Institut für Angewandte und Numerische Mathematik
Prof. Dr. Christian Wieners, Dipl.-Math. techn. Martin Sauter

Numerische Mathematik III — Programmier-Übungsblatt 3

Sommersemester 2010

Sei $\Omega \subset \mathbb{R}^2$ ein Polygongebiet und betrachten Sie auf Ω das Poisson-Problem

$$-\Delta u(x) = f(x) \quad \text{in} \quad \Omega\,, \qquad u(x) = g(x) \quad \text{auf} \quad \partial\Omega\,.$$

$\mathcal{T}_h = \{K_1, \dots, K_M\}$ sei eine zulässige Triangulierung von Ω in M Dreiecke. Ein Dreieck K_m ist durch die konvexe Hülle seiner Knotenpunkte $\mathcal{V}_m = \{z^{m,0}, z^{m,1}, z^{m,2}\}$ beschrieben, d.h. $K_m = \operatorname{conv} \mathcal{V}_m$. Die Menge aller Knotenpunkte ist $\mathcal{V}_h = \bigcup_{m=1}^M \mathcal{V}_m$. Zudem sei $|\mathcal{V}_h| = N$ die Anzahl der Gesamt-Knotenpunkte.

Zur Triangulierung $\mathcal{T}_h$ ist $\mathcal{S}_h \equiv \mathcal{S}_h^1 = \{v \in C(\overline{\Omega}) : v|_K \in \mathbb{P}_1 \ \text{ für alle } \ K \in \mathcal{T}_h\}$ der Raum der stückweise linearen Finite Elemente. Dieser lässt sich durch die Knotenbasis $\{\phi_z\}_{z\in\mathcal{V}_h} \subset \mathcal{S}_h$ beschreiben.

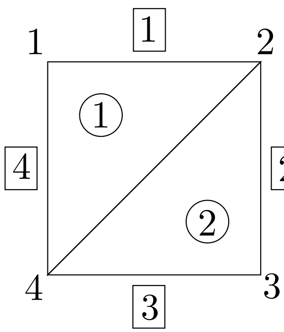
$$\mathcal{S}_h = \operatorname{span}\{\phi_z : z \in \mathcal{V}_h\}\,, \qquad \text{wobei} \qquad \phi_z(y) = \begin{cases} 1 & , z = y\,, \\ 0 & , z \neq y\,, \end{cases} \quad \text{für alle } \ y \in \mathcal{V}_h\,.$$

Der Träger einer Basisfunktion ist somit $\operatorname{supp}(\phi_z) = \bigcup_{\{m\in\mathbb{N}: z\in\mathcal{V}_m\}} K_m$. Damit lässt sich jedes $v_h \in \mathcal{S}_h$ durch $v_h = \sum_{z\in\mathcal{V}_h} v_h(z)\phi_z$ darstellen und nach einer Nummerierung der Knotenpunkte $\mathcal{V}_h$ erhält man die Darstellung $v_h = \sum_{k=1}^N \underline{v}_k \phi_k$, d.h. $v_h \in \mathcal{S}_h$ lässt sich durch einen Vektor $\underline{v} \in \mathbb{R}^N$ darstellen.

Für stetige Funktionen ist zudem der Knoten-Interpolationsoperator $I_h : C(\overline{\Omega}) \to \mathcal{S}_h$ durch $I_h v = \sum_{z\in\mathcal{V}} v(z)\phi_z \in \mathcal{S}_h$ definiert.

Beispieltriangulierung: $\Omega = (-1,1)^2$ wird durch zwei Dreiecke, vier Knotenpunkte und vier Randkanten beschrieben.

vertices		edges		triangles		
-1	1	1	2	2	1	4
1	1	2	3	3	2	4
1	-1	3	4			
-1	-1	4	1			



Die Triangulierung wird durch die drei Matrizen `vertices`, `edges` und `triangles` beschrieben.

