

- [28] Gegeben sei das lineare Gleichungssystem $A\underline{x} = \underline{f}$ mit einer invertierbaren, tridiagonalen Matrix

$$A = \begin{pmatrix} d_1 & c_1 & & & 0 \\ a_1 & d_2 & c_2 & & \\ & a_2 & d_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_{n-1} & d_n \end{pmatrix} \quad \text{und} \quad \underline{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}.$$

Für dieses Gleichungssystem leite man einen effizienten Gauß-Algorithmus her. Dazu zeige man, dass das gegebene Gleichungssystem äquivalent ist zu $\tilde{A}\underline{x} = \tilde{\underline{f}}$ mit

$$\tilde{A} = \begin{pmatrix} 1 & \tilde{c}_1 & & & 0 \\ & 1 & \tilde{c}_2 & & \\ & & 1 & \ddots & \\ & & & \ddots & \tilde{c}_{n-1} \\ 0 & & & & 1 \end{pmatrix} \quad \text{und} \quad \tilde{\underline{f}} = \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_n \end{pmatrix}.$$

Danach löse man das äquivalente Gleichungssystem $\tilde{A}\underline{x} = \tilde{\underline{f}}$.

- [29] Analog zu Lemma 1.56 zeige man für den Interpolationsoperator $I_h : H^1(0, 1) \rightarrow V_h$ die Fehlerabschätzung

$$|u - I_h(u)|_{H^1(0,1)}^2 \leq \sum_{k=1}^{n_h} h_k^2 \int_{T_k} |u''(x)|^2 dx$$

für alle $u \in \mathcal{C}^2[0, 1]$.

Programmierteil.

In der letzten Übung wurde die Aufstellung der Elementmatrizen und der Elementvektoren implementiert. In dieser Übung soll die globale Steifigkeitsmatrix und der globale Lastenvektor für das reine Neumann-Randwertproblem assembliert werden. Anschließend soll die Steifigkeitsmatrix und der Lastenvektor für das reine Neuman-Randwertproblem modiviziert werden, sodass auch Dirichlet- bzw. Robinrandbedingungen eingebaut werden können.

- [30] Man schreibe eine Assembler-Routine, die die globale Steifigkeitsmatrix

$$K_h = [K_{ij}]_{i,j=0}^{n_h} \in \mathbb{R}^{(n_h+1) \times (n_h+1)} \quad \text{mit} \quad K_{ij} = \int_a^b \varphi'_j(x) \varphi'_i(x) dx, \quad i, j = 0, \dots, n_h$$

für das reine Neumann-Randwertproblem und dem Rechengebiet (a, b) assembliert. Dazu berechne man für jedes Element einer gegebenen Zerlegung die lokale Steifigkeitsmatrix und assembliere diese in eine tridiagonale Matrix vom Typ **SMatrix**.

Man teste diese Routine für eine gegebene Triangulierung und gebe die berechnete Steifigkeitsmatrix in der Konsole aus.

- 31 Man schreibe eine Assemblier-Routine, die den globalen Lastenvektor

$$\underline{f}_h = [f_i]_{i=0}^{n_h} \in \mathbb{R}^{n_h+1} \quad \text{mit} \quad f_i = \int_a^b f(x) \varphi_i(x) dx, \quad i = 0, \dots, n_h$$

für eine gegebene Funktion $f : [a, b] \rightarrow \mathbb{R}$ assembliert. Dazu berechne man für jedes Element einer gegebenen Zerlegung den approximierten lokalen Elementvektor und assembliere diesen in den globalen Lastenvektor.

Man teste diese Routine für eine gegebene Triangulierung und gebe den berechneten Lastenvektor in der Konsole aus.

- 32 Man schreibe eine Routine, die die Steifigkeitsmatrix von Aufgabe 30 bzw. den Lastenvektor von Aufgabe 31 so modifiziert, sodass anstelle von Neuman-Randbedingungen auch Robin-Randbedingungen vorgegeben werden können, also

$$-u'(a) + \alpha_0 u(a) = g_0 \quad \text{oder} \quad u'(b) + \alpha_1 u(b) = g_1.$$

Man berechne mit dieser Routine die modifizierte Steifigkeitsmatrix bzw. den modifizierten Lastenvektor für einen geeigneten Testfall und gebe diese in der Konsole aus.

- 33 Man schreibe eine Routine, die die Steifigkeitsmatrix von Aufgabe 30 bzw. den Lastenvektor von Aufgabe 31 so modifiziert, sodass anstelle von Neuman-Randbedingungen auch Dirichlet-Randbedingungen vorgegeben werden können, also

$$u(a) = g_0 \quad \text{oder} \quad u(b) = g_1.$$

Man berechne mit dieser Routine die modifizierte Steifigkeitsmatrix bzw. den modifizierten Lastenvektor für einen geeigneten Testfall und gebe diese in der Konsole aus.