Matrixkompression und H-Matrizen

Skriptum zur Vorlesung

von

Markus Faustmann

Wien, am 7. März 2017

Dieses Skript basiert zu großen Teilen auf den Vorlesungsaufzeichnungen [Pra09], dem Buch [Hac09], welches dem interessierten Leser etliche weiterführende Informationen sowie einen guten Einstieg in die Materie liefert, sowie den Vorlesungsaufzeichnungen [Bör16, BGH06]. Die Bücher [Bör10, Beb08] waren hilfreiche Literatur beim Zusammenstellen des Skripts für die Kapitel $\mathcal{H}^2$-Matrizen und ACA und beinhalten etliche Erweiterungen der vorgestellten Materie.

Sämtliche Rechnungen wurden mit Hilfe des C-Softwarepakets HLIB [BG99] getätigt, welches in den Lecture Notes [BGH06] näher beschrieben wird. HLIB ist für nichtkommerziellen Gebrauch gratis verfügbar: http://www.hlib.org/.

Ein ganz großer Dank gebührt Prof. Dirk Praetorius für das Zurverfügungstellen seiner Unterlagen zur Vorlesung "Hierarchische Matrizen und Fast-Multipole-Methode".

Historisch gesehen ist die erste Erwähnung von hierarchischen Matrizen, welche eine algebraische Verallgemeinerung vorheriger Kompressionstechniken (z.B. Fast Multipole Method [Rok85, GR97]) darstellen, in der Arbeit von HACKBUSCH [Hac99] zu finden. Der große Vorteil von $\mathcal{H}$-Matrizen ist, dass sie im Gegensatz zu vorherigen Techniken auch schnelle arithmetische Operationen erlauben. Eine wichtige Arbeit hierbei war die Dissertation von GRASEDYCK [Gra01]. Für das effiziente und praktikable Assemblieren von $\mathcal{H}$-Matrizen seien die Arbeiten von BEBENDORF [Beb00] (ACA) sowie BÖRM, GRASEDYCK [BG05, BG04] (HCA,Interpolation) erwähnt.

Ein weitere Reduktion der Komplexität liefert das Format der $\mathcal{H}^2$-Matrizen (siehe z.B. Arbeiten von GIEBERMANN, HACKBUSCH & BÖRM [Gie01, HB02]).

Zu hierarchische Matrizen gibt es noch etliche weitere Arbeiten in den letzten beiden Jahrzehnten von obig genannten Autoren und vielen Weiteren. Tatsächlich sind $\mathcal{H}$-Matrizen und verwandte Themen noch immer ein aktives Forschungsgebiet mit etlichen interessanten, kürzlich erschienenen neuen Beiträgen, siehe beispielsweise [BG13, BKV15, Hac16, FMP16, FKS17].

Für Berechnungen seien neben dem obigem Softwarepaket HLIB noch die Pakete H2LIB (http://www.h2lib.org/), AHMED (https://github.com/xantares/ahmed) sowie HLIBPRO (http://www.hlibpro.com/) erwähnt.

Inhaltsverzeichnis

1 Einleitung

Das Ziel dieser Vorlesung ist, effiziente Verfahren für (gewisse) vollbesetzte Matrizen vorzustellen, sodass elementare Operationen mit Matrizen mit einem linearen Aufwand von (annähernd) $\mathcal{O}(N)$ durchgeführt werden können.

1.1 Lineare Komplexität

Sei $\phi : X_n \to Y_m$ ein Funktion, die ausgewertet werden soll, wobei $|X_n| = n$ Eingangsdaten und $|Y_m| = m$ Ausgangsdaten sind. Falls die alle Input und Output Daten mindestens einmal benötigt werden (also bspw. ϕ nicht trivial ist), dann ist der Aufwand der Auswertung mindestens $\mathcal{O}(N)$, wobei $N := \max(n, m)$. Wir sprechen hierbei von *linearer Komplexität*.

Im Fall von Operationen mit Matrizen $\mathbf{A} \in \mathbb{R}^{N \times N}$, kann $\mathcal{O}(N^2)$ als lineare Komplexität angesehen werden, was allerdings in praktischen Rechnungen dennoch einen hohen Aufwand darstellt. Hat man allerdings mehr Struktur, beispielsweise Diagonalmatrizen, schwachbesetzte Matrizen oder zirkuläre Matrizen (hier hat man jeweils nur $\mathcal{O}(N)$ verschiedene Eingangsdaten), so kann man Speicherung und Matrix-Vektor-Multiplikation in linear Komplexität $\mathcal{O}(N)$ bzw. *logarithmisch linearer Komplexität* $\mathcal{O}(N \log N)$ (FFT) durchführen.

Bei Diagonalmatrizen ist auch Addition, Multiplikation und Inversion in Komplexität $\mathcal{O}(N)$ durchführbar, bei schwachbesetzten Matrizen allerdings nicht, da beispielsweise $A + B$ deutlich weniger schwachbesetzt sein kann und Inversion im Allgemeinen sogar eine vollbesetzte Matrix liefert.

Generell kann man außer für Diagonalmatrizen nicht arithmetische Operationen mit Komplexität $\mathcal{O}(N)$ erwarten.

Wir werden in weiterer Folge versuchen bestimmte Klassen von Matrizen durch Matrizen $\mathbf{A}_{\mathcal{H}}$ (H-Matrizen) zu approximieren, für die sämtliche arithmetische Operationen sowie Speicherung und MVM in logarithmisch linearer Komplexität durchführbar sind.

Die vorgestellten Techniken können unter Anderem auf Matrizen

$$\mathbf{A}_{ij} = f(x_i, x_j), \quad x_i \in \mathbb{R}^d, i = 1, \ldots, N \tag{1.1}$$

angewendet werden, wobei f (asymptotisch) glatt ist (beispielsweise eine Kovarianzfunktion oder Gravitationspotentiale) sowie auf (Galerkin-)Diskretisierungen von Bilinearformen

$$\mathbf{A}_{ij} = a(\phi_i, \phi_j), \tag{1.2}$$

wobei $a : X_N \times X_N \to \mathbb{R}$ eine Bilinearform, X_N ein endlich dimensionaler Vektorraum und $\{\phi_i, i = 1 \dots N\}$ eine Basis von X_N ist. Arithmetische Operationen für derartige Matrizen haben im Allgemeinen Komplexität $\mathcal{O}(N^2)$.

1.2 Einführendes Beispiel

Auf dem Intervall $I = [0, 1]$ betrachten wir die Integralgleichung

$$V\phi(x) := \int_0^1 \log|x - y|\, \phi(y) dy = f(x), \quad x \in [0, 1].$$

Der Operator V ist ein Faltungsoperator mit Faltungskern $\kappa(x, y) := \log|x - y|$.

Sei $I = \bigcup_{i=0}^{N-1} I_i$ wobei $I_i = [\frac{i}{N}, \frac{i+1}{N}]$ und X_N der Raum der stückweise konstanten Polynome auf der Zerlegung $\{I_i : i = 0, \dots, N-1\}$. $\{\phi_i \in L^2(I) : i = 0, \dots, N-1\}$ seien die stückweise konstanten Basisfunktionen

$$\phi_i(x) = \begin{cases} 1 & x \in [\frac{i}{N}, \frac{i+1}{N}] \\ 0 & \text{sonst.} \end{cases}$$

Die zur Integralgleichung gehörige Bilinearform $a(\cdot, \cdot)$ erhält man mittels Multiplikation mit einer (geeigneten) Testfunktion und Integration über I. Unsere Galerkin-Matrix $\mathbf{A}$ ist also gegeben als

$$\mathbf{A}_{ij} = a(\phi_j, \phi_i) = \langle V\phi_j, \phi_i \rangle_{L^2(I)} = \int_{I_i} \int_{I_j} \log|x - y|\, dydx.$$

Da V ein Faltungsoperator ist und somit lediglich $\operatorname{supp} V\phi_i \subset \operatorname{supp} \phi_i + \operatorname{supp} \kappa = I$ gilt, ist die Matrix $\mathbf{A}$ vollbesetzt.

Wählt man beispielsweise $N = 32768$ so benötigt eine Speicherung von $\mathbf{A}$ 8 GB Speicher ($8N^2$ Byte).

Ziel: Reduktion des Speicheraufwands durch geeignete Approximation von $\mathbf{A}$.

Idee: Kernfunktion $\kappa(x, y)$ ist glatt, wenn $x \neq y$ bzw. für alle $(x, y) \in I_\tau \times I_\sigma$ wobei $I_\tau = [a, b]$, $I_\sigma = [c, d]$, $0 \leq a < b < c < d \leq 1$. Weiters gilt für $m \in \mathbb{N}_0$

$$\partial_x^m \kappa(x, y) = (-1)^{m-1}(m-1)!(x - y)^{-m}.$$

$\Longrightarrow$ approximiere $\kappa|_{I_\tau \times I_\sigma}$ durch abgeschnittene Taylorreihe um Punkt $x_0 = (a+b)/2$

$$\begin{aligned}\widetilde{\kappa}_k(x,y) &:= \sum_{m=0}^{k-1} \frac{1}{m!} \partial_x^m \kappa(x_0,y)(x-x_0)^m \\ &= \kappa(x_0,y) + \sum_{m=1}^{k-1} (-1)^{m-1} \frac{(m-1)!}{m!} \frac{(x-x_0)^m}{(x_0-y)^m}.\end{aligned}$$

Der Approximationsfehler lässt sich abschätzen durch

$$|\kappa(x,y) - \widetilde{\kappa}_k(x,y)| \leq \sum_{m=k}^{\infty} \frac{1}{m} \frac{|x-x_0|^m}{|x_0-y|^m} \leq \frac{1}{k} \frac{\left|\frac{x-x_0}{x_0-y}\right|^k}{1-\left|\frac{x-x_0}{x_0-y}\right|}.$$

Es gilt

$$\left|\frac{x-x_0}{x_0-y}\right| \leq \frac{\operatorname{diam}(I_\tau)/2}{\operatorname{diam}(I_\tau)/2 + \operatorname{dist}(I_\tau, I_\sigma)} = \frac{1}{1+2\frac{\operatorname{dist}(I_\tau,I_\sigma)}{\operatorname{diam}(I_\tau)}}.$$

Wir verlangen in weiterer Folge eine stärkere Bedingung als lediglich $b < c$ (oder $0 < \operatorname{dist}(I_\tau, I_\sigma)$), nämlich eine so genannte Zulässigkeitsbedingung

$$\operatorname{diam}(I_\tau) \leq \operatorname{dist}(I_\tau, I_\sigma). \tag{1.3}$$

Damit erhält man

$$\left|\frac{x-x_0}{x_0-y}\right| \leq \frac{1}{3}$$

und somit

$$|\kappa(x,y) - \widetilde{\kappa}_k(x,y)| \leq \frac{3}{2k} 3^{-k}.$$

Wir wählen Teilmengen $\tau, \sigma \subset \{1,\ldots,N\}$ sodass

$$I_\tau := \bigcup_{i\in\tau} \operatorname{supp}\phi_i, \quad I_\sigma := \bigcup_{i\in\sigma} \operatorname{supp}\phi_i$$

zusammenhängende Intervalle sind und die Zulässigkeitsbedingung (1.3) erfüllt ist.

Setzt man die Approximation in die Darstellung von $\mathbf{A}$ ein, so erhält man für alle $i \in \tau, j \in \sigma$, dass

$$\begin{aligned}\mathbf{A}_{ij} &= \int_I \int_I \phi_i(x) \log|x-y| \, \phi_j(y) dy dx = \int_{I_\tau} \int_{I_\sigma} \phi_i(x) \log|x-y| \, \phi_j(y) dy dx \\ &\simeq \int_{I_\tau} \int_{I_\sigma} \phi_i(x) \widetilde{\kappa}_k(x,y) \phi_j(y) dy dx \\ &= \sum_{m=0}^{k-1} \underbrace{\frac{1}{m!} \int_{I_\tau} \phi_i(x)(x-x_0)^m dx}_{\mathbf{V}_{im}} \underbrace{\int_{I_\sigma} \phi_j(y) \partial_x^m \kappa(x_0,y) dy}_{\mathbf{W}_{jm}}\end{aligned}$$

also

$$\mathbf{A}|_{\tau\times\sigma} \simeq \mathbf{V}\mathbf{W}^T$$

und $\mathbf{V} \in \mathbb{R}^{|\tau|\times k}$, $\mathbf{W} \in \mathbb{R}^{|\sigma|\times k}$. Eine Speicherung von $\mathbf{V}$ benötigt somit nur $k\,|\tau|$ Speicher, eine von $\mathbf{W}$ $k\,|\sigma|$, in Summe also $k(|\tau| + |\sigma|)$ anstelle von $|\tau|\,|\sigma|$ für $\mathbf{A}|_{\tau\times\sigma}$.

Es gilt die Fehlerabschätzung

$$\begin{aligned}
\left|\mathbf{A}_{ij} - (\mathbf{V}\mathbf{W}^T)_{ij}\right| &= \left|\int_I\int_I \phi_i(x)(\kappa(x,y) - \widetilde{\kappa}_k(x,y))\phi_j(y)dxdy\right| \\
&\lesssim \|\kappa - \widetilde{\kappa}_k\|_{L^\infty(I_i\times I_j)} \int_{I_i}\phi_i(x)dx \int_{I_j}\phi_j(y)dy \\
&\lesssim \frac{3}{2}3^{-k}\int_{I_i}\phi_i(x)dx\int_{I_j}\phi_j(y)dy = \frac{3}{2}N^{-2}3^{-k}
\end{aligned}$$

also exponentielle Konvergenz des Fehlers für jeden Matrixeintrag.

Zerlegt man also die Matrix $\mathbf{A}$ geeignet in zulässige Blöcke $\tau\times\sigma$ (vgl. Abbildung 1.1), und approximiert man auf diesen wie obig beschrieben, so erhält man eine Approximation, die exponentiell (z.B. in der Frobeniusnorm) gegen die ursprüngliche Matrix konvergiert und einen deutlich geringeren (sofern $k << \min(|\tau|, |\sigma|)$) Speicherbedarf hat.

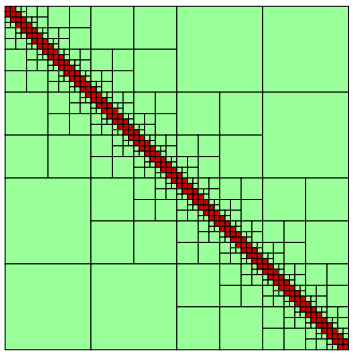


Abbildung 1.1: Block-Struktur einer hierarchischen Matrix

Tabelle 1.2 zeigt die Rechenzeiten und den Speicherbedarf für das Berechnen von der vollbesetzten Matrix $\mathbf{A}$ sowie für die Approximation $\mathbf{A}_{\mathcal{H}}$, bei der die Einträge in den grün markierten Blöcken in Abbildung 1.1 durch $\mathbf{V}\mathbf{W}^T$ ersetzt wurden. Weiters wird der relative Approximationsfehler angegeben. Man erkennt, dass der Fehler schnell konvergiert, der Speicheraufwand aber nur langsam wächst.

k	Speicherbedarf (MB)	Komprimiert auf (%)	Rechenzeit (s) Aufstellen von Matrix	relativer Fehler $\|\mathbf{A} - \mathbf{A}_{\mathcal{H}}\|_2 \,/\, \|\mathbf{A}\|_2$
Vollbesetzt	8192		253	
1	27.9	0.3	0.52	4.6e-02
2	41.4	0.5	0.67	6.9e-03
3	54.9	0.7	0.69	7.1e-04
4	68.4	0.8	0.71	1.7e-04
5	81.9	1.0	0.73	2.7e-05

Tabelle 1.1: Rechenzeiten und Speicheraufwand für das Berechnen von $\mathbf{A}$ und der Approximation $\mathbf{A}_{\mathcal{H}}$.

2 Algebraische Struktur von Hierarchischen Matrizen

Wir werden im Folgenden das Format der Hierarchischen Matrizen (H-Matrizen) formal definieren und deren Aufwand analysieren.

Im Allgemeinen ist es nicht möglich eine Niedrigrangfaktorisierung für die gesamte Matrix zu finden, aber oftmals können gewisse (große) Blöcke approximiert werden. Wir werden in diesem Abschnitt spezifizieren, welche Blöcke für eine Approximation geeignet sind. Wir beginnen mit der Definition eines Cluster-Baumes, welcher die den hierarchischen Matrizen zu Grunde liegende Blockstruktur bzw. Matrixpartition beschreiben wird.

2.1 Cluster-Baum

Sei $\mathcal{I} = \{1, \ldots, n\}$ eine Indexmenge. Eine Teilmenge $\tau \subset \mathcal{I}$ heißt *Cluster*.

Im Folgenden bezeichnet $|\tau|$ die Kardinalität der endlichen Menge τ.

Definition 2.1 (Cluster-Baum)**.** *Sei $n_{\text{blatt}} \in \mathbb{N}$. Ein* Cluster-Baum *$\mathbb{T}_{\mathcal{I}}$ ist ein binärer Baum mit den folgenden Eigenschaften*

- *$\mathcal{I}$ ist die Wurzel von $\mathbb{T}_{\mathcal{I}}$,*
- *Jeder Knoten $\tau \in \mathbb{T}_{\mathcal{I}}$ ist eine nicht-leere Teilmenge von $\mathcal{I}$,*
- *Jeder Knoten $\tau \in \mathbb{T}_{\mathcal{I}}$ ist entweder Blatt des Baumes oder hat genau zwei eindeutige Söhne $\tau', \tau'' \in \mathbb{T}_{\mathcal{I}}$ mit $\tau = \tau' \cup \tau''$, $\tau' \cap \tau'' = \emptyset$,*
- *Für jedes Blatt $\tau \in \mathbb{T}_{\mathcal{I}}$ gilt $|\tau| \leq n_{\text{blatt}}$.*

Die Menge der Söhne eines Clusters τ wird mit $\operatorname{sons}(\tau)$ *bezeichnet, die Menge der Blätter eines Cluster-Baumes mit* $\operatorname{leaves}(\mathbb{T}_{\mathcal{I}})$.

Die Konstante n_{blatt} hängt nicht von dem jeweiligen Cluster τ ab. Bei der Zerteilung von Blöcken hört man in den Berechnungen auf, wenn sie kleiner als n_{blatt} sind, Blöcke in der Blockstruktur werden also nicht zu klein (oder sogar einelementig), sodass für

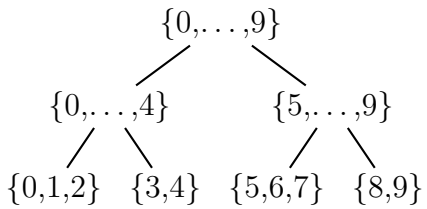


Abbildung 2.1: Cluster-Baum mit Wurzel $\mathcal{I} = \{0, \ldots, 9\}$ und $n_{\text{blatt}} = 3$.

diese eine Approximation nicht aufwändiger wäre als den exakten Block zu verwenden.

Ein Beispiel eines Cluster-Baumes mit Wurzel $\mathcal{I} = \{0, \ldots, 9\}$ und $n_{\text{blatt}} = 3$ findet sich in Abbildung 2.1. Dieser Cluster-Baum induziert eine Blockzerlegung eines Vektors:

x_0 x_1 x_2	x_3 x_4	x_5 x_6 x_7	x_8 x_9

Lemma 2.2. *Für einen Cluster-Baum $\mathbb{T}_\mathcal{I}$ gilt*

$$|\mathbb{T}_\mathcal{I}| \leq 2\,|\text{leaves}(\mathbb{T}_\mathcal{I})| - 1 \leq 2n - 1, \tag{2.1}$$

also die Anzahl der Knoten im Baum $\mathbb{T}_\mathcal{I}$ ist von der Ordnung $\mathcal{O}(n)$.

Beweis. Wir zeigen das Lemma mittels Induktion nach $n = |\mathcal{I}|$. Für $n = 1$ ist die Aussage klarerweise erfüllt. Für $n > 1$ und $|\text{leaves}(\mathbb{T}_\mathcal{I})| > 1$, betrachten wir die Söhne σ, τ der Wurzel $\mathcal{I}$ und wenden die Induktionsvoraussetzung für die Äste $\mathbb{T}_\sigma$, $\mathbb{T}_\tau$ mit Wurzeln σ, τ an. Das führt auf

$$\begin{aligned} |\mathbb{T}_\mathcal{I}| &= |\mathbb{T}_\sigma| + |\mathbb{T}_\tau| + 1 \leq \Big(2\,|\text{leaves}(\mathbb{T}_\sigma)| - 1\Big) + \Big(2\,|\text{leaves}(\mathbb{T}_\tau)| - 1\Big) + 1 \\ &= 2\,|\text{leaves}(\mathbb{T}_\mathcal{I})| - 1, \end{aligned}$$

wobei verwendet wurde, dass Äste eines Cluster-Baumes disjunkt sind. □

Eine wichtige Größe eines Cluster-Baumes ist die Baumtiefe, welche mit Hilfe der Level-Funktion angegeben wird.

Definition 2.3 (Level-Funktion, Tiefe)**.** *Für einen Cluster-Baum $\mathbb{T}_\mathcal{I}$ ist die* Level-Funktion $\text{level} : \mathbb{T}_\mathcal{I} \to \mathbb{N}_0$ *induktiv definiert durch*

$$\text{level}(\mathcal{I}) = 0 \quad \textit{und} \quad \text{level}(\tau') := \text{level}(\tau) + 1,$$

falls $\tau' \in \text{sons}(\tau)$ ist.

Die Tiefe *eines Cluster-Baumes ist*

$$\text{depth}(\mathbb{T}_\mathcal{I}) := \max_{\tau \in \mathbb{T}_\mathcal{I}} \text{level}(\tau).$$

Mit Hilfe von Cluster-Bäumen lassen sich Vektoren partitionieren. Für die Partition von Matrizen benötigt man hingegen so genannte Block-Cluster-Bäume.

Würde man obige Definition für die Wurzel $\mathcal{I} \times \mathcal{J}$ direkt verwenden, so würde man einen Block-Cluster-Baum mit potentiell $|\mathcal{I}|\,|\mathcal{J}|$-Knoten erhalten, was einen unnötig großen Aufwand darstellt. Im folgenden konstruieren wir einen Block-Cluster-Baum direkt aus den Cluster-Bäumen $\mathbb{T}_{\mathcal{I}}, \mathbb{T}_{\mathcal{J}}$.

Definition 2.4. *Für Indexmengen $\mathcal{I} = \{1, \ldots, n\}$, $\mathcal{J} = \{1, \ldots, m\}$ und Cluster-Bäume $\mathbb{T}_{\mathcal{I}}, \mathbb{T}_{\mathcal{J}}$ ist der Produkt-Cluster-Baum $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ definiert als induktiv definierter Baum mit*

- *Wurzel $\mathcal{I} \times \mathcal{J}$*
- *Jeder Knoten $b \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ hat die Darstellung $b = \tau \times \sigma$ mit $\tau \in \mathbb{T}_{\mathcal{I}}, \sigma \in \mathbb{T}_{\mathcal{J}}$.*
- *Die Söhne eines Knoten $b = \tau \times \sigma$ sind $\operatorname{sons}(b) = \emptyset$ oder gegeben durch*

$$\operatorname{sons}(\tau \times \sigma) := \begin{cases} \tau \times \operatorname{sons}(\sigma), & \text{falls } \operatorname{sons}(\tau) = \emptyset, \operatorname{sons}(\sigma) \neq \emptyset \\ \operatorname{sons}(\tau) \times \sigma, & \text{falls } \operatorname{sons}(\tau) \neq \emptyset, \operatorname{sons}(\sigma) = \emptyset \\ \operatorname{sons}(\tau) \times \operatorname{sons}(\sigma), & \text{falls } \operatorname{sons}(\tau) \neq \emptyset, \operatorname{sons}(\sigma) \neq \emptyset. \end{cases}$$

Bei der Analyse des Aufwands sämtlicher Operationen mit hierarchischen Matrizen werden stets sowohl die Baumtiefe als auch die Anzahl der Blätter des zugehörigen Cluster-Baumes auftreten.

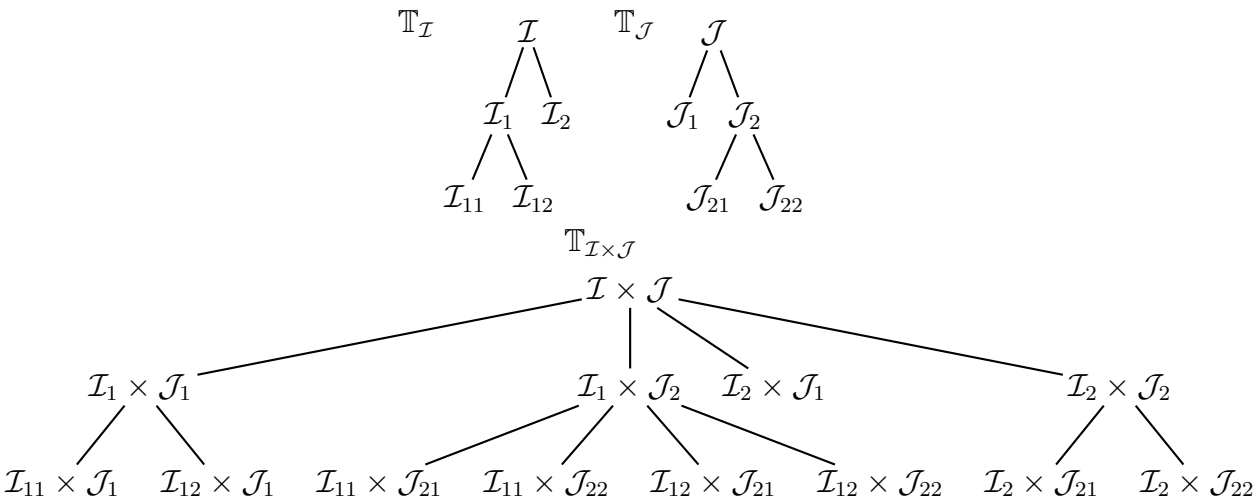


Abbildung 2.2: Beispiel eines Produkt-Cluster-Baumes $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$.

2.2 Zulässige Partitionen, hierarchische Matrizen

Im Folgenden wird spezifiziert auf welchen Blöcken geeignete niedrigdimensionale Approximationen gefunden werden können. Dies wird durch eine so genannte *Zulässigkeitsbedingung* entschieden (vgl. (1.3) für das 1D-Modellbeispiel). Die konkrete Wahl der Zulässigkeitsbedingung hängt oft von dem betrachteten Problem ab, wir werden in Definition 2.6 eine klassische Wahl vorstellen.

Wir assoziieren mit jedem $i \in \mathcal{I}$ eine Teilmenge $X_i \subset \mathbb{R}^d$ und schreiben für einen Cluster $\tau \subset \mathcal{I}$

$$X_\tau := \bigcup_{i \in \tau} X_i.$$

Die Mengen X_i können z.B. einpunktige Mengen oder (meistens, z.B. bei FEM/BEM) Träger von Basisfunktionen sein, also $X_i = \operatorname{supp} \phi_i$.

Abstrakt ist die Zulässigkeitsbedingung eine Abbildung

$$\mathrm{Adm} : \mathbb{T}_{\mathcal{I} \times \mathcal{J}} \to \{\mathrm{true}, \mathrm{false}\},$$

wobei $\mathrm{Adm}(\tau \times \sigma) = \mathrm{true} \implies \mathrm{Adm}(\tau' \times \sigma) = \mathrm{true}$ für $\tau' \in \mathrm{sons}(\tau)$ (und die selbe Aussage für σ).

Beispiel 2.5. *Wir betrachten abermals das 1D-Modellbeispiel aus Kapitel 1. Eine Variante der Zulässigkeitsbedingung ist* (1.3)*, also* $\mathrm{Adm}(\tau \times \sigma) = \mathrm{true}$ *falls*

$$\mathrm{diam}(X_\tau) \leq \mathrm{dist}(X_\tau, X_\sigma),$$

wobei Durchmesser und Distanz der Mengen X_τ und X_σ definiert sind als

$$\begin{aligned} \mathrm{diam}(X_\tau) &:= \max\{|x - y| : x, y \in X_\tau\} \\ \mathrm{dist}(X_\tau, X_\sigma) &:= \min\{|x - y| : x \in X_\tau, y \in X_\sigma\}. \end{aligned}$$

In der Praxis kann allerdings die Berechnung dieser Größen sehr aufwendig sein, weswegen üblicherweise die Mengen X_τ und X_σ durch achsenparallele Minimalquader, die diese Mengen enthalten, ersetzt werden.

Im Folgenden werden derartige Minimalquader, so genannte Bounding-Boxen, definiert, sowie eine Variante der Zulässigkeitsbedingung, die speziell bei $\mathcal{H}$-Matrizen mittels Interpolation von Kernfunktionen zum Einsatz kommt, vorgestellt.

Definition 2.6 (Bounding-Box, η-Zulässigkeit)**.** *Seien $\tau \subset \mathcal{I}, \sigma \subset \mathcal{J}$ Cluster und $\eta > 0$.*

- *Dann heißt der achsenorientierte Minimalquader $B_\tau := \Pi_{i=1}^d [a_i, b_i]$ mit $B_\tau \supseteq X_\tau$* Bounding-Box *für τ.*

- *Ein Cluster-Paar $\tau \times \sigma \subset \mathcal{I} \times \mathcal{J}$ heißt* η-zulässig, *falls zugehörige Bounding-Boxen B_τ, B_σ existieren, die folgende Bedingung erfüllen:*

$$\min\{\operatorname{diam}(B_\tau), \operatorname{diam}(B_\sigma)\} \leq \eta \operatorname{dist}(B_\tau, B_\sigma). \tag{2.2}$$

Die nachfolgende Graphik zeigt eine Illustration der Zulässigkeitsbedingung.

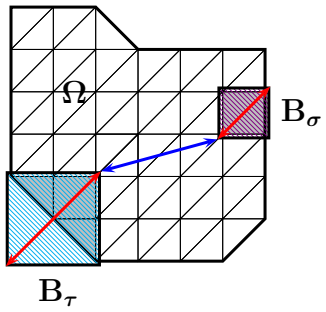


Abbildung 2.3: Bounding-Boxen und Zulässigkeit.

Basierend auf der Zulässigkeitsbedingung kann eine *zulässige Partition* der Produktindexmenge $\mathcal{I} \times \mathcal{J}$ erhalten werden.

Definition 2.7 (Zulässige Partition, Nah- und Fernfeld)**.** *Seien $\mathcal{I}, \mathcal{J}$ Indexmengen und* Adm *eine Zulässigkeitsbedingung.*

- *$\mathbb{P}$ heißt* auf einem Cluster-Baum $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ basierende Partition *der Produktindexmenge $\mathcal{I} \times \mathcal{J}$, falls*

$$\begin{aligned} & b, b' \in \mathbb{P} \quad \Longrightarrow \ (b = b' \ \vee \ b \cap b' = \emptyset) \\ & \bigcup_{b\in\mathbb{P}} b = \mathcal{I} \times \mathcal{J} \\ & \mathbb{P} \subset \mathbb{T}_{\mathcal{I}\times\mathcal{J}}. \end{aligned}$$

- *Eine derartige Partition heißt* zulässig *falls alle $\tau \times \sigma \in \mathbb{P}$ entweder* $\operatorname{Adm}(\tau \times \sigma) = \text{true}$ *erfüllen, oder* $\operatorname{sons}(\tau) = \emptyset \vee \operatorname{sons}(\sigma) = \emptyset$ *gilt.*

Für eine zulässige Partition $\mathbb{P}$ definieren wir das Fernfeld $\mathbb{P}_{\text{far}}$

$$\mathbb{P}_{\text{far}} := \{\tau \times \sigma \in \mathbb{P} \ : \ \tau \times \sigma \textit{ ist } \eta\textit{-zulässig}\} \tag{2.3}$$

und das Nahfeld $\mathbb{P}_{\text{near}}$ *als*

$$\mathbb{P}_{\text{near}} := \mathbb{P} \backslash \mathbb{P}_{\text{far}}. \tag{2.4}$$

Eine wichtige Größe für zulässige Partitionen wird durch die so genannte Schwachbesetztheitskonstante C_{sp} angegeben. Für schwachbesetzte Matrizen ist die maximale Anzahl an Nicht-Null-Elementen pro Zeile/Spalte

$$\max_{i\in\mathcal{I}} |\{j \in \mathcal{J} : \mathbf{A}_{ij} \neq 0\}| \quad \text{bzw.} \quad \max_{j\in\mathcal{J}} |\{i \in \mathcal{I} : \mathbf{A}_{ij} \neq 0\}|$$

ein Maß für die Schwachbesetztheit.

Die Konstante C_{sp} ist ein ähnliches Maß für Block-Partitionen und misst die maximale Anzahl an Zeilen-/Spaltenblöcken in der Partition. Daher wird C_{sp} in sämtlichen Rechen- und Speicheraufwandbetrachtungen auftreten.

Definition 2.8 (Schwachbesetztheitskonstante)**.** *Für eine zulässige Partition gemäß Definition 2.7 ist die* Schwachbesetztheitskonstante *C_{sp} gegeben als*

$$C_{\text{sp}} := \max\left\{\max_{\tau\in\mathbb{T}_{\mathcal{I}}} |\{\sigma \in \mathbb{T}_{\mathcal{J}} \,:\, \tau\times\sigma \in \mathbb{P}\}|, \max_{\sigma\in\mathbb{T}_{\mathcal{J}}} |\{\tau \in \mathbb{T}_{\mathcal{I}} \,:\, \tau\times\sigma \in \mathbb{P}\}|\right\}. \tag{2.5}$$

Das folgende Lemma liefert eine Abschätzung an die Anzahl der Blöcke in einer zulässigen Partition mit Hilfe von C_{sp}.

Lemma 2.9. *Für die Anzahl der Blöcke einer zulässigen Partition $\mathbb{P}$ gilt*

$$|\mathbb{P}| \leq (2\min\{m,n\} - 1)C_{\text{sp}}.$$

Beweis. Es gilt mit Lemma 2.2

$$|\mathbb{P}| = \sum_{\tau\times\sigma\in\mathbb{P}} 1 = \sum_{\tau\in\mathbb{T}_{\mathcal{I}}} |\{\sigma \in \mathbb{T}_{\mathcal{J}} : \tau\times\sigma \in \mathbb{P}\}| \leq \sum_{\tau\in\mathbb{T}_{\mathcal{I}}} C_{\text{sp}} \leq (2n-1)C_{\text{sp}}.$$

Vertauscht man die Rollen von τ und σ erhält man die gleiche Abschätzung mit m anstelle von n. □

Mit Hilfe einer zulässigen Partition können wir nun hierarchische Matrizen als Blockmatrizen definieren, wobei zulässige Blöcke niedrigen Rang haben.

Definition 2.10 ($\mathcal{H}$-Matrizen)**.** *Sei $\mathbb{P}$ eine zulässige Partition von $\mathcal{I}\times\mathcal{J}$ gemäß Definition 2.7. Eine Matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ heißt* $\mathcal{H}$-Matrix (hierarchische Matrix) *mit maximalem Rang r, falls für alle zulässigen Blöcke $\tau\times\sigma \in \mathbb{P}_{\text{far}}$ gilt, dass*

$$\mathbf{A}|_{\tau\times\sigma} = \mathbf{X}_{\tau\sigma}\mathbf{Y}_{\tau\sigma}^T$$

mit Matrizen $\mathbf{X}_{\tau\sigma} \in \mathbb{R}^{|\tau|\times r}$, $\mathbf{Y}_{\tau\sigma} \in \mathbb{R}^{|\sigma|\times r}$.

Wir schreiben $\mathcal{H}(r,\mathbb{P})$ für die Menge aller $\mathcal{H}$-Matrizen zur Partition $\mathbb{P}$ und maximalem blockweisen Rang r.

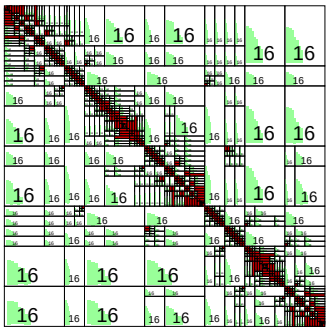


Abbildung 2.4: Beispiel einer Struktur einer $\mathcal{H}$-Matrix, rote Blöcke werden vollbesetzt gespeichert, grüne Blöcke haben Rang ≤ 16.

2.3 Speicheraufwand hierarchischer Matrizen

Proposition 2.11. *Der Speicherbedarf* N_{Storage} *einer* $\mathcal{H}$*-Matrix* $\mathbf{A} \in \mathcal{H}(r, \mathbb{P})$ *ist beschränkt durch*

$$N_{\text{Storage}} \leq C_{\text{sp}}(\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) + 1) \max\{n_{\text{blatt}}, r\}(m+n) = \mathcal{O}(rN \log N),$$

wobei $N = m + n$ *(falls* $\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) \sim \log(N)$*).*

Beweis. Nach Definition 2.10 ist der Speicheraufwand die Summe des Speicherbedarfs der zulässigen Blöcke (niedrigdimensional) und nicht zulässigen Blöcke (kleine Blockgröße). Für einen zulässigen Block $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ ist der Speicherbedarf beschränkt durch $r(|\tau| + |\sigma|)$. Ein nicht zulässiger Block $\tau \times \sigma \in \mathbb{P}_{\text{near}}$ muss nach Definition 2.7 entweder $|\tau| \leq n_{\text{blatt}}$ oder $|\sigma| \leq n_{\text{blatt}}$ erfüllen, und somit $|\tau||\sigma| \leq n_{\text{blatt}}(|\tau| + |\sigma|)$. Daher gilt

$$N_{\text{Storage}} = \sum_{\tau\times\sigma\in\mathbb{P}_{\text{far}}} r(|\tau| + |\sigma|) + \sum_{\tau\times\sigma\in\mathbb{P}_{\text{near}}} |\tau||\sigma| \leq \max\{n_{\text{blatt}}, r\} \sum_{\tau\times\sigma\in\mathbb{P}} (|\tau| + |\sigma|).$$

Mit Hilfe der Schwachbesetztheitskonstante C_{sp} aus Definition 2.8 erhalten wir

$$\sum_{\tau\times\sigma\in\mathbb{P}} |\tau| = \sum_{\ell=0}^{\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}})} \sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}} \\ \text{level}(\tau)=\ell}} \sum_{\substack{\sigma\in\mathbb{T}_{\mathcal{J}} \\ \tau\times\sigma\in\mathbb{P}}} |\tau| \leq C_{\text{sp}} \sum_{\ell=0}^{\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}})} \sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}} \\ \text{level}(\tau)=\ell}} |\tau|.$$

Auf jedem Level ℓ des Cluster-Baumes gilt laut Konstruktion von $\mathbb{T}_{\mathcal{I}}$, da Söhne disjunkt sind, dass

$$\sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}} \\ \text{level}(\tau)=\ell}} |\tau| \leq |\mathcal{I}| = n.$$

Somit folgt

$$\sum_{\tau\times\sigma\in\mathbb{P}} |\tau| \leq C_{\text{sp}}(\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) + 1)n,$$

und das selbe Argument für $|\sigma|$ liefert die gewünschte Aussage. □

Bemerkung 2.12. *Der Beweis von Proposition 2.11 liefert die Abschätzung*

$$\sum_{\tau\times\sigma\in\mathbb{P}} (|\tau| + |\sigma|) \leq C_{\mathrm{sp}}(\mathrm{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) + 1)(m+n), \tag{2.6}$$

welche in weiterer Folge öfters angewendet werden wird.

2.4 Konstruktion von zulässigen Partitionen

Ein Cluster-Baum und in weiterer Folge eine zulässige Partition kann auf verschiedene Arten konstruiert werden. Da die Anzahl der Blätter (vgl. C_{sp}) sowie die Baumtiefe in die Komplexitätsabschätzungen eingehen, möchte man einen Baum mit möglichst geringer Tiefe und dass Blätter so bald wie möglich zulässig werden.

Definition 2.13. *Wir nennen einen Cluster-Baum $\mathbb{T}_{\mathcal{I}}$* balanciert, *falls*

$$C_{\mathrm{bal}} := \min_{\tau\in\mathbb{T}_{\mathcal{I}}\setminus\mathrm{leaves}(\mathbb{T}_{\mathcal{I}})} \left\{ \frac{|\tau'|}{|\tau''|} \ : \ \tau', \tau'' \in \mathrm{sons}(\tau) \right\} \geq C > 0,$$

wobei die Konstante $C > 0$ nicht von n abhängt.

Der nachfolgende (abstrakte) Algorithmus erzeugt zunächst rekursiv mit dem Aufruf `CreateClusterTree(`$\emptyset$`,`$\mathcal{I}$`,`n_{blatt}`)` einen Cluster-Baum.

Algorithmus 2.14 (Erzeuge Cluster-Baum).

```
function CreateClusterTree(var T_I,τ,n_blatt)
if |τ| ≤ n_blatt
  return
else
  Zerlege τ in disjunkte Söhne τ', τ''
  Füge τ' zu T_I hinzu und rufe CreateClusterTree(T_I,τ',n_blatt) auf
  Füge τ'' zu T_I hinzu und rufe CreateClusterTree(T_I,τ'',n_blatt) auf
end
```

Algorithmus 2.14 spezifiziert nicht, wie ein Cluster τ zerlegt wird. Wir werden im Folgenden zwei verschiedene Möglichkeiten hierfür vorstellen, das so genannte *geometrische Clustering* und das *kardinalitäsbasierte Clustering*.

Für die tatsächliche Implementierung müssen die Mengen X_i, die die geometrische Information zugehörig zu einem Element der Indexmenge beinhalten, vorhanden sein. Um die Implementierung zu Vereinfachen ersetzt man die Mengen X_i durch so genannte *charakteristische Punkte* ξ_i, beispielsweise den Schwerpunkt von X_i.

