Prof. Dr. Angela Kunoth
Samuel Leweke

Abschlussprojekt: Adaptivität

In diesem Projekt soll das Gitter zum Lösen einer PDE mit einem adaptiven Verfahren an die exakte Lösung angepasst werden. Um die Genauigkeit der numerischen Lösung einer PDE zu erhöhen, kann das Gitter gleichmäßig verfeinert werden. Auf diese Weise können Strukturen in der Lösung (große Änderungen in Bereichen des Rechengebiets) global aufgelöst werden. Dieses Vorgehen ist jedoch ineffizient, da auch Teilgebiete, in denen sich die Lösung kaum ändert, höher aufgelöst werden und die Gitterweite von den feinsten Strukturen der Lösung diktiert wird.

Ein effizienteres Vorgehen besteht also darin, dort die Auflösung des Gitters zu erhöhen, wo die Lösung besondere Strukturen aufweist. In den Teilbereichen, in denen die Lösung glatt ist, kann selbige auch mit einem grobem Gitter ausreichend genau approximiert werden. Diese Anpassung des Gitters – und damit die angepasste Platzierung von Freiheitsgraden – wird von adaptiven Algorithmen realisiert.

Ein adaptives Verfahren besteht aus mehreren Subroutinen:

- `SOLVE`: Löst die diskretisierte PDE auf einem gegebenen Gitter.
- `ESTIMATE`: Schätze den lokalen Fehler der numerischen Lösung auf dem aktuellen Gitter. Dieser Schritt wird durch einen *a posteriori* Fehlerschätzer realisiert.
- `MARK`: Markiere die Gitterzellen, welche einen hohen Fehler aufweisen und verfeinert werden sollen.
- `REFINE`: Erzeuge aus der Markierung ein neues, lokal verfeinertes Gitter.

Beginnend mit einem groben (z. B. uniformen) Startgitter wird dieser Zyklus von Subroutinen so lange iteriert, bis eine vorgegebene Fehlerschranke unterschritten oder eine maximale Iterationszahl erreicht wird.

In diesem Projekt soll die eindimensionale PDE

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega = (0,1) \\ u &= 0 \qquad \text{auf } \partial\Omega \end{aligned} \tag{1}$$

adaptiv mit B-Splines gelöst werden. Dazu wird die schwache Formulierung

$$\text{Finde } u \in H_0^1(\Omega): \qquad a(u,v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}x =: L(v) \qquad \forall v \in H_0^1(\Omega) \tag{2}$$

verwendet. Der endlich-dimensionale Teilraum $V_h(\theta) \subset H_0^1(\Omega)$ ist durch B-Splines $N_{i,k,\theta}$ der Ordnung $k \geq 2$ mit nicht-uniformen Knotenvektor θ, der

$$\theta_1 = \ldots = \theta_k = 0 < \theta_{k+1} < \ldots < \theta_n < 1 = \theta_{n+1} = \ldots = \theta_{n+k} \tag{3}$$

erfüllt, gegeben. Um die Nullrandbedingung zu erfüllen, müssen die erste und letzte Basisfunktion entfernt werden:

$$V_h(\theta) := \operatorname{span}\{N_{i,k,\theta} : i = 2, \ldots, \#\theta - k - 1\} \subset H_0^1(\Omega). \tag{4}$$

Im Folgenden wird das Subskript k in $N_{i,k,\theta}$ zur besseren Übersichtlichkeit weggelassen.

Die PDE wird nun zu gegebenem Gitter (repräsentiert durch den Knotenvektor θ) und zugehörigem Teilraum $V_h(\theta)$ durch Einschränken der schwachen Formulierung (2) auf $V_h(\theta)$ diskretisiert. Dies führt auf ein lineares Gleichungssystem

$$\mathbf{A}\mathbf{u}_\theta = \mathbf{b} \tag{5}$$

mit

$$a_{ij} = \int_0^1 N'_{j,\theta}(x) N'_{i,\theta}(x) \,\mathrm{d}x, \qquad b_i = \int_0^1 f(x) N_{i,\theta}(x) \,\mathrm{d}x. \tag{6}$$

Das Lösen von (5) realisiert `SOLVE`.

