Theorie und Anwendungen Hierarchischer Matrizen

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität
zu Kiel

vorgelegt von

Lars Grasedyck

Kiel
2001

Theorie und Anwendungen Hierarchischer Matrizen

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität
zu Kiel

vorgelegt von

Lars Grasedyck

Kiel
2001

Referent:	Prof. Dr. Dr. h.c. Wolfgang Hackbusch
Korreferenten:	Priv. Doz. Jens Markus Melenk, PhD (Leipzig)
	Prof. Dr. Stefan Sauter (Zürich)
Tag der mündlichen Prüfung:	20. Juli 2001
Zum Druck genehmigt:	Kiel, den 20. Juli 2001

gez. T. Bauer
Der Dekan

Inhaltsverzeichnis

Zusammenfassung und Danksagung **6**

1. Einführung an einem Beispiel **7**
1.1. Die Kernfunktion . . . 7
1.2. Galerkinverfahren . . . 8
1.3. Approximation durch einen entarteten Kern . . . 8
1.4. Zulässigkeitsbedingung . . . 9
1.5. Darstellung zulässiger Blöcke im Galerkinverfahren . . . 9
1.6. Konstruktion zulässiger Blöcke . . . 10
1.7. Definition der Arithmetik . . . 11
1.8. Aufwand der Arithmetik . . . 14
1.9. Ziel der Arbeit . . . 14

2. $\mathbb{R}k$-Matrizen **15**
2.1. Definitionen und Grundlagen . . . 15
2.2. Auswertung . . . 16
2.3. Multiplikation . . . 17
2.4. Darstellungswechsel und Konvertierungen . . . 17
2.4.1. Gekürzte Singulärwertzerlegung . . . 17
2.4.2. Orthogonale Iteration . . . 20
2.4.3. Gekürzte QR-Zerlegung . . . 20
2.5. Addition . . . 24
2.6. Spektral- und Frobeniusnorm . . . 24
2.7. Komplexwertige $\mathbb{R}k$-Matrizen . . . 24

3. Hierarchische Partitionierung **26**
3.1. Partitionierung und Clusterung . . . 28
3.2. Die Zulässigkeitsbedingung . . . 33
3.2.1. Standard-Zulässigkeitsbedingung . . . 33
3.3. Partitionierung der Produkt-Indexmenge . . . 37
3.4. Arithmetik von $\mathcal{H}$-Bäumen . . . 38

4. Arithmetik Hierarchischer Matrizen **42**
4.1. Definitionen und Notationen . . . 42
4.2. Konvertierung . . . 42
4.2.1. Bestapproximation und Approximation . . . 42
4.2.2. Hierarchische Approximation . . . 44
4.3. Addition . . . 47
4.4. Multiplikation . . . 48
4.5. Inversion . . . 51
4.5.1. Block-Gauß-Elimination . . . 51
4.5.2. Newton-Iteration . . . 56

4.5.3. Geschachtelte Iteration 58
4.6. Normen 62

5. Komplexität für allgemeine Hierarchien **67**
5.1. Speicherbedarf 67
5.2. Auswertung 75
5.3. Bestapproximation, Approximation und hierarchische Approximation . . . 76
5.4. Addition 78
5.5. Multiplikation 79
5.6. Spektral- und Frobeniusnorm 87

6. Adaptive Arithmetik **88**
6.1. Grundlagen 88
6.2. Konvertierung 91
6.3. Addition und Multiplikation 91
6.4. Inversion 92

7. Approximationseigenschaft **95**
7.1. Notwendige und Hinreichende Bedingungen 95
7.2. Fredholmsche Integraloperatoren 96
7.3. Differentialoperatoren 100

8. Anwendungen **107**
8.1. Referenzproblem: Einfachschichtpotential 107
8.1.1. Approximation der Kernfunktion durch eine Taylorentwicklung . . 107
8.1.2. Dirichlet-Randwertaufgabe als Integralgleichung 1. Art für das Einfachschichtpotential 109
8.2. Partielle Differentialgleichungen 110
8.2.1. Das Modellproblem: Poisson-Gleichung 111
8.2.2. Ein nicht uniformes Gitter 113
8.2.3. Eine Bilinearform mit nicht konstanten Koeffizienten 114
8.3. Matrixgleichungen 117
8.3.1. Linear-quadratisches Kontrollproblem 117
8.3.2. Modellproblem: Wärmeleitungsgleichung 118
8.3.3. Algebraische Matrix-Riccati-Gleichung 119

Fazit **130**

A. Implementierung von $\mathcal{H}$-Matrizen in der Programmiersprache C **131**
A.1. Vorwort 131
A.2. Full-Matrix und $\mathcal{R}k$-Matrix 132
A.3. $\mathcal{H}$-Matrix 139
A.4. Newton-Iteration zur Berechnung von sign(M) 146

Index und Symbolverzeichnis **150**

Abstract

The modeling of physical properties often leads to the task of solving partial differential equations or integral equations. The results of some discretisation and linearisation process are matrix equations or linear systems of equations with special features. In the case of partial differential equations one exploits the local character of the differentiation by using some finite element method or finite difference scheme and gains a sparse system matrix.
In the case of (nonlocal) integral operators low rank approximations seem to be the method of choice. These are either given explicitly by some multipole method or panel clustering technique or implicitly by rank revealing decompositions.
Both types of matrices can be represented as so-called $\mathcal{H}$-matrices. In this thesis we investigate algorithms that perform the addition, multiplication and inversion of $\mathcal{H}$-matrices approximately. Under moderate assumptions the complexity of these new arithmetics is almost linear (linear up to logarithmic terms of order 1 to 3). The arithmetic operations can be performed adaptively, that is up to some given accuracy ε the relative error of the operations is zero.
The question arises under which circumstances the inverse of an $\mathcal{H}$-matrix can be approximated by an $\mathcal{H}$-matrix. For the techniques used in this thesis we need very restrictive assumptions, but the numerical examples in the last part indicate that the approximability does not depend on these assumptions.

Zusammenfassung

Die Modellierung physikalischer Probleme führt oft zu der Aufgabe, partielle Differentialgleichungen oder Integralgleichungen zu lösen. Eine Diskretisierung und Linearisierung dieser Gleichungen ergibt Matrixgleichungen bzw. Gleichungssysteme, die eine spezielle Struktur aufweisen. Bei partiellen Differentialgleichungen läßt sich der lokale Charakter der Differentiation mittels Finite-Element-Methoden, Differenzenverfahren oder ähnlicher Methoden ausnutzen, so daß man schwachbesetzte Matrizen erhält. Für Integralgleichungen haben sich Niedrigrang-Approximationen, die explizit durch Multipolmethoden oder Paneel-Clusterung, oder implizit durch rangweisende Zerlegungen erzeugt werden, als geeignete Verfahren erwiesen. Beide Matrixtypen lassen sich als sogenannte $\mathcal{H}$-Matrizen darstellen, für die in dieser Arbeit unter anderem die Operationen „Addition“, „Multiplikation“ und „Inversion“ einer approximativen (inexakten) Arithmetik untersucht werden. Es stellt sich heraus, daß unter moderaten Annahmen der Aufwand für diese neue Arithmetik fast linear (linear bis auf logarithmische Terme der Ordnung 1 bis 3) ist. Die Arithmetik läßt sich zudem adaptiv durchführen, d.h. zu einem vorgegebenen Approximationsfehler ε werden die Operationen bis auf einen relativen Fehler von ε exakt ausgeführt.
Die Darstellbarkeit der Inversen einer $\mathcal{H}$-Matrix kann nur unter sehr restriktiven Bedingungen gezeigt werden, in den numerischen Tests im letzten Abschnitt zeigt sich allerdings, daß zumindest bei Modellproblemen die Darstellbarkeit auch ohne diese Bedingungen möglich ist.

Danksagung

Ich danke

- Prof. Dr. Dr. h.c. Wolfgang Hackbusch für die Vergabe des Themas sowie seine zahlreichen neuen Anregungen und Ideen,
- Dr. Steffen Börm und Priv. Doz. Dr. Birgit Faermann für ihre Geduld beim Korrekturlesen und den Versuch, die Beweise in der Rohfassung zu verstehen,
- Jens Burmeister und allen Mitarbeitern des Lehrstuhls Praktische Mathematik an der CAU Kiel für die Unterstützung und gute Zusammenarbeit.

Teile dieser Arbeit sind während meiner Aufenthalte am Max-Planck-Institut für Mathematik in den Naturwissenschaften in Leipzig entstanden, motivierende neue Denkanstöße habe ich am Mathematischen Forschungsinstitut Oberwolfach erhalten und finanziell unterstützt wurde die Arbeit von der Deutschen Forschungsgemeinschaft im Rahmen des Projektes *Schnelle approximative Matrixoperationen* und des Graduiertenkollegs *Effiziente Algorithmen und Mehrskalenmethoden.*

1. Einführung an einem Beispiel

Erik Ivar Fredholm
1866-1927

Die von Erik Ivar Fredholm bereits 1900 in „Sur une nouvelle methode pour la resolution du probleme de Dirichlet“ behandelten und nach ihm benannten Fredholmschen Integralgleichungen zweiter Art

$$u(x) = \int_\Omega g(x,y)u(y)\,\mathrm{dy} + f(x) \quad \forall x \in \Omega \tag{1}$$

über dem Integrationsbereich $\Omega \subset \mathbb{R}^d$ (f, g gegeben, u ist gesucht) möchten wir für bestimmte sogenannte Kernfunktionen g lösen. Integralgleichungen dieser Art treten zum Beispiel bei der Überführung einer partiellen Differentialgleichung (auf einem Gebiet im $\mathbb{R}^d$) in eine Randintegralgleichung (auf der $d-1$-dimensionalen Oberfläche) auf. In Operatorschreibweise erhält man aus (1) die Gleichung $(I - \mathcal{K})u = f$, welche mit einer geeigneten Diskretisierungsmethode durch endlichdimensionale Probleme approximiert wird. Im allgemeinen führt eine Diskretisierung der (Rand-)Integralgleichung auf ein vollbesetztes Gleichungssystem, da der Operator $\mathcal{K}$, anders als bei Differentialoperatoren, nicht lokal ist. Spezielle Techniken (Fast Multipole, Paneel-Clusterung, Wavelets) sind erforderlich, um die Berechnung des diskreten Operators effizient durchzuführen. Die hier vorgestellte Methode ist verwandt mit der Paneel-Clusterung, deren grundlegende Idee im folgenden anhand eines einfachen Beispiels illustriert werden soll.

1.1. Die Kernfunktion

Die in der Randelementmethode auftretenden Kernfunktionen sind auf der Diagonalen ($x = y$) meist singulär (aber in einem geeigneten Sinne integrierbar) und weiter entfernt von der Diagonalen sehr glatt. Diese Eigenschaft nutzen wir in Abschnitt 1.3 zur Konstruktion einer Approximation von $\mathcal{K}$ aus, die ähnliche Eigenschaften wie ein lokaler Operator hat. Wir fixieren für die Einführung die Kernfunktion

$$g : \mathbb{R}^2 \to \mathbb{R}, \quad (x,y) \mapsto \log|x-y|, \tag{2}$$

welche für alle (x,y) mit $x \neq y$ unendlich oft differenzierbar ist:

$$\partial_1^j g(x,y) = (j-1)!\frac{1}{(x-y)^j}(-1)^{j-1},$$

$$\partial_2^j g(x,y) = -(j-1)!\frac{1}{(x-y)^j}.$$

Die Glattheit der partiellen Ableitungen $\iota \in \{1,2\}$ wird beschrieben durch

$$|\partial_\iota^j g(x,y)| \leq j!\frac{1}{|x-y|^j}, \quad j \in \mathbb{N}. \tag{3}$$

1.2. Galerkinverfahren

Bei der Diskretisierung mit dem Galerkinverfahren macht man den Ansatz

$$u(y) = \sum_{i=1}^{n} \phi_i(y) u_i \qquad \phi_i = \text{i-te Basisfunktion}$$

für die Lösung u und löst die schwache Formulierung

$$(\phi_l, (I - \mathcal{K})u)_{L^2(\Omega)} = (\phi_l, f)_{L^2(\Omega)} \qquad \forall l = 1, \ldots, n$$

in dem n-dimensionalen Raum $X_n := \text{span}\{\phi_1, \ldots, \phi_n\}$. Dazu müssen die Matrixeinträge

$$K_{li} = \int_\Omega \int_\Omega \phi_l(x) g(x,y) \phi_i(y) \, \mathrm{dy} \, \mathrm{dx}$$

berechnet werden. Die Integration über $\Omega \times \Omega$ möchten wir aufspalten in eine Integration über $\text{supp}\, \phi_l$ und eine über $\text{supp}\, \phi_i$. Dazu brauchen wir eine Darstellung oder Approximation von $g(x,y)$ in der Form

$$g(x,y) = \sum_{j \in J} g_1^j(x) g_2^j(y).$$

Kernfunktionen g, welche diese Darstellung besitzen, nennt man entartete oder ausgeartete Kerne. Offenbar ist die hier betrachtete Kernfunktion (2) kein entarteter Kern.

1.3. Approximation durch einen entarteten Kern

Eine Möglichkeit, eine Kernfunktion g durch einen entarteten Kern zu approximieren, besteht darin, ihn durch eine Taylorentwicklung endlicher Ordnung anzunähern. Dabei führt man die Taylorentwicklung entweder für die erste Variable x oder für die zweite Variable y durch. Wir beschränken uns hier auf den ersten Fall. Ersetzt man die Kernfunktion g durch ihre Taylorentwicklung

$$g_e(x,y) := \sum_{j=0}^{\mu-1} \frac{1}{j!} \partial_1^j g(x_0, y)(x - x_0)^j \tag{4}$$

bzgl. der ersten Variable in x_0 bis zur Ordnung μ, so entsteht ein Fehler

$$\begin{aligned} |g(x,y) - g_e(x,y)| &\leq \frac{1}{\mu!} |x - x_0|^\mu \max\{|\partial_1^\mu g(\xi, y)| \mid \xi \in [x_0, x]\} \\ &\overset{(3)}{\leq} \frac{1}{\mu!} |x - x_0|^\mu \mu! \max\left\{ \frac{1}{|\xi - y|^\mu} \Big| \xi \in [x_0, x] \right\} \\ &= \max\left\{ \left(\frac{|x - x_0|}{|\xi - y|} \right)^\mu \Big| \xi \in [x_0, x] \right\}. \end{aligned}$$

Um eine mit der Entwicklungsordnung μ besser werdende Approximation zu erreichen, muß $\frac{|x-x_0|}{|\xi-y|} < 1$ gelten.

1.4. Zulässigkeitsbedingung

Wir nennen ein Mengenprodukt $\sigma \times \tau \subset \mathbb{R}^2$ η-zulässig, falls die Zulässigkeitsbedingung

$$\min\{\operatorname{diam}(\sigma), \operatorname{diam}(\tau)\} \leq 2\eta \operatorname{dist}(\sigma, \tau) \tag{5}$$

erfüllt ist. Die Taylorentwicklung führen wir bzgl. der Variable aus der Menge mit dem kleineren Durchmesser (Länge des kürzesten Intervalls, das die Menge enthält) durch, o.B.d.A sei dies σ. Ist x_0 der Mittelpunkt von σ (Mittelpunkt des Intervalls) und $(x, y) \in \sigma \times \tau$, so gilt für alle $\xi \in [x_0, x]$:

$$\frac{|x - x_0|}{|\xi - y|} \leq \frac{\frac{1}{2}\operatorname{diam}(\sigma)}{\operatorname{dist}(\sigma, \tau)} \leq \eta.$$

Setzen wir $\eta < 1$ voraus, so erhalten wir eine bzgl. der Entwicklungsordnung μ exponentielle Konvergenz des Taylorpolynoms gegen die Kernfunktion. Wir erhalten also direkt aus der Zulässigkeitsbedingung die Approximationseigenschaft, welche mit kleiner werdendem η besser wird. Ein kleineres η verschärft allerdings die Zulässigkeitsbedingung, so daß man weniger zulässige Produkte $\sigma \times \tau$ findet.

1.5. Darstellung zulässiger Blöcke im Galerkinverfahren

Für diesen Abschnitt sei ein η-zulässiges Produkt $\sigma \times \tau$ sowie eine Entwicklungsordnung μ fixiert. Der entartete Kern g_e hat auf $\sigma \times \tau$ die Darstellung (4). Für Basisfunktionen mit $\operatorname{supp} \phi_l \subset \sigma$ und $\operatorname{supp} \phi_i \subset \tau$ kann man g durch den entarteten Kern (4) approximieren und erhält mit den Bezeichnungen

$$\begin{aligned} a_l^{(j,\sigma,\tau)} &:= \int_\sigma (x - x_0)^j \phi_l(x) \, \mathrm{dx} \\ b_i^{(j,\sigma,\tau)} &:= \int_\tau \frac{1}{j!} \partial_1^j g(x_0, y) \phi_i(y) \, \mathrm{dy} \end{aligned}$$

die Rang-μ-Darstellung

$$K_{li} = \sum_{j=0}^{\mu-1} a_l^{(j,\sigma,\tau)} b_i^{(j,\sigma,\tau)}, \qquad K|_{I_\sigma \times I_\tau} = \sum_{j=0}^{\mu-1} a^{(j,\sigma,\tau)} (b^{(j,\sigma,\tau)})^T,$$

von K auf der Produkt-Indexmenge $I_\sigma \times I_\tau := \{(l, i) \mid \operatorname{supp} \phi_l \subset \sigma, \operatorname{supp} \phi_i \subset \tau\}$.
Eine Rang-μ-Matrix $(K_{li})_{l \in I_\sigma, i \in I_\tau}$ hat zwei Eigenschaften, die wir ausnutzen werden: Zum einen sind nur $\mu \cdot (|I_\sigma| + |I_\tau|)$ Einträge zur Darstellung der Matrix zu berechnen (im Gegensatz zu $|I_\sigma| \cdot |I_\tau|$ bei einer vollbesetzten Darstellung) und zum anderen lassen sich die grundlegenden Matrixoperationen (Auswertung, Multiplikation, Addition) mit einem deutlich geringeren Aufwand als bei vollbesetzter Darstellung realisieren.

1.6. Konstruktion zulässiger Blöcke

Wir wollen zu einer gewählten Zulässigkeitsbedingung (5) möglichst große zulässige Produkte $\sigma \times \tau$ identifizieren. Dazu fixieren wir für den Rest der Einführung eine Diskretisierung mit dem Galerkinverfahren bei stückweise konstanten Basisfunktionen auf dem regelmäßig in $n = 2^p$ Teilintervalle unterteilten Gebiet $\Omega := [0, 1]$ und setzen den Parameter der Zulässigkeitsbedingung auf $\eta := 0.5$.
Aus der Menge aller möglichen Teilmengen $\sigma \times \tau$ von $\Omega \times \Omega$ müssen wir geeignete Kandidaten auf Zulässigkeit testen (alle zu testen wäre zu aufwendig). Ziel ist es, eine Partitionierung (**bis auf Randpunkte**) von $\Omega \times \Omega$ zu finden, welche aus möglichst vielen großen (zulässigen) Produkten und wenigen kleinen nicht zulässigen, welche dann vollbesetzt dargestellt werden, besteht. Die Elemente der Partition sollen aus der Vereinigung der Träger einer Teilmenge der Ansatzfunktionen bestehen. Eine Methode, eine

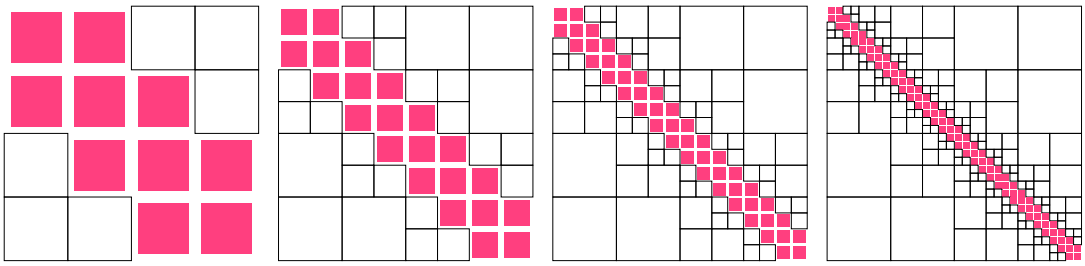


Abbildung 1: Partitionierungen für $p = 2, 3, 4, 5$. Der Punkt $(0, 0)$ liegt jeweils oben links, $(1, 1)$ unten rechts.