Algorithmus 2.15 (Zerlege geometrisch).

`function SplitGeometric(`τ`,var` τ'`, var` τ''`)`
erzeuge Boundingbox $B_\tau := \Pi[a_i, b_i]$ für τ (finde größte und kleinste Koordinaten in τ)
finde größte Kante (j mit $b_j - a_j = \max_{1\le i\le d} b_i - a_i$)
$B_{\tau'} := \Pi_{i=1}^{j-1}[a_i, b_i] \times [a_j, a_j + \frac{1}{2}(b_j - a_j)] \times \Pi_{i=j+1}^{d}[a_i, b_i]$
$B_{\tau''} := \Pi_{i=1}^{j-1}[a_i, b_i] \times [a_j + \frac{1}{2}(b_j - a_j), b_j] \times \Pi_{i=j+1}^{d}[a_i, b_i]$
`for` $i \in \tau$
  `if` $\xi_i \in B_{\tau'}$
    $\tau' = \tau' \cup \{i\}$
  `else`
    $\tau'' = \tau'' \cup \{i\}$
  `end`
`end`

Bemerkung 2.16.

- *Algorithmus 2.15 kann auf unbalancierte Bäume führen, also im Extremfall ($|\tau'| = |\tau| - 1$) wäre* $\operatorname{depth}(\mathbb{T}_\mathcal{I}) \sim \mathcal{O}(n)$.
- *Verlangt man beispielsweise für den Fall, dass $X_i = \operatorname{supp}\phi_i$ mit Basisfunktionen ϕ_i, zusätzlich, dass alle Mengen X_i etwa gleich groß sind und ähnliche Form haben (quasi-uniformes, formreguläres Gitter), dann gilt hingegen* $\operatorname{depth}(\mathbb{T}_\mathcal{I}) \sim \mathcal{O}(\log n)$.
- *In diesem Fall gilt auch $C_{\mathrm{sp}} \le C$, wobei die Konstante C unabhängig von n ist und die Konstruktion von $\mathbb{T}_\mathcal{I}$ hat einen Aufwand von $\mathcal{O}(n \log n)$.*

Eine Alternative zu Algorithmus 2.15 liefert die folgende Variante zur Zerlegung von Clustern.

Algorithmus 2.17 (Zerlege kardinalitätsbasiert).

`function SplitCardinality(`τ`,var` τ'`, var` τ''`)`
erzeuge Boundingbox $B_\tau := \Pi[a_i, b_i]$ für τ (finde größte und kleinste Koordinaten in τ)
finde größte Kante (j mit $b_j - a_j = \max_{1\le i\le d} b_i - a_i$)
sortiere $\tau = \{i_1, \ldots, i_{|\tau|}\}$ so dass $(\xi_{i_k})_j \le (\xi_{i_\ell})_j$ für $1 \le k \le \ell \le |\tau|$
$\tau' := \{i_1, \ldots, i_{\lceil|\tau|/2\rceil}\}$
$\tau'' := \{i_{\lceil|\tau|/2\rceil+1,\ldots,i_{|\tau|}}\}$

Bemerkung 2.18.

- *Bei kardinalitäsbasiertem Clustering erfüllen die Söhne stets $||\tau'| - |\tau''|| \le 1$, somit folgt $C_{\mathrm{bal}} \ge \frac{n_{\mathrm{blatt}}}{n_{\mathrm{blatt}}+2}$ und mit dem nachfolgenden Lemma, dass der Cluster-Baum stets Tiefe $\mathcal{O}(\log n)$ hat.*
- *Der Aufwand von Algorithmus 2.17 ist $\mathcal{O}(n \log^2 n)$.*

Lemma 2.19. *Sei $\mathbb{T}_{\mathcal{I}}$ ein balancierter Cluster-Baum. Dann gilt*

$$\operatorname{depth}(\mathbb{T}_{\mathcal{I}}) \leq \frac{1}{\log(1 + C_{\text{bal}})} \log(n) + 1.$$

Beweis. Sei $\tau \in \mathbb{T}_{\mathcal{I}} \setminus \operatorname{leaves}(\mathbb{T}_{\mathcal{I}})$ ein Cluster und $\tau', \tau'' \in \operatorname{sons}(\tau)$ seine Söhne. Dann gilt

$$\frac{|\tau|}{|\tau'|} = \frac{|\tau'| + |\tau''|}{|\tau'|} \geq 1 + C_{\text{bal}}.$$

Seien $v_1, \ldots, v_L$ Knoten eines Astes im Cluster-Baum $\mathbb{T}_{\mathcal{I}}$ wobei $L := \operatorname{depth}(\mathbb{T}_{\mathcal{I}})$. Aus obiger Abschätzung folgt $|v_{i+1}| \leq \frac{1}{1+C_{\text{bal}}} |v_i|$, also induktiv

$$|v_L| \leq \left(\frac{1}{1 + C_{\text{bal}}}\right)^{L-1} |\mathcal{I}| .$$

Logarithmieren führt auf die gewünschte Abschätzung $(L-1)\log(1+C_{\text{bal}}) \leq \log(n/|v_L|) \leq \log(n)$.

□

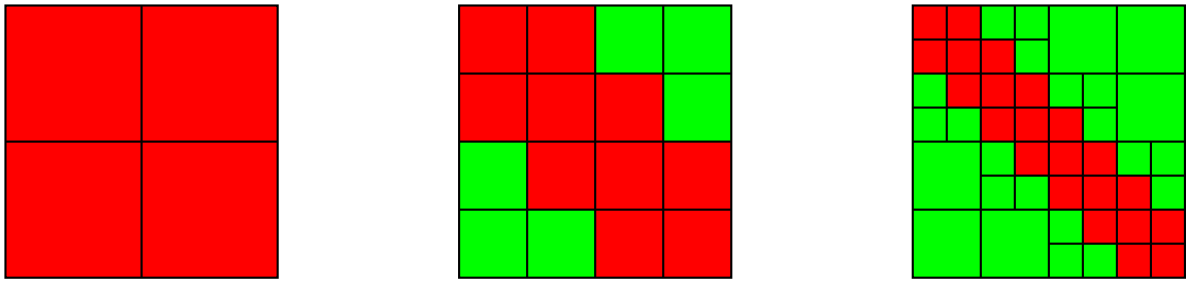


Abbildung 2.5: Blockzerlegung - zulässige Blöcke (grün), nicht zulässige Blöcke (rot)

Hat man Cluster-Bäume $\mathbb{T}_{\mathcal{I}}, \mathbb{T}_{\mathcal{J}}$, so kann man direkt mittels Definition 2.4 und der Zulässigkeitsbedingung Adm eine zulässige Partition $\mathbb{P} \subset \mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ induktiv konstruieren.

Algorithmus 2.20 (Erzeuge Block-Partition).

```
function CreateBlockPartitioning(var P_near, var P_far, τ, σ)
if τ × σ zulässig
  füge τ × σ zu P_far hinzu
elseif sons(τ) = ∅ = sons(σ)
  füge τ × σ zu P_near hinzu
else
  for all τ' × σ' ∈ sons(τ × σ)
    rufe CreateBlockPartitioning(var P_near, var P_far, τ', σ') auf
end
```

Bemerkung 2.21. *Die Zulässigkeitsbedingung basiert üblicherweise auf geometrischen Informationen gegeben durch die Mengen X_i. Mengen X_i, X_j, die z.B. nebeneinander liegen, müssen allerdings nicht benachbarte Indizes i, j haben. Nimmt man also einen zulässigen Matrix-Block $\tau \times \sigma$ so muss dieser entgegen der Intuition nicht zusammenhängend sein.*

Tatsächlich haben unsere Indexmengen $\mathcal{I}$, $\mathcal{J}$ keine vorgegebene Ordnung (die Technik der $\mathcal{H}$-Matrizen fordert keine Anordnung), es macht also in der Implementierung eventuell Sinn diese umzuordnen, z.B. $\{7, 2, 5, 4\} = \{2, 4, 7, 5\}$, falls z.B. $\{7, 2, 5, 4\}$ auf Grund der geometrischen Information in $\{2, 4\}$ und $\{7, 5\}$ zerlegt wird.

Bei der Implementierung muss man sich noch merken wie geometrische Freiheitsgrade zu tatsächlichen Indizes zu übersetzen sind (in HLib *heißt dies* `dof2idx`*).*

3 Approximation mit $\mathcal{H}$-Matrizen

Wir werden in diesem Abschnitt mögliche Methoden zur Konstruktion von Niedrigrang-Faktorisierungen angeben sowie deren Approximationsgüte analysieren. Eine naheliegende Frage hierbei ist, ob es möglich und praktikabel ist die blockweise Bestapproximation mit Rang r zu konstruieren.

Wir werden den Fehler zwischen der ursprünglichen Matrix und der konstruierten $\mathcal{H}$-Matrix in zwei Matrix-Normen analysieren, der Frobenius-Norm und der Spektralnorm.

Definition 3.1. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$. Die* Frobenius-Norm *ist definiert als*

$$\|\mathbf{A}\|_F := \left(\sum_{j=1}^{n}\sum_{k=1}^{m} |\mathbf{A}_{jk}|^2\right)^{1/2}. \tag{3.1}$$

Die Spektralnorm *oder* ℓ_2-Operatornorm *ist die Matrixnorm induziert von der euklidischen Norm*

$$\|\mathbf{A}\|_2 := \sup_{\substack{x\in\mathbb{R}^m\\ x\neq 0}} \frac{\|\mathbf{A}x\|_2}{\|x\|_2}. \tag{3.2}$$

Elementares Nachrechnen zeigt, dass $\|\cdot\|_F, \|\cdot\|_2$ tatsächlich Normen auf $\mathbb{R}^{n\times m}$ sind. Als Normen auf einem endlichdimensionalen Vektorraum sind diese somit äquivalent. Das nachfolgende Lemma gibt die (scharfen) Konstanten für die Normäquivalenz explizit an.

Lemma 3.2. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$. Es gilt:*

(i) Die Normen $\|\cdot\|_F, \|\cdot\|_2$ sind invariant unter orthogonalen Matrizen.

(ii) Es gilt

$$\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F \leq \min\{\sqrt{n}, \sqrt{m}\}\, \|\mathbf{A}\|_2 .$$

Beweis. (i) Mit einer orthogonalen Matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$ gilt

$$\|\mathbf{AQ}\|_2 = \sup_{\substack{x\in\mathbb{R}^m\\ x\neq 0}} \frac{\|\mathbf{AQ}x\|_2}{\|x\|_2} = \sup_{\substack{x\in\mathbb{R}^m\\ x\neq 0}} \frac{\|\mathbf{AQ}x\|_2}{\|\mathbf{Q}x\|_2} = \sup_{\substack{y\in\mathbb{R}^m\\ y\neq 0}} \frac{\|\mathbf{A}y\|_2}{\|y\|_2} = \|\mathbf{A}\|_2 ,$$

und klarerweise auch $\|\mathbf{QA}\|_2 = \|\mathbf{A}\|_2$ für $\mathbf{Q} \in \mathbb{R}^{n\times n}$. Für die Frobenius-Norm folgt

$$\|\mathbf{AQ}\|_F^2 = \sum_{j,k} (\mathbf{AQ})_{jk}^2 = \sum_{j,k} \left(\sum_i \mathbf{A}_{ji}\mathbf{Q}_{ik}\right)\left(\sum_\ell \mathbf{A}_{j\ell}\mathbf{Q}_{\ell k}\right) = \sum_{j,i,\ell} \mathbf{A}_{ji}\mathbf{A}_{j\ell}\left(\sum_k \mathbf{Q}_{ik}\mathbf{Q}_{\ell k}\right).$$

$\mathbf{Q}$ ist orthogonal, also ist die innerste Summe als das Skalarprodukt der i-ten und ℓ-ten Spalte von $\mathbf{Q}$ genau dann nicht gleich Null, wenn $i = \ell$ und gleich 1 in diesem Fall. Somit folgt

$$\|\mathbf{AQ}\|_F^2 = \sum_{j,i,\ell} \mathbf{A}_{ji}\mathbf{A}_{j\ell}\delta_{i\ell} = \sum_{j,i} \mathbf{A}_{ji}^2 = \|\mathbf{A}\|_F^2 .$$

(ii) Für $x \in \mathbb{R}^m$, impliziert die Cauchy-Schwarz-Ungleichung

$$\|\mathbf{A}x\|_2 = \Big(\sum_{j=1}^{n} \Big|\sum_{k=1}^{m} \mathbf{A}_{jk}x_k\Big|^2\Big)^{1/2} \leq \Big(\sum_{j=1}^{n} \Big[\sum_{k=1}^{m} |\mathbf{A}_{jk}|^2\Big]\Big[\sum_{k=1}^{m} |x_k|^2\Big]\Big)^{1/2} = \|\mathbf{A}\|_F \|x\|_2 .$$

Dividiert man durch $\|x\|_2$ und nimmt man das Supremum so folgt $\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F$. Die obere Abschätzung für die Frobenius Norm folgt aus (i) und dem nachfolgenden Satz 3.3. □

3.1 Bestapproximation

Wir werden im Folgenden für beliebige Matrizen Bestapproximationen mit Rang r konstruieren. Wählt man diese als Approximationen auf jedem zulässigen Block der Partition, so erhält man eine (quasi-)Bestapproximation im $\mathcal{H}$-Matrix Format.

3.1.1 Singulärwertzerlegung

Der entscheidende Ausgangspunkt für die Konstruktion einer Bestapproximation mit Rang r ist die Singulärwertzerlegung.

Satz 3.3 (Singulärwertzerlegung)**.** *Für jede Matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ existieren eine eindeutige Diagonalmatrix $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ gegeben durch*

- $\mathbf{\Sigma}_{jk} = \sigma_j\delta_{jk}$, *also (für $n \leq m$)* $\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & \dots & 0 & \sigma_n & 0 & \cdots & 0 \end{pmatrix}$
- *mit den* Singulärwerten $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{n,m\}} \geq 0$,

sowie orthogonale Matrizen $\mathbf{U} \in \mathbb{R}^{n\times n}$ und $\mathbf{V} \in \mathbb{R}^{m\times m}$ sodass die Singulärwertzerlegung

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3.3}$$

gilt. □

Wir fassen einige Eigenschaften der Singulärwertzerlegung im Folgenden zusammen.

Bemerkung 3.4.

i) Die Singulärwerte σ_j von $\mathbf{M}$ sind genau die positiven Wurzeln der Eigenwerte von $\mathbf{M}^T\mathbf{M}$.

ii) Die Singulärwerte σ_j hängen stetig von der Matrix $\mathbf{M}$ ab.

iii) Es gilt $\operatorname{rang}(\mathbf{M}) = \operatorname{rang}(\mathbf{\Sigma})$, *also* $\operatorname{rang}(\mathbf{M}) \leq r \iff \sigma_{r+1} = \cdots = \sigma_{\min\{n,m\}} = 0$.

Wir definieren die Menge aller Matrizen mit maximalem Rang r als

$$\mathbb{R}_r^{n\times m} := \{\mathbf{M} \in \mathbb{R}^{n\times m} \,:\, \operatorname{rang}(\mathbf{M}) \leq r\}. \tag{3.4}$$

$\mathbb{R}_r^{n\times m}$ ist nicht abgeschlossen bezüglich der Addition, aber topologisch abgeschlossen.

Korollar 3.5. *Sei $\mathbf{M} \in \mathbb{R}^{n\times m}$ beliebig. Dann gilt:*

i) $\mathbf{M} \in \mathbb{R}_r^{n\times m} \iff \exists$ Matrizen $\mathbf{X} \in \mathbb{R}^{n\times r}$ und $\mathbf{Y} \in \mathbb{R}^{m\times r}$ dass $\mathbf{M} = \mathbf{X}\mathbf{Y}^T$.

ii) Die Menge $\mathbb{R}_r^{n\times m}$ ist abgeschlossen in $\mathbb{R}^{n\times m}$.

Beweis. i) Aus der Faktorisierung $\mathbf{M} = \mathbf{X}\mathbf{Y}^T$ folgt

$$\operatorname{rang}(\mathbf{M}) \leq \min\{\operatorname{rang}(\mathbf{X}), \operatorname{rang}(\mathbf{Y})\} \leq r.$$

Für die Umkehrung sei $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ die Singulärwertzerlegung und $r \leq \min\{n, m\}$. Wir schreiben $\mathbf{U} = (\mathbf{U}_0, \mathbf{U}_1)$ und $\mathbf{V} = (\mathbf{V}_0, \mathbf{V}_1)$ mit $\mathbf{U}_0 \in \mathbb{R}^{n\times r}$ und $\mathbf{V}_0 \in \mathbb{R}^{m\times r}$. Sei $\mathbf{S}_r := \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$. Da aus $\operatorname{rang}(\mathbf{M}) \leq r$ folgt dass $\sigma_{r+1} = \cdots = \sigma_{\min\{n,m\}} = 0$, gilt auch $\mathbf{M} = \mathbf{U}_0\mathbf{S}_r\mathbf{V}_0^T$. Da $\mathbf{U}_0\mathbf{S}_r \in \mathbb{R}^{n\times r}$ ist die gewünschte Zerlegung gezeigt.

ii) Sei $(\mathbf{M}_i)_{i=1}^{\infty} \subset \mathbb{R}_r^{n\times m}$ eine (in einer beliebigen Matrixnorm) konvergente Folge, mit $\mathbf{M}_i \to \mathbf{M} \in \mathbb{R}^{n\times m}$. Zu zeigen ist $\operatorname{rang}(\mathbf{M}) \leq r$. Die stetige Abhängigkeit der Singulärwerte liefert $\lim_i \sigma(\mathbf{M}_i) = \sigma(\mathbf{M})$. Da alle Singulärwerte $\sigma_j(\mathbf{M}_i) = 0$ von $\mathbf{M}_i$ mit $j > r$ verschwinden ($\operatorname{rang}(\mathbf{M}_i) \leq r \;\forall i \in \mathbb{N}$) folgt auch $\sigma_j(\mathbf{M}) = 0$ für $j > r$, also $\operatorname{rang}(\mathbf{M}) \leq r$. □

3.1.2 Bestapproximation mit Niedrigrangmatrizen

Korollar 3.5 zeigt, dass eine Niedrigrang-Faktorisierung einfach durch Null-setzen aller Singulärwerte mit Index größer als r erhalten werden kann (die Singulärwerte sind absteigend geordnet!).

Formal wird das mit dem Operator

$$\mathcal{T}_r : \mathbb{R}^{n\times m} \to \mathbb{R}_r^{n\times m} = \{\mathbf{M} \in \mathbb{R}^{n\times m} \,:\, \operatorname{rang}(\mathbf{M}) \leq r\} \tag{3.5}$$

realisiert, der Bestapproximationseigenschaften in Frobenius und ℓ_2-Operatornorm hat.

Satz 3.6. *Sei* $\mathbf{M} \in \mathbb{R}^{n\times m}$ *beliebig,* $r \leq \min\{n, m\}$*, und* $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ *die Singulärwertzerlegung. Wir schreiben* $\mathbf{U} \in \mathbb{R}^{n\times n}$ *und* $\mathbf{V} \in \mathbb{R}^{m\times m}$ *als* $\mathbf{U} = (\mathbf{U}_0, \mathbf{U}_1)$ *sowie* $\mathbf{V} = (\mathbf{V}_0, \mathbf{V}_1)$ *wobei* $\mathbf{U}_0 \in \mathbb{R}^{n\times r}$ *und* $\mathbf{V}_0 \in \mathbb{R}^{m\times r}$*. Mit den Singulärwerten* σ_i *von* $\mathbf{M}$ *definieren wir* $\mathbf{S}_r := \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$*. Dann ist die Matrix*

$$\mathcal{T}_r\mathbf{M} := \mathbf{U}_0\mathbf{S}_r\mathbf{V}_0^T \in \mathbb{R}_r^{n\times m} \tag{3.6}$$

die Lösung des Minimierungsproblems

$$\|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_F = \min_{\mathbf{M}_r\in\mathbb{R}_r^{n\times m}} \|\mathbf{M} - \mathbf{M}_r\|_F = \left(\sum_{j=r+1}^{\operatorname{rang}(\mathbf{M})} \sigma_j^2\right)^{1/2}. \tag{3.7}$$

Es gilt auch

$$\|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_2 = \min_{\mathbf{M}_r\in\mathbb{R}_r^{n\times m}} \|\mathbf{M} - \mathbf{M}_r\|_2 = \sigma_{r+1}. \tag{3.8}$$

Beweis. Falls $\operatorname{rang}(\mathbf{M}) \leq r$, sind die Bestapproximations-Fehler klarerweise gleich Null und es ist nichts zu zeigen.

Es sei also $\operatorname{rang}(\mathbf{M}) > r$. Die Invarianz der Normen $\|\cdot\|_2, \|\cdot\|_F$ unter orthogonalen Matrizen liefert

$$\begin{aligned}\|\mathbf{M}\|_2 &= \left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right\|_2 = \|\mathbf{\Sigma}\|_2 = \sigma_1\\ \|\mathbf{M}\|_F &= \left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right\|_F = \|\mathbf{\Sigma}\|_F = \left(\sum_{j=1}^{\min\{n,m\}} \sigma_j^2\right)^{1/2}\end{aligned} \tag{3.9}$$

also speziell

$$\begin{aligned}\|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_2 &= \|\mathbf{\Sigma} - \mathbf{S}_r\|_2 = \sigma_{r+1}\\ \|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_F &= \|\mathbf{\Sigma} - \mathbf{S}_r\|_F = \left(\sum_{j>r} \sigma_j^2\right)^{1/2}.\end{aligned}$$

Wir zeigen nun, dass $\mathcal{T}_r\mathbf{M}$ tatsächlich eine Bestapproximation ist. Sei $\mathbf{M}_r \in \mathbb{R}_r^{n\times m}$ beliebig. Mit $u_j \in \mathbb{R}^n$ und $v_j \in \mathbb{R}^m$ bezeichnen wir die Spalten von $\mathbf{U} \in \mathbb{R}^{n\times n}$ und $\mathbf{V} \in \mathbb{R}^{m\times m}$.

1.Schritt: Mittels Induktion zeigen wir die Existenz von orthonormalen Vektoren $z_{r+1}, \ldots, z_{\operatorname{rang}(\mathbf{M})} \in \mathbb{R}^m$ mit

$$z_j \in \ker(\mathbf{M}_r) \cap \operatorname{span}\{v_1, \ldots, v_j\} =: V_j \quad \text{für alle } j = r+1, \ldots, \operatorname{rang}(\mathbf{M}). \tag{3.10}$$

Der Induktionsanfang ist für $j = r+1$. Wir zeigen mittels eines Dimensionsarguments, dass $V_{r+1} \neq \{0\}$. Aus $\operatorname{rang}(\mathbf{M}_r) \leq r$ folgt $\dim\ker(\mathbf{M}_r) \geq m - r$. Somit,

$$\dim\ker(\mathbf{M}_r) + j \geq (m-r) + (r+1) = m+1 \quad \text{also} \quad \ker(\mathbf{M}_r) \cap \operatorname{span}\{v_1, \ldots, v_{r+1}\} \neq \emptyset.$$

Wir können also einen (normierten) Vektor $z_{r+1} \in V_{r+1}$ wählen.

Für den Induktionsschritt seien orthonormale Vektoren $z_{r+1}, \dots, z_j$ gegeben und wir müssen die Existenz eines z_{j+1} zeigen. Abermals gilt

$$\dim \ker(\mathbf{M}_r) + (j+1) \geq (m-r) + (j+1), \quad \text{also} \quad \dim V_{j+1} \geq j - r + 1.$$

Laut Voraussetzung haben wir bereits $j - r$ orthonormale Vektoren $z_i \in V_i \subseteq V_{j+1}$ für $i = r+1, \dots, j$ und aus $\dim V_{j+1} \geq j - r + 1$ folgt somit die Existenz eines $z_{j+1} \in V_{j+1}$ sodass $\{z_{r+1}, \dots, z_{j+1}\}$ ein Orthonormalsystem ist.

2.Schritt: Wir zeigen nun $\|\mathbf{M} - \mathbf{M}_r\|_2 \geq \sigma_{r+1}$. Da die Spalten v_j von $\mathbf{V}$ orthonormal sind, gilt

$$\mathbf{M} z_{r+1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T z_{r+1} = \sum_{j=1}^{\operatorname{rang}(\mathbf{M})} \sigma_j (v_j \cdot z_{r+1}) u_j = \sum_{j=1}^{r+1} \sigma_j (v_j \cdot z_{r+1}) u_j.$$

Da $z_{r+1} \in \ker(\mathbf{M}_r)$, $\|z_{r+1}\|_2 = 1$ und die Spalten u_j von $\mathbf{U}$ orthogonal sind, erhalten wir

$$\begin{aligned} \|\mathbf{M} - \mathbf{M}_r\|_2^2 \geq \|(\mathbf{M} - \mathbf{M}_r) z_{r+1}\|_2^2 = \|\mathbf{M} z_{r+1}\|_2^2 &= \sum_{j=1}^{r+1} \sigma_j^2 |v_j \cdot z_{r+1}|^2 \\ &\geq \sigma_{r+1}^2 \sum_{j=1}^{r+1} |v_j \cdot z_{r+1}|^2. \end{aligned}$$

Mit $\sum_{j=1}^{r+1} |v_j \cdot z_{r+1}|^2 = \|z_{r+1}\|_2^2 = 1$ folgt die Aussage für die Spektralnorm.

3.Schritt: Wir zeigen schlussendlich $\|\mathbf{M} - \mathbf{M}_r\|_F^2 \geq \sum_{j=r+1}^{\operatorname{rang}(\mathbf{M})} \sigma_j^2$. Das Orthonormalsystem $\{z_{r+1}, \dots, z_{\operatorname{rang}(\mathbf{M})}\} \subset \mathbb{R}^m$ konstruiert in Schritt 1 kann zu einer Orthogonalbasis $\{z_1, \dots, z_m\}$ von $\mathbb{R}^m$ erweitert werden. Sei $\mathbf{Z} \in \mathbb{R}^{m \times m}$ eine orthogonale Matrix mit Spalten z_j. Die Invarianz der Frobeniusnorm unter $\mathbf{Z}$ sowie Schritt 2 implizieren

$$\begin{aligned} \|\mathbf{M} - \mathbf{M}_r\|_F^2 = \|(\mathbf{M} - \mathbf{M}_r)\mathbf{Z}\|_F^2 = \sum_{j=1}^{m} \|(\mathbf{M} - \mathbf{M}_r) z_j\|_2^2 &\geq \sum_{j=r+1}^{\operatorname{rang}(\mathbf{M})} \|(\mathbf{M} - \mathbf{M}_r) z_j\|_2^2 \\ &\geq \sum_{j=r+1}^{\operatorname{rang}(\mathbf{M})} \sigma_j^2, \end{aligned}$$

was die gewünschte Aussage zeigt. □

Bemerkung 3.7. *Gilt $\sigma_r = \sigma_{r+1}$, so kann man die r-te und $r+1$-te Spalte jeweils in $\mathbf{U}$ und $\mathbf{V}$ vertauschen und erhält im Allgemeinen eine andere Matrix $\mathcal{T}_r\mathbf{M}$ mit dem selben Approximationsfehler.*

Aus (3.9) *folgt*

$$\|\mathbf{M}\|_F^2 = \sum_{j=1}^{\min\{n,m\}} \sigma_j^2 \leq \min\{n,m\} \sigma_1^2 = \min\{n,m\} \|\mathbf{M}\|_2^2 .$$

Falls $\min\{n, m\} \geq 5$ kann die Singulärwertzerlegung nicht exakt mit endlich vielen arithmetischen Operationen bestimmt werden. In Golub, van Loan [GVL96, p. 254, Kap. 5.4.5] wird gezeigt, dass der Aufwand (abhängig von der Maschinengenauigkeit) von gängigen Algorithmen zur Berechnung der SVD einer Matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ beschränkt ist durch

$$4n^2m + 8nm^2 + 9m^3 \quad \text{oder} \quad 4n^2m + 22m^3 \tag{3.11}$$

abhängig von dem gewählten Algorithmus. Speziell haben diese Algorithmen kubischen Aufwand und sind daher für große Matrizen unpraktikabel. Für Niedrigrang-Matrizen ist die Berechnung der Singulärwertzerlegung allerdings deutlich billiger, worauf wir bei der Analyse der Arithmetik für $\mathcal{H}$-Matrizen zurückkommen werden.

3.1.3 Bestapproximation mit $\mathcal{H}$-Matrizen

Mit Hilfe der Bestapproximationen mit Rang r aus dem vorigen Kapitel werden wir einen Operator

$$\mathcal{T}_{\mathcal{H}(r,\mathbb{P})} : \mathbb{R}^{n\times m} \to \mathcal{H}(r, \mathbb{P}) \tag{3.12}$$

definieren, der eine Matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ auf ihre Bestapproximation in $\mathcal{H}(r, \mathbb{P})$ bezüglich der Frobenius-Norm abbildet, also

$$\left\|\mathbf{A} - \mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}\right\|_F = \min_{\mathbf{A}_\mathcal{H}\in\mathcal{H}(r,\mathbb{P})} \|\mathbf{A} - \mathbf{A}_\mathcal{H}\|_F . \tag{3.13}$$

Sei $\mathcal{T}_r$ der Bestapproximationsoperator mit Rang r, siehe (3.6). Auf den Fernfeld-Blöcken definieren wir $\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}$ als

$$(\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A})|_{\tau\times\sigma} := \mathcal{T}_r(\mathbf{A}|_{\tau\times\sigma}) \quad \text{für } \tau \times \sigma \in \mathbb{P}_{\text{far}}, \tag{3.14}$$

auf den Nahfeld-Blöcken wird der exakte Matrixblock verwendet

$$(\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A})|_{\tau\times\sigma} := \mathbf{A}|_{\tau\times\sigma} \quad \text{für } \tau \times \sigma \in \mathbb{P}_{\text{near}}. \tag{3.15}$$

Die Bestapproximationseigenschaften von $\mathcal{T}_r$ sowie die Definition der Frobenius-Norm liefern direkt das folgende Korollar zu Theorem 3.6.

Korollar 3.8. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$, dann ist $\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}$ eine Bestapproximation an $\mathbf{A}$ in $\mathcal{H}(r, \mathbb{P})$ in der Frobenius-Norm, also*

$$\left\|\mathbf{A} - \mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}\right\|_F = \min_{\mathbf{A}_\mathcal{H}\in\mathcal{H}(r,\mathbb{P})} \|\mathbf{A} - \mathbf{A}_\mathcal{H}\|_F .$$

Beweis. Für die Frobeniusnorm gilt $\|\mathbf{A}\|_F^2 = \sum_{\tau\times\sigma\in\mathbb{P}} \|\mathbf{A}|_{\tau\times\sigma}\|_F^2$, es genügt also blockweise eine Bestapproximation zu finden. Da $\mathcal{T}_r(\mathbf{A}|_{\tau\times\sigma})$ auf jedem Fernfeldblock laut Theorem 3.6 eine Bestapproximation ist und für $\tau\times\sigma \in \mathbb{P}_{\text{near}}$ gilt dass $(\mathbf{A}-\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A})|_{\tau\times\sigma} = 0$, folgt somit die gewünschte Eigenschaft. □

Für die Spektralnorm kann man nicht direkt von den blockweisen Normen auf die globale Norm $\|\cdot\|_2$ schließen. Das folgende Lemma liefert allerdings eine obere und untere Schranke für die globale Norm durch die blockweisen Normen.

Der Einfachheit halber betrachten wir nur stufentreue Partitionen, also Partitionen mit $\tau \times \sigma \in \mathbb{P} \implies \operatorname{level}(\tau) = \operatorname{level}(\sigma)$. Ein allgemeineres Resultat findet sich im Buch von Börm [Bör10].

Lemma 3.9. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$, und $\mathbb{P}$ eine stufentreue Partition von $\mathcal{I} \times \mathcal{J}$. Dann gilt*

$$\max_{\tau\times\sigma\in\mathbb{P}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 \leq \|\mathbf{A}\|_2 \leq C_{\mathrm{sp}} \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \max_{\substack{\tau\times\sigma\in\mathbb{P}\\ \operatorname{level}(\tau)=\ell}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 \,. \tag{3.16}$$

Beweis. Für den Beweis der unteren Schranke müssen wir eine präzisere Notation von Einschränkungen definieren. Wir schreiben

$$\mathbb{R}^\tau := \{x \in \mathbb{R}^n : x_i = 0, i \notin \tau\}$$

sowie

$$\mathbb{R}^{\tau\times\sigma} := \{\mathbf{A} \in \mathbb{R}^{n\times m} : \mathbf{A}_{ij} = 0, (i,j) \notin \tau \times \sigma\}.$$

Mit $y|_\tau$ sei die Projektion eines Vektors $y \in \mathbb{R}^n$ auf $\mathbb{R}^\tau$ bezeichnet.

Sei $\tau \times \sigma \in \mathbb{P}$ beliebig. Dann gilt für $x \in \mathbb{R}^m$

$$\|\mathbf{A}|_{\tau\times\sigma} x\|_2 \leq \|\mathbf{A}|_{\mathcal{I}\times\sigma} x|_\sigma\|_2 = \|\mathbf{A}x|_\sigma\|_2 \leq \|\mathbf{A}\|_2 \, \|x|_\sigma\|_2 \leq \|\mathbf{A}\|_2 \, \|x\|_2 \,,$$

und da $\tau \times \sigma \in \mathbb{P}$ beliebig war, somit die untere Schranke.

Für den Beweis der oberen Schranke in (3.16) seien $x \in \mathbb{R}^m$ und $y := \mathbf{A}x \in \mathbb{R}^n$. Da $\mathbb{P}$ Partition von $\mathcal{I} \times \mathcal{J}$ ist, folgt

$$\|\mathbf{A}x\|_2^2 = y \cdot (\mathbf{A}x) = \sum_j y_j (\mathbf{A}x)_j = \sum_{j,k} y_j \mathbf{A}_{jk} x_k = \sum_{\tau\times\sigma\in\mathbb{P}} y|_\tau \cdot (\mathbf{A}|_{\tau\times\sigma} x|_\sigma).$$

Somit impliziert Cauchy-Schwarz, dass

$$\begin{aligned}\|\mathbf{A}x\|_2^2 &\leq \sum_{\tau\times\sigma\in\mathbb{P}} \|y|_\tau\|_2 \, \|\mathbf{A}|_{\tau\times\sigma}\|_2 \, \|x|_\sigma\|_2 \\ &\leq \Big(\sum_{\tau\times\sigma\in\mathbb{P}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 \, \|y|_\tau\|_2^2 \Big)^{1/2} \Big(\sum_{\tau\times\sigma\in\mathbb{P}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 \, \|x|_\sigma\|_2^2 \Big)^{1/2}.\end{aligned}$$

Mit $\operatorname{level}(\tau) = \operatorname{level}(\sigma)$ für alle $\tau \times \sigma \in \mathbb{P}$ gilt

$$\varepsilon_\ell := \max_{\substack{\tau\times\sigma\in\mathbb{P}\\ \operatorname{level}(\tau)=\ell}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 = \max_{\substack{\tau\times\sigma\in\mathbb{P}\\ \operatorname{level}(\sigma)=\ell}} \|\mathbf{A}|_{\tau\times\sigma}\|_2$$

und mit $\sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}}\\ \operatorname{level}(\tau)=\ell}} \|y|_\tau\|_2^2 \le \|y\|_2^2$ folgt weiters

$$\sum_{\tau\times\sigma\in\mathbb{P}} \|\mathbf{A}|_{\tau\times\sigma}\|_2 \, \|y|_\tau\|_2^2 \le \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}}\\ \operatorname{level}(\tau)=\ell}} \sum_{\substack{\sigma\in\mathbb{T}_{\mathcal{J}}\\ \tau\times\sigma\in\mathbb{P}}} \varepsilon_\ell \, \|y|_\tau\|_2^2 \le C_{\mathrm{sp}} \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \varepsilon_\ell \sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}}\\ \operatorname{level}(\tau)=\ell}} \|y|_\tau\|_2^2$$
$$\le C_{\mathrm{sp}} \, \|y\|_2^2 \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \varepsilon_\ell$$

und die analoge Abschätzung für x anstelle von y. Schlussendlich erhalten wir

$$\|\mathbf{A}x\|_2^2 \le \Big(C_{\mathrm{sp}} \, \|y\|_2^2 \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \varepsilon_\ell\Big)^{1/2} \Big(C_{\mathrm{sp}} \, \|x\|_2^2 \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \varepsilon_\ell\Big)^{1/2} = C_{\mathrm{sp}} \, \|x\|_2 \, \|y\|_2 \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P})} \varepsilon_\ell.$$

Mit $y = \mathbf{A}x$ und dem Supremum über $x \in \mathbb{R}^n$, folgt das gewünschte Resultat. □

Analog zu Korollar 3.8 folgt somit das folgende quasi-Bestapproximationsresultat.

Korollar 3.10. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$ dann gilt für $\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}$ dass*

$$\left\|\mathbf{A} - \mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}\right\|_2 \le C_{\mathrm{sp}}(\operatorname{depth}(\mathbb{P}) + 1) \min_{\mathbf{A}_{\mathcal{H}}\in\mathcal{H}(r,\mathbb{P})} \|\mathbf{A} - \mathbf{A}_{\mathcal{H}}\|_2 .$$

Beweis. Sei $\mathbf{A}_{\mathcal{H}} \in \mathcal{H}(r,\mathbb{P})$ beliebig. Auf jedem zulässigen Block $\tau \times \sigma \in \mathbb{P}_{\mathrm{far}}$ gilt, da $\mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A}|_{\tau\times\sigma}$ laut Theorem 3.6 Bestapproximation ist, mit der unteren Abschätzung von Lemma 3.9, dass

$$\left\|(\mathbf{A} - \mathcal{T}_{\mathcal{H}(r,\mathbb{P})}\mathbf{A})|_{\tau\times\sigma}\right\|_2 \le \|(\mathbf{A} - \mathbf{A}_{\mathcal{H}})|_{\tau\times\sigma}\|_2 \le \|\mathbf{A} - \mathbf{A}_{\mathcal{H}}\|_2 .$$

Auf nicht zulässigen Blöcken ist der Fehler gleich Null und Summation über alle Blöcke sowie die obere Abschätzung von Lemma 3.9 zeigen das Resultat. □

3.2 $\mathcal{H}$-Matrizen mittels Interpolation

Für den Integraloperator mit Kernfunktion $\log|x-y|$ aus dem einführenden Beispiel in Kapitel 1 konnte einfach mittels Taylor-Approximation eine geeignete Niedrigrang-Faktorisierung konstruiert werden. Wir werden in diesem Kapitel allgemeiner eine Klasse von (Kern-)Funktionen betrachten, so genannte asymptotisch glatte Funktionen, für welche $\mathcal{H}$-Matrix Approximationen mittels Interpolation konstruiert werden können.

3.2.1 Chebyshev-Interpolation in 1D

Tensorielle Chebyshev Interpolation auf Bounding-Boxen liefert eine einfache und zuverlässige Methode zur Konstruktion von Niedrigrangfaktorisierungen. Wir beginnen zunächst mit eindimensionaler Interpolation.

Sei $k \in \mathbb{N}$ ein fixierter Grad und $x_j \in [a,b]$, $j = 1,\dots,k$ verschiedene, gegebene Stützstellen. Die zugehörigen *Lagrange-Polynome* sind gegeben als

$$L_j(x) = \prod_{\substack{\ell=1 \\ \ell\neq j}}^{k} \frac{x - x_\ell}{x_j - x_\ell}. \tag{3.17}$$

Klarerweise gilt $L_j \in \mathcal{P}^{k-1}$ und $L_j(x_\ell) = \delta_{j\ell}$, wobei $\delta_{j\ell}$ das Kronecker-Delta bezeichnet.

Mit Hilfe der Lagrange-Polynome können wir nun einen Interpolationsoperator $I_k : C[a,b] \to \mathcal{P}^{k-1}$ definieren durch

$$I_k u = \sum_{j=1}^{k} u(x_j) L_j. \tag{3.18}$$

Falls $u \in C[a,b]$, so ist $I_k u$ das eindeutige Polynom in $\mathcal{P}^{k-1}$ mit der Eigenschaft $u(x_j) = I_k u(x_j)$.

Proposition 3.11. *Sei $I_k : C[a,b] \to \mathcal{P}^{k-1}$ der Interpolationsoperator definiert in* (3.18)*. Dann gilt:*

(i) I_k ist eine Projektion, also $I_k(I_k u) = I_k u$ für alle $u \in C[a,b]$.

(ii) I_k ist linear und beschränkt mit

$$1 \leq \|I_k\| := \sup_{u\in C[a,b]} \frac{\|I_k u\|_{\infty,[a,b]}}{\|u\|_{\infty,[a,b]}} \leq \sup_{x\in[a,b]} \sum_{j=1}^{k} |L_j(x)| < \infty. \tag{3.19}$$

(iii) I_k hat die Approximationseigenschaft

$$\|u - I_k u\|_{\infty,[a,b]} \leq \frac{\left\|u^{(k)}\right\|_{\infty,[a,b]}}{k!} \max_{x\in[a,b]} \prod_{j=1}^{k} |x - x_j| \quad \text{für alle } u \in C^k[a,b]. \tag{3.20}$$

Beweis. (i) Da $I_k u$ und $I_k(I_k u)$ an den Stützstellen x_j den selben Wert haben, folgt die Projektionseigenschaft aus der Eindeutigkeit des Interpolationspolynoms.

(ii) Die Linearität ist offensichtlich. Für die untere Schranke der Abschätzung setzt man $u = q \in \mathcal{P}^{k-1}$. Die obere Schranke folgt mittels Dreiecksungleichung.

