Prof. Dr. Angela Kunoth
Samuel Leweke

Abschlussprojekt: Tensorprodukt Wavelets

Auf dem Intervall $(0,1)$ kann eine Waveletbasis aus randangepassten B-Spline Wavelets konstruiert werden. Die primale Generatorbasis besteht dann auf B-Splines, deren Verfeinerungseigenschaft genutzt wird. Beispielsweise besteht die primale Generatorbasis für $d = 2$ auf Level j aus $2^j + 1$ skalierten und translierten Hutfunktionen $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$, $k = 0, \dots, 2^j$, mit

$$\phi(x) \coloneqq \begin{cases} 1 + x & \text{für } x \in [-1, 0), \\ 1 - x & \text{für } x \in [0, 1), \\ 0 & \text{sonst.} \end{cases} \tag{1}$$

Am Rand des Intervalls sind jedoch Anpassungen notwendig, da hier eine andere Verfeinerungsgleichung gilt und Biorthogonalität sowie die Zahl der verschwindenden Momente erhalten werden soll. Schließlich hängen die Änderungen auch von der Randbedingung des unterliegenden Raumes ab.

Diese Anpassungen zeigen sich in den Strukturen der Transformationsmatrizen. Betrachten Sie zum Beispiel für B-Spline Wavelets mit $d = 2$, $\tilde{d} = 4$ und minimalem Level $j_0 = 3$ die Transformationsmatrix $\mathbf{M}_j = (\mathbf{M}_{j,0}, \mathbf{M}_{j,1})\colon \ell_2(\Delta_j \cup \nabla_j) \to \ell_2(\Delta_{j+1})$, wobei $\#\Delta_j = 2^j + 1$ und $\#\nabla_j = 2^j$. Bei der Anpassung an verschiedene Level bleiben die erste und letzte Spalte von $\mathbf{M}_{j,0}$ sowie die ersten und letzten beiden Spalten von $\mathbf{M}_{j,1}$ gleich. Nur die inneren Spalten werden entsprechend oft wiederholt.

Sei $\Phi_J = \{\phi_{J,k} : k \in \Delta_J\}$ eine Einzelskalenbasis und

$$\Psi^J = \Phi_{j_0} \cup \bigcup_{j=j_0}^{J-1} \Psi_j, \qquad \Psi_j \coloneqq \{\psi_{j,k} : k \in \nabla_j\}, \tag{2}$$

eine Waveletbasis auf $(0,1)$ zum Level J. Die Wavelet-Transformation $\mathbf{T}_J$ ist eine lineare Abbildung

$$(\Phi_{j_0}, \Psi_{j_0}, \Psi_{j_0+1}, \dots, \Psi_{J-1})^T = \mathbf{T}_J^T \Phi_J,$$

die sich aus der iterativen Anwendung der Matrizen $\mathbf{M}_j$ ergibt:

$$\mathbf{T}_J \coloneqq \mathbf{T}_{J,J-1} \cdots \mathbf{T}_{J,j_0}\colon \ell_2(\Delta_J) \to \ell_2(\Delta_J) \quad \text{mit} \quad \mathbf{T}_{J,j} \coloneqq \begin{pmatrix} \mathbf{M}_j & 0 \\ 0 & \mathbf{I}^{(\#\Delta_J)-(\#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J)\times(\#\Delta_J)}. \tag{3}$$

Aus einer Basis auf dem Intervall $(0,1)$ kann mittels Tensorieren eine Basis auf $(0,1)^n$, $d \geq 2$, gewonnen werden. Die Einzelskalenbasisfunktionen auf $(0,1)^n$ sind gegeben durch

$$\phi_{j,\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \phi_{j,k_i}(x_i), \qquad \mathbf{x} = (x_1, \dots, x_d), \tag{4}$$

wobei $\mathbf{k} = (k_1, \dots, k_d) \in (\Delta_j)^n$ nun ein Multi-Index ist. Damit ergibt sich $\Phi_{J,n} = \Phi_J \otimes \dots \otimes \Phi_J$. Für die Waveletbasis gibt es zwei verschiedene Ansätze:

- In der *anisotropen* Konstruktion werden die gesamten eindimensionalen Waveletbasen (2) einfach tensoriert:

$$\Psi^J_{n,\text{ani}} = \bigotimes_{i=1}^{n} \Psi^J. \tag{5}$$

  Eine Basisfunktion hat die Form

$$\psi^{\text{ani}}_{\mathbf{j},\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \psi_{j_i,k_i}(x_i), \tag{6}$$

wobei der Waveletindex $\lambda = (\mathbf{j}, \mathbf{k})$ nun aus zwei Multi-Indizes besteht.