Partitionierung zu erzeugen, ist zum Beispiel die sukzessive Bisektion von Ω: Zuerst wird die Zulässigkeit für $[0, 1] \times [0, 1]$ getestet, offenbar ist dieses Mengenprodukt nicht zulässig. $\Omega_1^{(0)} := [0, 1]$ wird unterteilt in $\Omega_1^{(1)} := [0, \frac{1}{2}]$ und $\Omega_2^{(1)} := [\frac{1}{2}, 1]$. Die Zulässigkeit wird für $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$, $[0, \frac{1}{2}] \times [\frac{1}{2}, 1]$, $[\frac{1}{2}, 1] \times [0, \frac{1}{2}]$ und $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$ getestet: Die Mengen haben jeweils den Abstand Null zueinander, sind also nicht zulässig. Wieder werden alle beteiligten Mengen unterteilt in

$$\Omega_i^{(2)} := [\frac{i-1}{4}, \frac{i}{4}], \quad i = 1, \ldots, 4$$

und paarweise auf Zulässigkeit getestet. Wie man leicht sieht, sind genau die Produkte $\Omega_i^{(2)} \times \Omega_j^{(2)}$ mit $|i-j| > 1$ zulässig, sie werden nicht weiter betrachtet. Die nicht zulässigen werden immer weiter unterteilt bis wir nach $p = \log_2(n)$ Schritten bei den sogenannten Paneelen

$$\Omega_i^{(p)} := [\frac{i-1}{n}, \frac{i}{n}], \quad i = 1, \ldots, n$$

angekommen sind. Diese Mengen werden nicht weiter unterteilt. Die so generierten Partitionierungen von $[0, 1] \times [0, 1]$ sind in Abbildung 1 für $p = 2, 3, 4, 5$ zu sehen, wobei die nicht zulässigen Paneel-Produkte eingefärbt wurden.

1.7. Definition der Arithmetik

Bislang haben wir eine (ungeordnete) Partitionierung aus zulässigen und nicht zulässigen Mengenprodukten $\sigma \times \tau$ generiert. Die Auswertung der Matrix $(x \mapsto Kx)$ läßt sich so bereits realisieren, für komplexere Operationen, wie zum Beispiel die Multiplikation zweier Matrizen, ist eine besser strukturierte Verwaltung der Partition jedoch vorteilhaft. Es bietet sich hier eine hierarchische Partitionierung von Ω und $\Omega \times \Omega$ in Form von Baumstrukturen T_Ω und $T_{\Omega\times\Omega}$ an.

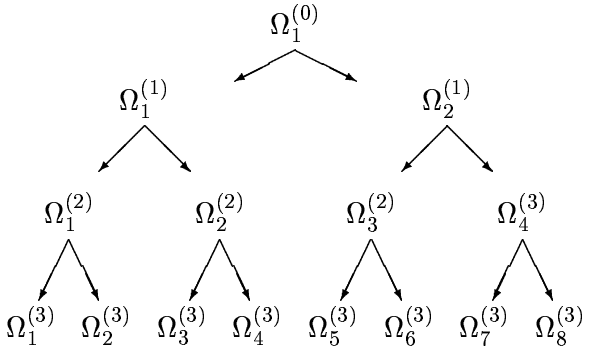


Abbildung 2: Der Baum T_Ω für $p = 3$

T_Ω ist durch die folgenden beiden Bedingungen charakterisiert:

- Die Wurzel von T_Ω ist $\Omega_1^{(0)} = [0, 1]$.
- Jeder Knoten $\Omega_i^{(l)}$, $1 \le i \le 2^l, 0 \le l \le p$, von T_Ω ist entweder ein Paneel und Blatt $(l = p)$ oder die Vereinigung seiner Söhne $\Omega_{2i-1}^{(l+1)}, \Omega_{2i}^{(l+1)}$.

Ebenso läßt sich $T_{\Omega\times\Omega}$ charakterisieren durch

- $\Omega_1^{(0)} \times \Omega_1^{(0)} = [0, 1] \times [0, 1]$ ist die Wurzel von $T_{\Omega\times\Omega}$;
- Jeder Knoten $\Omega_i^{(l)} \times \Omega_j^{(l)}$ des Baumes ist entweder ein Blatt (zulässig oder $l = p$) oder die Vereinigung seiner Söhne
  $\Omega_{2i-1}^{(l+1)} \times \Omega_{2i-1}^{(l+1)}$, $\Omega_{2i-1}^{(l+1)} \times \Omega_{2i}^{(l+1)}$, $\Omega_{2i}^{(l+1)} \times \Omega_{2i-1}^{(l+1)}$ und $\Omega_{2i}^{(l+1)} \times \Omega_{2i}^{(l+1)}$.

Es ist zu beachten, daß in T_Ω alle Paneele $\Omega_i^{(p)}$, $i = 1, \ldots, n$, enthalten sind (insgesamt $2n-1$ Knoten), während in $T_{\Omega\times\Omega}$ nur Produkte von Paneelen enthalten sind, deren sämtliche Vorfahren nicht zulässig waren (der komplette Baum mit allen Paneel-Produkten $\Omega_i^{(p)} \times \Omega_j^{(p)}$, $i, j = 1, \ldots, n$, würde mehr als n^2 Elemente enthalten).
Den Gebieten $\Omega_i^{(l)} \subset \Omega$ entsprechen die Indexmengen

$$I_i^l := \left\{ i \in \{1, \ldots, n\} \mid \operatorname{supp} \phi_i \subset \Omega_i^{(l)} \right\} = \left\{ (i-1)\frac{n}{2^l} + 1, \ \ldots \ , (i-1)\frac{n}{2^l} + \frac{n}{2^l} \right\}$$

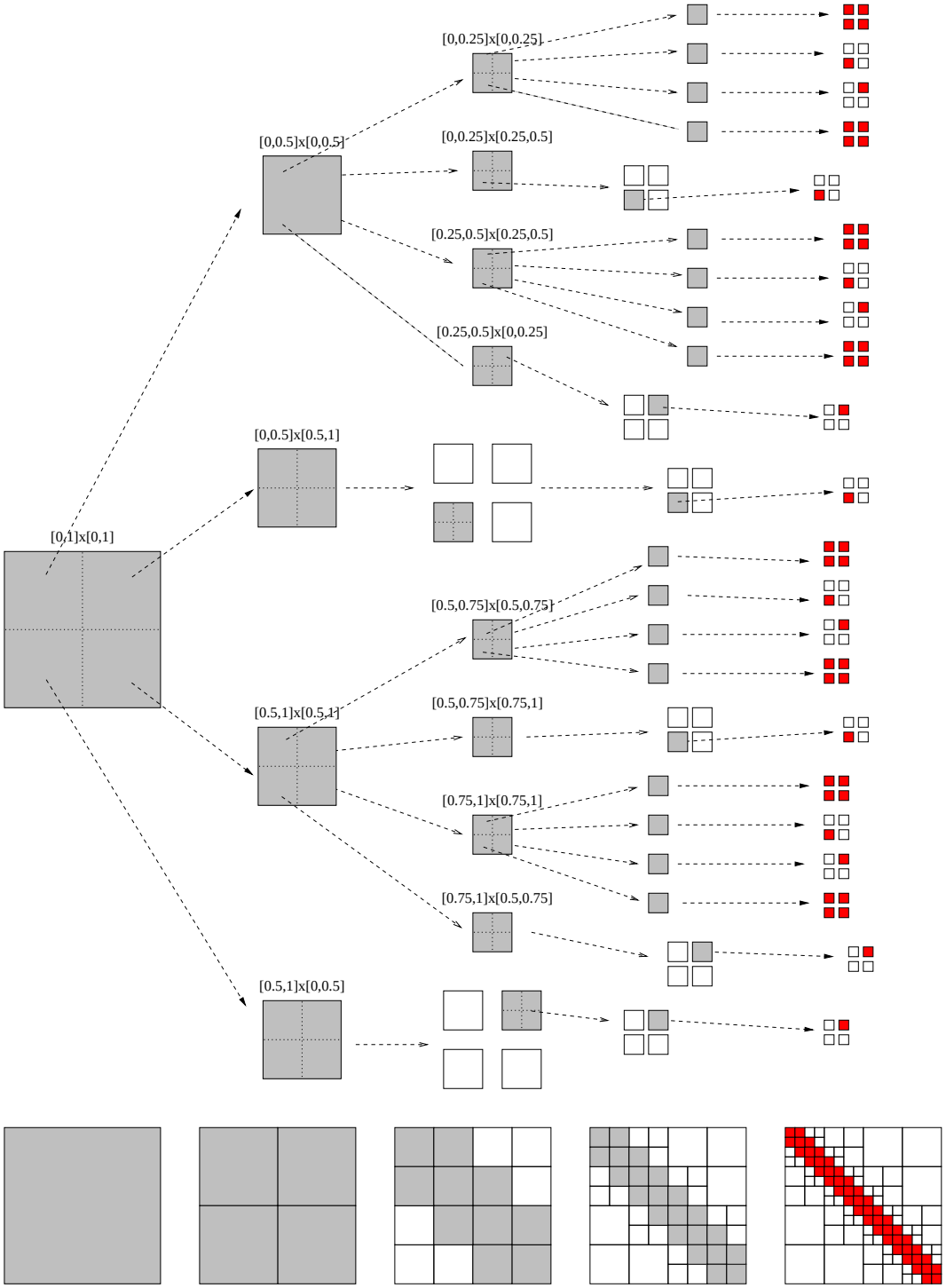


Abbildung 3: Hierarchische Partitionierung von $[0,1] \times [0,1]$ für $p = 4$

der Indizes von Basisfunktionen, deren Träger in $\Omega_i^{(l)}$ enthalten ist. Analog zu den Bäumen T_Ω und $T_{\Omega\times\Omega}$ erhalten wir die Bäume T_I und $T_{I\times I}$ zu den Indexmengen $I = \{1,\ldots,n\}$ und $I \times I$.
Die Menge der Hierarchischen Matrizen ($\mathcal{H}$-Matrizen) zu dem Baum $T_{I\times I}$ und einem vorgegebenen Rang $k \in \mathbb{N}_0$ ist definiert als

$$\mathcal{M}_{\mathcal{H},k}(T_{I\times I}) := \{M \in \mathbb{K}^{I\times I} \mid \operatorname{rang}(M_b) \leq k \quad \forall b \in \mathcal{L}^+(T_{I\times I})\},$$

wobei $\mathcal{L}^+(T_{I\times I})$ die zulässigen Blätter des Baumes bezeichnet und M_b die mit $b \subset I \times I$ korrespondierende Untermatrix von M ist. Die arithmetischen Operationen lassen sich per Induktion über die Blockgröße $|I_i^l| = 2^{p-l}$ definieren. Wir nehmen an, daß für alle zulässigen Blätter des Baumes $T_{I\times I}$ bereits die arithmetischen Operationen etabliert sind (Rang-k-Arithmetik, k der maximale Rang in den zulässigen Blöcken), sowie für die nicht zulässigen Blätter (1×1-Matrizen) die herkömmliche Arithmetik verwendet wird. Seien also

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \qquad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

drei $2^{p-l+1} \times 2^{p-l+1}$-Matrizen mit $2^{p-l} \times 2^{p-l}$-Untermatrizen A_{ij}, B_{ij}, C_{ij}, $i,j \in \{1,2\}$. Dann definieren wir die *formatierten* Operationen in der Menge $\mathcal{M}_{\mathcal{H},k}(T_{I\times I})$ als

- Auswertung

$$Ax = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix}$$

- Addition

$$C = A \oplus B := \begin{bmatrix} A_{11} \oplus B_{11} & A_{12} \oplus B_{12} \\ A_{21} \oplus B_{21} & A_{22} \oplus B_{22} \end{bmatrix}$$

- Multiplikation

$$C = A \odot B := \begin{bmatrix} A_{11} \odot B_{11} \oplus A_{12} \odot B_{21} & A_{11} \odot B_{12} \oplus A_{12} \odot B_{22} \\ A_{21} \odot B_{11} \oplus A_{22} \odot B_{12} & A_{21} \odot B_{12} \oplus A_{22} \odot B_{22} \end{bmatrix}$$

über die blockweisen Operationen. Die Auswertung entspricht der sukzessiven Anwendung des Operators in den Blättern. Die Addition wird ebenfalls für alle Blätter unabhängig voneinander durchgeführt, es ist allerdings nicht vorausgesetzt, daß die Rang-k-Addition in den zulässigen Blättern gleich der exakten Addition in den Blättern ist (der Rang erhöht sich), daher verwenden wir das Symbol $\oplus$ für die Addition und implizieren damit, daß nach der Addition eine Projektion auf die Struktur von C stattfindet. Die Multiplikation wird aus demselben Grund mit $\odot$ bezeichnet ($\odot$ hat eine höhere Priorität als $\oplus$). Die Addition $\oplus$ wird eine Bestapproximation bezüglich der Frobeniusnorm von $+$ in der Menge $\mathcal{M}_{\mathcal{H},k}(T_{I\times I})$ werden, die Multiplikation $\odot$ erfüllt dies mit einigen Modifikationen ebenfalls, daher sind die arithmetischen Operationen in diesem Sinne optimal. Die Inversion kann man in ähnlicher Weise definieren (zum Beispiel durch Block-Gauß-Elimination), sie läßt sich aber nicht mehr als Projektion der exakten Inversen auf $\mathcal{M}_{\mathcal{H},k}(T_{I\times I})$ realisieren, ohne daß der Aufwand für die Arithmetik übermäßig zunimmt.

1.8. Aufwand der Arithmetik

Die arithmetischen Operationen $\oplus, \odot$ lassen sich in $\mathcal{M}_{\mathcal{H},k}(T_{I\times I})$ im wesentlichen mit einem Aufwand von $O(n\log(n)^{c_{\oplus/\odot}}k^2)$ realisieren (logarithmisch-linear). Für die formatierte Addition ist $c_\oplus = 1$, für die formatierte Multiplikation $c_\odot = 2$. Dies kann man für die Addition recht leicht unter der Annahme beweisen, daß auf jeder Stufe $l = 0, \ldots, p$ des Baumes $T_{I\times I}$ höchstens $\kappa 2^l$ Blätter vorhanden sind. Dazu bezeichne $N_{\mathrm{R}k,\oplus}(j)$ den Aufwand der Addition zweier Rang-k-Matrizen der Größe j. Der Aufwand $N_{\mathcal{M}_{\mathcal{H},k},\oplus}$ der Addition ist dann gerade beschränkt durch

$$\sum_{l=0}^{p} \kappa 2^l N_{\mathrm{R}k,\oplus}(2^{p-l}).$$

Wie wir später sehen werden, ist $N_{\mathrm{R}k,\oplus}(j) \leq Ck^2 j$, also

$$N_{\mathcal{M}_{\mathcal{H},k},\oplus} \leq \sum_{l=0}^{p} \kappa 2^l Ck^2 2^{p-l} = \kappa C(p+1)k^2 n = O(k^2 n\log(n)).$$

Alternativ kann man direkt über Rekursionsformeln den Aufwand der Arithmetik für Modell-Partitionierungen bestimmen und erhält so konkrete Angaben über die benötigten Gleitkommaoperationen. Genauso wie der Aufwand der Addition ist der Speicherverbrauch auszurechnen: Da der Bedarf an Speicherplatz für eine $j \times j$-Rang-k-Matrix $O(kj)$ ist, ist der Speicherbedarf zur Darstellung einer Matrix aus $\mathcal{M}_{\mathcal{H},k}(T_{I\times I})$

$$N_{\mathcal{M}_{\mathcal{H},k},St} = O(kn\log(n)).$$

1.9. Ziel der Arbeit

Ziel dieser Arbeit ist es, die formatierte Arithmetik für allgemeine Baumstrukturen $T_{I\times J}$ zu etablieren. Dazu führen wir im nachfolgenden Abschnitt $\mathbf{R}k$-Matrizen als eine Darstellung für Niedrigrang-Matrizen ein. Abschnitt 3 und 4 befassen sich mit der Konstruktion der Bäume und Algorithmen für die Arithmetik hierarchischer Matrizen. Der Aufwand der (formatierten) Addition, Multiplikation und Inversion bei festgehaltenem Rang wird in Abschnitt 5 abgeschätzt.
Eine Alternative zur Arithmetik für festgehaltenen Rang wird in Abschnitt 6 vorgestellt. Dort wird der Rang in den zulässigen Blöcken der Matrix „adaptiv“ bestimmt.
Die Frage, welche Matrizen sich als $\mathcal{H}$-Matrizen darstellen lassen,wird in Abschnitt 7 angegangen.
Im letzten Teil der Arbeit soll die Anwendbarkeit der $\mathcal{H}$-Arithmetik auf verschiedene Probleme im Bereich der Diskretisierung elliptischer Randwertaufgaben studiert werden. Es werden dabei numerische Tests zur Diskretisierung und Inversion von Integraloperatoren und Differentialoperatoren sowie zur Auflösung von Matrixgleichungen durchgeführt.

2. Rk-Matrizen

Dieser Abschnitt kann als eigenständiges Kapitel unabhängig von dem Thema „Hierarchische Matrizen“ angesehen werden. Wir bestimmen die Komplexität der Operationen „Auswertung“, „Multiplikation“ und „Addition“ für $\mathbf{R}k$-Matrizen und werden dabei auf die Singulärwertzerlegung einer $\mathbf{R}k$-Matrix sowie deren Approximation eingehen.
Für den gesamten Abschnitt seien $n, m \in \mathbb{N}$ und $k \in \mathbb{N}_0$ vorgegeben.

2.1. Definitionen und Grundlagen

Definition 2.1 *($\mathbf{R}_{\leq k}$-Matrix)*
Eine Matrix $M \in \mathbb{R}^{n,m}$ bezeichnen wir als eine $\mathbf{R}_{\leq k}$-Matrix, wenn ihr Rang höchstens k ist. Die Menge aller $n \times m$-$\mathbf{R}_{\leq k}$-Matrizen bezeichnen wir mit $\mathbf{R}_{\leq k}(n, m)$.

Bemerkung 2.2 *(Idealeigenschaft von $\mathbf{R}_{\leq k}(n, m)$)*
$(\mathbf{R}_{\leq k}(n, m), +)$ ist keine Untergruppe von $(\mathbb{R}^{n,m}, +)$, da sie bezüglich der Addition nicht abgeschlossen ist, aber $\mathbf{R}_{\leq k}(n, m)$ ist ein Ideal von $(\mathbb{R}^{n,m}, \cdot)$.

Für praktische Zwecke reicht nicht allein die Kenntnis vom Rang einer Matrix aus, auch ihre Darstellung (z.B. auf einem Rechner) ist wichtig. Der Wechsel zwischen verschiedenen Darstellungen (LU-Zerlegung, Singulärwertzerlegung, schwachbesetzt oder vollbesetzt) kann manchmal sehr aufwendig oder praktisch nicht realisierbar sein. Eine Darstellung der $\mathbf{R}_{\leq k}$-Matrizen sind die $\mathbf{R}k$-Matrizen.

Definition 2.3 *($\mathbf{R}k$-Matrix)*
Eine Matrix $M \in \mathbb{R}^{n,m}$ bezeichnen wir als eine $\mathbf{R}k$-Matrix, wenn sie in der Darstellung

$$M = \sum_{i=1}^{k} a^i (b^i)^T \tag{6}$$

mit $a^i \in \mathbb{R}^n$, $b^i \in \mathbb{R}^m$ $(i = 1, \ldots, k)$ vorliegt. Die Menge aller $n \times m$-$\mathbf{R}k$-Matrizen bezeichnen wir mit $\mathbf{R}k(n, m)$. Seien I, J zwei Mengen und $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. Eine Matrix $M \in \mathbb{K}^{I \times J}$ wird als $\mathbf{R}k(I \times J)$-Matrix bezeichnet, wenn sie in der Darstellung (6) mit $a^i \in \mathbb{K}^I$, $b^i \in \mathbb{K}^J$ $(i = 1, \ldots, k)$ vorliegt.

Bemerkung 2.4 *(Indexmengen und Körper)*
Ist $|I| = n$ und $|J| = m$, so wird davon ausgegangen, daß eine Bijektion zwischen den Indexmengen I, J und den Abschnitten der natürlichen Zahlen vorliegt, so daß im folgenden nur noch $\mathbf{R}k(n, m)$-Matrizen betrachtet werden.
Auf die Behandlung komplexwertiger Matrizen wird in Abschnitt 2.7 eingegangen, im folgenden behandeln wir nur reellwertige Matrizen.

