

Prof. Dr. A. Klawonn
 J. Knepper, M. Sc.
 J. Weber, M. Sc.

24. Oktober 2018

3. Übung zu Wissenschaftliches Rechnen I

Aufgabe 1: (4 Punkte)

Um das System $S^{(j)}u_\Gamma^{(j)} = w_\Gamma^{(j)}$ auf dem Interface des Teilgebiets Ω_j zu lösen, würde man im Allgemeinen nicht das Schurkomplement $S^{(j)}$ explizit aufstellen und invertieren, da dies in der Praxis zu teuer im Hinblick auf Rechenzeit und Speicherressourcen ist.

Zeigen Sie, dass die Lösung des Systems

$$u_\Gamma^{(j)} = S^{(j)-1} w_\Gamma^{(j)}$$

alternativ durch Lösen eines geeigneten lokalen Systems auf Ω_j gefunden werden kann. Geben Sie dazu das entsprechende lokale System an und zeigen Sie, dass die Lösungen der beiden Systeme dieselben sind.

Aufgabe 2: (5 Punkte, 6 Bonuspunkte)

Sei $(\tau_h)_h$ eine reguläre Familie von Triangulierungen eines polygonalen Lipschitzgebiets $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, sowie $(V^h)_h$ die zugehörige affine Familie von Finite-Elemente-Räumen, die stückweise aus Polynomen vom Grad k bestehen. Sei $V^h = \text{span}(\varphi_1, \dots, \varphi_n)$, wobei φ_i die nodalen Basisfunktionen sind. Sei zudem $V^h \subset V := H_0^2(\Omega)$ und $h := \max_{T \in \tau_h} h_T$ mit $h_T = \text{diam}(T)$. Wir betrachten das Modellproblem

$$\begin{aligned} \Delta^2 u &= f \text{ in } \Omega, \\ \frac{\partial u}{\partial \nu} &= 0 \text{ auf } \partial\Omega, \\ u &= 0 \text{ auf } \partial\Omega. \end{aligned}$$

- i) (**Bonusaufgabe**) Zeigen Sie, dass die zugehörige Bilinearform $a(\cdot, \cdot)$ stetig und V -elliptisch ist.

Hinweise:

- Es gilt $\Delta^2 u = \Delta(\Delta u)$ mit $\Delta u = \sum_{i=1}^d (D_i^2 u) = \sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2}$.
- Benutzen Sie für die Herleitung der schwachen Formulierung die folgende Green'sche Formel: Für $u, v \in H^4(\Omega)$ gilt

$$\int_\Omega \Delta(\Delta u) \cdot v - \Delta u \cdot \Delta v \, dx = \int_{\partial\Omega} \left(v \frac{\partial}{\partial \nu} \Delta u - \Delta u \frac{\partial v}{\partial \nu} \right) d\sigma.$$

- Beachten Sie, dass die (schwachen) Ableitungen (1. Ordnung) von Funktionen in $H_0^2(\Omega)$ auf dem Rand Null sind.

- Zum Nachweis der Stetigkeit und V -Elliptizität von $a(\cdot, \cdot)$ zeigen Sie zunächst, dass $|u|_{H^2(\Omega)}^2 = \|\Delta u\|_{L^2(\Omega)}^2$ auf $H_0^2(\Omega)$ erfüllt ist. Wenden Sie dazu zwei mal auf $|u|_{H^2(\Omega)}^2 = \sum_{i,j=1}^d \|\frac{\partial^2 u}{\partial x_i \partial x_j}\|_{L^2(\Omega)}^2$ die Green'sche Formel

$$\int_{\Omega} w(D_i v) dx = - \int_{\Omega} (D_i w)v dx + \int_{\partial\Omega} wv\nu_i d\sigma, \quad w, v \in H^1(\Omega),$$

an.