(iii) Sei $y \in [a, b]$ fixiert. Wir betrachten die Funktion

$$f(x) := (u - I_k u)(x)\omega(y) - (u - I_k u)(y)\omega(x) \quad \text{mit } \omega(x) := \prod_{j=1}^{k}(x - x_j).$$

f hat $k+1$ Nullstellen in $[a, b]$. Mittels Induktion liefert der Satz von Rolle eine Nullstelle $\zeta \in (a, b)$ von $f^{(k)}$. Die Definition von f und $I_k u \in \mathcal{P}^{k-1}$ führen daher auf

$$0 = f^{(k)}(\zeta) = u^{(k)}(\zeta)\omega(y) - (u - I_k u)(y)k!$$

und somit

$$(u - I_k u)(y) = \frac{u^{(k)}(\zeta)}{k!}\omega(y).$$

Nimmt man das Supremum der Beträge über $\zeta = \zeta(y)$ sowie y, so folgt die gewünschte Aussage. □

Die Operatornorm $\|I_k\| =: \Lambda_k$ (oder auch *Lebesgue-Konstante*) ist unbeschränkt für $k \to \infty$ und hängt nur von der Wahl der Stützstellen ab. Tatsächlich existiert für jede Wahl von Stützstellen eine Konstante c, dass $\Lambda_k > \frac{2}{\pi}\log k - c$ (ERDÖS).

Eine gute Wahl der Stützstellen x_j, die zu dem langsamst möglichen Anstieg (logarithmisch) führt, sind die so genannten *Chebyshev Knoten* ζ_j, definiert auf dem Referenzintervall $[-1, 1]$ durch

$$\zeta_j = \cos\left(\frac{2j-1}{k}\,\frac{\pi}{2}\right) \quad \text{für } j = 1, \ldots, k.$$

Die Chebyshev Knoten sind die k Nullstellen der *Chebyshev Polynome*

$$C_k(\zeta) := \cos(k \arccos \zeta) \quad \text{für } \zeta \in [-1, 1]. \tag{3.21}$$

Mittels der bijektiven affinen Transformation

$$\psi(\zeta) = \frac{1}{2}\Big(a + b + \zeta(b - a)\Big)$$

können die Chebyshev Knoten einfach vom Referenzintervall auf ein beliebiges Intervall $[a, b]$ transformiert werden.

Seien schließlich $L_j \in \mathcal{P}^{k-1}$ die zu den transformierten Chebyshev-Knoten gehörigen Lagrange Polynome, dann nennen wir den induzierten Interpolationsoperator I_k den *Chebyshev-Interpolationsoperator*.

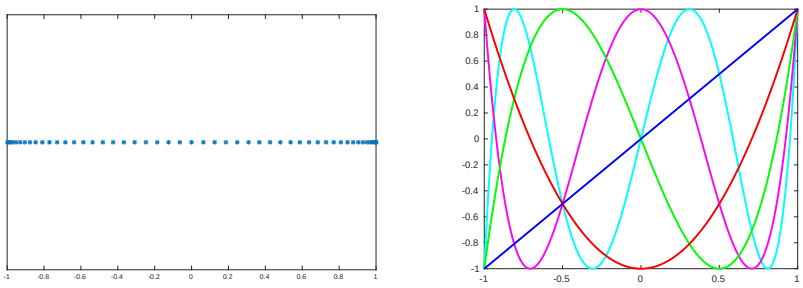


Abbildung 3.1: Chebyshev Knoten ($k = 50$) und die ersten 5 Chebyshev Polynome.

Satz 3.12. *Sei $u \in C[a,b]$ und I_k der Chebyshev-Interpolationsoperator. Dann gilt:*

(i) I_k erfüllt die Stabilitätsabschätzung

$$\|I_k u\|_{\infty,[a,b]} \leq \|I_k\| \, \|u\|_{\infty,[a,b]} \leq \left(\frac{2}{\pi}\log(k) + 1\right) \|u\|_{\infty,[a,b]} . \tag{3.22}$$

(ii) I_k besitzt die Approximationseigenschaft

$$\|u - I_k u\|_{\infty,[a,b]} \leq 4\,\frac{4^{-k}}{k!}\,(b-a)^k \left\|u^{(k)}\right\|_{\infty,[a,b]} . \tag{3.23}$$

Beweis. (i) Für die obere Schranke der Lebesgue-Konstante verweisen wir auf Rivlin [Riv74].
(ii) Wir betrachten zunächst das Referenzintervall $[-1,1]$.

Setzt man $\zeta = \cos\phi$ mit $\phi \in [0,\pi]$, dann gilt $C_k(\zeta) = \cos(k\phi)$ für das Chebyshev Polynom C_k. Das Additionstheorem für den Cosinus liefert

$$\cos(x) + \cos(y) = 2\cos\left(\frac{x+y}{2}\right)\cos\left(\frac{x-y}{2}\right) \quad \text{für alle } x, y \in \mathbb{R}.$$

Setzt man $x = (k+1)\phi$ und $y = (k-1)\phi$ ein, so erhält man die Rekursionsformel

$$2\zeta C_k(\zeta) - C_{k-1}(\zeta) = 2\cos(\phi)\cos(k\phi) - \cos((k-1)\phi) = \cos((k+1)\phi) = C_{k+1}(\zeta).$$

Mittels Induktion folgt daraus, dass C_k tatsächlich ein Polynom vom Grad k auf $[-1,1]$ ist und der führende Koeffizient ist 2^{k-1}. Somit gilt

$$1 = \|C_k\|_{\infty,[-1,1]} = 2^{k-1} \max_{\zeta\in[-1,1]} \prod_{j=1}^{k} |\zeta - \zeta_j|.$$

Setzt man das in (3.20) ein so folgt die Behauptung für das Referenzintervall.

Mit der affinen Transformation $\psi : [-1, 1] \to [a, b]$ folgt schlussendlich das Resultat für allgemeine Intervalle $[a, b]$, da $\|u \circ \psi\|_{\infty,[-1,1]} = \|u\|_{\infty,[a,b]}$ und

$$(u \circ \psi)^{(k)}(\zeta) = \left(\frac{b-a}{2}\right)^k u^{(k)}(\psi(\zeta)).$$

□

3.2.2 Asymptotisch glatte Kerne

Definition 3.13. *Seien $X, Y \subset \mathbb{R}^d$. Eine (Kern-)Funktion $\kappa : X \times Y \to \mathbb{R}$ heißt* asymptotisch glatt *falls $\kappa(x, y)$ für $x \neq y$ unendlich oft differenzierbar ist und Konstanten c_1, c_2 existieren sodass*

$$|\partial_x^\alpha \partial_y^\beta \kappa(x, y)| \leq c_1 \, (c_2|x - y|)^{-(|\alpha|+|\beta|+s)} (|\alpha + \beta|)! \tag{3.24}$$

für alle Multiindices $\alpha, \beta \in \mathbb{N}_0^d$ mit $|\alpha| + |\beta| \neq 0$. Der Parameter $s \in \mathbb{R}$ beschreibt die Ordnung der Singularität an $x = y$.

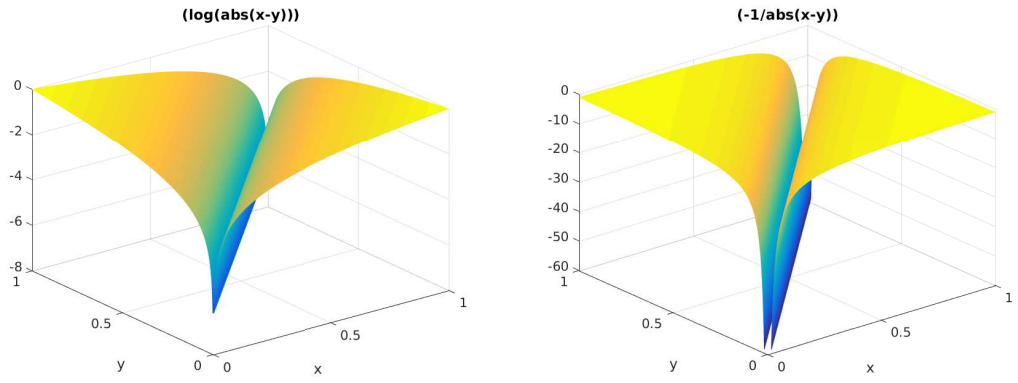


Abbildung 3.2: Die asymptotisch glatten Funktionen $\log|x - y|$ und $-\frac{1}{|x-y|}$.

Beispiel 3.14. *Man rechne nach, dass:*

- *Die Funktion $\kappa(x, y) = |x - y|^{-\nu}$ ist asymptotisch glatt mit $c_1 = c_2 = 1$ und $s = \nu$.*
- *Die Funktion $\kappa(x, y) = \log|x - y|$ ist asymptotisch glatt mit $c_1, c_2 = 1$ und $s = 0$.*
- *Produkte, Faltungen und Ableitungen von asymptotisch glatten Funktionen sind asymptotisch glatt.*

Das nachfolgende Resultat zeigt exponentielle Konvergenz der (eindimensionalen) Chebyshev Interpolation für asymptotisch glatte Funktionen κ auf zulässigen Bounding-Boxen.

Satz 3.15. *Sei κ asymptotisch glatt, $\eta > 0$ und $B_\tau, B_\sigma \subset \mathbb{R}$ Boxen mit* $\operatorname{diam}(B_\sigma) \leq \eta \operatorname{dist}(B_\tau, B_\sigma)$. *Sei $\mathcal{I}_k^\sigma$ der Chebyshev Interpolationsoperator in y-Richtung auf der Box B_σ. Dann gilt*

$$\|\kappa - \mathcal{I}_k^\sigma \kappa\|_{\infty, B_\tau \times B_\sigma} \leq 4c_1(\eta/c_2)^s \operatorname{diam}(B_\sigma)^{-s} \left(\frac{\eta}{4c_2}\right)^k. \tag{3.25}$$

Beweis. Satz 3.12 und Definition 3.13 implizieren

$$\begin{aligned}
\|\kappa - \mathcal{I}_k^\sigma \kappa\|_{\infty, B_\tau \times B_\sigma} &\leq 4 \frac{4^{-k}}{k!} \operatorname{diam}(B_\sigma)^k \left\|\partial_y^k \kappa\right\|_{\infty, B_\tau \times B_\sigma} \\
&\leq 4 \frac{4^{-k}}{k!} \operatorname{diam}(B_\sigma)^k \left[c_1 \Big(c_2 \operatorname{dist}(B_\tau, B_\sigma)\Big)^{-(k+s)} k!\right] \\
&\leq 4c_1(\eta/c_2)^s \operatorname{diam}(B_\sigma)^{-s} \Big(\frac{\eta}{4c_2}\Big)^k,
\end{aligned}$$

wobei im letzten Schritt $\frac{\operatorname{diam}(B_\sigma)}{\operatorname{dist}(B_\tau, B_\sigma)} \leq \eta$ verwendet wurde. □

Bemerkung 3.16. *Satz 3.15 liefert nur (exponentielle) Konvergenz für hinreichend kleine Zulässigkeitsparameter $\eta < 4c_2$.*

3.2.3 Verbesserte Fehlerabschätzung und tensorielle Interpolation

Wir werden in diesem Abschnitt eine verschärfte Fehlerabschätzung für die eindimensionale Chebyshev Interpolation vorstellen sowie die Resultate für $d > 1$ verallgemeinern.

Verbesserte Fehlerabschätzung*

Wir beginnen zunächst mit einem Approximationsresultat aus der komplexen Analysis. Hierfür benötigen wir den Begriff der Analytizität sowie eine spezielle Variante des Cauchy'schen Integralsatzes.

Sei $G \subset \mathbb{C}$ offen. Mit $B_r(w)$ bezeichnen wir die offene Kugel mit Radius r um den Punkt w. Eine Funktion $f : G \to \mathbb{C}$ heißt *analytisch* in G, wenn für jeden Punkt $w \in G$ ein $r > 0$ existiert mit $B_r(w) \subset G$ und

$$f(z) = \sum_{j=0}^{\infty} a_j (z - w)^j \quad \forall z \in B_r(w),$$

also wenn sich f um jeden Punkt in G in eine Potenzreihe entwickeln lässt.

Bemerkung 3.17. *Asymptotisch glatte Funktionen $\kappa : G \times G \to \mathbb{R}$ sind analytisch bezüglich x und y in $\{(x, y) \in G \times G, x \neq y\}$.*

Satz 3.18 (Cauchy'scher Integralsatz)**.** *Sei $B_{r,R}(0) := \{z \in \mathbb{C} : r < |z| < R\}$ ein offener Kreisring und $f : B_{r,R}(0) \to \mathbb{C}$ analytisch in $B_{r,R}(0)$. Dann gilt für alle Kreise $\partial B_\rho(0)$, $\partial B_{\rho'}(0)$ mit $r < \rho, \rho' < R$, dass*

$$\int_{\partial B_\rho(0)} f(\xi) d\xi = \int_{\partial B_{\rho'}(0)} f(\xi) d\xi.$$

Für $\rho > 0$ seien die so genannten *Bernstein-Ellipsen* definiert als

$$\mathcal{D}_\rho := \left\{ z \in \mathbb{C} \ : \ |z-1| + |z+1| < \rho + \frac{1}{\rho} \right\}, \qquad \mathcal{E}_\rho := \partial \mathcal{D}_\rho. \tag{3.26}$$

Klarerweise gilt $\mathcal{D}_\rho = \mathcal{D}_{\rho^{-1}}$ und $[-1, 1] \subseteq \mathcal{D}_\rho$ für alle $\rho > 1$. Eine elementare Rechnung zeigt, dass $|z| = \rho \implies \frac{1}{2}\left(z + \frac{1}{z}\right) \in \mathcal{E}_\rho$.

Die Transformation $J : z \mapsto \frac{1}{2}\left(z + \frac{1}{z}\right)$ heißt *Joukouwsky Transformation*.

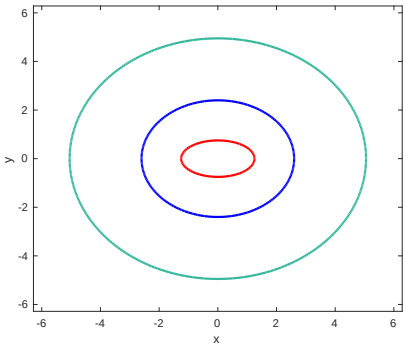


Abbildung 3.3: Bernstein-Ellipsen für $\rho = 2, 5, 10$.

Proposition 3.19. *Sei $u \in L^\infty(\mathcal{D}_\rho)$ analytisch in der Ellipse $\mathcal{D}_\rho$ mit $\rho > 1$. Dann gilt*

$$\min_{v \in \mathcal{P}_k} \|u - v\|_{\infty,[-1,1]} \leq 2\frac{\rho}{\rho - 1}\rho^{-(k+1)} \|u\|_{\infty,\mathcal{E}_\rho}$$

Beweis. Wir betrachten die Fourier-Reihe der 2π-periodischen Funktion $\widehat{u} := u \circ \cos : [-\pi, \pi] \to \mathbb{R}$, also

$$\widehat{u}(t) = \frac{a_0}{2} + \sum_{\ell=1}^{\infty} a_\ell \cos(\ell t) \quad a_\ell := \frac{1}{\pi}\int_{-\pi}^{\pi} \widehat{u}(t)\cos(\ell t) dt.$$

Da $\widehat{u}$ eine gerade Funktion ist, treten in der Entwicklung keine Sinus-Terme auf. Substituiert man $\cos(t) = x$, so erhält man eine Entwicklung von u in Chebyshev-Polynome

$$u(x) = \frac{a_0}{2} + \sum_{\ell=1}^{\infty} a_\ell C_\ell(x).$$

Da $|C_\ell(x)| \leq 1$ gilt

$$\left| u(x) - \frac{a_0}{2} - \sum_{\ell=1}^{k} a_\ell C_\ell(x) \right| \leq \sum_{\ell=k+1}^{\infty} |a_\ell| .$$

Da $\frac{a_0}{2} + \sum_{\ell=1}^{k} a_\ell C_\ell(x) \in \mathcal{P}_k$ benötigen wir also nur eine geeignete Abschätzung für $|a_\ell|$. Mit der Substitution $z = e^{it}$ erhält man

$$\begin{aligned} a_\ell &= \frac{1}{\pi} \int_{-\pi}^{\pi} u(\cos(t)) \cos(\ell t) dt = \frac{1}{\pi} \int_{-\pi}^{\pi} u(\mathrm{Re}(e^{it})) \,\mathrm{Re}(e^{i\ell t}) dt \\ &= \frac{1}{\pi} \int_{|z|=1} u(\mathrm{Re}(z)) \,\mathrm{Re}(z^\ell) \frac{1}{iz} dz. \end{aligned}$$

Auf der Einheitskreislinie gilt

$$\mathrm{Re}(z) = \frac{1}{2}(z + \overline{z}) = \frac{1}{2}\left(z + \frac{|z|^2}{z}\right) = \frac{1}{2}\left(z + \frac{1}{z}\right).$$

Somit gilt mit der Joukouwsky Transformation J, dass

$$\begin{aligned} a_\ell &= \frac{1}{2\pi i} \int_{|z|=1} u\left(Jz\right) \frac{z^\ell + z^{-\ell}}{z} dz \\ &= \frac{1}{2\pi i} \int_{|z|=1} u\left(Jz\right) z^{\ell-1} dz + \frac{1}{2\pi i} \int_{|z|=1} u\left(Jz\right) z^{-(\ell+1)} dz. \end{aligned}$$

Für $1 < \rho_1 < \rho$ verwenden wir den Cauchy'schen Integralsatz 3.18 für den Kreisring $B_{1/\rho,\rho}(0)$. Da die Joukouwsky Transformation jeden Kreis mit Radius ρ' auf die Ellipse $\mathcal{E}_{\rho'} \subset \mathcal{D}_\rho$ transformiert, ist $v(z) := u\left(\frac{1}{2}(z + \frac{1}{z})\right) = u \circ J(z)$ analytisch in $B_{1/\rho,\rho}(0)$. Somit sind beide Integranden der obigen Integrale analytisch in $B_{1/\rho,\rho}(0)$ und wir dürfen die Integrationswege im ersten Integral auf den Kreis $|z| = 1/\rho_1$ und im zweiten Integral auf den Kreis $|z| = \rho_1$ ändern. Es gilt also

$$\begin{aligned} a_\ell &= \frac{1}{2\pi i} \int_{|z|=\rho_1^{-1}} v(z) z^{\ell-1} dz + \frac{1}{2\pi i} \int_{|z|=\rho_1} v(z) z^{-(\ell+1)} dz \\ &\lesssim \frac{1}{2\pi} 2\pi \left\|u\right\|_{\infty,\mathcal{E}_{\rho_1}} \frac{1}{\rho_1} \rho_1^{-(\ell-1)} + \frac{1}{2\pi} 2\pi \left\|u\right\|_{\infty,\mathcal{E}_{\rho_1}} \rho_1 \rho_1^{-(\ell+1)} \\ &= 2\rho_1^{-\ell} \left\|u\right\|_{\infty,\mathcal{E}_{\rho_1}} . \end{aligned}$$

Somit folgt

$$\left| u(x) - \frac{a_0}{2} - \sum_{\ell=1}^{k} a_\ell C_\ell(x) \right| \leq \sum_{\ell=k+1}^{\infty} |a_k| \lesssim 2 \left\|u\right\|_{\infty,\mathcal{E}_{\rho_1}} \sum_{\ell=k+1}^{\infty} \rho_1^{-\ell} = 2\frac{\rho_1^{-k}}{\rho_1 - 1} \left\|u\right\|_{\infty,\mathcal{E}_{\rho_1}} .$$

Mit $\rho_1 \to \rho$ folgt die Aussage. □

Das folgende Resultat von Melenk [BLM05] liefert eine bessere Abschätzung für den Interpolationsfehler, womit exponentielle Konvergenz für beliebige Zulässigkeitsparameter η gezeigt werden kann.

Lemma 3.20. *Sei $[a,b] \subset \mathbb{R}$ und $u \in C^\infty[a,b]$ mit*

$$\left\|u^{(n)}\right\|_{\infty,[a,b]} \leq C_u \gamma_u^n n! \quad \forall n \in \mathbb{N}_0 \tag{3.27}$$

für Konstanten $C_u, \gamma_u \geq 0$. Dann gilt für alle $k \in \mathbb{N}_0$

$$\min_{v \in \mathcal{P}_k} \|u - v\|_{\infty,[a,b]} \leq C_u 4e(1 + \gamma_u(b-a))(k+1)\left(1 + \frac{2}{\gamma_u(b-a)}\right)^{-(k+1)}. \tag{3.28}$$

Beweis. Mit der affinen Transformation ψ kann man die Aussage abermals auf das Referenzintervall zurückführen. Hierbei gilt dann

$$\left\|(u \circ \psi)^{(n)}\right\|_{\infty,[-1,1]} \leq C_u \widehat{\gamma}_u^{-n} n!, \quad \widehat{\gamma}_u := \frac{2}{\gamma_u(b-a)}. \tag{3.29}$$

Aus dieser Abschätzung folgt, dass die Taylorreihe von $\widehat{u} := u \circ \psi$ in einem Punkt $\zeta \in (-1,1)$ auf einer Kugel $B_{\widehat{\gamma}_u}(\zeta) \subset \mathbb{C}$ um ζ mit Radius $\widehat{\gamma}_u$ konvergiert (z.B. Quotientenkriterium). Somit kann $\widehat{u}$ analytisch auf $\bigcup_{\zeta \in (-1,1)} B_{\widehat{\gamma}_u}(\zeta) =: G_{\widehat{\gamma}_u}$ fortgesetzt werden. Die Taylorentwicklung auf der Teilmenge $G_{\widehat{\gamma}_u/(1+\varepsilon)}$ um Punkte $\zeta \in (-1,1)$ für $\varepsilon > 0$ liefert

$$\|\widehat{u}\|_{\infty,G_{\widehat{\gamma}_u/(1+\varepsilon)}} \lesssim \left\|\sum_{\ell=0}^{\infty} \frac{\widehat{u}^{(\ell)}(\zeta)}{\ell!}(x-\zeta)^\ell\right\|_{\infty,G_{\widehat{\gamma}_u/(1+\varepsilon)}} \lesssim C_u \sum_{\ell=0}^{\infty}(1+\varepsilon)^{-\ell} = C_u \frac{1+\varepsilon}{\varepsilon}.$$

Eine elementare Rechnung zeigt, dass die Bernstein Ellipse $\mathcal{D}_{\rho_\varepsilon}$ mit $\rho_\varepsilon := 1 + \widehat{\gamma}_u/(1+\varepsilon)$ erfüllt, dass $\mathcal{D}_{\rho_\varepsilon} \subset G_{\widehat{\gamma}_u/(1+\varepsilon)}$. Insbesondere ist $\widehat{u}$ analytisch auf $\mathcal{D}_{\rho_\varepsilon}$. Proposition 3.19 impliziert nun

$$\min_{v \in \mathcal{P}_k} \|\widehat{u} - v\|_{\infty,[-1,1]} \leq 2\frac{\rho_\varepsilon}{\rho_\varepsilon - 1}\rho_\varepsilon^{-(k+1)} \|\widehat{u}\|_{\infty,\mathcal{E}_{\rho_\varepsilon}} \leq 2C_u \frac{\rho_\varepsilon}{\rho_\varepsilon - 1}\rho_\varepsilon^{-(k+1)} \frac{1+\varepsilon}{\varepsilon}.$$

Wählt man $\varepsilon = \frac{1}{k+1}$, dann folgt $1 + \varepsilon \leq 2$ sowie $\frac{\rho_\varepsilon}{\rho_\varepsilon - 1} \leq 1 + 2/\widehat{\gamma}_u$. Schlussendlich gilt

$$\begin{aligned}
\rho_\varepsilon^{-(k+1)} &= \left(1 + \frac{\widehat{\gamma}_u}{1+\varepsilon}\right)^{-(k+1)} = \frac{(1+\varepsilon)^{k+1}}{(1+\widehat{\gamma}_u)^{k+1}}\left(\frac{1+\widehat{\gamma}_u}{1+\widehat{\gamma}_u+\varepsilon}\right)^{k+1} \\
&\leq (1+\widehat{\gamma}_u)^{-(k+1)}\left(1 + \frac{1}{1+k}\right)^{k+1} \leq (1+\widehat{\gamma}_u)^{-(k+1)} e.
\end{aligned}$$

Setzt man dies ein, so folgt das gewünschte Resultat aus der Definition von $\widehat{\gamma}_u$. □

Korollar 3.21. *Unter den Voraussetzungen von Lemma 3.20 gilt für den Chebyshev Interpolationsoperator $I_k : C[a,b] \to \mathcal{P}^{k-1}$, dass*

$$\begin{aligned}\|u - I_k u\|_{\infty,[a,b]} &\leq (1+\Lambda_k) \min_{v\in\mathcal{P}_{k-1}} \|u-v\|_{\infty,[a,b]} \\ &\leq \Lambda_k C_u 8e(1+\gamma_u(b-a))k\left(1+\frac{2}{\gamma_u(b-a)}\right)^{-k}.\end{aligned}$$

Beweis. Sei $q \in \mathcal{P}_{k-1}$ beliebig. Dann gilt mit der Dreiecksungleichung, da I_k eine Projektion ist, und mit Satz 3.11, dass

$$\begin{aligned}\|u - I_k u\|_{\infty,[a,b]} &\lesssim \|u-q\|_{\infty,[a,b]} + \|q - I_k u\|_{\infty,[a,b]} = \|u-q\|_{\infty,[a,b]} + \|I_k(u-q)\|_{\infty,[a,b]} \\ &\lesssim \|u-q\|_{\infty,[a,b]} + \Lambda_k \|u-q\|_{\infty,[a,b]}.\end{aligned}$$

Die zweite Ungleichung folgt direkt aus Lemma 3.20 und $1 \leq \Lambda_k$. □

Tensorielle Chebyshev Interpolation auf Bounding Boxen

Sei $d \geq 1$. Wir betrachten die Bounding Box $B := \prod_{i=1}^d [a_i, b_i] \subset \mathbb{R}^d$. Ein weiterer Vorteil der Bounding Boxen besteht darin, dass sie Tensorprodukt-Struktur haben. Wir können daher einen Interpolationsoperator für $d > 1$ aus dem eindimensionalen Chebyshev Interpolationsoperator konstruieren.

Es seien für jedes $1 \leq j \leq k^d$ eindeutig ein zugehöriger Vektor $(j_1, \ldots, j_d) \in \{1, \ldots, k\}^d$ und Stützstellen

$$\mathbf{x}_j = (x_{j_1}^{(1)}, \ldots, x_{j_d}^{(d)})$$

gegeben, wobei $x_m^{(n)}$ den m-ten Chebyshev Knoten im Intervall $[a_n, b_n]$ bezeichnet. $L_{j_m}^{(m)} \in \mathcal{P}^{k-1}$ sei das Lagrange-Polynom zur Stützstelle $x_{j_m}^{(m)}$. Dann sind die tensoriellen Lagrange Polynome gegeben als

$$\mathbf{L}_j(y) = \prod_{m=1}^d L_{j_m}^{(m)}(y_m),$$

und es gilt $\mathbf{L}_j(\mathbf{x}_\ell) = \delta_{j\ell}$. Der **tensorielle Chebyshev Interpolationsoperator** ist dann für $u \in C(B)$ gegeben als

$$\boldsymbol{\mathcal{I}}_k u(y) = \sum_{j=1}^{k^d} u(\mathbf{x}_j)\mathbf{L}_j(y).$$

Wir definieren den Interpolationsoperator in der i-ten Variable $\mathcal{I}_k^{(i)} : C(B) \to \mathcal{P}^{k-1}$ durch

$$(\mathcal{I}_k^{(i)} u)(y) = \sum_{\ell=0}^{k} u(y_1, \ldots, y_{i-1}, x_\ell^{(i)}, y_{i+1}, \ldots, y_d) L_\ell^{(i)}(y_i).$$

Somit gilt für $\boldsymbol{\mathcal{I}}_k$ die Darstellung

$$\boldsymbol{\mathcal{I}}_k = \mathcal{I}_k^{(d)} \circ \cdots \circ \mathcal{I}_k^{(1)} =: \prod_{i=1}^{d} \mathcal{I}_k^{(i)}. \tag{3.30}$$

Mit Hilfe des eindimensionalen Stabilitäts- und Approximationsresultates aus Satz 3.12 folgt der nachfolgende Satz für $d > 1$.

Satz 3.22. *Sei $u \in C(B)$ und $\boldsymbol{\mathcal{I}}_k$ der tensorielle Chebyshev-Interpolationsoperator. Dann gilt:*

(i) $\boldsymbol{\mathcal{I}}_k$ ist eine Projektion, also $\boldsymbol{\mathcal{I}}_k^2 = \boldsymbol{\mathcal{I}}_k$.

(ii) $\boldsymbol{\mathcal{I}}_k$ ist linear und beschränkt mit

$$\|\boldsymbol{\mathcal{I}}_k u\|_{\infty,B} \leq \Lambda_k^d \, \|u\|_{\infty,B} \quad \text{für alle } u \in C(B). \tag{3.31}$$

(iii) Die Approximationseigenschaften der eindimensionalen Interpolationsoperatoren $\mathcal{I}_k^{(i)}$ übertragen sich auf $\boldsymbol{\mathcal{I}}_k$ mittels

$$\|u - \boldsymbol{\mathcal{I}}_k u\|_{\infty,B} \leq \sum_{\ell=1}^{d} \Lambda_k^{\ell-1} \left\| u - \mathcal{I}_k^{(\ell)} u \right\|_{\infty,[a_\ell,b_\ell]}. \tag{3.32}$$

Beweis. (i) Die Operatoren $\mathcal{I}_k^{(i)}$ kommutieren ($\mathcal{I}_k^{(i)}\mathcal{I}_k^{(\ell)} = \mathcal{I}_k^{(\ell)}\mathcal{I}_k^{(i)}$ für alle $i, \ell = 1, \ldots, d$).

Da die Operatoren $\mathcal{I}_k^{(i)}$ in einer Variable interpolieren und in den anderen die Identität sind, folgt aus Proposition 3.11 (i), dass $\mathcal{I}_k^{(i)}\mathcal{I}_k^{(i)} = \mathcal{I}_k^{(i)}$. Somit gilt aus der Kommutatoreigenschaft und der Kompositionsdarstellung auch $\boldsymbol{\mathcal{I}}_k\boldsymbol{\mathcal{I}}_k = \boldsymbol{\mathcal{I}}_k$.

(ii) Die Operatoren $\mathcal{I}_k^{(i)}$ erfüllen nach Proposition 3.11 (ii)

$$\left\|\mathcal{I}_k^{(i)} v\right\|_{\infty,B} \leq \Lambda_k \, \|v\|_{\infty,B} \quad \forall v \in C(B) \tag{3.33}$$

und somit folgt aus $\boldsymbol{\mathcal{I}}_k = \mathcal{I}_k^{(d)} \circ \cdots \circ \mathcal{I}_k^{(1)}$

$$\|\boldsymbol{\mathcal{I}}_k u\|_{\infty,B} \leq \prod_{i=1}^{d} \|\mathcal{I}_k^{(i)}\| \, \|u\|_{\infty,B} \leq \Lambda_k^d \, \|u\|_{\infty,B}.$$

(iii) Mit einer Teleskopsumme schreiben wir

$$u - \boldsymbol{\mathcal{I}}_k u = u - \Big(\prod_{i=1}^{d} \mathcal{I}_k^{(i)}\Big) u = \sum_{\ell=1}^{d} \Big(\prod_{i=1}^{\ell-1} \mathcal{I}_k^{(i)} u - \prod_{i=1}^{\ell} \mathcal{I}_k^{(i)} u\Big) = \sum_{\ell=1}^{d} \Big(\prod_{i=1}^{\ell-1} \mathcal{I}_k^{(i)}\Big)(u - \mathcal{I}_k^{(\ell)} u).$$

Mit (3.33) folgt schließlich

$$\|u - \boldsymbol{\mathcal{I}}_k u\|_{\infty} \leq \sum_{\ell=1}^{d} \left\| \Big(\prod_{i=1}^{\ell-1} \mathcal{I}_k^{(i)}\Big)(u - \mathcal{I}_k^{(\ell)} u) \right\|_{\infty} \leq \sum_{\ell=1}^{d} \Lambda_k^{\ell-1} \left\| u - \mathcal{I}_k^{(\ell)} u \right\|_{\infty},$$

was den Beweis beendet. □

Bemerkung 3.23. *Verwendet man für die Operatoren $\mathcal{I}_k^{(\ell)}$ die Approximations- und Stabilitätseigenschaften von Satz 3.12, so folgt ein analoges Resultat (mit zusätzlichem Faktor $d\Lambda_k^{d-1}$) zu Satz 3.15 auch für $d > 1$. Da Λ_k nur logarithmisch wächst, folgt abermals exponentielle Konvergenz falls $\eta < 4c_2$.*

Das nachfolgende Resultat zeigt schließlich exponentielle Konvergenz des Approximationsfehlers der tensoriellen Chebyshev-Interpolation für asymptotisch glatte Funktionen ohne Einschränkungen an den Zulässigkeitsparameter.

Korollar 3.24. *Sei κ asymptotisch glatt, $\eta > 0$ und $B_\tau, B_\sigma \subset \mathbb{R}^d$ Boxen mit $\operatorname{diam}(B_\sigma) \leq \eta \operatorname{dist}(B_\tau, B_\sigma)$. Sei $\boldsymbol{\mathcal{I}}_k^\sigma$ der tensorielle Chebyshev Interpolationsoperator in y-Richtung auf der Box B_σ. Dann gilt mit einer Konstante $C > 0$, dass*

$$\|\kappa - \boldsymbol{\mathcal{I}}_k^\sigma \kappa\|_{\infty, B_\tau \times B_\sigma} \leq C \operatorname{dist}(B_\tau, B_\sigma)^{-s} \Lambda_k^d \left(1 + \frac{\eta}{c_2}\right) k \left(1 + \frac{2c_2}{\eta}\right)^{-k}. \tag{3.34}$$

Beweis. Seien $x \in B_\tau, y \in B_\sigma$. Aus

$$|x - y| \geq \operatorname{dist}(x, B_\sigma) \geq \operatorname{dist}(B_\tau, B_\sigma)$$

und der Definition asymptotisch glatter Funktionen (Definition 3.13) folgt

$$\left\|\partial_{y_i}^n \kappa(x, \cdot)\right\|_{\infty, B_\sigma} \leq \sup_{y \in B_\sigma} c_1 \left(c_2 |x - y|\right)^{-(n+s)} n! \leq c_1 \left(c_2 \operatorname{dist}(B_\tau, B_\sigma)\right)^{-(n+s)} n!.$$

Somit ist für jede fixierte Koordinate y_i die Voraussetzung von Lemma 3.20 mit

$$C_u = c_1 (c_2 \operatorname{dist}(B_\tau, B_\sigma))^{-s} \quad \text{und} \quad \gamma_u = (c_2 \operatorname{dist}(B_\tau, B_\sigma))^{-1}$$

erfüllt. Mit der Darstellung $\boldsymbol{\mathcal{I}}_k^\sigma = \prod_{i=1}^d \mathcal{I}_k^{(i)}$ aus (3.30) gilt für jeden der eindimensionalen Chebyshev Interpolationsoperatoren $\mathcal{I}_k^{(i)}$ mit Korollar 3.21, dass

$$\left\|\kappa(x, \cdot) - \mathcal{I}_k^{(i)} \kappa(x, \cdot)\right\|_{\infty, B_\sigma} \leq C \Lambda_k C_u \left(1 + \frac{\operatorname{diam}(B_\sigma)}{c_2 \operatorname{dist}(B_\tau, B_\sigma)}\right) k \left(1 + \frac{2c_2 \operatorname{dist}(B_\tau, B_\sigma)}{\operatorname{diam}(B_\sigma)}\right)^{-k}.$$

Die Voraussetzung $\operatorname{diam}(B_\sigma) \leq \eta \operatorname{dist}(B_\tau, B_\sigma)$ impliziert schließlich

$$\left\|\kappa(x, \cdot) - \mathcal{I}_k^{(i)} \kappa(x, \cdot)\right\|_{\infty, B_\sigma} \leq C \operatorname{dist}(B_\tau, B_\sigma)^{-s} \Lambda_k \left(1 + \frac{\eta}{c_2}\right) k \left(1 + \frac{2c_2}{\eta}\right)^{-k}.$$

Satz 3.22 liefert nun

$$\begin{aligned} \|\kappa(x, \cdot) - \boldsymbol{\mathcal{I}}_k^\sigma \kappa(x, \cdot)\|_{\infty, B_\sigma} &\leq \sum_{\ell=1}^d \Lambda_k^{\ell-1} \left\|\kappa(x, \cdot) - \mathcal{I}_k^{(\ell)} \kappa(x, \cdot)\right\|_{\infty, B_\sigma} \\ &\leq C \operatorname{dist}(B_\tau, B_\sigma)^{-s} \Lambda_k^d \left(1 + \frac{\eta}{c_2}\right) k \left(1 + \frac{2c_2}{\eta}\right)^{-k}. \end{aligned}$$

Nimmt man das Supremum über $x \in B_\tau$, so folgt die gewünschte Aussage. □

Bemerkung 3.25. *Da stets* $\left(1 + \frac{2c_2}{\eta}\right) > 1$ *folgt exponentielle Konvergenz für beliebige Zulässigkeitsparameter* η*.*

Bemerkung 3.26. *Interpoliert man in x-Richtung anstatt in y-Richtung, so erhält man das selbe Resultat unter der Voraussetzung* $\operatorname{diam}(B_\tau) \leq \eta \operatorname{dist}(B_\tau, B_\sigma)$*.*

3.2.4 $\mathcal{H}$-Matrix Approximation

Wir wollen schlussendlich die Approximationen an asymptotisch glatte Kernfunktionen auf Matrizen $\mathbf{A}$ übertragen, die aus Diskretisierungen dieser Funktionen erhalten wurden, und somit $\mathcal{H}$-Matrix Approximationen erzeugen sowie deren Fehler kontrollieren.

Wir konstruieren $\mathcal{H}$-Matrix Approximationen für 3 verschiede Klassen von Matrizen $\mathbf{A} \in \mathbb{R}^{n \times m}$.

Annahme an die Matrix/Blockpartition. Sei $\mathbf{A}$ eine Matrix mit Einträgen gegeben durch einen der drei folgenden Fälle:

$$\mathbf{A}_{ij} = \kappa(x_i, y_j) \quad \text{für alle } i = 1, \ldots, n \text{ und } j = 1, \ldots, m \tag{A1}$$

mit gegebenen Evaluationspunkten $x_i, y_j \in \mathbb{R}^d$. In diesem Fall sind die Teilmengen X_i für die Konstruktion des Clusterbaumes $\mathbb{T}_\mathcal{I}$ gegeben als $X_i = \{x_i\}$ und für $\mathbb{T}_\mathcal{J}$ als $X_j = \{y_j\}$.

$$\mathbf{A}_{ij} = \int_\Omega \kappa(x_i, y)\psi_j(y)\, dy \quad \text{für alle } i = 1, \ldots, n \text{ und } j = 1, \ldots, m \tag{A2}$$

mit Auswertungspunkten $x_i \in \mathbb{R}^d$ und Funktionen $\psi_j \in L^1(\Omega)$, wobei $\Omega \subset \mathbb{R}^d$ ein beschränktes Gebiet ist. Hier sind die Teilmengen $X_i = \{x_i\}$ für $\mathbb{T}_\mathcal{I}$ und $X_j = \operatorname{supp} \psi_j$ für $\mathbb{T}_\mathcal{J}$.

$$\mathbf{A}_{ij} = \int_\Omega \int_\Omega \phi_i(x)\kappa(x, y)\psi_j(y)\, dy\, dx \quad \text{für alle } i = 1, \ldots, n \text{ und } j = 1, \ldots, m \tag{A3}$$

mit Funktionen $\phi_i \in L^1(\Omega)$ and $\psi_j \in L^1(\Omega)$. Hier sind die Teilmengen $X_i = \operatorname{supp} \phi_i$ für $\mathbb{T}_\mathcal{I}$ und $X_j = \operatorname{supp} \psi_j$ für $\mathbb{T}_\mathcal{J}$.

Die Block-Partition $\mathbb{P}$ sei basierend auf der Zulässigkeitsbedingung Adm mit $\operatorname{Adm}(\tau \times \sigma) = \text{true}$ falls

$$\min\{\operatorname{diam}(B_\tau), \operatorname{diam}(B_\sigma)\} \leq \eta \operatorname{dist}(B_\tau, B_\sigma) \tag{3.35}$$

für beliebiges, fixiertes $\eta > 0$ und Bounding Boxen $B_\tau \supset X_\tau, B_\sigma \supset X_\sigma$ gegeben.

Bemerkung 3.27. *Wir erinnern nochmals an die Notation*

$$\mathbb{R}^{\tau\times\sigma} := \{\mathbf{A}\in\mathbb{R}^{n\times m} \,:\, \mathbf{A}_{ij} = 0 \text{ für } (i,j)\notin\tau\times\sigma\}$$

aus dem Beweis von Lemma 3.9.