Bemerkung 2.5 *($\mathbf{R}k(V, W)$-Matrizen)*
Sei V ein k-dimensionaler Unterraum des $\mathbb{R}^n$ und W ein k-dimensionaler Unterraum von $\mathbb{R}^m$. Die Menge der $\mathbf{R}k(V, W)$-Matrizen ist dann definiert als die Menge der $\mathbf{R}k$-Matrizen, deren Vektoren aus der Darstellung (6) die Bedingung $a^i \in V$, $b^i \in W$ für $i =$

$1, \ldots, k$ *erfüllen.* $\mathbf{R}k(V, W)$ *ist bzgl. der Addition abgeschlossen und damit (im Gegensatz zu* $\mathbf{R}k(n, m)$*) eine Untergruppe von* $(\mathbb{R}^{n,m}, +)$.

Bemerkung 2.6 *(Singulärwertzerlegung)*
Eine Matrix $M \in \mathbb{R}^{n,m}$, $l := \operatorname{rang}(M)$, *welche in Singulärwertzerlegung*

$$M = U\Sigma V^T$$

mit $\Sigma = \operatorname{diag}\{\sigma_1, \ldots, \sigma_l, 0, \ldots, 0\} \in \mathbb{R}^{n,m}$ *und unitären* $U \in \mathbb{R}^{n,n}, V \in \mathbb{R}^{m,m}$ *gegeben ist, ist insbesondere eine* $\mathbf{R}l(n, m)$*-Matrix*

$$M = \sum_{i=1}^{l} u^i \sigma_i (v^i)^T$$

mit den Spaltenvektoren u^i *von* U *und den Spaltenvektoren* v^i *von* V.

Bemerkung 2.7 *(*$\mathbf{R}k$*-Darstellung allgemeiner Matrizen)*
Eine Matrix $M \in \mathbb{R}^{n,m}$, $l := \min(n, m)$, *läßt sich über die Einheitsvektoren* $\{e^i\}_{i=1}^{l}$ *immer auch als* $\mathbf{R}l(n, m)$*-Matrix*

$$M = \sum_{i=1}^{l} a^i (e^i)^T \quad \textit{bzw.} \quad M = \sum_{i=1}^{l} e^i (b^i)^T$$

mit den Spaltenvektoren a^i *bzw. Zeilenvektoren* b^i *von* M *darstellen.*

2.2. Auswertung

Die Matrix-Vektor-Multiplikation einer $\mathbf{R}k(n, m)$-Matrix

$$M = \sum_{i=1}^{k} a^i (b^i)^T$$

mit einem Vektor $x \in \mathbb{R}^m$ von rechts (bzw. $y \in \mathbb{R}^n$ von links) läßt sich als Summation über die k Vektoren a^i (bzw. b^i) mit der Skalierung $(b^i)^T x$ (bzw. $y^T a^i$) schreiben:

$$Mx = \sum_{i=1}^{k} a^i (b^i)^T x = \sum_{i=1}^{k} a^i ((b^i)^T x),$$

$$y^T M = y^T \sum_{i=1}^{k} a^i (b^i)^T = \sum_{i=1}^{k} (y^T a^i)(b^i)^T.$$

Der Aufwand zur Berechnung wird mit $N_{\mathbf{R}k \cdot V}(n, m)$ (bzw. $N_{V \cdot \mathbf{R}k}(n, m)$) bezeichnet und ist

$$\begin{aligned} N_{\mathbf{R}k \cdot V}(n, m) &= 2k(n + m) - k - n \\ N_{V \cdot \mathbf{R}k}(n, m) &= 2k(n + m) - k - m. \end{aligned}$$

2.3. Multiplikation

Zur Multiplikation einer $\mathbf{R}k(n,m)$-Matrix

$$M = \sum_{i=1}^{k} a^i (b^i)^T$$

mit einer beliebigen Matrix F benötigt man lediglich die k-fache Auswertung der Matrix bzw. ihrer Transponierten:

$$\begin{aligned} MF &= (\sum_{i=1}^{k} a^i (b^i)^T) F = \sum_{i=1}^{k} a^i (F^T b^i)^T, \\ FM &= F(\sum_{i=1}^{k} a^i (b^i)^T) = \sum_{i=1}^{k} (Fa^i)(b^i)^T. \end{aligned}$$

Der Aufwand für die Multiplikation einer $\mathbf{R}k(n,m')$- mit einer $\mathbf{R}k(m',m)$-Matrix wird mit $N_{\mathbf{R}k\cdot\mathbf{R}k}(n,m',m)$ bezeichnet und liegt bei

$$N_{\mathbf{R}k\cdot\mathbf{R}k}(n,m',m) = 2k^2(m' + \min\{n,m\}) - k^2 - k\min\{n,m\}.$$

2.4. Darstellungswechsel und Konvertierungen

Wir haben bereits gesehen, daß die $\mathbf{R}_{\leq k}$-Matrizen als $\mathbf{R}k$-Matrizen dargestellt werden können. Es gilt nun, für eine beliebige Matrix M eine Approximation (und Darstellung) in $\mathbf{R}k$ zu finden. Unter der Vielzahl von Algorithmen, welche dies bewerkstelligen, zeichnet sich die gekürzte Singulärwertzerlegung dadurch aus, daß sie sowohl in der Frobeniusnorm als auch in der Spektralnorm eine Bestapproximation von M in der Menge der $\mathbf{R}_{\leq k}$-Matrizen liefert.

2.4.1. Gekürzte Singulärwertzerlegung

Gegeben sei eine $\mathbf{R}_{\leq k'}(n,m)$-Matrix M. Wir fixieren eine Singulärwertzerlegung

$$M = U\Sigma V^T = \begin{bmatrix} u^1 & \cdots & u^n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_{k'} & & \\ & & & 0 & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} (v^1)^T \\ \vdots \\ (v^m)^T \end{bmatrix} \tag{7}$$

von M.

Definition 2.8 *(Gekürzte Singulärwertzerlegung)*
Die zu einer Singulärwertzerlegung (7) und $\tilde{k} \in \{0,\ldots,k'\}$ gehörende Matrix

$$\tilde{M} := U\tilde{\Sigma}V^T = \begin{bmatrix} u^1 & \cdots & u^{\tilde{k}} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{\tilde{k}} \end{bmatrix} \begin{bmatrix} (v^1)^T \\ \vdots \\ (v^{\tilde{k}})^T \end{bmatrix} \tag{8}$$

bezeichnen wir als eine (auf Rang $\tilde{k} \leq k'$) gekürzte Singulärwertzerlegung von M.

Bemerkung 2.9 *(Mehrdeutigkeit der gekürzten Singulärwertzerlegung)*
Eine gekürzte Singulärwertzerlegung $\tilde{M}$ von M hängt von der Wahl der Singulärwertzerlegung (7) ab, falls die Singulärwerte nicht einfach sind.

Satz 2.10 *(Gekürzte Singulärwertzerlegung liefert Bestapproximation)*
Eine auf Rang $\tilde{k}$ gekürzte Singulärwertzerlegung $\tilde{M}$ von M ist eine Bestapproximierende (bezüglich der Spektral- und Frobeniusnorm) von M in der Menge der $\mathbf{R}_{\leq\tilde{k}}$-Matrizen:

$$\begin{aligned}\min_{\text{rang}(B)\leq\tilde{k}} \|M-B\|_2 &= \|M-\tilde{M}\|_2 = \sigma_{\tilde{k}+1},\\ \min_{\text{rang}(B)\leq\tilde{k}} \|M-B\|_F &= \|M-\tilde{M}\|_F = \sqrt{\sum_{i=\tilde{k}+1}^{k'} \sigma_i^2}.\end{aligned}$$

Beweis: (Vgl. [6], Theorem 2.5.3, für 2-Norm)
Im Falle $\tilde{k} = k'$ ist die Behauptung trivial, sei also nun $\tilde{k} < k'$. Es gilt

$$\min_{\text{rang}(B)\leq\tilde{k}} \|M-B\|_2 \leq \|M-\tilde{M}\|_2 = \|U^T(M-\tilde{M})V\|_2 = \|\Sigma-\tilde{\Sigma}\|_2 = \sigma_{\tilde{k}+1},$$

$$\min_{\text{rang}(B)\leq\tilde{k}} \|M-B\|_F \leq \|M-\tilde{M}\|_F = \|U^T(M-\tilde{M})V\|_F = \|\Sigma-\tilde{\Sigma}\|_F = \sqrt{\sum_{i=\tilde{k}+1}^{k'} \sigma_i^2}.$$

Zu zeigen bleibt $\min_{\text{rang}(B)\leq\tilde{k}} \|M-B\| \geq \|M-\tilde{M}\|$. Sei dazu B vorgegeben und eine Singulärwertzerlegung $B = XDY^T$ von B fixiert. Aus $\text{rang}(B) \leq \tilde{k}$ folgt direkt

$$B\text{span}\{y^{\tilde{k}+1},\ldots,y^m\} = \{0\},$$

so daß wir wegen $(m-\tilde{k})+\tilde{k}+1 > m$ einen normierten Vektor

$$0 \neq z^{\tilde{k}+1} \in \text{span}\{y^{\tilde{k}+1},\ldots,y^m\} \cap \text{span}\{v^1,\ldots,v^{\tilde{k}+1}\}$$

finden. Es gilt dann

$$\begin{aligned}\|M-B\|_2 &\geq \|(M-B)z^{\tilde{k}+1}\|_2 = \|Mz^{\tilde{k}+1}\|_2 = \sqrt{\sum_{i=1}^{\tilde{k}+1}(\sigma_i(v^i)^T z^{\tilde{k}+1})^2}\\ &\geq \sigma_{\tilde{k}+1}\sqrt{\sum_{i=1}^{\tilde{k}+1}((v^i)^T z^{\tilde{k}+1})^2} = \sigma_{\tilde{k}+1}.\end{aligned}$$

Damit ist die Behauptung für die Spektralnorm bewiesen. Wir beweisen per Induktion, daß wir ein Orthonormalsystem von Vektoren $z^{\tilde{k}+1},\ldots,z^{k'}$ finden mit

$z^j \in \text{span}\{y^{\tilde{k}+1}, \ldots, y^m\} \cap \text{span}\{v^1, \ldots, v^j\}$ und $\|(M-B)z^j\|_2 \geq \sigma_j$ für $j = \tilde{k}+1, \ldots, k'$. Seien also $z^{\tilde{k}+1}, \ldots, z^{j-1}$ bereits konstruiert (Induktionsanfang s.o.). Der Raum

$$W_j := \text{span}\{y^{\tilde{k}+1}, \ldots, y^m\} \cap \text{span}\{v^1, \ldots, v^j\}$$

erfüllt

$$\begin{aligned} \dim(W_j) &\geq j - \tilde{k}, \\ z^i &\in W_j, \quad , i = \underbrace{\tilde{k}+1, \ldots, j-1}_{j-\tilde{k}-1 \text{ Stück}}. \end{aligned}$$

Daher finden wir einen Vektor $z^j \in W_j$, der orthonormal zu den vorigen z^i ist. Dieser Vektor erfüllt dann

$$\|(M-B)z^j\|_2 = \|Mz^j\|_2 = \sqrt{\sum_{i=1}^{j} (\sigma_i (v^i)^T z^j)^2} \geq \sigma_j \sqrt{\sum_{i=1}^{j} ((v^i)^T z^j)^2} = \sigma_j.$$

Das Orthonormalsystem $z^{\tilde{k}+1}, \ldots, z^{k'}$ wird zu einer Orthonormalbasis $z^1, \ldots, z^n$ ergänzt. Dann ist $Z := [z^1 \cdots z^n]$ unitär und es folgt die Behauptung für die Frobeniusnorm:

$$\begin{aligned} \|M-B\|_F &= \|(M-B)Z\|_F \\ &= \sqrt{\sum_{i=1}^{n} \|(M-B)Ze^i\|_2^2} \qquad {\scriptstyle e^i = \text{i-ter Einheitsvektor}} \\ &\geq \sqrt{\sum_{i=\tilde{k}+1}^{k'} \|(M-B)z^i\|_2^2} \geq \sqrt{\sum_{i=\tilde{k}+1}^{k'} \sigma_i^2}. \end{aligned}$$

■

Bemerkung 2.11 *(Aufwand der (gekürzten) Singulärwertzerlegung)*
Im allgemeinen benötigt man zur Berechnung der Singulärwertzerlegung einer $n \times m$-Matrix $O(n^3 + m^3)$ Operationen (unter der Annahme, daß nach wenigen Golub-Kahan-SVD-Schritten jeweils ein Nebendiagonalelement gegen Null konvergiert, siehe [6]).
Für die Projektion einer beliebigen Matrix $M \in \mathbb{R}^{n,m}$ auf $\mathbf{R}k(n,m)$ wäre es nun interessant, eine Methode zur direkten Berechnung der auf Rang k gekürzten Singulärwertzerlegung zu haben, welche einen geringeren Aufwand als die gesamte Singulärwertzerlegung hat. Für beliebige Matrizen ist dafür zur Zeit noch kein Verfahren bekannt, wenngleich es zahlreiche Methoden zur Approximation der gekürzten Singulärwertzerlegung gibt.
Im folgenden geben wir einen Algorithmus an, der mit $O(k^3) + O(k^2(n+m))$ Operationen die Singulärwertzerlegung einer $\mathbf{R}k$-Matrix berechnet. Auch dort ist der Aufwand „$O(k^3)$“ nicht exakt zu bestimmen, da ein $k \times k$-Eigenwertproblem (SVD) gelöst werden muß. In den Anwendungen dieser Arbeit ist stets $k \ll n, m$, so daß „$O(k^2(n+m))$“ der dominante Aufwand ist.

Algorithmus 2.12 *(Singulärwertzerlegung von $\mathbf{R}k$-Matrizen)*
Gegeben sei eine $\mathbf{R}k(n,m)$-Matrix M in der Darstellung $M = AB^T$, $A \in \mathbb{R}^{n,k}, B \in \mathbb{R}^{m,k}$ mit Rang $k' \leq k$. Bestimme QR-Zerlegungen

$$\begin{aligned} A &= Q_A R_A, \quad Q_A \in \mathbb{R}^{n,k}, R_A \in \mathbb{R}^{k,k} \\ B &= Q_B R_B \quad Q_B \in \mathbb{R}^{m,k}, R_B \in \mathbb{R}^{k,k} \end{aligned}$$

von A, B (nur die ersten k Spalten der unitären Matrizen!) und eine Singulärwertzerlegung

$$R_A R_B^T = U \Sigma V^T$$

von $R_A R_B^T$. Dann ist

$$M = AB^T = Q_A U \cdot \Sigma \cdot (Q_B V)^T$$

eine Singulärwertzerlegung von M.
Aufwand: *(vgl. [6, 5.2.9 und 5.4.5])*

QR-Zerlegung von A:	$4nk^2$	
QR-Zerlegung von B:	$4mk^2$	
Multiplikation von $R_A R_B^T$:		$2k^3$
SVD von $R_A R_B^T$:		$21k^3$
Multiplikation von $Q_A U, Q_B V$:	$nk^2 + mk^2$	
Gesamt: $N_{\mathbf{R}k,SVD}(n,m) =$	$5(n+m)k^2$	$+23k^3$

2.4.2. Orthogonale Iteration

Als Alternative zur Berechnung der exakten gekürzten Singulärwertzerlegung kann man auch die Orthogonale Iteration ([6], 7.3.2) zur Bestimmung der Eigenvektoren $v^1, \ldots, v^{\tilde{k}}$ von $M^T M$ verwenden. Die Konvergenz der Iteration hängt dann allerdings entscheidend von der Matrix M ab und kann langsam sein, wenn die Singulärwerte dicht beieinander liegen und eine höhere Genauigkeit erzielt werden soll. Die Analyse entspricht dem Vorgehen in Satz 4.31. Der Aufwand für einen Iterationsschritt liegt bei $k^2(2n + 6m) - k^2 - km - \frac{4}{3}k^3$, vorbereitend ($M^T M$ ausmultiplizieren) sind allerdings $2k^2(n + m) - k^2 - km$ Operationen nötig und anschließend müssen die Vektoren Mv^i mit einem Aufwand von $2k^2(n + m) - k^2 - km$ ausgerechnet werden. Der Aufwand der Orthogonalen Iteration ist für $(n+m) \gg k$ höher als der für die Berechnung der exakten Singulärwertzerlegung.

2.4.3. Gekürzte QR-Zerlegung

Gegeben sei eine $\mathbf{R}k$-Matrix

$$M = \sum_{i=1}^{k} a^i (b^i)^T \quad \in \mathbb{R}^{n,m}$$

und $k' := \text{rang}(M)$. Gesucht ist eine QR-Zerlegung

$$M = QR\Pi = \sum_{i=1}^{k'} q^i (r^i)^T \Pi \qquad \left(\begin{array}{c} q^i = \text{i-te Spalte von } Q, \\ r^i = \text{i-te Zeile von } R \end{array} \right)$$

von M derart, daß Q unitär, R eine obere Dreiecksmatrix und Π eine Permutationsmatrix ist. Eine auf Rang $\tilde{k} \leq k'$ gekürzte QR-Zerlegung $\tilde{M}$ von M ist dann

$$\tilde{M} := \sum_{i=1}^{\tilde{k}} q^i ((r^i)^T \Pi).$$

Einen Algorithmus zur Berechnung der gekürzten QR-Zerlegung findet man in [29] (Gram-Schmidt-Version). Um den Aufwand für $\mathbf{R}k$-Matrizen abschätzen zu können und mit der Notation konform zu bleiben, wird der Algorithmus hier noch einmal angegeben.

Algorithmus 2.13 *(Gekürzte QR-Zerlegung mit Gram-Schmidt)*
In dem Vektor p wird die Permutationsmatrix Π gespeichert, zu Beginn sind $R := 0$ und $Q := 0$. Zuerst werden die Normen

$$v_j := \|M_j\|_2^2, \quad j = 1, \ldots, m,$$

der Spalten von M berechnet und wir setzen $p_i := i$, $i = 1, \ldots, m$. Nacheinander werden nun die Vektoren q^i und r^i für $i = 1, \ldots, k'$ so bestimmt, daß k' Spalten der Matrix exakt wiedergegeben und die restlichen möglichst gut approximiert werden. In v werden die Normen der Spalten des Restes $M - \sum_{j=1}^{i-1} q^j ((r^j)^T \Pi)$ gespeichert. Für jedes $i = 1, \ldots, k'$ ist folgendes zu tun:

1. *(Pivotindex bestimmen) Wähle einen Index $j \in \{i, \ldots, k'\}$ so, daß v_j maximal ist. Vertausche die Inhalte von p_j und p_i sowie von v_{p_i} und v_{p_j}.*
2. *(R pivotieren) Vertausche r^i und r^j.*
3. *(Q um eine Spalte erweitern) Setze $q^i := M_{p_i} - \sum_{\nu=1}^{i-1} q^\nu r_\nu^i$.*
4. *(Diagonalelement von R berechnen) Berechne $r_i^i := \|q^i\|_2$. Abbruch, falls $r_i^i \approx 0$.*
5. *(Q normieren) Normalisiere $q^i := q^i / r_i^i$.*
6. *(Rest von R berechnen) Setze $r_i^\nu := (q^j)^T M_{p_\nu}$ für $\nu = i+1, \ldots, k'$.*
7. *(Spaltennorm aktualisieren) $v_\nu := v_\nu - (r_i^\nu)^2$ für $\nu = i+1, \ldots, k'$.*

Der **Aufwand** *zur Berechnung der gekürzten QR-Zerlegung liegt für eine $\mathbf{R}k(n,m)$-Matrix bei $O(k^2(n+m))$:*

$2k^2(n+m) - k - m + (2k-1)m$	*Spaltennormen berechnen*
$\frac{1}{2}k^2$	*Pivotindex bestimmen*

k^2	*R pivotieren*
$3k^2n$	*Spalte von Q berechnen*
$2kn$	*Diagonale von R*
kn	*Q normieren*
$k^2(n+m) - \frac{1}{2}(k^2+kn)$	*Rest von R berechnen*
k^2	*Spaltennorm aktualisieren*

Insgesamt $6k^2n + 3k^2m + \frac{5}{2}kn + 2km + 2k^2 \quad -2m - k$

Zu beachten ist, daß zur Wahl der Pivotspalten die Normen

$$v_j = \|M_j\|_2^2, \quad j = 1, \ldots, m$$

der Spalten M_j *von* M *berechnet werden müssen, was über die Formel*

$$\|M_j\|_2^2 = (M^T M)_{jj}$$

erfolgt.

Die gekürzte Singulärwertzerlegung ist (besonders für großen Rang k) aufwendiger als die gekürzte QR-Zerlegung, die Approximation kann aber besser sein, was die folgenden Beispiele demonstrieren.