Matrix	Dim.	Inhalt
`vertices`	$\mathbb{R}^{N,2}$	Knotenpunkte in $\mathcal{V}_h$ (N Knotenpunkte)
`edges`	$\mathbb{R}^{E,2}$	Indizes der Eckpunkte der Randkanten (E Randkanten)
`triangles`	$\mathbb{R}^{M,3}$	Indizes der Dreiecks-Knotenpunkte (M Dreiecke)

Die Beispielgeometrie $\Omega = (a,b)^2$ können Sie mit der Matlab-Funktion

```
[vertices,edges,triangles] = fe_rect(r,a,b)
```

erstellen. Dabei ist `r` die Anzahl der Verfeinerungen der Anfangstriangulierung (`r=0` entspricht dabei der Triangulierung in zwei Dreiecke). Zudem können Sie mit

```
fe_plot(vertices,edges,triangles,u)
```

eine Finite Element Funktion $u_h \in \mathcal{S}_h$ über ihre Darstellung $\underline{u} \in \mathbb{R}^N$ zeichnen. Dabei muss `u(k)`$= u_h($`vertices(k,:)`$)$ gelten.

Die Funktionen `fe_rect` und `fe_plot` finden Sie auf der Homepage der Vorlesung.

Berechnung von Integralen: Oftmals muss auf Ω ein Integral berechnet werden. Dies geschieht i.A. näherungsweise anhand einer Quadraturformel auf der Triangulierung. Mit $\widehat{z}^0 = (0,0)$, $\widehat{z}^1 = (1,0)$ und $\widehat{z}^2 = (0,1)$ sei das Referenzdreieck $\widehat{K} = \operatorname{conv}\{\widehat{z}^0, \widehat{z}^1, \widehat{z}^2\}$ definiert. Ein beliebiges Dreieck $K = \operatorname{conv}\{z^0, z^1, z^2\}$ lässt sich dann über die affine Transformation $\varphi_K : \widehat{K} \to K$,

$$\varphi_K(\widehat{x}) = F_K\widehat{x} + d_K \qquad \text{mit} \quad F_K = \begin{bmatrix} z^1 - z^0 & z^2 - z^0 \end{bmatrix} \in \mathbb{R}^{2,2}\,, \quad d_K = z^0 \in \mathbb{R}^2\,,$$

darstellen. Insbesondere gilt $\varphi_K(\widehat{z}^i) = z^i$ und mit $\varphi_m \equiv \varphi_{K_m}$, $F_m \equiv F_{K_m}$, $J_m \equiv \det D\varphi_{K_m} = \det F_m$ und dem Transformationssatz für Integrale folgt

$$\int_\Omega f(x)\,dx = \sum_{m=1}^M \int_{K_m} f(x)\,dx = \sum_{m=1}^M \int_{\widehat{K}} f\big(\varphi_m(\widehat{x})\big)\,|J_m(\widehat{x})|\,d\widehat{x}\,.$$

Dies bedeutet, dass das Integral elementweise berechnet werden kann und die Quadratur auf einem Dreieck auf eine Quadraturformel auf dem Referenzdreieck $\widehat{K}$, wie z.B.

$$\int_{\widehat{K}} f(\widehat{x})\,d\widehat{x} \approx \tfrac{1}{6}\Big(f(\tfrac{1}{2},0) + f(\tfrac{1}{2},\tfrac{1}{2}) + f(0,\tfrac{1}{2})\Big)\,,$$

zurückgeführt werden kann.

bitte wenden

Aufgabe 7 **(Abgabe: 27. Mai)**

Erzeugen Sie mit `fe_rect` eine Triangulierung von $\Omega = (0,1)^2$ für unterschiedliche Stufen der Gitterverfeinerung und zeichnen Sie folgende Funktionen.

(a) Die Basisfunktion $\phi_{(0,0)}$ und für r≥ 1 auch $\phi_{(\frac{1}{2},\frac{1}{2})}$ und $\phi_{(\frac{1}{2},0)}$.

