Technische Universität Graz
Institut für Angewandte Mathematik
Prof. Dr. Olaf Steinbach
Julia Hauser

Numerische Mathematik 3

Finite Elemente, 4. Mai 2020

Punktesystem

Da alle restlichen Übungsblätter mit einer digitalen Abgabe verbunden sind, führen wir ein Punktesystem ein. Die Punkte werden in eure Note einfließen und eine ungefähre Benotung vorgeben. Diese könnt ihr am Ende mit dem Prüfungsgespräch beeinflussen. Wie wir dieses abhalten werden, wird von den Umständen am Ende des Semesters abhängen.

Abgabeform

Bitte kopiert eure Ergebnisse in ein seperates Text/LaTex/Pdf-File und kommentiert diese dort. Da sehr viele verschiedene Versionen von Matlab und Octave in diesem Kurs verwendet werden und ich teilweise ein anderes Betriebsystem verwende, können einige Formatierungen verloren gehen. Deswegen kopiert eure Ergebnisse in ein seperates File, sodass diese nicht verloren gehen können.

Teil 1 - 1D Finite Element Methode

Im letzten Übungsblatt haben wir betrachtet wie eine Yukawa-Gleichung mit Neumannbedingungen in 1D gelöst werden kann. Jedoch haben wir die Approximationseigenschaften bis jetzt lediglich aus Bildern abgeleitet. Nun werden wir ebenso den $L^2(\Omega)$- und $H^1(\Omega)$- Fehler betrachten. Also den Fehler im Funktionenraum der Ansatz- und Testfunktionen. Dafür betrachten wir verschiedene rechte Seiten und die Fehler der resultierenden Lösungen.

Aufgabe 10: (4 Punkte) Erweitere die Routine `LoeseNeumann1d()` von Aufgabe 7 auf das inhomogene Neumann-Randwertproblem

$$-u''(x) + u(x) = f(x) \quad \text{für } x \in (a,b), \qquad -u'(a) = g_N(a) =: \texttt{na}, \quad u'(b) = g_N(b) =: \texttt{nb}$$

mit rechter Seite $f \in L^2(a,b)$ und Neumann-Datum $g_N\colon \{a,b\} \to \mathbb{R}$. Dazu implementiere in MATLAB/Octave die Assemblierung der rechten Seite $\vec{F}_f \in \mathbb{R}^M$,

$$\vec{F}_f[k] := \int_a^b f(x)\varphi_k(x)\mathrm{d}x, \quad k = 1,\dots,M,$$

mit Hilfe der lokalen Vektoren

$$\vec{F}_f^l := \begin{pmatrix} \int_{x_{l_1}}^{x_{l_2}} f(x)\varphi_{l_1}(x)\mathrm{d}x \\ \int_{x_{l_1}}^{x_{l_2}} f(x)\varphi_{l_2}(x)\mathrm{d}x \end{pmatrix}$$

und der Benutzung der Mittelpunktsregel

$$\int_{x_{l_1}}^{x_{l_2}} g(x)\mathrm{d}x \approx (x_{l_2} - x_{l_1})\, g\left(\frac{x_{l_1} + x_{l_2}}{2}\right)$$

auf jedem Element (x_{l_1}, x_{l_2}). Weiters implementiere in MATLAB/Octave die Berechnung von

$$\|u - u_h\|_{L^2(a,b)} \quad \text{und} \quad \|u' - u_h'\|_{L^2(a,b)}.$$

Dafür laufe über die Elemente τ_l, transformiere auf das Referenzelement und verwende eine 3-Punkt-Gauß-Formel, siehe die Quadraturformeln für Simplexe im Zusatz vom 2. Übungsblatt.

Teste auf $(a,b) := (0,2)$ die Implementierung für das gleichmäßige Netz mit den Knoten $x_k = 2\frac{k-1}{M-1}$ für $k = 1,\dots,M$. Dazu erstelle eine Fehlertabelle für $M = 2^{L+1} + 1$ für $L = 0,\dots,10$ in der Form

L	M	$\|u-u_h\|_{L^2(0,2)}$	eoc	$\|u-u_h\|_{H^1(0,2)}$	eoc
0	3	1.1340e+00	-	3.7455e+00	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

für die Lösung $u(x) = -2x^3 + 9x^2 - 12x + 4$.