Als nächstes muss in `ESTIMATE` der Fehler von

$$u_\theta = \sum_{i=2}^{\#\theta-k-1} (\mathbf{u}_\theta)_i \, N_{i,\theta} \in V_h(\theta) \tag{7}$$

geschätzt werden. Dazu wird das Residuum $r\colon \Omega \to \mathbb{R}$ definiert als $r = \Delta u_\theta + f$. Es kann gezeigt werden, dass es für Splineräume $V_h(\theta) \subset C^1(\Omega)$ eine vom Gitter θ unabhängige konstante $C > 0$ gibt, sodass

$$\|u - u_\theta\|_{H_0^1(\Omega)} \leq C\varepsilon_\theta\left(u_\theta\right), \qquad \varepsilon_\theta^2\left(u_\theta\right) := \sum_{i=1}^{\#\theta-1} \left(\theta_{i+1} - \theta_i\right)^2 \|r\|_{L_2([\theta_i,\theta_{i+1}])}^2, \tag{8}$$

wobei $u \in H_0^1(\Omega)$ die exakte Lösung von (1) ist. Die Größe

$$\varepsilon_{\theta,i}^2\left(u_\theta\right) := \left(\theta_{i+1} - \theta_i\right)^2 \|r\|_{L_2([\theta_i,\theta_{i+1}])}^2, \qquad i = 1, \ldots, \#\theta - 1, \tag{9}$$

liefert somit einen Indikator für den Beitrag eines Knotenspanns $[\theta_i, \theta_{i+1}]$ zum Gesamtfehler.

Mit dem Fehlerindikator $\varepsilon_{\theta,i}\left(u_\theta\right)$ können diejenigen Teilintervalle $[\theta_i, \theta_{i+1}]$ verfeinert werden, die am meisten zum Fehler beitragen. Eine häufig gewählte Strategie für `MARK` ist das sogenannte *Dörfler-Marking*. Hierbei werden Teilintervalle $\mathcal{M} \subset \{1, \ldots, \#\theta - 1\}$ mit den größten Beiträgen $\varepsilon_{\theta,i}\left(u_\theta\right)$ markiert bis

$$\sqrt{\sum_{i\in\mathcal{M}} \varepsilon_{\theta,i}^2\left(u_\theta\right)} \geq \varrho\, \varepsilon_\theta\left(u_\theta\right) \tag{10}$$

für ein gegebenes $\varrho \in (0, 1]$ erfüllt ist.

Im letzten Schritt, `REFINE`, wird ein neues Gitter $\tilde{\theta}$ aus θ und den Markierungen $\mathcal{M}$ erzeugt. In diesem Projekt wird ein ausgewähltes Teilintervall $[\theta_i, \theta_{i+1}]$ mit $i \in \mathcal{M}$ einfach in zwei gleich große Stücke $[\theta_i, (\theta_i + \theta_{i+1})/2]$ und $[(\theta_i + \theta_{i+1})/2, \theta_{i+1}]$ zerteilt. Das heißt, zwischen θ_i und θ_{i+1} wird der Knoten $(\theta_i + \theta_{i+1})/2$ eingefügt.

Aufgaben

a) Schreiben Sie eine Matlab-Funktion `v = EvalSpline(x, knots, order, coeffs, diff)`, welche die `diff`-te Ableitung (`diff` ≥ 0) des B-Splines

$$S_{\texttt{knots}}(x) = \sum_{i=1}^{\#\texttt{knots}-\texttt{order}} \texttt{coeffs}_i \, N_{i,\texttt{order},\texttt{knots}}^{(\texttt{diff})}(x) \tag{11}$$

auswertet. Die Auswertung erfolgt an allen Punkten des (beliebigen) Arrays `x` und der Rückgabewert `v` ist ein Array derselben Größe.

b) Schreiben Sie eine Matlab-Funktion `coeffs = Solve(knots, order, rhsF)`, welche zu einem gegebenen Gitter $\theta =$ `knots` das zur PDE (1) gehörende lineare System (5) löst. Die rechte Seite f der PDE (1) wird als Function Handle in `rhsF` übergeben. Die Koeffizienten der diskreten Lösung u_θ aus (7) werden als Vektor `coeffs` zurückgegeben.

