Ausgabe: 07.11.2019

Übung 5

Abgabe: 14.11.2019, 12:00 Uhr

Aufgabe 10 ($2 + 4 + 4 + 4 = 14$ Punkte)

Betrachten Sie das Poisson-Problem

$$\begin{aligned} -\Delta u &= 13\pi^2 \sin(2\pi x) \sin(3\pi y) \qquad && \text{in } \Omega = (0,1) \times (0,1) \\ u &= 0 && \text{auf } \partial\Omega \end{aligned}$$

mit der variationellen Formulierung

$$\begin{aligned} &\text{Finde } u \in H_0^1(\Omega): \\ &a(u,\varphi) := \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla \varphi(\mathbf{x}) \,\mathrm{d}\mathbf{x} = \int_\Omega 13\pi^2 \sin(2\pi x_1) \sin(3\pi x_2)\ \varphi(\mathbf{x}) \,\mathrm{d}\mathbf{x} =: L(\varphi) \quad \forall \varphi \in H_0^1(\Omega). \end{aligned} \tag{1}$$

Ziel dieser Aufgabe ist die Implementierung eines Galerkin-Verfahrens mit Tensorprodukt B-Splines zur Lösung von (1). Für einen Level $J \in \mathbb{N}$ verwenden die B-Splines $N_{i,k}$ der Ordnung k auf dem Intervall $[0,1]$ dabei den erweiterten äquidistanten Knotenvektor

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

Als endlich-dimensionalen Teilraum $V_J \subset H_0^1(\Omega)$ zum Level J werden Linearkombinationen von Tensorprodukt B-Splines $N_{i,k}$ verwendet, wobei jeweils die erste und letzte Basisfunktion ausgelassen werden.

$$V_J := \operatorname{span}\left\{ N_{\mathbf{i},k} \,\middle|\, N_{\mathbf{i},k} \colon \Omega \to \mathbb{R},\ (x,y) \mapsto N_{i_1,k}(x)\ N_{i_2,k}(y);\quad \mathbf{i} = (i_1,i_2);\ i_1, i_2 = 2, \ldots, 2^J + k - 2 \right\}. \tag{2}$$

Die Lösung $u_J \in V_J$ hat die Darstellung

$$u_J(x,y) := \sum_{i_1,i_2=2}^{2^J+k-2} u_{J,(i_1,i_2)}\ N_{(i_1,i_2),k}(x,y) = \sum_{i_1=2}^{2^J+k-2} \sum_{i_2=2}^{2^J+k-2} u_{J,(i_1,i_2)}\ N_{i_1,k}(x)\ N_{i_2,k}(y), \tag{3}$$

wobei die Koeffizienten in lexikographischer Reihenfolge sortiert werden:

$$\mathbf{u}_J := (u_{J,(2,2)}, u_{J,(2,3)}, \ldots, u_{J,(2,2^J+k-2)}, u_{J,(3,2)}, \ldots, u_{J,(2^J+k-2,2^J+k-2)})^T.$$

a) Im Galerkin-Verfahren für (1) entsteht das lineare System

$$\mathbf{A}_J \mathbf{u}_J = \mathbf{b}_J.$$

Zeigen Sie, dass sich $\mathbf{A}_J$ darstellen lässt als

$$\mathbf{A}_J = \mathbf{S}_J \otimes \mathbf{M}_J + \mathbf{M}_J \otimes \mathbf{S}_J,$$

wobei $\mathbf{S}_J$ die *Steifigkeitsmatrix* und $\mathbf{M}_J$ die *Massenmatrix* in einer Raumdimension ist:

$$(\mathbf{S}_J)_{i,j} := \int_0^1 \frac{\mathrm{d}N_{i,k}}{\mathrm{d}x} \frac{\mathrm{d}N_{j,k}}{\mathrm{d}x} \,\mathrm{d}x, \qquad (\mathbf{M}_J)_{i,j} := \int_0^1 N_{i,k}\, N_{j,k} \,\mathrm{d}x.$$

Hierbei bezeichnet allgemein

$$\mathbf{B} \otimes \mathbf{C} := \begin{bmatrix} b_{11}\mathbf{C} & \cdots & b_{1n}\mathbf{C} \\ \vdots & \ddots & \vdots \\ b_{m1}\mathbf{C} & \cdots & b_{mn}\mathbf{C} \end{bmatrix} \in \mathbb{R}^{mp \times nq}$$

das *Kronecker-Produkt* (manchmal auch *Tensorprodukt* genannt) zweier Matrizen $\mathbf{B} \in \mathbb{R}^{m \times n}$ und $\mathbf{C} \in \mathbb{R}^{p \times q}$.

