

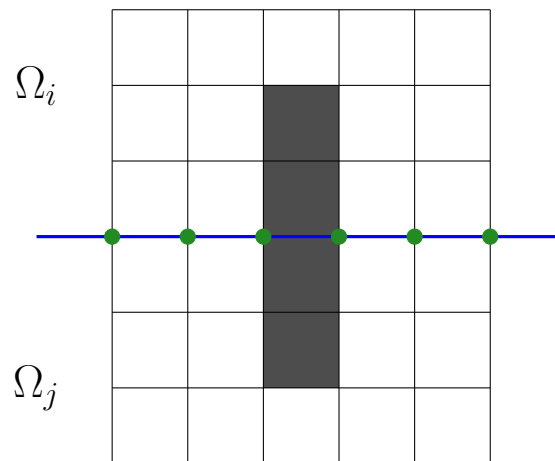
Prof. Dr. A. Klawonn
J. Knepper, M. Sc.
J. Weber, M. Sc.

05. Dezember 2018

9. Übung zu Wissenschaftliches Rechnen I

Aufgabe 1 (FETI-DP: B_D -Operator: Skalierung): (2 + 2 = 4 Punkte)

In der folgenden Abbildung ist eine Teilgebietskante (blau) und der Ausschnitt eines FE-Gitters dargestellt. In grün sind duale Knoten markiert. In dunkelgrau markierten Elementen sei $\rho := \rho_A$ und sonst $\rho := \rho_B$, wobei $\rho_A \gg \rho_B > 0$.



Zu einem FE-Knoten x wird die entsprechende Zeile von $B^{(i)}$ mit $\delta_j^+(x)$ und von $B^{(j)}$ mit $\delta_i^+(x)$ skaliert, um $B_D^{(i)}$ bzw. $B_D^{(j)}$ zu erhalten. Die Koeffizientenfunktion ρ ist, anders als bisher, nicht konstant auf den Teilgebieten. Wir definieren

$$\delta_j^+(x) := \frac{\rho_{\max}(x^{(j)})}{\rho_{\max}(x^{(j)}) + \rho_{\max}(x^{(i)})},$$

wobei $\rho_{\max}(x^{(j)})$ der maximale Koeffizient eines Elements ist, welches den Knoten x enthält und zudem in Ω_j liegt, d.h.

$$\rho_{\max}(x^{(j)}) := \max_{\substack{T \in \tau_h \\ T \subset \Omega_j \\ x \in T}} \rho(T).$$

Siehe auch die Beschreibung zur Programmieraufgabe im Anhang.

- Zeigen Sie, dass $\delta_j^+ = \delta_i^+$ konstant auf der Kante ist.
- Modifizieren Sie das Beispiel leicht, sodass a) nicht mehr erfüllt ist.

Aufgabe 2 (FETI-DP: P_D -Abschätzung): (7 + 9 = 16 Punkte)

Seien alle Ecken primal. Wir betrachten den Beweis der Abschätzung

$$|P_D w|_{\tilde{S}}^2 \leq C \left(1 + \log\left(\frac{H}{h}\right)\right)^2 |w|_{\tilde{S}}^2 \quad \forall w \in \tilde{W}.$$

Die Koeffizientenfunktion sei wie in Aufgabe 1 gewählt, d.h. insbesondere nicht konstant auf den Teilgebieten.

- Vollziehen Sie den Beweis zur P_D -Abschätzung nach und zeigen Sie, an welchen Stellen der Beweis für die Koeffizientenfunktion aus Aufgabe 1 nicht mehr funktioniert.
- Modifizieren Sie die Stellen und leiten Sie eine Konditionszahlabschätzung her.

Schreiben Sie in a) die kritischen Stellen aus dem Beweis der Vorlesung ab und füllen Sie die restlichen Bereiche mit Platzhaltern, sodass die Beweisstruktur nachvollziehbar ist. Ähnlich gehen Sie in b) vor, indem Sie die kritischen Bereiche durch Ihre Modifikationen ersetzen.

Hinweise:

- Betrachten Sie die Ungleichungen, in denen die diskret ρ -harmonische Fortsetzung $\mathcal{H} := \mathcal{H}^\rho$ bzw. die Energieminimalität dieser Fortsetzung verwendet wird.
- Die Konstante C in der P_D -Abschätzung hängt nun vom Koeffizientensprung ab.

