Ausgabe: 31.10.2019

# Übung 4

Abgabe: 07.11.2019, 12:00 Uhr

**Aufgabe 9** (20 Punkte)

Betrachten Sie das Poisson-Problem

$$\begin{aligned} -\Delta u &= \sin(2\pi x) \quad &&\text{in } \Omega = (0,1) \\ u &= 0 &&\text{auf } \partial\Omega = \{0,1\} \end{aligned}$$

mit der variationellen Formulierung

$$\text{Finde } u \in H_0^1(\Omega): \quad a(u,\varphi) := \int_\Omega u'(x)\,\varphi'(x)\,\mathrm{d}x = \int_\Omega \sin(2\pi x)\,\varphi(x)\,\mathrm{d}x =: L(\varphi) \quad \forall \varphi \in H_0^1(\Omega). \tag{1}$$

Ziel dieser Aufgabe ist die Implementierung eines Galerkin-Verfahrens mit B-Splines zur Lösung von (1). Die B-Splines der Ordnung $k$ auf dem Intervall $[a,b]$ verwenden dabei einen erweiterten Knotenvektor

$$\theta_1 = \ldots = \theta_k = a < \theta_{k+1} \le \theta_{k+2} \ldots \le \theta_n < b = \theta_{n+1} = \ldots = \theta_{n+k}.$$

a) Schreiben Sie eine Funktion `v = Nik(x, i, knots, order, diff)`, welche die `diff`-te Ableitung $N_{i,k}^{(\mathtt{diff})}$ einer B-Spline Basisfunktion $N_{i,k}$ der Ordnung $k = \mathtt{order}$ an der Stelle $x$ auswertet. Der erweiterte Knotenvektor wird in `knots` übergeben.

   Nutzen Sie hierzu die rekursive Darstellung der Ableitung

$$N'_{i,k}(x) = (k-1)\left[\frac{N_{i,k-1}(x)}{\theta_{i+k-1} - \theta_i} - \frac{N_{i+1,k-1}(x)}{\theta_{i+k} - \theta_{i+1}}\right],$$

   wobei die Quotienten auf 0 gesetzt werden, sollte der jeweilige Nenner 0 werden (zusammenfallende Knoten).

   **Hinweis:** Verwenden Sie zur Auswertung der $N_{i,k}$ das Neville-Schema aus Übung 3.

b) Schreiben Sie eine Funktion `A = MatAssembly(knots, order, diff, startI, endI)`, welche die Matrix $\mathbf{A} = (a_{ij})_{i,j=\mathtt{startI}}^{\mathtt{endI}}$ mit

$$a_{ij} = \int_a^b N_{j,k}^{(\mathtt{diff})} N_{i,k}^{(\mathtt{diff})}\,\mathrm{d}x \qquad\qquad i,j = \mathtt{startI}, \ldots, \mathtt{endI}$$

   aufbaut, wobei `diff` wieder den Grad der Ableitung angibt. Beachten Sie, dass nur die B-Splines mit $i,j = \mathtt{startI}, \ldots, \mathtt{endI}$ in der Matrix berücksichtigt werden.

   **Hinweis:** Verwenden Sie, dass $\operatorname{supp}(N_{i,k}) = [\theta_i, \theta_{i+k}]$ und berechnen Sie daher nur die Integrale der $N_{i,k}$, $N_{j,k}$ mit überlappendem Träger. Benutzen Sie dazu wiederum, dass die $N_{i,k}$ (und deren Ableitungen) auf den Intervallstücken $[\theta_i, \theta_{i+1}]$ Polynome sind und sich daher die Integrale mit Gauß-Quadratur exakt berechnen lassen. Überlegen Sie sich dazu die benötigte Anzahl an Gauß-Knoten in Abhängigkeit von Ordnung $k = \mathtt{order}$ und Differentiationsgrad `diff`.

c) Schreiben Sie eine Funktion `b = VecAssembly(g, knots, order, startI, endI, gaussLegNodes)`, welche den Vektor $\mathbf{b} = (b_i)_{i=\mathtt{startI}}^{\mathtt{endI}}$ mit

$$b_i = \int_a^b g(x) N_{i,k}(x)\,\mathrm{d}x \qquad\qquad i = \mathtt{startI}, \ldots, \mathtt{endI}$$

   aufbaut, wobei $g\colon [a,b] \to \mathbb{R}$ durch das Function-Handle `g` gegeben ist. Beachten Sie, dass nur die B-Splines mit $i = \mathtt{startI}, \ldots, \mathtt{endI}$ in dem Vektor berücksichtigt werden. Da $g$ unbekannt ist, kann die Anzahl der Gauß-Knoten für die Quadratur nicht automatisch bestimmt werden und wird vom Benutzer in `gaussLegNodes` übergeben.