Eine Basisfunktion kann also ein Tensorprodukt der eindimensionalen Waveletbasisfunktionen auf verschiedenen Leveln sein. Dies hat einen quader-artigen Träger mit verschiedenen Kantenlängen, die auch sehr unterschiedlich lang sein können, zur Folge, was den Namen *anisotrop* rechtfertigt.

Die Wavelettransformation $\mathbf{T}_J^{\mathrm{ani},n}$ ergibt sich hier als Tensorprodukt der eindimensionalen Transformationen:

$$\mathbf{T}_J^{\mathrm{ani},n} \coloneqq \mathbf{T}_{J,J-1}^{\mathrm{ani},n} \cdots \mathbf{T}_{J,j_0}^{\mathrm{ani},n} \quad \text{mit} \quad \mathbf{T}_{J,j}^{\mathrm{ani},n} \coloneqq \bigotimes_{i=1}^{n} \mathbf{T}_{J,j}. \tag{7}$$

- In der *isotropen* Konstruktion werden nur die Basisfunktionen des gleichen Levels tensoriert. Sei $e \in \{0,1\}$ und das eindimensionale isotrope Wavelet gegeben durch

$$\psi_{j,k,e}^{\mathrm{iso}}(x) \coloneqq \begin{cases} \phi_{j,k}(x) & \text{für } e = 0, k \in \nabla_{j,0}, \\ \psi_{j,k}(x) & \text{für } e = 1, k \in \nabla_{j,1}, \end{cases} \qquad \nabla_{j,e} \coloneqq \begin{cases} \Delta_j & \text{für } e = 0, \\ \nabla_j & \text{für } e = 1. \end{cases} \tag{8}$$

Das isotrope Wavelet auf $(0,1)^n$ ist das Tensorprodukt der eindimensionalen isotropen Wavelets:

$$\psi_{j,\mathbf{k},\mathbf{e}}^{\mathrm{iso}}(\mathbf{x}) = \prod_{i=1}^{n} \psi_{j,k_i,e_i}^{\mathrm{iso}}(x_i), \qquad \mathbf{e} = (e_1, \ldots, e_d) \in \{0,1\}^n, \mathbf{k} = (k_1, \ldots, k_d) \in \bigtimes_{i=1}^{n} \nabla_{j,e_i} =: \nabla_{j,\mathbf{e}}. \tag{9}$$

Enthält der Multi-Index e, der auch als Zahl in Binärdarstellung interpretiert werden kann, an einer Position eine 0, ist eine Einzelskalenfunktion im Tensorprodukt enthalten. Eine besondere Rolle spielt dabei der Index $\mathbf{e} = \mathbf{0}$, der zu den Tensorprodukten aus reinen Einzelskalenfunktionen gehört.

Die isotrope Waveletbasis ist durch das Zusammenfassen aller isotropen Wavelets des gleichen Levels definiert:

$$\Psi_{n,\mathrm{iso}}^{J} = \left\{ \psi_{j_0,\mathbf{k},\mathbf{0}}^{\mathrm{iso}} : \mathbf{k} \in \nabla_{j,\mathbf{0}} \right\} \cup \bigcup_{j=j_0}^{J-1} \bigcup_{\mathbf{0} \neq \mathbf{e} \in \{0,1\}^n} \left\{ \psi_{j,\mathbf{k},\mathbf{e}}^{\mathrm{iso}} : \mathbf{k} \in \nabla_{j,\mathbf{e}} \right\}. \tag{10}$$

Der Waveletindex ist hier $\lambda = (j, \mathbf{k}, \mathbf{e})$.