Programmieraufgabe: (15 + 10 = 25 Punkte) **Abgabe bis: 19. Dez.**

Im Folgenden betrachten wir auf $\Omega = (0, 1)^2$ das Variationsproblem: Finde $u \in H_0^1(\Omega)$, sodass

$$\int_{\Omega} \rho \nabla u \cdot \nabla v \, dx = \int_{\Omega} v \, dx \quad \forall v \in H_0^1(\Omega).$$

Sei Ω durch eine strukturierte Dreieckszerlegung ($P1$) diskretisiert und sei ρ **konstant auf den Elementen**. Wir partitionieren Ω in $N \times N$ quadratische Teilgebiete mit jeweils $2 \cdot 10^2$ Elementen.

- Implementieren Sie den FETI-DP-Algorithmus zur Lösung des vorkonditionierten Systems

$$M^{-1} F \lambda = M^{-1} d,$$

wobei M^{-1} der Dirichlet-Vorkonditionierer nach Farhat, Mandel und Roux (1994) ist, d.h.

$$M^{-1} := B_{D,\Gamma} S_{\Gamma\Gamma} B_{D,\Gamma}^T = \sum_{i=1}^N B_{D,\Gamma}^{(i)} S_{\Gamma\Gamma}^{(i)} B_{D,\Gamma}^{(i),T}, \quad (1)$$

mit $S_{\Gamma\Gamma}^{(i)} := K_{\Gamma\Gamma}^{(i)} - K_{\Gamma I}^{(i)} (K_{II}^{(i)})^{-1} K_{I\Gamma}^{(i)}$ (hierbei enthält I keine Dirichletrandknoten¹). Wählen Sie alle Eckknoten primal, dann gilt zudem (Warum?)

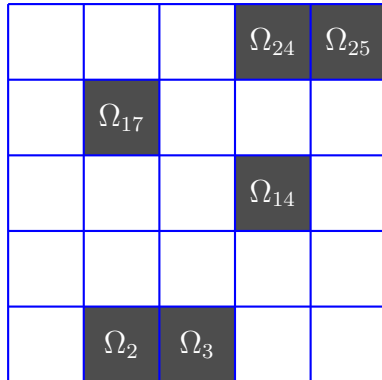
$$M^{-1} = \sum_{i=1}^N B_{D,\Delta}^{(i)} S_{\Delta\Delta}^{(i)} B_{D,\Delta}^{(i),T},$$

mit $S_{\Delta\Delta}^{(i)} = K_{\Delta\Delta}^{(i)} - K_{\Delta I}^{(i)} (K_{II}^{(i)})^{-1} K_{I\Delta}^{(i)}$ (hierbei enthält I , wie vorher, nur innere Knoten und weder Dirichletrandknoten noch primale Knoten¹).

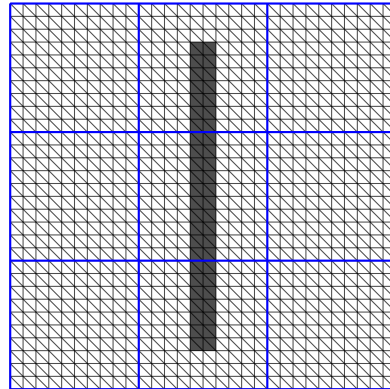
Stellen Sie M^{-1} nicht explizit auf, sondern schreiben Sie eine Funktion, welche $M^{-1}x$ auswertet. Stellen Sie dabei auch $S_{\Gamma\Gamma}^{(i)}$ bzw. $S_{\Delta\Delta}^{(i)}$ nicht explizit auf.

¹Man kann auch die Dirichletrandknoten zu I hinzufügen. Dadurch wird die Matrix jedoch ein wenig größer.

