

Angewandte Numerik 2

Abgabetermin: Freitag 24.01.2014, **vor der Übung**

Raumänderung (Vorankündigung):

Am **07. Februar 2014** finden wegen der Promotionsfeier der Fakultät für Ingenieurwissenschaften und Informatik die Übungen zu Angewandte Numerik 2 im Raum 43.2.103 statt.

Aufgabe 26 (*Programmieraufgabe, Finite-Elemente-Methode in 1D mit Hut-Funktionen*) (16 Punkte)

Programmieren Sie die Finite Elemente Methode für

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

- Leiten Sie die schwache Formulierung der Differentialgleichung her.
- Stellen Sie für die Hut-Basis ein Gleichungssystem der Form $(A + B + C)u = f$ auf.
- Berechnen Sie die Einträge der drei Matrizen A, B und C auf dem Papier. Auf einem Element sollen die Koeffizienten $a(x), b(x)$ und $c(x)$ als konstant (angenähert durch den Wert des Mittelpunkts des jeweiligen Elements) angenommen werden.
- Betrachten Sie nun die gegebene Funktion `solve_1dfem.m` und die vorhandenen Gitterdaten `coordinates.dat`, `elements.dat` und `dirichlet.dat`. Sie finden die entsprechenden Dateien im zip-Archiv `fem1d.zip` auf der Homepage. Versuchen Sie zu verstehen, was im Code gemacht wird!
- Vervollständigen Sie die Routine `fem1d.m`, welche die Finite-Elemente-Lösung von (1) berechnet. Sie dürfen dabei annehmen, dass die Koeffizienten $a(x), b(x)$ und $c(x)$ in $\Omega = (0, 1)$ konstant sind. Achten Sie darauf, dass das Gitter nicht notwendigerweise äquidistante Gitterabstände haben muss.
- 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 $u(0) = u(1) = 0$.

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt11** an angewandte.numerik@uni-ulm.de (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.