- Nutzen Sie zudem die Poincaré-Ungleichung, um $\|\cdot\|_{L^2(\Omega)}$ und $|\cdot|_{H^1(\Omega)}$ abzuschätzen.
- ii) Zeigen Sie, dass für die spektrale Kondition der Steifigkeitsmatrix $K := (a(\varphi_i, \varphi_j))_{i,j=1}^n$ die Abschätzung

$$\kappa_2(K) \leq C \frac{h^d}{(\min_{T \in \tau_h} h_T)^{d+4}}$$

gilt.

Hinweis: Sie dürfen ohne Beweis die inverse Abschätzung für eine reguläre Triangulierung verwenden, d.h es gilt

$$\|v_h\|_{H^k(\Omega)} \leq \frac{c}{(\min_{T \in \tau_h} h_T)^k} \|v_h\|_{L^2(\Omega)} \quad \forall v_h \in V^h.$$

Aufgabe 3: (10 Punkte)

Betrachten Sie den in der Vorlesung behandelten Dirichlet-Dirichlet-Algorithmus zur Lösung von $-\Delta u = f$, $u|_{\partial\Omega} = 0$ auf einem Polygon $\Omega \subset \mathbb{R}^2$, partitioniert in zwei polygonale Teilgebiete $\Omega_i \subset \Omega$.

Wir nehmen an, dass im n -ten Iterationsschritt $r_{\Gamma}^n = 0$ erfüllt ist. Zeigen Sie, dass dann die zusammengesetzte Lösung $u := (u^{(1),n}, u^{(2),n})$ das global assemblierte System $K_{BB}u_B = f_B$ löst, wobei $B = (I, \Gamma) = (I^{(1)}, I^{(2)}, \Gamma)$ die inneren Knoten von Ω beschreibt.

Hinweis: Überlegen Sie sich und begründen Sie, warum

$$K_{\Gamma\Gamma} = K_{\Gamma\Gamma}^{(1)} + K_{\Gamma\Gamma}^{(2)}$$

und

$$f_{\Gamma} = f_{\Gamma}^{(1)} + f_{\Gamma}^{(2)}$$

gilt.

Algorithmus: Sei $n \geq 1$, $\theta \in (0, \theta_{\max})$, $\lambda_{\Gamma}^1 := 0$ sowie stets $\lambda_{\Gamma}^{(1),n} = \lambda_{\Gamma}^n$ und $\lambda_{\Gamma}^{(2),n} = -\lambda_{\Gamma}^n$. Wir benutzen im Folgenden dieselben Variablen für die diskrete und stetige Beschreibung.

- Neumannprobleme in Ω_i :

$$\begin{aligned} -\Delta u^{(i),n} &= f && \text{in } \Omega_i \\ u^{(i),n} &= 0 && \text{auf } \partial\Omega_i \setminus \Gamma \\ \frac{\partial u^{(i),n}}{\partial \nu_i} &= \lambda_{\Gamma}^{(i),n} && \text{auf } \Gamma \end{aligned}$$

- Definiere Residuum:

$$r_{\Gamma}^n := u_{\Gamma}^{(1),n} - u_{\Gamma}^{(2),n}$$

- Dirichletprobleme in Ω_i : (harmonische Fortsetzung der Randwerte)

$$\begin{aligned} -\Delta\psi^{(i),n} &= 0 && \text{in } \Omega_i \\ \psi^{(i),n} &= 0 && \text{auf } \partial\Omega_i \setminus \Gamma \\ \psi^{(i),n} &= r_\Gamma^n && \text{auf } \Gamma \end{aligned}$$

- Korrigiere Flussapproximation:

$$\lambda_\Gamma^{n+1} = \lambda_\Gamma^n - \theta \cdot \left(\frac{\partial\psi^{(1),n}}{\partial\nu_1} + \frac{\partial\psi^{(2),n}}{\partial\nu_2} \right) \quad \text{auf } \Gamma$$

