Nichtkonforme und gemischte Finite-Element-Methoden

Frühlingssemester 2016
Prof. Dr. H. Harbrecht

Universität Basel

Programmieraufgabe 1.

zu bearbeiten bis **Dienstag, 15.3.2016**.

In dieser Programmieraufgabe lösen wir das Poisson-Problem

$$\begin{aligned} -\Delta u(\mathbf{x}) &= f(\mathbf{x}) \qquad \mathbf{x} \in \Omega := (0,1)^2 \\ u(\mathbf{x}) &= 0 \qquad\quad \mathbf{x} \in \Gamma := \partial\Omega \end{aligned}$$

mithilfe linearer Crouzeix-Raviart-Elemente. Zu diesem Zweck kann ein Grossteil der Programmieraufgaben aus dem letzten Semester wiederverwendet werden. Zusätzlich benötigen wir noch eine Funktion, die ein nichtkonformes Gitter erstellt, sowie eine Funktion, die nichtkonforme Elemente visualisiert.

Aufgabe 1. Schreiben Sie eine Funktion

```
function [P_nc,F_nc,B_nc] = c2nc(P,F,B),
```

die ein konformes Gitter, gegeben durch die Punkteliste $\mathbf{P}$, die Elementliste $\mathbf{F}$ und die Randbedingungen $\mathbf{B}$, in ein nichtkonformes Gitter transformiert. Dazu kann der folgende Algorithmus verwendet werden:

Input : Punktliste $\mathbf{P}$, Elementliste $\mathbf{F}$, Randbedingungen $\mathbf{B}$
Output : Nichtkonforme Punktliste $\mathbf{P}_{nc}$, nichtkonforme Elementliste $\mathbf{F}_{nc}$, nichtkonforme Randbedingungen $\mathbf{B}_{nc}$
initialisiere $\mathbf{K} =$ `Kantenliste`$(\mathbf{P},\mathbf{F},\mathbf{B})$
for *i=1:length*$\big(\mathbf{K}\big)$ **do**
  for *j=1:length*$\big(\mathbf{K}(i,:)\big)$ **do**
    berechne den Mittelpunkt der Kante $\big[i, \mathbf{K}(i,j)\big]$, hänge ihn an $\mathbf{P}_{nc}$ an
    und setze $\mathbf{L}(i,j) := \mathit{length}(\mathbf{P}_{nc}) + 1$
    if $\mathbf{B}(i) = 0$ & $\mathbf{B}\big(\mathbf{K}(i,j)\big) = 0$ **then**
      setze $\mathbf{B}_{nc}\big(\mathbf{L}(i,j)\big) := 0$
    else
      setze $\mathbf{B}_{nc}\big(\mathbf{L}(i,j)\big) := 1$
    end
  end
end
for *i=1:length*$\big(\mathbf{F}\big)$ **do**
  for *j=1:3* **do**
    finde den Index (k_1,k_2) der Kante $\big[\mathbf{F}(i,j), \mathbf{F}(i, 1+j \mod 3)\big]$ in $\mathbf{K}$
    setze $\mathbf{F}_{nc}(i, 1+j \mod 3) := \mathbf{L}(k_1,k_2)$
  end
end

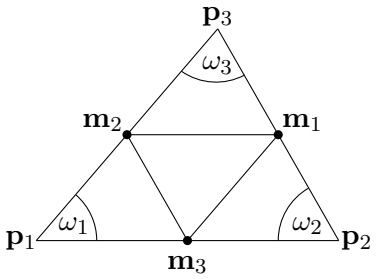
Aufgabe 2. Implementieren Sie zwei Funktionen

```
function S=fem_nc(P_nc,F_nc,B_nc)
```

und

```
function M=mass_nc(P_nc,F_nc,B_nc),
```

die die Steifigkeits- und Massenmatrix für nichtkonforme Elemente aufstellen. Die Massenmatrix $\mathbf{M}$ wird später benötigt. Bei der Steifigkeitsmatrix gilt es folgendes zu beachten:
Das Dreieck $\triangle(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ (siehe unten) geht aus dem Dreieck der Seitenmittelpunkte $\triangle(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ durch Spiegelung der Eckpunkte an den jeweils gegenüberliegenden Seiten hervor.



