Prof. Dr. Angela Kunoth
Samuel Leweke

Abschlussprojekt: Isogeometric Analysis

In diesem Projekt soll das Paradigma der *isogeometric analysis* (kurz IGA) eingeführt werden. Dadurch können PDEs auf komplexen Geometrien gelöst werden, die üblicherweise in CAD-Software (*Computer Aided Design*) erstellt werden. Während traditionelle (isoparametrische) Finite Elemente Verfahren solche Geometrien mit ihrem Gitter nur approximieren können, wird die Geometrie bei IGA exakt berücksichtigt.

Auf einem (komplexen) Gebiet $\Omega \subset \mathbb{R}^2$ soll die partielle Differentialgleichung

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

gelöst werden. Dieses *physikalische Gebiet* Ω wird durch eine Abbildung $\mathbf{F}\colon \hat{\Omega} \to \Omega$ parametrisiert, wobei $\hat{\Omega} = [0,1]^2$ das *parametrische Gebiet* genannt wird.

In diesem Projekt soll die Abbildung $\mathbf{F}$ durch eine B-Spline Oberfläche der Form

$$\mathbf{F}(\mathbf{x}) = \sum_{\mathbf{i}} \mathbf{C}_{\mathbf{i}} N_{i_1}(x_1) N_{i_2}(x_2), \qquad \mathbf{x} = (x_1, x_2) \in \hat{\Omega},\ \mathbf{i} = (i_1, i_2) \in \mathcal{I}_1 \times \mathcal{I}_2, \tag{2}$$

gegeben sein, wobei die Punkte $\mathbf{C}_{\mathbf{i}} \in \mathbb{R}^2$ *Kontrollpunkte* genannt werden und $\mathcal{I}$ eine Indexmenge bezeichnet. Die Abbildung $\mathbf{F}$ ist also eine Art Linearkombination von Tensorprodukt B-Splines, wobei die Koeffizienten Punkte im $\mathbb{R}^2$ sind. Im Folgenden nehmen wir an, dass die Abbildung $\mathbf{F}$ und ihre Umkehrabbildung glatt sind ($\mathbf{F}$ ist ein Diffeomorphismus).

In der IGA wird ein *isoparametrischer* Ansatz für den diskreten Teilraum $\hat{V}_h \subset H_0^1(\hat{\Omega})$ gewählt. Das bedeutet, dass der Raum für die Geometrie auch für die Approximation benutzt wird und die beiden mit durch die Abbildung $\mathbf{F}$ miteinander verknüpft werden.

$$\hat{V}_h = \operatorname{span}\{\mathbf{x} \mapsto N_{\mathbf{i}}(\mathbf{x}) = N_{i_1}(x_1) N_{i_2}(x_2) : \mathbf{i} = (i_1, i_2) \in \mathcal{I}_1 \times \mathcal{I}_2\} \tag{3}$$

$$V_h = \operatorname{span}\{N_{\mathbf{i}} \circ \mathbf{F}^{-1} : N_{\mathbf{i}} \in \hat{V}_h\}. \tag{4}$$

Hier ist der parametrische Raum $\hat{V}_h \subset H_0^1(\hat{\Omega})$ mit den gleichen (offenen) Knotenvektoren definiert, die auch für $\mathbf{F}$ in (2) verwendet werden. Der Raum $\hat{V}_h$ wird dann mittels $\mathbf{F}$ auf das physikalische Gebiet Ω abgebildet, wodurch $V_h \subset H_0^1(\Omega)$ entsteht.

Die Tatsache, dass die gleichen Räume für die Darstellung der Geometrie und die Approximation benutzt werden, ist der Grund für den Zusatz „iso-geometrisch". Hierbei ist jedoch hervorzuheben, dass die Räume nicht exakt gleich sein müssen. Vielmehr muss der geometrische Raum aus (2) im Approximationsraum $\hat{V}_h$ enthalten sein ($\hat{V}_h$ darf also deutlich feiner sein).

Dieser Ansatz wird gewählt, weil sichergestellt werden soll, dass V_h beliebige affine Funktionen enthält. Dies scheint insbesondere in der linearen Elastizitätstheorie wichtig zu sein (sogenannte *patch tests*). Ist $A\colon \mathbb{R}^2 \to \mathbb{R}$ eine beliebige affine Funktion, so ist $A \circ \mathbf{F}$ ein Tensorprodukt B-Spline aus $\hat{V}_h$ dessen Koeffizienten aus $A(\mathbf{C}_{\mathbf{i}})$ bestehen. Dann ist aber $(A \circ \mathbf{F}) \circ \mathbf{F}^{-1} = A \in V_h$.