Oftmals wird notationell schlampig (so auch in Definition 2.10) die Einschränkung $\mathbf{A}|_{\tau\times\sigma}$ *als Matrix in* $\mathbb{R}^{|\tau|\times|\sigma|}$ *betrachtet, die Zeilen und Spalten zugehörig zu* τ *und* σ *beinhaltet. Versteht man unter der Einschränkung eigentlich die Projektion* $\Pi_{\mathbb{R}^{\tau\times\sigma}}\mathbf{A}$ *auf* $\mathbb{R}^{\tau\times\sigma}$ *durch Null-setzen der Einträge, die nicht zu* $\tau\times\sigma$ *gehören, so hängen diese beiden Interpretationen folgendermaßen formal zusammen:*

Für einen Cluster $\tau\subset\mathcal{I}$ *wählen wir eine Bijektion* $\pi_\tau : \tau\to\{1,\dots,|\tau|\}$*. Dann ist* $(\mathbf{A}|_{\tau\times\sigma})_{\pi_\tau(i),\pi_\sigma(j)} = (\Pi_{\mathbb{R}^{\tau\times\sigma}}\mathbf{A})_{ij}$*. In diesem Sinne wird oftmals für die Einschränkung* $\mathbf{A}|_{\tau\times\sigma} = \Pi_{\mathbb{R}^{\tau\times\sigma}}\mathbf{A}\in\mathbb{R}^{\tau\times\sigma}\simeq\mathbb{R}^{|\tau|\times|\sigma|}$ *geschrieben.*

Assemblieren der $\mathcal{H}$-Matrix. Wir werden die $\mathcal{H}$-Matrix Approximation $\mathbf{A}_\mathcal{H}$ blockweise aufstellen, wobei wir drei Fälle unterscheiden:

(1) Für $\tau\times\sigma\in\mathbb{P}_{\text{near}}$, definieren wir $\mathbf{A}_\mathcal{H}|_{\tau\times\sigma} := \mathbf{A}|_{\tau\times\sigma}$.

(2) Für $\tau\times\sigma\in\mathbb{P}_{\text{far}}$ mit $\operatorname{diam}(B_\tau)\le\operatorname{diam}(B_\sigma)$, approximieren wir $\mathbf{A}|_{\tau\times\sigma}$, indem wir die Kernfunktion $\kappa(x,y)$, die die Matrixeinträge festlegt ((A1)-(A3)), durch ihre (tensorielle) Interpolation in x-Richtung

$$\kappa_k^{\tau\sigma}(x,y) := \mathcal{I}_k^\tau\kappa(x,y) = \sum_{\ell=1}^{k^d}\kappa(x_\ell^\tau,y)\mathbf{L}_\ell^\tau(x)$$

ersetzen. Beispielsweise, sind die Einträge $\mathbf{A}_{ij}$ gegeben durch (A2), dann erhält man die Blockapproximation als

$$\mathbf{A}_{ij}\approx(\mathbf{A}_\mathcal{H})_{ij} := \int_\Omega\kappa_k^{\tau\sigma}(x_i,y)\psi_j(y)\,dy = \sum_{\ell=1}^{k^d}\mathbf{L}_\ell^\tau(x_i)\int_\Omega\kappa(x_\ell^\tau,y)\psi_j(y)\,dy.$$

Definiert man Matrizen $\mathbf{V}_{\tau\sigma}\in\mathbb{R}^{\tau\times k^d}$ and $\mathbf{W}_{\tau\sigma}\in\mathbb{R}^{\sigma\times k^d}$ (Rang $\le k^d$) durch

$$(\mathbf{V}_{\tau\sigma})_{i\ell} := \mathbf{L}_\ell^\tau(x_i)\;\; i\in\tau \quad\text{und}\quad (\mathbf{W}_{\tau\sigma})_{j\ell} := \int_\Omega\kappa(x_\ell^\tau,y)\psi_j(y)\,dy \quad j\in\sigma,$$

so erhalten wir

$$\mathbf{A}|_{\tau\times\sigma}\approx\mathbf{A}_\mathcal{H}|_{\tau\times\sigma} = \mathbf{V}_{\tau\sigma}\mathbf{W}_{\tau\sigma}^T.$$

(3) Für $\tau\times\sigma\in\mathbb{P}_{\text{far}}$ mit $\operatorname{diam}(B_\sigma)\le\operatorname{diam}(B_\tau)$, approximieren wir $\mathbf{A}|_{\tau\times\sigma}$, indem wir die Kernfunktion $\kappa(x,y)$, die die Matrixeinträge festlegt ((A1)-(A3)), durch ihre (tensorielle) Interpolation in y-Richtung

$$\kappa_k^{\tau\sigma}(x,y) := \mathcal{I}_k^\sigma\kappa(x,y) = \sum_{\ell=1}^{k^d}\kappa(x,y_\ell^\sigma)\mathbf{L}_\ell^\sigma(y)$$

ersetzen. Für den Fall (A2) führt das auf

$$\mathbf{A}_{ij} \approx (\mathbf{A}_k)_{ij} = \int_\Omega \kappa_k^{\tau\sigma}(x_i, y)\psi_j(y)\,dy = \sum_{\ell=1}^{k^d} \kappa(x_i, y_\ell^\sigma) \int_\Omega \mathbf{L}_\ell^\sigma(y)\psi_j(y)\,dy.$$

Definiert man Matrizen $\mathbf{V}_{\tau\sigma} \in \mathbb{R}^{\tau \times k^d}$ und $\mathbf{W}_{\tau\sigma} \in \mathbb{R}^{\sigma \times k^d}$ (Rang $\leq k^d$) durch

$$(\mathbf{V}_{\tau\sigma})_{i\ell} := \kappa(x_i, y_\ell^\sigma) \;\; i \in \tau \quad \text{und} \quad (\mathbf{W}_{\tau\sigma})_{j\ell} := \int_\Omega \mathbf{L}_\ell^\tau(y)\psi_j(y)\,dy \;\; j \in \sigma,$$

so erhält man abermals die Faktorisierung

$$\mathbf{A}|_{\tau\times\sigma} \approx \mathbf{A}_\mathcal{H}|_{\tau\times\sigma} = \mathbf{V}_{\tau\sigma}\mathbf{W}_{\tau\sigma}^T.$$

Wir haben also insgesamt für eine gegebene zulässige Partition eine $\mathcal{H}$-Matrix Approximation $\mathbf{A}_\mathcal{H} \in \mathcal{H}(k^d, \mathbb{P})$ konstruiert, die $\mathbf{A}$ approximiert. Hierbei kann man $\mathbf{A}_\mathcal{H}$ assemblieren **ohne** die gesamte Matrix $\mathbf{A}$ aufzustellen.

Mit Korollar 3.24 gilt

$$\|\kappa - \kappa_k^{\tau\sigma}\|_{\infty, B_\tau \times B_\sigma} \leq C \max_{\tau\times\sigma\in\mathbb{P}} (\operatorname{dist}(B_\tau, B_\sigma)^{-s})\Lambda_k^d \left(1 + \frac{\eta}{c_2}\right) k \left(1 + \frac{2c_2}{\eta}\right)^{-k} =: C_k. \quad (3.36)$$

Die folgende Proposition gibt schließlich eine a-priori Abschätzung für den Approximationsfehler der konstruierten $\mathcal{H}$-Matrix mit Interpolation in der Frobeniusnorm.

Proposition 3.28. *Sei $\mathbb{P}$ eine zulässige Partition und $\mathbf{A}_\mathcal{H} \in \mathcal{H}(k^d, \mathbb{P})$ die konstruierte $\mathcal{H}$-Matrix Approximation an $\mathbf{A} \in \mathbb{R}^{n\times m}$ mittels Interpolation. Sei C_k der Interpolationsfehler* (3.36) *von Korollar 3.24. Mit den Annahmen aus diesem Abschnitt folgt:*
(1) Seien die Einträge von $\mathbf{A}$ gegeben durch (A1)*, dann gilt*

$$\|\mathbf{A} - \mathbf{A}_\mathcal{H}\|_F \leq C_k\sqrt{n}\sqrt{m}. \quad (3.37)$$

(2) Seien die Einträge von $\mathbf{A}$ gegeben durch (A2)*, dann gilt*

$$\|\mathbf{A} - \mathbf{A}_\mathcal{H}\|_F \leq C_k\sqrt{n}\sqrt{m} \max_{j=1,\dots,m} \|\psi_j\|_{L^1(\Omega)} \quad (3.38)$$

(3) Seien die Einträge von $\mathbf{A}$ gegeben durch (A3)*, dann gilt*

$$\|\mathbf{A} - \mathbf{A}_\mathcal{H}\|_F \leq C_k\sqrt{n}\sqrt{m} \max_{i=1,\dots,n} \|\phi_i\|_{L^1(\Omega)} \max_{j=1,\dots,m} \|\psi_j\|_{L^1(\Omega)}. \quad (3.39)$$

Beweis. Da $\mathbb{P}$ Partition von $\mathcal{I} \times \mathcal{J}$ ist, existiert für alle Indices $(i, j) \in \mathcal{I} \times \mathcal{J}$ ein eindeutiger Block $\tau \times \sigma \in \mathbb{P}$ mit $(i, j) \in \tau \times \sigma$. Falls $\tau \times \sigma \in \mathbb{P}_{\text{near}}$, gilt $\mathbf{A}_{ij} = (\mathbf{A}_\mathcal{H})_{ij}$ und der elementweise Fehler ist gleich Null.

Für $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ unterscheiden wir die spezielle Gestalt der Matrixeinträge:
(1) Falls (A1) gilt, folgt aus $x_i \in X_\tau \subseteq B_\tau$ und $y_j \in X_\sigma \subseteq B_\sigma$, dass

$$\left|\mathbf{A}_{ij} - (\mathbf{A}_{\mathcal{H}})_{ij}\right| = \left|\kappa(x_i, y_j) - \kappa_k^{\tau\sigma}(x_i, y_j)\right| \leq \|\kappa - \kappa_k^{\tau\sigma}\|_{\infty, B_\tau \times B_\sigma} \leq C_k,$$

also gilt für die Frobenius-Norm

$$\|\mathbf{A} - \mathbf{A}_{\mathcal{H}}\|_F^2 = \sum_{i,j} \left|\mathbf{A}_{ij} - (\mathbf{A}_{\mathcal{H}})_{ij}\right|^2 \leq C_k^2 nm.$$

(2) Falls (A2) gilt, implizieren die Hölder Ungleichung sowie $\operatorname{supp}(\psi_j) \subseteq X_\sigma \subseteq B_\sigma$, dass

$$\begin{aligned}
\left|\mathbf{A}_{ij} - (\mathbf{A}_{\mathcal{H}})_{ij}\right| &= \left| \int_\Omega \Big(\kappa(x_i, y) - \kappa_k^{\tau\sigma}(x_i, y)\Big)\psi_j(y)\, dy \right| \\
&= \left| \int_{B_\sigma} \Big(\kappa(x_i, y) - \kappa_k^{\tau\sigma}(x_i, y)\Big)\psi_j(y)\, dy \right| \\
&\leq \|\kappa(x_i, \cdot) - \kappa_k^{\tau\sigma}(x_i, \cdot)\|_{\infty, B_\sigma} \|\psi_j\|_{L^1(B_\sigma)} \\
&\leq C_k \max_{j=1,\dots,m} \|\psi_j\|_{L^1(\Omega)}.
\end{aligned}$$

Summation über alle i, j liefert das gewünscht Resultat.
(3) Falls (A3) gilt, führt die gleiche Argumentation auf

$$\begin{aligned}
\left|\mathbf{A}_{ij} - (\mathbf{A}_{\mathcal{H}})_{ij}\right| &= \left| \int_{B_\tau} \int_{B_\sigma} \phi_i(x)\Big(\kappa(x, y) - \kappa_k^{\tau\sigma}(x, y)\Big)\psi_j(y)\, dy\, dx \right| \\
&\leq \|\kappa - \kappa_k^{\tau\sigma}\|_{\infty, B_\tau \times B_\sigma} \|\phi_i\|_{L^1(B_\tau)} \|\psi_j\|_{L^1(B_\sigma)} \\
&\leq C_k \max_{i=1,\dots,n} \|\phi_i\|_{L^1(\Omega)} \max_{j=1,\dots,m} \|\psi_j\|_{L^1(\Omega)},
\end{aligned}$$

womit die Proposition gezeigt ist. □

Bemerkung 3.29. *Tensorielle Interpolation erzeugt $\mathcal{H}$-Matrix Approximationen mit Rang k^d, da in jeder Koordinate ein Interpolationspolynom mit Grad $k-1$ erzeugt wird. Da oftmals für das Erreichen einer gewünschten Genauigkeit in manchen Koordinaten ein kleinerer Interpolationsgrad ausreicht, ist der Rang nicht optimal (beispielsweise wenn über eine $(d-1)$-dimensionale Mannigfaltigkeit integriert wird). Eine Möglichkeit den Rang nachträglich zu reduzieren wird in Kapitel 4 vorgestellt.*

Bemerkung 3.30. *Hat man Matrizen $\mathbf{A}$, die von Diskretisierungen (beispielsweise von Integraloperatoren) stammen, dann wächst die Dimension bei Verfeinerung der Zerlegung. Da der Interpolationsfehler C_k noch von $\max_{\tau\times\sigma\in\mathbb{P}}(\operatorname{dist}(B_\tau, B_\sigma)^{-s})$ abhängt, muss eventuell auch der Rang k erhöht werden, damit sich die Genauigkeit nicht verschlechtert. Typischerweise genügt hierbei $k = \mathcal{O}(\log(n+m))$.*

Bemerkung 3.31. *In vielen Anwendungen (siehe auch das Beispiel in Kapitel 1) erfüllen die Funktionen ϕ_i , dass $\|\phi_i\|_{L^1(\Omega)} = \mathcal{O}(1/n)$ und die Funktionen ψ_j, dass $\|\psi_j\|_{L^1(\Omega)} = \mathcal{O}(1/m)$, was zu besseren Fehlerabschätzungen in den Fällen* (A2) *und* (A3) *führt.*

3.3 Adaptive Kreuzapproximation (ACA)

Eine Methode effizient Niedrigrang-Faktorisierungen direkt aus den Matrixeinträgen zu konstruieren liefert die so genannte *adaptive Kreuzapproximation (ACA)*.

Die Idee hierbei ist (geschickt) Zeilen und Spalten ("Kreuze") der Matrix auszuwählen und so eine Niedrigrangfaktorisierung als Produkt dieser zu erhalten.

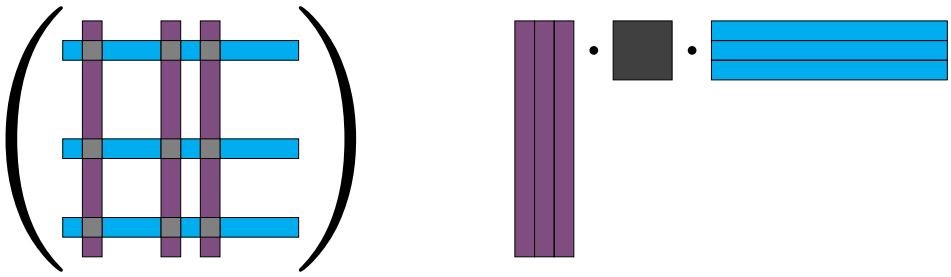


Abbildung 3.4: Kreuzapproximation

Man wählt also Zeilen $\tau^* \subset \tau$ und Spalten $\sigma^* \subset \sigma$ aus, dass $\mathbf{M} \in \mathbb{R}^{\tau\times\sigma}$ durch die Faktorisierung

$$\mathbf{M} \sim \mathbf{M}|_{\tau\times\sigma^*}\mathbf{S}\mathbf{M}|_{\tau^*\times\sigma}$$

mit einer geeigneten Matrix $\mathbf{S} \in \mathbb{R}^{\sigma^*\times\tau^*}$ ersetzt wird. Für das Produkt $\mathbf{M}|_{\tau\times\sigma^*}\mathbf{S}\mathbf{M}|_{\tau^*\times\sigma}$ gilt klarerweise

$$\operatorname{rang}(\mathbf{M}|_{\tau\times\sigma^*}\mathbf{S}\mathbf{M}|_{\tau^*\times\sigma}) \leq \min\{|\tau^*|, |\sigma^*|\}.$$

Wählt man also maximal r Zeilen in τ^* oder Spalten in σ^* aus, so erhält man eine Rang-r-Approximation.

Die Existenz von $\tau^*, \sigma^*, \mathbf{S}$, sodass die Kreuzapproximation $\mathbf{M}|_{\tau\times\sigma^*}\mathbf{S}\mathbf{M}|_{\tau^*\times\sigma}$ eine ansprechende Genauigkeit erzielt, ist im Allgemeinen nicht klar. Der folgende Satz liefert ein Kriterium hierfür.

Satz 3.32. *Sei $\mathbf{M} \in \mathbb{R}^{\tau\times\sigma}$ und $r \in \mathbb{N}$ mit $r < \min\{|\tau|, |\sigma|\}$. Für $\varepsilon > 0$ erfülle die Bestapproximation $\mathcal{T}_r\mathbf{M}$ mit Rang r, dass $\|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_2 \leq \varepsilon$. Dann existieren $\tau^* \subset \tau$, $\sigma^* \subset \sigma$ mit $|\tau^*| = |\sigma^*| = r$ sowie eine Matrix $\mathbf{S} \in \mathbb{R}^{\sigma^*\times\tau^*}$ sodass*

$$\|\mathbf{M} - \mathbf{M}|_{\tau\times\sigma^*}\mathbf{S}\mathbf{M}|_{\tau^*\times\sigma}\|_2 \leq \varepsilon\left(1 + 2\sqrt{r}(\sqrt{|\tau|} + \sqrt{|\sigma|})\right). \tag{3.40}$$

Beweis. Siehe Goreinov-Tyrtyshnikov-Zamarashkin [GTZ97]. □

Obwohl der Beweis in [GTZ97] gewissermaßen konstruktiv ist, liefert das Existenzresultat keine praktikable Antwort, wie die Zeilen- und Spaltenindices zu wählen sind. Weiters muss der Rang r, sodass $\|\mathbf{M} - \mathcal{T}_r\mathbf{M}\|_2 \leq \varepsilon$ a-priori bekannt sein.

Bemerkung 3.33. *Eine mögliche Wahl für die Indexmengen und die Matrix* $\mathbf{S}$*, die auf eine quasi-Bestapproximation führt, wäre* τ^*, σ^* *mit* $|\tau^*| = |\sigma^*| = r$ *so zu wählen, dass* $|\det \mathbf{M}|_{\tau^* \times \sigma^*}| = \max\{|\det \mathbf{M}|_{\tau' \times \sigma'}| \ : \ |\tau'| = |\sigma'| = r\}$ *und* $\mathbf{S} := \mathbf{M}|^{-1}_{\tau^* \times \sigma^*}$*. Allerdings ist die Lösung dieses Maximierungsproblems nicht in einem vertretbaren Aufwand bestimmbar.*

Der folgende Algorithmus liefert bei vorgegebenem r eine Kreuzapproximation $\mathbf{R} := \sum_{\ell=1}^{r} a^{(\ell)} b^{(\ell)^T}$ mit $\operatorname{rang}(\mathbf{R}) \leq r$.

Algorithmus 3.34 (Kreuzapproximation).

```
function CrossApproximation(M, var a, var b, r)
% a = (a^(1), ..., a^(r)) ∈ R^{n×r}, b = (b^(1), ..., b^(r)) ∈ R^{m×r}
for ℓ = 1 : r
   finde gutes Pivotelement (i*, j*)
   Setze δ := M_{i*j*}
   if δ = 0
      return
   else
      Setze a_i^(ℓ) := M_{ij*}          (j*-te Spalte von M)
      Setze b_j^(ℓ) := M_{i*j}/δ        (Skalierte i*-te Zeile von M)
      Subtrahiere Rang-1-Update M := M − a^(ℓ) b^(ℓ)^T.
   end
end
```

Die Wahl eines geeigneten Pivotelements $\mathbf{M}_{i^*j^*}$ ist entscheidend für Algorithmus 3.34. Eine Wahl wäre: finde (i^*, j^*) sodass

$$|\mathbf{M}_{i^*j^*}| = \max_{i,j} |\mathbf{M}_{ij}| \,.$$

Man spricht hierbei von Kreuzapproximation mit *voller Pivotsuche*.

Das nachfolgende Lemma zeigt, dass die Kreuzapproximation Rang-r-Matrizen in r-Schritten reproduziert.

Lemma 3.35. *Sei* $\mathbf{M} \in \mathbb{R}^{n \times m}$ *mit* $\operatorname{rang}(\mathbf{M}) \leq r$ *und die Vektoren* $a^{(\ell)}, b^{(\ell)}, \ell = 1, \ldots, r$ *berechnet von Algorithmus 3.34. Dann gilt*

$$\mathbf{R}_r := \sum_{\ell=1}^{r} a^{(\ell)} b^{(\ell)^T} = \mathbf{M}.$$

Beweis. Wir definieren $\mathbf{X}_{r'} := \mathbf{M} - \mathbf{R}_{r'}$ und zeigen mit Induktion, dass $\operatorname{rang}(\mathbf{X}_{r'}) = r - r'$ für alle $r' \leq r$.

Laut Voraussetzung gilt $\operatorname{rang}(\mathbf{M}) = r$, also der Induktionsanfang für $r' = 0$.

Für den Induktionsschritt gelte $\operatorname{rang}(\mathbf{X}_{r'}) = r - r'$. Dann ist $\mathbf{X}_{r'+1} = \mathbf{X}_{r'} - a^{(r'+1)} b^{(r'+1)^T}$. Da $a^{(r'+1)}$ laut Algorithmus 3.34 die j^*-te Spalte von $\mathbf{X}_{r'}$ ist, folgt $\operatorname{im}(\mathbf{X}_{r'+1}) \subset \operatorname{im}(\mathbf{X}_{r'})$ sowie $\dim \operatorname{im}(\mathbf{X}_{r'+1}) \geq \dim \operatorname{im}(\mathbf{X}_{r'}) - 1$. Es gilt

$$e_{i^*}^T \cdot \mathbf{X}_{r'} = b^{(r'+1)^T} \delta \neq 0,$$

also $e_{i^*} \notin \operatorname{im}(\mathbf{X}_{r'})^\perp$, aber

$$\begin{aligned} e_{i^*}^T (\mathbf{X}_{r'} - a^{(r'+1)} b^{(r'+1)^T}) &= b^{(r'+1)^T} \delta - e_{i^*}^T a^{(r'+1)} b^{(r'+1)^T} \\ &= b^{(r'+1)^T} \delta - \delta b^{(r'+1)^T} = 0, \end{aligned}$$

also $e_{i^*} \in \operatorname{im}(\mathbf{X}_{r'+1})^\perp$. Somit erhalten wir $\dim \operatorname{im}(\mathbf{X}_{r'+1})^\perp > \dim \operatorname{im}(\mathbf{X}_{r'})^\perp$, woraus $\dim \operatorname{im}(\mathbf{X}_{r'+1}) = \dim \operatorname{im}(\mathbf{X}_{r'}) - 1$ und mit der Induktionsvoraussetzung $\dim \operatorname{im}(\mathbf{X}_{r'+1}) = r - r' - 1$ folgt. Für $r' = r$ folgt dann das gewünschte Resultat. □

Unser Ziel ist zu zeigen, dass Algorithmus 3.34 tatsächlich eine Kreuzapproximation konstruiert, wofür wir nachfolgendes Lemma benötigen, das im gewissen Sinne eine Interpolationseigenschaft der Kreuzapproximation beschreibt.

Lemma 3.36. *Sei $\mathbf{M} \in \mathbb{R}^{n \times m}$ mit $r := \operatorname{rang}(\mathbf{M}) \geq 1$ und $\mathbf{R}_r := \sum_{\ell=1}^{r} a^{(\ell)} b^{(\ell)^T}$ von Algorithmus 3.34 berechnet. Dann gilt für alle Zeilen- und Spaltenpivots i^* und j^*, dass*

$$\mathbf{R}_r e_{j^*} = \mathbf{M}|_{\tau \times j^*}, \qquad e_{i^*}^T \mathbf{R}_r = \mathbf{M}|_{i^* \times \sigma}. \tag{3.41}$$

Beweis. Wir zeigen nur die erste Aussage für j^*, die Aussage für i^* folgt analog. Seien $i_{r'}^*, j_{r'}^*$ die Pivotelemente im r'-ten Schritt des Algorithmus. Dann gilt $a^{(r'+1)} = (\mathbf{M} - \mathbf{R}_{r'})|_{\tau \times j_{r'+1}^*}$, $b^{(r'+1)} = ((\mathbf{M} - \mathbf{R}_{r'})|_{i_{r'+1}^* \times \sigma})/\delta$ mit $\delta = (\mathbf{M} - \mathbf{R}_{r'})_{i_{r'+1}^* j_{r'+1}^*}$. Es gilt also

$$\begin{aligned} \mathbf{R}_{r'+1} e_{j_{r'+1}^*} &= \mathbf{R}_{r'} e_{j_{r'+1}^*} + a^{(r'+1)} b^{(r'+1)^T} e_{j_{r'+1}^*} = \mathbf{R}_{r'} e_{j_{r'+1}^*} + a^{(r'+1)} b_{j_{r'+1}^*}^{(r'+1)} \\ &= \mathbf{R}_{r'} e_{j_{r'+1}^*} + a^{(r'+1)} = \mathbf{R}_{r'} e_{j_{r'+1}^*} + (\mathbf{M} - \mathbf{R}_{r'})|_{\tau \times j_{r'+1}^*} = \mathbf{M}|_{\tau \times j_{r'+1}^*}. \end{aligned}$$

Mittels Induktion nach $r' = 0 \ldots, r$ (mit obiger Zeile mit $r = 0$ als Induktionsanfang) folgt für alle Pivotelemente i_k^*, j_k^* mit $k < r' + 1$, aus der Induktionsvoraussetzung, dass $(\mathbf{M} - \mathbf{R}_{r'+1})|_{\tau \times j_k^*} = 0$ und somit

$$\mathbf{R}_{r'+1} e_{j_k^*} = \mathbf{M}|_{\tau \times j_k^*} = \mathbf{R}_{r'} e_{j_k^*},$$

womit das Lemma gezeigt ist. □

Im Folgenden betrachten wir (notationell schlampig) die Einschränkungen $\mathbf{M}|_{\tau \times \sigma}$ als Matrix in $\mathbb{R}^{|\tau| \times |\sigma|}$.

Proposition 3.37. *Sei* $\mathbf{M} \in \mathbb{R}^{n\times m}$ *mit* $\operatorname{rang}(\mathbf{M}) \geq r$ *und* $\mathbf{R}_r := \sum_{\ell=1}^{r} a^{(\ell)} b^{(\ell)^T}$ *von Algorithmus 3.34 berechnet. Seien* $\tau^* \subset \tau$ *und* $\sigma^* \subset \sigma$ *die Mengen der Zeilen- und Spaltenpivots der ersten* r*-Schritte von Algorithmus 3.34. Dann ist* $\mathbf{M}|_{\tau^*\times\sigma^*}$ *invertierbar und es gilt*

$$\mathbf{R}_r = \mathbf{M}|_{\tau\times\sigma^*}\left(\mathbf{M}|_{\tau^*\times\sigma^*}\right)^{-1}\mathbf{M}|_{\tau^*\times\sigma}. \tag{3.42}$$

Beweis. Da $\operatorname{rang}(\mathbf{M}) \geq r$ gilt, terminiert Algorithmus 3.34 nicht in den ersten r-Schritten (ein geeignetes Pivotelement soll nicht 0 sein), somit folgt $\operatorname{rang}(\mathbf{M}|_{\tau^*\times\sigma^*}) = r$, also ist $\mathbf{M}|_{\tau^*\times\sigma^*} \in \mathbb{R}^{r\times r}$ invertierbar.

Mit dem ℓ-ten Spaltenpivotelement $j_\ell \in \sigma^*$ folgt schließlich

$$\mathbf{M}|_{\tau\times\sigma^*}\left(\mathbf{M}|_{\tau^*\times\sigma^*}\right)^{-1}\mathbf{M}|_{\tau^*\times\sigma}\, e_{j_\ell} = \mathbf{M}|_{\tau\times\sigma^*}\left(\mathbf{M}|_{\tau^*\times\sigma^*}\right)^{-1}\mathbf{M}|_{\tau^*\times\{j_\ell\}} = \mathbf{M}|_{\tau\times\sigma^*}\, e_{j_\ell} = \mathbf{M}|_{\tau\times\{j_\ell\}}.$$

Mit dem vorigen Lemma folgt auch $\mathbf{R}_r e_{j_\ell} = \mathbf{M}|_{\tau\times\{j_\ell\}}$, also stimmen die j_ℓ-ten Spalten überein. Da $\ell \leq r$ beliebig war, folgt das gewünschte Resultat. □

Kreuzapproximation mit voller Pivotsuche hat zwei große Nachteile. Einerseits müssen sämtliche Matrixeinträge $\mathbf{M}_{i,j}$ bereits berechnet sein und andererseits hat die Suche nach dem Pivotelement quadratischen Aufwand.

Eine praktikablere Alternative zur vollen Pivotsuche ist die <u>*partielle Pivotsuche*</u>:

- Wähle hierfür zunächst einen beliebigen Zeilenpivot i^*.
- Berechne in jedem Schritt für zuvor bestimmtes i^* ein j^*, sodass

$$j^* = \operatorname*{argmax}_{j\in\sigma} \left|(\mathbf{M} - \mathbf{R}_{\ell-1})_{i^*\times j}\right|.$$

- Mit j^* erhält man das Zeilenpivotelement i^* für den nächsten Durchlauf durch

$$i^* = \operatorname*{argmax}_{i\in\tau} \left|(\mathbf{M} - \mathbf{R}_{\ell-1})_{i\times j^*}\right|.$$

Man sucht also die neuen Pivotelemente in den Zeilen und Spalten zugehörig zu den vorherigen Pivotelementen. Die Bestimmung der Zeilen- und Spaltenpivots benötigt also nur bereits zuvor berechnete Einträge.

Bemerkung 3.38. *Ein Blick auf Algorithmus 3.34 zeigt, dass die Berechnung des Updates* $\mathbf{M} - a^{(\ell)} b^{(\ell)^T}$ *auch quadratischen Aufwand hat. Tatsächlich wird aber für die Berechnung von* $a^{(\ell)}, b^{(\ell)}$ *gar nicht die gesamte Matrix benötigt, sondern nur die Spalte* $(\mathbf{M} - \mathbf{R}_{\ell-1})_{\tau\times j*}$ *und Zeile* $(\mathbf{M} - \mathbf{R}_{\ell-1})_{i^*\times\sigma}$.

Eliminiert man also den Update-Schritt und verändert wie beschrieben die Berechnung von $a^{(\ell)}, b^{(\ell)}$*, so führt partielle Pivotsuche auf einen Algorithmus mit (annähernd) linearem Aufwand bei dem nicht sämtliche Matrixeinträge zuvor berechnet werden mussten.*

Oftmals ist nicht klar, wie der Rang r *a-priori* gewählt werden muss, damit eine geeignet genaue Approximation gefunden werden kann. Läuft man also die Schleife in Algorithmus 3.34 nicht r-mal durch, sondern bis ein gewisses Abbruchkriterium erreicht wird, so erhält ein Verfahren, das *adaptive Kreuzapproximation (ACA)* genannt wird.

Algorithmus 3.39 (ACA).

`function ACA(`$\mathbf{M}$`, var` a`, var` b`,` $r_{\max}$`,` ε`)`
`%` $a = (a^{(1)}, \dots, a^{(r)}) \in \mathbb{R}^{n\times r}, b = (b^{(1)}, \dots, b^{(r)}) \in \mathbb{R}^{m\times r}$
Wähle i^* beliebig mit $\exists j : \mathbf{M}_{i^*j} \neq 0$
`for` $\ell = 1 : r_{\max}$
  Setze $b_j^{(\ell)} := \mathbf{M}_{i^*j} - \sum_{k=1}^{\ell-1} a_{i^*}^{(k)} b_j^{(k)}$
  Bestimme $j^* := \operatorname{argmax}_j \left|b_j^{(\ell)}\right|$
  Setze $\delta := b_{j^*}^{(\ell)}$.
  Setze $a_i^{(\ell)} := \mathbf{M}_{ij^*} - \sum_{k=1}^{\ell-1} a_i^{(k)} b_{j^*}^{(k)}$
  `if` $\delta \neq 0$
    Setze $b^{(\ell)} := b^{(\ell)}/\delta$
  `if` $\left\|a^{(\ell)}\right\|_2 \left\|b^{(\ell)}\right\|_2 \leq \varepsilon \left\|a^{(1)}\right\|_2 \left\|b^{(1)}\right\|_2$
    `return`
  Bestimme $i^* := \operatorname{argmax}_i \left|a_i^{(\ell)}\right|$
`end`

Die (relative) Fehlerschranke $\left\|a^{(\ell)}\right\|_2 \left\|b^{(\ell)}\right\|_2 \leq \varepsilon \left\|a^{(1)}\right\|_2 \left\|b^{(1)}\right\|_2$ ist motiviert durch folgende Idee:

Nimmt man an, dass der Fehler fällt, also $\|\mathbf{M} - \mathbf{R}_\ell\|_2 \lesssim \|\mathbf{M} - \mathbf{R}_{\ell-1}\|_2$, und dass $\mathbf{M} - \mathbf{R}_{\ell-1} \simeq \mathbf{R}_\ell - \mathbf{R}_{\ell-1}$, sowie $\|\mathbf{M}\|_2 \simeq \|\mathbf{R}_1\|_2$, dann gilt

$$\begin{aligned}\|\mathbf{M} - \mathbf{R}_\ell\|_2 \lesssim \|\mathbf{M} - \mathbf{R}_{\ell-1}\|_2 \simeq \|\mathbf{R}_\ell - \mathbf{R}_{\ell-1}\|_2 = \left\|a^{(\ell)} b^{(\ell)^T}\right\|_2 \leq \left\|a^{(\ell)}\right\|_2 \left\|b^{(\ell)}\right\|_2 \\ \leq \varepsilon \left\|a^{(1)}\right\|_2 \left\|b^{(1)}\right\|_2 = \varepsilon \|\mathbf{R}_1\|_2 = \varepsilon \|\mathbf{M}\|_2 ,\end{aligned}$$

also der relative Fehler der Kreuzapproximation ist beschränkt durch eine gegebene Schranke ε.

Kann man in nicht-quadratischem Aufwand ein Start-Zeilenpivotelement i^* auswählen, dann ist die Komplexität von Algorithmus 3.39 von der Ordnung $\mathcal{O}(r^2(|\tau| + |\sigma|))$. Das nachfolgende Gegenbeispiel zeigt, dass dies im Allgemeinen nicht der Fall ist.

Beispiel 3.40. *Sei $\mathbf{M} \in \mathbb{R}^{n\times n}$ gegeben mit $\mathbf{M}_{ij} = 1$ für $i = i_0$ und $j = j_0$ und $\mathbf{M}_{ij} = 0$ sonst. Zu jedem Auswahlkriterium kann man i_0, j_0 wählen, dass die Nicht-Null-Spalte erst im n-ten Schritt gefunden wird. Die Gesamtkomplexität wäre dann abermals quadratisch.*

Das vorige Gegenbeispiel zeigt, dass ACA keine optimale Methode für schwachbesetzte Matrizen ist. Das Gegenbeispiel ist allerdings nicht typisch bei Diskretisierungen (von asymptotisch glatten Kernfunktionen, bzw Integraloperatoren).

Auch wenn ACA in der Anwendung (meistens) gute Resultate liefert, existiert auch für Diskretisierungen asymptotisch glatter Funktionen kein (allgemeines) theoretisches Konvergenzresultat. Ein Grund hierfür ist das folgende Gegenbeispiel.

Beispiel 3.41. *Sei* $\kappa : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, $(x, y) \mapsto (y_1 - x_1 - 1)(y_2 - x_2 - 1) \log \|x - y\|$, *dann ist* κ *asymptotisch glatt (Übung!) in* $\Omega := [0, 1]^2$ *und am Rand* $\Gamma := \partial\Omega$.

Sind die Matrixeinträge $\mathbf{M}_{ij}$ *gegeben durch Auswertungen*

$$\mathbf{M}_{ij} = \kappa(x_i, x_j)$$

mit gleichmäßig verteilten Punkten $x_i \in \Gamma$, $i = 1, \ldots, n$ *am Rand. Sei* $\eta > 0$ *und* $\delta := \frac{\eta}{2(1+\eta)}$ *und Cluster*

$$\begin{aligned} \tau_1 &:= \{i \in \mathcal{I} : (x_i)_1 \in [0, \delta], (x_i)_2 = 0\} & \tau_2 &:= \{i \in \mathcal{I} : , (x_i)_1 = 0, (x_i)_2 \in [0, \delta]\} \\ \sigma_1 &:= \{i \in \mathcal{I} : (x_i)_1 = 1, (x_i)_2 \in [1 - \delta, 1]\} & \sigma_2 &:= \{i \in \mathcal{I} : (x_i)_1 \in [1 - \delta, 1], (x_i)_2 = 1\} \end{aligned}$$

gegeben. Dann sind $\tau = \tau_1 \cup \tau_2$, $\sigma = \sigma_1 \cup \sigma_2$ η*-zulässig, da* $\operatorname{diam}(B_\tau) = \operatorname{diam}(B_\sigma) = \sqrt{2}\delta = \sqrt{2}\frac{\eta}{2(1+\eta)}$ *und* $\operatorname{dist}(B_\tau, B_\sigma) = \sqrt{2} - 2\sqrt{2}\delta = \sqrt{2}\frac{1}{1+\eta}$.

Für alle $i \in \tau_1$, $j \in \sigma_2$ *gilt* $\mathbf{M}_{ij} = 0$ *genau wie für* $i \in \tau_2$, $j \in \sigma_1$. *Somit hat die Matrix* $\mathbf{M}$ *Blockstruktur, wobei die beiden Diagonalblöcke entkoppelt sind, also*

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{\tau_1 \times \sigma_1} & 0 \\ 0 & \mathbf{M}_{\tau_2 \times \sigma_2} \end{pmatrix}$$

Hierbei versagt ACA auf Grund der partiellen Pivotsuche: wir wählen den ersten Pivotindex i^* *in* τ_1, *dann liefert das Kriterium* $j^* = \operatorname{argmax}_{j \in \mathcal{I}} |\mathbf{M}_{i^* \times j}|$, *dass* $j^* \in \sigma_1$. *Das nächste Zeilenpivotelement wird wegen* $i^* = \operatorname{argmax}_{i \in \mathcal{I}} \left|a^{(1)}\right|_{i \times j^*}$ *abermals in* $i^* \in \tau_1$ *gewählt, und somit wird auch das nächste Spaltenpivotelement in* $j^* \in \sigma_1$ *gewählt. Es sind also sämtliche Pivotelemente* $(i^*, j^*) \in \tau_1 \times \sigma_1$ *bis* $r > \min(|\tau_1|, |\sigma_1|)$. *Der Algorithmus sieht also den Block* $\mathbf{M}_{\tau_2 \times \sigma_2}$ *gar nicht und es gilt*

$$\|\mathbf{M} - \mathbf{R}_r\|_2 \geq \|\mathbf{M}_{\tau_2 \times \sigma_2}\|_2 .$$

Auch für den Fall von Diskretisierungen von Integraloperatoren existiert für den so genannten Doppelschicht-Operator ein ähnliches Gegenbeispiel, siehe z.B. [BGH06].

Es existieren aber auch positive Resultate zur Konvergenz von ACA, allerdings unter zusätzlichen Annahmen. Beispielsweise konvergiert ACA unter geeigneten Annahmen an die Auswertungspunkte x_j, siehe [Beb00].

Bebendorf erklärt in [Beb08], dass die Wahl des Zeilenpivots i^* entscheidend für die Konvergenz ist. Andere Wahlen der Pivotelemente führen beispielsweise auf den so genannten Algorithmus ACA+ ([BGH06]), für den obiges Gegenbeispiel nicht gilt, aber der im worst-case quadratischen Aufwand hat. In [Beb08] wird eine kompliziertere Wahl von i^* vorgestellt, für die unter zusätzlichen Stabilitätsannahmen Konvergenz für Matrizen der Form (A2)-(A3) folgt.

3.4 Hybride Kreuzapproximation

Eine Möglichkeit die Probleme von ACA zu umgehen, aber dennoch nicht den Rang unnötig groß zu wählen (wie bei Interpolation) ist die Kombination dieser beiden Methoden, die so genannte *hybride Kreuzapproximation (HCA)*. Im Gegensatz zu ACA funktioniert HCA nicht rein algebraisch, also nur durch Kenntnisse der Matrixeinträge, es muss zusätzlich die Funktion $\kappa(\cdot,\cdot)$, die die Matrixeinträge erzeugt, bekannt sein.

Die Grundidee der HCA ist die tensoriellen Chebyshev-Knoten auf einer Bounding-Box als Pivotelemente für die Kreuzapproximation zu verwenden.

Seien hierfür $\tau \subset \mathcal{I}$, $\sigma \subset \mathcal{J}$ Cluster und $B_\tau, B_\sigma \subset \mathbb{R}^d$ Bounding Boxen, $x_j^\tau \in B_\tau$, $j = 1,\dots,k^d$ die tensoriellen Chebyshev-Interpolationspunkte in B_τ und y_ℓ^σ, $\ell = 1,\dots,k^d$ die Punkte in B_σ und $\mathbf{L}_j^\tau(x)$, $\mathbf{L}_\ell^\sigma(y)$ die zugehörigen Lagrange Polynome.

