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

Übungsblatt 9

(Besprechung Di. 16.7. 2013)

Aufgabe 29 *(2D FEM)*

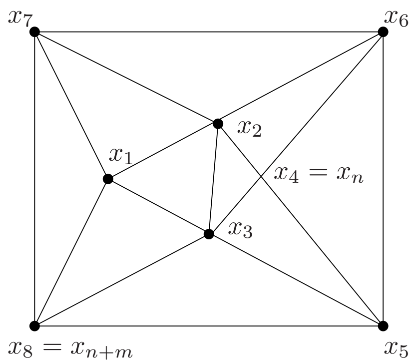
Wir betrachten für $\Omega \subset \mathbb{R}^2$ das Problem:
Finde $u \in C^2(\Omega) \cup C(\overline{\Omega})$ mit

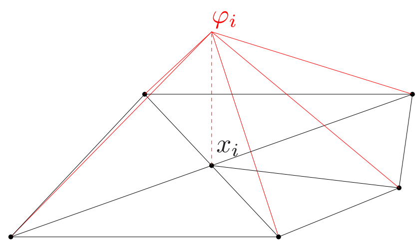
$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \subset \mathbb{R}^2 \\ u &= g \quad \text{auf } \Gamma = \partial\Omega. \end{aligned}$$

Wir schreiben $u = u_0 + u_g$, wobei $u_g \in H^1(\Omega)$ und $u_0 \in H_0^1(\Omega)$. Die schwache Formulierung lautet dann:
Finde $u_0 \in H_0^1(\Omega)$ so, dass

$$\int_\Omega \nabla u_0 \nabla v \, dx = \int_\Omega f v \, dx - \int_\Omega \nabla u_g \nabla v \, dx \quad \forall v \in H^1(\Omega).$$

Um das Problem zu diskretisieren zerlegen wir Ω in Dreiecke (wir nehmen dazu an, dass Ω polygonial berandet ist) und definieren die stückweise lineare Basisfunktion φ_i so, dass $\varphi_i(x_j) = \delta_{i,j}$, wobei x_j Eckpunkt eines Dreiecks ist (siehe Grafik). Nun betrachten wir das schwache Problem auf einem endlich dimensionalen Raum $X_{h,0} \subset H_0^1(\Omega)$ mit





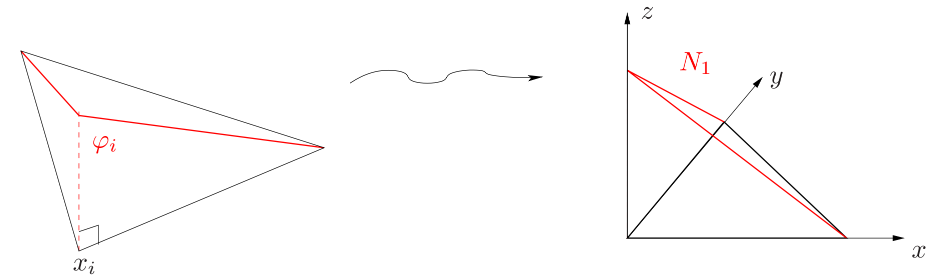
Basis $\{\varphi_1, \ldots, \varphi_n\}$ und ersetzen u_g durch die Interpolierende $u_{g,h} \in X_h \subset H^1(\Omega)$ mit Basis $\{\varphi_1, \ldots, \varphi_{n+m}\}$. Es gilt $u_{g,h} = \sum\limits_{i: x_i \in \Gamma} \varphi_i u_D(x_i)$. Damit erhalten wir das Problem:
Finde $u_{0,h} \in X_{h,0}$

$$\begin{aligned} \int_\Omega \nabla u_{0,h} \nabla v_h \, dx &= \int_\Omega f v_h \, dx - \int_\Omega \nabla u_{g,h} \nabla v_h \, dx \quad \forall v_h \in X_h \\ &= \int_\Omega f v_h \, dx - \sum_{i: x_i \in \Gamma} u_D(x_i) \int_\Omega \nabla \varphi_i \nabla v_h \, dx \quad \forall v_h \in X_h. \end{aligned}$$

Um die Galerkin-Matrix $A = (a_{i,j})$ aufzustellen, gehen wir wie folgt vor:

$$a_{i,j} = \int_\Omega \nabla \varphi_i \nabla \varphi_j dx = \sum_{T \in \operatorname{supp}(\varphi_i) \cap \operatorname{supp}(\varphi_j)} \int_T \nabla \varphi_i \nabla \varphi_j dx.$$

Anstatt nun in einer Schleife über alle Basisfunktionen zu laufen, laufen wir über alle Dreiecke und berechnen für alle Basisfunktionen, die auf dem jeweiligen Dreieck "leben" den Integral-Anteil. Diesen addieren wir dann zu dem Matrix Eintrag an der entsprechenden Stelle in der Galerkin Matrix. Um das Integral über ein Dreieck zu berechnen, transformieren wir das Dreieck T auf das Referenzdreieck $\hat{T} := \operatorname{conv}\{(0,0),(1,0),(0,1)\}$. Auf dem Referenzdreieck sind die shape functions $N_1(s,t) = 1 - s - t$, $N_2(s,t) = s$ und $N_3(s,t) = t$ definiert. Jede shape function entspricht dem Anteil einer Basisfunktion auf dem Dreieck T.



(a) Leiten Sie Formeln für $\int_T \nabla\varphi_i \nabla\varphi_j dx$, $i, j \in 1, 2, 3$, wobei $T = \text{conv}\{x_1, x_2, x_3\}$ ist. (insgesamt 9 Integrale).

Für die Implementierung brauchen wir noch Datenstrukturen um das Gitter aus Dreiecken zu verwalten. Hierfür verwenden wir die Matrizen `coordinates`, `elements` und `dirichlet`. `coordinates` ist eine $(n+m \times 2)$-Matrix, welche die Eckpunkte der Dreiecke enthält, `elements` ist eine $nE \times 3$ Matrix, welche für jedes Dreieck die Nummern der drei Eckpunkte in `coordinates` enthält und `dirichlet` ist eine $nD \times 2$ Matrix, welche für jedes Randstück die Nummern der beiden Eckpunkte in `coordinates` enthält.

(b) Zeichnen Sie das Gitter für die folgenden Datenstrukturen:

$$coordinates = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0.5 & 0.5 \end{pmatrix} \qquad elements = \begin{pmatrix} 1 & 2 & 5 \\ 2 & 3 & 5 \\ 3 & 4 & 5 \\ 4 & 1 & 5 \end{pmatrix} \qquad dirichlet = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 1 \end{pmatrix}$$

(c) Schreiben Sie die Schleife zum Aufstellen der Galerkin-Matrix in Pseudocode auf.

Aufgabe 30

(a) Laden Sie das File fem2d.zip von der homepage herunter. Öffnen Sie die Datei `skript_fem.m` und schauen Sie den Code genau durch. Versuchen Sie zu verstehen was im Einzelnen passiert.

(b) Vervollständigen Sie die Funktion `stima.m` sowie die fehlenden Stellen im Skript `skript_fem.m`.

(c) Lassen Sie ihr Skript für die Beispiele `square` und `lshape` laufen. Testen Sie auch eigene Geometrien.