Numerical Algorithms

Winter Semester 2014/2015
Dozent: Prof. Dr. Beuchler
Assistent: Katharina Hofer

universitätbonn

Aufgabenblatt 5.

Abgabedatum **Theorie: 11.11.2014, Programm: 18.11.2014**.

1. **C++/C. [6 Punkte.]** Verfeinerung:

   (a) (OPTIONAL: Nur notwendig falls eigener Code verwendet wird): Schreibe eine Routine die ein Netz das aus Elementen und Knoten besteht, erzeugt. Weiters benötigt man eine Routine die Zugriff auf den Platz im Lösungsvektor bzw. auf die Polynomgrade für Kanten bietet. Vergiss nicht, dass sich der Polynomgrad, die Anzahl der Knoten und Kanten und der Platz im Lösungsvektor nach jeder Verfeinerung verändert.

   (b) Implementiere die *hp*-Verfeinerung, d.h. implementiere

   ```
   void Mesh1D::refinehpFEM(const std::vector<int>& hFEM,
     const std::vector<int>& pFEM);
   ```

   Alle Kanten in hFEM sollen h verfeinert werden, alle Kanten in pFEM p verfeinert. (Hinweis: Wenn eine Kante im aktuellen Netz nicht verwendet wird, wird die private Variable solvec_ in Mesh1D::partEdge() auf -10 gesetzt.)

   (c) Aufteilung der Speicherplätze im Lösungsvektor: Jeder Knoten und jede Kante (abhängig vom Polynomgrad) braucht einen Platz im Lösungsvektor. Im globalen Vektor sollen alle Hutfunktionen zuerst auftreten, anschließend folgen die Komponenten der Kantenbasisfunktionen. Implementiere dazu die folgende Routine

   ```
   void Mesh1D::set_solvec();
   ```

   (d) Teste die Routine, verwende dazu das Anfangsintervall $[-1, 1]$ mit einem Dirichletknoten in -1 und einem Neumannknoten in 1. Mache drei h-Verfeinerungen auf der linken Seite (also immer in dem Element das den Punkt -1 enthält), und erhöhe den Polynomgrad bei allen Kanten die nicht h verfeinert werden. Gib eine Skizze des Netzes an (entweder im pdf oder auf Papier, die Informationen die dazu benötigt werden, erhält man mit der print-Methode der Mesh-Klasse).

2. **C++/C. [6 Punkte.]** Assemblierung der rechten Seite:

   (a) Schreibe eine Routine die für die lokalen Speicherplätze die globalen Speicherplätze (im Lösungsvektor) zurückgibt:

   ```
   void FE1D::get_solvecs(Vector & solvec);
   ```

   (b) Implementiere mit obiger Routine die Assemblierung der rechten Seite in

   ```
   void ElementMatVec::assembleRHS(Vector & rhs, Functor1D & func);
   ```

   (c) Verwende das Netz $[-1, 1]$ mache drei h-Verfeinerungen auf der linken Seite (also immer in dem Element das den Punkt -1 enthält), und erhöhe den Polynomgrad bei allen Kanten die nicht h verfeinert werden. Teste die Assemblierung der rechten Seite mit $f_1(x) = 1$ und $f_2(x) = x^2$.

3. **Theoriebeispiel. [6 Punkte.]** Sei $\hat{T}$ das Referenzdreieck mit den Eckpunkten $V_1 = (-1,-1)$, $V_2 = (1,-1)$, $V_3 = (0,1)$. Die Hutfunktionen dieses Dreiecks seien durch die baryzentrischen Koordinaten

$$\lambda_i(V_j) = \delta_{ij} \qquad i,j = 1,2,3$$

gegeben. Die Kantenbubbles auf $\hat{T}$ seien gegeben durch

$$g_i^{E_j}(x,y) := \hat{p}_i^0\left(\frac{\lambda_{e_2} - \lambda_{e_1}}{\lambda_{e_1} + \lambda_{e_2}}\right)(\lambda_{e_1} + \lambda_{e_2})^i \qquad i = 2,\dots,p$$

wobei $E_j = [e_1, e_2]$ die Kante mit den Knoten e_1, e_2 ist und $E_1 = [V_1, V_2], E_2 = [V_2, V_3], E_3 = [V_3, V_1]$. Es sei

$$h_{ij}(x,y) := \hat{p}_j^{2i-1}(2\lambda_3 - 1).$$

Die Elementbubbles auf $\hat{T}$ sind gegeben durch

$$g_{ij}^C(x,y) := h_{ij}(x,y) g_i^{E_1}(x,y)$$

(HINWEIS: $\hat{p}_n^\alpha(-1) = 0$ und $\hat{p}_n^0(1) = 0$ für $n \geq 2$.)

   (a) Sei T das Dreieck mit den Knoten V_2, $V_4 = (2,2)$ und V_3. Bestimme die baryzentrischen Koordinaten für $\hat{T}$ und für T.

   (b) Gib die Basisfunktionen auf $\hat{T}$ und auf T in x und y an.

4. **Theoriebeispiel. [3+7 Punkte.]** Konstruiere analog zu Aufgabe 3) mit Hilfe der baryzentrischen Koordinaten Basisfunktionen für den Referenztetraeder mit den Knoten $V_1 = (-1,-1,-1)$, $V_2 = (1,-1,-1)$, $V_3 = (0,1,-1)$, $V_4 = (0,0,1)$.

   (a) Berechne die baryzentrischen Koordinaten λ_i.

   (b) **ZUSATZ:** Gib die Kantenbubbles, Flächenbubbles und Elementbubbles in Abhängigkeit der baryzentrischen Koordinaten λ_i an.

5. **Theoriebeispiel. [4 Punkte.]** Löse die Differentialgleichung

$$-r\left(rR'(r)\right)' = -\lambda_k^2 R$$

mit $\lambda_k = 2k/3$.

6. **Theoriebeispiel. [5 Punkte.]** Beweise Korollar 1.8 aus der Vorlesung.