10 Show *Poincaré's inequality*: There exists a constant $C_P > 0$ such that

$$\|v\|_{L_2(0,1)} \;\le\; C_P\Big\{\Big(\int_0^1 v(x)\,dx\Big)^2 + |v|^2_{H^1(0,1)}\Big\}^{1/2} \qquad \forall v \in H^1(0,1)\,.$$

Hint: Integrate the identity

$$v(y) \;=\; v(x) + \int_x^y v'(z)\,dz$$

over the whole interval (0, 1) with respect to x. The rest of the proof is then similar to the one of Friedrichs' inequality (see your lecture notes).

11 Take a look at exercise **06** on the pure Neumann problem and show that the weak formulation (2.4) has a solution if and only if (2.5) holds, and that the solution is unique up to an additive constant.
Hint: Use Poincaré's inequality to show the coercivity of $a(w, v)$ on $\widehat{V}$.

12 Let V be a Hilbert space, $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ a symmetric bilinear form satisfying $a(v, v) \ge 0$ for all $v \in V$, and $F \in V^*$ with $V_0 \subset V$. Show directly that the variational formulation

$$\text{find } u \in V_g: \quad a(u, v) = \langle F, v\rangle \quad \forall v \in V_0$$

with $V_g = g + V_0$ is equivalent to the minimization problem

$$J(u) = \inf_{v\in V_g} J(v) \qquad \text{with} \qquad J(v) = \frac{1}{2}a(v, v) - \langle F, v\rangle\,.$$

Hint: Modify the corresponding proof from your lecture notes, where the special case $a(u, v) = (u, v)_V$ with $V_0 = V_g = V$ is treated.

Programming

Let $\Omega = (0, 1)$, $\Gamma_D = \{0\}$, and $\Gamma_R = \{1\}$. Consider the following one-dimensional boundary value problem: Find $u(x)$ such that

$$\begin{aligned} -u''(x) &= f(x) && \text{for } x \in \Omega\,,\\ u(x) &= g_D(x) && \text{for } x \in \Gamma_D\,,\\ u'(x) &= \alpha(x)\big(g_R(x) - u(x)\big) && \text{for } x \in \Gamma_R\,. \end{aligned} \tag{3.1}$$

We discretize this problem using the FEM with Courant elements. Consider the nodes $0 = x_0 < x_1 < \cdots < x_{n_h} = 1$ which define a mesh (subdivision) $\mathcal{T}_h$ of Ω with the elements $T_k = (x_{k-1}, x_k)$, $k = 1, \ldots, n_h$. We introduce the finite element space

$$V^h := \{v_h \in C(\overline{\Omega}) : v_{h|T} \in P_1 \text{ for all } T \in \mathcal{T}_h\}$$

whose basis is given by the nodal basis functions φ_i, $i = 0, \ldots, n_h$, defined by

$$\varphi_i(x_j) = \delta_{ij} \quad \text{for } i, j = 0, \ldots, n_h\,.$$

13 Write a function `ElementStiffnessMatrix(↓xa, ↓xb, ↑element_matrix)` which for given nodes `xa` $= x_{k-1}$ and `xb` $= x_k$ returns the element stiffness matrix `element_matrix` $= K_h^{(k)}$ on the element T_k, defined by

$$K_h^{(k)} = \begin{bmatrix} \int_{T_k} \left(\varphi'_{k-1}(x)\right)^2 dx & \int_{T_k} \varphi'_{k-1}(x)\,\varphi'_k(x)\,dx \\ \int_{T_k} \varphi'_k(x)\,\varphi'_{k-1}(x)\,dx & \int_{T_k} \left(\varphi'_k(x)\right)^2 dx \end{bmatrix} \qquad \text{for } k = 1, \ldots, n_h\,.$$

Hint: You can use the type `typedef double Mat22[2][2];` to represent a two-by-two matrix.

14 Write a function `ElementLoadVector(↓(*f)(x), ↓xa, ↓xb, ↑element_vector)` which for a given function `f` $= f \in C[0,1]$ and the nodes `xa` $= x_{k-1}$ and `xb` $= x_k$ returns the 2-dimensional element load vector `element_vector` $= f_h^{(k)}$ on the element T_k, defined by

$$f_h^{(k)} = \begin{pmatrix} \int_{T_k} f(x)\,\varphi_{k-1}(x)\,dx \\ \int_{T_k} f(x)\,\varphi_k(x)\,dx \end{pmatrix} \qquad \text{for } k = 1, \ldots, n_h\,.$$

Use the trapezoidal rule to approximate above integrals:

$$\int_a^b g(x)\,dx \;\simeq\; \frac{b-a}{2}\big[g(a) + g(b)\big]\,.$$

Hint: You can use the following types and function header:

```
typedef double (*RealFunction)(double x);
typedef double Vec2[2];
void ElementLoadVector (RealFunction f, double xa, double xb, Vec2& element_vector);
```

15 Define an efficient data type `Matrix` for the stiffness matrix K_h exploiting the fact that it is tridiagonal. Make sure that your data type allows access to the matrix entries.

Provide your solution on a USB stick or send it by e-mail before Monday 9.45 a.m.