---

25 Gegeben sei ein Hilbertraum V mit einem abgeschlossenen Teilraum $V_h \subset V$. Für ein $u \in V$ zeige man, dass die folgenden Problemstellungen äquivalent sind

a) Gesucht ist $u_h \in V_h$, sodass

$$\|u - u_h\|_V = \inf_{v_h \in V_h} \|u - v_h\|_V .$$

b) Gesucht ist $u_h \in V_h$, sodass

$$(u_h, v_h)_V = (u, v_h)_V \qquad \text{für alle } v_h \in V_h .$$

26 Man konstruiere quatratische Basisfunktionen $\hat{\varphi}_0, \hat{\varphi}_1, \hat{\varphi}_2 \in \mathbb{P}_2(\hat{T})$ für das Referenzelement $\hat{T} = (0, 1)$, sodass für die Punkte $\xi_0 = 0$, $\xi_1 = \frac{1}{2}$ und $\xi_2 = 1$

$$\hat{\varphi}_i(\xi_j) = \delta_{ij} = \begin{cases} 1 & \text{für } i = j, \\ 0 & \text{für } i \neq j \end{cases}$$

für alle $i, j = 0, 1, 2$ gilt. Für diese Basisfunktionen berechne man die Element-Steifigkeitsmatrix am Referenzelement $\hat{T}$

$$\hat{K}_h := \int_{\hat{T}} \underline{\phi}'(\xi) \underline{\phi}'(\xi)^\top \mathrm{d}\xi,$$

wobei $\underline{\phi} := (\hat{\varphi}_0, \hat{\varphi}_1, \hat{\varphi}_2)^\top$ der Vektor der quadratischen Formfunktionen ist.

Programmierteil.

---

27 Man erstelle eine Funktion

```
void ElementStiffnessMatrix(double xa, double xb, Mat22& elMat);
```

die für ein gegebenes Element $T_K = (x_{k-1}, x_k) = (\mathtt{xa}, \mathtt{xb})$ einer Zerlegung die Element-Steifigkeitsmatrix

$$K_h^{(k)} = \int_{T_k} \begin{pmatrix} \varphi'_{k-1}(x) \\ \varphi'_k(x) \end{pmatrix} \begin{pmatrix} \varphi'_{k-1}(x) \\ \varphi'_k(x) \end{pmatrix}^\top \mathrm{d}x = \frac{1}{h_k} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

für stückweise lineare und stetige Funktionen berechnet und in dem Matrixtyp `elMat` speichert. Weiters implementiere man eine Methode `void TestElementMatrix()` welche eine Element-Steifigkeitsmatrix berechnet und die berechnete Werte in der Konsole ausgibt.

28 Man implementiere eine Funktion

```
void ElementLoadVector(RealFunction f, double xa, double xb,
                       Vec2& elVec);
```

welche für eine gegebene Funktion $\texttt{f} = f \in \mathcal{C}[0,1]$ und für ein gegebenes Element $T_K = (x_{k-1}, x_k) = (\texttt{xa}, \texttt{xb})$ eine Approximation des Element-Lastenvektors

$$\underline{f}_h^{(k)} = \int_{T_k} f(x) \begin{pmatrix} \varphi_{k-1}(x) \\ \varphi_k(x) \end{pmatrix} \mathrm{d}x$$

mit Hilfe der Trapezregel berechnet und im Datentyp `elVec` speichert. Weiters implementiere man eine Methode `void TestElementVector()` welche für ein gegebenes Element den Element-Lastenvektor berechnet und die berechneten Werte in der Konsole ausgibt.

$\boxed{29}$ Man implementiere eine Routine, die effizient ein lineares Gleichungssystem der Form

$$K_h \underline{u}_h = \underline{f}_h$$

löst. Dabei ist $K_h \in \mathbb{R}^{n_h \times n_h}$ eine Tridiagonalmatrix, also vom Typ `SMatrix` und $\underline{f}_h \in \mathbb{R}^{n_h}$ eine gegebener Vektor.

Man teste diese Routine für ein geeignetes Gleichungssystem und überprüfe die Richtigkeit der berechneten Lösung $\underline{u}_h \in \mathbb{R}^{n_h}$.