Wir betrachten im Folgenden Matrizen $\mathbf{M}$, deren Einträge (A3) erfüllen, für Matrizen mit (A1) oder (A2) gelten analoge Überlegungen. Im Gegensatz zu Abschnitt 3.2.4 interpolieren wir in beiden Variablen, wir betrachten also

$$\widetilde{\kappa}_k^{\tau\sigma}(x,y) := \mathcal{I}_k^\tau \mathcal{I}_k^\sigma \kappa(x,y) = \sum_{j=1}^{k^d}\sum_{\ell=1}^{k^d} \mathbf{L}_j^\tau(x)\kappa(x_j^\tau, y_\ell^\sigma)\mathbf{L}_\ell^\sigma(y).$$

Die Matrix $\mathbf{S} \in \mathbb{R}^{k^d \times k^d}$ beinhaltet die Auswertungen

$$\mathbf{S}_{j\ell} := \kappa(x_j^\tau, y_\ell^\sigma).$$

Wir können also auf jedem zulässigem Block $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ die Approximation mittels Interpolation anschreiben als

$$\mathbf{M}|_{\tau\times\sigma} = \mathbf{V}^{\tau\sigma}\mathbf{S}(\mathbf{W}^{\tau\sigma})^T,$$

wobei $\mathbf{V}^{\tau\sigma} \in \mathbb{R}^{\tau\times k^d}$, $\mathbf{W}^{\tau\sigma} \in \mathbb{R}^{\sigma\times k^d}$ gegeben sind durch

$$(\mathbf{V}^{\tau\sigma})_{i\ell} := \int_\Omega \mathbf{L}_\ell^\tau(x)\phi_i(x)\,dx \quad \text{und} \quad (\mathbf{W}^{\tau\sigma})_{j\ell} := \int_\Omega \mathbf{L}_\ell^\sigma(y)\psi_j(y)\,dy.$$

Die Punkte x_j^τ, y_ℓ^σ sind im gesamten Volumen von $B_\tau \times B_\sigma$ verteilt (nicht nur am Rand wie bei Gegenbeispiel 3.41), daher kann die Matrix $\mathbf{S}$ mittels ACA approximiert werden (siehe [Beb00]). Da $\mathbf{S} \in \mathbb{R}^{k^d\times k^d}$ könnte für kleine k sogar eine volle Pivotsuche verwendet werden.

Seien $\tau^*, \sigma^* \subset \{1,\dots,k^d\}$ die Zeilen und Spaltenpivots, dann liefert ACA sowie Proposition 3.37

$$\mathbf{S} \approx \mathbf{S}_r = \mathbf{S}|_{k^d\times\sigma^*}\left(\mathbf{S}|_{\tau^*\times\sigma^*}\right)^{-1}\mathbf{S}|_{\tau^*\times k^d}.$$

Das Produkt $\mathbf{V}^{\tau\sigma}\mathbf{S}|_{k^d\times\sigma^*}$ erfüllt

$$(\mathbf{V}^{\tau\sigma}\mathbf{S}|_{k^d\times\sigma^*})_{ij} = \sum_{\ell=1}^{k^d} \mathbf{V}^{\tau\sigma}_{i\ell}\mathbf{S}_{\ell j} = \sum_{\ell=1}^{k^d} \int_\Omega \mathbf{L}^\tau_\ell(x)\phi_i(x)dx\ \kappa(x^\tau_\ell, y^\sigma_j) = \int_\Omega \phi_i(x)\mathcal{I}^\tau_k\kappa(x, y^\sigma_j)dx$$

und analog

$$(\mathbf{S}|_{\tau^*\times k^d}(\mathbf{W}^{\tau\sigma})^T)_{ij} = \int_\Omega \psi_j(y)\mathcal{I}^\sigma_k\kappa(x^\tau_i, y)dy.$$

Ersetzt man die Interpolationen durch die Kernfunktion κ, so erhält man

$$\mathbf{M}|_{\tau\times\sigma} \approx \mathbf{AGB}^T$$

mit

$$\mathbf{A}_{ij} := \int_\Omega \phi_i(x)\kappa(x, y^\sigma_j)\,dx, \quad \mathbf{B}_{ij} := \int_\Omega \psi_j(y)\kappa(x^\tau_i, y)\,dy \quad \text{und} \quad \mathbf{G} := \left(\mathbf{S}|_{\tau^*\times\sigma^*}\right)^{-1}.$$

Die Einträge der Approximation $\mathbf{AGB}^T$ erfüllen

$$\begin{aligned}(\mathbf{AGB}^T)_{ij} &= \int_\Omega\int_\Omega \phi_i(x)\widetilde{\kappa}(x, y)\psi_j(y)dxdy,\\ \widetilde{\kappa}(x, y) &:= \sum_{\nu\in\tau^*}\sum_{\mu\in\sigma^*} \kappa(x, y^\sigma_\mu)\mathbf{G}_{\mu\nu}\kappa(x^\tau_\nu, y).\end{aligned}$$

Die Funktion $\widetilde{\kappa}$ ist also genau eine Kreuzapproximation von κ und der Umweg über Interpolation ist nur für die korrekte Wahl der Pivots geschehen. Die Menge, in der die Pivots gesucht werden ist hierbei eingeschränkt, der Fehler hängt daher sowohl von der Genauigkeit der Interpolation als auch der Kreuzapproximation ab.

Proposition 3.42. *Seien $\tau \subset \mathcal{I}, \sigma \subset \mathcal{J}$ Cluster und B_τ, B_σ zugehörige Bounding-Boxen. Dann gilt*

$$\|\kappa - \widetilde{\kappa}\|_{\infty, B_\tau\times B_\sigma} \le \varepsilon_{\mathrm{Int}}(\kappa) + \varepsilon_{\mathrm{Int}}(\widetilde{\kappa}) + \Lambda_k^{2d}\varepsilon_{\mathrm{ACA}},$$

wobei $\varepsilon_{\mathrm{Int}}(g) := \|g - \mathcal{I}^\tau_k\mathcal{I}^\sigma_k g\|_\infty$ den Interpolationsfehler für eine Funktion g bezeichnet und $\varepsilon_{\mathrm{ACA}} := \max_{i,j}|\mathbf{S}_{ij} - (\mathbf{S}_r)_{ij}|$ den Fehler der Kreuzapproximation für die Matrix $\mathbf{S}$ bezeichnet.

Beweis. Mit Dreiecksungleichung folgt

$$\begin{aligned}\|\kappa - \widetilde{\kappa}\|_{\infty, B_\tau\times B_\sigma} &\lesssim \|\kappa - \mathcal{I}^\tau_k\mathcal{I}^\sigma_k\kappa\|_{\infty, B_\tau\times B_\sigma} + \|\widetilde{\kappa} - \mathcal{I}^\tau_k\mathcal{I}^\sigma_k\widetilde{\kappa}\|_{\infty, B_\tau\times B_\sigma} + \|\mathcal{I}^\tau_k\mathcal{I}^\sigma_k(\kappa - \widetilde{\kappa})\|_{\infty, B_\tau\times B_\sigma}\\ &\lesssim \varepsilon_{\mathrm{Int}}(\kappa) + \varepsilon_{\mathrm{Int}}(\widetilde{\kappa}) + \|\mathcal{I}^\tau_k\mathcal{I}^\sigma_k(\kappa - \widetilde{\kappa})\|_{\infty, B_\tau\times B_\sigma}.\end{aligned}$$

In den Interpolationspunkten x^τ_i, y^σ_j gilt $\widetilde{\kappa}(x^\tau_i, y^\sigma_j) = (\mathbf{S}_r)_{ij}$, also

$$\left|\kappa(x^\tau_i, y^\sigma_j) - \widetilde{\kappa}(x^\tau_i, y^\sigma_j)\right| = |\mathbf{S}_{ij} - (\mathbf{S}_r)_{ij}| \le \varepsilon_{ACA}.$$

Schlussendlich erfüllt die Lebesgue-Konstante (1D) $\Lambda_k = \sup_{x\in[a,b]} \sum_{j=1}^{k} |L_j(x)|$ und somit

$$\|\mathcal{I}_k^\tau \mathcal{I}_k^\sigma(\kappa - \widetilde{\kappa})\|_{\infty, B_\tau \times B_\sigma} \leq \Lambda_k^{2d} \sup_{i,j=1,\ldots,k} \left|\kappa(x_i^\tau, y_j^\sigma) - \widetilde{\kappa}(x_i^\tau, y_j^\sigma)\right| \leq \Lambda_k^{2d} \varepsilon_{ACA},$$

womit die Proposition gezeigt ist. □

Bemerkung 3.43. *Der Interpolationsfehler von κ wurde für asymptotisch glatte κ bereits betrachtet. Der Interpolationsfehler von $\widetilde{\kappa}$ kann mit ähnlichen Überlegungen behandelt werden ($\widetilde{\kappa}$ ist ebenfalls asymptotisch glatt), allerdings treten bei den Abschätzungen noch die Koeffizienten $\mathbf{G}_{\mu\nu}$ auf, weswegen der Interpolationsgrad (möglicherweise) hierbei größer gewählt werden muss. Numerische Beispiele zeigen allerdings, dass dies ein rein theoretisches Artefakt ist (man erwartet auch theoretisch, dass der Fehler nicht größer wird, wenn Interpolation durch exakten Kern ersetzt wird).*

Wir zeigen nachfolgend zwei Beispiele, die die Stärken und Schwächen von Interpolation, ACA und HCA zeigen.

Wir betrachten zunächst auf der Einheitskugel $B_1(0) \subset \mathbb{R}^d$ den Integraloperator

$$V\phi(x) = \int_{\partial\Omega} \frac{1}{4\pi\,|x-y|} \phi(y) ds_y, \quad x \in \partial\Omega.$$

Sei $\mathcal{T}_h = \{T_1, \ldots, T_n\}$ eine (approximative) Zerlegung der Kugel mit Dreiecken T_i und ϕ_i die charakteristische Funktion für das Dreieck T_i, also $\phi_i(x) = 1$ für $x \in T_i$ und $\phi_i(x) = 0$ sonst. Die Matrix $\mathbf{A}$ zugehörig zu V mit den Einträgen

$$\mathbf{A}_{ij} = \int_{\partial\Omega} V\phi_j(x)\phi_i(x) ds_x = \int_{T_i}\int_{T_j} \frac{1}{4\pi\,|x-y|} ds_y ds_x$$

ist eine Matrix der Form (A3) (die Funktion $\kappa(x,y) = \frac{1}{4\pi|x-y|}$ ist asymptotisch glatt).

Für $n = 20000$ zeigen die nachfolgenden Tabellen den Rang, relativen Fehler $\frac{\|\mathbf{A}-\mathbf{A}_\mathcal{H}\|_2}{\|\mathbf{A}\|_2}$ in der Spektralnorm sowie den Speicherbedarf (in MB) und die benötigte Zeit zum Assemblieren von $\mathbf{A}_\mathcal{H}$ für Interpolation (mit wachsendem Grad k), ACA (mit fallender Toleranz ε_{ACA}) und HCA (mit fallender Toleranz ε_{ACA} und steigendem Polynomgrad k). Die vollbesetzte Matrix $\mathbf{A}$ würde 3052 MB Speicher benötigen.

$\mathbf{A}_\mathcal{H}$ mit Interpolation

k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
2	8	0.0022	294.99	4.62
3	27	2.05e-04	905.55	9.97
4	64	3.08e-05	2094.53	21.49
5	125	3.63e-06	4054.75	42.82
6	216	4.05e-07	6979.02	78.63
7	343	8.66e-08	11060.13	137.89
8	512	1.65e-08	16490.91	224.02

$\mathbf{A}_\mathcal{H}$ mit ACA

ε_{ACA}	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
1e-01	3	0.0207	83.76	5.92
1e-02	5	0.0021	132.82	7.45
1e-03	9	3.73e-04	171.06	8.75
1e-04	12	2.70e-05	225.73	10.76
1e-05	15	2.50e-06	279.96	13.03
1e-06	20	2.24e-07	342.10	15.61
1e-07	24	3.02e-08	407.53	18.13

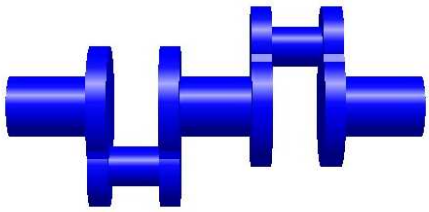
$\mathbf{A}_\mathcal{H}$ mit HCA

ε_{ACA}	k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
1e-01	2	6	0.0043	130.60	4.32
1e-02	3	13	8.77e-04	198.61	5.20
1e-03	4	21	1.06e-04	312.18	7.28
1e-04	5	36	3.22e-06	459.29	10.94
1e-05	6	52	6.77e-07	641.63	18.04
1e-06	7	70	4.43e-08	835.51	28.63
1e-07	8	94	3.44e-09	1041.39	42.66

Man erkennt für alle drei Verfahren Konvergenz, wobei ACA den geringsten Rang und somit auch den geringsten Speicherbedarf und die geringste Assemblierungszeit bei vergleichbarer Genauigkeit benötigt. HCA benötigt zwar etwas größere Blockränge, hat aber einen vergleichbaren Aufwand zu ACA.

Bei Interpolation hingegen erkennt man, dass der Rang wegen der Abhängigkeit von k^d sehr schnell wächst und für eine vergleichbare Genauigkeit zu ACA/HCA sehr große Ränge benötigt werden. Das spiegelt sich auch direkt im Speicherbedarf und den Rechenzeiten wieder.

Als zweites Beispiel sei die Geometrie Ω die Kurbelwelle:



Am Rand $\partial\Omega$ betrachten wir den Integraloperator

$$K\varphi(x) = \int_{\partial\Omega} \partial_{n_y} \frac{1}{4\pi\,|x-y|}\varphi(y)ds_y, \quad x \in \partial\Omega.$$

Mit einer Zerlegung $\mathcal{T}_h = \{T_1, \ldots, T_n\}$ des Randes $\partial\Gamma$ mit Dreiecken und charakteristischen Funktionen φ_i von T_i ist die zugehörige Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ gegeben durch

$$\mathbf{A}_{ij} = \int_{\partial\Omega} K\varphi_j(x)\varphi_i(x)ds_x = \int_{T_i}\int_{T_j} \partial_{n_y} \frac{1}{4\pi\,|x-y|} ds_y ds_x,$$

also abermals von der Gestalt (A3).

Für $n = 16352$ betrachten wir die selben Größen wie im vorherigen Beispiel. Die volle Matrix $\mathbf{A}$ benötigt hier 2040 MB Speicher.

$\mathbf{A}_{\mathcal{H}}$ mit Interpolation

k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
2	8	0.0893	272.82	4.79
3	27	0.0189	834.89	10.43
4	64	0.0038	1929.44	21.76
5	125	9.58e-04	3733.97	41.74
6	216	2.37e-04	6425.98	76.15
7	343	6.57e-05	10182.95	129.42
8	512	1.78e-05	15182.39	224.10

$\mathbf{A}_{\mathcal{H}}$ mit ACA

ε_{ACA}	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
1e-01	7	0.0236	113.83	7.11
1e-02	15	0.0040	175.40	9.84
1e-03	25	0.0027	249.13	13.38
1e-04	39	0.0027	332.05	16.28
1e-05	56	8.13e-04	418.92	20.41
1e-06	70	8.97e-04	510.34	25.27
1e-07	86	8.81e-04	601.27	31.15
1e-08	106	8.07e-04	690.32	36.96

$\mathbf{A}_{\mathcal{H}}$ mit HCA

ε_{ACA}	k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
1e-01	2	7	0.1085	163.75	4.19
1e-02	3	15	0.0230	268.79	6.26
1e-03	4	29	0.0139	405.20	8.80
1e-04	5	45	0.0069	572.73	13.75
1e-05	6	64	0.0029	769.86	22.74
1e-06	7	91	2.28e-04	974.44	37.05
1e-07	8	117	2.58e-05	1165.11	54.15

Für Interpolation und HCA erkennt man abermals Konvergenz für steigenden Polynomgrad bzw. fallendes ε_{ACA}. ACA hingegen konvergiert nicht wie gewünscht.

4 $\mathcal{H}$-Arithmetik

Eine der größten Stärken hierarchischer Matrizen ist, dass diese eine (approximative) Arithmetik mit Hilfe der Best-Approximationseigenschaft der Singulärwertzerlegung besitzen.

4.1 Singulärwertzerlegung von Niedrigrangmatrizen

In Abschnitt 3.1.1 haben wir gesehen, dass die SVD kubischen Aufwand besitzt. Für Matrizen mit Rang r, wobei $r < \min\{n, m\}$ kann die SVD allerdings mit deutlich geringerem Aufwand bestimmt werden.

Lemma 4.1. *Sei die Rang-r-Matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ in faktorisierter Form $\mathbf{M} = \mathbf{V}\mathbf{W}^T$ mit $\mathbf{V} \in \mathbb{R}^{n\times r}, \mathbf{W} \in \mathbb{R}^{m\times r}$ gegeben. Dann kann die Singulärwertzerlegung von $\mathbf{M}$ mit*

$$6r^2(n+m) + 23r^3 = \mathcal{O}\Big(r^2(n+m) + r^3\Big) \tag{4.1}$$

arithmetischen Operationen berechnet werden.

Beweis. Wir schreiben den Beweis in algorithmischer Form, sodass auch die Implementierung einer SVD für Niedrigrangmatrizen erklärt wird.

Wir beginnen mit der QR-Zerlegung einer Matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$, also der Faktorisierung $\mathbf{M} = \mathbf{Q}\mathbf{R}$ mit einer Matrix $\mathbf{Q} \in \mathbb{R}^{n\times m}$ mit orthogonalen Spalten und einer oberen Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{m\times m}$. Die Berechnung der QR-Zerlegung benötigt laut Golub, van Loan [p. 232, Kap. 5.2.9] $4nm^2$ Operationen.

Mit der Faktorisierung $\mathbf{M} = \mathbf{V}\mathbf{W}^T$ gehen wir wie folgt vor:

- Berechne die QR-Faktorisierung $\mathbf{V} = \mathbf{Q}_V\mathbf{R}_V$ in $4nr^2$ Operationen.
- Berechne die QR-Faktorisierung $\mathbf{W} = \mathbf{Q}_W\mathbf{R}_W$ in $4mr^2$ Operationen.
- Berechne die Matrix-Matrix Multiplikation $\mathbf{R} := \mathbf{R}_V\mathbf{R}_W^T$ in $< 2r^3$ Operationen.
- Berechne die SVD $\mathbf{R} = \mathbf{U}_0\mathbf{\Sigma}\mathbf{V}_0^T$ in $21r^3$ Operationen.
- Berechne orthogonale Matrix $\mathbf{U}_1 := \mathbf{Q}_V\mathbf{U}_0$ in $< 2nr^2$ Operationen.
- Berechne orthogonale Matrix $\mathbf{V}_1 := \mathbf{Q}_W\mathbf{V}_0$ in $< 2mr^2$ Operationen.

Gesamt benötigen wir also weniger als $6r^2(n+m)+23r^3$ arithmetische Operationen zur Berechnung der Faktorisierung

$$\begin{aligned}\mathbf{M} = \mathbf{V}\mathbf{W}^T &= \mathbf{Q}_V\mathbf{R}_V\mathbf{R}_W^T\mathbf{Q}_W^T = \mathbf{Q}_V(\mathbf{U}_0\mathbf{\Sigma}\mathbf{V}_0^T)\mathbf{Q}_W^T = (\mathbf{Q}_V\mathbf{U}_0)\mathbf{\Sigma}(\mathbf{Q}_W\mathbf{V}_0)^T \\ &= \mathbf{U}_1\mathbf{\Sigma}\mathbf{V}_1^T,\end{aligned}$$

womit das Lemma gezeigt ist. □

Bemerkung 4.2. *Die Spektralnorm $\|\mathbf{M}\|_2$ kann wegen $\|\mathbf{M}\|_2 = \sigma_1$ mit dem gleichen Aufwand berechnet werden.*

Für die Frobenius Norm gilt $\|\mathbf{M}\|_F = \left(\sum_{j=1}^r \sigma_j^2\right)^{1/2}$. Somit benötigt man $2r-1$ zusätzliche Operationen für das Berechnen der Summe und eine zusätzliche Operation für die Wurzel, also $2r$ zusätzliche Operationen.

Der Aufwand zur Berechnung der Bestapproximation $\mathcal{T}_r\mathbf{A}$ mittels SVD einer beliebigen Matrix $\mathbf{A}$ ist auf Grund des (zumindest quadratischen) Aufwands der SVD zu hoch. Zusätzlich muss man die gesamte Matrix $\mathbf{A}$ bereits assembliert haben, was im Allgemeinen ebenfalls quadratischen Aufwand hat.

Hat man allerdings eine $\mathcal{H}$-Matrix $\mathbf{A}_\mathcal{H} \in \mathcal{H}(r,\mathbb{P})$ gegeben, so ist die Berechnung der Bestapproximation mit $r' < r$ wegen Lemma 4.1 günstig.

Proposition 4.3. *Sei $\mathbb{P}$ zulässige Partition, $\mathbf{A} \in \mathcal{H}(r,\mathbb{P})$ und $r' < r$. Dann benötigt man*

$$23r^3|\mathbb{P}_{\text{far}}| + C_{\text{sp}}(\text{depth}(\mathbb{P})+1)(6(n+m)r^2 + nr') = \mathcal{O}(r^3N + r^2N\log N) \tag{4.2}$$

arithmetische Operationen zur Berechnung der Bestapproximation $\mathcal{T}_{\mathcal{H}(r')}\mathbf{A}$, wobei zulässige Blöcke in faktorisierter Form gespeichert werden und $N = n+m$.

Beweis. Für $\tau\times\sigma \in \mathbb{P}_{\text{near}}$ ist nichts zu tun.

Für $\tau\times\sigma \in \mathbb{P}_{\text{far}}$ verwenden wir Lemma 4.1 und bestimmen die reduzierte SVD $\mathbf{A}|_{\tau\times\sigma} = \mathbf{X}_{\tau\sigma}\mathbf{Y}_{\tau\sigma}^T = \mathbf{U}_r\mathbf{S}_r\mathbf{W}_r^T$ in $6r^2(|\tau|+|\sigma|)+23r^3$ Operationen. Nimmt man nur die ersten r' Spalten von $\mathbf{U}_{r'}, \mathbf{S}_{r'}, \mathbf{W}_{r'}$, dann erhält man die reduzierte faktorisierte Form mittels der Multiplikation $\mathbf{V}_{r'} := \mathbf{U}_{r'}\mathbf{S}_{r'} \in \mathbb{R}^{\tau\times r'}$ in $|\tau|\,r'$ Operationen. Für den Gesamtaufwand folgt also

$$\sum_{\tau\times\sigma\in\mathbb{P}_{\text{far}}} \left(6r^2(|\tau|+|\sigma|) + 23r^3 + |\tau|r'\right) \le 23r^3|\mathbb{P}_{\text{far}}| + 6r^2\sum_{\tau\times\sigma\in\mathbb{P}_{\text{far}}}(|\tau|+|\sigma|) + r'\sum_{\tau\times\sigma\in\mathbb{P}_{\text{far}}}|\tau|.$$

Bemerkung 2.12 liefert

$$\sum_{\tau\times\sigma\in\mathbb{P}_{\text{far}}}(|\tau|+|\sigma|) \le \sum_{\tau\times\sigma\in\mathbb{P}}(|\tau|+|\sigma|) \le C_{\text{sp}}(\text{depth}(\mathbb{P})+1)(n+m),$$

womit das gewünschte Resultat gezeigt ist. □

4.2 Matrix-Vektor-Multiplikation

Als erste arithmetische Operation werden wir die Matrix-Vektor-Multiplikation (MVM) betrachten. Bei vollbesetzten Matrizen $\mathbf{A} \in \mathbb{R}^{n\times m}$ besteht die MVM, also die Berechnung von $y_j = (\mathbf{A}x)_j = \sum_{\ell=1}^{m} \mathbf{A}_{j\ell}x_\ell$ aus m Multiplikationen und $m-1$-Addition. Gesamt erhält man also einen Aufwand von $n(2m-1)$, also quadratischen Aufwand.

Mit einer Partition $\mathbb{P}$ von $\mathcal{I}\times\mathcal{J}$ kann die MVM klarerweise blockweise durchgeführt werden, also

$$(\mathbf{A}x)_j = \sum_{\substack{\tau\times\sigma\in\mathbb{P}\\ j\in\tau}} \left(\mathbf{A}|_{\tau\times\sigma}x|_\sigma\right)_j$$

für $1 \le j \le n$. Der nachfolgende rekursive Algorithmus, aufgerufen durch

`HMVM(0,`$\mathcal{I}$`,`$\mathcal{J}$`,`$\mathbf{A}$`,`x`)`

berechnet $y = \mathbf{A}_{\mathcal{H}}x$, also die MVM einer $\mathcal{H}$-Matrix mit einem Vektor mit Hilfe dieser blockweisen Darstellung. Zusätzlich wird die Block-Partition im Algorithmus on-the-fly erzeugt.

Algorithmus 4.4 (MVM mit $\mathcal{H}$-Matrices).

```
function HMVM(var y,τ,σ,A,x)
if τ × σ zulässig
  y|_τ := y|_τ + X_τσ Y_τσ^T x|_σ
elseif sons(τ) ≠ ∅ und sons(σ) ≠ ∅
  for all τ' × σ' ∈ sons(τ × σ)
    rufe HMVM(y,τ',σ',A,x) auf
else /* nicht zulässiger Nahfeld-block */
  y|_τ := y|_τ + A|_τ×σ x|_σ
end
```

Die Multiplikation $\mathbf{X}_{\tau\sigma}\mathbf{Y}_{\tau\sigma}^T x|_\sigma$ wird hierbei in zwei Schritten berechnet, zuerst $z = \mathbf{Y}_{\tau\sigma}^T x|_\sigma$ und dann $\mathbf{X}_{\tau\sigma}z$. Der Aufwand der $\mathcal{H}$-MVM ist dann von der selben Größenordnung wie bei der Speicherung von $\mathcal{H}$-Matrizen.

Proposition 4.5. *Für die Anzahl arithmetischer Operationen N_{MVM} für die $\mathcal{H}$-Matrix MVM mit $\mathbf{A} \in \mathcal{H}(r,\mathbb{P})$ gilt*

$$N_{\mathrm{Storage}} \le N_{\mathrm{MVM}} \le 2N_{\mathrm{Storage}},$$

mit N_{Storage} von Proposition 2.11.

Beweis. Auf den Nahfeld-Blöcken $\tau\times\sigma \in \mathbb{P}_{\mathrm{near}}$ wird die MVM in Algorithmus 4.4 wie üblich durchgeführt und benötigt somit $|\tau|(2|\sigma|-1)$ Operationen. Das Update von $y|_\tau$

benötigt zusätzlich $|\tau|$ Additionen. Der Speicherbedarf eines Nahfeld-Blockes ist $|\tau||\sigma|$. Es gilt

$$|\tau||\sigma| \leq 2|\tau||\sigma| = |\tau|(2|\sigma| - 1) + |\tau|,$$

also die gewünschte Abschätzung für alle $\tau \times \sigma \in \mathbb{P}_{\text{near}}$.

Die hintereinander ausgeführten Multiplikationen $\mathbf{X}_{\tau\sigma}\mathbf{Y}_{\tau\sigma}^T x|_\sigma$ auf Fernfeld-Blöcken $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ benötigen $|\tau|(2r-1)+r(2|\sigma|-1)$ Operationen und das Update von $y|_\tau$ benötigt abermals zusätzlich $|\tau|$ Additionen. Der Speicherbedarf der Fernfeld-Blöcke ist $r(|\tau| + |\sigma|)$. Es gilt

$$r(|\tau| + |\sigma|) = |\tau|r + r|\sigma| \leq |\tau|(2r - 1) + r(2|\sigma| - 1) + |\tau| \leq 2r(|\tau| + |\sigma|),$$

also die gewünschte Aussage auf allen Fernfeld-Blöcken. Summation über alle Blöcke liefert die gewünschte Aussage. □

4.3 $\mathcal{H}$-Addition

Das einfache Beispiel

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

zeigt, dass die Menge aller Matrizen mit Rang r nicht abgeschlossen bezüglich der Addition ist.

Sei $\mathbb{P}$ eine zulässige Partition von $\mathcal{I} \times \mathcal{J}$ und $\mathbf{A}, \mathbf{B} \in \mathcal{H}(r) := \mathcal{H}(r, \mathbb{P})$, dann erfüllt die Matrix $\mathbf{A} + \mathbf{B}$ lediglich $\mathbf{A} + \mathbf{B} \in \mathcal{H}(2r)$. Für Matrizen mit Rang $2r$ kann aber nach Proposition 4.3 die Projektion auf $\mathcal{H}(r)$ billig berechnet werden. Somit können wir die *$\mathcal{H}$-Matrix Addition* (approximativ) definieren als

$$\oplus_r : \mathcal{H}(r) \times \mathcal{H}(r) \to \mathcal{H}(r), \quad \mathbf{A} \oplus_r \mathbf{B} := \mathcal{T}_{\mathcal{H}(r)}(\mathbf{A} + \mathbf{B}) \tag{4.3}$$

mit dem Bestapproximationsoperator $\mathcal{T}_{\mathcal{H}(r)}$ aus (3.12).

Der folgende rekursive Algorithmus, aufgerufen durch

`HAddition(var C,A,B,`$\mathcal{I}$`,`$\mathcal{J}$`,r),`

berechnet die $\mathcal{H}$-Addition $\mathbf{C} := \mathbf{A} \oplus_r \mathbf{B}$.

Algorithmus 4.6 ($\mathcal{H}$-Addition mit fixiertem lokalen Rang).

```
function HAddition(var C,A,B,τ,σ,r)
if τ × σ ∈ P_far
  C|_{τ×σ} := T_r(A|_{τ×σ} + B|_{τ×σ})
elseif sons(τ × σ) ≠ ∅
  for all τ' × σ' ∈ sons(τ × σ)
    rufe HAddition(C,A,B,τ',σ',r) auf
else /* nicht zulässiger Nahfeld-block */
  C|_{τ×σ} := A|_{τ×σ} + B|_{τ×σ}
end
```

Die Addition im Fernfeld wird hierbei nicht eintragsweise durchgeführt, sondern in der faktorisierten Form $\mathbf{A}|_{\tau\times\sigma} = \mathbf{V}_1\mathbf{W}_1^T$ und $\mathbf{B}|_{\tau\times\sigma} = \mathbf{V}_2\mathbf{W}_2^T$ mit Matrizen $\mathbf{V}_j \in \mathbb{R}^{\tau\times r}$ und $\mathbf{W}_j \in \mathbb{R}^{\sigma\times r}$: Wir definieren $\mathbf{V} := (\mathbf{V}_1, \mathbf{V}_2) \in \mathbb{R}^{\tau\times 2r}$ und $\mathbf{W} := (\mathbf{W}_1, \mathbf{W}_2) \in \mathbb{R}^{\sigma\times 2r}$. Dann gilt

$$\mathbf{A}|_{\tau\times\sigma} + \mathbf{B}|_{\tau\times\sigma} = \mathbf{V}\mathbf{W}^T,$$

also die *exakte Addition* von Fernfeld Blöcken ist prinzipiell eine Kopie von Speicher. $\mathbf{C}|_{\tau\times\sigma} = \mathcal{T}_r(\mathbf{V}\mathbf{W}^T)$ wird schlussendlich mit SVD berechnet.

Proposition 4.7. *Die $\mathcal{H}$-Addition zweier Matrizen $\mathbf{A}, \mathbf{B} \in \mathcal{H}(r)$ benötigt weniger als*

$$\begin{aligned} &184r^3|\mathbb{P}_{\text{far}}| + C_{\text{sp}}(\text{depth}(\mathbb{P}) + 1)\max\{24r^2 + r, n_{\text{blatt}}\}(n + m) \\ &\quad = \mathcal{O}(r^3N + r^2N\log N) \end{aligned} \tag{4.4}$$

arithmetische Operationen, wobei $N = n + m$.

Beweis. Auf Nahfeldblöcken $\tau \times \sigma \in \mathbb{P}_{\text{near}}$ ist die Addition eintragsweise und exakt in $|\tau||\sigma| \leq n_{\text{blatt}}(|\tau| + |\sigma|)$ Operationen durchgeführt. Für Fernfeldblöcke benötigen wir mit Proposition 4.3

$$6(2r)^2(|\tau| + |\sigma|) + 23(2r)^3 + |\tau|r \leq (24r^2 + r)(|\tau| + |\sigma|) + 184r^3$$

Operationen zum Berechnen von $\mathcal{T}_r(\mathbf{A}|_{\tau\times\sigma} + \mathbf{B}|_{\tau\times\sigma})$ in faktorisierter Form. Die üblichen Summationen über alle Blöcke zeigen das gewünschte Resultat. □

Bemerkung 4.8. *Die $\mathcal{H}$-Addition kann von der Genauigkeit beliebig schlecht sein, wie das folgende Beispiel zeigt:*

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \oplus_1 \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \mathcal{T}_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

*Ein Alternative hierzu wäre die so genannte **adaptive $\mathcal{H}$-Addition**, welche mit Hilfe der Rekompression in Abschnitt 4.6 vorgestellt wird.*

4.4 $\mathcal{H}$-Multiplikation

Wir werden in diesem Abschnitt das Produkt $\mathbf{A} \odot_r \mathbf{B}$ im $\mathcal{H}$-Matrix-Format definieren.

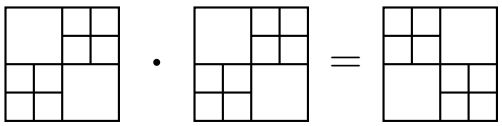
Seien hierfür Indexmengen $\mathcal{I}, \mathcal{J}, \mathcal{K}$ gegeben, sowie Cluster-Bäume $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}, \mathbb{T}_{\mathcal{J}\times\mathcal{K}}$ und $\mathcal{H}$-Matrizen $\mathbf{A} \in \mathcal{H}(r, \mathbb{P}_{\mathcal{I}\mathcal{J}})$, $\mathbf{B} \in \mathcal{H}(r, \mathbb{P}_{\mathcal{J}\mathcal{K}})$ mit zulässigen Partitionen $\mathbb{P}_{\mathcal{I}\mathcal{J}}$ basierend auf $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ und $\mathbb{P}_{\mathcal{J}\mathcal{K}}$ basierend auf $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$ (die Zulässigkeitsbedingungen können hierbei unterschiedlich sein).

Die $\mathcal{H}$-Matrix-Matrix-Multiplikation ist im Vergleich zur $\mathcal{H}$-Addition eine deutlich kompliziertere Operation. Ein Grund hierfür ist, dass bei einer blockweisen Bestimmung des Produkts

$$(\mathbf{A} \cdot \mathbf{B})|_{\tau\times\rho} = \sum_{j\in\mathcal{J}} \mathbf{A}|_{\tau\times\{j\}}\mathbf{B}|_{\{j\}\times\rho} \quad \forall \tau \subseteq \mathcal{I}, \rho \subseteq \mathcal{K} \tag{4.5}$$

die innere Summe die Blockpartition beeinflusst. Die nachfolgenden Beispiel zeigen, dass hierbei sowohl Vergröberungen, als auch Verfeinerungen der Blockstruktur auftreten können:

Tatsächlich kann sich auch die Blockstruktur bei der Multiplikation komplett verändern:



4.4.1 Produktbäume und Komplexität der exakten Multiplikation

Wir müssen also im Folgenden in effizienter Weise die Blockstruktur des Produkts beschreiben.

Definition 4.9. *Seien $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ und $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$ stufentreue Produkt-Cluster-Bäume, dann definieren wir den* induzierten Produktbaum *$\mathbb{T}_{\mathcal{I}\times\mathcal{J}} \cdot \mathbb{T}_{\mathcal{J}\times\mathcal{K}} =: \mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ induktiv als Baum mit Wurzel $\mathcal{I} \times \mathcal{K}$ und*

$$\begin{aligned}\operatorname{sons}(\tau \times \rho) := \{\tau' \times \rho' \,:\, &\tau' \in \mathbb{T}_{\mathcal{I}}, \rho' \in \mathbb{T}_{\mathcal{K}},\\ &\exists\, \sigma, \sigma' \in \mathbb{T}_{\mathcal{J}} :\ \tau' \times \sigma' \in \operatorname{sons}(\tau \times \sigma), \sigma' \times \rho' \in \operatorname{sons}(\sigma \times \rho)\}.\end{aligned}$$

Die Blätter des induzierten Produktbaumes $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ bilden die induzierte Partition *$\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}} := \operatorname{leaves}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}})$.*

Lemma 4.10. *Seien $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ und $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$ stufentreue Produkt-Cluster-Bäume und $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ der induzierte Produktbaum. Dann gilt*

$$\operatorname{depth}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}) \leq \min\{\operatorname{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}), \operatorname{depth}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}})\}. \tag{4.6}$$

Für die Schwachbesetztheitskonstante $C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}})$ von $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ folgt

$$C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}) \leq C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}}). \tag{4.7}$$

Beweis. Die erste Abschätzung folgt direkt aus Definition 4.9.

Sei $\tau \in \mathbb{T}_{\mathcal{I}}$. Dann gilt

$$\begin{aligned}\{\rho \in \mathbb{T}_{\mathcal{K}} : \tau \times \rho \in \mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}\} &\subseteq \{\rho \in \mathbb{T}_{\mathcal{K}} : \exists \sigma \in \mathbb{T}_{\mathcal{J}},\ \tau \times \sigma \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}},\ \sigma \times \rho \in \mathbb{T}_{\mathcal{J}\times\mathcal{K}}\} \\ &= \bigcup_{\substack{\sigma \in \mathbb{T}_{\mathcal{J}} \\ \tau\times\sigma \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}}}} \{\rho \in \mathbb{T}_{\mathcal{K}} : \sigma \times \rho \in \mathbb{T}_{\mathcal{J}\times\mathcal{K}}\}.\end{aligned}$$

Somit folgt

$$\begin{aligned}\left|\{\rho \in \mathbb{T}_{\mathcal{K}} : \tau \times \rho \in \mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}\}\right| &\leq \sum_{\substack{\sigma \in \mathbb{T}_{\mathcal{J}} \\ \tau\times\sigma \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}}}} \left|\{\rho \in \mathbb{T}_{\mathcal{K}} : \sigma \times \rho \in \mathbb{T}_{\mathcal{J}\times\mathcal{K}}\}\right| \\ &\leq C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}) C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}}),\end{aligned}$$

also die zeilenweise Abschätzung. Die spaltenweise Abschätzung folgt analog. □

Wir werden zunächst die exakte Multiplikation, also die Summe (4.5), effizient umschreiben, sodass ein Faktor immer eine Niedrigrangmatrix ist.

Wir schreiben $\mathcal{F}^{(\ell)}(\tau) := \hat{\tau}$ für den eindeutigen Vorfahren $\hat{\tau} \in \mathbb{T}_{\mathcal{I}}$ eines Blattes $\tau \in \mathbb{T}_{\mathcal{I}}$ mit $\operatorname{level}(\hat{\tau}) = \ell$.

Für ein Blatt $\tau \times \rho \in \mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ definieren wir die Menge

$$\mathcal{U}_\ell(\tau \times \rho) := \{\sigma \in \mathbb{T}_{\mathcal{J}} : \mathcal{F}^{(\ell)}(\tau) \times \sigma \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}}, \sigma \times \mathcal{F}^{(\ell)}(\rho) \in \mathbb{T}_{\mathcal{J}\times\mathcal{K}},\ \text{zumindest einer ist Blatt}\}.$$

Lemma 4.11. *Sei $\tau \times \rho \in \operatorname{leaves}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}})$ ein Blatt des Produktbaumes. Dann sind die Mengen $\mathcal{U}_\ell(\tau \times \rho)$, $\ell = 0, \ldots, \operatorname{level}\tau \times \rho$ eine disjunkte Partition von $\mathcal{J}$, also es gilt*

$$\bigcup_{\ell=0}^{\operatorname{level}(\tau\times\rho)} \mathcal{U}_\ell(\tau \times \rho) = \mathcal{J}. \tag{4.8}$$

Für die Mächtigkeit der Mengen $\mathcal{U}_\ell(\tau \times \rho)$ gilt

$$|\mathcal{U}_\ell(\tau \times \rho)| \leq \min\{C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}), C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}})\}. \tag{4.9}$$

Beweis. **1.Schritt:** Wir zeigen, dass die Mengen $\mathcal{U}_\ell(\tau \times \rho)$ disjunkt sind: Seien $\ell_1 \leq \ell_2$ und $\sigma_1 \in \mathcal{U}_{\ell_1}(\tau \times \rho)$, $\sigma_2 \in \mathcal{U}_{\ell_2}(\tau \times \rho)$ mit $\sigma_1 \cap \sigma_2 \neq \emptyset$. Aus $\operatorname{level}(\sigma_1) = \ell_1 \leq \ell_2 = \operatorname{level}(\sigma_2)$ sowie $\sigma_1, \sigma_2 \in \mathbb{T}_{\mathcal{J}}$ folgt da $\mathbb{T}_{\mathcal{J}}$ Cluster-Baum ist, dass $\sigma_2 \subseteq \sigma_1$. Es gilt somit

$$\mathcal{F}^{(\ell_2)}(\tau) \times \sigma_2 \subseteq \mathcal{F}^{(\ell_1)}(\tau) \times \sigma_1 \quad \text{und} \quad \sigma_2 \times \mathcal{F}^{(\ell_2)}(\rho) \subseteq \sigma_1 \times \mathcal{F}^{(\ell_1)}(\rho).$$

Per definitionem ist eine der Übermengen ein Blatt und somit gilt hierbei Gleichheit der Mengen, also $\sigma_2 = \sigma_1$ sowie $\ell_1 = \ell_2$.

2.Schritt: $\mathcal{J} \subseteq \bigcup_\ell \mathcal{U}_\ell(\tau \times \rho)$: Sei $j \in \mathcal{J}$. Wir definieren $\tau_0 := \mathcal{I} = \mathcal{F}^{(0)}(\tau)$, $\sigma_0 := \mathcal{J}$, und $\rho_0 := \mathcal{K} = \mathcal{F}^{(0)}(\rho)$. Falls weder $\tau_0 \times \sigma_0$ noch $\sigma_0 \times \rho_0$ Blatt ist, existieren Söhne $\tau_1 \times \sigma_1$ und $\tilde{\sigma}_1 \times \rho_1$ mit $j \in \sigma_1 \cap \tilde{\sigma}_1$. Aus $\operatorname{level}(\sigma_1) = 1 = \operatorname{level}(\tilde{\sigma}_1)$ folgt somit $\sigma_1 = \tilde{\sigma}_1$. Schreitet man induktiv nach unten zum Level ℓ bis entweder $\tau_\ell \times \sigma_\ell$ oder $\sigma_\ell \times \rho_\ell$ ein Blatt ist, dann gilt $j \in \sigma_\ell \in \mathcal{U}_\ell(\tau \times \rho)$.