Aufgabe 11: (4 Punkte) Betrachte weiterhin die Gleichung von Aufgabe 10 für die eindeutigen Lösungen

$$u_1(x) = x^{7/4} \qquad \text{und} \qquad u_2(x) = x^{5/4} \quad \text{für } x \in (0,2) =: (a,b).$$

a) Zeige, dass $u_1 \in H^2(0,2)$, $u_2 \in H^1(0,2)$ und $u_2 \notin H^2(0,2)$ gelten.

b) Erstelle Fehlertabellen wie in Aufgabe 10 für diese Lösungen für die Level $L = 0, \ldots, 12$, wenn für den lokalen Lastenvektor $\vec{F}_f^l$ einmal die Mittelpunktsregel und einmal die 3-Punkt-Gauß-Formel verwendet wird.

c) Für diese entsprechenden rechten Seiten f_1 und f_2 ist eine analytische Integration möglich. Berechne die analytische Integration für f_1 und f_2 (zum Beispiel mit Mathematica/Wolfram Alpha) und implementiere in MATLAB der lokalen Lastenvektoren $\vec{F}_{f_1}^l$ und $\vec{F}_{f_2}^l$. Erstelle wieder die Fehlertabellen wie in Aufgabe 10 für die Lösungen von Level $L = 0, \ldots, 12$, wobei die analytische Integration für den lokalen Lastenvektor verwendet werden soll.

Teil 2 - 2D Finite Element Methode

Nun möchten wir ebenso 2D Probleme lösen. Dafür bauen wir einen kompletten Code für die Yukawa-Gleichung mit Neumanndaten am Rand in Aufgabe 12. Danach betrachten wir die Poisson Gleichung mit Dirichlet- und Neumanndaten. Dabei könnt ihr sehen, wie unterschiedlich die Gleichungen gehandhabt werden. Warum Stabilisationen benötigt werden, werdet ihr in der Vorlesung erfahren.

Aufgabe 12: (6 Punkte) Betrachte das Neumann-Randwertproblem der Yukawa-Gleichung

$$-\Delta u(x) + u(x) = f(x) \quad \text{für } x \in \Omega, \qquad \frac{\partial u}{\partial n}(x) = g_N(x) \quad \text{für } x \in \partial\Omega =: \Gamma,$$

wobei $f \in L^2(\Omega)$ und $g_N \in L^2(\Gamma)$ gegebene Funktionen sind. Die eindeutig lösbare Variationsformulierung dazu ist: Variationsformulierung ist:
Gesucht ist $u \in H^1(\Omega)$ sodass

$$\int_\Omega \nabla u \cdot \nabla v \; dx + \int_\Omega uv \; dx = \int_\Omega fv \; dx + \int_{\partial\Omega} g_N v \; ds_x$$

für alle $v \in H^1(\Omega)$.
Für das Einheitsquadrat $\Omega := (0,1)^2$ betrachte die Triangulierung $\mathcal{T}_N$, welche durch die Dateien `coordinates.dat`, `elements.dat` und `neumann.dat` beschrieben wird. Die Dateien `coordinates.dat` und `elements.dat` haben die gleiche Struktur wie im 1D Fall. In der Datei `neumann.dat` beschreibt die j-te Zeile die Indizes j_1 und j_2 der Knoten der Neumann-Kante e_j.

a) Schreibe die Routine

```
function [A_el] = stiff2dEl(x_l1, x_l2, x_l3)
% Berechnung der lokalen Steifigkeitsmatrix für das Dreieck mit
% den Eckpunkten x_l1, x_l2, x_l3
%
% x_l1 ... Erster Eckpunkt des Dreiecks
% x_l2 ... Zweiter Eckpunkt des Dreiecks
% x_l3 ... Dritter Eckpunkt des Dreiecks
% A_el ... Lokale Steifigkeitsmatrix des Dreiecks
```

um die lokale Steifigkeitsmatrix gegeben durch

$$\begin{aligned} A_h^l[i,j] &= \int_{\mathcal{T}_l} \nabla\varphi_{l_j}(x)\cdot\nabla\varphi_{l_i}(x)\mathrm{d}x \\ &= \frac{|\det J_l|}{2}(J_l^{-T}\nabla_\xi\psi_j)\cdot(J_l^{-1}\nabla_\xi\psi_i) \end{aligned}$$

zu erstellen. Teste diese für das gegebene Dreieck $x_{l_1}=(0,0)^\top$, $x_{l_2}=(1,0)^\top$, $x_{l_3}=(0,1)^\top$.

