Prof. Dr. Stefan Funken
Dipl.-Math Markus Bantle
Institut für Numerische Mathematik
Universität Ulm

Übungsblatt 8

(Besprechung Do. 4.7. 2013)

Aufgabe 28 *(FEM)*

Wir betrachten die Differentialgleichung

$$-\big(a(x)u'(x)\big)' + b(x)u'(x) + c(x)u(x) = f \quad x \in \Omega = (0,1). \tag{1}$$

für die wir in Aufgabe 26 die schwache Formulierung und die Matrix-Koeffizienten hergeleitet haben.
In dieser Aufgabe beschäftigen wir uns mit der Implementierung. Das Gitter (hier nur 1D) soll in den Matrizen `coordinates` $\in \mathbb{R}^{n_C \times 1}$ und `elements` $\in \mathbb{R}^{n_E \times 2}$ gespeichert werden. `coordinates` enthält die Koordinaten der Gitterpunkte und `elements` enthält für jedes Intervall des Gitters die Indizes der beiden Endpunkte in `coordinates`.

(a) Zeichen Sie (auf einem Blatt Papier) das Gitter für

`coordinates` $= \begin{pmatrix} 0 & 0.5 & 0.3 & 0.6 & 1.0 & 0.9 \end{pmatrix}^T$

`elements` $= \begin{pmatrix} 1 & 3 & 2 & 4 & 6 \\ 3 & 2 & 4 & 6 & 5 \end{pmatrix}^T$

Der Vektor `dirichlet` enthält die Indizes der Dirichlet Knoten, in unserem Fall ist Dirichlet gegeben durch `dirichlet`$= \begin{pmatrix} 1 & 5 \end{pmatrix}^T$. Als nächstes wollen wir uns mit der uniformen Gitter-Verfeinerung beschäftigen. Hierbei wird jedes Element (d.h. Intervall des Gitters) halbiert. Man geht wie folgt vor:

(1) Berechne für jedes Element den Mittelpunkt und hänge die Mittelpunkte hinten an `coordinates` an. (Der Mittelpunkt des ersten (letzten) Elements ist hierbei der erste (letzte) neue Punkt in `coordinates`)

(2) Berechne die neuen Elemente. Hierbei nutzt man aus, dass man weiß in welcher Reihenfolge man die neuen Gitterpunkte hinten an `coordinates` angehängt hat.

In Matlab lässt sich diese Verfeinerung elegant und effizient realisieren.

(b) Schreiben Sie die Matrizen `coordinates` und `elements` für das verfeinerte Gitter von Teil (a) auf. Implementieren Sie eine Funktion

```
[coordinates,elements] = function refineMesh(coordinates,elements)
```

welche die Gitterverfeinerung wie beschrieben durchführt.
Hinweis: Bei geschickter Programmierung benötigen Sie keine `for`-Schleife und nur 2-3 Zeilen Code.

Der nächste Schritt ist die Assemblierung der System-Matrix. Hierbei gehen wir elementweise vor. Auf dem j-ten Element $T_j = [x_j, x_{j+1}]$ des Gitters müssen wir nur die zwei Basis-Funktionen φ_j und φ_{j+1} betrachten, da alle anderen Basis-Funktionen auf diesem Element null sind. Für das j-te Element berechnen wir also für die Matrix A die 2×2 Matrix

$$\begin{pmatrix} \int_{T_j} a(x) \cdot \varphi_j'(x) \cdot \varphi_j'(x)dx & \int_{T_j} a(x) \cdot \varphi_j'(x) \cdot \varphi_{j+1}'(x)dx \\ \int_{T_j} a(x) \cdot \varphi_{j+1}'(x) \cdot \varphi_j'(x)dx & \int_{T_j} a(x) \cdot \varphi_{j+1}'(x) \cdot \varphi_{j+1}'(x)dx \end{pmatrix}.$$

Diese 2×2 Matrix wird dann an entsprechender Stelle auf die Gesamt-Matrix A addiert.

(c) Stellen Sie dieses Vorgehen am Beispiel von Aufgabenteil (a) schematisch auf dem Papier dar.

Für die rechte Seite des Gleichungssystems gehen wir auch elementweise vor.

(d) Schreiben Sie eine Funktion `[A,B,C,b] = assemble(coordinates,elements,f)`, welche die Matrizen A, B und C (siehe Aufgabe 26) und die rechte Seite b elementweise assembliert. `f` ist hierbei ein function handle für die rechte Seite der Differentialgleichung.

(e) Vervollständigen Sie das Skript `main.m` (von der Vorlesungs-Homepage herunterladen), in dem das Gitter eingelesen wird und dann die Differentialgleichung (1) numerisch gelöst wird. Hierbei wird angenommen, dass die Koeffizienten $a(x), b(x), c(x)$ in $\Omega = (0, 1)$ konstant sind. Testen Sie das Programm mit folgenden Daten:

- $a = 1$, $b = 0$, $c = 0$, $f = 1$ und $u(0) = u(1) = 0$,
- $a = 1$, $b = 0$, $c = 0$, $f = 1$ und $u(0) = 0$,$u(1) = 1$,
- $a = 1$, $b = 0$, $c = 0$, $f = x$ und $u(0) = u(1) = 0$,
- $a = 1$, $b = 0$, $c = 1/100$, $f = 1$ und $u(0) = u(1) = 0$,
- $a = 1$, $b = 1/10$, $c = 0$, $f = 1$ und $u(0) = u(1) = 0$.