

## 6. Übungsblatt (erschienen am 15.01.2020)

### Aufgabe 6.1 (Schriftliche Aufgabe)

Sei  $H$  Hilbertraum,  $b : H \times H \rightarrow \mathbb{R}$  und  $l : H \rightarrow \mathbb{R}$  seien eine Bilinear- und eine Linearform, welche die Voraussetzungen von Satz 2.41 (Lax-Milgram) erfüllen (mit  $\beta > 0$  und  $C > 0$ ). Seien  $u, v \in H$ .

- (a) Die durch das Skalarprodukt  $\langle u, v \rangle_b := b(u, v)$  induzierte Norm  $\|u\|_b := \sqrt{\langle u, u \rangle_b}$  heißt *Energienorm*. Zeigen Sie, dass diese äquivalent zur  $H$ -Norm ist

$$\sqrt{\beta} \|u\| \leq \|u\|_b \leq \sqrt{C} \|u\| \quad \forall u \in H.$$

- (b) Sei  $\bar{u} \in H$  die eindeutige Lösung von  $b(\bar{u}, v) = l(v)$  für alle  $v \in H$ . Weiterhin sei  $V$  ein endlichdimensionaler abgeschlossener Teilraum von  $H$  versehen mit  $\langle \cdot, \cdot \rangle_b$  bzw.  $\|\cdot\|_b$ . Dann bezeichnet  $\tilde{u}$  die *Galerkin-Projektion* von  $\bar{u}$  auf  $V$ . Zeigen Sie folgende Variante des Céa-Lemmas:

- (i)  $\|\bar{u} - \tilde{u}\|_b = \inf_{v \in V} \|\bar{u} - v\|_b$ .  
 (ii)  $\|\bar{u} - \tilde{u}\| \leq \sqrt{\frac{C}{\beta}} \inf_{v \in V} \|\bar{u} - v\|$ .

### Aufgabe 6.2 (Votieraufgabe)

Wir übernehmen die Notation von Übungsblatt 5 und insbesondere von Aufgabe 5.2. Für  $u, \tilde{u}, v \in V^T$ ,  $D \subset \Omega$  und  $\tau \in \mathbb{R}$  seien die Bilinearform und die Linearform

$$b(u, v) := \int_{\Omega} \tau \nabla u(x) \cdot \nabla v(x) + u(x)v(x) \, dx \quad f(v) := \int_{\Omega} \tau \chi_D(x)v(x) + \tilde{u}(x)v(x) \, dx$$

gegeben. Dabei gilt wegen  $\tilde{u} \in V^T$ , dass auf jedem Dreieck  $T_k$  die Funktion  $\tilde{u}$  eindeutig durch die drei Koeffizienten  $\tilde{\mathbf{u}}_a, \tilde{\mathbf{u}}_b, \tilde{\mathbf{u}}_c \in \mathbb{R}$  gegeben ist durch  $\tilde{u}(x) = \tilde{\mathbf{u}}_a \Lambda_a(x) + \tilde{\mathbf{u}}_b \Lambda_b(x) + \tilde{\mathbf{u}}_c \Lambda_c(x)$ . Bestimmen Sie wieder die *Elementsteifigkeitsmatrix*

$$B_{T_k} := \begin{pmatrix} b_{T_k}(\Lambda_a, \Lambda_a) & b_{T_k}(\Lambda_a, \Lambda_b) & b_{T_k}(\Lambda_a, \Lambda_c) \\ b_{T_k}(\Lambda_b, \Lambda_a) & b_{T_k}(\Lambda_b, \Lambda_b) & b_{T_k}(\Lambda_b, \Lambda_c) \\ b_{T_k}(\Lambda_c, \Lambda_a) & b_{T_k}(\Lambda_c, \Lambda_b) & b_{T_k}(\Lambda_c, \Lambda_c) \end{pmatrix}$$

sowie  $f_{T_k}(\Lambda_a)$ ,  $f_{T_k}(\Lambda_b)$  und  $f_{T_k}(\Lambda_c)$ . Dabei dürfen Sie bei  $f_{T_k}(\Lambda_a)$ ,  $f_{T_k}(\Lambda_b)$  und  $f_{T_k}(\Lambda_c)$  davon ausgehen, dass  $T_k$  entweder komplett im Teilgebiet  $D$  oder gar nicht in  $D$  enthalten ist.