b) Implementiere mit Hilfe der Funktionen `mass2dEl` aus Aufgabe 8 und `stiff2dEl` die Assemblierung der (globalen) Massematrix $M_h \in \mathbb{R}^{M\times M}$ und der (globalen) Steifigkeitsmatrix $A_h \in \mathbb{R}^{M\times M}$,. Laufe dafür über die Elemente in `elements.dat`.Teste die Assemblierungen mit den gegebenen Netzdaten und vergleiche die Matrizen mit den zur Verfügung gestellten Ergebnissen.

c) Implementiere mit Hilfe der Funktionen `rhs2dEl` und `rhsNeumann2dEl` aus Aufgabe 8 die Assemblierung der rechten Seite $\vec{F}=\vec{F}_f+\vec{F}_N\in\mathbb{R}^M$. Teste mit den gegebenen Netzdaten, $f(x_1,x_2)=g_N(x_1,x_2)=1$ und vergleiche die Vektoren mit den zur Verfügung gestellten Ergebnissen.

d) Wähle f und g_N, sodass $u(x_1,x_2)=x_1^2$ Lösung des oben gestellten Randwertproblems ist. Löse das resultierende Gleichungssystem der Diskretisierung und plotte die exakte Lösung u (`surf`) und die Näherungslösung $u_h\in S_h^1(\mathcal{T}_N)$ mit dem Befehl

```
trisurf(elements,coordinates(:,1),coordinates(:,2),uh,'facecolor','interp').
```

e) Es gibt verschiedene Möglichkeiten zu überprüfen bzw. festzustellen, ob in Berechnungen tatsächlich eine bestimmte („vorhergesagte“) Konvergenzordnung für ein Verfahren erreicht wird. Gehe von der Annahme aus, dass das („vorhergesagte“) Fehlerverhalten dem (idealem) Gesetz $\mathrm{err}(h)=Ch^s$ mit $s>0$, $C>0$, dem Diskretisierungsparameter (zB „Maschenweite“) $h>0$ und dem Fehler $\mathrm{err}(h)$ gehorcht.
Angenommen, für zwei verschiedene Diskretisierungen mit positiven Parametern $h_1\neq h_2$ seien die Fehler $\mathrm{err}(h_1)>0$ und $\mathrm{err}(h_2)>0$ berechenbar. Zeige, dass sich dann s durch

$$s=\frac{\ln\frac{\mathrm{err}(h_1)}{\mathrm{err}(h_2)}}{\ln\frac{h_1}{h_2}}=:\mathrm{eoc}$$

bestimmen lässt, wobei eoc als experimentelle Konvergenzordnung (experimental order of convergence) bezeichnet wird. (Hier wird angenommen, dass alle Rechnungen exakt durchgeführt werden können.)

f) Verwende die Routinen von Aufgabe 9 und erstelle eine Fehlertabelle für die Verfeinerungslevel $L=0,\ldots,7$ in der Form

L	M	$\|u-u_h\|_{L^2(\Omega)}$	eoc	$\|u-u_h\|_{H^1(\Omega)}$	eoc
0	9	2.6487e-02	-	2.7811e-01	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

für die exakte Lösung $u(x) = x_1^2$ für $\Omega = (0,1)^2$. Achte darauf, die Wurzel erst beim Gesamtfehler zu ziehen. Als Ausgangstriangulierung verwende das gegebene Netz und verfeinere gleichmäßig mit der gegebenen Routine `refineMesh()`, welche `provideGeometricData` verwendet. Die Funktion `provideGeometricData` weist den Meshdaten die zugehörigen Knoten und Kanten zu, aber wird für das Beispiel nicht benötigt.
Was ist zu beobachten?

