ is V_0-coercive.

Programming.
In Tutorial 4 we assembled the stiffness matrix and load vector corresponding to the pure Neumann problem with homogeneous Neumann boundary conditions. Here we consider other types of boundary conditions and will solve the FE system.

$\boxed{29}$ Write a function

```
void ImplementRobinBC (int i, double g, double alpha,
                       SMatrix& mat, Vector& vec);
```

to implement the Robin boundary condition

$$\begin{aligned} -u'(0) + \alpha\, u(0) &= g_1 \qquad \text{if } i = 0, \\ u'(1) + \alpha\, u(1) &= g_1 \qquad \text{if } i = n_h\,, \end{aligned}$$

for given values `g`$=g_1$, `alpha`$=\alpha$ at the boundary node x_i identified by the index `i`$=i \in \{0, n_h\}$. Compare with Exercise $\boxed{02}$ (Tutorial 1).

`ImplementRobinBC` must update (modify) the stiffness matrix `mat` and the load vector `vec` which were previously computed by `AssembleStiffnessMatrix` and `AssembleLoadVector`.

$\boxed{30}$ Write a function

```
void ImplementDirichletBC (int i, double g, SMatrix& mat,
                                          Vector& vec);
```

which implements the Dirichlet boundary condition

$$u(x_i) \;=\; g_0$$

for a given value `g`$=g$ at the boundary node x_i identified by the index `i`$=i$. The function `ImplementDirichletBC` must update the stiffness matrix `mat` and the load vector `vec`, after having applied `AssembleStiffnessMatrix`, `AssembleLoadVector`, and `ImplementRobinBC`.

Instead of *deleting* rows or columns from the matrix, you should stay with the $(n_h + 1) \times (n_h + 1)$ matrix using the following "decoupling" technique:
For $i = 0$, the resulting system should look like

$$\begin{pmatrix} K_{00} & 0 & \dots & 0 \\ 0 & K_{11} & \dots & K_{1n_h} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & K_{n_h1} & \dots & K_{n_hn_h} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n_h} \end{pmatrix} = \begin{pmatrix} K_{00}\, g_0 \\ f_1 - K_{10}\, g_0 \\ \vdots \\ f_{n_h} - K_{n_h0}\, g_0 \end{pmatrix},$$

where $[K_{ij}]_{i,j=0}^{n_h}$ and $[f_i]_{i=0}^{n_h}$ are the stiffness matrix and load vector *before* the call.

$\boxed{31}$ Solve the system $K_h\, \underline{u}_h = \underline{f}_h$ in optimal complexity using Gaussian elimination exploiting the tridiagonal structure (and maybe also the symmetry).
Hint: If you are lazy, take your inspiration from
`http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`

Use this algorithm to solve the boundary value problem

$$\begin{aligned} -u''(x) &= f(x), \\ u(0) &= g_0\,, \\ u'(1) &= g_1\,, \end{aligned}$$

with $f(x) = 2\,x + 1$, $g_0 = 3$, and $g_1 = 0.5$.