Für die Wavelettransformation wird zunächst die Transformationsmatrix $\mathbf{M}_{j,\mathbf{e}}$ benötigt:

$$\mathbf{M}_{j,\mathbf{e}} \coloneqq \bigotimes_{i=1}^{n} \mathbf{M}_{j,e_i}, \tag{11}$$

wobei $\mathbf{M}_{j,0}$, $\mathbf{M}_{j,1}$ die gewöhnlichen 1D Transformationsmatrizen sind. Aus diesen Matrizen wird die isotrope Verfeinerungsmatrix $\mathbf{M}_j^{\mathrm{iso}}$ aufgebaut:

$$\mathbf{M}_j^{\mathrm{iso}} \coloneqq \left(\mathbf{M}_{j,(0,\ldots,0,0)}, \mathbf{M}_{j,(0,\ldots,0,1)}, \mathbf{M}_{j,(0,\ldots,1,0)}, \mathbf{M}_{j,(0,\ldots,1,1)}, \ldots, \mathbf{M}_{j,(1,\ldots,1,1)} \right) \in \mathbb{R}^{(\#\Delta_{j+1})^n \times (\#\Delta_{j+1})^n}. \tag{12}$$

Die Wavelettransformation $\mathbf{T}_J^{\mathrm{ani},n}$ ergibt sich analog zu (3):

$$\mathbf{T}_J^{\mathrm{iso},n} \coloneqq \mathbf{T}_{J,J-1}^{\mathrm{iso},n} \cdots \mathbf{T}_{J,j_0}^{\mathrm{iso},n} \quad \text{mit} \quad \mathbf{T}_{J,j}^{\mathrm{iso},n} \coloneqq \begin{pmatrix} \mathbf{M}_j^{\mathrm{iso}} & 0 \\ 0 & \mathbf{I}^{(\#\Delta_J)^n - (\#\Delta_{j+1})^n} \end{pmatrix}. \tag{13}$$

In diesem Projekt sollen Tensorprodukt B-Spline Wavelets auf $\Omega =: (0,1)^2$, also $n = 2$, zur Diskretisierung der PDE

$$\begin{aligned} -\Delta u + u &= f \quad \text{in } \Omega \\ \frac{\partial u}{\partial n} &= 0 \quad \text{auf } \Gamma \coloneqq \partial\Omega \end{aligned} \tag{14}$$

verwendet werden. Dabei soll die die Transformation in Waveletkoordinaten dazu genutzt werden, um die Kondition des diskreten Gleichungssystems der PDE (14) zu verbessern. Die Äquivalenz der $\|\cdot\|_{H^1(\Omega)}$-Norm und der $\|\cdot\|_{\ell_2}$-Norm der Waveletkoordinaten führt zu einer (asymptotisch) optimalen Kondition, das heißt, die Kondition hängt nicht mehr vom (maximalen) Level J oder der Gitterweite h ab.

Zunächst gilt nur eine Normäquivalenz zwischen der $\|\cdot\|_{L_2(\Omega)}$-Norm und der $\|\cdot\|_{\ell_2}$-Norm der Waveletkoordinaten. Um aus dieser $L_2(\Omega)$-Waveletbasis eine $H^s(\Omega)$-Waveletbasis ($s \in \mathbb{N}$) mit entsprechender Äquivalenz der $H^s(\Omega)$-Norm zu erhalten, muss eine Diagonalskalierung vorgenommen werden. Für anisotrope Wavelets ist $\mathbf{\Psi}^{J,s}_{n,\text{ani}} = \mathbf{D}^{\text{ani}}_s \mathbf{\Psi}^{J}_{n,\text{ani}}$ eine Riesz-Basis für $H^s(\Omega)$ mit der Matrix

$$\left(\mathbf{D}^{\text{ani}}_s\right)_{\lambda,\lambda'} = 2^{-s\|\|\lambda\|\|_\infty}\delta_{\lambda,\lambda'}, \qquad |\lambda| = |(\mathbf{j},\mathbf{k})| \coloneqq \mathbf{j}, \quad \|\mathbf{j}\|_\infty \coloneqq \max_{i=1,\dots,n} |j_i|. \tag{15}$$