Das weitere Vorgehen entspricht dem gewöhnlichen Galerkin-Verfahren. Die Bilinearform $a\colon H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ und das Funktional $L\colon H_0^1(\Omega) \to \mathbb{R}$, die zur PDE (1) gehören, werden auf den endlich-dimensionalen Teilraum V_h eingeschränkt und so ein lineares Gleichungssystem

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

gewonnen. Hierbei ist zu beachten, dass mittels des Transformationssatzes vom physikalischen Gebiet Ω zum parametrischen Gebiet $\hat{\Omega}$ gewechselt wird. Für $\mathbf{i}, \mathbf{j} \in \mathcal{I}_1 \times \mathcal{I}_2$ gilt also

$$\begin{aligned} a_{\mathbf{i},\mathbf{j}} &= \int_{\Omega} \nabla N_{\mathbf{j}}(\mathbf{F}^{-1}(\mathbf{x})) \cdot \nabla N_{\mathbf{i}}(\mathbf{F}^{-1}(\mathbf{x}))\, \mathrm{d}\mathbf{x} \\ &= \int_{\hat{\Omega}} \left(\left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right)^{-T} \nabla N_{\mathbf{j}}(\mathbf{x}) \right) \cdot \left(\left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right)^{-T} \nabla N_{\mathbf{i}}(\mathbf{x}) \right) \left| \det \left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right) \right| \mathrm{d}\mathbf{x}, \end{aligned} \tag{6}$$

$$b_{\mathbf{i}} = \int_{\Omega} f(\mathbf{x}) N_{\mathbf{i}}(\mathbf{F}^{-1}(\mathbf{x}))\, \mathrm{d}\mathbf{x} = \int_{\hat{\Omega}} f(\mathbf{F}(\mathbf{x})) N_{\mathbf{i}}(\mathbf{x}) \left| \det \left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right) \right| \mathrm{d}\mathbf{x}. \tag{7}$$

Die Tensorprodukt-Struktur der Basisfunktionen in $\hat{V}_h$ kann hier, im Gegensatz zu den Übungsaufgaben, nicht mehr ausgenutzt werden um die Matrix $\mathbf{A}$ in (6) aus den 1D-Matrizen mittels Kronecker-Produkt aufzubauen.

Aufgaben

a) Zeigen Sie, dass $A \circ \mathbf{F}$ ein Tensorprodukt B-Spline mit Koeffizienten $A(\mathbf{C_i})$ ist.

b) Zeigen Sie mittels des Transformationssatzes die Gleichungen (6) und (7).

c) Schreiben Sie eine Matlab-Funktion `[physX, physY] = isogeoMap(knotsGeoX, orderGeoX, knotsGeoY, orderGeoY, ctrlPts, paramX, paramY)`, die die Abbildung $\mathbf{F}$ aus (2) umsetzt. Die Tensorprodukt B-Splines bestehen dabei aus dem B-Spline mit Knotenvektor `knotsGeoX` und Ordnung `orderGeoX` in x-Richtung und aus dem B-Spline mit Knotenvektor `knotsGeoY` und Ordnung `orderGeoY` in y-Richtung. Die Kontrollpunkte $C_{\mathbf{i}}$ sind in der zweispaltigen Matrix `ctrlPts` gegeben, wobei jede Zeile einen Punkt enthält. Die Abbildung $\mathbf{F}$ soll an den Punkten (`paramX`, `paramY`) ausgewertet werden, die beliebige Arrays (mit gleicher Größe) sein können. Das Ergebnis sind die Arrays `physX` und `physY`, welche die gleiche Größe wie `paramX` und `paramY` haben.

   Verwenden Sie die lexikographische Sortierung für die Kontrollpunkte und die Nummerierung der tensorierten Basisfunktionen.

d) Schreiben Sie eine Matlab-Funktion `J = isogeoMapJac(knotsGeoX, orderGeoX, knotsGeoY, orderGeoY, ctrlPts, paramX, paramY)`, die die Jacobi-Matrix $\mathrm{d}\mathbf{F}/\mathrm{d}\mathbf{x}$ der Abbildung $\mathbf{F}$ aus (2) umsetzt. Die Parameter ergeben sich wie in Teil c). Das Ergebnis ist ein 3D-Array `J`, welches in den ersten beiden Dimensionen jeweils die Jacobi-Matrix an einem Punkt enthält. Das heißt, `J(:, :, k)` enthält die Jacobi-Matrix am Punkt (`paramX(k)`, `paramY(k)`).

e) Schreiben Sie eine Matlab-Funktion `isogeoGrid(knotsGeoX, orderGeoX, knotsGeoY, orderGeoY, ctrlPts, J)`, welche für einen Level $J \in \mathbb{N}$ das uniforme Gitter $(i/2^J, k/2^J)$ für $i, k = 0, \ldots, 2^J$ im parametrischen Gebiet $\hat{\Omega}$ auf das physikalische Gebiet Ω abbildet und plottet.