Aufgabe 13: (4 Punkte) Implementiere in MATLAB das Galerkin-Variationsproblem für die Poisson-Gleichung mit inhomogenen Dirichlet-Randbedingungen

$$-\Delta u = f \text{ in } \Omega, \quad \gamma_0^{\text{int}} u = g_D \text{ auf } \partial\Omega$$

mit $f \in L^2(\Omega)$ und $g_D \in H^{1/2}(\partial\Omega)$ für stückweise lineare und global stetige Ansatzfunktionen für Gebiete mit polygonaler Berandung in $\mathbb{R}^2$. Es sei vorausgesetzt, dass die Dirichlet-Randdaten g_D stetig sind. Gehe dafür wie in 1D vor. Das heißt, die approximierte Lösung ist durch

$$u_h(x) = \sum_{x_k \in \Omega} \overline{u}_k \varphi_k(x) + \sum_{x_k \in \partial\Omega} g_D(x_k)\varphi_k(x)$$

mit den Knoten x_k gegeben, wobei die $\overline{u}_k$ zu berechnen sind. Um die Freiheitsgrade und Randknoten zu bestimmen, verwende man

```
freenodes = setdiff(1:M, unique(dirichlet));
boundary = setdiff(1:M, freenodes);
```

mit der Anzahl M der Knoten x_k. Ordne die Matrix und rechte Seite nach den Freiheitsgraden und Randknoten. Schreibe das geordnete System auf und überlege, welche Zeilen wegfallen. (Vergleiche den 1D Fall aus dem Tutorium.) Forme das restliche System so um, dass nur noch die Matrix `A(freenodes,freenodes)` auf der linken Seite steht. Teste das Programm für die Lösungen $u(x) = x_1^2 + x_2^2$ und $u(x) = (x_1^2 + x_2^2)^{0.3}$, mittels das gegebene Netz. Verfeinere gleichmäßig und erstelle eine Tabellen in der Form von Aufgabe 12f) erstellt. Was ist zu beobachten?

Aufgabe 14: (4 Punkte) Implementiere in MATLAB das Galerkin-Variationsproblem für die Poisson-Gleichung mit inhomogenen Neumann-Randbedingungen

$$-\Delta u = f \text{ in } \Omega, \quad \frac{\partial u}{\partial n} = g_N \text{ auf } \partial\Omega$$

mit $f \in L^2(\Omega)$ und $g_N \in L^2(\partial\Omega)$ für stückweise lineare und global stetige Ansatzfunktionen für Gebiete mit polygonaler Berandung in $\mathbb{R}^2$. Dazu verwende die folgende Variationsformulierung:
Gesucht ist $u \in H^1(\Omega)$, sodass

$$\langle \nabla u, \nabla v \rangle_\Omega + \langle u, \mathbf{1} \rangle_\Omega \langle v, \mathbf{1} \rangle_\Omega = \langle f, v \rangle_\Omega + \left\langle g_N, \gamma_0^{\text{int}} v \right\rangle_{\partial\Omega}$$

für alle $v \in H^1(\Omega)$ gilt.
Leite das lineare Gleichungssystem

$$(A_h + \vec{a}\,\vec{a}^\top)\vec{u} = \vec{F}$$

mit dem Stabilisierungsvektor $\vec{a} \in \mathbb{R}^M$,

$$\vec{a}[j] = \int_\Omega \varphi_j(x)\mathrm{d}x, \quad j = 1, \dots, M,$$

her. Führe die Assemblierung des Stabilisierungsvektors $\vec{a}$ wieder elementweise durch, zeige dabei dass

$$\int_{\tau_l} \varphi_j(x)\mathrm{d}x = \frac{1}{6}\,|\det J_l| \qquad \text{für } x_j \in \overline{\tau_l}$$

gilt. Teste das Programm mit dem Netz von Aufgabe 12 für die Lösung $u(x) = x_1^2 - \frac{1}{3}$,. Verfeinere gleichmäßig und erstelle eine Tabelle in der Form von Aufgabe 12f).