Beispiel 2.14 *(Approximation bei unvollständiger QR-Zerlegung)*
Für die $(n^2+1) \times (n^2+1)$-**R2**-*Matrix*

$$M := \begin{bmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix}$$

berechnen wir die auf Rang 1 *gekürzte QR-Zerlegung*

$$M^{QR} = \begin{bmatrix} n & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

sowie die gekürzte Singulärwertzerlegung

$$M^{SVD} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix}.$$

Die Approximationsfehler liegen dann bei

$$\begin{aligned} \left\|M - M^{QR}\right\|_2 &= n^2, \\ \left\|M - M^{SVD}\right\|_2 &= n. \end{aligned}$$

Folgerung 2.15 *(Zeilenpivotwahl zum Kürzen)*
In Beispiel 2.14 sieht die komplette QR-Zerlegung von M folgendermaßen aus:

$$M = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{n} \\ \vdots & \vdots \\ 0 & \frac{1}{n} \end{bmatrix} \begin{bmatrix} n & 0 & \cdots & 0 \\ 0 & n & \cdots & n \end{bmatrix}$$

Hier wäre es sinnvoll gewesen, erst die komplette (auf Rang 2 gekürzte) QR-Zerlegung der Matrix M zu berechnen und anschließend die Vektoren r^i mit der größten Norm zur Approximation zu wählen. Bei diesem Vorgehen hätte man in dem Beispiel die Bestapproximation wie in der Singulärwertzerlegung erzielen können.

Beispiel 2.16 *(Approximation bei Zeilenpivotwahl)*
Wir berechnen nun die auf Rang 1 gekürzte QR-Zerlegung der $\mathbf{R}2(n,n)$-Matrix

$$M := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

mit anschließender Zeilenpivotwahl aus Folgerung 2.15. Das Ergebnis der gekürzten QR-Zerlegung von M ist

$$M^{QR} = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{3} & \frac{2}{3}\sqrt{3} & 0 & \cdots & 0 \\ 0 & \frac{\sqrt{6}}{3} & 0 & \cdots & 0 \end{bmatrix},$$

so daß die erste Zeile zur Approximation genommen wird. Der Approximationsfehler ist dann $\sqrt{6}/3 \approx 0.816$, während die Bestapproximation aus der Singulärwertzerlegung nur einen Fehler von $\sqrt{2.5 - \sqrt{17}/2} \approx 0.662$ aufweist. Das gleiche Resultat erhält man mit ähnlichen Beispielen auch für höhere Ränge.

Folgerung 2.17 *(Zusammenfassung)*
Die gekürzte QR-Zerlegung eignet sich sehr gut zur Berechnung der QR-Zerlegung einer $\mathbf{R}k(n,m)$-Matrix, da sie mit einer festen Zahl von Rechenschritten und einem Aufwand von $O(k^2(n+m))$ die exakte Zerlegung berechnen kann. Außerdem können mit ihrer Hilfe beliebige Matrizen M mit wenigen Matrix-Vektor-Multiplikationen durch eine $\mathbf{R}k$-Matrix approximiert werden. In diesem Fall kann die Approximation jedoch sehr ungenau sein. Um eine $\mathbf{R}k$-Matrix auf niedrigeren Rang zu kürzen, eignet sich die QR-Zerlegung ohne Zeilenpivotwahl nicht und die QR-Zerlegung mit Zeilenpivotwahl nur wenig, da mit etwas mehr Aufwand die Singulärwertzerlegung eine zuverlässigere Approximation bietet.

2.5. Addition

Die Addition von s $\mathbf{R}k_\nu(n,m)$-Matrizen

$$M^{(\nu)} = \sum_{i=1}^{k_\nu} a^i_{(\nu)} (b^i_{(\nu)})^T \qquad \nu = 1, \ldots, s.$$

ergibt im allgemeinen eine $\mathbf{R}k(n,m)$-Matrix mit Rang $k = \sum_{i=1}^{\nu} k_\nu$. Dies erfordert lediglich ein Kopieren der Vektoren a, b in die gewünschte Zielmatrix. Mit

$$\oplus : \mathbf{R}k'(n,m) \times \mathbf{R}k''(n,m) \to \mathbf{R}\tilde{k}(n,m)$$

bezeichnen wir die formatierte Addition, welche das Ergebnis der Addition einer $\mathbf{R}k'$- und einer $\mathbf{R}k''$-Matrix auf eine $\mathbf{R}\tilde{k}$-Matrix kürzt. Implizit wird hier die Wahl einer Konvertierung (zum Kürzen) vorausgesetzt. Dies ist, wenn nichts anderes erwähnt wird, die gekürzte Singulärwertzerlegung (Algorithmus 2.12), also eine bezüglich des euklidischen Skalarproduktes orthogonale Projektion auf die Menge der $\mathbf{R}\tilde{k}$-Matrizen. Der Aufwand der formatierten Addition innerhalb der $\mathbf{R}k$-Matrizen ist gemäß Algorithmus 2.12 beschränkt durch

$$N_{\mathbf{R}k,\oplus}(n,m) \leq 20k^2(n+m) + 184k^3.$$

2.6. Spektral- und Frobeniusnorm

Von einer $\mathbf{R}k(n,m)$-Matrix $R = AB^T$, $A \in \mathbb{R}^{n,k}$, $B \in \mathbb{R}^{m,k}$, lassen sich die Spektral- und Frobeniusnorm mit der gekürzten Singulärwertzerlegung berechnen, da diese Normen invariant unter orthogonalen Transformationen sind. Die Berechnung besteht also aus der Aufstellung der Matrix $R_A R_B^T$ in Algorithmus 2.12 mit einem Aufwand von $4(n+m)k^2 + 2k^3$ und der Berechnung der Singulärwertzerlegung von $R_A R_B^T$ mit einem Aufwand von $21k^3$, insgesamt also

$$N_{\mathbf{R}k,\|\cdot\|}(n,m) = 4(n+m)k^2 + 23k^3.$$

2.7. Komplexwertige $\mathrm{R}k$-Matrizen

Die Arithmetik komplexwertiger $\mathbf{R}k$-Matrizen (hier mit $\mathbb{C}k$ bezeichnet) läßt sich analog zu der bereits definierten reellwertigen durchführen. Folgende Änderungen sind zu beachten (Aufwand jeweils reelle Gleitkommaoperationen):

- Auswertung:
  $N_{\mathbb{C}k \cdot V} = 8k(n+m) - 2k - 2n, \qquad N_{V \cdot \mathbb{C}k} = 8k(n+m) - 2k - 2m.$
- Multiplikation:
  $N_{\mathbb{C}k \cdot \mathbb{C}k}(n, m', m) = 8k^2(m' + \min\{n,m\}) - 2k^2 - 2k\min\{n,m\}.$

- Singulärwertzerlegung, Algorithmus 2.12:
  Mit Hilfe der komplexwertigen QR-Zerlegung (vgl. [6, 5.2.10]) von A, B erhält man unitäre Matrizen $Q_A \in \mathbb{C}^{n,k}, Q_B \in \mathbb{C}^{m,k}$ und reelle Matrizen $R_A, R_B \in \mathbb{R}^{k,k}$, so daß mit der reellen SVD der Algorithmus fortgeführt werden kann. Der Aufwand ist $N_{\mathbb{C}k,SVD} = O(k^2(n+m) + k^3)$.

- Formatierte Addition:
  $N_{\mathbb{C}k,\oplus} = O(k^2(n+m) + k^3)$.

In den üblichen Softwarepaketen zur numerischen linearen Algebra sind die benötigten Routinen (insbesondere QR-Zerlegung) für komplexwertige Matrizen bereits enthalten, so daß auch bei der Implementierung von $\mathbb{C}k$-Matrizen alles analog zu $\mathbf{R}k$-Matrizen erfolgen kann.

3. Hierarchische Partitionierung

Die Hierarchische Partitionierung der Indexmenge $I \times J$ einer Matrix ist das Bindeglied zwischen dem zugrundeliegenden (diskretisierten) Problem und der abstrakten $\mathcal{H}$-Arithmetik, die wir definieren wollen. Ihre Erzeugung gliedert sich in zwei Teile:

1. Die Indexmenge I (bzw. J) wird hierarchisch unterteilt. Diese Unterteilung kann ganz verschieden motiviert sein, da hier das zugrundeliegende Problem eine Rolle spielt. Die Hierarchie wird verwaltet in Form von Baumstrukturen, die nachfolgend definiert werden.

2. Die Indexmenge $I \times J$ wird hierarchisch unterteilt. Die zu bildenden Teilmengen von $I \times J$ sind Produkte aus Teilmengen von I und J, die in (1.) konstruiert wurden. Eine von dem zugrundeliegenden Problem abhängige *Zulässigkeitsbedingung* gibt an, welche Produkte geeignet sind.

Definition 3.1 *(Baum)*
Ein Mengentupel $T = (V, E)$ mit $E \subset V \times V$ nennen wir Baum mit Knoten V und Kanten E, wenn die folgenden Bedingungen erfüllt sind:

1. *Es gibt genau ein Element $\mathrm{root}(T) \in V$, so daß $(v, \mathrm{root}(T)) \notin E$ für alle $v \in V$ gilt. Dieses Element heißt Wurzel des Baumes.*

2. *Zu jedem Knoten $v \in V \setminus \{\mathrm{root}(T)\}$ gibt es einen Weg $(v_i)_{i=0}^{n}$ der Länge $n \in \mathbb{N}$ von der Wurzel $v_0 = \mathrm{root}(T)$ zu dem Knoten $v_n = v$ mit $(v_i, v_{i+1}) \in E$ für $i = 0, \ldots, n-1$.*

3. *Es gibt keine Zyklen (ein Weg von einem Knoten zu sich selbst).*

Notation 3.2 *(Baumstrukturen)*
Für einen Baum $T = (V, E)$ verwenden wir folgende Bezeichnungen:

- *Die Länge p_T des längsten Weges in T heißt* Tiefe *des Baumes.*
- *Mit „$q \in T$“ ist stets „$q \in V$“ gemeint.*
- *$S(q) := S_T(q) := \{v \in V \mid (q, v) \in E\}$ ist die Menge der Söhne eines Knotens $q \in T$.*
- *$T^{(0)} := \{\mathrm{root}(T)\}$ ist die erste Stufe des Baumes (enthält nur die* Wurzel*).*
- *$T^{(i)}$, $i = 1, \ldots, p_T$, ist die Menge der Söhne von Elementen aus $T^{(i-1)}$, i heißt die* Stufe *von $T^{(i)}$.*
- *$\mathcal{L}(T) := \{q \in T \mid \forall v \in V : (q, v) \notin E\}$ sind die Blätter des Baumes.*
- *$\mathcal{L}(T, i) := \mathcal{L}(T) \cap T^{(i)}$ sind die Blätter des Baumes auf der Stufe $i = 0, \ldots, p_T$.*
- *$\mathcal{L}(T, \leq i) := \mathcal{L}(T) \cap (T^{(i)} \cup \cdots \cup T^{(0)})$ sind die Blätter auf den Stufen $0, \ldots, i \leq p_T$.*

Definition 3.3 *(Hierarchischer Partitionsbaum, $\mathcal{H}$-Baum, L_T)*
Wir nennen einen Baum $T = (V, E)$ mit $V \subset \mathcal{P}(I) \setminus \{\emptyset\}$ ($\mathcal{P}$ = Potenzmenge) und $\mathrm{root}(T) = I$ einen hierarchischen Partitionsbaum, oder kurz $\mathcal{H}$-Baum, der Menge I, falls für alle $t \in T \setminus \mathcal{L}(T)$ gilt:

$$t = \dot{\bigcup_{s \in S(t)}} s.$$

Mit $L_T := \{l \in \mathbb{N}_0 \mid \mathcal{L}(T, l) \neq \emptyset\}$ bezeichnen wir die Menge der Stufen von T, auf denen sich Blätter befinden.

Bemerkung 3.4 *($\mathcal{H}$-Baum bildet stufenweise Partition)*
Sei T ein $\mathcal{H}$-Baum von I und setze $P^{(i)} := T^{(i)} \cup \mathcal{L}(T, \leq i-1)$ für $i = 0, \ldots, p_T$. Dann gilt für jede Stufe $i \in \{0, \ldots, p_T\}$ des Baumes

$$I = \dot{\bigcup_{t \in P^{(i)}}} t,$$

d.h. jede Stufe definiert eine Partition $P^{(i)}$ der Indexmenge. Allgemeiner gilt für jeden Teilbaum T' von T, der die Wurzel I enthält, $I = \dot{\bigcup}_{t \in \mathcal{L}(T')} t$.

Bemerkung 3.5 *(Allgemeiner $\mathcal{H}$-Baum, erweiterte Notation)*
In der Definition 3.3 ist jeder Knoten des Baumes T eine Teilmenge von I. Gelegentlich ist es nützlich, dieselbe Indexmenge $I' \subset I$ auf mehreren Stufen des Baumes zu haben, was hier ausgeschlossen wird. Eine Alternative besteht darin, daß jeder Knoten des Baumes T auf der Stufe i aus einem Paar (I', i) mit $I' \subset I$ besteht. Dadurch wären die Knoten (I', i) und $(I', i+1)$ verschieden, beinhalten aber dieselbe Indexmenge.

Definition 3.6 *(Hierarchischer Produktpartitionsbaum, $\mathcal{H}_\times$-Baum)*
Wir nennen einen Baum $T_{I \times J}$ einen hierarchischen Produktpartitionsbaum von $I \times J$, falls $T_{I \times J}$ ein $\mathcal{H}$-Baum von $I \times J$ ist und zusätzlich jeder Knoten q des Baumes die Gestalt $q = r \times s$, $r \subset I, s \subset J$ hat. Abkürzend schreiben wir auch $\mathcal{H}_\times$-Baum.

Definition 3.7 *(Das Kreuzprodukt $\otimes$ zweier $\mathcal{H}$-Bäume)*
Sei T_I ein $\mathcal{H}$-Baum von I und T_J ein $\mathcal{H}$-Baum von J. Der kanonische $\mathcal{H}_\times$-Baum $T_I \otimes T_J$ ist ein spezieller $\mathcal{H}_\times$-Baum der Tiefe $p = \min\{p_{T_I}, p_{T_J}\}$ von $I \times J$ mit der Eigenschaft, daß für alle $r \times s \in T_I \otimes T_J$ gilt: $r \in T_I, s \in T_J$ und

$$S_{T_I \otimes T_J}(r \times s) = \{r' \times s' \mid r' \in S_{T_I}(r), s' \in S_{T_J}(s)\}.$$

Bemerkung 3.8 *(Struktur des Kreuzproduktes)*
Sei T_I ein $\mathcal{H}$-Baum von I und T_J ein $\mathcal{H}$-Baum von J. Für den kanonischen $\mathcal{H}_\times$-Baum $T_I \otimes T_J$ gilt auf jeder Stufe $i = 0, \ldots, \min\{p_{T_I}, p_{T_J}\}$

$$(T_I \otimes T_J)^{(i)} = T_I^{(i)} \times T_J^{(i)}.$$

Beispiel 3.9 *($\mathcal{H}$-Bäume)*
In Abbildung 4 sind drei $\mathcal{H}$-Bäume dargestellt. T_I und T_J sind $\mathcal{H}$-Bäume verschiedener Tiefe von $\{1, 2, 3\}$. Die Blätter der Stufe 2 von T_J sind nicht relevant für das Kreuzprodukt $T_I \otimes T_J$, da T_I eine geringere Tiefe als T_J hat.

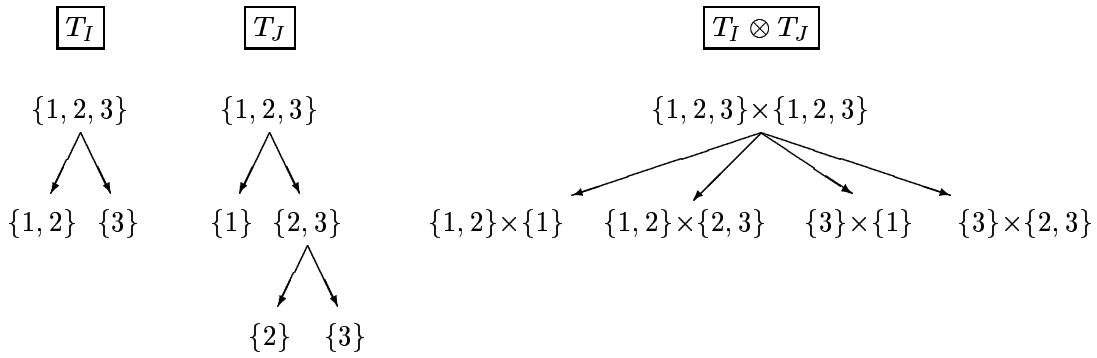


Abbildung 4: $\mathcal{H}$-Bäume T_I, T_J und der $\mathcal{H}_\times$-Baum $T_I \times T_J$.

3.1. Partitionierung und Clusterung

In der Einleitung des Kapitels wurde bereits erwähnt, daß die Unterteilung der Indexmenge I problemabhängig ist. Für die (in den hier betrachteten Anwendungen) wichtigen Problemklassen kann man allerdings eine allgemeine Strategie zur Clusterung der Indizes angeben.
In diesem Abschnitt fixieren wir eine Indexmenge I der Mächtigkeit n und eine Menge von Punkten $m_i \in \mathbb{R}^d$, $d \in \mathbb{N}$, $i \in I$. Diese Punkte könnten Kollokationspunkte aus einer Randelemente-Diskretisierung einer Integralgleichung oder Mittelpunkte der (lokalen) Träger von Basisfunktionen einer Finite-Elemente-Diskretisierung einer partiellen Differentialgleichung sein. Die Indizes $i \in I$ sollen stufenweise zu immer größeren Gruppen zusammengefaßt werden, die dazu korrespondierenden Gruppen von Punkten nennen wir Cluster. Die Cluster einer Stufe sollen einen kleinen Durchmesser im Vergleich zu ihren Abständen haben.

Beispiel 3.10 *(Binäre Raumzerlegung (BSP))*
Der Begriff „binäre Raumzerlegung“ (engl.: Binary Space Partitioning) stammt ursprünglich aus der Computergraphik und wird dort zum Beispiel für das Verfolgen von Strahlen durch einen Raum (Raytracing) eingesetzt. Zuerst wurde es in [3] vorgestellt und später für zahlreiche Anwendungen modifiziert. Die Idee der binären Raumzerlegung besteht darin, bei der Cluster-Erzeugung nicht bei den einelementigen Clustern anzufangen und diese zu akkumulieren, sondern mit den Mengen aller Punkte und Indizes ($\Omega_1^{(0)} := \{m_i | i \in I\}$, $I_1^{(0)} := I$ *) zu starten und diese sukzessive in zwei*

- *gleichmächtige Mengen (Indexmengen $I_1^{(1)}, I_2^{(1)}$) oder*
- *gleichgroße Mengen (Durchmesser von $\Omega_1^{(1)}, \Omega_2^{(1)}$)*

zu zerteilen. Im ersten Fall erhält man einen kardinalitätsbalancierten Baum, im zweiten Fall einen geometrisch balancierten Baum, siehe Abbildung 5. Der Algorithmus zur Zerteilung eines Knotens $I_l^{(p)}$ in seine zwei Söhne $I_{2l-1}^{(p+1)}, I_{2l}^{(p+1)}$ sieht wie folgt aus:

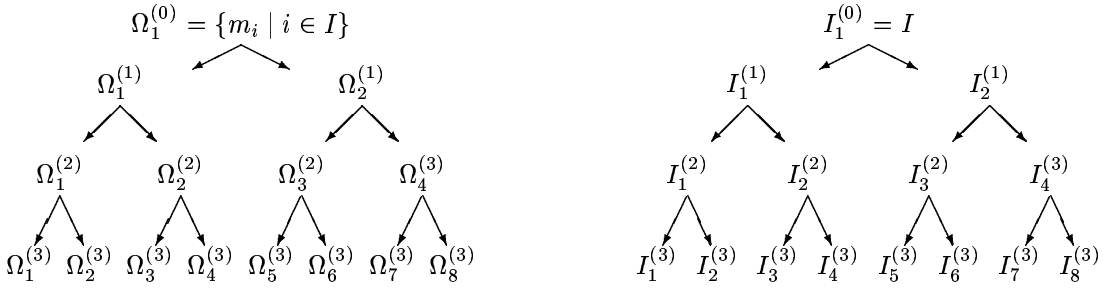


Abbildung 5: Der Baum T_I (rechts) und die entsprechende Struktur für die Punktmenge $\Omega_1^{(0)}$ (links) für $p = 3$

1. *Wähle einen Vektor $e \in E \subset \mathbb{R}^d \setminus \{0\}$ und sortiere die Indizes $j \in I_l^{(p)}$ nach der Größe des euklidischen Skalarproduktes $\langle e, m_j \rangle$ (häufig wählt man e als einen der Einheitsvektoren, so daß die Punkte bezüglich einer Koordinatenrichtung sortiert werden).*

2. *Bestimme einen Index j^* so, daß*

   - *(kardinalitätsbalanciert)*
     $\left|\{j \in I_l^{(p)} \mid \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle\}\right| \approx \left|\{j \in I_l^{(p)} \mid \langle e, m_j \rangle > \langle e, m_{j^*} \rangle\}\right|$
   - *(geometrisch balanciert)*
     $\operatorname{diam}(\{m_j \mid j \in I_l^{(p)}, \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle\})$
     $\approx \operatorname{diam}(\{m_j \mid j \in I_l^{(p)}, \langle e, m_j \rangle > \langle e, m_{j^*} \rangle\})$

   ist und setze

   $$I_{2l-1}^{(p+1)} := \{j \in I_l^{(p)} \mid \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle\}, \qquad \Omega_{2l-1}^{(p+1)} := \{m_j \mid j \in I_{2l-1}^{(p+1)}\}$$
   $$I_{2l}^{(p+1)} := \{j \in I_l^{(p)} \mid \langle e, m_j \rangle > \langle e, m_{j^*} \rangle\}, \qquad \Omega_{2l}^{(p+1)} := \{m_j \mid j \in I_{2l}^{(p+1)}\}.$$

   (Falls mehrere Punkte m_j die gleiche Koordinate $\langle e, m_j \rangle$ wie m_{j^*} haben, so teilt man die Indizes so auf, daß die Söhne möglichst gleichmächtig sind)

Den Vektor e wählt man so, daß die Durchmesser der Mengen $\Omega_{2l-1}^{(p+1)}$ und $\Omega_{2l}^{(p+1)}$ möglichst klein werden. Die einfachste Wahl ist hier, die Durchmesser ρ_j von $\Omega_l^{(p)}$ projiziert auf die j-te Koordinatenachse zu bestimmen und $e := e_j$ korrespondierend zum größten ρ_j zu setzen. Einelementige Knoten werden nicht weiter unterteilt und sind Blätter. In manchen Fällen kann es sinnvoll sein, Knoten nicht weiter zu unterteilen, etwa falls man eine Mindestmächtigkeit der Knoten vorschreiben möchte. Ein Beispiel für die so entstehenden Cluster ist in Abbildung 6 gegeben.

