DEPARTMENT MATHEMATIK
UND INFORMATIK
DER UNIVERSITÄT ZU KÖLN

Sommersemester 2019

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

03. April 2019

1. Übung zu Wissenschaftliches Rechnen II

- Richten Sie sich eine Entwicklungsumgebung für Python 3 und FEniCS ein. Weitere Informationen finden Sie auf der Übungswebseite.
- Schauen Sie sich die Python-Beispielprogramme (→ Übungswebseite) an und testen Sie diese.
- Testen Sie zur Überprüfung Ihrer FEniCS-Installation das Poissonbeispiel aus dem FEniCS-Tutorial.

Programmieraufgabe 1: (20 Punkte)
Implementieren Sie in Python (ohne FEniCS) ein FEM-Programm zum Diffusionsproblem $-\Delta u = 1$ mit Dirichletnullrand in 2D fuer eine strukturierte Zerlegung von $(0,1)^2$ in $2 \cdot 10^2$ Dreiecke. Nutzen Sie $\mathcal{P}_1$-Ansatzfunktionen.

Zusätzlich zu den auf der Übungswebseite zur Verfügung gestellten Beispielprogrammen, können folgende Befehle und Codeschnipsel hilfreich sein.

- Das Python-Äquivalent zum Matlab-Befehl `trisurf`:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D # mpl_toolkits gehoert zu
   matplotlib.
# [...]
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_trisurf(x[:,0], x[:,1], u, triangles=tri,
                cmap=plt.cm.Spectral_r,
                antialiased=False)
plt.show()
```

- Wie in Matlab, kann auch in Python ein logischer Vektor durch einen Vergleich der Art `x[:,0] == 1` erzeugt werden. Der Befehl `numpy.logical_or.reduce(( A,B,C,D ))` ist das Äquivalent zum Matlabbefehl `(A | B | C | D)`.
- Die Befehle `numpy.nonzero(A)[0]` oder `numpy.where(A)[0]` entsprechen bei Vektoren dem Matlab-Befehl `find(A)`.
- Die aus Matlab bekannte `spy(A)`-Funktion befindet sich in `matplotlib.pylab.*`. Achtung: Es muss nachfolgend immer der Befehl `matplotlib.pylab.show()` ausgeführt werden; siehe `trisurf`-Beispiel.

- Python (hier: Numpy) differenziert nicht bezüglich $1 \times n$ und $n \times 1$ Vektoren; das Transponieren von Vektoren hat keinen Effekt. Für zwei Zeilenvektoren `a` und `b` erzeugt `[a',b']` in Matlab eine Matrix. In Python kann dazu `numpy.column_stack((a,b))` genutzt werden.
- Ist `ind` ein Indexvektor (`integer` oder `logical`), so kann in Matlab mit `A(ind,ind)` auf die entsprechende Teilmatrix zugegriffen werden. In Python ist dies nur möglich, sofern die Indizes wie folgt angegeben werden: `A[0:3,0:2]`. Dies extrahiert eine 3×2-Matrix. Liegt ein Indexvektor `ind` vor, so kann der Numpy-Befehl `ix_` genutzt werden: `A[numpy.ix_(ind,ind)]`.

Programmieraufgabe 2: (6 Punkte)
Implementieren Sie in FEniCS das Diffusionsproblem $-\Delta u(x,y) = \sin(2\pi x)$ auf $\Omega = (0,1)^2$ mit $u|_{\partial\Omega} = 0$. Zerlegen Sie Ω in $2 \cdot 3^2$ viele Dreiecke und plotten Sie die Lösung: Vergleichen Sie $\mathcal{P}_1$- und $\mathcal{P}_2$-Elemente.

- Modifizieren Sie das Beispielprogramm aus dem FEniCS-Tutorial.
- Nutzen Sie zum Plotten der Lösung den folgenden Code: (Sei `u` die FE-Lösung)

```
import numpy as np
mesh = UnitSquareMesh(30, 30)
x = mesh.coordinates()
tri = mesh.cells()
uEval = np.empty(x.shape[0])
for i in range(0,x.shape[0]):
   uEval[i] = u(x[i,0],x[i,1])
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_trisurf(x[:,0], x[:,1], uEval, triangles=tri,
    cmap=plt.cm.Spectral_r, antialiased=False)
plt.show()
```

Programmieraufgabe 3: (6 Punkte)
Ändern Sie Ihren Code aus der 2. Programmieraufgabe ab, sodass $-\operatorname{div}(\rho\nabla u) = 1$ gelöst wird. Hierbei sei

$$\rho(x) := \begin{cases} 10^6, & x \in [0.3, 0.7]^2, \\ 1, & \text{sonst.} \end{cases}$$

Nutzen Sie $2 \cdot 10^2$ Dreiecke und $\mathcal{P}_1$-Basisfunktionen. Plotten Sie die Lösung.

- Zur Implementierung einer komplexeren Funktion kann eine `UserExpression` genutzt werden. Muster:

```
class Koeffizient(UserExpression):
   def __init__(self, **kwargs):
      UserExpression.__init__(self, **kwargs)
   def eval(self, value, x):
```

```
    # Funktionsdefinition gehoert hierhin.
    # Nutze x[0],...,x[d-1] und setze value[0].
rho = Koeffizient()
```

Abgabe: Bis Mittwoch, 17.04.2019, 14:00 Uhr.