3.Schritt: Die Abschätzung der Mächtigkeit der Mengen $\mathcal{U}_\ell(\tau \times \rho)$ folgt schließlich aus

$$\begin{aligned}\left|\mathcal{U}_\ell(\tau \times \rho)\right| &\leq \left|\{\sigma \in \mathbb{T}_{\mathcal{J}} \,:\, \mathcal{F}^{(\ell)}(\tau) \times \sigma \in \mathbb{T}_{\mathcal{I}\times\mathcal{J}}\}\right| \leq C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}),\\ \left|\mathcal{U}_\ell(\tau \times \rho)\right| &\leq \left|\{\sigma \in \mathbb{T}_{\mathcal{J}} \,:\, \sigma \times \mathcal{F}^{(\ell)}(\rho) \in \mathbb{T}_{\mathcal{J}\times\mathcal{K}}\}\right| \leq C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}}),\end{aligned}$$

womit das Lemma gezeigt ist. □

Aus Lemma 4.11 folgt speziell

$$(\mathbf{AB})|_{\tau\times\rho} = \sum_{j\in\mathcal{J}} \mathbf{A}|_{\tau\times\{j\}}\mathbf{B}|_{\{j\}\times\rho} \overset{!}{=} \sum_{\ell=0}^{\operatorname{level}(\tau\times\rho)} \sum_{\sigma\in\mathcal{U}_\ell(\tau\times\rho)} \mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho}. \tag{4.10}$$

Die nachfolgende Proposition beschreibt die exakte Multiplikation zweier $\mathcal{H}$-Matrizen.

Proposition 4.12. *Seien $\mathbf{A} \in \mathcal{H}(r_A, \mathbb{P}_{\mathcal{I}\mathcal{J}})$ mit einer zulässigen Partition $\mathbb{P}_{\mathcal{I}\mathcal{J}}$ basierend auf dem Baum $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ und $\mathbf{B} \in \mathcal{H}(r_B, \mathbb{P}_{\mathcal{J}\mathcal{K}})$ mit einer zulässigen Partition $\mathbb{P}_{\mathcal{J}\mathcal{K}}$ basierend auf dem Baum $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$. Sei $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ der induzierte Produktbaum. Dann erfüllt das exakte Produkt $\mathbf{C} = \mathbf{AB}$, dass $\mathbf{C} \in \mathcal{H}(r_C, \mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}})$, wobei der Rang r_C beschränkt ist durch*

$$r_C \leq C_{\mathrm{sp}}(\operatorname{depth}+1)\max\{r_A, r_B, n_{\mathrm{blatt}}\}. \tag{4.11}$$

Die Matrix $\mathbf{C}$ kann in weniger als

$$2(\operatorname{depth}+1)^2 C_{\mathrm{sp}}^3 \max\{r_A, r_B, n_{\mathrm{blatt}}\}^2(|\mathcal{I}| + 2|\mathcal{J}| + |\mathcal{K}|) = \mathcal{O}(r^2 N \log^2 N), \tag{4.12}$$

arithmetischen Operationen berechnet werden, wobei $C_{\mathrm{sp}} := \max\{C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}), C_{\mathrm{sp}}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}})\}$, $\operatorname{depth} := \max\{\operatorname{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}}), \operatorname{depth}(\mathbb{T}_{\mathcal{J}\times\mathcal{K}})\}$, $N := |\mathcal{I}|+|\mathcal{J}|+|\mathcal{K}|$, und $r := \max\{r_A, r_B\}$.

Beweis. **1.Schritt:** Wir zeigen (4.11):

Es gilt $(\text{level}(\tau \times \rho) + 1) \leq \text{depth}(\mathbb{T}) + 1 \leq \text{depth} + 1$. Somit hat die Multiplikation mittels der Darstellung (4.10) mit Lemma 4.11 weniger als $C_{\text{sp}}\,(\text{depth} + 1)$ Additionen. Es genügt also $\text{rang}(\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho}) \leq \max\{r_A, r_B, n_{\text{blatt}}\}$ für $\sigma \in \mathcal{U}_\ell(\tau \times \rho)$ zu zeigen:

Aus der Definition von $\mathcal{U}_\ell(\tau \times \rho)$ folgt entweder $\mathcal{F}^{(\ell)}(\tau) \times \sigma$ oder $\sigma \times \mathcal{F}^{(\ell)}(\rho)$ oder beide sind Blatt. Daher folgt

- entweder $\text{rang}(\mathbf{A}|_{\tau\times\sigma}) \leq \text{rang}(\mathbf{A}|_{\mathcal{F}^{(\ell)}(\tau)\times\sigma}) \leq \max\{r_A, n_{\text{blatt}}\}$
- oder $\text{rang}(\mathbf{B}|_{\sigma\times\rho}) \leq \text{rang}(\mathbf{B}|_{\sigma\times\mathcal{F}^{(\ell)}(\rho)}) \leq \max\{r_B, n_{\text{blatt}}\}$.

Zusammengesetzt folgt somit $\text{rang}(\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho}) \leq \max\{r_A, r_B, n_{\text{blatt}}\}$.

2.Schritt: Wir schätzen den Aufwand der Multiplikation ab:

Um $\mathbf{C} = \mathbf{AB}$ im $\mathcal{H}$-Matrix Format zu erhalten benötigen wir blockweise Faktorisierungen

$$(\mathbf{AB})|_{\tau\times\rho} = \mathbf{V}_{\tau\rho}\mathbf{W}_{\tau\rho}^T \quad \text{mit } \mathbf{V}_{\tau\rho} \in \mathbb{R}^{\tau\times r_C}, \mathbf{W}_{\tau\rho} \in \mathbb{R}^{\rho\times r_C}. \tag{4.13}$$

Hat man für jeden Summanden $\mathbf{A}|_{\tau\times\sigma_i}\mathbf{B}|_{\sigma_i\times\rho} = \mathbf{V}_i\mathbf{W}_i^T$ von (4.10) eine derartige Faktorisierung mit Rang $\widehat{r}_C := \max\{r_A, r_B, n_{\text{blatt}}\}$, dann ist

$$\mathbf{V}_{\tau\rho} := (\mathbf{V}_1, \mathbf{V}_2, \dots), \mathbf{W}_{\tau\rho} := (\mathbf{W}_1, \mathbf{W}_2, \dots)$$

die gewünschte Faktorisierung. Nach Definition von $\mathcal{U}_\ell(\tau \times \rho)$ ist entweder $\mathcal{F}^{(\ell)}(\tau) \times \sigma$ oder $\sigma \times \mathcal{F}^{(\ell)}(\rho)$ oder beide ein Blatt. Daher treten nur folgende Fälle auf:

- $\mathcal{F}^{(\ell)}(\tau) \times \sigma$ ist zulässig. Daher ist der Block $\mathbf{A}|_{\mathcal{F}^{(\ell)}(\tau)\times\sigma} = \mathbf{VW}^T$ faktorisiert gespeichert. Somit,

$$\mathbf{A}|_{\tau\times\sigma} = \mathbf{V}|_{\tau\times\sigma}\mathbf{W}^T \quad \text{und} \quad \mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho} = \mathbf{V}|_{\tau\times\sigma}(\mathbf{W}^T\mathbf{B}|_{\sigma\times\rho}) = \mathbf{V}|_{\tau\times\sigma}(\mathbf{B}|_{\sigma\times\rho}^T\mathbf{W})^T =: \mathbf{V}_i\mathbf{W}_i^T.$$

- $\sigma \times \mathcal{F}^{(\ell)}(\rho)$ ist zulässig, also $\mathbf{B}_{\sigma\times\mathcal{F}^{(\ell)}(\rho)} = \mathbf{VW}^T$. Somit,

$$\mathbf{B}|_{\sigma\times\rho} = \mathbf{VW}|_{\sigma\times\rho}^T \quad \text{und} \quad \mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho} = (\mathbf{A}|_{\tau\times\sigma}\mathbf{V})\mathbf{W}|_{\sigma\times\rho}^T =: \mathbf{V}_i\mathbf{W}_i^T.$$

- $\mathcal{F}^{(\ell)}(\tau) \times \sigma$ und $\sigma \times \mathcal{F}^{(\ell)}(\rho)$ sind nicht zulässig mit $|\sigma| \leq n_{\text{blatt}}$. Dann,

$$\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho} = \mathbf{A}|_{\tau\times\sigma}(\mathbf{B}|_{\sigma\times\rho}^T)^T =: \mathbf{V}_i\mathbf{W}_i^T.$$

- $\mathcal{F}^{(\ell)}(\tau) \times \sigma$ ist nicht zulässig mit $|\tau| \leq |\mathcal{F}^{(\ell)}(\tau)| \leq n_{\text{blatt}}$. Dann,

$$\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho} = \mathbf{I}\,(\mathbf{B}|_{\sigma\times\rho}^T\mathbf{A}|_{\tau\times\sigma}^T)^T =: \mathbf{V}_i\mathbf{W}_i^T.$$

- $\sigma \times \mathcal{F}^{(\ell)}(\rho)$ ist nicht zulässig mit $|\rho| \leq |\mathcal{F}^{(\ell)}(\rho)| \leq n_{\text{blatt}}$. Dann,

$$\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho} = (\mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho})\mathbf{I} =: \mathbf{V}_i\mathbf{W}_i^T.$$

Für die Berechnung von $\mathbf{V}_i$ und $\mathbf{W}_i$ benötigen wir also entweder Kopien oder maximal $\widehat{r}_C$ Matrix-Vektor-Multiplikationen von links mit $\mathbf{B}|_{\sigma\times\rho}^T$ oder $\mathbf{A}|_{\tau\times\sigma}$. Somit folgt für die Komplexität der Multiplikation mit Proposition 4.5 und Lemma 4.11

$$\begin{aligned}
N_{\text{mult}}(\mathbf{AB}) &\leq \sum_{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}} \sum_{\ell=0}^{\text{level}(\tau\times\rho)} \sum_{\sigma\in\mathcal{U}_\ell(\tau\times\rho)} \widehat{r}_C\Big(N_{\text{MVM}}(\mathbf{A}|_{\tau\times\sigma}) + N_{\text{MVM}}(\mathbf{B}|_{\sigma\times\rho}^T)\Big) \\
&\leq 2\,\widehat{r}_C \sum_{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}} \sum_{\ell=0}^{\text{level}(\tau\times\rho)} \sum_{\sigma\in\mathcal{U}_\ell(\tau\times\rho)} \Big(N_{\text{Storage}}(\mathbf{A}|_{\tau\times\sigma}) + N_{\text{Storage}}(\mathbf{B}|_{\sigma\times\rho})\Big) \\
&= 2\,\widehat{r}_C \sum_{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}} \Big(N_{\text{Storage}}(\mathbf{A}|_{\tau\times\mathcal{J}}) + N_{\text{Storage}}(\mathbf{B}|_{\mathcal{J}\times\rho})\Big) \\
&= 2\,\widehat{r}_C \sum_{\ell=0}^{\text{depth}} \sum_{\substack{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}} \\ \text{level}(\tau)=\ell}} \Big(N_{\text{Storage}}(\mathbf{A}|_{\tau\times\mathcal{J}}) + N_{\text{Storage}}(\mathbf{B}|_{\mathcal{J}\times\rho})\Big).
\end{aligned}$$

Teilt man die zweite Summe weiter auf, so folgt mit Lemma 4.10

$$\begin{aligned}
\sum_{\ell=0}^{\text{depth}} \sum_{\substack{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}} \\ \text{level}(\tau)=\ell}} N_{\text{Storage}}(\mathbf{A}|_{\tau\times\mathcal{J}}) &= \sum_{\ell=0}^{\text{depth}} \sum_{\substack{\tau\in\mathbb{T}_{\mathcal{I}} \\ \text{level}(\tau)=\ell}} \sum_{\substack{\rho\in\mathbb{T}_{\mathcal{K}} \\ \tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}}} N_{\text{Storage}}(\mathbf{A}|_{\tau\times\mathcal{J}}) \\
&\leq C_{\text{sp}}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}})\,\Big(\text{depth}+1\Big)N_{\text{Storage}}(\mathbf{A}) \\
&\leq C_{\text{sp}}^2\,\Big(\text{depth}+1\Big)N_{\text{Storage}}(\mathbf{A}). \qquad (4.14)
\end{aligned}$$

Proposition 2.11 liefert

$$\begin{aligned}
N_{\text{Storage}}(\mathbf{A}) &\leq C_{\text{sp}}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}})\Big(\text{depth}(\mathbb{T}_{\mathcal{I}\times\mathcal{J}})+1\Big)\max\{r_A, n_{\text{blatt}}\}\,(|\mathcal{I}|+|\mathcal{J}|) \\
&\leq C_{\text{sp}}(\text{depth}+1)\widehat{r}_C\,(|\mathcal{I}|+|\mathcal{J}|).
\end{aligned}$$

Setzt man dies in (4.14) ein und verwendet die selbe Argumentation für den Term mit $N_{\text{Storage}}(\mathbf{B}|_{\mathcal{J}\times\rho})$ liefert dies schließlich

$$\begin{aligned}
N_{\text{mult}}(\mathbf{AB}) &\leq 2\,\widehat{r}_C \sum_{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}} \Big(N_{\text{Storage}}(\mathbf{A}|_{\tau\times\mathcal{J}}) + N_{\text{Storage}}(\mathbf{B}|_{\mathcal{J}\times\rho})\Big) \\
&\leq 2\,C_{\text{sp}}^3\,(\text{depth}+1)^2\,\widehat{r}_C^2\,\Big(|\mathcal{I}|+2|\mathcal{J}|+|\mathcal{K}|\Big),
\end{aligned}$$

was den Beweis beendet. □

4.4.2 Vorgeschriebene Zielpartitionen

Oftmals möchte man bei der Multiplikation nicht die Zielpartition verändern, also nicht die induzierte Partition verwenden, sondern eventuell die Partition der Multiplikanden, oder eine andere Partition vorschreiben. Hierbei muss klarerweise eine Verbindung zwischen der gewünschten Partition und der induzierten Partition hergestellt werden. Im Folgenden geschieht dies mit der so genannten Idempotenz-Konstante, die misst wie gut Blätter des Produktbaumes in dem gewünschten Baum aufgelöst werden, also wieviele Blätter des Produktbaumes in einem Blatt des gewünschten Baumes liegen.

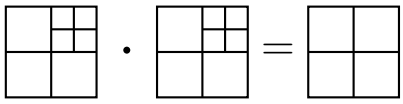
Definition 4.13. *Seien $\mathbb{T}_{\mathcal{I}}, \mathbb{T}_{\mathcal{J}}, \mathbb{T}_{\mathcal{K}}$ Clusterbäume und $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$, $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$, und $\mathbb{T}_{\mathcal{I}\times\mathcal{K}}$ die zugehörigen Block-Cluster-Bäume. Sei $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ der von $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$, $\mathbb{T}_{\mathcal{J}\times\mathcal{K}}$ induzierte Produktbaum für $\mathcal{I}\times\mathcal{K}$. Für $\tau\in\mathbb{T}_{\mathcal{I}}$ und $\rho\in\mathbb{T}_{\mathcal{K}}$ definieren wir*

$$C_{\mathrm{id}}(\tau\times\rho) := \left|\left\{\tau'\times\rho'\in S^*(\tau)\times S^*(\rho) \,:\, \exists\sigma'\in\mathbb{T}_{\mathcal{J}},\ \tau'\times\sigma'\in\mathbb{T}_{\mathcal{I}\times\mathcal{J}},\ \sigma'\times\rho'\in\mathbb{T}_{\mathcal{J}\times\mathcal{K}}\right\}\right|,$$

*wobei $S^*(\tau) := \{\tau'\in\mathbb{T}_{\mathcal{I}} : \tau'\subseteq\tau\}$ und $S^*(\rho) := \{\rho'\in\mathbb{T}_{\mathcal{K}} : \rho'\subseteq\rho\}$ jeweils die Nachfahren von τ und ρ bezeichnet. Die **Idempotenz-Konstante** ist schlussendlich definiert als*

$$C_{\mathrm{id}}(\mathbb{T}_{\mathcal{I}\times\mathcal{K}}) := \max\{C_{\mathrm{id}}(\tau\times\rho) : \tau\times\rho\in\mathrm{leaves}(\mathbb{T}_{\mathcal{I}\times\mathcal{K}})\}.$$

Bemerkung 4.14. *Im Fall $\mathcal{I}=\mathcal{J}=\mathcal{K}$ heißt ein Block-Cluster-Baum idempotent, falls $\mathbb{T}_{\mathcal{I}\times\mathcal{I}} = \mathbb{T}_{\mathcal{I}\times\mathcal{I}}\cdot\mathbb{T}_{\mathcal{I}\times\mathcal{I}}$. Idempotenz impliziert $C_{\mathrm{id}}(\mathbb{T}_{\mathcal{I}\times\mathcal{I}}) = 1$, aber die Umkehrung gilt nicht, wie das folgende Beispiel zeigt:*



Hier ist $C_{\mathrm{id}}(\mathbb{T}_{\mathcal{I}\times\mathcal{I}}) = 1$, aber der Produkt-Baum gröber als der ursprüngliche Baum.

Beim Übergang von der induzierten Partition zu einer gewünschten Partition kann es vorkommen, dass Blöcke vergröbert werden müssen. Die nachfolgende Bemerkung zeigt, dass dies effizient möglich ist.

Bemerkung 4.15. *Sei $\tau\times\rho\in\mathbb{T}_{\mathcal{I}\times\mathcal{K}}$ mit vier zulässigen Söhnen $\tau_i\times\rho_j\in\mathbb{P}_{\mathrm{far}}$, $i,j\in\{1,2\}$, wobei*

$$\mathbf{A}|_{\tau_i\times\rho_j} = \mathbf{V}_{ij}\mathbf{W}_{ij}^T \qquad \text{mit } \mathbf{V}_{ij}\in\mathbb{R}^{|\tau_i|\times r} \text{ und } \mathbf{W}_{ij}\in\mathbb{R}^{|\rho_j|\times r}.$$

Definiert man Matrizen

$$\mathbf{V} := \begin{pmatrix} \mathbf{V}_{11} & 0 & \mathbf{V}_{12} & 0 \\ 0 & \mathbf{V}_{21} & 0 & \mathbf{V}_{22} \end{pmatrix} \in \mathbb{R}^{|\tau|\times 4r} \quad \text{und} \quad \mathbf{W} := \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{21} & 0 & 0 \\ 0 & 0 & \mathbf{W}_{12} & \mathbf{W}_{22} \end{pmatrix} \in \mathbb{R}^{|\rho|\times 4r},$$

dann gilt

$$\mathbf{V}\mathbf{W}^T = \begin{pmatrix} \mathbf{V}_{11}\mathbf{W}_{11}^T & \mathbf{V}_{12}\mathbf{W}_{12}^T \\ \mathbf{V}_{21}\mathbf{W}_{21}^T & \mathbf{V}_{12}\mathbf{W}_{22}^T \end{pmatrix} = \begin{pmatrix} \mathbf{A}|_{\tau_1\times\rho_1} & \mathbf{A}|_{\tau_1\times\rho_2} \\ \mathbf{A}|_{\tau_2\times\rho_1} & \mathbf{A}|_{\tau_2\times\rho_2} \end{pmatrix} = \mathbf{A}|_{\tau\times\rho}.$$

Der Vater-Block $\mathbf{A}|_{\tau\times\rho}$ hat also einen maximalen Rang von $4r$. Mit Lemma 4.1 kann die SVD und somit eine Niedrigrangfaktorisierung von $\mathbf{A}|_{\tau\times\rho}$ in weniger als

$$6(4r)^2(|\tau| + |\rho|) + 23(4r)^3$$

arithmetischen Operationen berechnet werden.

Proposition 4.16. *Seien $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}, \mathbb{T}_{\mathcal{J}\times\mathcal{K}}, \mathbb{T}_{\mathcal{I}\times\mathcal{K}}$ Cluster-Bäume und $\mathbb{P}_{\mathcal{I}\mathcal{J}}, \mathbb{P}_{\mathcal{J}\mathcal{K}}, \mathbb{P}_{\mathcal{I}\mathcal{K}}$ gegebene Partitionen basierend auf den jeweiligen Cluster-Bäumen. Seien $\mathbf{A} \in \mathcal{H}(r_A, \mathbb{P}_{\mathcal{I}\mathcal{J}})$ und $\mathbf{B} \in \mathcal{H}(r_B, \mathbb{P}_{\mathcal{J}\mathcal{K}})$ $\mathcal{H}$-Matrizen. Dann ist das Produkt $\mathbf{C} := \mathbf{A}\mathbf{B}$ in $\mathcal{H}(r_C, \mathbb{P}_{\mathcal{I}\mathcal{K}})$ wobei der maximale blockweise Rang r_C beschränkt ist durch*

$$r_C \leq C_{\mathrm{id}} C_{\mathrm{sp}} (\mathrm{depth} + 1) \max\{r_A, r_B, n_{\mathrm{blatt}}\}. \tag{4.15}$$

Sei $r \leq r_C$. Dann benötigt die Berechnung von $\mathcal{T}_{\mathcal{H}(r)}(\mathbf{C})$ weniger als

$$\begin{aligned} N_{\mathrm{MM,r}}(\mathbf{A}\mathbf{B}) \leq\ & 23 C_{\mathrm{id}}^3 C_{\mathrm{sp}}^3 (\mathrm{depth}+1)^3 \max\{r_A, r_B, n_{\mathrm{blatt}}\}^3 |\mathbb{P}_{\mathrm{far}}^{\mathcal{I}\times\mathcal{K}}| \\ & + 9 C_{\mathrm{id}}^2 C_{\mathrm{sp}}^3 (\mathrm{depth}+1)^2 \max\{r_A, r_B, n_{\mathrm{blatt}}\}^2 (|\mathcal{I}| + |\mathcal{J}| + |\mathcal{K}|) \\ & = \mathcal{O}(\max\{r_A, r_B, n_{\mathrm{blatt}}\}^3 N \log^3 N) \end{aligned} \tag{4.16}$$

arithmetische Operationen, wobei $N = |\mathcal{I}| + |\mathcal{J}| + |\mathcal{K}|$.

Beweis. Sei $\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ der induzierte Produktbaum, $\mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}}$ die induzierte Partition, und $\widetilde{r}_C := C_{\mathrm{sp}}(\mathrm{depth}+1)\max\{r_A, r_B, n_{\mathrm{blatt}}\}$. Laut Proposition 4.12 gilt $\mathbf{C} \in \mathcal{H}(\widetilde{r}_C, \mathbb{P}_{\mathcal{I}\mathcal{J}\mathcal{K}})$. Wir zeigen, dass nicht-disjunkte Blätter im Produktbaum sowie dem Baum $\mathbb{T}_{\mathcal{I}\times\mathcal{K}}$ vollständig ineinander enthalten sein müssen, was schließlich die korrekte Rangabschätzung liefert.

1.Schritt: Seien $\tau\times\rho \in \mathrm{leaves}(\mathbb{T}_{\mathcal{I}\times\mathcal{K}})$ und $\widetilde{\tau}\times\widetilde{\rho} \in \mathrm{leaves}(\mathbb{T}_{\mathcal{I}\mathcal{J}\mathcal{K}})$ mit $(\tau\times\rho)\cap(\widetilde{\tau}\times\widetilde{\rho}) \neq \emptyset$. Dann gilt entweder $\tau\times\rho \subseteq \widetilde{\tau}\times\widetilde{\rho}$ oder $\widetilde{\tau}\times\widetilde{\rho} \subseteq \tau\times\rho$:

Um dies einzusehen, sei $(i,k) \in (\tau\times\rho)\cap(\widetilde{\tau}\times\widetilde{\rho})$ mit $i \in \tau\cap\widetilde{\tau}$ und $k \in \rho\cap\widetilde{\rho}$. Aus der Definition eines Cluster-Baumes folgt dann

$$\tau \subseteq \widetilde{\tau} \quad \text{oder} \quad \widetilde{\tau} \subseteq \tau \qquad \text{und} \qquad \rho \subseteq \widetilde{\rho} \quad \text{oder} \quad \widetilde{\rho} \subseteq \rho.$$

Aus der stufentreue folgt $\mathrm{level}(\tau) = \mathrm{level}(\rho)$ und $\mathrm{level}(\widetilde{\tau}) = \mathrm{level}(\widetilde{\rho})$ und somit entweder

$$\tau \subseteq \widetilde{\tau} \quad \text{und} \quad \rho \subseteq \widetilde{\rho} \qquad \text{oder} \qquad \widetilde{\tau} \subseteq \tau \quad \text{und} \quad \widetilde{\rho} \subseteq \rho.$$

2.Schritt: Für jedes Blatt $\tau\times\rho \in \mathbb{T}_{\mathcal{I}\times\mathcal{K}}$ gilt $\mathrm{rang}(\mathbf{C}|_{\tau\times\rho}) \leq r_C$:

Ist $\widetilde{\tau} \times \widetilde{\rho} \in \operatorname{leaves}(\mathbb{T}_{\mathcal{IJK}})$ mit $\tau \times \rho \subseteq \widetilde{\tau} \times \widetilde{\rho}$, dann folgt mit $C_{\mathrm{id}} \geq 1$ und $r_C = C_{\mathrm{id}}\widetilde{r}_C$, dass

$$\operatorname{rang}(\mathbf{C}|_{\tau\times\rho}) \leq \operatorname{rang}(\mathbf{C}|_{\widetilde{\tau}\times\widetilde{\rho}}) \leq \widetilde{r}_C \leq r_C.$$

Andernfalls ist das Blatt $\tau \times \rho \in \operatorname{leaves}(\mathbb{T}_{\mathcal{I}\times\mathcal{K}})$ Vereinigung von Blättern $\widetilde{\tau} \times \widetilde{\rho} \in \mathbb{T}_{\mathcal{IJK}}$. Somit folgt

$$\operatorname{rang}(\mathbf{C}|_{\tau\times\rho}) \leq \left|\{\widetilde{\tau} \times \widetilde{\rho} \in \operatorname{leaves}(\mathbb{T}_{\mathcal{IJK}}) \,:\, \widetilde{\tau} \times \widetilde{\rho} \subseteq \tau \times \rho\}\right| \widetilde{r}_C \leq C_{\mathrm{id}}\widetilde{r}_C = r_C.$$

3.Schritt: Abschätzung für den Rechenaufwand:

Wir berechnen zunächst das Produkt $\mathbf{C} = \mathbf{AB} \in \mathcal{H}(\widetilde{r}_C, \mathbb{P}_{\mathcal{IJK}})$ mit der induzierten Partition in maximal

$$2(\operatorname{depth}+1)^2 C_{\mathrm{sp}}^3 \max\{r_A, r_B, n_{\mathrm{blatt}}\}^2(|\mathcal{I}| + 2|\mathcal{J}| + |\mathcal{K}|) \tag{4.17}$$

arithmetischen Operationen (Proposition 4.12). Es bleibt dann noch die Konversion von Blöcken von $\mathbf{C}$ auf die Blockstruktur von $\mathbb{P}_{\mathcal{I}\times\mathcal{K}}$, wobei zwischen zulässigen und nichtzulässigen Blöcken unterschieden wird.

1.Fall: Sei $\tau \times \rho \in \mathbb{P}_{\mathcal{I}\times\mathcal{K}}$ mit $\tau \times \rho \subseteq \widetilde{\tau} \times \widetilde{\rho}$ für ein $\widetilde{\tau} \times \widetilde{\rho} \in \mathbb{T}_{\mathcal{IJK}}$.

Hierbei gilt $\mathbf{C}|_{\widetilde{\tau}\times\widetilde{\rho}} = \widetilde{\mathbf{V}}\widetilde{\mathbf{W}}^T$ (vgl. Beweis von Proposition 4.12), was eine Faktorisierung $\mathbf{C}|_{\tau\times\rho} = \mathbf{V}\mathbf{W}^T$ mit $\mathbf{V} \in \mathbb{R}^{\tau\times\widetilde{r}_C}$ und $\mathbf{W} \in \mathbb{R}^{\rho\times\widetilde{r}_C}$ induziert.

- Falls $\tau \times \rho \in \mathbb{P}_{\mathrm{near}}^{\mathcal{I}\times\mathcal{K}}$, muss $\mathbf{C}|_{\tau\times\rho} = \mathbf{V}\mathbf{W}^T$ in eine vollbesetzte Matrix umgewandelt werden. Das benötigt $|\tau|$ Matrix-Vektor-Multiplikationen mit $\mathbf{V}$, womit der Aufwand abschätzbar ist durch

  $$|\tau||\rho|(2\widetilde{r}_C - 1) \leq 2n_{\mathrm{blatt}}\widetilde{r}_C\,(|\tau| + |\rho|) \leq 2\widetilde{r}_C^2\,(|\tau| + |\rho|),$$

  da $\widetilde{r}_C \geq n_{\mathrm{blatt}}$.

- Falls $\tau \times \rho \in \mathbb{P}_{\mathrm{far}}^{\mathcal{I}\times\mathcal{K}}$, benötigt man mit Proposition 4.3 weniger als

  $$6\widetilde{r}_C^2(|\tau| + |\rho|) + 23\widetilde{r}_C^3 + r|\tau|$$

  Operationen zur Berechnung von $\mathcal{T}_r(\mathbf{C}|_{\tau\times\rho})$ in faktorisierter Form.

2.Fall: Sei $\tau \times \rho \in \mathbb{P}_{\mathcal{I}\times\mathcal{K}}$ mit $\tau \times \rho \supseteq \widetilde{\tau} \times \widetilde{\rho}$ für ein $\widetilde{\tau} \times \widetilde{\rho} \in \mathbb{T}_{\mathcal{IJK}}$. Dann wird $\tau \times \rho$ maximal in C_{id} Blöcke $\widetilde{\tau}_i \times \widetilde{\rho}_i$ zerteilt.

- Für den Fall $\tau \times \rho \in \mathbb{P}_{\mathrm{near}}^{\mathcal{I}\times\mathcal{K}}$, gehen wir auf jedem Block $\widetilde{\tau}_i \times \widetilde{\rho}_i$ wie in Fall 1 vor, was auf eine maximale Komplexität von

  $$\sum_i 2\widetilde{r}_C^2\,(|\widetilde{\tau}_i| + |\widetilde{\rho}_i|) \leq 2C_{\mathrm{id}}\widetilde{r}_C^2\,(|\tau| + |\rho|)$$

  führt.

- Im Fall $\tau \times \rho \in \mathbb{P}_{\text{far}}^{\mathcal{I}\times\mathcal{K}}$ muss die Blockstruktur vergröbert werden. Wir gehen hierbei wie in Bemerkung 4.15 vor und erhalten eine Faktorisierung $\mathbf{C}|_{\tau\times\rho} = \mathbf{V}\mathbf{W}^T$ aus den Faktorisierungen von $\mathbf{C}_{\widetilde{\tau}_i\times\widetilde{\rho}_i}$. Die Berechnung und Kompression auf Rang-r dieser benötigt maximal

$$6(C_{\text{id}}\widetilde{r}_C)^2(|\tau| + |\rho|) + 23(C_{\text{id}}\widetilde{r}_C)^3 + |\tau|r$$

  Operationen (vgl. Proposition 4.3).

Gesamt ist also der Aufwand der Transformation N_{trans} sämtlicher Nah- und Fernfeldblöcke beschränkt durch

$$\begin{aligned}
\sum_{\tau\times\rho\in\mathbb{P}_{\mathcal{I}\times\mathcal{K}}} N_{\text{trans}}(\mathbf{C}|_{\tau\times\rho}) &= \sum_{\tau\times\rho\in\mathbb{P}_{\text{far}}^{\mathcal{I}\times\mathcal{K}}} N_{\text{trans}}(\mathbf{C}|_{\tau\times\rho}) + \sum_{\tau\times\rho\in\mathbb{P}_{\text{near}}^{\mathcal{I}\times\mathcal{K}}} N_{\text{trans}}(\mathbf{C}|_{\tau\times\rho}) \\
&\leq \sum_{\tau\times\rho\in\mathbb{P}_{\text{far}}^{\mathcal{I}\times\mathcal{K}}} 23C_{\text{id}}^3\widetilde{r}_C^3 + (6C_{\text{id}}^2\widetilde{r}_C^2 + r)(|\tau| + |\rho|) + 2C_{\text{id}}\widetilde{r}_C^2 \sum_{\tau\times\rho\in\mathbb{P}_{\text{near}}^{\mathcal{I}\times\mathcal{K}}} (|\tau| + |\rho|) \\
&\leq 23C_{\text{id}}^3\widetilde{r}_C^3|\mathbb{P}_{\text{far}}^{\mathcal{I}\times\mathcal{K}}| + 7C_{\text{sp}}(\text{depth}+1)C_{\text{id}}^2\widetilde{r}_C^2(|\mathcal{I}| + |\mathcal{K}|) \\
&= C_{\text{sp}}^3(\text{depth}+1)^3\Big(23C_{\text{id}}^3\max\{r_A, r_B, n_{\text{blatt}}\}^3|\mathbb{P}_{\text{far}}^{(\mathcal{I}\times\mathcal{K})}| \\
&\quad + 7C_{\text{id}}^2\max\{r_A, r_B, n_{\text{blatt}}\}^2(|\mathcal{I}| + |\mathcal{K}|)\Big)
\end{aligned}$$

Gemeinsam mit (4.17) folgt schließlich das gewünschte Resultat. □

Bemerkung 4.17. *Die Berechnung von*

$$\mathcal{T}_r\Big((\mathbf{AB})|_{\tau\times\rho}\Big) = \mathcal{T}_r\Big(\sum_{\ell=0}^{\text{level}(\tau\times\rho)} \sum_{\sigma\in\mathcal{U}_\ell(\tau\times\rho)} \mathbf{A}|_{\tau\times\sigma}\mathbf{B}|_{\sigma\times\rho}\Big)$$

kann relativ teuer werden. Eine Möglichkeit den Aufwand zu reduzieren ist, nach jeder Addition die Bestapproximation zu berechnen. In der Literatur wird dies fast truncation *genannt.*

4.5 $\mathcal{H}$-Inversion und $\mathcal{H}$-LU-Zerlegung

Basierend auf der $\mathcal{H}$-Addition und Multiplikation können noch weitere arithmetische Operationen im $\mathcal{H}$-Matrix Format definiert werden, wie beispielsweise die $\mathcal{H}$-Inversion oder die $\mathcal{H}$-LU-Zerlegung.

Man könnte wie zuvor die $\mathcal{H}$-Inversion einfach mittels der Bestapproximation definieren als $\mathcal{T}_{\mathcal{H}(r)}(\mathbf{A}^{-1})$. Dies ist allerdings nicht praktikabel, da hierfür die Inverse Matrix berechnet werden muss. Wir werden im Folgenden eine andere Variante der $\mathcal{H}$-Inversion angeben, die keine exakte Inversion von $\mathbf{A}$ benötigt.

Im Gegensatz zu Addition und Multiplikation liefert diese allerdings im Allgemeinen nicht die Best-Approximationen, also

$$\mathtt{Inv}_{\mathcal{H}(r)}(\mathbf{A}) \neq \mathcal{T}_{\mathcal{H}(r)}(\mathbf{A}^{-1}).$$

Wir werden im Folgenden eine mögliche Berechnung der $\mathcal{H}$-Inversen mit Hilfe von Schur Komplementen vorstellen.

Lemma 4.18. *Sei $\mathbf{A} \in \mathbb{R}^{n\times n}$ so, dass alle ersten Submatrizen $(a_{jk})_{j,k=1,\dots,m}$ für $1 \leq m \leq n$ invertierbar sind (oder äquivalent $\mathbf{A}$ eine LU-Zerlegung besitzt). Wir schreiben $\mathbf{A}$ als Blockmatrix*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}, \tag{4.18}$$

*mit $\mathbf{A}_{11} \in \mathbb{R}^{n_1\times n_1}$, $\mathbf{A}_{12} \in \mathbb{R}^{n_1\times n_2}$, $\mathbf{A}_{21} \in \mathbb{R}^{n_2\times n_1}$, $\mathbf{A}_{22} \in \mathbb{R}^{n_2\times n_2}$, wobei $n = n_1 + n_2$. Dann sind $\mathbf{A}_{11}$ sowie das **Schur Komplement***

$$\mathbf{S} := \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \tag{4.19}$$

invertierbar. Die Inverse von $\mathbf{A}$ kann dann geschrieben werden als Blockmatrix

$$\mathbf{A}^{-1} = \begin{pmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{S}^{-1} \end{pmatrix}. \tag{4.20}$$

Beweis. $\mathbf{A}_{11}$ ist laut Voraussetzung invertierbar, ebenso wie $\mathbf{A}$. Wir verwenden den Blockmatrix-Ansatz für die Inverse $\mathbf{B} := \mathbf{A}^{-1}$:

$$\begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \overset{!}{=} \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} \end{pmatrix}$$

Die zweite Spalte liefert $\mathbf{0} = \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22}$, also $\mathbf{B}_{12} = -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{B}_{22}$, sowie $\mathbf{I} = \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} = -\mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{B}_{22} + \mathbf{A}_{22}\mathbf{B}_{22} = \mathbf{S}\mathbf{B}_{22}$. Somit folgt $\mathbf{S}\mathbf{B}_{22} = \mathbf{I}$ und $\mathbf{S}$ muss invertierbar sein. Die Formel (4.20) folgt durch einfaches Multiplizieren $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. □

Die Darstellung (4.20) der Inversen liefert direkt einen rekursiven Algorithmus zur Berechnung einer $\mathcal{H}$-Matrix-Approximation an $\mathbf{A}_{\mathcal{H}}^{-1}$: Man geht von oben rekursiv durch den Baum, invertiert nicht zulässige Blätter exakt und ersetzt in (4.20) die arithmetischen Operationen durch $\oplus_r$ und $\odot_r$.

Obwohl die derartige Berechnung eine $\mathcal{H}$-Matrix liefert, ist der Fehler zu der exakten Inversen im Allgemeinen nicht kontrollierbar (vgl. Beispiel bei $\mathcal{H}$-Addition). Dennoch funktioniert der $\mathcal{H}$-Inversionsalgorithmus in der Praxis gut.

Die Voraussetzung von Lemma 4.18 sind für $\mathcal{H}$-Matrizen $\mathbf{A}_{\mathcal{H}}$ üblicherweise erfüllt, sofern die ursprüngliche Matrix $\mathbf{A}$ diese erfüllt.

Lemma 4.19. *Sei $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ und die LU-Zerlegung von $\mathbf{A}$ existiere. Falls der Fehler $\|\mathbf{A}-\mathbf{B}\|_2$ hinreichend klein ist, besitzt die Matrix $\mathbf{B}$ ebenfalls eine LU-Zerlegung.*

Beweis. Die Determinante hängt stetig von den Matrixeinträgen ab und invertierbare Matrizen $\mathbf{M} \in \mathbb{R}^{n\times n}$ sind charakterisiert durch $\det(\mathbf{M}) \neq 0$. Also formen die Matrizen

$$\{\mathbf{M} \in \mathbb{R}^{n\times n} \ : \ \mathbf{M} \text{ regulär}\} = \{\mathbf{M} \in \mathbb{R}^{n\times n} \ : \ \det(\mathbf{M}) \neq 0\}$$

eine offene Teilmenge von $\mathbb{R}^{n\times n}$. Somit existiert für jede erste Submatrix $\mathbf{A}_m \in \mathbb{R}^{m\times m}$ von $\mathbf{A}$ ein $\varepsilon_m > 0$ so dass die Submatrix $\mathbf{B}_m$ von $\mathbf{B}$ invertierbar ist sofern $\|\mathbf{A}_m - \mathbf{B}_m\|_2 \leq \varepsilon_m$. Falls $\|\mathbf{A}-\mathbf{B}\|_2 \leq \min_{m=1,\dots,n} \varepsilon_m$, folgt somit das gewünschte Resultat. □

Ähnlich zur rekursiven $\mathcal{H}$-Inversion kann auch rekursiv eine $\mathcal{H}$-LU-Zerlegung bestimmt werden. Sei $\mathbf{A} \in \mathbb{R}^{n\times n}$ und es existiere die LU-Zerlegung von $\mathbf{A}$. Wir machen den Ansatz

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ 0 & \mathbf{U}_{22} \end{pmatrix}, \tag{4.21}$$

was auf folgende Rechenschritte zur Bestimmung der $\mathcal{H}$-LU-Zerlegung führt:

- Berechne $\mathbf{L}_{11}$ und $\mathbf{U}_{11}$ aus $\mathbf{A}_{11} = \mathbf{L}_{11}\mathbf{U}_{11}$.
- Berechne $\mathbf{U}_{12}$ aus $\mathbf{A}_{12} = \mathbf{L}_{11}\mathbf{U}_{12}$.
- Berechne $\mathbf{L}_{21}$ aus $\mathbf{A}_{21} = \mathbf{L}_{21}\mathbf{U}_{11}$.
- Berechne $\mathbf{L}_{22}$ und $\mathbf{U}_{22}$ aus $\mathbf{A}_{22} = \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22}$.

Beginnt man mit einer $\mathcal{H}$-Matrix $\mathbf{A}_{\mathcal{H}}$ und ersetzt man abermals die arithmetischen Operationen durch $\mathcal{H}$-Arithmetik, so erhält man eine LU-Zerlegung im $\mathcal{H}$-Matrix-Format.