Lemma 3.11 *(Eigenschaften eines BSP-$\mathcal{H}$-Baumes)*
Der $\mathcal{H}$-Baum T_I der Indexmenge I sei mit dem BSP-Algorithmus kardinalitätsbalanciert erzeugt worden. Dann gilt:

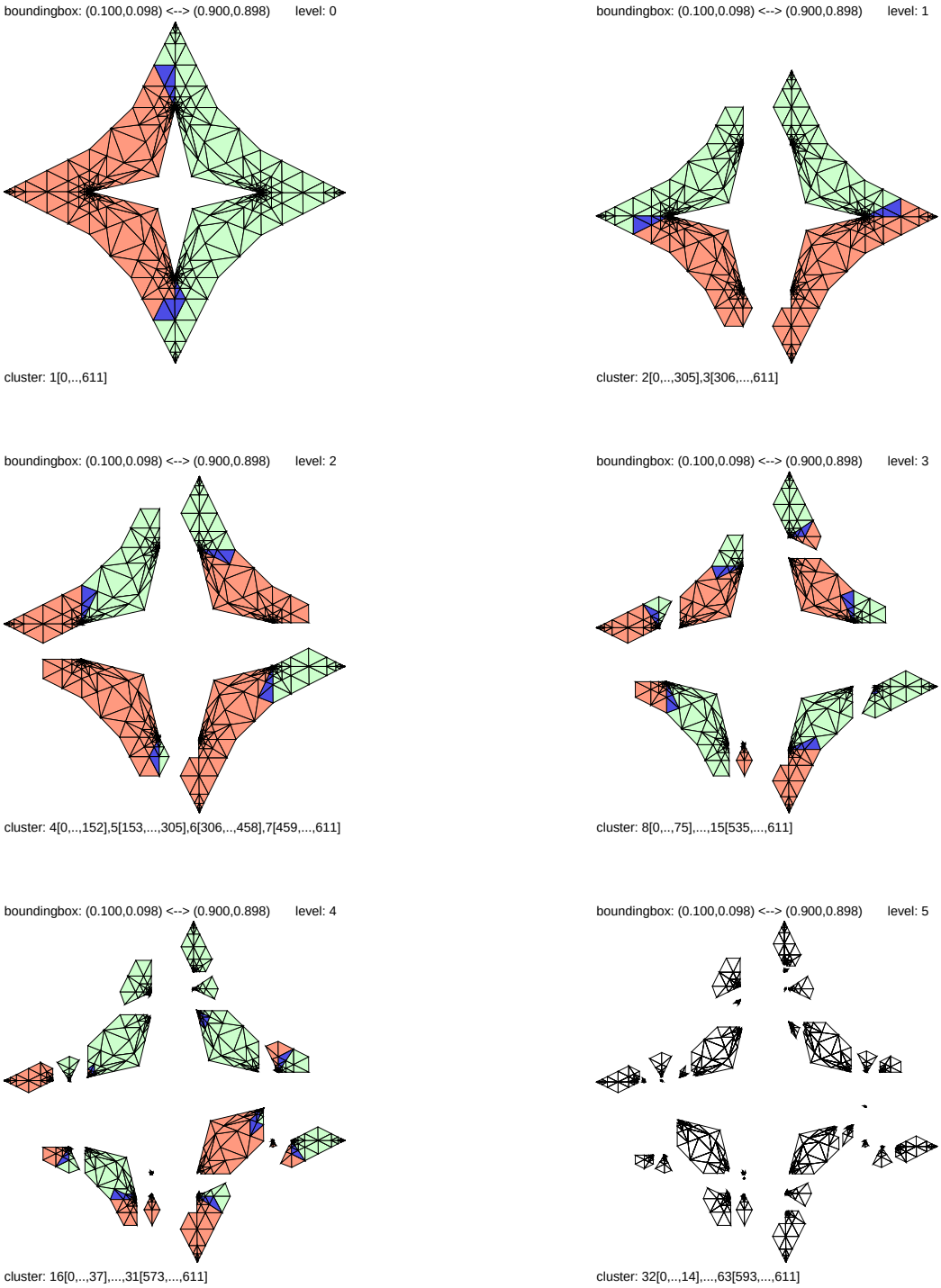


Abbildung 6: Das Sterngebiet wird trianguliert und die Menge der inneren Gitterpunkte kardinalitätsbalanciert aufgeteilt. Der Bereich, in dem sich die Träger der (linearen Knoten-) Basisfunktionen überlappen, ist dunkel (blau) eingefärbt. Die zwei Cluster auf der Stufe 1 (rechts oben) werden bzgl. der y-Koordinate unterteilt. Nach 5 Unterteilungen sind die Cluster klein genug und werden nicht weiter geteilt.

1. *Der Baum T_I hat eine Tiefe von*

$$p_T = \begin{cases} \log_2(n) & \text{falls } n \text{ eine Zweierpotenz ist,} \\ \lfloor \log_2(n) \rfloor + 1 & \text{sonst.} \end{cases}$$

2. *Auf jeder Stufe $p = 0, \ldots, p_T - 1$ befinden sich 2^p Knoten, auf der Stufe p_T sind höchstens n Knoten.*

3. *Jeder Knoten auf der Stufe p hat eine Mächtigkeit von $O(2^{log_2(n)-p})$.*

4. *Die Blätter von T_I sind genau die einelementigen Teilmengen von I.*

5. *Der Aufwand des BSP-Algorithmus ist $O(n \log_2(n))$.*

Beweis:

1. *Von einer Stufe p zur Stufe $p+1$ werden die Knoten (Indexmengen) halbiert, so daß nach $\log_2(n)$ Stufen ($\lfloor \log_2(n) \rfloor + 1$ falls n nicht Zweierpotenz) die Indexmengen einelementig sind.*

2. *Die Anzahl der Knoten von einer Stufe p zur nächsten kann sich höchstens verdoppeln, so daß auf der Stufe p maximal 2^p Knoten vorhanden sind.*

3. *Auf der Stufe p wurde die Indexmenge bereits $p-1$-mal (fast) halbiert, also umfaßt sie höchstens noch $|I_l^p| \leq 2n/2^p = 2^{\log_2(n)-p+1}$ Elemente.*

4. *Nicht einelementige Teilmengen werden weiter unterteilt.*

5. *Das Sortieren hat für einen Knoten auf der Stufe p nach 3. einen Aufwand von $O(2^{\log_2(n)-p})$ (lineare Medianbestimmung, siehe [22, II.4.]), summiert über alle 2^p Knoten der Stufe und alle $\log_2(n)$ (bzw. $\lfloor \log_2(n) \rfloor + 1$) Stufen erhalten wir einen Gesamtaufwand von*

$$N_{BSP}(n) = O(n \log_2(n)).$$

Ist $|E| = O(1)$, so kann man für die Gesamtindexmenge I die Sortierung für alle $e \in E$ durchführen (Aufwand $O(n \log_2(n))$ mit Heap-Sort). Auf den weiteren Stufen kann man bei der Bisektion die geordneten Listen entsprechend mit einem Aufwand $O(n)$ (dieselbe Ordnung aber geringer als bei der Medianbestimmung) in je zwei Teile aufteilen, welche dann schon sortiert sind.

■

Beispiel 3.12 *(Modellfall $d = 1$)*
Aus der Indexmenge $I = \{1, \ldots, n\}$, $n = 2^{p_T}$, $p_T \in \mathbb{N}$, wird mit dem BSP-Algorithmus ein $\mathcal{H}$-Baum T_I von I generiert. Wir nehmen an, daß die Punkte m_j in derselben Weise geordnet sind wie die Indexmenge I. Dann sind die Knoten des Baumes T_I

$$I_1^{(0)} = \{1, \ldots, n\},$$

$$I_1^{(1)} = \{1, \ldots, \frac{n}{2}\}, \quad I_2^{(1)} = \{\frac{n}{2} + 1, \ldots, n\},$$
$$I_l^{(p)} = \{\frac{n}{2^p}(l-1) + 1, \ldots, \frac{n}{2^p} l\}, \quad l = 1, \ldots, 2^p, \quad p = 0, \ldots, p_T.$$

Beispiel 3.13 *(Modellfall $d = 2$)*
Gegeben seien $p_T \in 2\mathbb{N}$, $n = 2^{p_T}$ und Punkte

$$m_{(i,j)} := \left(\frac{i}{2^{p_T/2} + 1}, \frac{j}{2^{p_T/2} + 1} \right) \in \mathbb{R}^2,$$

$i, j \in \{1, \ldots, 2^{p_T/2}\}$. Die Punkte $m_{(i,j)}$ entsprechen den inneren Gitterpunkten einer regelmäßigen Triangulation von $[0,1]^2$. Der Baum T_I wird mit dem BSP-Algorithmus erzeugt, wobei der Vektor e alternierend $e_1 = (1,0)$ und $e_2 = (0,1)$ ist. Die Wurzel des Baumes ist $I_1^{(0)} = \{1, \ldots, n\}$. Auf der ersten Stufe wird die Punktmenge in die mit den kleineren und die mit den größeren x-Koordinaten aufgeteilt:

$$\begin{aligned} I_1^{(1)} &= \{(i,j) \mid j \in \{1, \ldots, 2^{\frac{p_T}{2}}\}, i \in \{1, \ldots, 2^{\frac{p_T}{2}-1}\}\}, \\ I_2^{(1)} &= \{(i,j) \mid j \in \{1, \ldots, 2^{\frac{p_T}{2}}\}, i \in \{1 + 2^{\frac{p_T}{2}-1}, \ldots, 2^{p_T/2}\}\}. \end{aligned}$$

Auf der folgenden Stufe erfolgt die Aufteilung bzgl. der y-Koordinaten und auf der p-ten Stufe ($p = 2, \ldots, p_T$) läßt sich der l-te Knoten mit Hilfe der Binärdarstellung $(l-1) = \sum_{\nu=0}^{p-1} 2^\nu \beta_\nu$, $\beta_\nu \in \{0,1\}$, angeben:

$$I_l^{(p)} = \left\{ (i,j) \;\middle|\; i \in \sum_{\nu=0}^{\lfloor \frac{p-1}{2} \rfloor} 2^{\frac{p_T}{2}-\nu} \beta_{2\nu} + \left[1, 2^{\lfloor \frac{p_T - p}{2} \rfloor}\right], \right.$$
$$\left. j \in \sum_{\nu=0}^{\lfloor \frac{p-2}{2} \rfloor} 2^{\frac{p_T}{2}-1-\nu} \beta_{2\nu+1} + \left[1, 2^{\lfloor \frac{p_T - p + 1}{2} \rfloor}\right] \right\}.$$

Beispiel 3.14 *(Modellfall $d = 3$)*
Gegeben seien $p_T \in 3\mathbb{N}$, $n = 2^{p_T}$ und innere Punkte

$$m_{(i,j,r)} := \left(\frac{i}{2^{p_T/3} + 1}, \frac{j}{2^{p_T/3} + 1}, \frac{r}{2^{p_T/3} + 1} \right) \in \mathbb{R}^3,$$

$i, j, r \in \{1, \ldots, 2^{p_T/3}\}$, einer regelmäßigen Triangulation von $[0,1]^3$. Der mit dem BSP-Algorithmus erzeugte Baum (e ist alternierend $e_1 = (1,0,0)$, $e_2 = (0,1,0)$ und $e_3 = (0,0,1)$) besitzt auf der Stufe $p = 3, \ldots, p_T$ die Knoten

$$I_l^{(p)} = \left\{ (i,j,r) \;\middle|\; i \in \sum_{\nu=0}^{\lfloor \frac{p-1}{3} \rfloor} 2^{\frac{p_T}{3}+1-\nu} \beta_{3\nu} + \left[1, 2^{\lfloor \frac{p_T - p}{3} \rfloor}\right], \right.$$
$$j \in \sum_{\nu=0}^{\lfloor \frac{p-2}{3} \rfloor} 2^{\frac{p_T}{3}-\nu} \beta_{3\nu+1} + \left[1, 2^{\lfloor \frac{p_T - p + 1}{3} \rfloor}\right],$$
$$\left. r \in \sum_{\nu=0}^{\lfloor \frac{p-3}{3} \rfloor} 2^{\frac{p_T}{3}-1-\nu} \beta_{3\nu+2} + \left[1, 2^{\lfloor \frac{p_T - p + 2}{3} \rfloor}\right] \right\}$$

mit $l = 1, \ldots, 2^p$ *und der Binärdarstellung* $(l-1) = \sum_{\nu=0}^{p-1} 2^\nu \beta_\nu$, $\beta_\nu \in \{0, 1\}$.

Bemerkung 3.15 *(Andere Konstruktionen des $\mathcal{H}$-Baumes)*

1. *In [21] (4.2.1 und Anhang C) wird ein Algorithmus vorgestellt, welcher dem BSP-Algorithmus in der geometrisch balancierten Variante entspricht. Der Vektor e wird dabei für jeden aufzuteilenden Knoten neu bestimmt als* $e := m_i - m_j$ *für zwei Punkte* m_i, m_j *des Clusters mit maximalem Abstand zueinander. Die Sortierung fällt in diesem Fall weg (Punkte werden danach gruppiert, welchem der beiden Punkte* m_i, m_j *sie näher sind), allerdings läßt sich für die Bestimmung der Punkte* m_i *und* m_j *kein Algorithmus mit befriedigender Komplexität angeben.*

2. *In [5] (Algorithmus 3) wird der BSP-Algorithmus in der kardinalitätsbalancierten Version durchgeführt und „geometrische Bisektion" genannt. Der Vektor e wird entsprechend der längsten Kante einer achsenparallelen „Box" gewählt.*

3. *In [26] wird der $\mathcal{H}$-Baum aus einer vorhandenen Gitterhierarchie extrahiert, indem für jedes Element τ einer Diskretisierungsstufe die Indizes des korrespondierenden Knotens des $\mathcal{H}$-Baumes aus den Söhnen von τ in der Gitterhierarchie ermittelt werden.*

3.2. Die Zulässigkeitsbedingung

Das Kreuzprodukt $\otimes$ zweier $\mathcal{H}$-Bäume T_I, T_J liefert auf kanonische Weise einen $\mathcal{H}$-Baum von $I \times J$. $T_I \otimes T_J$ enthält zwar nicht alle Produkte $r \times s$ von Knoten $r \in T_I$ und $s \in T_J$, die Anzahl der Elemente in $T_I \otimes T_J$ ist allerdings nur durch $O(|T_I| \cdot |T_J|)$ beschränkt, so daß das Kreuzprodukt nur mit quadratischem Aufwand berechen- oder speicherbar ist. Die Blätter eines $\mathcal{H}$-Baumes $T_{I \times J}$ kennzeichnen Blöcke der Matrix, die durch $\mathbf{R}k$-Matrizen repräsentiert werden können. Sind alle Blätter des Baumes einelementig, so erhält man die übliche vollbesetzte Darstellung der Matrix. Die Entscheidung, wann ein Knoten des Baumes nicht weiter unterteilt werden muß, weil bereits eine $\mathbf{R}k$-Darstellung des korrespondierenden Blockes der Matrix möglich ist, wird über die sogenannte *Zulässigkeitsbedingung*

$$Z : T_{I \times J} \to \{\text{„zulässig"}, \text{„nicht zulässig"}\}$$

geregelt. In diese Bedingung geht wieder das diskretisierte Problem ein. Für einen Teil der Probleme, die wir später untersuchen werden, ist die folgende Zulässigkeitsbedingung hinreichend.

3.2.1. Standard-Zulässigkeitsbedingung

Gegeben seien Indexmengen I und J, $\mathcal{H}$-Bäume T_I von I und T_J von J sowie Mengen $\{\tau_i \mid i \in I\}, \{\sigma_j \mid j \in J\}$, deren Elemente Teilmengen des $\mathbb{R}^d$ sind und die zugrundeliegende Geometrie charakterisieren (zum Beispiel Träger von Basisfunktionen oder Kollokationspunkte).

Definition 3.16 *(η-zulässig, Z_η)*
Ein Knoten $r \times s \in T_I \otimes T_J$ heißt η-zulässig (zum Parameter $\eta \in \mathbb{R}$), falls für $\tau := \bigcup_{i \in r} \tau_i$ und $\sigma := \bigcup_{j \in s} \sigma_j$

$$\min\{\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\} \leq 2\eta \operatorname{dist}(\tau, \sigma) \tag{9}$$

ist. Entsprechend nennen wir ein Mengenprodukt $\tau \times \sigma$ zulässig, wenn (9) gilt. Die Standard-Zulässigkeitsbedingung Z_η (zum Parameter η) ist definiert als

$$Z_\eta(r \times s) := \begin{cases} \text{„zulässig“} & \text{falls } r \times s \ \eta\text{-zulässig ist,} \\ \text{„nicht zulässig“} & \text{sonst.} \end{cases} \tag{10}$$

Hier bezeichnet $\operatorname{diam}(M)$ das Infimum der Durchmesser aller Kugeln, die die Menge M enthalten, und $\operatorname{dist}(\tau, \sigma)$ das Infimum der Abstände aller Punkte aus τ zu Punkten aus σ. Die Wahl des Parameters η hängt von dem zu diskretisierenden Operator ab. Die Berechnung von Durchmesser und Abstand ist in den höherdimensionalen Fällen ($d > 1$) nicht unproblematisch; es empfiehlt sich, diese durch leicht zu berechnende Approximationen $\widetilde{\operatorname{diam}}$ und $\widetilde{\operatorname{dist}}$ zu ersetzen. Die Güte einer Approximation der Standard-Zulässigkeitsbedingung wird in der folgenden Definition charakterisiert.

Definition 3.17 *(Zuverlässigkeit und Effizienz)*
Eine Familie $(\tilde{Z}_\eta)_{\eta \in \mathbb{R}_{>0}}$ von Zulässigkeitsbedingungen heißt zuverlässig, *falls für alle $\eta \in \mathbb{R}_{>0}$*

$$\tilde{Z}_\eta^{-1}(\text{„zulässig“}) \subset Z_\eta^{-1}(\text{„zulässig“})$$

ist, also unter der Standard-Zulässigkeitsbedingung Z_η nicht zulässige Knoten auch unter $\tilde{Z}_\eta$ nicht zulässig sind. Sie heißt effizient, *falls es ein $\eta_0 \in \mathbb{R}_{>0}$ und ein $C \in \mathbb{R}_{>0}$ gibt, so daß für alle $\eta \in (0, \eta_0)$ ein $\tilde{\eta} \in [C\eta, \eta]$ mit*

$$\tilde{Z}_\eta^{-1}(\text{„zulässig“}) \supset Z_{\tilde{\eta}}^{-1}(\text{„zulässig“})$$

existiert. Diese Eigenschaft stellt sicher, daß ein unter $\tilde{Z}_\eta$ nicht zulässiger Knoten auch unter $Z_{\tilde{\eta}}$ für ein höchstens um den Faktor C kleineres $\tilde{\eta}$ nicht zulässig ist.

