

### 3. Übung zur Numerik partieller Differentialgleichungen II

**Hinweis:** Schreiben Sie bitte jede Aufgabe auf ein neues Blatt und auf **jedes Blatt ihren Namen**. Auf die erste Seite Ihrer Übung schreiben Sie bitte zusätzlich Ihre Matrikelnummer.

#### Aufgabe 1 (6 Punkte):

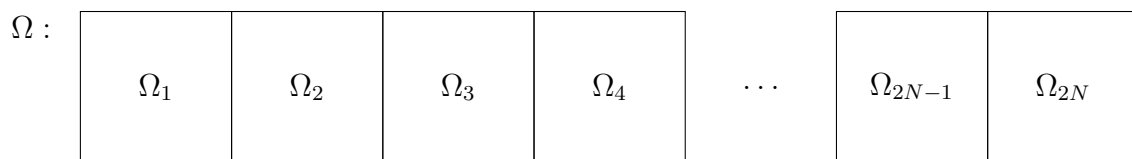
Sei  $A$  eine symmetrische, positiv definite Matrix,  $\alpha = \alpha_{\text{opt}}$  und  $R_\alpha = I - \alpha A$ . Zeigen Sie: Für das nicht vorkonditionierte stationäre Richardson Verfahren gilt

$$\|e^{(k+1)}\|_A \leq \rho(R_\alpha) \|e^{(k)}\|_A, \quad k \geq 0.$$

*Hinweis:* Begründen Sie warum  $A^{-1/2}$  mit  $A^{-1} = A^{-1/2} A^{-1/2}$  existiert.

#### Aufgabe 2 (3 Punkte):

Betrachten Sie den Querschnitt eines Stabes  $\bar{\Omega} = [0, 10] \times [0, 1]$ . Zerlegen Sie  $\Omega$  in  $2N$  viele Teilgebiete  $\Omega_1, \dots, \Omega_{2N}$ .



Verallgemeinern Sie die diskrete Beschreibung des Dirichlet-Neumann-Algorithmus' aus der Vorlesung für zwei Teilgebiete (siehe (1.9)) auf den Fall von  $2N$  Teilgebieten, bei denen Dirichlet-Randwertprobleme auf  $\Omega_i$  mit  $i = 1, 3, \dots$  und Neumann-Randwertprobleme auf  $\Omega_i$  mit  $i = 2, 4, \dots$  gelöst werden.

**Programmieraufgabe (18 Punkte):** Betrachten Sie  $\Omega$  und eine Zerlegung in Teilgebiete wie in Aufgabe 2.

Schreiben Sie einen Dirichlet-Neumann-Gebietszerlegungs-Algorithmus in Matlab oder C/C++ (dabei darf PETSc verwendet werden), um folgendes Randwertproblem zu lösen

$$\begin{aligned} -\Delta u &= 1 & \text{in } \Omega &= (0, 10) \times (0, 1), \\ u &= 0 & \text{auf } \partial\Omega. \end{aligned}$$

Verwenden Sie dazu eine Finite Elemente Diskretisierung der Schrittweite  $h = 1/16$  auf  $\Omega$ . Lösen Sie auf Teilgebieten mit ungeradem Index ein Dirichlet-Problem und auf Teilgebieten mit geradem Index ein Neumann-Problem.

Gehen Sie dazu wie folgt vor:

1. Legen Sie ein m-File **dirichletneumann.m** an, dessen Übergabewerte **n=1/h** und **N** sind.

2. Verwenden Sie Matlab-Cell-Arrays wie
 

```
elements = cell(1,2*N);
for i=1:2*N
elements{i} = ...
end
```

 usw., um Punkt-, Elementlisten, Steifigkeitsmatrizen, Lastvektoren etc. pro Teilgebiet  $\Omega_i$ ,  $i = 1, \dots, 2N$  speichern zu können.
3. Verwenden Sie ihre Module zur Erstellung der Geometrie, zur Auswertung der Basisfunktionen, der rechten Seite, der Erstellung der lokalen Steifigkeitsmatrix und des lokalen Lastvektors aus Übung 1. Unterscheiden Sie für einen Punkt auf  $\Gamma := \{x \in \partial\Omega_i \cap \partial\Omega_j \mid 1 \leq i, j \leq 2N, x \notin \partial\Omega\}$  ob er links oder rechts vom Teilgebiet auf  $\Gamma$  liegt. Das heißt: Verwenden Sie zwei verschiedene Abbildungen von der gesamten lokalen Punktmenge auf Teilmengen  $\Gamma_{i,\text{links}} := \{x \in \partial\Omega_i \cap \partial\Omega_{i-1} \mid x \notin \partial\Omega\}$  und  $\Gamma_{i,\text{rechts}} := \{x \in \partial\Omega_i \cap \partial\Omega_{i+1} \mid x \notin \partial\Omega\}$ . Aus diesen Beiden kann auch eine Abbildung auf das gesamte lokale Interface  $\Gamma_i = \partial\Omega_i \cap \Gamma$ ,  $1 \leq i \leq 2N$  erstellt werden.
4. Wählen Sie  $u_\Gamma^0 = 0$ , eine maximale Iterationszahl von 100 und  $N = 4$ .
5. Berechnen Sie nun für die Teilgebiete  $\Omega_1, \Omega_3, \Omega_5, \dots$  die Dirichlet-Randwertprobleme mit entsprechen Informationen  $u_{\Gamma_i}^n$ .
6. Berechnen Sie nun für die Teilgebiete  $\Omega_2, \Omega_4, \Omega_6, \dots$  die Neumann-Randwertprobleme mit entsprechen Fluss-Informationen aus den Dirichlet-Problemen.
7. Berechnen Sie für jedes Teilgebiet sowohl das Residuum, welches im Vergleich von  $u_i^n$  mit den Werten  $u_{i-1}^n$  und  $u_{i+1}^n$  auf  $\Gamma_i$  entsteht, d.h.

$$r_{\Gamma,u}^{(i)} := \begin{pmatrix} r_{\Gamma_{i,\text{links}},u}^{(i)} \\ r_{\Gamma_{i,\text{rechts}},u}^{(i)} \end{pmatrix} := \begin{pmatrix} (u_i^n - u_{i-1}^n)|_{\Gamma_{i,\text{links}}} \\ (u_i^n - u_{i+1}^n)|_{\Gamma_{i,\text{rechts}}} \end{pmatrix}$$

als auch den Unterschied des Flusses wie in Übung 1 durch  $r_\Gamma$  ausgedrückt; hier entsprechend:  $r_{\Gamma,\lambda}^{(i)} := (r_{\Gamma_{i,\text{links}},\lambda}^{(i)}, r_{\Gamma_{i,\text{rechts}},\lambda}^{(i)})^T$ . Bestimmen Sie über alle Teilgebiete das Maximum über diese beiden Residuen. Brechen Sie die Iteration ab, sobald dieses Maximum über alle Teilgebiete kleiner als TOL ist. Anderenfalls beginnen Sie wieder bei 5.

8. Geben Sie gegen Ende des Programms die Anzahl Iterationen und das maximale Residuum im letzten Schritt für  $\text{TOL} = 1e - 10$  und  $\theta = 0.3$  aus. Plotten Sie den Verlauf des maximalen Residuums über die Iterationen sowie die berechnete Lösung  $u$ , die sich aus  $u_i = u\{i\}$ ,  $i = 1, \dots, 2N$  auf  $\Omega$  zusammensetzt.

### Abgabe des Programmiererteils

- Das ausführbare Programm (Programmcode inkl. **Startdatei**) bitte an Lara Gutberlet (lgutber1@smail.uni-koeln.de) mit Betreff der Form: Uebung1, Nachname, Vorname schicken. **Unkommentierter Programmcode wird nicht angenommen!**
- Packen Sie Ihre Dateien in ein Archiv (Formate: .zip, oder .tar.gz) mit Dateinamen der Form ueb01\_nachname\_vorname.zip.
- Geben Sie bitte immer eine **ausgedruckte Version** Ihres Programmcodes ab, falls dies in der Aufgabenstellung nicht anders vermerkt wurde.

**Abgabedatum: 11. November 2016 bis 12:00 Uhr. Im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts.**