(b) Die Funktion $I_h u$, wobei $u(x) = \sin(\pi x_1)(x_2^2 - 1)x_2$ sei (vgl. Aufgabe 1). Schreiben Sie dazu eine Funktion

```
u = fe_interpol(vertices,funchandle),
```

die den Knoten-Interpolationsoperator implementiert. Dabei ist `funchandle` ein sog. "function handle" (ähnlich einem Zeiger auf eine Funktion in C++). Falls bspw. die Funktion u in `u_func.m` implementiert ist, übergeben Sie `@u_func`. Innerhalb von `fe_interpol` können Sie via `funchandle(...)` die übergebene Funktion auswerten.

Aufgabe 8 (mündlich)

Sei $u_h \in \mathcal{S}_h$ eine Finite Element-Funktion. Schreiben Sie eine Matlab-Funktion, die für beliebiges $x \in \Omega$ sowohl $u_h(x)$ als auch $Du_h(x)$ berechnet.

Aufgabe 9 **(Abgabe 10. Juni)**

Für hinreichend glatte Funktionen $u : \Omega \to \mathbb{R}$ gelten mit dem Knoten-Interpolationsoperator $I_h : C(\overline{\Omega}) \to \mathcal{S}_h$ die Abschätzungen

$$\|u - I_h u\|_0 \leq C_1\, h^2\, \|u\|_2\,,$$
$$\|u - I_h u\|_1 \leq C_2\, h\, \|u\|_2\,,$$

mit Konstanten $C_1, C_2 > 0$ und der Gitterweite $h = \max_{K\in\mathcal{T}_h} \operatorname{diam}(K)$. Schreiben Sie eine Matlab-Funktion

```
[L2fehler,H1fehler] = fe_fehler(vertices,triangles,u,u_exacthandle),
```

die sowohl den L^2-Fehler $\|u - I_h u\|_0$, als auch den H^1-Fehler $\|u - I_h u\|_1$ zu einer gegebenen Triangulierung berechnet. Berechnen Sie dazu $u_h = I_h u$ mithilfe der Funktion `fe_interpol` aus Aufgabe 7, und übergeben Sie anschließend u_h und das "function handle" der Funktion an `fe_fehler`.

Benutzen Sie für die Berechnung des Integrals die oben angegebene Quadraturformel. Verwenden Sie zu Testzwecken die Funktion u aus Aufgabe 7(b) und die Lösung des Interface-Problems aus Aufgabe 6. Welche Konvergenzordnungen beobachten Sie jeweils?

Organisatorisches:

Am **Donnerstag, 20. Mai** wird erneut die Vorlesung und der erste Teil des Rechnerpraktikums getauscht, d.h. um 9.45 Uhr findet der theoretische Teil des Praktikums statt, und von 15.45–17.15 Uhr findet die Vorlesung statt. Anschließend beginnt der praktische Teil des Rechnerpraktikums im Pool-Raum.

Attestierung:

Die mit **(Abgabe)** markierten Aufgaben können Sie sich im Rechnerpraktikum attestieren lassen. Dieses findet immer **donnerstags, ab 15.45 Uhr** statt. Soweit über die Mailingliste nichts anderes bekannt gegeben wird, ist zuerst der Seminarraum SR1 Treffpunkt.

Service/Material:

Unter http://www.math.kit.edu/ianm3/lehre/numa32010s/ finden Sie die Homepage zur Vorlesung. Dort werden neben den aktuellen Praktikumsblättern in unregelmäßigen Abständen auch Übersichtsfolien und Beispielprogramme zur Vorlesung bereitgestellt.

Rechnerpraktikum:

Zur aktiven Teilnahme am Rechnerpraktikum müssen Sie sich registrieren. Den entsprechenden Link finden Sie auf der Homepage.

Sprechstunden:

Prof. Dr. Christian Wieners: Mittwoch, 10.00-12.00 Uhr.
Dipl.-Math. techn. Martin Sauter: Donnerstag, 10.00-11.30 Uhr.