Definition 3.18 *(Bounding-Box)*
Die Bounding-Box $Q(\tau)$ einer beschränkten Menge $\tau \subset \mathbb{R}^d$ ist definiert als

$$Q(\tau) := \left\{ x \in \mathbb{R}^d \;\middle|\; \inf_{y \in \tau} y_i \leq x_i \leq \sup_{y \in \tau} y_i \right\}.$$

Definition 3.19 *(Einfache Zulässigkeitsbedingung Z_η^{simple})*
Die approximativen Distanzen und Durchmesser definieren wir für beschränkte Mengen $\tau, \sigma \subset \mathbb{R}^d$ durch

$$\begin{aligned} \operatorname{diam}^{\text{simple}}(\tau) &:= \operatorname{diam}(Q(\tau)), \\ \operatorname{dist}^{\text{simple}}(\tau, \sigma) &:= \operatorname{dist}(Q(\tau), Q(\sigma)) \end{aligned}$$

und die approximative Zulässigkeitsbedingung $Z_\eta^{\text{simple}} : T_I \otimes T_J \to \{$„zulässig“, „nicht zulässig“$\}$ für $\eta \in \mathbb{R}_{>0}$ durch (9),(10) mit $\operatorname{diam}^{\text{simple}}, \operatorname{dist}^{\text{simple}}$ anstelle von diam, dist.

Bemerkung 3.20 *(Ineffizienz von Z_η^{simple})*
Die Familie $(Z_\eta^{\text{simple}})_{\eta\in\mathbb{R}_{>0}}$ von einfach zu berechnenden Zulässigkeitsbedingungen ist zuverlässig, aber, wie das Beispiel in Abbildung 7 zeigt, nicht effizient.

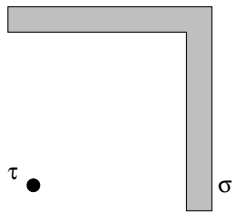


Abbildung 7: Das Cluster-Paar $\tau\times\sigma$ ist für alle $\eta\in\mathbb{R}_{>0}$ bei hinreichend kleinem Radius von τ unter Z_η zulässig, aber für kein $\eta\in\mathbb{R}_{>0}$ unter der zu scharfen Zulässigkeitsbedingung Z_η^{simple} zulässig.

Definition 3.21 *(Effiziente Zulässigkeitsbedingung Z_η^{eff})*
Die approximativen Distanzen und Durchmesser definieren wir für beschränkte Mengen $\tau,\sigma\subset\mathbb{R}^d$ durch

$$\begin{aligned}\operatorname{diam}^{\text{eff}}(\tau) &:= \operatorname{diam}(Q(\tau)),\\ \operatorname{dist}^{\text{eff}}(\tau,\sigma) &:= \begin{cases}\operatorname{dist}(Q(\tau),\sigma) & \textit{falls}\quad \operatorname{diam}^{\text{eff}}(\tau)\le\operatorname{diam}^{\text{eff}}(\sigma)\\ \operatorname{dist}(\tau,Q(\sigma)) & \textit{falls}\quad \operatorname{diam}^{\text{eff}}(\tau)>\operatorname{diam}^{\text{eff}}(\sigma)\end{cases}\end{aligned}$$

und die approximative Zulässigkeitsbedingung $Z_\eta^{\text{eff}}: T_I\otimes T_J\to\{$„zulässig“, „nicht zulässig“$\}$ für $\eta\in\mathbb{R}_{>0}$ durch (9),(10) mit $\operatorname{diam}^{\text{eff}},\operatorname{dist}^{\text{eff}}$ *anstelle von* $\operatorname{diam},\operatorname{dist}$.

Lemma 3.22 *(Effizienz von Z_η^{eff})*
Sei $\eta_0:=1$ und $C:=\frac{1}{3\sqrt{d}}$. Dann gilt für alle $\eta\in(0,\eta_0)$ und $\tilde\eta:=\frac{1}{(1+2\eta)\sqrt{d}}\eta\in[C\eta,\eta]$:

$$(Z_\eta^{\text{eff}})^{-1}(\text{„zulässig“})\supset Z_{\tilde\eta}^{-1}(\text{„zulässig“}),$$

d.h. Z_η^{eff} ist eine effiziente und zuverlässige Approximation für die Standard-Zulässigkeitsbedingung.

Beweis: Sei $\tau\times\sigma$ zulässig unter $Z_{\tilde\eta}$, $\eta\in(0,\eta_0)$. Dann ist

$$\min\{\operatorname{diam}(\tau),\operatorname{diam}(\sigma)\}\le 2\tilde\eta\operatorname{dist}(\tau,\sigma)$$

und o.B.d.A. $\operatorname{diam}^{\text{eff}}(\tau)\le\operatorname{diam}^{\text{eff}}(\sigma)$. Aus

$$\begin{aligned}\operatorname{diam}^{\text{eff}}(\tau) &= \sqrt{\sum_{i=1}^{d}\left(\max_{x\in\tau}x_i-\min_{x\in\tau}x_i\right)^2} \le\sqrt{d}\operatorname{diam}(\tau) \qquad\text{und}\\ \operatorname{dist}^{\text{eff}}(\tau,\sigma) &= \operatorname{dist}(Q(\tau),\sigma) \ge\operatorname{dist}(\tau,\sigma)-\operatorname{diam}^{\text{eff}}(\tau) \qquad\text{folgt}\end{aligned}$$

$$
\begin{aligned}
\frac{1}{\sqrt{d}}\mathrm{diam}^{\mathrm{eff}}(\tau) &= \min\{\frac{1}{\sqrt{d}}\mathrm{diam}^{\mathrm{eff}}(\tau), \frac{1}{\sqrt{d}}\mathrm{diam}^{\mathrm{eff}}(\sigma)\} \\
&\leq \min\{\mathrm{diam}(\tau), \mathrm{diam}(\sigma)\} \\
&\leq 2\tilde{\eta}\mathrm{dist}\,(\tau,\sigma) \\
&\leq 2\tilde{\eta}(\mathrm{dist}^{\mathrm{eff}}(\tau,\sigma) + \mathrm{diam}^{\mathrm{eff}}(\tau)), \\
\left(1 - \frac{2\eta}{1+2\eta}\right)\mathrm{diam}^{\mathrm{eff}}(\tau) &\leq 2\frac{\eta}{1+2\eta}\mathrm{dist}^{\mathrm{eff}}(\tau,\sigma), \\
\min\{\mathrm{diam}^{\mathrm{eff}}(\tau), \mathrm{diam}^{\mathrm{eff}}(\sigma)\} &\leq 2\eta\mathrm{dist}^{\mathrm{eff}}(\tau,\sigma).
\end{aligned}
$$

■

Bemerkung 3.23 *(Zur Wahl der Approximation für die Zulässigkeitsbedingung)*
Die Erzeugung des $\mathcal{H}$-Baumes T_I aus der Indexmenge I ($|I| = n$) ließ sich mit dem BSP-Algorithmus mit einem Aufwand von $O(n\log(n))$ realisieren. Es ist also erstrebenswert, auch den $\mathcal{H}_\times$-Baum $T_{I\times J}$ mit logarithmisch-linearem Aufwand aufzustellen. Dazu werden die Knoten des Baumes $T_{I\times J}$ stufenweise mit der Wurzel beginnend auf Zulässigkeit getestet, zum Beispiel mit der Standard-Zulässigkeitsbedingung. Die Ermittlung des dafür benötigten exakten Abstandes und Durchmessers zweier beliebiger Teilmengen $\sigma, \tau \subset \mathbb{R}^d$ ist nicht durchführbar, aber die Mengen σ, τ haben die spezielle Struktur

$$\tau = \bigcup_{i\in r} \tau_i, \qquad \sigma = \bigcup_{j\in s} \sigma_j.$$

Die Elemente τ_i, σ_j beschreiben Träger von Basisfunktionen (üblicherweise aus wenigen Simplices zusammengesetzt) oder enthalten nur wenige Punkte, so daß für diese Elemente Durchmesser und Abstand untereinander leicht berechnet werden können. Die Bestimmung des Abstandes von σ zu τ läßt sich dann mit quadratischem Aufwand $O((|r|+|s|)^2)$ durchführen. Zur Verringerung der Kosten kann man die approximative Zulässigkeitsbedingung Z_η^{eff} verwenden, ohne das Risiko einzugehen, nicht zulässige Knoten als zulässig anzuerkennen, und ohne die Effizienz zu verlieren. Die Auswertung von Z_η^{eff} läßt sich mit $O(|r| + |s|)$ Berechnungen des Abstandes der Elemente aus τ, σ realisieren, so daß der Aufwand zur Bestimmung der Zulässigkeit eines Knotens $(r, s) \in T_I \otimes T_J$ proportional zur Mächtigkeit der Indexmengen r, s ist.

Bemerkung 3.24 *(Paneel-Clusterung und Z_η^{simple})*
Für das Paneel-Clusterungs-Verfahren wird jeweils die Zulässigkeit einer Menge τ_i (Kollokationspunkt oder Träger einer Basisfunktion) zu einem Cluster $\sigma = \cup_{j\in s}\sigma_j$ geprüft. In diesem Fall läßt sich der Abstand mit einem Aufwand von $O(|s|)$ exakt bestimmen. Der Radius hingegen ist auch dort nicht trivial zu bestimmen (vgl. [21] Anhang C)).
In der Praxis wird im Zusammenhang mit dem BSP-Algorithmus meistens die einfache Zulässigkeitsbedingung Z_η^{simple} verwendet, da sie besonders schnell auswertbar ist und die Ineffizienz dort konstruktionsbedingt keine große Rolle spielt (bei gleichmäßiger Verteilung der die Geometrie charakterisierenden Punkte sind die Cluster annähernd Quader).

3.3. Partitionierung der Produkt-Indexmenge

Wir fixieren zwei Indexmengen I, J und $\mathcal{H}$-Bäume T_I, T_J. Gesucht ist ein $\mathcal{H}_\times$-Baum von $I \times J$ mit möglichst wenig Blättern, die alle zulässig sind oder eine vorgegebene Mächtigkeit unterschreiten. Einen solchen $\mathcal{H}_\times$-Baum nennen wir *minimal zulässig*. Die Erzeugung eines minimal zulässigen $\mathcal{H}_\times$-Baumes von $I \times J$ ist nicht trivial, sie vereinfacht sich jedoch erheblich, wenn man auf jeder Stufe des Baumes nur Produkte aus Knoten derselben Stufe aus T_I und T_J zuläßt, was zur nachfolgenden Definition führt.

Definition 3.25 *(Aus T_I, T_J gebildeter $\mathcal{H}$-Baum, zulässiger $\mathcal{H}$-Baum)*
Sei $T = (V', E')$ ein $\mathcal{H}_\times$-Baum von $I \times J$. Wir sagen T wurde aus T_I und T_J gebildet, *falls für $T_I \otimes T_J = (V, E)$ gilt:*

$$V' \subset V, \qquad E' \subset E.$$

Im Fall $T_I = T_J$ sagen wir auch „aus T_I gebildet“. T heißt zulässig *(bezüglich einer Zulässigkeitsbedingung Z und einer Blockgröße $b_{min} \in \mathbb{N}$), falls für alle $\sigma \times \tau \in \mathcal{L}(T)$ gilt:*

$$Z(\sigma \times \tau) = \text{„zulässig“} \quad \text{oder} \quad |\sigma \times \tau| \leq b_{min}. \tag{11}$$

Beispiel 3.26 *(Berechnung des minimal zulässigen aus T_I, T_J gebildeten $\mathcal{H}_\times$-Baumes)*
Gegeben sei eine Zulässigkeitsbedingung Z und eine Blockgröße $b_{min} \in \mathbb{N}$. T heißt minimal zulässig aus T_I, T_J gebildet *(bzgl. Z und b_{min}), falls T unter allen zulässigen aus T_I, T_J gebildeten $\mathcal{H}_\times$-Bäumen eine minimale Zahl von Blättern besitzt. Durch diese Bedingung ist T bereits eindeutig bestimmt und läßt sich konstruieren:*

Auf der Stufe 0 befindet sich gemäß der Definition die Wurzel $\text{root}(T) := I \times J$. *Ist*

$$Z(I \times J) = \text{„zulässig“} \quad \text{oder} \quad |I \times J| \leq b_{min},$$

so sind wir fertig. Andernfalls besitzt $I \times J$ genau die Söhne $S_T(I \times J) = \{\sigma \times \tau \mid \sigma \in S_{T_I}(I), \tau \in S_{T_J}(J)\}$. Für jede Stufe $i = 0, \ldots, \min\{p_{T_I}, p_{T_J}\}$ wird so jeder Knoten $\sigma \times \tau$ der Stufe i darauf geprüft, ob er die Bedingung (11) erfüllt. Ist dies nicht der Fall, so hat er die durch $S_{T_I}(\sigma)$ und $S_{T_J}(\tau)$ eindeutig bestimmten Söhne.
Ist eine der Mengen $S_{T_I}(\sigma)$ oder $S_{T_J}(\tau)$ leer, aber der Knoten erfüllt nicht die Bedingung (11), so gibt es keinen zulässigen aus T_I, T_J gebildeten $\mathcal{H}_\times$-Baum. Dies kann durch die Voraussetzung $p_{T_I} = p_{T_J} =: p$ (alle Bäume besitzen dieselbe Tiefe) und

$$\forall \tau \in \mathcal{L}(T_I) \cup \mathcal{L}(T_J): \qquad |\tau| \leq \sqrt{b_{min}} \quad \wedge \quad \tau \in T_I^{(p)} \cup T_J^{(p)}$$

ausgeschlossen werden.
Sei T der minimal zulässige aus T_I, T_I gebildete $\mathcal{H}_\times$-Baum für den Baum T_I zur Abbildung 6. Die Blätter von T bilden dann eine Partition von $I \times I$ und sind in Abbildung 8 dargestellt.

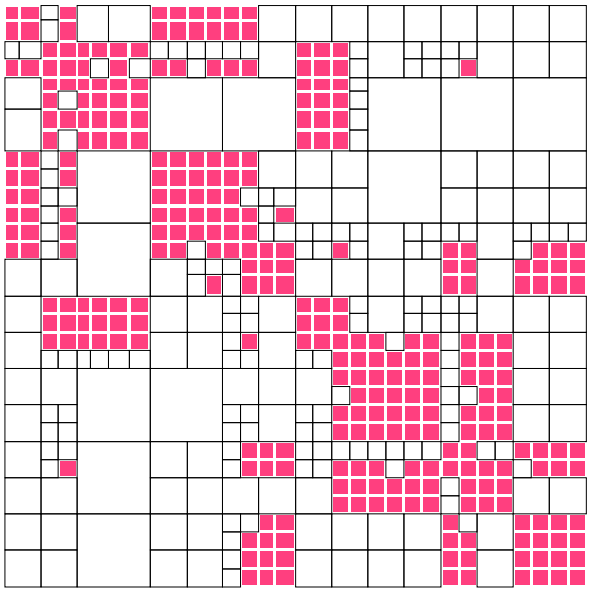


Abbildung 8: Der aus dem $\mathcal{H}$-Baum in Abbildung 6 gebildete $\mathcal{H}_\times$-Baum (Standard-Zulässigkeitsbedingung $Z_{0.8}$) definiert durch seine Blätter eine Partition von $I \times I$. Die zulässigen Blätter sind weiß, die nicht zulässigen rot (dunkel) gefärbt.

3.4. Arithmetik von $\mathcal{H}$-Bäumen

Ein zulässiges Blatt eines $\mathcal{H}_\times$-Baumes beschreibt einen Indexbereich, der für den korrespondierenden Block in einer Matrix eine besondere Darstellung ($\mathbf{R}k$-Matrix) erlaubt. Bei der Addition oder Multiplikation zweier Matrizen zu unterschiedlichen $\mathcal{H}_\times$-Bäumen kann es nützlich sein zu wissen, in welcher Struktur das (exakte) Ergebnis liegt. Diese Information kann man direkt aus den $\mathcal{H}_\times$-Bäumen gewinnen, ohne die ursprüngliche Zulässigkeitsbedingung oder Eigenschaften der Ergebnismatrix zu kennen.
Gegeben seien zwei $\mathcal{H}$-Bäume T_1, T_2. Wir wollen untersuchen, welche Struktur bei der Addition zweier Matrizen M^1, M^2 entsteht, wenn für jedes Blatt $s \in T_1$ (bzw. $t \in T_2$) der Block $M^1|_s$ (bzw. $M^2|_t$) eine $\mathbf{R}k$-Matrix ist. Gesucht ist also ein Baum $T_1 + T_2$, so daß für alle Blätter r von $T_1 + T_2$ der Block $(M^1 + M^2)|_r$ eine $\mathbf{R}2k$-Matrix ist.
Seien $s \in T_1$ und $t \in T_2$ Blätter der Bäume. Für $r := s \cap t$ gilt

$$(M^1 + M^2)|_r = (M^1|_s)|_r + (M^2|_t)|_r,$$

also ist $(M^1 + M^2)|_r$ eine $\mathbf{R}2k$-Matrix. Die Blätter des Summenbaumes sollten demnach Durchschnitte aus Blättern von T_1, T_2 sein. Die darüberliegende Hierarchie erhält man wie in Definition 3.27, indem von der Wurzel an jeweils Durchschnitte aus den Knoten der Bäume gebildet werden.

Definition 3.27 *(Addition von $\mathcal{H}$-Bäumen)*
Seien T und T' zwei $\mathcal{H}$-Bäume der Indexmenge I. Setze für alle $i \in \mathbb{N}_0$

$$\mathcal{B}_i := \{\tau \cap \tau' \mid \tau \in T^{(i)} \cup \mathcal{L}(T, \leq i-1), \tau' \in T'^{(i)} \cup \mathcal{L}(T', \leq i-1)\}.$$

Die Summe $T+T'$ der $\mathcal{H}$-Bäume mit $(T+T')^{(i)} \subset \mathcal{B}_i$ für alle $i \in \mathbb{N}_0$ wird von der Wurzel an aufsteigend definiert durch $\mathrm{root}(T+T') := I$ *und*

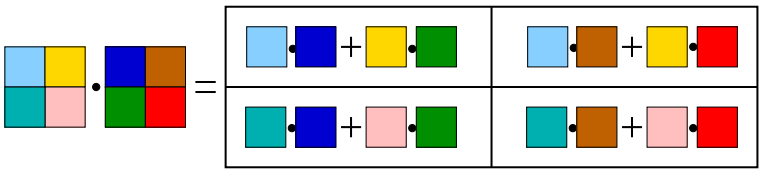
$$S_{T+T'}(\tau \cap \tau') := \begin{cases} \{\sigma \cap \sigma' \mid \sigma \in S_T(\tau), \sigma' \in S_{T'}(\tau')\} \setminus \{\emptyset\} & \tau \notin \mathcal{L}(T) \ \wedge \ \tau' \notin \mathcal{L}(T') \\ \{\sigma \cap \tau' \mid \sigma \in S_T(\tau)\} \setminus \{\emptyset\} & \tau \notin \mathcal{L}(T) \ \wedge \ \tau' \in \mathcal{L}(T') \\ \{\sigma' \cap \tau \mid \sigma' \in S_{T'}(\tau')\} \setminus \{\emptyset\} & \tau \in \mathcal{L}(T) \ \wedge \ \tau' \notin \mathcal{L}(T') \\ \emptyset & \tau \in \mathcal{L}(T) \ \wedge \ \tau' \in \mathcal{L}(T') \end{cases}.$$

Beweis der Wohldefiniertheit: *Sei $i \in \mathbb{N}_0$, $\tau, \tilde{\tau} \in T^{(i)} \cup \mathcal{L}(T, \leq i-1)$ und $\tau', \tilde{\tau}' \in T'^{(i)} \cup \mathcal{L}(T', \leq i-1)$. Wir zeigen $\tau \cap \tau' = \tilde{\tau} \cap \tilde{\tau}' \Rightarrow \tau = \tilde{\tau}$ per Induktion über i; der Induktionsanfang $i = 0$ ist klar. Sei also $\tau \cap \tau' = \tilde{\tau} \cap \tilde{\tau}'$. Der Fall $\tau \cap \tau' = \emptyset$ ist nach Definition der Söhne in $T+T'$ für die Stufen aus $\mathbb{N}$ ausgeschlossen. Aus $\tau \cap \tau' = \tau \cap \tau' \cap \tilde{\tau} \cap \tilde{\tau}' \subset \tau \cap \tilde{\tau}$ folgt $\tau \cap \tilde{\tau} \neq \emptyset$ und aus $\tau \cap \tau' = \tau \cap \tau' \cap \tilde{\tau} \cap \tilde{\tau}' \subset \tau' \cap \tilde{\tau}'$ folgt $\tau' \cap \tilde{\tau}' \neq \emptyset$. Die Knoten einer Stufe zusammen mit den Blättern auf kleineren Stufen bilden nach Bemerkung 3.4 eine Partition der Indexmenge, also muß $\tau = \tilde{\tau}$ und $\tau' = \tilde{\tau}'$ gelten.*

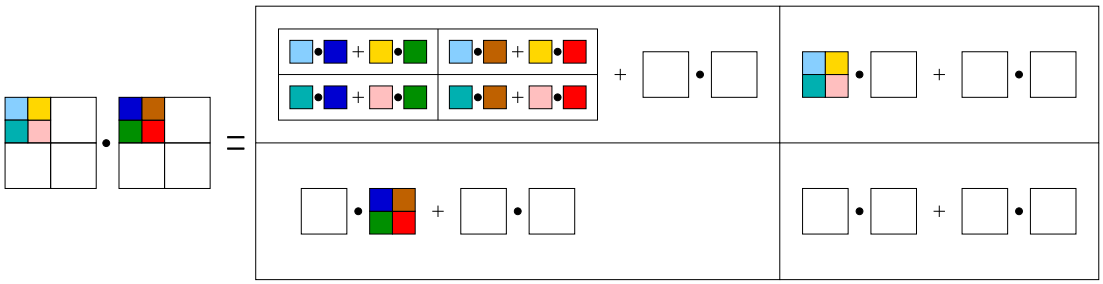
Bemerkung 3.28 *(Ergebnis der Addition)*
Die Blätter der Summe zweier $\mathcal{H}$-Bäume T, T' sind per Definition Durchschnitte aus Blättern der $\mathcal{H}$-Bäume. Die Stufenzahl $p_{T+T'}$ ist gleich dem Maximum der Stufen der $\mathcal{H}$-Bäume. Die Zahl der Knoten in $T+T'$ kann sich erheblich gegenüber T, T' auf bis zu $|T| \cdot |T'|$ erhöhen, vgl. Abbildung 9.