Für isotrope Wavelets ist $\mathbf{\Psi}^{J,s}_{n,\text{iso}} = \mathbf{D}^{\text{iso}}_s \mathbf{\Psi}^{J}_{n,\text{iso}}$ eine Riesz-Basis für $H^s(\Omega)$ mit der Matrix

$$\left(\mathbf{D}^{\text{iso}}_s\right)_{\lambda,\lambda'} = 2^{-s|\lambda|}\delta_{\lambda,\lambda'}, \qquad |\lambda| = |(j,\mathbf{k},\mathbf{e})| \coloneqq j. \tag{16}$$

Aufgaben

a) Schreiben Sie eine Matlab-Funktion `c = FWTani(d, maxLevel)`, welche die schnelle anisotrope Wavelettransformation $\mathbf{T}^{\text{ani},2}_{\texttt{maxLevel}}$ aus (7) mit den randangepassten Wavelets durchführt. Die Detailkoeffizienten `d` auf Level `maxLevel` sollen dabei in Einzelskalenkoeffizienten `c` transformiert werden. Verwenden Sie zur Implementierung das Pyramid-Schema aus der Vorlesung.

Hinweis: Den Aufbau und die Koeffizienten der Transformationsmatrix $\mathbf{M}_j$ entnehmen Sie der Matrix $\mathbf{M}_4$, die Sie auf der Homepage der Vorlesung herunterladen können (`M24_4.mat`).

b) Schreiben Sie eine Matlab-Funktion `c = FWTaniTrans(d, maxLevel)`, welche die transponierte Version $(\mathbf{T}^{\text{ani},2}_{\texttt{maxLevel}})^T$ von `FWTani` durchführt. Verwenden Sie ebenfalls das Pyramid-Schema.

c) Schreiben Sie eine Matlab-Funktion `c = FWTiso(d, maxLevel)`, welche die schnelle isotrope Wavelettransformation $\mathbf{T}^{\text{iso},2}_{\texttt{maxLevel}}$ aus (13) mit den randangepassten Wavelets durchführt. Die Detailkoeffizienten `d` auf Level `maxLevel` sollen dabei in Einzelskalenkoeffizienten `c` transformiert werden. Verwenden Sie zur Implementierung das Pyramid-Schema aus der Vorlesung.

Hinweis: Den Aufbau und die Koeffizienten der Transformationsmatrix $\mathbf{M}_j$ entnehmen Sie der Matrix $\mathbf{M}_4$, die Sie auf der Homepage der Vorlesung herunterladen können (`M24_4.mat`).

d) Schreiben Sie eine Matlab-Funktion `c = FWTisoTrans(d, maxLevel)`, welche die transponierte Version $(\mathbf{T}^{\text{iso},2}_{\texttt{maxLevel}})^T$ von `FWTiso` durchführt. Verwenden Sie ebenfalls das Pyramid-Schema.

e) Schreiben Sie eine Matlab-Funktion `ds = DiagScale(d, s, maxLevel, isAni)`, die die Diagonalskalierungsmatrix $\mathbf{D}^{\text{ani}}_{\texttt{s}}$ (für `isAni == true`) bzw. $\mathbf{D}^{\text{iso}}_{\texttt{s}}$ (für `isAni == false`) auf den Vektor der Detailkoeffizienten `d` mit maximalem Level `maxLevel` anwendet. Die Matrizen sind in (15) und (16) definiert und das Ergebis der Operation wird als Vektor `ds` zurückgegeben.

f) Schreiben Sie eine Matlab-Funktion `[S, M] = StiffMass1D(J)`, welche die (1D) Steifigkeitsmatrix `S` und die Massenmatrix `M` zum Level `J` berechnet, wobei als Basisfunktionen die Einzelskalenbasis $\phi_{\texttt{J},k}$ verwendet wird. Die Matrizen sind gegeben durch

$$\texttt{S}_{i,k} = \int_\Omega \nabla\phi_{\texttt{J},k} \cdot \nabla\phi_{\texttt{J},i}\,\mathrm{d}x, \qquad \texttt{M}_{i,k} = \int_\Omega \phi_{\texttt{J},k}\phi_{\texttt{J},i}\,\mathrm{d}x, \qquad i,k = 1,\dots,2^{\texttt{J}}+1. \tag{17}$$

g) Schreiben Sie eine Matlab-Funktion `[u, k] = WCG(A, b, J)`, welche das vorkonditionierte CG-Verfahren in Waveletkoordinaten anwendet, um das diskrete System $\texttt{Au} = \texttt{b}$ der PDE (14) auf Level `J` zu lösen.