- ii) Testen Sie Ihre Implementierung mit der Toleranz 10^{-8} bzgl. des relativen unvorkonditionierten Residuums und $\lambda^{(0)} := 0$ als Startvektor. Testen Sie jeweils M^{-1} wie in (1) angegeben und die Identität $M^{-1} = I$. Geben Sie in beiden Fällen die benötigte Anzahl Iterationen und die Konditionszahl schätzung aus dem Lanczos-Prozess an. Lösen Sie zusätzlich das global assemblierte System mit Ihrer CG-Implementierung und geben Sie auch hier die benötigte Anzahl Iterationen und die Konditionszahl schätzung aus. Testen Sie die folgenden Koeffizientenfunktionen:



Koeffizientenfunktion (i)
 $N = 5$



Koeffizientenfunktion (ii)
 $N = 3$

$$\rho = \begin{cases} 10^6, & \text{in } \Omega_i, i \in \{2, 3, 14, 17, 24, 25\}, \\ 1, & \text{sonst.} \end{cases}, \quad \rho = \begin{cases} 10^6, & \frac{16}{30} \geq x \geq \frac{14}{30} \wedge \frac{27}{30} \geq y \geq \frac{3}{30}, \\ 1, & \text{sonst.} \end{cases}$$

Es ist $\rho = 10^6$ in dunkelgrau gefärbten Gebieten bzw. Elementen und sonst $\rho = 1$.

Plotten Sie in beiden Fällen jeweils die Teilgebietslösungen wie in der Programmieraufgabe auf dem 2. Übungsblatt.

- iii) *Bonusaufgabe:* Die Anzahl Iterationen für die Beispiele i) und ii), die FETI-DP benötigt, sind recht gering; auch wenn kein Vorkonditionierer verwendet wird und auch bei hohen Konditionszahlen. Dies liegt daran, dass es wenige „problematische“ Stellen gibt und zudem die Anzahl Teilgebiete sehr niedrig ist. Mit einer Zufallsverteilung lässt sich einfach eine Koeffizientenfunktion generieren, die (meist) in einem harten Problem resultiert. Dazu definiert man die globale Koeffizientenfunktion `rho` wie folgt:

```
% Setze alle Koeffizienten der Elemente auf 'rhoMin'.
rho = rhoMin*ones(size(tri,1),1);

% Setze nun alle 'rhoMax' Koeffizienten.
rho(rand(length(rho),1) < 0.25) = rhoMax;
```

Mit wachsender Anzahl Teilgebiete wird nun auch die Anzahl Iterationen steigen.

Hinweise:

- Nutzen Sie die Lösung des global assemblierten Systems, um die Korrektheit Ihrer Implementierung zu überprüfen.
- Bei (i) müssen nur die Teilgebietssteifigkeitsmatrizen für $-\Delta u = 0$ mit ρ skaliert werden und bei (ii) die Elementsteifigkeitsmatrizen, um die korrekten Steifigkeitsmatrizen (für $\rho \neq 1$) zu assemblieren. Um nicht zwei verschiedene Versionen implementieren zu müssen, ist es jedoch sinnvoll die zweite Variante auch für (i) zu nutzen.

- Die Lösung des global assemblierten Systems mit CG wird viele Schritte benötigen. Sofern Sie die maximale Iterationszahl beschränken, setzen Sie diese auf 10 000.
- Beachten Sie auch die Implementierungshinweise im Anhang.

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, 12. Dezember 2018, 16:00 Uhr.
- Programmieraufgabe: Bis Mittwoch, 19. Dezember 2018, 16:00 Uhr.

Anhang - Implementierungshinweise

Koeffizientenvektor: Mit einer leichten Erweiterung der Partitionierung in der Funktion `meshPartSquare` lässt sich eine „local-2-global-map“ (`l2g_tri_sd`) für die Elemente bestimmen. Die Erweiterung wird auf der Übungsseite zur Verfügung gestellt.

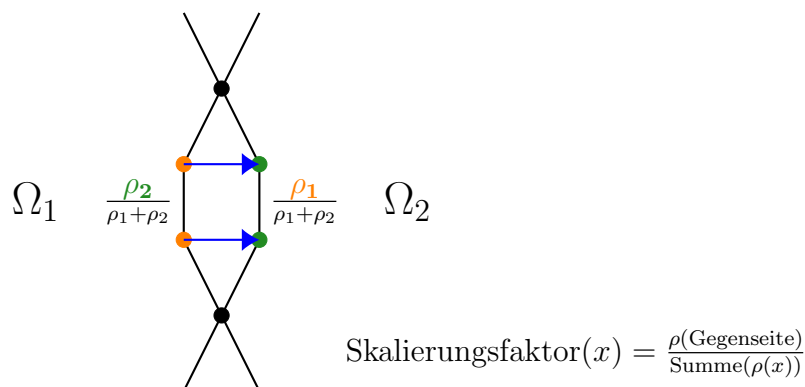
Schreiben Sie zwei Funktionen, die zunächst jeweils zur globalen Triangulierung einen Koeffizientenvektor `rho` erzeugen, der zu jedem Element den zugehörigen Koeffizienten angibt. Sind die Koeffizienten konstant auf den Teilgebieten, so lässt sich dies einfach mittels Zugriff auf `rho(l2g_tri_sd{i})` erreichen. Für das „Kanalproblem“ berechnen Sie, wie in `meshPartSquare`, den Schwerpunkt der Dreiecke und bestimmen anschließend über die in der Aufgabenstellung angegebenen Koordinaten alle Elemente, die innerhalb des Kanals liegen.