4.6 Rekompression von $\mathcal{H}$-Matrizen

Im Folgenden wollen wir das $\mathcal{H}$-Matrix-Format weiter ausdünnen, also weitere Informationen weglassen ohne den Fehler zu vergrößern. Wir haben bisher stets Best-Approximationen auf einen festen, vorgegebenen Rang r betrachtet. Kontrolliert man hingegen bei der Projektion nicht den Rang, sondern den Fehler, so kann adaptiv jener Rang *a-posteriori* bestimmt werden, der tatsächlich für die gewünschte Genauigkeit benötigt wird.

Betrachtet man beispielsweise $\mathcal{H}$-Matrizen mittels Interpolation, so zeigen die numerischen Beispiele aus Abschnitt 3, dass im Vergleich zu anderen Methoden (ACA,HCA) ein deutlich größerer Rang als benötigt verwendet wird. Mit Hilfe der nachfolgenden Rekompressionstechniken kann dieser a-posteriori (nach dem Assemblieren/oder auch on-the-fly) wieder geeignet reduziert werden.

Sei also eine $\mathcal{H}$-Matrix $\mathbf{A} \in \mathcal{H}(r, \mathbb{P}_{\mathcal{I}\mathcal{J}})$ mit einer Adm-zulässigen Partition $\mathbb{P}_{\mathcal{I}\mathcal{J}}$ basierend auf einem Cluster-Baum $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$ gegeben. Sei $\delta > 0$. Wir suchen eine Matrix $\mathbf{A}^{\mathrm{comp}} \in \mathcal{H}(r^{\mathrm{comp}}, \mathbb{P}^{\mathrm{comp}}_{\mathcal{I}\mathcal{J}})$ mit den folgenden Eigenschaften

- $\mathbf{A}^{\mathrm{comp}}$ benötigt weniger Speicher und erlaubt schnellere Berechnungen.
- $\mathbb{P}^{\mathrm{comp}}_{\mathcal{I}\mathcal{J}}$ ist eine Partition basierend auf einer schwächeren Zulässigkeitsbedingung $\mathrm{Adm}^{\mathrm{comp}}$, also

  $$\tau \times \sigma \text{ ist Adm-zulässig} \Rightarrow \tau \times \sigma \text{ ist } \mathrm{Adm}^{\mathrm{comp}}\text{-zulässig}.$$

  Wir haben somit potentiell mehr und größere zulässige Blöcke.
- r^{comp} ist für jeden Block separat so klein wie möglich gewählt.
- Der relative Fehler ist beschränkt durch

  $$\frac{\|\mathbf{A} - \mathbf{A}^{\mathrm{comp}}\|_2}{\|\mathbf{A}\|_2} \leq \delta.$$

Die Rekompression teilt sich in 3 Schritte auf:

- Rekompression zulässiger Blöcke: Es genügt eventuell ein Rang $r' < r$ für ähnliche Genauigkeit, also bestimme $\mathcal{T}_{r'}(\mathbf{A}_{\tau\times\sigma})$.
- Rekompression nicht-zulässiger Blöcke: Die Zulässigkeitsbedingung ist nur ein hinreichendes Kriterium, es sind also im Allgemeinen mehr nicht-zulässige Blöcke vorhanden als notwendig. Eventuell können also nicht-zulässige Blöcke in zulässige umgewandelt werden.
- Rekompression der Blockpartition: Die Partition $\mathbb{P}_{\mathcal{I}\mathcal{J}}$ beinhaltet eventuell zu viele Blöcke bzw. die Tiefe ist größer als notwendig. Die Blockstruktur kann eventuell vergröbert werden.

4.6.1 Blockweise Rekompression

Wir beginnen mit einer blockweisen Rekompression, wobei das nachfolgende Lemma die zentrale Beobachtung für die Rekompressionsstrategie liefert.

Lemma 4.20. *Sei $\mathbf{A} \in \mathcal{H}(r)$ und $\varepsilon > 0$. Definiere eine Matrix $\widetilde{\mathbf{A}} \in \mathcal{H}(r)$ wie folgt: Für jeden Block $\tau \times \sigma \in \mathbb{P}$ und $k := \mathrm{rang}(\mathbf{A}|_{\tau\times\sigma})$, suchen wir $r' \in \{1, ..., k-1\}$ mit*

$$\sigma_{r'+1}(\mathbf{A}|_{\tau\times\sigma}) \leq \varepsilon\, \sigma_1(\mathbf{A}|_{\tau\times\sigma}), \tag{4.22}$$

und setzen $r' := k$ sofern kein derartiges r' existiert. Wir definieren dann

$$\widetilde{\mathbf{A}}|_{\tau\times\sigma} := \mathcal{T}_{r'}(\mathbf{A}|_{\tau\times\sigma}).$$

Dann ist der relative Fehler beschränkt durch

$$\frac{\left\|\mathbf{A}-\widetilde{\mathbf{A}}\right\|_2}{\|\mathbf{A}\|_2} \leq C_{\mathrm{sp}}(\mathrm{depth}(\mathbb{P})+1)\,\varepsilon \tag{4.23}$$

Beweis. Der Rang r' kann für jeden Block $\tau\times\sigma\in\mathbb{P}$ verschieden sein, also $r' = r'(\tau,\sigma)$. Der Einfachheit halber lassen wir die Abhängigkeit von $\tau\times\sigma$ weg. Wir haben $\sigma_{k+1}(\mathbf{A}|_{\tau\times\sigma}) = 0$ für $k = \mathrm{rang}(\mathbf{A}|_{\tau\times\sigma})$. Die blockweise Abschätzung der Spektralnorm aus Lemma 3.9 liefert mit (4.22)

$$\begin{aligned}
\left\|\mathbf{A}-\widetilde{\mathbf{A}}\right\|_2 &\leq C_{\mathrm{sp}} \sum_{\ell=0}^{\mathrm{depth}(\mathbb{P})} \max_{\substack{\tau\times\sigma\in\mathbb{P}\\ \mathrm{level}(\tau)=\ell}} \left\|\mathbf{A}|_{\tau\times\sigma}-\widetilde{\mathbf{A}}|_{\tau\times\sigma}\right\|_2 \\
&= C_{\mathrm{sp}} \sum_{\ell=0}^{\mathrm{depth}(\mathbb{P})} \max_{\substack{\tau\times\sigma\in\mathbb{P}_{\mathrm{far}}\\ \mathrm{level}(\tau)=\ell}} \|\mathbf{A}|_{\tau\times\sigma}-\mathcal{T}_{r'}(\mathbf{A}|_{\tau\times\sigma})\|_2 \\
&= C_{\mathrm{sp}} \sum_{\ell=0}^{\mathrm{depth}(\mathbb{P})} \max_{\substack{\tau\times\sigma\in\mathbb{P}_{\mathrm{far}}\\ \mathrm{level}(\tau)=\ell}} \sigma_{r'+1}(\mathbf{A}|_{\tau\times\sigma}) \\
&\leq C_{\mathrm{sp}}(\mathrm{depth}(\mathbb{P})+1)\,\varepsilon\,\|\mathbf{A}\|_2\,,
\end{aligned} \tag{4.24}$$

wobei wir $\sigma_1(\mathbf{A}|_{\tau\times\sigma}) = \|\mathbf{A}|_{\tau\times\sigma}\|_2 \leq \|\mathbf{A}\|_2$ verwendet haben. □

Lemma 4.20 liefert direkt einen Algorithmus zur Rekompression von Fernfeld-Blöcken.

Algorithmus 4.21 (Rekompression von Fernfeldblöcken).

`function CompressPfar(var` $\mathbf{A}^{\mathrm{comp}}$`,`$\mathbf{A}$`,`$\tau$`,`$\sigma$`,`$\varepsilon$`)`
Berechne SVD von $\mathbf{A}|_{\tau\times\sigma}$
Suche minimales $r'\in\{1,\ldots,r-1\}$ mit (4.22).
`if` r' existiert
  Definiere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathcal{T}_{r'}(\mathbf{A}|_{\tau\times\sigma})$ in faktorisierter Form.
`else`
  Definiere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathbf{A}|_{\tau\times\sigma}$.
`end`

Der Aufwand von Algorithmus 4.21 ist mit Lemma 4.1/Proposition 4.3 beschränkt durch

$$6r^2(|\tau|+|\sigma|) + 23r^3 + r'|\tau| + r, \tag{4.25}$$

da maximal r zusätzliche Operationen für das Bestimmen von r' benötigt werden.

Für Nahfeld-Blöcke verwenden wir einen ähnlichen Ansatz. Allerdings benötigen wir eine Faktorisierung für die effiziente Berechnung der SVD bei zu großen Blöcken. Im Folgenden geschieht dies mit der QR-Zerlegung (SVD direkt hätte kubische Komplexität).

Algorithmus 4.22 (Rekompression von Nahfeld-Blöcken).

`function CompressPnear(var` $\mathbf{A}^{\mathrm{comp}}$`,`$\mathbf{A}$`,`$\tau$`,`$\sigma$`,`$\varepsilon$`)`
`if` $|\tau| \leq n_{\mathrm{blatt}}$ `und` $|\sigma| \leq n_{\mathrm{blatt}}$
  Berechne SVD von $\mathbf{A}|_{\tau\times\sigma}$ direkt.
`else`
  `if` $|\sigma| \leq |\tau|$ `/*` somit $|\sigma| \leq n_{\mathrm{blatt}} < |\tau|$ `*/`
    Berechne reduzierte QR-Zerlegung $\mathbf{A}|_{\tau\times\sigma} = \mathbf{QR}$ mit $\mathbf{Q} \in \mathbb{R}^{\tau\times\sigma}$ und $\mathbf{R} \in \mathbb{R}^{\sigma\times\sigma}$.
    Definiere $\mathbf{V} := \mathbf{Q}$ und $\mathbf{W} := \mathbf{R}^T$.
    Berechne SVD $\mathbf{A}|_{\tau\times\sigma} = \mathbf{VW}^T$.
  `else /*` also $|\tau| \leq n_{\mathrm{blatt}} < |\sigma|$ `*/`
    Berechne reduzierte QR-Zerlegung $\mathbf{A}|_{\tau\times\sigma}^T = \mathbf{QR}$ mit $\mathbf{Q} \in \mathbb{R}^{\sigma\times\tau}$ und $\mathbf{R} \in \mathbb{R}^{\tau\times\tau}$.
    Definiere $\mathbf{V} := \mathbf{R}^T$ und $\mathbf{W} := \mathbf{Q}$.
    Berechne SVD von $\mathbf{A}|_{\tau\times\sigma} = \mathbf{VW}^T$.
  `end`
`end`
Suche minimales $r' \in \{1, \ldots, r\}$ mit (4.22).
`if` r' existiert
  Definiere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathcal{T}_{r'}(\mathbf{A}|_{\tau\times\sigma})$ in faktorisierter Form.
`else`
  `if` $\operatorname{rang}(\mathbf{A}|_{\tau\times\sigma}) \leq r$ `/*` also $\operatorname{rang}(\mathbf{A}|_{\tau\times\sigma}) = r$ `*/`
    Definiere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathbf{A}|_{\tau\times\sigma}$ aber in faktorisierter Form.
  `else`
    Definiere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathbf{A}|_{\tau\times\sigma}$.
  `end`
`end`

Der Aufwand von Algorithmus 4.22 kann mit ähnlichen Überlegungen wie für Proposition 4.3 beschränkt werden durch

$$11 \max\{n_{\mathrm{blatt}}^2, r\}(|\tau| + |\sigma|) + n_{\mathrm{blatt}}^2 + 23 n_{\mathrm{blatt}}^3 + r \tag{4.26}$$

Operationen.

Kombiniert man die Kompressionstechniken für Nah- und Fernfeld, so kann man Block für Block die Matrix $\mathbf{A}^{\mathrm{comp}}$ bestimmen mit relativem Fehler (4.23) und Aufwand von Ordnung $\mathcal{O}(r^3 N + r^2 N \log N)$, wobei $N = n + m$.

Bemerkung 4.23. *Lemma 4.20 suggeriert die Wahl* $\varepsilon = \left[C_{\mathrm{sp}}(\operatorname{depth}(\mathbb{P}) + 1)\right]^{-1} \delta$*, sodass*

der Rekompressionsfehler kleiner δ ist. Numerische Experimente zeigen, dass eine Wahl $\varepsilon = \delta$ ebenfalls genügt.

Für eine Kontrolle des relativen Fehlers in der Frobenius-Norm kann man Kriterium (4.22) *durch*

$$\sum_{j=r'+1}^{\operatorname{rang}(\mathbf{A}|_{\tau\times\sigma})} \sigma_j(\mathbf{A}|_{\tau\times\sigma})^2 \leq \varepsilon^2 \left\|\mathbf{A}|_{\tau\times\sigma}\right\|_F^2 \tag{4.27}$$

ersetzen.

4.6.2 Vergröberung der Block-Partition

Wir erinnern an die Vergröberungsstrategie von Bemerkung 4.15, also 4 zulässige Sohnblöcke mit Rang kleiner r werden zusammengefasst mittels

$$\mathbf{V} := \begin{pmatrix} \mathbf{V}_{11} & 0 & \mathbf{V}_{12} & 0 \\ 0 & \mathbf{V}_{21} & 0 & \mathbf{V}_{22} \end{pmatrix} \in \mathbb{R}^{\tau\times 4r} \quad \text{und} \quad \mathbf{W} := \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{21} & 0 & 0 \\ 0 & 0 & \mathbf{W}_{12} & \mathbf{W}_{22} \end{pmatrix} \in \mathbb{R}^{\sigma\times 4r},$$

dann gilt

$$\mathbf{V}\mathbf{W}^T = \begin{pmatrix} \mathbf{V}_{11}\mathbf{W}_{11}^T & \mathbf{V}_{12}\mathbf{W}_{12}^T \\ \mathbf{V}_{21}\mathbf{W}_{21}^T & \mathbf{V}_{12}\mathbf{W}_{22}^T \end{pmatrix} = \begin{pmatrix} \mathbf{A}|_{\tau_1\times\sigma_1} & \mathbf{A}|_{\tau_1\times\sigma_2} \\ \mathbf{A}|_{\tau_2\times\sigma_1} & \mathbf{A}|_{\tau_2\times\sigma_2} \end{pmatrix} = \mathbf{A}|_{\tau\times\sigma}.$$

Hierbei hat der neue Vaterblock einen Rang von maximal $4r$.

Die Überlegung ist also einen Block zu vergröbern, sofern der Speicherbedarf für die 4 Söhne größer ist als der für den mittels Kriterium (4.22) rekomprimierten Vaterblock.

Für stufentreue Produkt-Cluster-Bäume und sofern jeder Sohn-Block festen Rang r hat, reduziert sich dies darauf, ob ein r' mit $r' < 2r$ gefunden werden kann, das Kriterium (4.22) erfüllt. Wir verlangen im Nachfolgenden die stärkere Bedingung $r' < r := \max\{r(\tau' \times \sigma') : \tau' \times \sigma' \in \operatorname{sons}(\tau \times \sigma)\}$.

Bei der (rekursiven) Implementierung der Vergröberung müssen folgende Schritte abgearbeitet werden:

- Überprüfe ob alle Söhne zulässig sind.
- Berechne SVD von $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma}$.
- Suche minimales $r' \in \{1, \ldots, r\}$ sodass (4.22) für $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma}$ gilt.
- Falls r' existiert, entferne alle Söhne $\tau' \times \sigma' \in \operatorname{sons}(\tau \times \sigma)$ und lösche $\mathbf{A}^{\mathrm{comp}}|_{\tau'\times\sigma'}$.
- Füge $\tau \times \sigma$ zu $\mathbb{P}_{\mathrm{far}}$ hinzu und speichere $\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma} := \mathcal{T}_{r'}(\mathbf{A}^{\mathrm{comp}}|_{\tau\times\sigma})$ in faktorisierter Form.

Proposition 4.24. *Sei $\mathbf{A} \in \mathcal{H}(r, \mathbb{P})$ und $\varepsilon > 0$. Die mit derartiger Vergröberung berechnete Matrix $\mathbf{A}^{\mathrm{comp}} \in \mathcal{H}(r, \mathbb{P}^{\mathrm{comp}})$ erfüllt*

$$\frac{\|\mathbf{A} - \mathbf{A}^{\mathrm{comp}}\|_2}{\|\mathbf{A}\|_2} \leq C_{\mathrm{sp}} \operatorname{depth}(\mathbb{P})(\operatorname{depth}(\mathbb{P}) + 1)\, \varepsilon. \tag{4.28}$$

und kann in weniger als

$$\mathcal{O}(r^2 N \log^2 N + r^3 N), \qquad N := n + m$$

Operationen berechnet werden.

Beweis. Wir schreiben die rekursive Vergröberung induktiv auf.

- Definiere $(\mathbf{A}^{(0)}, \mathbb{P}^{(0)}) := (\mathbf{A}, \mathbb{P})$.
- Für $j = 1, \ldots, \operatorname{depth}(\mathbb{P})$ und gegebene $(\mathbf{A}^{(j-1)}, \mathbb{P}^{(j-1)})$, erhält man $(\mathbf{A}^{(j)}, \mathbb{P}^{(j)})$ aus $(\mathbf{A}^{(j-1)}, \mathbb{P}^{(j-1)})$ indem man in dem Schritt nur Blöcke $\tau \times \sigma \in \mathbb{T}_{\mathcal{I}} \times \mathbb{T}_{\mathcal{J}}$ mit $\operatorname{level}(\tau \times \sigma) = \operatorname{depth}(\mathbb{P}) - j$ zur Vergröberung zulässt.

Läuft man den gesamten Baum durch, so folgt schließlich $\mathbf{A}^{\mathrm{comp}} = \mathbf{A}^{(\operatorname{depth}(\mathbb{P}))}$. Da $\mathbb{P}^{(j)}$ Vergröberungen von $\mathbb{P}$ sind, sind die zugehörigen Schwachbesetztheitskonstanten beschränkt durch C_{sp}. Mittels Dreiecksungleichung und Lemma 3.9 folgt somit

$$\begin{aligned}
\|\mathbf{A} - \mathbf{A}^{\mathrm{comp}}\|_2 &\leq \sum_{j=1}^{\operatorname{depth}(\mathbb{P})} \left\|\mathbf{A}^{(j-1)} - \mathbf{A}^{(j)}\right\|_2 \\
&\leq C_{\mathrm{sp}} \sum_{j=1}^{\operatorname{depth}(\mathbb{P})} \sum_{\ell=0}^{\operatorname{depth}(\mathbb{P}^{(j)})} \max_{\substack{\tau\times\sigma\in\mathbb{P}^{(j)} \\ \operatorname{level}(\tau\times\sigma)=\ell}} \left\|\mathbf{A}^{(j-1)}|_{\tau\times\sigma} - \mathbf{A}^{(j)}|_{\tau\times\sigma}\right\|_2 .
\end{aligned}$$

Mittels Definition von $\mathbf{A}^{(j)}$ folgt also sofern vergröbert wird

$$\begin{aligned}
\left\|\mathbf{A}^{(j-1)}|_{\tau\times\sigma} - \mathbf{A}^{(j)}|_{\tau\times\sigma}\right\|_2 \leq \left\|\mathbf{A}^{(j-1)}|_{\tau\times\sigma} - \mathcal{T}_{r'}(\mathbf{A}^{(j-1)}|_{\tau\times\sigma})\right\|_2 &= \sigma_{r'+1}(\mathbf{A}^{(j-1)}|_{\tau\times\sigma}) \\
&\leq \varepsilon\sigma_1(\mathbf{A}^{(j-1)}|_{\tau\times\sigma}).
\end{aligned}$$

Mit

$$\sigma_1(\mathbf{A}^{(j-1)}|_{\tau\times\sigma}) = \sigma_1(\mathbf{A}|_{\tau\times\sigma}) = \|\mathbf{A}|_{\tau\times\sigma}\|_2 \leq \|\mathbf{A}\|_2 ,$$

folgt die gewünschte Abschätzung aus $\operatorname{depth}(\mathbb{P}^{(j)}) \leq \operatorname{depth}(\mathbb{P})$.

Die Bestimmung des Rechenaufwands erhält man bei Betrachtung eines Schrittes von $\mathbf{A}^{(j-1)}$ zu $\mathbf{A}^{(j)}$ und den üblichen Überlegungen (Aufwand der SVD, Berechnung von r', Summation über Blöcke). □

Bemerkung 4.25. *Die Fehlerabschätzung in Proposition 4.24 ist abermals pessimistisch. Numerische Experimente zeigen*

$$\frac{\|\mathbf{A} - \mathbf{A}^{\mathrm{comp}}\|_2}{\|\mathbf{A}\|_2} \leq \delta$$

mit $\varepsilon = \delta$ anstelle von $\varepsilon = \left(C_{\mathrm{sp}} \operatorname{depth}(\mathbb{P})(\operatorname{depth}(\mathbb{P}) + 1)\right)^{-1} \delta$.

4.6.3 Gesamte On-The-Fly-Rekompression

Die Rekompression inklusive Vergröberung kann auch direkt während der Assemblierung durchgeführt werden.

Algorithmus 4.26 (On-The-Fly-Rekompression).

```
function BuildCompressedHMatrix(var A,τ,σ,ε1,ε2)
if τ × σ zulässig
  Berechne A|τ×σ = VW^T in faktorisierter Form.
  rufe CompressPfar(A,A,τ,σ,ε1) auf
elseif sons(τ × σ) ≠ ∅
  for all τ' × σ' ∈ sons(τ) × sons(σ)
    rufe BuildCompressedHMatrix(A,τ',σ',ε1,ε2) auf
    if τ' × σ' ∉ P_far
      return
    end
  end
  Berechne SVD von A|τ×σ.
  Suche minimales r' ∈ {1,...,r} so dass (4.22) gilt mit ε = ε2.
  if r' existiert
    Füge τ × σ zu P_far hinzu und speichere A|τ×σ := T_r'(A|τ×σ) faktorisiert.
    for all τ' × σ' ∈ sons(τ × σ)
      Entferne τ' × σ' aus P_far und lösche die Sohn-Blöcke A|τ'×σ'.
  end
else /* nicht-zulässiger Block der nicht weiter zerteilt werden kann */
  Berechne vollbesetzten Block A|τ×σ.
  rufe CompressPnear(A,A,τ,σ,ε1) auf
end
```

Der Aufwand des Algorithmus setzt sich einfach aus den einzelnen Teilen zusammen, also blockweiser Kompression und Vergröberung, welche zuvor bereits analysiert worden sind. Für den Fehler gilt

$$\frac{\|\mathbf{A} - \mathbf{A}^{\mathrm{comp}}\|_2}{\|\mathbf{A}\|_2} \leq C_{\mathrm{sp}}(\mathrm{depth}(\mathbb{P}) + 1)\Big(\varepsilon_1 + \mathrm{depth}(\mathbb{P})\varepsilon_2\Big) \tag{4.29}$$

mit Dreiecksungleichung, Lemma 4.20 und Proposition 4.24.

Bemerkung 4.27. *Wir haben bei der $\mathcal{H}$-Addition gesehen, dass der Fehler bei $\mathbf{A} \oplus_r \mathbf{B}$ beliebig schlecht sein kann. Die angesprochene* ***adaptive $\mathcal{H}$-Addition*** *berechnet einfach eine Rekompression der Summe $\mathbf{A} + \mathbf{B} \in \mathcal{H}(2r)$, also eine Matrix $\mathbf{C} \in \mathcal{H}(2r)$ mit*

$$\frac{\|\mathbf{A} + \mathbf{B} - \mathbf{C}\|_2}{\|\mathbf{A} + \mathbf{B}\|_2} \leq \varepsilon.$$

Wir betrachten abermals das zweite Beispiel aus Abschnitt 3.4 (Integraloperator K auf Kurbelwelle). Für festes $\varepsilon = 10^{-5}$ berechnen wir die rekomprimierte $\mathcal{H}$-Matrix mit Interpolation on-the-fly mit obigem Algorithmus. Die folgenden Tabellen vergleichen die rekomprimierte Matrix mit der zunächst berechneten Matrix.

$\mathbf{A}_{\mathcal{H}}$ mit Interpolation

k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
2	8	0.0893	272.82	4.79
3	27	0.0189	834.89	10.43
4	64	0.0038	1929.44	21.76
5	125	9.58e-04	3733.97	41.74
6	216	2.37e-04	6425.98	76.15
7	343	6.57e-05	10182.95	129.42
8	512	1.78e-05	15182.39	224.10

$\mathbf{A}_{\mathcal{H}}$ rekomprimiert

k	Rang	rel. Fehler $\|\cdot\|_2$	Speicher (MB)	Zeit (s)
2	36	0.0843	204.09	19.77
3	57	0.0196	284.64	42.89
4	91	0.0040	259.98	72.32
5	72	9.08e-04	239.65	99.84
6	66	2.36e-04	233.97	142.66
7	68	6.61e-05	233.07	186.39
8	69	2.11e-06	232.93	242.97

Man erkennt, dass der Speicherbedarf enorm reduziert wird, da beim Assemblieren mit Interpolation (wie erwartet) viel redundante Information hinzugefügt wird. Der Sprung im Rang kann mit der Vergröberung der Blockstruktur erklärt werden (HLib vergleicht $r' < 2r$). Allerdings steigen die Rechenzeiten.

5 $\mathcal{H}^2$-Matrizen

Das Ziel dieses Abschnittes ist das Format der $\mathcal{H}$-Matrizen mit einer weiteren Hierarchie zu versehen und somit den Aufwand weiter zu reduzieren. Tatsächlich kann der Logarithmus, der üblicherweise durch die Baumtiefe in die Komplexitätsabschätzungen eingeht, vermieden werden.

5.1 Motivation und uniforme $\mathcal{H}$-Matrizen

Wir betrachten abermals Matrizen $\mathbf{A} \in \mathbb{R}^{n\times m}$ der Form (A3), also

$$\mathbf{A}_{ij} = \int_\Omega \int_\Omega \phi_i(x)\kappa(x,y)\psi_j(y)\,dy\,dx \quad \text{für } i = 1,\dots,n \text{ und } j = 1,\dots,m \tag{A3}$$

mit einer asymptotisch glatten Funktion κ.

Im Gegensatz zu Abschnitt 3.2.4 interpolieren wir wie in Abschnitt 3.4 in beiden Variablen, wir betrachten also

$$\widetilde{\kappa}_k^{\tau\sigma}(x,y) := \mathcal{I}_k^\tau \mathcal{I}_k^\sigma \kappa(x,y) = \sum_{j=1}^{k^d}\sum_{\ell=1}^{k^d} \mathbf{L}_j^\tau(x)\kappa(x_j^\tau, y_\ell^\sigma)\mathbf{L}_\ell^\sigma(y)$$

mit tensoriellen Chebyshev Knoten $x_j^\tau \in B_\tau$, $j = 1,\dots,k^d$ sowie y_ℓ^σ, $\ell = 1,\dots,k^d$ und zugehörigen Lagrange-Polynomen $\mathbf{L}_j^\tau, \mathbf{L}_\ell^\sigma$.

Bemerkung 5.1. *Verlangt man die verschärfte Zulässigkeitsbedingung*

$$\max\{\operatorname{diam}(B_\tau), \operatorname{diam}(B_\sigma)\} \le \eta \operatorname{dist}(B_\tau, B_\sigma), \tag{5.1}$$

so folgt analog zu Korollar 3.24 mit Dreiecksungleichung

$$\|\kappa - \widetilde{\kappa}_k^{\tau\sigma}\|_{\infty, B_\tau\times B_\sigma} \le C \operatorname{dist}(B_\tau, B_\sigma)^{-s} \Lambda_k^{2d} \left(1 + \frac{\eta}{c_2}\right) k \left(1 + \frac{2c_2}{\eta}\right)^{-k}$$

mit dem Faktor Λ_k^{2d} anstelle von Λ_k^d.

Die Matrix $\mathbf{S}^{\tau\sigma} \in \mathbb{R}^{k^d\times k^d}$ beinhaltet die Auswertungen

$$\mathbf{S}_{j\ell}^{\tau\sigma} := \kappa(x_j^\tau, y_\ell^\sigma)$$

und auf jedem Block $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ ist die Approximation mittels Interpolation

$$\mathbf{A}|_{\tau\times\sigma} = \mathbf{V}^\tau \mathbf{S}^{\tau\sigma} (\mathbf{W}^\sigma)^T,$$

wobei $\mathbf{V}^\tau \in \mathbb{R}^{\tau \times k^d}$, $\mathbf{W}^\sigma \in \mathbb{R}^{\sigma \times k^d}$ gegeben sind durch

$$(\mathbf{V}^\tau)_{i\ell} := \int_\Omega \mathbf{L}^\tau_\ell(x)\phi_i(x)\,dx \quad \text{und} \quad (\mathbf{W}^\sigma)_{j\ell} := \int_\Omega \mathbf{L}^\sigma_\ell(y)\psi_j(y)\,dy.$$

Die Matrizen $\mathbf{V}^\tau$, $\mathbf{W}^\sigma$ hängen nicht von der Funktion κ ab, sondern nur von den Lagrange Polynomen und den gegebenen Basisfunktionen. Somit können diese auch exakt (ohne Quadraturfehler!) mittels Gauß-Quadratur berechnet werden.

Diese Beobachtung, dass $\mathbf{V}^\tau$, $\mathbf{W}^\sigma$ nicht von κ abhängen, motiviert die folgende Definition.

Definition 5.2. *Sei $\mathbb{P}$ eine zulässige Partition basierend auf dem Block-Clusterbaum $\mathbb{T}_{\mathcal{I}\times\mathcal{J}}$. Für alle Cluster $\tau \in \mathbb{T}_\mathcal{I}$ und $\sigma \in \mathbb{T}_\mathcal{J}$ seien Matrizen $\mathbf{V}^\tau \in \mathbb{R}^{\tau\times r}$, $\mathbf{W}^\sigma \in \mathbb{R}^{\sigma\times r}$ gegeben.*

*Ein Matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ heißt **uniforme $\mathcal{H}$-Matrix**, falls für alle $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ eine **Kopplungs-Matrix** $\mathbf{S}^{\tau\sigma} \in \mathbb{R}^{r\times r}$ existiert mit*

$$\mathbf{A}|_{\tau\times\sigma} = \mathbf{V}^\tau \mathbf{S}^{\tau\sigma} (\mathbf{W}^\sigma)^T. \tag{5.2}$$

*Die Familie von Matrizen $V = (\mathbf{V}^\tau)_{\tau\in\mathbb{T}_\mathcal{I}}$, $W = (\mathbf{W}^\sigma)_{\sigma\in\mathbb{T}_\mathcal{J}}$ heißen **Cluster-Basen**. Wir schreiben $\mathcal{H}(r, \mathbb{P}, V, W)$ für die Menge aller uniformen $\mathcal{H}$-Matrizen mit Cluster-Basen V, W.*

Im Gegensatz zu der Menge $\mathcal{H}(r, \mathbb{P})$ ist die Menge aller uniformen $\mathcal{H}$-Matrizen mit fixen Cluster-Basen $\mathcal{H}(r, \mathbb{P}, V, W)$ ein Untervektorraum von $\mathbb{R}^{n\times m}$, da

$$(\mathbf{A} + \mathbf{B})|_{\tau\times\sigma} = \mathbf{V}^\tau(\mathbf{S}^{\tau\sigma}_\mathbf{A} + \mathbf{S}^{\tau\sigma}_\mathbf{B})(\mathbf{W}^\sigma)^T.$$

Wir benötigen also für die Addition keine Kürzung mittels SVD sondern können diese exakt durchführen.

Wir betrachten abermals die Lagrange-Polynome. Die Projektionseigenschaft der Interpolation führt auf eine **zweite Hierarchie:** Sei $\tau' \in \text{sons}(\tau)$, dann gilt

$$\mathbf{L}^\tau_\ell(x) = \sum_{\nu=1}^{k^d} \mathbf{L}^\tau_\ell(x^{\tau'}_\nu)\mathbf{L}^{\tau'}_\nu(x). \tag{5.3}$$

Setzt man das in die Definition von $\mathbf{V}^\tau$ ein, so folgt für $i \in \tau' \in \text{sons}(\tau)$

$$\mathbf{V}^\tau_{i\ell} = \int_\Omega \phi_i(x)\mathbf{L}^\tau_\ell(x)\,dx = \sum_{\nu=1}^{k^d} \mathbf{L}^\tau_\ell(x^{\tau'}_\nu)\mathbf{V}^{\tau'}_{i\nu}.$$

Die Matrix $\mathbf{V}^\tau$ kann also blockweise geschrieben werden als

$$\mathbf{V}^\tau|_{\tau'} = \mathbf{V}^{\tau'}\mathbf{T}^{\tau'\tau}, \tag{5.4}$$

mit so genannten **Transfer-Matrizen** $\mathbf{T}^{\tau'\tau} \in \mathbb{R}^{k^d \times k^d}$ definiert durch

$$\mathbf{T}^{\tau'\tau}_{\nu\ell} := \mathbf{L}^\tau_\ell(x^{\tau'}_\nu). \tag{5.5}$$

Der Übergang von Vatercluster zu Sohncluster wird also nur durch die Transfermatrizen beschrieben. Selbes gilt für $\mathbf{W}^\sigma$, also $\mathbf{W}^\sigma|_{\sigma'} = \mathbf{W}^{\sigma'}\mathbf{T}^{\sigma'\sigma}$

Der zentrale Vorteil hierbei ist, dass nur die Matrizen $\mathbf{V}^\tau, \mathbf{W}^\sigma$ für Blätter im Cluster-Baum gespeichert werden müssen sowie die (kleinen) Transfermatrizen. Diese Beobachtung führt zu einer deutlichen Reduktion im Speicherbedarf und zu der Definition der $\mathcal{H}^2$-Matrizen.

5.2 $\mathcal{H}^2$-Matrizen und deren Komplexität

Definition 5.3. *Eine uniforme $\mathcal{H}$-Matrix $\mathbf{A} \in \mathcal{H}(r, \mathbb{P}, V, W)$ heißt **$\mathcal{H}^2$-Matrix**, falls für alle $\tau \in \mathbb{T}_\mathcal{I}$, $\sigma \in \mathbb{T}_\mathcal{J}$ **Transfer-Matrizen** $\mathbf{T}^{\tau'\tau}_V \in \mathbb{R}^{r\times r}$ und $\mathbf{T}^{\sigma'\sigma}_W \in \mathbb{R}^{r\times r}$ existieren, so dass*

$$\mathbf{V}^\tau|_{\tau'} = \mathbf{V}^{\tau'}\mathbf{T}^{\tau'\tau}_V \quad \text{für alle } \tau \in \mathbb{T}_\mathcal{I}, \tau' \in \operatorname{sons}(\tau) \tag{5.6}$$

sowie

$$\mathbf{W}^\sigma|_{\sigma'} = \mathbf{W}^{\sigma'}\mathbf{T}^{\sigma'\sigma}_W \quad \text{für alle } \sigma \in \mathbb{T}_\mathcal{J}, \sigma' \in \operatorname{sons}(\sigma). \tag{5.7}$$

*Gilt diese Voraussetzung, so heißen die Cluster-Basen V, W **geschachtelt**. Wir schreiben $\mathcal{H}^2(r, \mathbb{P}, V, W)$ für die Menge aller $\mathcal{H}^2$-Matrizen mit maximalem blockweisem Rang r zur Partition $\mathbb{P}$ mit geschachtelten Cluster-Basen V, W.*

Uniforme $\mathcal{H}$-Matrizen und $\mathcal{H}^2$-Matrizen sind klarerweise auch $\mathcal{H}$-Matrizen. Bei der Speicherung von $\mathcal{H}^2$-Matrizen werden allerdings nur die Nahfeld-Blöcke und auf dem Fernfeld die Kopplungsmatrizen $\mathbf{S}^{\tau\sigma}$, die Cluster-Basen $\mathbf{V}^\tau$, $\mathbf{W}^\sigma$ auf den Blättern sowie die Transfermatrizen $\mathbf{T}^{\tau'\tau}_V$, $\mathbf{T}^{\sigma'\sigma}_W$ benötigt.

Diese Beobachtung führt auf lineare Komplexität des Speicherbedarfs. Zusätzlich verlangen wir, damit auch das Nahfeld lineare Komplexität hat, dass für Nahfeldblöcke $\tau \times \sigma \in \mathbb{P}_{\text{near}}$ folgt, dass $|\tau|, |\sigma| \leq n_{\text{blatt}}$ (vgl. auch die verschärfte Zulässigkeitsbedingung im vorigen Abschnitt).

Proposition 5.4. *Sei $\mathbb{P}$ eine zulässige Partition, wobei $\tau \times \sigma \in \mathbb{P}_{\text{near}} \implies |\tau|, |\sigma| \leq n_{\text{blatt}}$. Der Speicherbedarf N_{Storage} einer $\mathcal{H}^2$-Matrix $\mathbf{A}_{\mathcal{H}^2} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$ lässt sich beschränken durch*

$$N_{\text{Storage}} \leq \left(|\mathbb{T}_\mathcal{I}| + |\mathbb{T}_\mathcal{J}|\right)\left(r^2 + \frac{1}{2}C_{\text{sp}} \max\{n^2_{\text{blatt}}, r^2\}\right) + (n+m)r = \mathcal{O}(r^2 N),$$

wobei $N = n + m$.

Beweis. Wir müssen nur zusätzlich zu den Cluster-Basen auf den Blättern und den Transfermatrizen die Kopplungsmatrizen speichern. Die Kopplungsmatrizen $\mathbf{S}^{\tau\sigma}$ benötigen r^2 Speicher. Nahfeld Blöcke benötigen nach Voraussetzung maximal n_{blatt}^2 Speicher. Nach Definition der Schwachbesetztheitskonstante gilt $|\mathbb{P}| \leq C_{\text{sp}} \min\{|\mathbb{T}_{\mathcal{I}}|, |\mathbb{T}_{\mathcal{J}}|\}$. Für die Fernfeld-Blöcke und Kopplungsmatrizen benötigen wir also gesamt weniger als

$$\begin{aligned} |\mathbb{P}| \max\{r^2, n_{\text{blatt}}^2\} &\leq C_{\text{sp}} \min\{|\mathbb{T}_{\mathcal{I}}|, |\mathbb{T}_{\mathcal{J}}|\} \max\{r^2, n_{\text{blatt}}^2\} \\ &\leq \frac{1}{2} C_{\text{sp}} (|\mathbb{T}_{\mathcal{I}}| + |\mathbb{T}_{\mathcal{J}}|) \max\{r^2, n_{\text{blatt}}^2\} \end{aligned}$$

Speicher. Jede Transfer-Matrix benötigt r^2 Speicher. Da weniger als $|\mathbb{T}_{\mathcal{I}}|$ Vater-Sohn Paare in $\mathbb{T}_{\mathcal{I}}$ auftreten, benötigen alle Transfermatrizen maximal

$$r^2(|\mathbb{T}_{\mathcal{I}}| + |\mathbb{T}_{\mathcal{J}}|)$$

Speicher. Die Cluster-Basen V, W müssen nur für die Blätter gespeichert werden. Der Speicheraufwand ist also beschränkt durch

$$\sum_{\tau \in \text{leaves}\, \mathbb{T}_{\mathcal{I}}} |\tau| r + \sum_{\sigma \in \text{leaves}\, \mathbb{T}_{\mathcal{J}}} |\sigma| r \leq r(n+m),$$

womit das gewünschte Resultat gezeigt ist. □

Bemerkung 5.5. *Für balancierte Bäume kann das vorige Resultat verbessert werden, da hierfür $|\mathbb{T}_{\mathcal{I}}| \sim n/n_{\text{blatt}}$ gilt. Wählt man also $n_{\text{blatt}} \sim r$, so hat man einen Speicherbedarf von $\mathcal{O}(rN)$.*

5.2.1 Matrix-Vektor Multiplikation mit $\mathcal{H}^2$-Matrizen

Für $\mathcal{H}^2$-Matrizen kann ebenfalls eine Arithmetik angegeben werden. Haben hierbei die betrachteten $\mathcal{H}^2$-Matrizen das selbe Format, so können die arithmetischen Operationen in linearem Aufwand (ohne Logarithmus) durchgeführt werden.

Wir werden im Folgenden exemplarisch die Matrix-Vektor-Multiplikation mit $\mathcal{H}^2$-Matrizen genauer erklären.

Sei $\mathbf{A} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$. Wie bei $\mathcal{H}$-Matrizen schreiben wir blockweise

$$(\mathbf{A}x)_i = \sum_{\substack{\tau \times \sigma \in \mathbb{P}_{\text{near}} \\ i \in \tau}} (\mathbf{A}|_{\tau\times\sigma} x|_\sigma)_i + \sum_{\substack{\tau \times \sigma \in \mathbb{P}_{\text{far}} \\ i \in \tau}} (\mathbf{V}^\tau \mathbf{S}^{\tau\sigma} (\mathbf{W}^\sigma)^T x|_\sigma)_i$$

mit $1 \leq i \leq m$. Man sieht also, dass die Multiplikation am Fernfeld in drei Schritten durchgeführt wird.

1.Schritt: *Vorwärtstransformation*: Bestimme $\widehat{x}_\sigma := (\mathbf{W}^\sigma)^T x|_\sigma$ für alle $\sigma \in \mathbb{T}_{\mathcal{J}}$.

2.Schritt: *Multiplikation*: Für jeden Fernfeld-Block $\tau \times \sigma$ wird mit $\mathbf{S}^{\tau\sigma}$ multipliziert, also

$$\widehat{y}_\tau := \sum_{\substack{\sigma \in \mathbb{T}_\mathcal{J} \\ \tau\times\sigma \in \mathbb{P}_{\text{far}}}} \mathbf{S}^{\tau\sigma} \widehat{x}_\sigma \quad \forall \tau \in \mathbb{T}_\mathcal{I}.$$

3.Schritt: *Rückwärtstransformation*: Berechne $y := \sum_{\tau\in\mathbb{T}_\mathcal{I}} (\mathbf{V}^\tau \widehat{y}_\tau)$.