Zu den Neumannproblemen gehören die diskretisierten Systeme

$$\begin{pmatrix} K_{II}^{(i)} & K_{I\Gamma}^{(i)} \\ K_{\Gamma I}^{(i)} & K_{\Gamma\Gamma}^{(i)} \end{pmatrix} \cdot \begin{pmatrix} u_I^{(i),n} \\ u_\Gamma^{(i),n} \end{pmatrix} = \begin{pmatrix} f_I^{(i)} \\ f_\Gamma^{(i)} + \lambda_\Gamma^{(i),n} \end{pmatrix}$$

und zu den Dirichletproblemen die Systeme

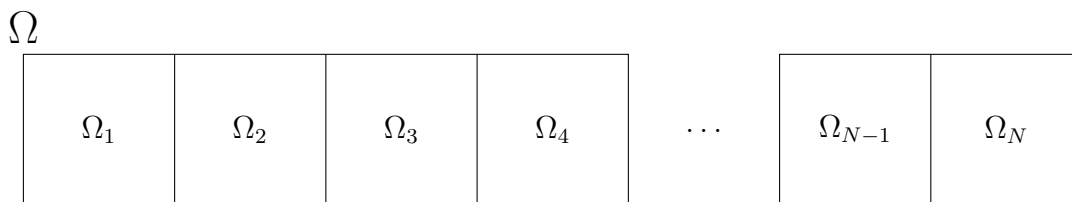
$$K_{II}^{(i)}\psi_I^{(i),n} + K_{I\Gamma}^{(i)}r_\Gamma^n = 0.$$

Die Flussapproximationen auf dem Interface erhält man über

$$\frac{\partial\psi^{(i),n}}{\partial\nu_i} = K_{\Gamma I}^{(i),n}\psi_I^{(i),n} + K_{\Gamma\Gamma}^{(i),n}\psi_\Gamma^{(i),n}.$$

Aufgabe 4: (4 Punkte)

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



Verallgemeinern Sie die Beschreibung des Dirichlet-Dirichlet-Algorithmus für zwei Teilgebiete aus Aufgabe 3 auf N Teilgebiete. Nutzen Sie als Muster die Algorithmusbeschreibung in Aufgabe 3. Beachten Sie auch den zweiten Teil der Programmieraufgabe, insbesondere den Hinweis.

Programmieraufgabe (Dirichlet-Dirichlet): (20 + 5 Punkte) **Abgabe bis 7. Nov.**

Sei das Modellproblem aus Aufgabe 4 gegeben.

- Implementieren Sie den Dirichlet-Dirichlet-Algorithmus. Testen Sie Ihre Implementierung für $f \equiv 1$, einer strukturierten Dreieckszerlegung ($P1$) mit $2 \cdot 40^2$ Elementen pro Teilgebiet sowie $N = 5$ und $\theta = 0.25$. Verwenden Sie für das Konvergenzkriterium eine Toleranz von 10^{-8} für das absolute Residuum und plotten Sie die Teilgebietslösungen wie in der Programmieraufgabe auf dem 2. Übungsblatt bis das absolute Residuum kleiner als 10^{-2} ist (drucken Sie nur die erste Iterierte aus).

Vergleichen Sie die benötigte Iterationsanzahl mit der für das (nicht-vorkonditionierte) CG-Verfahren zur Lösung des globalen Systems $Ku = b$ (ebenfalls bzgl. des absoluten Residuums und der Toleranz 10^{-8}), wobei hier K die global assemblierte Steifigkeitsmatrix mit eliminierten Randwerten und b der zugehörige Lastvektor ist. Geben Sie in beiden Fällen das finale globale Residuum $\|Ku - b\|_2$ an.

In Aufgabe 3 haben wir für zwei Teilgebiete gezeigt, dass der Dirichlet-Dirichlet-Algorithmus bei Konvergenz die Galerkinlösung berechnet; analog zeigt man die Aussage für N Teilgebiete auf einem Streifen. In der Vorlesung wurde für zwei Teilgebiete das Problem so umformuliert, dass die Iteration durch

$$\lambda_{\Gamma}^{n+1} = \lambda_{\Gamma}^n + \theta(S^{(1)} + S^{(2)}) \cdot (d_{\Gamma} - F\lambda_{\Gamma}^n)$$

beschrieben wird, wobei

$$F = (S^{(1)})^{-1} + (S^{(2)})^{-1}, \quad d_{\Gamma} = (S^{(1)})^{-1}g_{\Gamma}^{(1)} + (S^{(2)})^{-1}g_{\Gamma}^{(2)}, \quad g_{\Gamma}^{(i)} = f_{\Gamma}^{(i)} - K_{\Gamma I}^{(i)}(K_{II}^{(i)})^{-1}f_I^{(i)}.$$