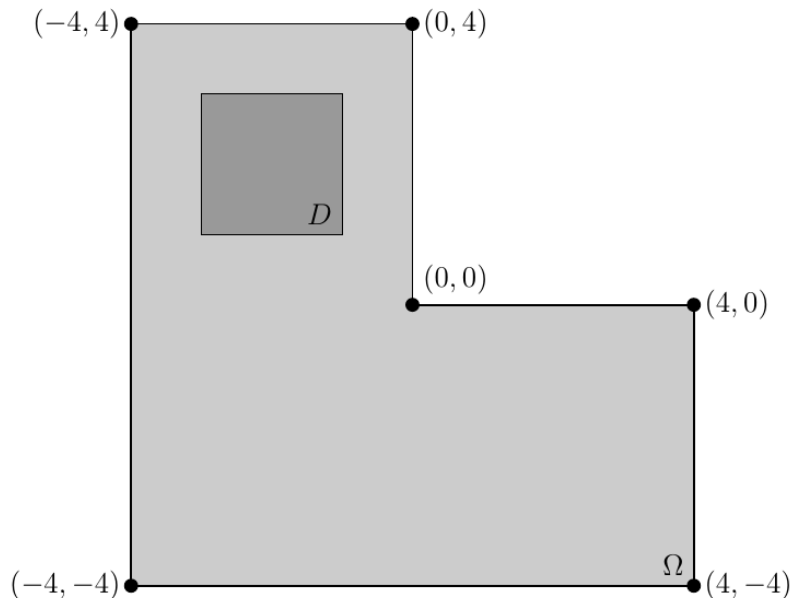
### Aufgabe 6.3 (Programmieraufgabe)

Wir betrachten das Gebiet

$$\Omega = ((-4, 4) \times (-4, 4)) \cap \{(x, y) \in \mathbb{R}^2 : x < 0 \text{ oder } y < 0\}$$

mit dem Teilgebiet

$$D = [-3, -1] \times [1, 3].$$



Wir betrachten die *parabolische* partielle Differentialgleichung

$$u_t - \Delta u = \chi_D, \quad u|_{\partial\Omega} = 0, \quad u(x, 0) = 0. \quad (1)$$

und wollen die *Rothe-Methode* (oder *horizontale Linienmethode*) verwenden, um eine Approximation an die wahre Lösung zu erhalten. Dabei wird (1) bezüglich der Zeit diskretisiert und es ist in jedem Zeitschritt ein elliptisches Randwertproblem zu lösen. Für eine feste Zeitschrittweite  $\tau \in \mathbb{R}$  sei also  $t_i := i\tau$ ,  $i = 1, 2, \dots$  und  $u_i(x) \approx u(x, t_i)$  bezeichne die Approximation an die wahre Lösung zum Zeitpunkt  $t_i$ .

Wir verwenden das implizite Euler-Verfahren um  $u_i(x)$  aus  $u_{i-1}(x)$  zu bestimmen, d.h.

$$u_i(x) - \tau \Delta u_i(x) = u_{i-1}(x) + \tau \chi_D(x), \quad x \in \Omega, \quad u_i(x) = 0, \quad x \in \partial\Omega.$$

Schreiben Sie eine **MATLAB**-Funktion, welche mit Hilfe der Rothe-Methode und den zweidimensionalen linearen finiten Elementen, welche auf dem 5. Übungsblatt behandelt wurden, eine Approximation an die wahre Lösung von (1) bestimmt (verwenden Sie beispielsweise  $\tau = \frac{1}{10}$  und berechnen Sie die Approximationen der ersten 1000 Zeitschritte). Visualisieren Sie ihr Ergebnis.

*Hinweis:* Falls Sie  $\Omega$  mit dem `distmesh`-Paket triangulieren wollen, so können Sie das mit Hilfe der `dpoly`-Funktion realisieren (vgl. drittes Beispiel auf <http://persson.berkeley.edu/distmesh/>). Überlegen Sie sich, wie Sie mit Hilfe der `dpoly`-Funktion auch die Indizes der Randknoten bestimmen können.

## Hinweise zur Übungsblattbearbeitung:

- Zu **schriftlichen Aufgaben** soll eine Ausarbeitung/Lösung angefertigt werden, die bis zum 27.01.2020 um 12:00 Uhr in dem Postkasten Ihrer Übungsleiterin (Nummer 17) im 3. Stock der Robert-Mayer-Str. 6-8 abzugeben ist. Sollte ein Übungstermin nicht wahrgenommen werden können, so kann die Abgabe der schriftlichen Aufgabe auch bis zum obigen Zeitpunkt an Ihre Übungsleiterin geschickt werden.
- Zu **Programmieraufgaben** soll bis zum 27.01.2020 um 12:00 Uhr eine **kommentierte** Ausarbeitung in MATLAB-Code an Ihre Übungsleiterin geschickt werden. Bitte beginnen Sie die Betreffzeile Ihrer E-Mail mit "**PDGL6\_2020\_Gruppennummer:**" (wenn Sie z.B. in Gruppe 2 sind, so soll die Betreffzeile mit "PDGL6\_2020\_2:" beginnen).
- Zu **Votieraufgaben** wird keine schriftliche Abgabe verlangt. Die Lösung wird in der Übung besprochen.
- Alle Aufgaben von Übungsblatt 6 werden in den Übungen am 28.01.2020 besprochen.