Bei der Vorwärtstransformation hängt das Matrix-Vektor Produkt nicht von $\tau \times \sigma$ ab, sondern nur von σ und muss somit nur einmal für jeden Cluster σ bestimmt werden. Gleiches gilt für die Rückwärtstransformation und Cluster τ.

Die Vektoren $\widehat{x}_\sigma, \widehat{y}_\tau$ benötigen hierbei zusätzlichen Speicher, aber dieser zusätzliche Bedarf ist von Ordnung $\mathcal{O}(n+m)$, also linear.

Lemma 5.6. *Die Berechnung der Nahfeld-Anteile sowie die Addition zu den Fernfeld-Anteilen benötigt* $2C_{\text{sp}} n_{\text{blatt}}^2 \min\{|\mathbb{T}_\mathcal{I}|, |\mathbb{T}_\mathcal{J}|\}$ *arithmetische Operationen.*

Beweis. Die Anzahl der Nahfeld-Blöcke ist beschränkt durch $|\mathbb{P}| \leq C_{\text{sp}} \min\{|\mathbb{T}_\mathcal{I}|, |\mathbb{T}_\mathcal{J}|\}$. Für jeden Block $\tau \times \sigma \in \mathbb{P}_{\text{near}}$ benötigen wir weniger als $n_{\text{blatt}}(2n_{\text{blatt}} - 1)$ Operationen für die exakte MVM $\mathbf{A}|_{\tau\times\sigma} x|_\sigma$ (wir haben in diesem Abschnitt zusätzlich verlangt, dass Nahfeld-Blöcke $|\tau|, |\sigma| \leq n_{\text{blatt}}$ erfüllen). Schlussendlich benötigen wir weniger als n_{blatt} Additionen für die Addition zu den (bereits berechneten) Fernfeld-Anteilen. □

Wir werden schlussendlich die Komplexität der Berechnung der Fernfeld-Anteile betrachten.

Lemma 5.7. *Der Multiplikationsschritt kann in weniger als* $2C_{\text{sp}} r^2 |\mathbb{T}_\mathcal{I}|$ *arithmetischen Operationen durchgeführt werden.*

Beweis. Für festes $\tau \in \mathbb{T}_\mathcal{I}$ hat die Summe über σ mit $\tau \times \sigma \in \mathbb{P}_{\text{far}}$ maximal C_{sp} Summanden. Jeder Summand ist ein Vektor der Länge r bestimmt mittels MVM mit einer $r \times r$-Matrix, was in $r(2r-1)$ Operationen durchgeführt werden kann. □

Die Vorwärts- und Rückwärtstransformation werden mit Hilfe der Transfermatrizen durchgeführt. Für $\sigma \in \mathbb{T}_\mathcal{J} \setminus \text{leaves}(\mathbb{T}_\mathcal{J})$ gilt

$$\begin{aligned} \widehat{x}_\sigma &= (\mathbf{W}^\sigma)^T x|_\sigma = \sum_{\sigma' \in \text{sons}(\sigma)} (\mathbf{T}^{\sigma'\sigma})^T (\mathbf{W}^{\sigma'})^T x|_{\sigma'} \\ &= \sum_{\sigma' \in \text{sons}(\sigma)} (\mathbf{T}^{\sigma'\sigma})^T \widehat{x}_{\sigma'}. \end{aligned} \tag{5.8}$$

Auf den Blättern $\sigma \in \text{leaves}(\mathbb{T}_\mathcal{J})$ berechnet man das Produkt $(\mathbf{W}^\sigma)^T x|_\sigma$ wie üblich. Somit hat man einen rekursiven Algorithmus für die Vorwärtstransformation, der auch (auf Grund der Verwendung der Transfermatrizen) *schnelle Vorwärtstransformation* genannt wird.

Algorithmus 5.8 (Schnelle Vorwärtstransformation).

```
function FastForwardTrafo(σ, x, var x̂)
if sons(σ) = ∅
  x̂_σ := (W^σ)^T x|_σ
else
  for σ' ∈ sons(σ)
    FastForwardTrafo(σ',x,x̂)
    x̂_σ := x̂_σ + (T^{σ'σ})^T x̂_{σ'}
  end
end
```

Lemma 5.9. *Die schnelle Vorwärtstransformation berechnet in weniger als* $4r^2|\mathbb{T}_{\mathcal{J}}| + r(2m-1)$ *arithmetischen Operationen die Vektoren* $\widehat{x}_\sigma$ *für alle* $\sigma \in \mathbb{T}_{\mathcal{J}}$.

Beweis. Die Multiplikation $(\mathbf{W}^\sigma)^T x|_\sigma$ für jedes Blatt $\sigma \in \operatorname{leaves}(\mathbb{T}_{\mathcal{J}})$ benötigt $r(2|\sigma|-1)$ Operationen. Summiert man über alle Blätter, so ist der Aufwand hierfür beschränkt durch $r(2m-1)$.

Ist $\sigma \in \mathbb{T}_{\mathcal{J}}$ kein Blatt, dann existieren genau zwei Söhne σ' und σ'' und mit (5.8) gilt

$$\widehat{x}_\sigma = (\mathbf{T}^{\sigma'\sigma})^T \widehat{x}_{\sigma'} + (\mathbf{T}^{\sigma''\sigma})^T \widehat{x}_{\sigma''}.$$

Pro Sohn benötigt das Update von $\widehat{x}_\sigma$ durch $\widehat{x}_{\sigma'}$ also weniger als $r(2r-1)+r = 2r^2$ Operationen. Im Baum $\mathbb{T}_{\mathcal{J}}$ gibt es weniger als $|\mathbb{T}_{\mathcal{J}}|$ Vaterknoten, also weniger als $2\,|\mathbb{T}_{\mathcal{J}}|$ Söhne. □

Für die schnelle Rückwärtstransformation überlegt man sich, dass für Cluster τ_1, τ_1' mit $\tau_1' \in \operatorname{sons}(\tau_1)$, $i \in \tau_1'$ gilt

$$\begin{aligned} y_i &= \sum_{\substack{\tau \in \mathbb{T}_{\mathcal{I}} \\ i \in \tau}} (\mathbf{V}^\tau \widehat{y}_\tau)_i = (\mathbf{V}^{\tau_1} \widehat{y}_{\tau_1})_i + (\mathbf{V}^{\tau_1'} \widehat{y}_{\tau_1'})_i + \sum_{\substack{\tau \in \mathbb{T}_{\mathcal{I}} \setminus \{\tau_1, \tau_1'\} \\ i \in \tau}} (\mathbf{V}^\tau \widehat{y}_\tau)_i \\ &= (\mathbf{V}^{\tau_1'}(\mathbf{T}^{\tau_1'\tau_1} \widehat{y}_{\tau_1} + \widehat{y}_{\tau_1'}))_i + \sum_{\substack{\tau \in \mathbb{T}_{\mathcal{I}} \setminus \{\tau_1, \tau_1'\} \\ i \in \tau}} (\mathbf{V}^\tau \widehat{y}_\tau)_i. \end{aligned}$$

Somit erhält man abermals einen rekursiven Algorithmus, der auch *schnelle Rückwärtstransformation* genannt wird.

Algorithmus 5.10 (Schnelle Rückwärtstransformation).

```
function FastBackwardTrafo(τ, var y, ŷ)
if sons(τ) = ∅
  y|_τ := V^τ ŷ_τ
else
  for τ' ∈ sons(τ)
```

```
    ŷ_τ' := ŷ_τ' + T^{τ'τ} ŷ_τ
    FastBackwardTrafo(τ', y, ŷ)
  end
end
```

Lemma 5.11. *Die schnelle Rückwärtstransformation benötigt weniger als* $4r^2|\mathbb{T}_\mathcal{I}| + n(2r-1)$ *arithmetische Operationen.*

Beweis. Analog zur Vorwärtstransformation. □

Setzt man nun die jeweiligen Teilschritte zusammen, so erhält man die Gesamtkomplexität der Matrix-Vektor-Multiplikation mit $\mathcal{H}^2$-Matrizen.

Proposition 5.12. *Sei* $\mathbf{A} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$ *und* $x \in \mathbb{R}^m$. *Die Anzahl an arithmetischen Operationen* N_{MVM} *zur Berechnung der Matrix-Vektor-Multiplikation* $\mathbf{A}x$ *ist beschränkt durch*

$$N_{\mathrm{MVM}} \leq \left(2C_{\mathrm{sp}}(r^2 + n_{\mathrm{blatt}}^2) + 8r^2\right) \max\{|\mathbb{T}_\mathcal{I}|, |\mathbb{T}_\mathcal{J}|\} + 2r(n+m) = \mathcal{O}(r^2 N)$$

Wir haben also lineare Komplexität. □

5.2.2 Projektion auf $\mathcal{H}^2$-Matrix-Format

Wir werden in diesem Abschnitt die Konversion von beliebigen Matrizen sowie von $\mathcal{H}$-Matrizen auf das $\mathcal{H}^2$-Format untersuchen.

Für spezielle Cluster-Basen lässt sich dies einfach durchführen:

Definition 5.13. *Eine Cluster-Basis* $(\mathbf{V}^\tau)_{\tau \in \mathbb{T}_\mathcal{I}}$ *heißt* ***orthogonal****, falls*

$$(\mathbf{V}^\tau)^T \mathbf{V}^\tau = \mathbf{I} \quad \forall \tau \in \mathbb{T}_\mathcal{I}.$$

Aus der Orthogonalität der Spalten von $\mathbf{V}^\tau$ folgt, dass $\mathbf{V}^\tau(\mathbf{V}^\tau)^T$ als Abbildung von $\mathbb{R}^\tau \to \mathbb{R}^\tau$ die Orthogonalprojektion auf den von $\mathbf{V}^\tau$ erzeugten Unterraum (das Bild von $\mathbf{V}^\tau$) bezüglich des euklidischen Skalarprodukts ist.

Wir versehen den Raum der $\mathbb{R}^{n\times m}$-Matrizen mit dem Frobenius Innenprodukt

$$\langle \mathbf{A}, \mathbf{B} \rangle_F := \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \mathbf{A}_{ij}\mathbf{B}_{ij} = \mathrm{spur}(\mathbf{A}\mathbf{B}^T),$$

dann ist $(\mathbb{R}^{n\times m}, \langle\cdot,\cdot\rangle_F)$ ein Hilbertraum. Somit ist die Bestapproximation aus dem Raum $\mathcal{H}^2(r, \mathbb{P}, V, W)$ gerade die Orthogonalprojektion auf diesen Unterraum.

Lemma 5.14. *Seien* $(\mathbf{V}^\tau)_{\tau\in\mathbb{T}_\mathcal{I}}$, $(\mathbf{W}^\sigma)_{\sigma\in\mathbb{T}_\mathcal{J}}$ *orthogonale Cluster-Basen.*

Die Abbildung $\mathbf{A} \mapsto \mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}$ *als Abbildung von* $\mathbb{R}^{\tau\times\sigma} \to \mathbb{R}^{\tau\times\sigma}$ *ist die orthogonale Projektion auf den Unterraum* $\{\mathbf{A} \in \mathbb{R}^{\tau\times\sigma} : \mathrm{im}(\mathbf{A}) \subset \mathrm{im}(\mathbf{V}^\tau)\}$ *bezüglich des Frobenius-Skalarprodukts.*

Im gleichen Sinn ist die Abbildung $\mathbf{A} \mapsto \mathbf{A}\mathbf{W}^\sigma(\mathbf{W}^\sigma)^T$ *orthogonale Projektion auf das Bild von* $\mathbf{W}^\sigma$.

Beweis. Wir zeigen nur die Aussage für $\mathbf{V}^\tau$, die Aussage für $\mathbf{W}^\sigma$ folgt analog indem man Zeilen statt Spalten verwendet.

Die Projektionseigenschaft folgt aus der Orthogonalität von $\mathbf{V}^\tau$.

Seien $\mathbf{A}_{\tau,j}$, $j \in \sigma$ die Spalten von $\mathbf{A}$. Das Frobenius Skalarprodukt lässt sich nach Definition spaltenweise mit dem euklidischen Skalarprodukt schreiben als

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_{j \in \sigma} \langle \mathbf{A}_{\tau,j}, \mathbf{B}_{\tau,j} \rangle_2 .$$

Somit gilt, da $\mathbf{V}^\tau (\mathbf{V}^\tau)^T$ Orthogonalprojektion bezüglich des euklidischen Skalarprodukts ist, dass

$$\begin{aligned} \left\langle \mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{A}, \mathbf{B} \right\rangle_F &= \sum_{j \in \sigma} \left\langle (\mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{A})_{\tau,j}, \mathbf{B}_{\tau,j} \right\rangle_2 = \sum_{j \in \sigma} \left\langle \mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{A}_{\tau,j}, \mathbf{B}_{\tau,j} \right\rangle_2 \\ &= \sum_{j \in \sigma} \left\langle \mathbf{A}_{\tau,j}, \mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{B}_{\tau,j} \right\rangle_2 = \left\langle \mathbf{A}, \mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{B} \right\rangle_F , \end{aligned}$$

womit die Orthogonalität der Projektion gezeigt ist. □

Mittels orthogonaler Cluster-Basen $(\mathbf{V}^\tau)_{\tau \in \mathbb{T}_\mathcal{I}}$, $(\mathbf{W}^\sigma)_{\sigma \in \mathbb{T}_\mathcal{J}}$ lässt sich die Projektion blockweise definieren als

$$\Pi_{\mathcal{H}^2}(\mathbf{A})|_{\tau \times \sigma} = \begin{cases} \Pi_{\tau \times \sigma}(\mathbf{A}|_{\tau \times \sigma}) & \text{für } \tau \times \sigma \in \mathbb{P}_{\text{far}}, \\ \mathbf{A}|_{\tau \times \sigma} & \text{sonst}, \end{cases}$$

wobei

$$\Pi|_{\tau \times \sigma} : \mathbb{R}^{\tau \times \sigma} \to \mathbb{R}^{\tau \times \sigma}, \quad \mathbf{A} \mapsto \mathbf{V}^\tau (\mathbf{V}^\tau)^T \mathbf{A} \mathbf{W}^\sigma (\mathbf{W}^\sigma)^T .$$

Mit Lemma 5.14 folgt, dass $\Pi_{\mathcal{H}^2}$ die orthogonale Projektion auf $\mathcal{H}^2(r, \mathbb{P}, V, W)$ ist, es gilt also

$$\| \mathbf{A} - \Pi_{\mathcal{H}^2}(\mathbf{A}) \|_F = \min_{\mathbf{X} \in \mathcal{H}^2(r,\mathbb{P},V,W)} \| \mathbf{A} - \mathbf{X} \|_F .$$

Die Berechnung der Projektion kann prinzipiell für jede Matrix bestimmt werden, für vollbesetzte Matrizen hat diese allerdings quadratischen Aufwand.

Für $\mathcal{H}$-Matrizen geht dies allerdings effizienter: Sei eine $\mathcal{H}$-Matrix $\mathbf{A}_\mathcal{H} \in \mathcal{H}(r, \mathbb{P})$ gegeben. Wir wollen aus dieser eine $\mathcal{H}^2$-Matrix $\mathbf{A}_{\mathcal{H}^2} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$ mit gleicher Partition konstruieren. Hierfür muss die Faktorisierung $\mathbf{A}|_{\tau \times \sigma} = \mathbf{X}_{\tau\sigma} \mathbf{Y}_{\tau\sigma}^T$ umgewandelt werden in

$$\Pi_{\mathcal{H}^2}(\mathbf{A})|_{\tau \times \sigma} = \mathbf{V}^\tau \mathbf{S}^{\tau\sigma} (\mathbf{W}^\sigma)^T, \quad \text{wobei } \mathbf{S}^{\tau\sigma} := (\mathbf{V}^\tau)^T \mathbf{X}_{\tau\sigma} \mathbf{Y}_{\tau\sigma}^T \mathbf{W}^\sigma .$$

- Die Berechnung von $(\mathbf{V}^\tau)^T \mathbf{X}_{\tau\sigma}$ hat einen Aufwand von $2\,|\tau|\, r^2$,
- die Berechnung von $\mathbf{Y}_{\tau\sigma}^T \mathbf{W}^\sigma$ einen Aufwand von $2\,|\sigma|\, r^2$,
- die Multiplikation beider Matrizen einen Aufwand von $2r^3$.

Gesamt erhalten wir also mit Bemerkung 2.12 einen Aufwand von

$$\begin{aligned} 2 \sum_{\tau \times \sigma \in \mathbb{P}_{\text{far}}} r^2 (|\tau| + |\sigma|) + r^3 &\leq 2r^2 C_{\text{sp}} (\text{depth}(\mathbb{P}) + 1) N + 4r^3 C_{\text{sp}} N \\ &= \mathcal{O}(r^2 N \log N + r^3 N). \end{aligned}$$

5.2.3 Niedrigrangstruktur von $\mathcal{H}^2$-Matrizen

Die Approximation einer Matrix mit $\mathcal{H}$-Matrizen hängt nur davon ab ob man die Matrix auf Fernfeld-Blöcken durch Niedrigrang Matrizen approximieren kann. Eine derartige Charakterisierung ist für $\mathcal{H}^2$-Matrizen nicht so einfach, da eine Kopplung von Blöcken mittels geschachtelter Cluster-Basen besteht. Man muss also größere Teilmatrizen betrachten.

Definition 5.15. *Für einen Cluster τ bezeichnen wir die Menge aller Vorfahren im Cluster-Baum $\mathbb{T}_\mathcal{I}$ mit* $\operatorname{pred}(\tau)$.

Für $\tau \in \mathbb{T}_\mathcal{I}, \sigma \in \mathbb{T}_\mathcal{J}$ definieren wir die ***Fernfeld-Indizes*** *als*

$$\mathcal{F}_\tau := \bigcup_{\tau^+ \in \operatorname{pred}(\tau)} \bigcup_{\substack{\sigma \in \mathbb{T}_\mathcal{J} \\ \tau^+ \times \sigma \in \mathbb{P}_{\text{far}}}} \sigma$$

$$\mathcal{F}_\sigma := \bigcup_{\sigma^+ \in \operatorname{pred}(\sigma)} \bigcup_{\substack{\tau \in \mathbb{T}_\mathcal{I} \\ \tau \times \sigma^+ \in \mathbb{P}_{\text{far}}}} \tau.$$

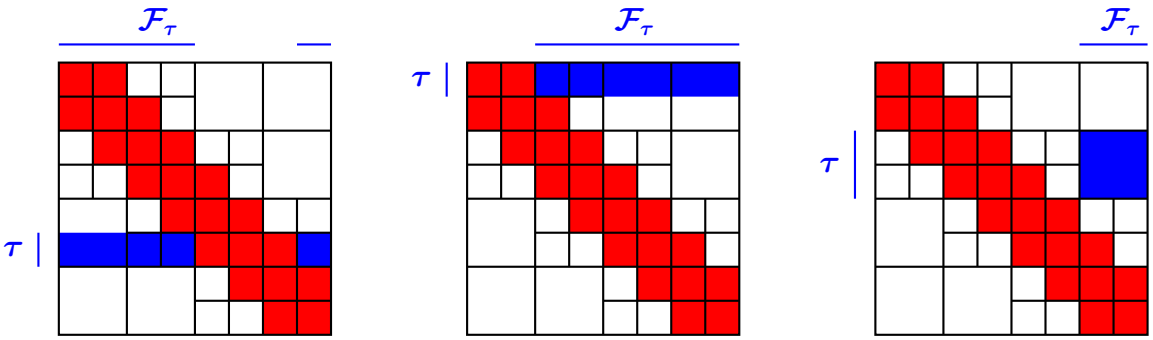


Abbildung 5.1: Die Mengen $\tau \times \mathcal{F}_\tau$ (blau), nicht zulässige Blöcke sind rot markiert

Lemma 5.16. *Sei $\tau \in \mathbb{T}_\mathcal{I}$. Für jedes $j \in \mathcal{F}_\tau$ existiert ein eindeutiger Cluster $\tau^+ \in \operatorname{pred}(\tau)$ sowie genau ein $\sigma \in \mathbb{T}_\mathcal{J}$ mit $\tau^+ \times \sigma \in \mathbb{P}$ sowie $j \in \sigma$. Eine analoge Aussage folgt auch für σ und $\mathcal{F}_\sigma$.*

Beweis. Sei $j \in \mathcal{F}_\tau$ und $\tau_1^+, \tau_2^+ \in \operatorname{pred}(\tau)$ und $\sigma_1, \sigma_2 \in \mathbb{T}_\mathcal{J}$ mit $\tau_1^+ \times \sigma_1 \in \mathbb{P}$, $\tau_2^+ \times \sigma_2 \in \mathbb{P}$ sowie $j \in \sigma_1$, $j \in \sigma_2$.

Sei $i \in \tau$, dann folgt auch $i \in \tau_1^+$, $i \in \tau_2^+$ und somit $(i,j) \in \tau_1^+ \times \sigma_1$ und $(i,j) \in \tau_2^+ \times \sigma_2$. Da die Partition $\mathbb{P}$ eine disjunkte Zerlegung ist, folgt also $\tau_1^+ = \tau_2^+$ und $\sigma_1 = \sigma_2$. □

Bemerkung 5.17. *Wir haben bei geschachtelten Cluster-Basen nur Transfermatrizen $\mathbf{T}^{\tau' \tau}$ für $\tau' \in \operatorname{sons}(\tau)$. Bei der Menge $\mathcal{F}_\sigma$ treten allerdings auch frühere Vorfahren auf. In*

natürlicher Art und Weise kann man erweiterte Transfermatrizen $\mathbf{T}^{\widehat{\tau}\tau}$ für $\widehat{\tau} \in \mathbb{T}_{\mathcal{I}}$, $\widehat{\tau} \subset \tau$ als Produkt der Transfermatrizen der dazwischengelegenen Vater-Sohn Paare schreiben:

$$\mathbf{T}^{\widehat{\tau}\tau} = \prod_{\ell=\operatorname{level}\widehat{\tau}}^{\operatorname{level}\tau+1} \mathbf{T}^{\mathcal{F}_\ell(\widehat{\tau})\mathcal{F}_{\ell-1}(\widehat{\tau})}.$$

Hier bezeichnet $\mathcal{F}_\ell(\widehat{\tau})$ wie in vorigen Kapiteln den eindeutigen Vorfahren von $\widehat{\tau}$ mit Level ℓ im Baum $\mathbb{T}_{\mathcal{I}}$.

Die nachfolgende Proposition liefert eine Niedrigrang-Charakterisierung für $\mathcal{H}^2$-Matrizen. Für den Beweis benötigen wir ein Resultat aus der linearen Algebra, dass für jede Matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ eine partielle Inverse existiert, also eine Matrix $\mathbf{R} \in \mathbb{R}^{m\times n}$ mit $\mathbf{A} = \mathbf{ARA}$.

Proposition 5.18. *Sei $\mathbf{A} \in \mathbb{R}^{n\times m}$ und geschachtelte Cluster-Basen V, W mit Rang r gegeben. Dann ist $\mathbf{A} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$ genau dann, wenn Matrizen $\mathbf{X}_\sigma \in \mathbb{R}^{\mathcal{F}_\sigma\times r}$, $\mathbf{Y}_\tau \in \mathbb{R}^{\mathcal{F}_\tau\times r}$ existieren, sodass*

$$\begin{aligned} \mathbf{A}|_{\tau\times\mathcal{F}_\tau} &= \mathbf{V}^\tau\mathbf{Y}_\tau^T \qquad \forall\tau \in \mathbb{T}_{\mathcal{I}} && (5.9)\\ \mathbf{A}|_{\mathcal{F}_\sigma\times\sigma} &= \mathbf{X}_\sigma(\mathbf{W}^\sigma)^T \qquad \forall\sigma \in \mathbb{T}_{\mathcal{J}}. && (5.10) \end{aligned}$$

Beweis. Seien für alle $\tau \in \mathbb{T}_{\mathcal{I}}, \sigma \in \mathbb{T}_{\mathcal{J}}$ Matrizen $\mathbf{X}_\sigma \in \mathbb{R}^{\mathcal{F}_\sigma\times r}$, $\mathbf{Y}_\tau \in \mathbb{R}^{\mathcal{F}_\tau\times r}$ gegeben mit (5.9). Wir müssen also für eine $\mathcal{H}^2$-Matrix noch die Kopplungsmatrizen $\mathbf{S}^{\tau\sigma} \in \mathbb{R}^{r\times r}$ konstruieren. Sei $\tau\times\sigma \in \mathbb{P}_{\text{far}}$, dann gilt $\tau \subset \mathcal{F}_\sigma$ und $\sigma \subset \mathcal{F}_\tau$ und

$$(\mathbf{X}_\sigma)|_{\tau\times r}(\mathbf{W}^\sigma)^T = (\mathbf{X}_\sigma(\mathbf{W}^\sigma)^T)|_{\tau\times\sigma} = \mathbf{A}|_{\tau\times\sigma} = (\mathbf{V}^\tau\mathbf{Y}_\tau^T)|_{\tau\times\sigma} = \mathbf{V}^\tau(\mathbf{Y}_\tau|_{\sigma\times r})^T.$$

Mit der partiellen Inversen $\mathbf{R} \in \mathbb{R}^{r\times\tau}$ zu $\mathbf{V}^\tau$ folgt

$$\begin{aligned} \mathbf{A}|_{\tau\times\sigma} = \mathbf{V}^\tau(\mathbf{Y}_\tau|_{\sigma\times r})^T &= \mathbf{V}^\tau\mathbf{RV}^\tau(\mathbf{Y}_\tau|_{\sigma\times r})^T = \mathbf{V}^\tau\mathbf{RA}|_{\tau\times\sigma} = \mathbf{V}^\tau\mathbf{R}(\mathbf{X}_\sigma)|_{\tau\times r}(\mathbf{W}^\sigma)^T\\ &=: \mathbf{V}^\tau\mathbf{S}^{\tau\sigma}(\mathbf{W}^\sigma)^T, \end{aligned}$$

also ist $\mathbf{A} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$.

Umgekehrt sei $\mathbf{A} \in \mathcal{H}^2(r, \mathbb{P}, V, W)$. Sei $j \in \mathcal{F}_\tau$. Dann liefert Lemma 5.16 eindeutige Elemente $\tau^+ \in \operatorname{pred}(\tau)$, $\sigma \in \mathbb{T}_{\mathcal{I}}$ mit $\tau^+\times\sigma \in \mathbb{P}_{\text{far}}$, $j \in \sigma$. Die Definition einer $\mathcal{H}^2$-Matrix liefert gemeinsam mit den obig definierten erweiterten Transfermatrizen, dass

$$\mathbf{A}|_{\tau\times\sigma} = (\mathbf{A}|_{\tau^+\times\sigma})|_{\tau\times\sigma} = \mathbf{V}^{\tau^+}|_{\tau\times r}\mathbf{S}^{\tau^+\sigma}(\mathbf{W}^\sigma)^T = \mathbf{V}^\tau\mathbf{T}^{\tau\tau^+}\mathbf{S}^{\tau^+\sigma}(\mathbf{W}^\sigma)^T.$$

Dann ist die Matrix $\mathbf{Y}_\tau \in \mathbb{R}^{\mathcal{F}_\tau\times r}$ eindeutig definiert durch

$$\mathbf{Y}_\tau|_{\sigma\times r} := \mathbf{W}^\sigma(\mathbf{S}^{\tau^+\sigma})^T(\mathbf{T}^{\tau\tau^+})^T \quad \forall\tau^+ \in \operatorname{pred}(\tau), \sigma \in \mathbb{T}_{\mathcal{J}}, \text{ mit } \tau^+\times\sigma \in \mathbb{P}_{\text{far}}$$

und es gilt (5.9). Die Konstruktion von $\mathbf{X}_\sigma$ erfolgt analog. □

5.2.4 Konstruktion von Cluster-Basen

Im Allgemeinen sind die Cluster-Basen nicht a-priori bekannt, womit obige Projektion nicht berechnet werden kann. Wir stellen im Folgenden eine Konstruktion dieser vor.

Das nachfolgende Lemma zeigt, dass die Konstruktion der Cluster-Basen separat durchgeführt werden kann, da der Fehler beschränkt ist durch die jeweiligen einzelnen Projektionsfehler.

Lemma 5.19. *Seien $(\mathbf{V}^\tau)_{\tau\in\mathbb{T}_\mathcal{I}}$, $(\mathbf{W}^\sigma)_{\sigma\in\mathbb{T}_\mathcal{J}}$ orthogonale Cluster-Basen und $\mathbf{A}\in\mathbb{R}^{\tau\times\sigma}$. Dann gilt*

$$\|\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A}\|_2^2 \leq \left\|\mathbf{A}-\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}\right\|_2^2 + \left\|\mathbf{A}^T-\mathbf{W}^\sigma(\mathbf{W}^\sigma)^T\mathbf{A}^T\right\|_2^2.$$

Umgekehrt gilt auch

$$\left\|\mathbf{A}-\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}\right\|_2^2 \leq \|\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A}\|_2^2, \quad \left\|\mathbf{A}^T-\mathbf{W}^\sigma(\mathbf{W}^\sigma)^T\mathbf{A}^T\right\|_2^2 \leq \|\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A}\|_2^2.$$

Beweis. Sei $z\in\mathbb{R}^\tau$, $x:=\mathbf{A}z$ und $y:=\Pi_{\tau\times\sigma}\mathbf{A}z$. Dann gilt da $\mathbf{P}:=\mathbf{V}^\tau(\mathbf{V}^\tau)^T$ orthogonale Projektion ist

$$\begin{aligned}\|(\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A})z\|_2^2 &= \|x-y\|_2^2 = \|x-\mathbf{P}x\|_2^2 + \|y-\mathbf{P}x\|_2^2\\ &= \left\|x-\mathbf{V}^\tau(\mathbf{V}^\tau)^Tx\right\|_2^2 + \left\|\mathbf{P}(\mathbf{A}\mathbf{W}^\sigma(\mathbf{W}^\sigma)^Tz-x)\right\|_2^2\\ &\leq \left\|(\mathbf{A}-\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A})z\right\|_2^2 + \left\|(\mathbf{A}-\mathbf{A}\mathbf{W}^\sigma(\mathbf{W}^\sigma)^T)z\right\|_2^2.\end{aligned}$$

Da z beliebig war folgt also die erste Aussage.

Aus der dritten Gleichheit folgt

$$\|(\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A})z\|_2 \geq \left\|x-\mathbf{V}^\tau(\mathbf{V}^\tau)^Tx\right\|_2 = \left\|(\mathbf{A}-\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A})z\right\|_2$$

und nimmt man das Supremum über z die zweite Aussage.

Die Aussage für $\mathbf{W}^\sigma$ folgt mit dem selben Argument, wobei man $(\mathbf{A}-\Pi_{\tau\times\sigma}\mathbf{A})^T$ betrachtet und $\|\mathbf{B}\|_2 = \left\|\mathbf{B}^T\right\|_2$ verwendet. □

Bemerkung 5.20. *Das selbe Resultat gilt auch für die Frobenius-Norm.*

Konstruiert man also die Cluster Basen V, W separat, so wird der Fehler maximal um einen Faktor $\sqrt{2}$ größer.

Wir werden nur eine Konstruktion der Cluster-Basis V vorstellen, eine Konstruktion für W funktioniert analog bei Betrachtung der transponierten Matrix.

Proposition 5.18 suggeriert hierbei nach einer Niedrigrang-Faktorisierung der Submatrix

$$\mathbf{A}_\tau := \mathbf{A}|_{\tau\times\mathcal{F}_\tau}$$

zu suchen. Wir beginnen bei den Blättern.

1. **τ ist Blatt:** Dann bestimmen wir die Singulärwertzerlegung von $\mathbf{A}_\tau = \mathbf{G}\mathbf{\Sigma}\mathbf{Q}^T$. Nimmt man nur die ersten r-Spalten von $\mathbf{G} = (g_1, \dots, g_{|\tau|})$ und definiert man
$$\mathbf{V}^\tau := (g_1, \dots, g_r),$$
dann ist $\mathbf{V}^\tau$ orthogonal und $\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}_\tau = \mathbf{G}\mathbf{\Sigma}_r\mathbf{Q}^T$, wobei $\mathbf{\Sigma}_r = \operatorname{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0)$. Proposition 3.6 liefert dann die Fehlerabschätzung
$$\left\|\mathbf{A}_\tau - \mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}_\tau\right\|_2 \leq \sigma_{r+1}$$
sowie die Bestapproximationseigenschaft von $\mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}_\tau$.

2. **τ ist kein Blatt:** Dann hat τ Söhne $\tau_1, \tau_2 \in \operatorname{sons}(\tau)$. Die Definition der geschachtelten Basen impliziert, dass Transfermatrizen $\mathbf{T}_{\tau_i\tau}$ existieren müssen, sodass
$$\mathbf{V}^\tau = \begin{pmatrix} \mathbf{V}^{\tau_1} & 0 \\ 0 & \mathbf{V}^{\tau_2} \end{pmatrix} \begin{pmatrix} \mathbf{T}^{\tau_1\tau} \\ \mathbf{T}^{\tau_2\tau} \end{pmatrix} =: \mathbf{U}^\tau\mathbf{T}^\tau, \tag{5.11}$$
wobei die orthogonalen Matrizen $\mathbf{V}^{\tau_i}$ induktiv bereits berechnet sind. Da $\mathbf{V}^{\tau_i}$ orthogonal sind ist auch $\mathbf{U}^\tau$ orthogonal. Wir wollen, dass $\mathbf{V}^\tau$ ebenfalls orthogonal ist, also
$$\mathbf{I} = (\mathbf{V}^\tau)^T\mathbf{V}^\tau = (\mathbf{U}^\tau\mathbf{T}^\tau)^T\mathbf{U}^\tau\mathbf{T}^\tau = (\mathbf{T}^\tau)^T\mathbf{T}^\tau.$$
Wir müssen also eine geeignete orthogonale Matrix $\mathbf{T}^\tau$ finden.

Sei $z \in \mathbb{R}^{\mathcal{F}_\tau}$ und $x := \mathbf{A}_\tau z$. Mit Pythagoras folgt
$$\begin{aligned}
\left\|(\mathbf{A}_\tau - \mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 &= \left\|x - \mathbf{U}^\tau\mathbf{T}^\tau(\mathbf{T}^\tau)^T(\mathbf{U}^\tau)^Tx\right\|_2^2 \\
&= \left\|x - \mathbf{U}^\tau(\mathbf{U}^\tau)^Tx\right\|_2^2 + \left\|\mathbf{U}^\tau(\mathbf{U}^\tau)^Tx - \mathbf{U}^\tau\mathbf{T}^\tau(\mathbf{T}^\tau)^T(\mathbf{U}^\tau)^Tx\right\|_2^2 \\
&= \left\|(\mathbf{A}_\tau - \mathbf{U}^\tau(\mathbf{U}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 + \left\|\mathbf{U}^\tau((\mathbf{U}^\tau)^T\mathbf{A}_\tau - \mathbf{T}^\tau(\mathbf{T}^\tau)^T(\mathbf{U}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 \\
&= \left\|(\mathbf{A}_\tau - \mathbf{U}^\tau(\mathbf{U}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 + \left\|((\mathbf{U}^\tau)^T\mathbf{A}_\tau - \mathbf{T}^\tau(\mathbf{T}^\tau)^T(\mathbf{U}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2.
\end{aligned}$$
Definiert man $\widehat{\mathbf{A}}_\tau := (\mathbf{U}^\tau)^T\mathbf{A}_\tau$, dann folgt
$$\left\|(\mathbf{A}_\tau - \mathbf{V}^\tau(\mathbf{V}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 = \left\|(\mathbf{A}_\tau - \mathbf{U}^\tau(\mathbf{U}^\tau)^T\mathbf{A}_\tau)z\right\|_2^2 + \left\|(\widehat{\mathbf{A}}_\tau - \mathbf{T}^\tau(\mathbf{T}^\tau)^T\widehat{\mathbf{A}}_\tau)z\right\|_2^2.$$
Da $\mathbf{U}^\tau$ aus den Cluster Basis Matrizen der Söhne von τ besteht, ist der erste Term auf der rechten Seite gerade der gemachte Approximationsfehler bei den Sohn Blöcken. Der zweite Term beschreibt den Approximationsfehler an $\widehat{\mathbf{A}}_\tau$ der Orthogonalprojektion $\mathbf{T}^\tau(\mathbf{T}^\tau)^T$. Für diesen können wir wie bei den Blättern vorgehen und eine SVD bestimmen, also
$$\widehat{\mathbf{A}}_\tau = \widehat{\mathbf{G}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{Q}}^T$$
und wählen die ersten r Spalten von $\widehat{\mathbf{G}}$ für $\mathbf{T}^\tau$. Dann sind schlussendlich die Cluster Basis Matrizen und Transfermatrizen bestimmt durch (5.11).

Bemerkung 5.21. *Die Matrizen $\widehat{\mathbf{A}}_\tau$ direkt zu berechnen ist ineffizient. Stattdessen können Hilfsmatrizen $\mathbf{X}_\tau := (\mathbf{V}^\tau)^T \mathbf{A}_\tau \in \mathbb{R}^{r \times \mathcal{F}_\tau}$ betrachtet werden, aus denen man $\widehat{\mathbf{A}}_\tau$ mittels*

$$\widehat{\mathbf{A}}_\tau = \begin{pmatrix} \mathbf{X}_{\tau_1}|_{r \times \mathcal{F}_\tau} \\ \mathbf{X}_{\tau_2}|_{r \times \mathcal{F}_\tau} \end{pmatrix}$$

erhält. Die Matrizen $\mathbf{X}_\tau$ können mittels

$$\mathbf{X}_\tau = (\mathbf{V}^\tau)^T \mathbf{A}_\tau = (\mathbf{T}^\tau)^T (\mathbf{U}^\tau)^T \mathbf{A}_\tau = (\mathbf{T}^\tau)^T \widehat{\mathbf{A}}_\tau$$

effizient rekursiv mittels den Transfermatrizen bestimmt werden.

Bemerkung 5.22. *Der Aufwand für die Bestimmung der Cluster Basen für eine beliebige Matrix kann mit den üblichen Methoden abgeschätzt werden und ist beschränkt durch*

$$Crnm + Cr^2 \left|\mathbb{T}_{\mathcal{I}}\right| m.$$

Wir haben also quadratischen Aufwand für die Berechnung der quasi-Bestapproximation im Gegensatz zum kubischen Aufwand bei $\mathcal{H}$-Matrizen.

Hat man allerdings eine $\mathcal{H}$-Matrix gegeben, so können orthogonale Cluster Basen mit einem Aufwand von

$$C(\operatorname{depth}(\mathbb{P}) + 1)(r^2(n+m) + r^3 \left|\mathbb{T}_{\mathcal{I}}\right|)$$

also logarithmisch linearem Aufwand bestimmt werden. Details finden sich in [Bör10].

Literaturverzeichnis

[Beb00] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[Beb08] M. Bebendorf. *Hierarchical Matrices*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2008.

[BG99] S. Börm and L. Grasedyck. *H-Lib - a library for $\mathcal{H}$- and $\mathcal{H}^2$-matrices*. available at http://www.hlib.org, 1999.

[BG04] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72(3-4):325–332, 2004.

[BG05] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Numer. Math.*, 101(2):221–249, 2005.

[BG13] S. Börm and J. Gördes. Low-rank approximation of integral operators by using the Green formula and quadrature. *Numer. Algorithms*, 64(3):567–592, 2013.

[BGH06] S. Börm, L. Grasedyck, and W. Hackbusch. *Hierarchical Matrices*. Lecture Notes, Kiel, 2006.

[BKV15] M. Bebendorf, C. Kuske, and R. Venn. Wideband nested cross approximation for Helmholtz problems. *Numer. Math.*, 130(1):1–34, 2015.

[BLM05] Steffen Börm, Maike Löhndorf, and Jens M. Melenk. Approximation of integral operators by variable-order interpolation. *Numer. Math.*, 99(4):605–643, 2005.

[Bör10] S. Börm. *Efficient numerical methods for non-local operators*, volume 14 of *EMS Tracts in Mathematics*. European Mathematical Society (EMS), Zürich, 2010.

[Bör16] S. Börm. *Numerical Methods for Non-local Operators*. Lecture Notes, Kiel, 2016.

[FKS17] M. Feischl, F. Kuo, and I.H. Sloan. Fast random field generation with h-matrices. *https://arxiv.org/abs/1702.08637*, 2017.

[FMP16] Markus Faustmann, Jens Markus Melenk, and Dirk Praetorius. Existence of $\mathcal{H}$-matrix approximants to the inverses of BEM matrices: the simple-layer operator. *Math. Comp.*, 85(297):119–152, 2016.

[Gie01] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67(3):183–207, 2001.

[GR97] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace in three dimensions. In *Acta Numerica 1997*, pages 229–269. Cambridge University Press, 1997.

[Gra01] L. Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. PhD thesis, Universität Kiel, 2001.

[GTZ97] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[Hac99] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[Hac09] W. Hackbusch. *Hierarchische Matrizen: Algorithmen und Analysis*. Springer, 2009.

[Hac16] Wolfgang Hackbusch. New estimates for the recursive low-rank truncation of block-structured matrices. *Numer. Math.*, 132(2):303–328, 2016.

[HB02] W. Hackbusch and S. Börm. Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices. *Computing*, 69(1):1–35, 2002.

[Pra09] D. Praetorius. *Hierarchische Matrizen und Fast Multipole Method*. Lecture Notes, Wien, 2009.

[Riv74] Th. J. Rivlin. *The Chebyshev polynomials*. Wiley-Interscience, New York, 1974.

[Rok85] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.*, 60:187–207, 1985.