Vollkommen analog zeigt man für den Fall mit N Teilgebieten, dass

$$\lambda_{\Gamma}^{n+1} = \lambda_{\Gamma}^n + \theta \left(\sum_{i=1}^N S^{(i)} \right) \cdot (d_{\Gamma} - F\lambda_{\Gamma}^n), \quad (1)$$

wobei

$$F = \sum_{i=1}^N (S^{(i)})^{-1}, \quad d_{\Gamma} = \sum_{i=1}^N (S^{(i)})^{-1}g_{\Gamma}^{(i)}, \quad g_{\Gamma}^{(i)} = f_{\Gamma}^{(i)} - K_{\Gamma I}^{(i)}(K_{II}^{(i)})^{-1}f_I^{(i)}.$$

Vorschrift (1) entspricht gerade dem Richardson-Verfahren

$$\lambda^{n+1} = \lambda^n + \alpha M^{-1} \tilde{r}^n$$

mit Residuum $\tilde{r}^n = d_{\Gamma} - F\lambda^n$ zur Lösung von $F\lambda_{\Gamma} = d_{\Gamma}$ mit $M^{-1} = \sum_{i=1}^N S^{(i)}$ als Vorkonditionierer und $\alpha = \theta$. Die Konvergenz des Verfahrens ist nun äquivalent dazu, dass der Spektralradius der Matrix $B := I - \alpha M^{-1}F$ kleiner als 1 ist.

- b) Berechnen Sie für das Modellproblem aus i) mit $\max(\text{abs}(\text{eig}(B)))$ den Spektralradius von B . Variieren Sie θ und geben Sie eine Schätzung für θ_{\max} an. Überprüfen Sie die Konvergenz anhand Ihres Programmes aus a).

Hinweis: Teilgebiet Ω_i interagiert nur mit einem Teil des Interfaces Γ . Dadurch sind obige Definitionen, wie z.B. $F = \sum_{i=1}^N (S^{(i)})^{-1}$, streng genommen nicht wohldefiniert. Hierbei ist immer gemeint, dass die Werte von, z.B. $S^{(i)}$ oder $(S^{(i)})^{-1}$, an die entsprechende Stelle geschrieben werden; siehe auch die mathematische Beschreibung unten.

Der folgende exemplarische Code berechnet den Spektralradius von B . Dabei ist der Indexvektor `gamma` vom Matlabtyp `logical` und gibt global die Knoten des Interfaces an; `l2g_sd{i}` gibt die globalen Knotennummern des i -ten Teilgebiets an; `cGamma{i}` die Interfaceknoten im i -ten Teilgebiet bezüglich der lokalen Nummerierung.

```
invM = zeros(nnz(gamma));
F = zeros(nnz(gamma));
mapGamma = zeros(length(gamma),1);
mapGamma(gamma) = 1:nnz(gamma);
for i = 1:N
    % Stelle Indexmap fuer Gamma_i --> Gamma auf.
    mapGamma_i = mapGamma(l2g_sd{i}(cGamma{i}));

    % Aktualisiere 'F' und 'inv(M)'.
    invM(mapGamma_i,mapGamma_i) = invM(mapGamma_i,mapGamma_i)+cS{i};
    F(mapGamma_i,mapGamma_i) = F(mapGamma_i,mapGamma_i)+inv(cS{i});
end
B = eye(nnz(gamma)) - theta*invM*F;
rho = max(abs(eig(B))); % Spektralradius von 'B'.
```