Erzeugen Sie nun mit Hilfe von `l2g_tri_sd` für jedes Teilgebiet die lokalen Koeffizientenvektoren `cRho{i}`.

Assemblierung: Mit Hilfe von `cRho` und `rho` lässt sich mit der Skalierung der Elementsteifigkeitsmatrizen mit dem entsprechenden Koeffizienten die Assemblierung anpassen.

B_D -Operator: (alle Eckknoten sind primal)

Beispiel, wenn ρ konstant auf den Teilgebieten ist:



$B_{D,\Delta}$ hat dann (abhängig von der Nummerierung) die folgende Gestalt:

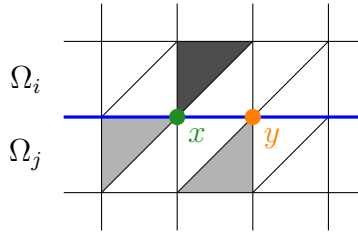
$$B_{D,\Delta} = \left(\begin{array}{cc|cc} \frac{\rho_2}{\rho_1 + \rho_2} & 0 & -\frac{\rho_1}{\rho_1 + \rho_2} & 0 \\ 0 & \frac{\rho_2}{\rho_1 + \rho_2} & 0 & -\frac{\rho_1}{\rho_1 + \rho_2} \end{array} \right)$$

Ist ρ nur konstant auf den Elementen, jedoch nicht auf den Teilgebieten, so benötigen wir eine allgemeinere Beschreibung: Sei τ_h die Triangulierung und x ein dualer Knoten, der u.a. im j -ten Teilgebiet liegt, dann definiere den maximalen Koeffizient über alle Elemente, die in Ω_j liegen und x enthalten, als

$$\rho_{\max}(x^{(j)}) := \max_{\substack{T \in \tau_h \\ T \subset \Omega_j \\ x \in T}} \rho(T).$$

Beispiel: In der folgenden Abbildung sei $\rho = 10^{12}$ auf dunkelgrauen Elementen und $\rho = 10^6$ auf hellgrauen Elementen und sonst $\rho = 1$. Es gilt

$$\rho_{\max}(x^{(j)}) = 10^6, \quad \rho_{\max}(y^{(j)}) = 10^6, \quad \rho_{\max}(x^{(i)}) = 10^{12}, \quad \rho_{\max}(y^{(i)}) = 1.$$



Sei nun $x^{(i)}$ ein dualer Knoten im i -ten Teilgebiet und $x^{(j)}$ der zugehörige Knoten im j -ten Teilgebiet, sodass $x^{(i)} \rightarrow x^{(j)}$ mit einem Lagrangeschen Multiplikator gekoppelt sind. Sei o.B.d.A der Eintrag in B_D zu $x^{(i)}$ positiv und entsprechend zu $x^{(j)}$ negativ, dann ist der Skalierungsfaktor in Ω_i zu $x^{(i)}$ gerade

$$\frac{\rho_{\max}(x^{(j)})}{\rho_{\max}(x^{(j)}) + \rho_{\max}(x^{(i)})}$$

und in Ω_j zu $x^{(j)}$

$$\frac{\rho_{\max}(x^{(i)})}{\rho_{\max}(x^{(j)}) + \rho_{\max}(x^{(i)})}.$$

Dass in einem Teilgebiet der Koeffizient aus dem Nachbargebiet benötigt wird und zudem der maximale (lokale) Koeffizient, macht die Implementierung von B_D deutlich aufwendiger als die des unskalierten B .

