Prof. Dr. Angela Kunoth
Samuel Leweke

Abschlussprojekt: Wavelet Nested Iteration CG

Die Äquivalenz der $\|\cdot\|_{H^1}$-Norm und der $\|\cdot\|_{\ell_2}$-Norm der Waveletkoordinaten bedeutet, dass das diskrete Gleichungssystem einer elliptischen PDE (asymptotisch) in Waveletkoordinaten optimal konditioniert ist. Die Kondition hängt also nicht mehr vom (maximalen) Level J oder der Gitterweite h ab.

Eine Iteration des CG-Verfahrens senkt den Fehler in der diskreten Energie-Norm $\|\mathbf{v}\|_{\mathbf{A}_J} \coloneqq \sqrt{\mathbf{v}^T \mathbf{A}_J \mathbf{v}}$ daher um den konstanten Faktor $q \in (0,1)$:

$$\|\mathbf{u}_J - \mathbf{u}_J^{(k)}\|_{\mathbf{A}_J} \leq 2q^k \|\mathbf{u}_J - \mathbf{u}_J^{(0)}\|_{\mathbf{A}_J}, \qquad q \coloneqq \frac{\sqrt{\kappa_2(\mathbf{A}_J)} - 1}{\sqrt{\kappa_2(\mathbf{A}_J)} + 1}, \qquad k \geq 0,$$

wobei $\mathbf{u}_J$ die exakte diskrete Lösung des Systems $\mathbf{A}_J \mathbf{u}_J = \mathbf{b}$ auf Level $J \in \mathbb{N}$ ist.

Aus Übung 2 ist bekannt, dass es genügt, das diskrete System bis auf Diskretisierungsfehlergenauigkeit 2^{-J} zu lösen. Wird das CG-Verfahren verwendet, werden $O(J\sqrt{\kappa_2(\mathbf{A}_J)})$ Iterationen auf Level J benötigt, um einen (relativen) Energie-Fehler von 2^{-J} zu erreichen. Selbst bei optimaler Vorkonditionierung ($\kappa_2(\mathbf{A}_J) \sim 1$ für $J \to \infty$) bleibt der logarithmische Faktor $J \sim -\log(N)$ bestehen, wobei N die Anzahl der Freiheitsgrade bezeichnet.

Dieser Faktor kann durch *nested iteration* entfernt werden, was einen *optimalen* Aufwand von $O(N)$ Operationen bedeutet. Im Nested Iteration Conjugate Gradient (NICG) Algorithmus wird das diskrete System auf dem gröbsten Level j_0 zunächst exakt (bzw. bis auf Maschinengenauigkeit) gelöst. Die Lösung des nächsten Levels $j+1$ wird mit dem CG-Verfahren berechnet, welches mit der Lösung des vorherigen Levels j gestartet wird. Hier werden nun so viele Iterationen ausgeführt, bis der Fehler den aktuellen Diskretisierungsfehler $2^{-(j+1)}$ unterschreitet. Dies wird iterativ fortgesetzt, bis das feinste Level J erreicht wird.

Nested Iteration Conjugate Gradient Algorithmus

1. Löse auf kleinstem Level j_0 exakt (d. h. bis auf Maschinengenauigkeit): $\mathbf{A}_{j_0}\mathbf{u}_{j_0} = \mathbf{b}_{j_0}$.
2. Setze $j \coloneqq j_0 + 1$ und $\mathbf{v}_{j_0} \coloneqq \mathbf{u}_{j_0}$.
3. Prolongation $(\mathbf{v}_j^{(0)})$ von $\mathbf{v}_{j-1}$ auf Level j durch Anfügen von 0en (in Waveletkoordinaten): $(\mathbf{v}_j^{(0)})^T \coloneqq (\mathbf{v}_{j-1}^T, \mathbf{0}^T)$.
4. Starte das CG-Verfahren zur Lösung von $\mathbf{A}_j\mathbf{u}_j = \mathbf{b}_j$ mit $\mathbf{v}_j^{(0)}$ und iteriere bis ein Fehler von 2^{-j} erreicht wurde um $\mathbf{v}_j$ zu erhalten.
5. Ist das feinste Level $j = J$ erreicht, breche ab. Ansonsten setze $j \coloneqq j+1$ und gehe zu Schritt 3.

In diesem Projekt sollen das CG-Verfahren mit und ohne (optimale) Vorkonditionierung sowie das NICG-Verfahren verglichen werden. Dazu soll die eindimensionale PDE

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

verwendet werden.