b) Schreiben Sie eine Funktion `b = VecAssembly2D(g, knots, order, startI, endI, gaussLegNodes)`, welche den Vektor $\mathbf{b}_J = (b_{\mathbf{i}})$ mit

$$b_{\mathbf{i}} = \int_\Omega g(x_1, x_2) N_{i_1,k}(x_1) N_{i_2,k}(x_2)\,\mathrm{d}x, \qquad \mathbf{i} = (i_1, i_2) = (\texttt{startI}, \texttt{startI}), \ldots, (\texttt{endI}, \texttt{endI})$$

aufbaut, wobei $g\colon \Omega \to \mathbb{R}$ durch das Function-Handle `g` gegeben ist. Beachten Sie, dass nur die B-Splines mit $i_1, i_2 = \texttt{startI}, \ldots, \texttt{endI}$ in dem Vektor berücksichtigt werden. Die Anzahl der Gauß-Knoten für die Quadratur 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 $N_{\mathbf{i},k}$ statt über ganz Ω. Teilen Sie dazu wiederum das Integral über den Träger in die Teilstücke $[\theta_{i_1+s}, \theta_{i_1+s+1}] \times [\theta_{i_2+\ell}, \theta_{i_2+\ell+1}]$, $s, \ell = 0, \ldots, k-1$, auf und approximieren Sie jedes dieser Teilintegrale mittels Tensorprodukt Gauß-Quadratur.

c) Schreiben Sie eine Funktion `v = SplineInt2D(g, coeffs, knots, order, startI, endI, gaussLegNodes)`, welche ein Integral der Form

$$\mathsf{v} \approx \int_\Omega g(x_1, x_2, s(\mathbf{x}))\,\mathrm{d}\mathbf{x}$$

approximiert, wobei $g\colon \Omega \times \mathbb{R} \to \mathbb{R}$ beliebig ist und der Spline s die folgende Darstellung hat:

$$s(\mathbf{x}) = \sum_{i_1, i_2 = \texttt{startI}}^{\texttt{endI}} \texttt{coeffs}_{(i_1,i_2)}\ N_{(i_1,i_2),\texttt{order}}(x_1, x_2) = \sum_{i_1=\texttt{startI}}^{\texttt{endI}} \sum_{i_2=\texttt{startI}}^{\texttt{endI}} \texttt{coeffs}_{(i_1,i_2)}\ N_{i_1,\texttt{order}}(x_1)\ N_{i_2,\texttt{order}}(x_2).$$

Hinweis: Teilen Sie das Gebiet in die Teilstücke $[\theta_{i_1}, \theta_{i_1+1}] \times [\theta_{i_2}, \theta_{i_2+1}]$, $i_1, i_2 = k, \ldots, n$, auf und approximieren Sie jedes Teilintegral mittels Tensorprodukt Gauß-Quadratur. Verwenden Sie dabei, dass $\operatorname{supp}(N_{i,k}) = [\theta_i, \theta_{i+k}]$ und somit nur wenige $N_{(i_1,i_2),k}$ in diesem Teilstück nicht verschwinden.

d) Lösen Sie unter Verwendung der oben erstellten Funktionen das Problem (1) für $k = 2, \ldots, 4$ und $J = 2, \ldots, 5$. Berechnen Sie für jede Kombination (J, k) den $L_2(\Omega)$-Fehler der numerischen Lösung $u_{J,k}$ bezüglich der exakten Lösung

$$u(x, y) = \sin(2\pi x)\sin(3\pi y).$$

Plotten Sie für jede Ordnung k eine Fehlerkurve (Gitterweite 2^{-J} gegen $\|u_{J,k} - u\|_{L_2(\Omega)}$ für $J = 2, \ldots, 5$) in einen gemeinsamen doppelt-logarithmischen (`loglog`) Plot.

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

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

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

J	$k = 2$	$k = 3$	$k = 4$
3			
$\vdots$			
5			

Berechnen Sie ferner die Kondition $\kappa(\mathbf{A}_J)$ der Matrix $\mathbf{A}_J$ für jede Kombination (J, k) mittels der Matlab Funktion `condest`. Tragen Sie für jede Ordnung k die Kondition $\kappa(\mathbf{A}_J)$ gegen die Gitterweite 2^{-J} für $J = 2, \ldots, 5$ in einem gemeinsamen doppelt-logarithmischen (`loglog`) Plot ab.