Implementierung: Erzeugen Sie für die Teilgebiete einen Cell-Array `cRhoMax` und einen globalen Vektor `rhoMaxSum` für die Summe der Koeffizienten eines Knotens. In `cRhoMax{i}` speichern Sie für das i -te Teilgebiet $\rho_{\max}(x^{(i)})$ für alle lokalen Knoten $x \in \Omega_i^h$. Laufen Sie dazu über alle lokalen Elemente und bestimmen Sie über

$$\text{cRhoMax}\{i\}(\text{ind}) = \max(\text{cRhoMax}\{i\}(\text{ind}), \text{rhoElement});$$

den maximalen anliegenden Koeffizienten eines Knotens; hierbei ist `ind` ein Knotenindex.

Die Summe der Koeffizienten können Sie anschließend einfach mit Hilfe von `l2g_sd` (Knoten-ID-Mapping) und `cRhoMax` bestimmen.

Als Nächstes bestimmen Sie mittels einer „Two-Pass“-Methode die Skalierungsfaktoren des B_D -Operators. Legen Sie dazu einen globalen Vektor `scalings` mit zwei Spalten an. Iterieren Sie über alle Teilgebiete: Im i -ten Teilgebiet schreiben Sie (für alle Knoten; siehe auch die Bemerkung unten) in die erste Spalte den Skalierungsfaktor $\frac{\rho_{\max}(x^{(i)})}{\rho_{\max}(x^{(j)}) + \rho_{\max}(x^{(i)})}$, sofern der entsprechende Eintrag in B bzw. B_D positiv gesetzt werden soll. In die zweite Spalte schreiben Sie entsprechend den Skalierungsfaktor, falls er negativ gesetzt werden soll. Das heißt pro Teilgebiet füllen Sie immer nur eine Spalte pro Knoten aus.

Ob der Wert positiv oder negativ ist, können Sie wie bei der Aufstellung von B ermitteln. Nutzen Sie dazu einen logischen globalen Vektor, der auf `true` gesetzt wird, sofern der entsprechende Knoten bereits „entdeckt“ wurde.

Bemerkung: Es werden mit dieser Vorgehensweise auch nicht-duale Knoten behandelt. Dies ist kein Problem, da später die entsprechenden Werte aus `scalings` zu dualen Knoten extrahiert werden; die restlichen Werte werden verworfen. Diese Vorgehensweise vereinfacht die Implementierung, da keine zusätzlichen ID-Mappings berechnet werden müssen.

In einem zweiten Durchlauf können Sie nun im i -ten Teilgebiet (nehmen wir an, der Skalierungswert ist positiv) den Skalierungsfaktor des j -ten Teilgebiets ermitteln, indem Sie auf die 2. Spalte von `scalings` zugreifen. Sie können nun, wie vorher $B^{(i)}$, also `cB{i}`, und den skalierten Sprungoperator $B_D^{(i)}$, also `cBD{i}`, aufstellen.

Anhand des Beispiels von oben: 1. Durchlauf:

$$\Omega_1 : \mathbf{scalings} \leftarrow \begin{pmatrix} \vdots & \vdots \\ \frac{\rho_1}{\rho_1 + \rho_2} & 0 \\ \vdots & \vdots \end{pmatrix} \longrightarrow \Omega_2 : \mathbf{scalings} \leftarrow \begin{pmatrix} \vdots & \vdots \\ \frac{\rho_1}{\rho_1 + \rho_2} & \frac{\rho_2}{\rho_1 + \rho_2} \\ \vdots & \vdots \end{pmatrix}$$

Beim zweiten Durchlauf wird wieder das 1. Teilgebiet vor dem 2. Teilgebiet bearbeitet. Dadurch sind die dualen Knoten in Ω_1 als „noch nicht entdeckt“ markiert. Daher weiß man, dass in der zweiten Spalte von $\mathbf{scalings}$ der passende Skalierungsfaktor steht und dieser nicht-negiert in B_D geschrieben werden muss. Wenn dann das 2. Teilgebiet bearbeitet wird, sind die entsprechenden dualen Knoten (die auch zum 1. Teilgebiet gehören) als „bereits entdeckt“ markiert. Daher steht in der 1. Spalte von $\mathbf{scalings}$ der Skalierungsfaktor, der zudem negiert werden muss.