   Gehen Sie dazu folgendermaßen vor: Das Gitter besteht aus $2^J + 1$ waagerechten und senkrechten Linien (z. B. von $(3/2^J, 0)$ bis $(3/2^J, 1)$)). Diskretisieren Sie jede dieser Linien mit $2^{J+4} + 1$ Punkten und bilden Sie jeden dieser Punkte mit $\mathbf{F}$ in das physikalische Gebiet Ω ab. Zeichnen Sie die lineare Interpolation dieser abgebildeten Punkte für jede Linie in einen gemeinsamen Plot.

f) Schreiben Sie eine Matlab-Funktion `A = isogeoMat(knotsGeoX, orderGeoX, knotsGeoY, orderGeoY, ctrlPts, orderPar, J)`, welche die Matrix $\mathbf{A}$ aus (6) aufstellt. Dabei bezeichnet $J \in \mathbb{N}$ den Level des uniformen parametrischen Gitters. Wählen Sie für beide B-Spline Basisfunktionen N_{i_1}, N_{i_2} im Tensorproduktansatz von $\hat{V}_h$ die Ordnung $k =$ `orderPar` und den erweiterten äquidistanten Knotenvektor

$$\theta_1 = \ldots = \theta_k = 0 < 2^{-J} < 2 \cdot 2^{-J} < 3 \cdot 2^{-J} < \ldots < 1 - 2^{-J} < 1 = \theta_{n+1} = \ldots = \theta_{n+k}. \tag{8}$$

   Beachten Sie, dass die jeweils erste und letzte Basisfunktion beider Splines ausgelassen wird, um die Dirichlet-Randbedingung zu erfüllen. Das heißt, $\mathcal{I}_1 = \mathcal{I}_2 = \{2, \ldots, 2^J + k - 2\}$.

   Teilen Sie die Integrale in die relevanten Teilstücke $[\theta_i, \theta_{i+1}] \times [\theta_j, \theta_{j+1}]$ auf und verwenden Sie 2D-Gauß-Quadratur. Benutzen Sie die vorherigen Aufgabenteile zur Berechnung von $\mathbf{F}$ und $\mathrm{d}\mathbf{F}/\mathrm{d}\mathbf{x}$.

g) Schreiben Sie eine Matlab-Funktion `b = isogeoVec(knotsGeoX, orderGeoX, knotsGeoY, orderGeoY, ctrlPts, orderPar, J, f)`, welche den Vektor **b** aus (7) aufstellt. Die Funktion f aus (1) ist dabei durch das Function-Handle `f` gegeben. Die übrigen Parameter entsprechen denen aus Teil f).

h) Betrachten Sie (1) für die rechte Seite $f(\mathbf{x}) = 5\pi^2 \sin(2\pi x_1)\sin(\pi x_2)$ mit der analytischen Lösung $u(\mathbf{x}) = \sin(2\pi x_1)\sin(\pi x_2)$ auf dem L-förmigen Gebiet $\Omega = (-1,1)^2 \setminus (0,1)^2$. Für dieses physikalische Gebiet sollen zwei Parametrisierungen $\mathbf{F}_1$ und $\mathbf{F}_2$ betrachtet werden:

   1) Quadratische Splines ($k = 3$) mit Knotenvektoren `knotsX = [0 0 0 0.5 0.5 1 1 1]` und `knotsY = [0 0 0 1 1 1]` sowie den Kontrollpunkten `ctrlPts = [-1, 1; -0.6, 1; 0, 1; -1, 0; -0.55, 0; 0, 0.5; -1, -1; -0.5, -0.5; 0, 0; 0, -1; 0, -0.55; 0.5, 0; 1, -1; 1, -0.6; 1, 0]`. Hier ist der Spline in x_1-Richtung nur stetig bei $\hat{x}_1 = 0.5$.

   2) Quadratische Splines ($k = 3$) mit Knotenvektoren `knotsX = [0 0 0 0.5 1 1 1]` und `knotsY = [0 0 0 1 1 1]` sowie den Kontrollpunkten `ctrlPts = [-1, 1; -0.65, 1; 0, 1; -1, -1; -0.7, 0; 0, 0; -1, -1; 0, -0.7; 0, 0; 1, -1; 1, -0.65; 1, 0]`. Hier sind beide Splines stetig differenzierbar.

   Plotten Sie für beide Parametrisierungen von Ω das uniforme Gitter zum Level $J = 3$ mit der Funktion `isogeoGrid` aus Teil e).

   Führen Sie für jede der beiden Parametrisierungen die folgenden Schritte durch:

   - Lösen Sie unter Verwendung der oben erstellten Funktionen das Problem für $k =$ `orderPar` $= 2, \ldots, 4$ 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:

   $$\|u_{J,k} - u\|^2_{L_2(\Omega)} = \int_{\hat{\Omega}} \left(u(\mathbf{F}(\mathbf{x})) - \sum_{\mathbf{i}} (\mathbf{u}_{J,k})_{\mathbf{i}} N_{\mathbf{i}}(\mathbf{x}) \right)^2 \left| \det \left(\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right) \right| \mathrm{d}\mathbf{x}.$$

   - 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

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

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

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

   - Plotten Sie für jede Ordnung k die jeweils letzte Lösung $u_{5,k}$ auf dem physikalischen Gebiet Ω. Wählen Sie hierfür ein geeignetes Plot-Gitter.

Literatur

[HCB05] T.J.R. Hughes, J.A. Cottrell und Y. Bazilevs. „Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement“. *Computer Methods in Applied Mechanics and Engineering* 194.39 (2005), S. 4135–4195. DOI: `https://doi.org/10.1016/j.cma.2004.10.008`.