Das vorkonditionierte System lautet

$$\mathbf{D}\mathbf{T}^T_{\texttt{J}}\texttt{A}\mathbf{T}_{\texttt{J}}\mathbf{D}\mathbf{v} = \mathbf{D}\mathbf{T}^T_{\texttt{J}}\texttt{b}, \tag{18}$$

wobei die Lösung $\texttt{u} = \mathbf{T}_{\texttt{J}}\mathbf{D}\mathbf{v}$ ist und $\mathbf{D}$ die Diagonalskalierungsmatrix aus Teil e) mit $\texttt{s} = 1$ meint.

Außer der Lösung `u` soll auch die Anzahl an benötigten Iterationen `k` zurückgegeben werden, die benötigt wurden, um Diskretisierungsfehlergenauigkeit $2^{-\texttt{J}}$ zu erreichen. Verwenden Sie die Matlab-Funktion `pcg()`.

h) Betrachten Sie das Problem (14) mit

$$f(\mathbf{x}) = 4\pi^2 \cos(2\pi x_1)\cos(3\pi x_2), \qquad u(x) = \cos(2\pi x_1)\cos(3\pi x_2). \tag{19}$$

Schreiben Sie ein Programm, welches das (normale) CG-Verfahren und das vorkonditionierte CG-Verfahren (PCG) miteinander vergleicht. Lösen Sie dazu das Problem für die Level $j = 4, 5, \ldots, 8$ mit jedem der beiden Verfahren und jeder der beiden Wavelet-Varianten (anisotrop / isotrop). Zeichnen Sie die Anzahl der Iterationen `k` auf, die zum Erreichen der Diskretisierungsfehlergenauigkeit 2^{-j} nötig sind. Tragen Sie dann diese Iterationszahlen gegen den Level j in einen gemeinsamen Plot ab, wobei die y-Achse (Iterationen) logarithmisch skaliert wird.

Berechnen Sie für beide Wavelet-Varianten die Kondition der Matrix

$$\mathbf{D}\mathbf{T}_j^T\mathbf{A}\mathbf{T}_j\mathbf{D}$$

aus Teil g) für die Level $j = 4, 5, \ldots, 8$. Tragen Sie die Konditionen zusammen mit der Kondition von $\mathbf{A}$ in einem gemeinsamen Plot gegen das Level j ab, wobei die y-Achse (Kondition) logarithmisch skaliert wird.

Verwenden Sie Tensorprodukte der in Teil f) erstellten 1D Matrizen um die Systemmatrix $\mathbf{A}$ des Problems aufzubauen. Ebenfalls können Sie die Tensorprodukt-Struktur der rechten Seite f ausnutzen, um den Vektor $\mathbf{b}$ zu erzeugen.

i) Erklären Sie die in Teil h) erzielten Ergebnisse.

Literatur

[DKU99] Wolfgang Dahmen, Angela Kunoth und Karsten Urban. „Biorthogonal Spline Wavelets on the Interval—Stability and Moment Conditions“. *Applied and Computational Harmonic Analysis* 6.2 (1999), S. 132–196. DOI: https://doi.org/10.1006/acha.1998.0247.

[Pab15] Roland Pabel. „Adaptive Wavelet Methods for Variational Formulations of Nonlinear Elliptic PDEs on Tensor-Product Domains“. Diss. Universität zu Köln, Mai 2015.

[Urb08] Karsten Urban. *Wavelet Methods for Elliptic Partial Differential Equations*. Oxford University Press, Nov. 2008. DOI: 10.1093/acprof:oso/9780198526056.001.0001.