Abbildung 9: Beispiele für Summen von $\mathcal{H}_\times$-Bäumen (Tiefe 1)

Seien T, T' zwei $\mathcal{H}_\times$-Bäume. Wir wollen nun untersuchen, welche Struktur das Produkt zweier Matrizen M, M' hat, wenn für jedes Blatt $s \in T$ (bzw. $t \in T'$) die Matrix $M|_s$ (bzw. $M'|_t$) eine $\mathbf{R}k$-Matrix ist. Damit die auftretenden Produkte von Untermatrizen zu den durch T, T' gebildeten Blockgrenzen passen, beschränken wir uns auf den Fall, daß T aus T_I, T_J und T' aus $T_J, T_{I'}$ gebildet wurde und $T_I, T_J, T_{I'}$ Binärbäume der Tiefe > 1 sind. Das Produkt läßt sich dann wie folgt blockweise beschreiben:



In einem Block  ist das Ergebnis eine $\mathbf{R}2k$-Matrix, falls bei jedem der beiden Summanden  und  einer der Faktoren eine $\mathbf{R}k$-Matrix ist. Gehören bei einem Summanden beide Faktoren zu Knoten der Bäume, die keine Blätter sind, so läßt sich das Produkt weiter aufschlüsseln:



Die Struktur der Summe wird wie in Definition 3.27 ermittelt, allerdings gilt hier (da die $\mathcal{H}_\times$-Bäume jeweils den $\mathcal{H}$-Baum T_J gemeinsam haben) für die Blätter zu den Summanden $t \subset s$ oder $s \subset t$ falls $s \cap t \neq \emptyset$. Rekursiv erhält man die in Definition 3.29 beschriebene hierarchische Darstellung der Struktur des Produktes.

Definition 3.29 *(Produkt von $\mathcal{H}_\times$-Bäumen)*
Seien $T_I, T_J, T_{I'}$ $\mathcal{H}$-Bäume, T ein aus T_I,T_J gebildeter und T' ein aus $T_J, T_{I'}$ gebildeter $\mathcal{H}_\times$-Baum. Das Produkt $T \cdot T'$ der $\mathcal{H}$-Bäume wird von der Wurzel an aufsteigend definiert durch $\mathrm{root}(T \cdot T') := I \times I'$ *und*

$$S_{T\cdot T'}(\tau \times \tau') := \left\{\sigma \times \sigma' \mid \exists \tilde{\tau}, \tilde{\sigma} \in T_J : \sigma \times \tilde{\sigma} \in S_T(\tau \times \tilde{\tau}) \quad \wedge \quad \tilde{\sigma} \times \sigma' \in S_{T'}(\tilde{\tau} \times \tau')\right\}.$$

Bemerkung 3.30 *(Produkt ist $\mathcal{H}_\times$-Baum)*
Das Produkt aus Definition 3.29 ist ein aus $T_I, T_{I'}$ gebildeter $\mathcal{H}_\times$-Baum.

Beweis: Sei $\tau \times \tau' \in (T \cdot T')^{(i)}$ ein Knoten der Stufe i. Nach Definition ist offenbar $\tau \in T_I^{(i)}$ und $\tau' \in T_{I'}^{(i)}$, also $T \cdot T'$ aus $T_I, T_{I'}$ gebildet. Zu zeigen bleibt, daß $T \cdot T'$ ein $\mathcal{H}$-Baum ist. Sei also $\tau \times \tau'$ kein Blatt von $T \cdot T'$. Dann gibt es $\tilde{\tau} \in T_J$ derart, daß $\tau \times \tilde{\tau}$ und $\tilde{\tau} \times \tau'$ Knoten von T und T' sind, die beide keine Blätter sind. Entsprechend gehören alle Söhne zu T bzw. T' und somit $\sigma \times \sigma'$ zu $T \cdot T'$ für alle $\sigma \in S_{T_I}(\tau), \sigma' \in S_{T_{I'}}(\tau')$, d.h.

$$\tau \times \tau' = \bigcup_{\sigma \times \sigma' \in S_{T\cdot T'}(\tau \times \tau')} \sigma \times \sigma'.$$

Es bleibt die Disjunktheit zu zeigen. Ist $\sigma \times \sigma' \cap s \times s' \neq \emptyset$, so ist auch $\sigma \cap s \neq \emptyset$ und $\sigma' \cap s' \neq \emptyset$. Mit Bemerkung 3.4 folgt $\sigma = s$ und $\sigma' = s'$. ■

Beispiel 3.31 *(Produkte von $\mathcal{H}_\times$-Bäumen)*
Die $\mathcal{H}$-Bäume $T_I, T_J, T_{I'}$ sind in diesem Beispiel alle identisch zu dem in Abbildung 10 dargestellten Binärbaum, $I = J = I' = \{1, \ldots, n\}, n = 2^p, p \in \mathbb{N}$. Die $\mathcal{H}_\times$-Bäume T, T' sind die minimal zulässigen aus T_I, T_J und $T_J, T_{I'}$ gebildeten $\mathcal{H}_\times$-Bäume (siehe Beispiel 3.26) und unterscheiden sich nur in der Wahl der Zulässigkeitsbedingung Z für T und Z' für T'.

1. *Beispiel: Die Zulässigkeitsbedingungen Z und Z' seien gleich. $Z(\sigma \times \tau) =$ „zulässig“ gilt genau dann, wenn $\sigma \cap \tau = \emptyset$ ist. Dann ist das Produkt der Bäume wieder von derselben Struktur:*

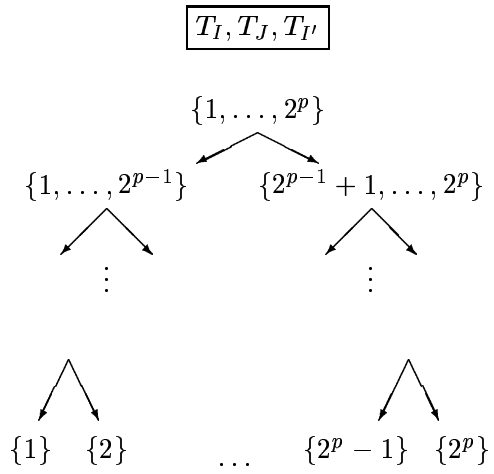
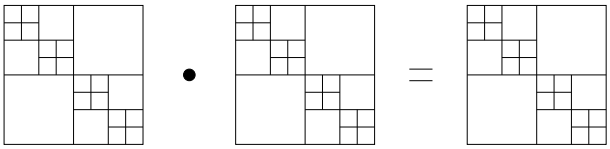
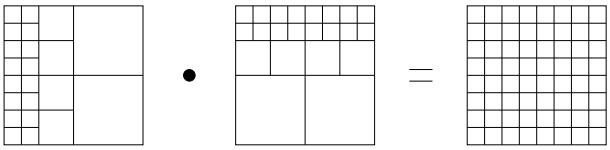


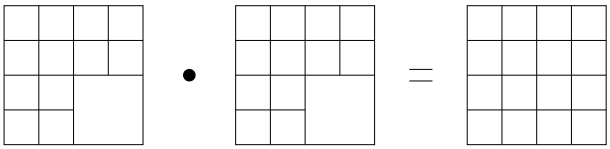
Abbildung 10: Der $\mathcal{H}$-Baum ist ein Binärbaum.



2. Beispiel: $Z(\sigma \times \tau) =$ „zulässig“ gelte genau dann, wenn $\tau \cap \{1\} = \emptyset$ ist und $Z'(\sigma \times \tau) =$ „zulässig“ gilt genau dann, wenn $\sigma \cap \{1\} = \emptyset$ ist. Die Bäume T und T' besitzen, wie man leicht nachrechnet, $O(n \log(n))$ Knoten, während $T \cdot T'$ aus $O(n^2)$ Knoten besteht:



3. Beispiel: Das $\mathcal{H}$-Baum-Produkt ist nicht idempotent (vgl. Definition 5.21), denn das Produkt kann „feiner“ sein:



4. Arithmetik Hierarchischer Matrizen

4.1. Definitionen und Notationen

Definition 4.1 *(Hierarchische Matrix, $\mathcal{H}$-Matrix, darstellbar)*
*Seien I, J zwei Mengen, T ein $\mathcal{H}_\times$-Baum von $I \times J$, Z eine Zulässigkeitsbedingung auf T und $k : \mathcal{L}(T) \to \mathbb{N}_0$ (*Rangverteilung*).*
Eine Matrix $M \in \mathbb{R}^{I \times J}$ heißt $\mathcal{H}$-Matrix bzgl. T, Z, k, falls für jedes zulässige Blatt $b \in \mathcal{L}(T)$ der korrespondierende Matrix-Block $M_b = (M_{ij})_{(i,j)\in b}$ eine $\mathbf{R}k(b)$-Matrix ist (für eine $\mathbf{R}k(b)(b)$-Matrix schreiben wir kurz $\mathbf{R}k(b)$-Matrix, da aus dem Zusammenhang immer ersichtlich ist, ob der Rang k von dem Block b abhängig ist).
Eine Matrix $A \in \mathbb{R}^{I \times J}$ heißt „als $\mathcal{H}$-Matrix darstellbar“ bzgl. T, Z, k, falls für jedes zulässige Blatt $b \in \mathcal{L}(T)$ der korrespondierende Matrix-Block $A_b = (A_{ij})_{(i,j)\in b}$ eine $\mathbf{R}_{\leq k(b)}$-Matrix ist.

Notation 4.2 *(Hierarchische Matrizen, $\mathcal{M}_{\mathcal{H},k}(T, Z)$)*
Die Menge aller $\mathcal{H}$-Matrizen bzgl. eines $\mathcal{H}_\times$-Baumes T von $I \times J$, einer Zulässigkeitsbedingung Z auf T und einer Rangverteilung $k : \mathcal{L}(T) \to \mathbb{N}_0$ wird mit $\mathcal{M}_{\mathcal{H},k}(T, Z)$ bezeichnet. Die auf diese Weise definierten Mengen $\mathcal{M}_{\mathcal{H},k}(T, Z)$ nennen wir „Klassen von $\mathcal{H}$-Matrizen“ (obwohl sie keine Klasseneinteilung der Matrizen sind, da sie sich überlappen).

Notation 4.3 *(Einschränkung und Fortsetzung einer Matrix)*
Die Einschränkung einer auf $I \times J$ definierten Matrix M auf eine Teilmenge $b \subset I \times J$ bezeichnen wir mit $M|_b$. Die Fortsetzung einer auf $I \times J$ definierten Matrix M auf eine Obermenge $b \supset I \times J$ bezeichnen wir mit $M|^b$ und definieren sie als

$$M|^b_{ij} := \begin{cases} M_{ij} & (i,j) \in I \times J, \\ 0 & \text{sonst.} \end{cases}$$

4.2. Konvertierung

Eine Konvertierung ist die Approximation einer Matrix aus einer Klasse von $\mathcal{H}$-Matrizen in einer anderen Klasse von $\mathcal{H}$-Matrizen. Dies schließt die praktische Berechnung der expliziten Darstellung mit ein. Im Fall der Konvertierung von $\mathcal{M}_{\mathcal{H},k}$ nach $\mathcal{M}_{\mathcal{H},k'}$, $k' < k$, sprechen wir auch von *Kürzen* statt *Konvertieren*. Da keine Voraussetzungen an die $\mathcal{H}$-Bäume oder Rangverteilungen gestellt werden, sind hier vollbesetzte und $\mathbf{R}k$-Matrizen mit eingeschlossen. Prinzipiell unterscheidet man die Bestapproximations-Konvertierungen, welche in einer gegebenen Norm eine Matrix mit minimalem Abstand bestimmen, und die Approximations-Konvertierungen, welche lediglich eine Näherung an eine Bestapproximation liefern.

4.2.1. Bestapproximation und Approximation

Wir fixieren zwei $\mathcal{H}_\times$-Bäume T, T' der Produkt-Indexmenge $I \times J$. Gegeben ist eine $\mathcal{H}$-Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$, gesucht ist eine Matrix $M' \in \mathcal{M}_{\mathcal{H},k'}(T', Z')$, welche unter allen

Elementen aus $\mathcal{M}_{\mathcal{H},k'}(T',Z')$ einen minimalen Abstand zu M in der Frobenius-Norm hat (diese nennen wir eine Bestapproximation bezüglich der Frobeniusnorm):

$$\begin{aligned}\|M - M'\|_F &= \min_{\tilde{M}\in\mathcal{M}_{\mathcal{H},k'(T',Z')}} \|M - \tilde{M}\|_F \\ &= \min_{\tilde{M}\in\mathcal{M}_{\mathcal{H},k'(T',Z')}} \sum_{b\in\mathcal{L}(T')} \|M|_b - \tilde{M}|_b\|_F\end{aligned}$$

Offenbar genügt es, die Approximation für alle Blätter $b \in \mathcal{L}(T')$ unabhängig voneinander durchzuführen. In nicht zulässigen Blöcken $b \in \mathcal{L}(T')$ kann man die vollbesetzte Matrix $(M_{ij})_{(i,j)\in b}$ zur exakten Darstellung verwenden (vgl. aber auch Bemerkung 2.6). In zulässigen Blöcken $b \in \mathcal{L}(T')$ benötigt man eine Rang-$k'(b)$-Approximation (und Darstellung) von $M|_b$. Es bieten sich zwei Möglichkeiten an, diese zu berechnen:

1. Die vollbesetzte Matrix $M|_b$ wird aufgestellt und anschließend mit der gekürzten Singulärwertzerlegung (Definition 2.8) für vollbesetzte Matrizen eine Bestapproximation berechnet.

2. Die Matrix $M|_b$ liegt in blockweiser $\mathbf{R}k$-Darstellung mit $b = \dot\cup_{\nu=1}^s b_\nu$ vor:

   $$M|_{b_\nu} = R_\nu, \quad R_\nu \in \mathbf{R}k(b_\nu), \quad \nu = 1\ldots, s.$$

   Ergänzt man die b_ν-Matrizen R_ν zu b-Matrizen $R_\nu|^b$, so gilt $M|_b = \sum_{\nu=1}^s R_\nu|^b$. Die Rang-$\hat{k}$-Matrix $\sum_{\nu=1}^s R_\nu|^b$ ($\hat{k} := \sum_{\nu=1}^s k(b_\nu)$) kann mit der gekürzten Singulärwertzerlegung (Algorithmus 2.12) für $\mathbf{R}k$-Matrizen auf den Rang $k'(b)$ gekürzt werden. Auch hier erhält man die Bestapproximation von $M|_b$ und erzielt für $\hat{k} \ll \sqrt{|b|}$ eine Reduktion der Kosten zur Berechnung.

Notation 4.4 *(Gekürzte Matrix $M_{\mathcal{H}}$)*
Mit $M_{\mathcal{H}}$ bezeichnen wir eine Bestapproximation (bezüglich der Frobeniusnorm) von M in $\mathcal{M}_{\mathcal{H},k'}(T',Z')$. Genau genommen müßte man die Bezeichnung $M_{\mathcal{H},k',T',Z'}$ für die Menge aller Bestapproximierenden verwenden, da aber aus dem Zusammenhang klar ist, welche Klasse von $\mathcal{H}$-Matrizen gemeint ist, und die Mehrdeutigkeit von $M_{\mathcal{H}}$ bekannt ist, verzichten wir auf die überladene Notation.

Die Berechnung der Bestapproximation bezüglich anderer Normen ist ungleich schwieriger, da sich die Approximation dann nicht für die Blätter der Matrix unabhängig voneinander durchführen läßt ($\rightarrow$ globale Optimierung nötig) und die $\mathcal{H}$-Matrix-Klassen keine Unterräume der Matrizen sind ($\rightarrow$ iterative Verfahren ungünstig). In Abschnitt 6 werden wir für spezielle $\mathcal{H}$-Matrizen die Abschätzung

$$\max_{b\in T} \|M|_b\|_2 \;\le\; \|M\|_2 \;\le\; C_{sp} p_T \| \max_{b\in T} \|M|_b\|_2$$

zeigen, so daß bis auf den Faktor p_T (Baumtiefe) die Bestapproximation in der Spektralnorm äquivalent zur blockweisen Bestapproximation in der Spektralnorm ist.
Ist man lediglich an einer (groben) Approximation der Matrix M in der Menge

$\mathcal{M}_{\mathcal{H},k'}(T',Z')$ interessiert, so kann man die Bestapproximation in den zulässigen Blättern durch die in Abschnitt 2.4 erwähnten Näherungen ersetzen und erhält so eine approximative Konvertierung.
Liegt eine Matrix X in der Nähe einer durch eine $\mathcal{H}$-Matrix approximierbaren Matrix M, so liefert das folgende Lemma eine Aussage über die Approximierbarkeit von X durch eine $\mathcal{H}$-Matrix.

Lemma 4.5 *(Störungslemma für Konvertierung)*
Seien $M, X \in \mathbb{R}^{I\times J}$ und $M_{\mathcal{H}}, X_{\mathcal{H}}$ entsprechende Bestapproximationen in $\mathcal{M}_{\mathcal{H},k'}(T',Z')$. Dann gilt

$$\begin{aligned} \|X - X_{\mathcal{H}}\|_F &\leq \|X - M\|_F + \|M - M_{\mathcal{H}}\|_F, \\ \|M - X_{\mathcal{H}}\|_F &\leq 2\|X - M\|_F + \|M - M_{\mathcal{H}}\|_F. \end{aligned}$$

Beweis: $\|X - X_{\mathcal{H}}\|_F \overset{Bestapx}{\leq} \|X - M_{\mathcal{H}}\|_F \leq \|X - M\|_F + \|M - M_{\mathcal{H}}\|_F.$ ■

4.2.2. Hierarchische Approximation

In Abschnitt 4.2.1 wurde die Approximation einer Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ durch eine Matrix $M' \in \mathcal{M}_{\mathcal{H},k'}(T',Z')$ für jedes Blatt $b \in \mathcal{L}(T')$ einzeln durchgeführt. Dies führt in den zulässigen Blättern zu der Aufgabe, eine $\mathcal{H}$-Matrix $M|_b$ durch eine $\mathbf{R}k'$-Matrix zu approximieren.
Wir fixieren einen $\mathcal{H}_\times$-Baum T der Produkt-Indexmenge $I \times J$ und eine $\mathcal{H}$-Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$. Gesucht ist eine Matrix $R \in \mathbf{R}k'(I \times J)$, welche die Matrix M „gut“ approximiert, aber schneller als die Bestapproximation zu berechnen ist. Zuerst wird für alle Blöcke $b \in \mathcal{L}(T)$ eine Rang-k'-Approximation R_b von $M|_b$ berechnet (erster Schritt in Abbildung 11). Danach wird auf den Stufen $T^{(p_T-1)}, \ldots, T^{(0)}$ für alle Blöcke b der Stufe eine Rang-k'-Approximation berechnet. Ist b kein Blatt, so können die Rang-k'-Approximationen für die Söhne von b benutzt werden: Ist $S(b) = \{b_1, \ldots, b_{|S(b)|}\}$ und sind die Rang-k'-Approximationen R_{b_i} von $M|_{b_i}$, $i = 1, \ldots, |S(b)|$ bereits berechnet, so definieren wir die Approximation R_b von $M|_b$ als die auf Rang k' gekürzte Singulärwertzerlegung der Matrix $\sum_{i=1}^{|S(b)|} R_{b_i}|^b$, deren Rang $k' \cdot |S(b)|$ ist (zweiter und dritter Schritt in Abbildung 11).
Die so berechnete hierarchische Rang-k'-Approximation $R = R_{I\times J}$ von M kann deutlich schlechter als die Bestapproximation sein.