Dies hat zur Folge, dass die lokale Steifigkeitsmatrix für nichtkonforme Elemente ein Vielfaches der Matrix für konforme Elemente ist, d.h.

$$\mathbf{A}_T = 2 \begin{bmatrix} \cot(\omega_3) + \cot(\omega_2) & -\cot(\omega_3) & -\cot(\omega_2) \\ -\cot(\omega_3) & \cot(\omega_3) + \cot(\omega_1) & -\cot(\omega_1) \\ -\cot(\omega_2) & -\cot(\omega_1) & \cot(\omega_2) + \cot(\omega_1) \end{bmatrix}.$$

Sie können also Ihren Code aus dem vorangegangenen Semester wiederverwenden.
Die rechte Seite des Gleichungssystems wird nun durch eine Funktion

```
function R=rhs(P_nc,F_nc,B_nc,f)
```

realisiert. Da man für nichtkonforme Elemente das Skalarprodukt (f, ϕ_i) nicht wie bisher durch Interpolation bestimmen kann, muss man die zweidimensionale Mittelpunktregel benutzen, um es auszurechnen. Dies geschieht wie folgt:

Input : Nichtkonforme Punktliste $\mathbf{P}_{nc}$, nichtkonforme Elementliste $\mathbf{F}_{nc}$,
nichtkonforme Randbedingungen $\mathbf{B}_{nc}$, rechte Seite f
Output : approximierte rechte Seite $\mathbf{R}$
initialisiere $\mathbf{R} := zeros\big(length(\mathbf{P}_{nc}), 1\big)$
for *i=1:length*$\big(\mathbf{F}_{nc}\big)$ **do**
  berechne die Fläche $\mathbf{A}$ des Dreiecks mit den Seitenmittelpunkten
  $\mathbf{P}_{nc}\big(\mathbf{F}_{nc}(i,:)\big)$
  berechne den Schwerpunkt $\mathbf{S}$ des Dreiecks gegeben durch $\mathbf{P}_{nc}\big(\mathbf{F}_{nc}(i,:)\big)$
  setze $\mathbf{R}\big(\mathbf{F}_{nc}(i,:)\big) := \mathbf{R}\big(\mathbf{F}_{nc}\big(i,:)\big) + \mathbf{A} \cdot f(\mathbf{S})/3$
end
$\mathbf{R}(\mathbf{B}_{nc} = 0) := 0$

Aufgabe 3. Erstellen Sie eine Funktion

```
function visfunc_nc(u_nc,P_nc,F_nc),
```

die die Lösung des Poisson-Problems auf einem nichtkonformen Gitter visualisiert. Modifizieren Sie dafür ihre Visualisierung für konforme Finite Elemente so, dass sie auch an den Kanten unstetige Funktionen darstellen kann.

Aufgabe 4. Lösen Sie das Poisson-Problem auf dem Einheitsquadrat mit dem CG-Verfahren und der rechten Seite $f(\mathbf{x}) = 2\pi^2 \sin(\pi x)\sin(\pi y)$.
Die exakte Lösung ist $u(\mathbf{x}) = \sin(\pi x)\sin(\pi y)$. Bestätigen Sie, dass der L^2-Fehler der Lösung, d.h. $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$, quadratisch konvergiert. Berechnen Sie dafür die L^2-Orthoprojektion $\mathbf{Pu}$ von $\mathbf{u}$ und zeigen Sie, dass der relative Fehler $\|\mathbf{Pu} - \mathbf{u}_h\|_2$ quadratisch konvergiert. Sie können dafür wie folgt vorgehen:

Input : Nichtkonforme Punktliste $\mathbf{P}_{nc}$, nichtkonforme Elementliste $\mathbf{F}_{nc}$, nichtkonforme Randbedingungen $\mathbf{B}_{nc}$, exakte Lösung $\mathbf{u}$, approximierte Lösung $\mathbf{u}_h$
Output : Fehlerschätzer $\mathbf{e}$
setze $\mathbf{Pu} :=$ `rhs`$(\mathbf{P}_{nc}, \mathbf{F}_{nc}, \mathbf{B}_{nc}, \mathbf{u})$
berechne $\mathbf{M} :=$ `mass_nc`$(\mathbf{P}_{nc}, \mathbf{F}_{nc}, \mathbf{B}_{nc})$
löse $\mathbf{Pu} :=$ `CG`$(\mathbf{M}, \mathbf{Pu})$
setze $\mathbf{e} := \|\mathbf{Pu} - \mathbf{u}_h\|_2 / \|\mathbf{Pu}\|_2$