101.405 Einführung in Scientific Computing Montag, 23.04.2012
Dirk Praetorius
Michael Karkulik

Übung 5

Aufgabe 1. Betrachten Sie die Differentialgleichung

$$\begin{aligned} -u'' + u' + u &= f \quad \text{auf } (a,b) \\ u(a) = u(b) &= 0. \end{aligned} \tag{1}$$

In der 4. Übung haben sie die Finite-Elemente-Lösung von (1) mit einer uniformen Version berechnet. Heute betrachten wir ein adaptives Verfahren.

1. Sie können annehmen, das der Vektor `coordinates` sortiert ist. Der Elementvektor `elements` muss jedoch nicht sortiert sein. Vergewissern Sie sich, das Ihre Funktion `fem1D` auch in diesem Fall richtig arbeitet.

2. Seien u_ℓ die FE-Lösung zum Gitter $\mathcal{T}_\ell$ und $\widehat{u}_\ell$ die FE-Lösung zum uniform verfeinerten Gitter $\widehat{\mathcal{T}_\ell}$. Man kann den Fehler $|u - u_\ell|^2_{H^1(a,b)}$ schätzen durch $\rho_\ell^2 := |u_\ell - \widehat{u}_\ell|^2_{H^1(a,b)}$. Letzterer Ausdruck ist explizit berechenbar und Summe der lokalen Grössen $\rho_\ell^2(T) = |u_\ell - \widehat{u}_\ell|^2_{H^1(T)}$ für $T \in \mathcal{T}_\ell$. Schreiben Sie eine Funktion `hh2est`, die $(\rho_\ell^2(T))_{T\in\mathcal{T}_\ell}$ berechnet. Input sind die Knotenvektoren zu u_ℓ und $\widehat{u}_\ell$, die beiden Gitter $\mathcal{T}_\ell$ und $\widehat{\mathcal{T}_\ell}$ gegeben durch `coordinates`, `elements` $\in \mathbb{R}^N$ sowie `coordinates_fine` und `elements_fine`. Da wir annehmen das die Elemente nicht sortiert sein müssen, bekommt die Funktion noch ein Array `father2son` $\in \mathbb{R}^{N\times 2}$, welches folgende Information enthält: Wird ein Element $T_i \in \mathcal{T}_\ell$ verfeinert zu den Elementen $T_j \in \widehat{\mathcal{T}_\ell}$ und $T_k \in \widehat{\mathcal{T}_\ell}$, d.h.: $\overline{T_i} = \overline{T_j} \cup \overline{T_k}$, so ist `father2son`$(i) = [j,k]$.
   Wird ein Element $T_i \in \mathcal{T}_\ell$ nicht verfeinert, dann ist `father2son`$(i) = [i,i]$.

3. Laden Sie sich von der Homepage der Übung die Datei `hh2fem1d.m` herunter, welche den adaptiven Algorithmus aus der Vorlesung implementiert. Sie müssen nun zwei Funktionen zur Verfügung stellen: `fem1D` und `hh2est`. Durch Aufruf `hh2fem1D(N)` wird der adaptive Algorithmus mit N Verfeinerungsschritten gestartet. In den Zeilen 15-31 können sie zwischen der glatten Lösung aus der 4. Übung und der singulären Lösung $u(x) = x^{0.9}(1-x)$ wählen. Geplottet wird der H^1-Fehler, der Fehlerschätzer und die optimale Konvergenzordnung. Verwenden Sie die singuläre Funktion auch in ihrer uniformen Version der 4. Übung. Vergleichen Sie die Ergebnisse.

4. Ein adaptiver Algorithmus, der von ρ_ℓ gesteuert wird, hat den entscheidenden Nachteil, das in jedem Schritt $\mathcal{T}_\ell$ sowie das verfeinerte Gitter $\widehat{\mathcal{T}_\ell}$ berechnet werden muss. Ein adaptiver Algorithmus wird aber klarerweise $\widehat{u}_\ell$ zurückliefern, in diesem Sinne ist die Berechnung von u_ℓ nur ein Zwischenergebnis. Verwendet man den Fehlerschätzer

   $$\widetilde{\rho}_\ell^2 = |\widehat{u}_\ell - I_\ell \widehat{u}_\ell|_{H^1(a,b)},$$

   wobei $I_\ell : \mathcal{S}_0^1(\widehat{\mathcal{T}_\ell}) \to \mathcal{S}_0^1(\mathcal{T}_\ell)$ der nodale Interpolationsoperator in den Knoten von $\mathcal{T}_\ell$ ist, so kann die Berechnung von u_ℓ vermieden werden. Schreiben Sie auf Basis von `hh2fem1D.m` eine Funktion `hh2fem1DNew.m`, die einen adaptiven Algorithmus auf Basis von $\widetilde{\rho}_\ell$ implementiert.