   **Hinweis:** Verwenden Sie, dass $\operatorname{supp}(N_{i,k}) = [\theta_i, \theta_{i+k}]$ und berechnen Sie daher nur die Integrale über den jeweiligen Träger der $N_{i,k}$ statt über ganz $[a,b]$. Teilen Sie dazu wiederum das Integral über den Träger in die Teilstücke $[\theta_{i+s}, \theta_{i+s+1}]$, $s = 0, \ldots, k-1$, auf und approximieren Sie jedes dieser Teilintegrale mittels Gauß-Quadratur.

d) Für Level $J \in \mathbb{N}$ und Ordnung $k \geq 2$ wird nun der äquidistante Knotenvektor

$$\theta_1 = \ldots = \theta_k = 0 < 2^{-J} < 2 \cdot 2^{-J} < 3 \cdot 2^{-J} < \ldots < 1 - 2^{-J} < 1 = \theta_{n+1} = \ldots = \theta_{n+k} \tag{2}$$

verwendet. Lösen Sie unter Verwendung der oben erstellen Funktionen das Problem (1) für $k = 2, \ldots, 5$ und $J = 2, \ldots, 6$.

Berechnen Sie für jede Kombination $(J, k)$ den $L_2(\Omega)$-Fehler der numerischen Lösung $u_{J,k}$ bezüglich der exakten Lösung

$$u(x) = \frac{\sin(2\pi x)}{4\pi^2}.$$

Plotten Sie für jede Ordnung $k$ eine Fehlerkurve (Gitterweite $2^{-J}$ gegen $\|u_{J,k} - u\|_{L_2(\Omega)}$ für $J = 2, \ldots, 6$) in einen gemeinsamen doppelt-logarithmischen (`loglog`) Plot.

Schätzen Sie für jede Ordnung $k$ und jeweils zwei aufeinanderfolgende Level $J - 1$, $J$ die Konvergenzrate

$$\mathrm{eoc}(J, k) = \frac{\ln\left(\frac{\|u_{J,k} - u\|_{L_2(\Omega)}}{\|u_{J-1,k} - u\|_{L_2(\Omega)}}\right)}{\ln\left(\frac{2^{-J}}{2^{-(J-1)}}\right)}.$$

Geben Sie eine Tabelle mit den (empirischen) Konvergenzraten $\mathrm{eoc}(J, k)$ aus:

| $J$ | $k = 2$ | $k = 3$ | $k = 4$ | $k = 5$ |
|---|---|---|---|---|
| 3 | | | | |
| $\vdots$ | | | | |
| 6 | | | | |

**Hinweis:** Zur Einhaltung der Nullrandwerte entfernen Sie die erste und letzte B-Spline Basisfunktion, sodass nur die inneren Basisfunktionen verwendet werden. Nutzen Sie hierzu die Parameter `startI` und `endI` aus den obigen Funktionen.

e) Führen Sie die gleiche Untersuchung aus Teil d) für das Funktional

$$\tilde{L}(\varphi) := \int_\Omega x^{-1/4}\, \varphi(x)\, \mathrm{d}x, \qquad \varphi \in H_0^1(\Omega)$$

anstelle von $L$ in (1) durch. Die exakte Lösung von $a(\tilde{u}, \varphi) = \tilde{L}(\varphi)$ für alle $\varphi \in H_0^1(\Omega)$ ist

$$\tilde{u}(x) = -\frac{16}{21} x(x^{3/4} - 1).$$

Erklären Sie Ihre Beobachtungen.

**Hilfsmittel:**

- Wenn Sie Ihre eigene Implementation des Neville-Schemas aus Übung 3 nicht benutzen möchten, können Sie eine (geschützte, vektorisierte) Implementation von der Homepage der Vorlesung herunterladen. Diese hat die Signatur `v = Neville(x, coeffs, knots, order)`.
- Auf der Homepage zur Vorlesung finden Sie die geschützte Implementation der Gauß-Quadratur mit Signatur `I = gaussInt(f, a, b, nodes)` zur Berechnung von

$$\int_a^b f(x)\, \mathrm{d}x.$$

  Dabei ist `f` ein Function-Handle des Integranden (vektorisiert). Für die Anzahl der Gauß-Knoten gilt $1 \leq$ `nodes` $\leq 8$. Polynome bis zum Grad $2 \cdot$ `nodes` $- 1$ werden exakt integriert.
- Falls Ihre Implementierung eines Aufgabenteils nicht funktioniert, können Sie die entsprechende (geschützte) Implementierung von der Homepage der Vorlesung verwenden.