Die mathematischere Beschreibung des obigen Sachverhalts geschieht über Restriktionsmatrizen. Sei n_Γ die Knotenanzahl auf Γ und n_{Γ_i} die auf $\Gamma_i := \Gamma \cap \bar{\Omega}_i$. Mit $R_{\Gamma \rightarrow \Gamma_i} \in \mathbb{R}^{n_{\Gamma_i} \times n_\Gamma}$ bezeichnen wir die Matrix, welche die Funktion einer Indexabbildung von Γ auf Γ_i übernimmt. Sei beispielsweise der j -te Knoten von Γ_i gegeben und j_Γ der zugehörige Index in Γ , so hat $R_{\Gamma \rightarrow \Gamma_i}$ in der j -ten Zeile bzw. der Spalte j_Γ den Eintrag 1. In jeder Zeile von $R_{\Gamma \rightarrow \Gamma_i}$ steht somit genau eine 1 und sonst nur Nullen. Die Indexabbildung von Γ auf Γ_i geschieht über die Anwendung von $R_{\Gamma \rightarrow \Gamma_i}$. Die inverse Indexabbildung ist mit der Transponierten gegeben, d.h. es gilt $R_{\Gamma_i \rightarrow \Gamma} = R_{\Gamma \rightarrow \Gamma_i}^T$. Die Matrix F kann somit dargestellt werden als

$$F = \sum_{i=1}^N R_{\Gamma \rightarrow \Gamma_i}^T (S^{(i)})^{-1} R_{\Gamma \rightarrow \Gamma_i}.$$

Der Vorkonditionierer und die rechte Seite sind definiert über

$$M^{-1} = \sum_{i=1}^N R_{\Gamma \rightarrow \Gamma_i}^T S^{(i)} R_{\Gamma \rightarrow \Gamma_i}, \quad d_\Gamma = \sum_{i=1}^N R_{\Gamma \rightarrow \Gamma_i}^T (S^{(i)})^{-1} g_\Gamma^{(i)}.$$

Allgemeine Hinweise zum Programmiereteil

- Der Code **muss sinnvoll kommentiert** sein. Ein nicht kommentiertes Programm gilt als nicht erfolgreich bearbeitet.
- Das Programm muss ausführbar sein, ohne Änderungen am Code vornehmen zu müssen (d.h. ein Klick auf „Ausführen“ muss ausreichen). Schreiben Sie daher ein oder mehrere Skripte für die Teilaufgabe(n). Benennen Sie das Skript / die Skripte sinnvoll (z.B. `aufg1c.m`).
- Schreiben Sie bitte Funktionen in eigene Dateien und nicht in Skriptdateien (*Ausnahme*: anonyme Funktionen der Art `f = @(x) x.^2;`).
- Enthält ihr Code mehrere Funktionen, so ist jede Funktion in eine eigene Datei zu schreiben. *Ausnahme*: Die Funktion wird ausschließlich von anderen Funktionen derselben Datei aufgerufen. In diesem Fall steht an oberster Stelle der Funktionsdatei die Funktion, welche von außerhalb (z.B. von einem Skript) aufgerufen wird.

Abgabe des Programmiereteils

- Packen Sie Ihre Dateien in ein Archiv (Formate: `.zip`, oder `.tar.gz`) mit einem Dateinamen der Art:

`ueb01_nachname_vorname.zip`

- Den Quellcode schicken Sie bitte an die E-Mail-Adresse Ihrer Übungsgruppenleiter / Übungsgruppenleiterinnen, mit einem Betreff der Art:

Betreff: Uebung1, Nachname, Vorname

- Geben Sie bitte immer eine **ausgedruckte Version** Ihrer Programmcodes mit den schriftlichen Aufgaben ab (\rightarrow Kasten), sofern dies in der Aufgabenstellung nicht eindeutig anders vermerkt wurde.

- Sofern es zur sinnvollen Lösung der Aufgabenstellung nötig ist, drucken Sie bitte auch die Ausgabe von Matlab aus. Dies sollte nicht zwei DIN-A4-Seiten überschreiten. Gleiches gilt für Grafiken.

Abgabe im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts.

- Theorie: Bis Mittwoch, 31. Oktober 2018, 16:00 Uhr.
- Programmieraufgabe: Bis Mittwoch, 07. November 2018, 16:00 Uhr.