Hinweis:

- Zur Einhaltung der Nullrandwerte entfernen Sie in jeder Dimension jeweils die erste und letzte B-Spline Basisfunktion, sodass nur die inneren Basisfunktionen verwendet werden (siehe (2)). Nutzen Sie hierzu die Parameter `startI` und `endI` aus den obigen Funktionen.
- Achten Sie darauf, dass die Steifigkeitsmatrix $\mathbf{S}_J$ und Massematrix $\mathbf{A}_J$ *sparse* sind.
- Verwenden Sie `SplineInt2D` zur Auswertung des $L_2(\Omega)$-Fehlers mit einer entsprechenden Funktion `g`.

Hilfsmittel:

- Wenn Sie Ihre Funktionen aus Übung 3 und 4 nicht benutzen möchten, können Sie (geschützte, vektorisierte) Implementationen von der Homepage der Vorlesung herunterladen.
- Auf der Homepage zur Vorlesung finden Sie die geschützte Implementation der Tensorprodukt Gauß-Quadratur mit Signatur `I = gaussInt2D(f, a, b, nodes)` zur Berechnung von

  $$\int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x, y) \, \mathrm{d}y \, \mathrm{d}x.$$

  Dabei ist `f` ein Function-Handle des Integranden (vektorisiert). Die oberen und unteren Grenzen werden in den Vektoren $\mathtt{b} = [b_1, b_2]$ und $\mathtt{a} = [a_1, a_2]$ angegeben. Für die Anzahl der Gauß-Knoten gilt $1 \leq \mathtt{nodes} \leq 8$. Polynome bis zum Grad $2 \cdot \mathtt{nodes} - 1$ werden exakt integriert.

  Alternativ kann die Funktion benutzt werden, um die Stützstellen und Gewichte der Quadraturformel zu berechnen. Der Aufruf `[X, W] = gaussInt2D(f, a, b, nodes)` liefert eine Matrix $\mathtt{X} \in \mathbb{R}^{\mathtt{nodes}^2 \times 2}$ mit einer Stützstelle pro Zeile und einen Vektor aus Gewichten $\mathtt{W} \in \mathbb{R}^{\mathtt{nodes}^2}$, sodass

  $$\int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x, y) \, \mathrm{d}y \, \mathrm{d}x \approx \sum_{i=1}^{\mathtt{nodes}} \mathtt{W}_i f(\mathtt{X}_{i,1}, \mathtt{X}_{i,2}).$$

- Falls Ihre Implementierung eines Aufgabenteils nicht funktioniert, können Sie die entsprechende (geschützte) Implementierung von der Homepage der Vorlesung verwenden.

Aufgabe 11 (6 Punkte)

Betrachten Sie auf dem Intervall $[a, b]$ einen kubischen Spline ($k = 4$) zum erweiterten Knotenvektor:

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

Der Spline $S \in \mathcal{S}_{k,\Delta}$ interpoliere die Funktion $f \in C^2([a, b])$ an den Stellen $x_i = \theta_{i+3}$ für $i = 1, \ldots, n-2$, d.h.

$$S(x_i) = f(x_i), \qquad i = 1, \ldots, n.$$

Die verbleibenden zwei Freiheitsgrade werden durch die folgende Forderung (vollständiger kubischer Spline) festgelegt:

$$S'(a) = f'(a), \qquad S'(b) = f'(b).$$

Sei $h \coloneqq \max\{|x_{i+1} - x_i| : i = 1, \ldots, n-3\}$ der maximale Abstand zwischen zwei benachbarten Stützstellen.

a) Zeigen Sie, dass $\displaystyle\int_a^b (f'' - S'')S'' \, \mathrm{d}x = 0$.

b) Zeigen Sie, dass $\|f' - S'\|_{\infty,[a,b]} \leq \sqrt{h}\|f''\|_{L_2([a,b])}$, wobei $\|f\|_{\infty,[a,b]} \coloneqq \sup_{x\in[a,b]}|f(x)|$ die Supremumsnorm bezeichnet.

   Hinweis: Satz von Rolle, Hauptsatz der Differential- und Integralrechnung, Teil a).

c) Zeigen Sie, dass $\|f - S\|_{\infty,[a,b]} \leq \dfrac{1}{2}h^{3/2}\|f''\|_{L_2([a,b])}$.