Technische Universität Chemnitz
Prof. Dr. R. Herzog, Dr. M. Winkler

Chemnitz, 18. Juni 2018
Abgabe am 2. Juli 2018

Numerik partieller Differentialgleichungen

Übung 11

Aufgabe 24: Sattelpunktproblem für Poisson-Problem mit reinen Neumann-Randbedingungen

Wir greifen das Poisson-Problem mit homogenen reinen Neumann-Randbedingungen aus Satz 9.11 auf. Die hierbei auftretende Steifigkeitsmatrix A ist nicht invertierbar. Damit das System $A\vec{u} = \vec{f}$ eine Lösung besitzt, muss eine Kompatibilitätsbedingung erfüllt sein (vgl. Übung 7, Aufgabe 18).

Geben Sie $\ker(A)$ an und bestimmen Sie eine einzeilige Matrix B, sodass die Systemmatrix im Sattelpunktproblem $\mathcal{K} = \begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix}$ invertierbar ist.

Was ist mit der Kompatibilitätsbedingung passiert?

Aufgabe 25: Aspektverhältnis und Formregularität

(a) Wie lautet das Aspektverhältnis von
  (a) einem gleichseitigen Dreieck,
  (b) eines achsenparallelen d-dimensionalen Quaders sowie
  (c) dem Einheitssimplex in $\mathbb{R}^d$?

(b) Zeigen Sie Bemerkung 14.14 (f). Folgern Sie daraus, dass es für die Formregularität einer Familie $\{\mathcal{T}_h\}_{h>0}$ von Dreiecksgittern notwendig und hinreichend ist, wenn der kleinste Winkel in jedem Dreieck der Familie größer als eine Konstante $c > 0$ ist, d.h. gleichmäßig von 0 entfernt ist.

(c) Beweisen Sie, dass die alternative Definition des Begriffes der Formregularität aus Bemerkung 14.14 (b) äquivalent zur Definition 14.13 (c) ist.

Hinweis: Für konvexe Mengen $K \subset \mathbb{R}^d$ gilt (siehe P. Steinhagen, 1922)

$$\frac{\sqrt{d+2}}{d+1} \cdot \underline{d}_K \leq \varrho_K,$$

wobei $\underline{d}_K$ die Breite von K, also den kleinsten Abstand zweier paralleler Stützhyperebenen, die K einschließen, bezeichnet.

Hausaufgabe 20: Implementierung adaptiver finite Elemente in 2D

Diese Aufgabe basiert auf der Implementierung aus Übung 10, Aufgabe 23. Ziel ist es ein adaptives Verfahren unter Verwendung eines residualen Fehlerschätzers zu implementieren. Die Schritte

`SOLVE` $\rightarrow$ `ESTIMATE` $\rightarrow$ `MARK` $\rightarrow$ `REFINE`

sollen dabei nacheinander umgesetzt werden:

`ESTIMATE` : Hier soll der lokale (zellweise) Fehlerschätzer aus Satz 16.4 implementiert werden. Geeignete Quadraturformeln für die Zellresiduen sind der Routine `quadrature_unit_triangle_area.m` zu entnehmen. Beachte, dass diese Quadraturformeln nur für das Einheitsquadrat gültig sind. Man muss hier wieder mit der affinen Transformation arbeiten. Vergleiche dazu die Implementierung von `area_integrator.m`.

`MARK` : Verwende hier die Dörfler-Strategie aus Abschnitt 16.3. Eine erste Implementierung ist in der Datei `doerfler_marking.m` aus Übung 4, Aufgabe 9 zu finden. Die Kopfzeile dieser Funktion sollte

```
function ref_elem = doerfler_marking(eta2,sigma)
```

lauten. Dabei sind im Vektor `eta2` die Quadrate der lokalen Fehlerschätzer η_K enthalten (siehe Schritt `ESTIMATE`). Der Rückgabewert `ref_elem` ist ein Vektor mit den Zellindizes, die verfeinert werden sollen.

`REFINE` : Schaue dazu in den Hilfetext der Datei `myrefinemesh.m`. Die dort implementierte Funktion muss mit einem dritten Parameter aufgerufen werden um eine lokale Verfeinerung zu realisieren. Dieser entspricht dem Rückgabewert `ref_elem` der Funktion `doerfler_marking`.

Teste die Implementierung auf verschiedenen Gebieten und mit verschiedenen Eingangsdaten und beobachte, an welchen Stellen das Netz verfeinert wird. Beispielsweise sollte das in folgenden Fällen geschehen:

- An der einspringenden Ecke eines L-förmigen Gebiets (verwende dazu `geometry = 'lshapeg';`)
- Im Fall von $\mathcal{P}_1$-Elementen an den Stellen, an denen die Rechte Seite f einen großen Funktionswerte aufweist.
- An springenden Quelltermen oder Koeffizienten.