Als Waveletbasis auf dem Intervall Ω werden randangepasste B-Spline Wavelets mit $d = 2$, $\tilde{d} = 4$ und minimalem Level $j_0 = 3$ verwendet. Das heißt, die primale Generatorbasis auf Level j besteht aus $2^j + 1$ skalierten und

translierten Hutfunktionen $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$, $k = 0, \ldots, 2^j$, mit

$$\phi(x) \coloneqq \begin{cases} 1 + x & \text{für } x \in [-1, 0), \\ 1 - x & \text{für } x \in [0, 1), \\ 0 & \text{sonst.} \end{cases} \tag{2}$$

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 $\#\nabla_j = 2^j$, muss an den Rändern des Intervalls angepasst werden. 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.

Aufgaben

a) Schreiben Sie eine Matlab-Funktion `c = FWT(d, maxLevel)`, welche die schnelle Wavelettransformation $\mathbf{T}_{\texttt{maxLevel}}$ 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 = FWTtrans(d, maxLevel)`, welche die transponierte Version $\mathbf{T}^T_{\texttt{maxLevel}}$ von `FWT` durchführt. Verwenden Sie ebenfalls das Pyramid-Schema.

c) Schreiben Sie eine Matlab-Funktion `ds = DiagScale(d, s, maxLevel)`, die eine Diagonalskalierung $\mathbf{D}$ der Detailkoeffizienten `d` mit Level `maxLevel` vornimmt:

$$\texttt{d} = \begin{pmatrix} \mathbf{c}_{j_0} \\ \mathbf{d}_{j_0} \\ \vdots \\ \mathbf{d}_{\texttt{maxLevel}-1} \end{pmatrix} \mapsto \begin{pmatrix} 2^{-j_0\texttt{s}}\mathbf{c}_{j_0} \\ 2^{-j_0\texttt{s}}\mathbf{d}_{j_0} \\ \vdots \\ 2^{-(\texttt{maxLevel}-1)\texttt{s}}\mathbf{d}_{\texttt{maxLevel}-1} \end{pmatrix} =: \mathbf{D}\texttt{d}. \tag{3}$$

d) Schreiben Sie eine Matlab-Funktion `[S, M] = StiffMass(J)`, welche die 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

$$\texttt{S}_{i,k} = \int_\Omega \nabla\phi_{\texttt{J},k} \cdot \nabla\phi_{\texttt{J},i}\,\mathrm{d}x, \qquad \texttt{M}_{i,k} = \int_\Omega \phi_{\texttt{J},k}\phi_{\texttt{J},i}\,\mathrm{d}x, \qquad i,k = 1, \ldots, 2^{\texttt{J}} + 1. \tag{4}$$

e) Schreiben Sie eine Matlab-Funktion `[u, k] = WCG(b, J)`, welche das vorkonditionierte CG-Verfahren in Waveletkoordinaten anwendet, um das diskrete System $\mathbf{A}\texttt{u} = \texttt{b}$ der PDE (1) auf Level `J` zu lösen. Die Matrix $\mathbf{A}$ ist dabei gegeben durch $\mathbf{A} = \texttt{S} + \texttt{M}$ mit `S`, `M` aus Teil d).

Das vorkonditionierte System lautet

$$\mathbf{D}T_{\texttt{J}}^T\mathbf{A}T_{\texttt{J}}\mathbf{D}\mathbf{v} = \mathbf{D}T_{\texttt{J}}^T\texttt{b}, \tag{5}$$

wobei die Lösung $\texttt{u} = T_{\texttt{J}}\mathbf{D}\mathbf{v}$ ist und $\mathbf{D}$ die Diagonalskalierungsmatrix aus Teil c) 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()`.

f) Schreiben Sie eine Matlab-Funktion `[u, k] = NICG(b, J)`, welche den oben beschriebenen NICG-Algorithmus anwendet, um das System aus Teil e) zu lösen. Geben Sie neben der Lösung (in Einzelskalenbasis) auch die Anzahl `k` der CG-Iterationen auf Level `J` zurück. Verwenden Sie als gröbstes Level $j_0 = 3$.

g) Betrachten Sie das Problem (1) mit

$$f(x) = \sin(2\pi x), \qquad u(x) = \frac{2\pi\left(e^{1-x} - e^{x}\right) + (1+e)\sin(2\pi x)}{(1+e)(1+4\pi^2)}. \tag{6}$$

Schreiben Sie ein Programm, welches das (normale) CG-Verfahren, das vorkonditionierte CG-Verfahren (PCG) und das NICG-Verfahren miteinander vergleicht. Lösen Sie dazu das Problem für die Level $j = 5, 6, \ldots, 12$ mit jedem der drei Verfahren. 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.

h) Erklären Sie die in Teil g) 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.

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