   Hinweis: Verwenden Sie die Funktionen aus Übung 4.

c) Schreiben Sie eine Matlab-Funktion `err = Estimate(knots, order, coeffs, rhsF)`, die den Fehlerindikator (9) auf allen Teilintervallen $i = 1, \ldots, \#\theta - 1$ auswertet und als Vektor `err` zurückliefert. Die Koeffizienten der diskreten Lösung u_θ werden in `coeffs` und die rechte Seite f der PDE (1) wird als Function Handle in `rhsF` übergeben.

d) Schreiben Sie eine Matlab-Funktion `idxRef = Mark(knots, err, rho)`, welche diejenigen Teilintervalle $[\theta_i, \theta_{i+1}]$ auswählt, die verfeinert werden sollen. Der Fehlerindikator wird als Vektor `err` übergeben und `rho` bezeichnet den Parameter ρ in (10). Der Rückgabewert `idxRef` kann ein Vektor mit den Indizes der zu verfeinernden Teilintervalle sein, oder ein Vektor aus `true / false`, der für jedes Intervall festlegt, ob es verfeinert werden soll (`idxRef(i) == true` bedeutet, dass $[\theta_i, \theta_{i+1}]$ verfeinert werden soll).

e) Schreiben Sie eine Matlab-Funktion `knotsRef = Refine(knots, idxRef)`, welche das aktuelle Gitter $\theta =$ `knots` in den durch `idxRef` markierten Intervallen verfeinert. Die Verfeinerung wird durch Halbierung der markierten Intervalle erreicht. Das neue Gitter wird als Knotenvektor `knotsRef` zurückgegeben.

f) Schreiben Sie ein Matlab-Programm, welches die PDE (1) mit rechter Seite

$$f(x) = x^{-1/4} \tag{12}$$

und analytischer Lösung

$$u(x) = \frac{16}{21} x \left(1 - x^{3/4}\right), \qquad u'(x) = -\frac{4}{21}\left(7x^{3/4} - 4\right) \tag{13}$$

numerisch löst. Das Problem soll sowohl mit uniformen Gittern der Level $J = 3, \ldots, 8$ als auch adaptiv mit $\rho = 0.9$ und 10 Iterationen des `SOLVE-ESTIMATE-MARK-REFINE` Zyklus ausgehend von einem uniformen Startgitter zum Level $J = 3$.

Speichern Sie in jeder Iteration des adaptiven Verfahrens das aktuelle Gitter. Erzeugen Sie am Ende der Iteration einen Plot der Gitter, indem Sie die Punkte $(\theta_i^{(j)}, j)$ in einen gemeinsamen Plot zeichnen, wobei j der Index der adaptiven Iteration und $\theta_i^{(j)}$ der i-te Knoten im Knotenvektor $\theta^{(j)}$ aus selbiger Iteration j ist.

Zeichnen Sie jeweils für jede Iteration (bzw. jedes Level) den $L_2(\Omega)$-Fehler und den Fehler in der $H_0^1(\Omega)$-Seminorm

$$|v|_{H_0^1(\Omega)} := \sqrt{\int_\Omega (v')^2 \,\mathrm{d}x} \tag{14}$$

auf und stellen Sie diese für beide Verfeinerungsarten in einem gemeinsamen doppelt-logarithmischen Plot dar, wobei die die Fehler gegen die Anzahl an Freiheitsgraden (Länge des Vektors `coeffs`) aufgetragen werden sollen.

Berechnen Sie empirische Konvergenzraten (siehe z. B. Übung 4) für beide Normen und Verfeinerungsarten und geben Sie diese als Tabelle aus.

Führen sie den obigen Vergleich für quadratische ($k =$ `order` $= 3$) und kubische Splines ($k = 4$) durch.

Literatur

[BG16] Annalisa Buffa und Carlotta Giannelli. „Adaptive isogeometric methods with hierarchical splines: Error estimator and convergence“. *Mathematical Models and Methods in Applied Sciences* 26.01 (Jan. 2016), S. 1–25. DOI: 10.1142/S0218202516500019.