Beispiel 4.6 *(Schlechte hierarchische Approximation)*
Sei $n \in \mathbb{N}$ und der n-stufige $\mathcal{H}$-Baum T_J der Indexmenge $J := \{1, \ldots, n\}$ definiert durch $T_J^{(i)} := \{\{1, \ldots, n-i\}, \{n+1-i\}, \ldots, \{n\}\}$. Zur Indexmenge $I := \{1, 2\}$ ist der $\mathcal{H}$-Baum T_I mit $T_I^{(i)} = \{\{1, 2\}\}$, $i = 0, \ldots, n-1$, gegeben (siehe Abbildung 12). Hier benutzen wir die in Bemerkung 3.5 erwähnte Verallgemeinerung eines $\mathcal{H}$-Baumes (selbe Indexteilmenge auf mehreren Stufen). Setze $T := T_I \otimes T_J$. Die Matrix $M \in \mathcal{M}_{\mathcal{H},1}(T,Z)$

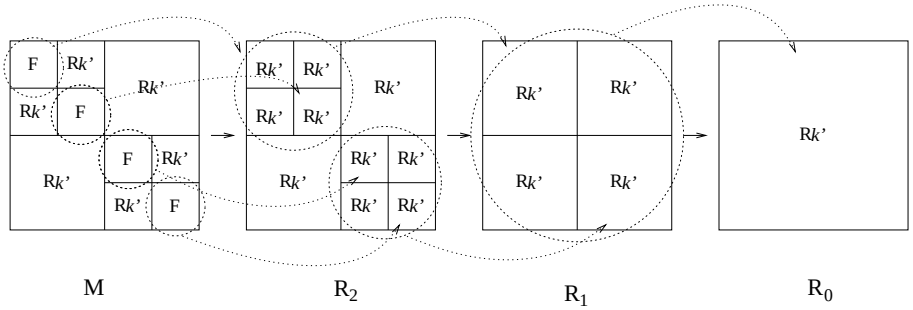


Abbildung 11: Im ersten Schritt werden alle vollbesetzten Blöcke (F) in $\mathbf{R}k'$-Matrizen ($\mathbf{R}k'$) konvertiert, danach jeweils die Söhne eines Knotens in einer $\mathbf{R}k'$-Matrix zusammengefaßt.

($Z \equiv$ „nicht zulässig“) habe für ein $\varepsilon \in \mathbb{R}_{>0}$ die Gestalt

$$M = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1+\varepsilon & 0 & \cdots & 0 \end{bmatrix}.$$

Die Rang-1-Bestapproximation von M ist

$$M^{\text{best}} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

und führt zu einem Approximationsfehler von $\|M - M^{\text{best}}\|_2 = \|M - M^{\text{best}}\|_F = 1 + \varepsilon$. Die hierarchische Rang-1-Approximation von M ist

$$M^{\text{hier}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1+\varepsilon & 0 & \cdots & 0 \end{bmatrix}$$

und führt zu einem Approximationsfehler von $\|M - M^{\text{hier}}\|_2 = \|M - M^{\text{hier}}\|_F = \sqrt{n-1}$.

Satz 4.7 *(Güte der hierarchischen Approximation)*
Die hierarchische Rang-k'-Approximation R von M erfüllt

$$\|R - M\| \leq (2^{p_T+1} + 1)\|R^{\text{best}} - M\|,$$

wobei R^{best} die Bestapproximation von M in $\mathbf{R}_{\leq k'}$ und $\|\cdot\|$ die Frobeniusnorm ist.

Beweis: Mit R_i bezeichnen wir die durch die Matrizen R_b, $b \in T^{(i)}$, auf der Stufe $i = 0, \ldots, p_T$ definierte blockweise Rang-k'-Approximation von M (siehe Abbildung 11). Von einer Stufe i zur nächstkleineren Stufe $i-1$ wurde blockweise eine Bestapproximation bestimmt, so daß

$$\|R_i - R_{i-1}\| \leq \|R_i - \tilde{R}\| \tag{12}$$

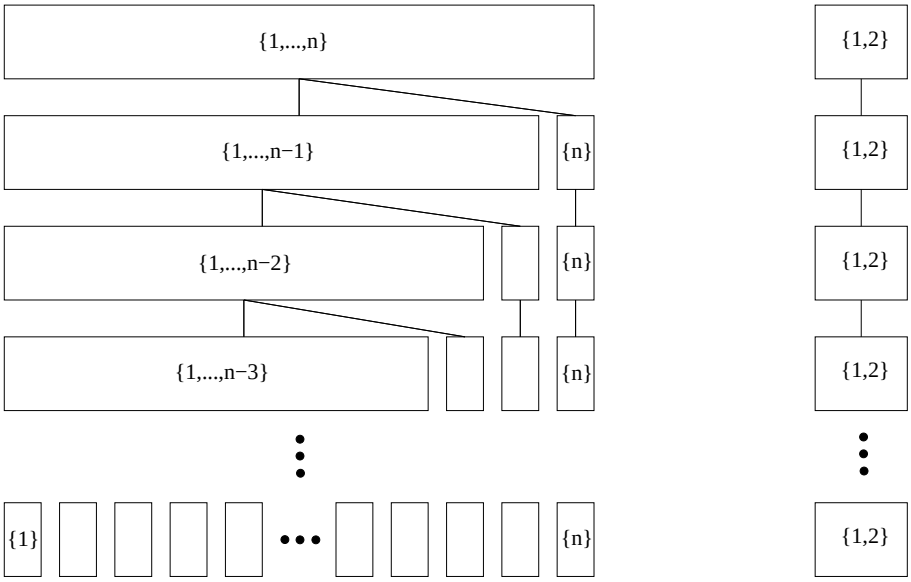


Abbildung 12: Die $\mathcal{H}$-Bäume T_J und T_I.

für alle $\tilde{R} \in \mathbb{R}^{I \times J}$ mit $\tilde{R}|_b \in \mathbf{R}_{\leq k'}(b)$ für alle $b \in T^{(i-1)}$ gilt. Insbesondere sind auch

$$\begin{aligned} \|R_i - R_{i-1}\| &\leq \|R_i - R^{\text{best}}\|, & (13) \\ \|M - R_{p_T}\| &\leq \|M - R^{\text{best}}\| & (14) \end{aligned}$$

stets erfüllt. Zunächst wird per Induktion bewiesen, daß der Abstand $\|R_i - R^{\text{best}}\|$ für $i = p_T, \ldots, 0$ durch $2^{p_T - i}\|R_{p_T} - R^{\text{best}}\|$ beschränkt ist. Der Induktionsanfang $i = p_T$ ist klar. Aus

$$\begin{aligned} \|R_i - R^{\text{best}}\| &\leq \|R_{i+1} - R_i\| + \|R_{i+1} - R^{\text{best}}\| \\ &\overset{(13)}{\leq} 2\|R_{i+1} - R^{\text{best}}\| \end{aligned}$$

schließt man den Induktionsschritt. Es folgt

$$\begin{aligned} \|R - M\| &= \|\sum_{i=0}^{p_T-1} R_i - R_{i+1} + R_{p_T} - M\| \\ &\leq \sum_{i=0}^{p_T-1} \|R_i - R_{i+1}\| + \|R_{p_T} - M\| \\ &\overset{(13)(14)}{\leq} \sum_{i=0}^{p_T-1} \|R^{\text{best}} - R_{i+1}\| + \|R^{\text{best}} - M\| \\ &\leq \sum_{i=0}^{p_T-1} 2^{p_T-i-1}\|R_{p_T} - R^{\text{best}}\| + \|R^{\text{best}} - M\| \end{aligned}$$

$$
\begin{aligned}
&= 2^{p_T-1}\sum_{i=0}^{p_T-1} 2^{-i}\|R_{p_T}-R^{\text{best}}\| + \|R^{\text{best}}-M\| \\
&\leq 2^{p_T}\|R_{p_T}-R^{\text{best}}\| + \|R^{\text{best}}-M\| \\
&\leq 2^{p_T}(\|M-R_{p_T}\| + \|M-R^{\text{best}}\|) + \|R^{\text{best}}-M\| \\
&\overset{(14)}{\leq} (2^{p_T+1}+1)\|M-R^{\text{best}}\|.
\end{aligned}
$$

■

4.3. Addition

Die Summe zweier $\mathcal{H}$-Matrizen M, M' aus der gleichen Klasse $\mathcal{M}_{\mathcal{H},k}(T,Z)$ liegt im allgemeinen nicht wieder in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ sondern in $\mathcal{M}_{\mathcal{H},2k}(T,Z)$. Liegen den beiden Matrizen nicht mehr dieselben $\mathcal{H}_\times$-Bäume zugrunde, so braucht man eine differenziertere Beschreibung der Zielstruktur.

Lemma 4.8 *(Ergebnis der Addition)*
Die Addition zweier $\mathcal{H}$-Matrizen M, M' aus unterschiedlichen Klassen $\mathcal{M}_{\mathcal{H},k}(T,Z)$, $\mathcal{M}_{\mathcal{H},k'}(T',Z')$ ergibt eine Matrix $M+M'$, die in der Klasse $\mathcal{M}_{\mathcal{H},k+k'}(T+T',Z+Z')$ darstellbar ist, wobei $+$ für $\mathcal{H}$-Bäume in Abschnitt 3.4 definiert wurde und die Rangverteilung $k+k' : T+T' \to \mathbb{N}_0$ und Zulässigkeitsbedingung $Z+Z' : T+T' \to \{$„zulässig“, „nicht zulässig“$\}$ durch

$$
\begin{aligned}
(k+k')(\tau\cap\tau') &:= k(\tau)+k'(\tau'), \\
(Z+Z')(\tau\cap\tau') &:= \begin{cases} \text{„zulässig“} & Z(\tau) = \text{„zulässig“} = Z'(\tau') \\ \text{„nicht zulässig“} & \text{sonst} \end{cases}
\end{aligned}
$$

für $\tau \in T$ und $\tau' \in T'$ definiert ist (in der Baumsumme sind alle Knoten Durchschnitt eines Knotens aus T und eines aus T').

Beweis: Sei $\tau\cap\tau' \in \mathcal{L}(T+T')$ ein zulässiges Blatt. Nach Definition von $Z+Z'$ sind $\tau \in \mathcal{L}(T)$ und $\tau' \in \mathcal{L}(T')$ beide zulässig. Es folgt $M|_\tau \in \mathbf{R}_{\leq k}(\tau)$ und $M'|_{\tau'} \in \mathbf{R}_{\leq k'}(\tau')$, also $M|_{\tau\cap\tau'} \in \mathbf{R}_{\leq k}(\tau\cap\tau')$ und $M'|_{\tau'\cap\tau} \in \mathbf{R}_{\leq k'}(\tau'\cap\tau)$. Somit ist $(M+M')|_{\tau'\cap\tau} \in \mathbf{R}_{\leq k+k'}(\tau'\cap\tau)$. ■

Von besonderem Interesse ist die *formatierte* Addition

$$
\oplus : \mathcal{M}_{\mathcal{H},k}(T,Z) \times \mathcal{M}_{\mathcal{H},k'}(T',Z') \to \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T},\tilde{Z}),
$$

die das Ergebnis $M+M'$ der Addition in eine Matrix $\tilde{M} \in \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T},\tilde{Z})$ konvertiert. Die formatierte Addition ist nicht eindeutig, da die Wahl einer geeigneten Konvertierung (üblicherweise Bestapproximation) vorausgesetzt wird und deren Ergebnis nicht eindeutig ist.

Bemerkung 4.9 *(Berechnung der formatierten Addition)*
Für die Berechnung der formatierten Addition unterscheidet man zwei Fälle: Die Bestapproximation und eine zweistufige (einfachere) Approximation. In beiden Fällen kann man es vermeiden, die Summenmatrix $M + M'$ für $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$, $M' \in \mathcal{M}_{\mathcal{H},k'}(T', Z')$, vor dem Konvertieren aufzustellen.

1. *Bestapproximation: Die Approximation wird für jedes Blatt $\tilde{b} \in \mathcal{L}(\tilde{T})$ einzeln durchgeführt. Dazu werden alle Blätter $b \in \mathcal{L}(T)$ und $b' \in \mathcal{L}(T')$ mit $b \cap \tilde{b} \neq \emptyset$ und $b' \cap \tilde{b} \neq \emptyset$ ermittelt (maximal $p_T, p_{T'}$ Stufen). Mit Hilfe der Fortsetzung erhält man die Darstellung*

$$(M + M')|_{\tilde{b}} = \sum_{\substack{b \in \mathcal{L}(T) \\ b \cap \tilde{b} \neq \emptyset}} M|_{\tilde{b} \cap b}^{\tilde{b}} + \sum_{\substack{b' \in \mathcal{L}(T') \\ b' \cap \tilde{b} \neq \emptyset}} M'|_{\tilde{b} \cap b'}^{\tilde{b}}.$$

   Ist $\tilde{b}$ nicht zulässig, so wird die exakte Summe (vollbesetzt) berechnet. Andernfalls wird die Summe wie in Abschnitt 4.2.1 gekürzt. Dafür müssen die Blöcke $M|_b, M'|_{b'}$ in nicht zulässigen Blättern als $\mathbf{R}\nu$-Matrizen behandelt werden ($\nu = \min\{|r|, |s|\}$, $\tilde{b} = r \times s$ bzw. $b' = r \times s$). Falls zuviele Blätter in der Summation nicht zulässig sind, ist es effizienter, erst eine vollbesetzte Summation und anschließend eine Singulärwertzerlegung (der vollbesetzten Matrix) zum Kürzen durchzuführen.

2. *Zweistufige Approximation: Im ersten Schritt wird die Matrix M nach $\hat{M} \in \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$ konvertiert. Danach wird zu $\hat{M}$ die Matrix M' addiert und das Ergebnis nach $\mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$ konvertiert. Die Kostenersparnis erklärt sich folgendermaßen: Werden s $\mathbf{R}k(n, m)$-Matrizen gleichzeitig addiert, so ergibt sich (Algorithmus 2.12 mit $s \cdot k$ statt k) ein Aufwand von $s^2 5(n+m)k^2 + s^3 23k^3$. Wird nach jeder einzelnen Addition das Kürzen vorgenommen, so reduziert sich der Aufwand auf*

$$(s-1)20(n+m)k^2 + (s-1)184(n+m)k^3.$$

   Bei stark unterschiedlichen Bäumen $T, T', \tilde{T}$ kann es außerdem sinnvoll sein, für jeden Knoten des Baumes eine $\mathbf{R}k(b)$- bzw. $\mathbf{R}k(b')$-Approximation R_b bzw. $R'_{b'}$ zu bestimmen (siehe Abschnitt 4.2.2) und zur Berechnung von $\tilde{M}|_{\tilde{b}}$ die $\mathbf{R}k$-Approximationen $R_b|_{\tilde{b}}, R'_{b'}|_{\tilde{b}}$ zu addieren, wobei b, b' die kleinsten $\tilde{b}$ enthaltenden Knoten in T, T' sind.

4.4. Multiplikation

In Beispiel 3.31 wurde bereits erwähnt, daß das Baumprodukt nicht idempotent ist, so daß selbst in einfachen Fällen die Struktur des Produktes zweier $\mathcal{H}$-Matrizen nicht offensichtlich ist.

Definition 4.10 *(Vorfahren eines Knotens, $\tau^{(j)}$)*
Sei T ein $\mathcal{H}$-Baum und $\tau \in T^{(i)}$. Für $j \in \{0, \ldots, i\}$ definieren wir $\tau^{(j)}$ als den Knoten aus $T^{(j)}$, der τ enthält. $\tau^{(j)}$ bezeichnen wir als den Vorfahren von τ auf der j-ten Stufe.

Lemma 4.11 *(Ergebnis der Multiplikation)*
Gegeben seien zwei $\mathcal{H}$-Matrizen M, M' aus unterschiedlichen Klassen $\mathcal{M}_{\mathcal{H},k}(T,Z)$, $\mathcal{M}_{\mathcal{H},k'}(T',Z')$, T sei ein aus T_I, T_J und T' ein aus $T_J, T_{I'}$ gebildeter $\mathcal{H}_\times$-Baum. Für ein Blatt $\tau \times \tau' \in \mathcal{L}(T \cdot T', i)$ (Baumprodukt aus Abschnitt 3.4) definieren wir die Rangverteilung $k \cdot k' : \mathcal{L}(T \cdot T') \to \mathbb{N}_0$ und Zulässigkeitsbedingung $Z \cdot Z' : \mathcal{L}(T \cdot T') \to \{$ „zulässig“, „nicht zulässig“$\}$ wie folgt:

$$U_j := \{\tilde{\tau} \in T_J^{(j)} \mid \quad \tau^{(j)} \times \tilde{\tau} \in T \ \wedge \ \tilde{\tau} \times \tau'^{(j)} \in T' \ \wedge \ (\tau^{(j)} \times \tilde{\tau} \in \mathcal{L}(T) \ \vee \ \tilde{\tau} \times \tau'^{(j)} \in \mathcal{L}(T'))\}, \quad 0 \le j \le i$$

$$(k \cdot k')(\tau \times \tau') := \sum_{j=0}^{i} \sum_{\tilde{\tau} \in U_j} \min(k(\tau^{(j)} \times \tilde{\tau}), k'(\tilde{\tau} \times \tau'^{(j)}))$$

$$(Z \cdot Z')(\tau \times \tau') \quad := \quad \begin{cases} \text{„zulässig“} & \text{falls für alle } j = 0, \ldots, i \text{ und } \tilde{\tau} \in U_j : \\ & Z(\tau^{(j)} \times \tilde{\tau}) = \text{„zulässig“ oder} \\ & Z'(\tilde{\tau} \times \tau'^{(j)}) = \text{„zulässig“} \\ \text{„nicht zulässig“} & \text{sonst.} \end{cases}$$

Das Produkt $M \cdot M'$ ist dann in $\mathcal{M}_{\mathcal{H},k\cdot k'}(T \cdot T', Z \cdot Z')$ darstellbar, und es gilt für $\tau \times \tau' \in \mathcal{L}(T \cdot T', i)$:

$$(M \cdot M')|_{\tau \times \tau'} = \sum_{j=0}^{i} \sum_{\tilde{\tau} \in U_j} \left(M|_{\tau^{(j)} \times \tilde{\tau}} \cdot M'|_{\tilde{\tau} \times \tau'^{(j)}} \right)|_{\tau \times \tau'} \tag{15}$$

Beweis: Sei $\tau \times \tau' \in \mathcal{L}(T \cdot T', i)$.

Zwischenbehauptung:

$$J = \dot{\bigcup_{0 \le j \le i}} U_j$$

Disjunktheit: Sei $\tilde{\tau}_1 \in U_j$ und $\tilde{\tau}_2 \in U_{j'}$. O.B.d.A. sei $\tau^{(j)} \times \tilde{\tau}_1 \in \mathcal{L}(T)$.

1. Fall $j = j'$: $T_J^{(j)} \cup \mathcal{L}(T_J, \le j)$ ist eine Partition von J (Bemerkung 3.4), also $\tilde{\tau}_1 = \tilde{\tau}_2$ oder $\tilde{\tau}_1 \cap \tilde{\tau}_2 = \emptyset$.
2. Fall $j' \ne j$: O.B.d.A. sei $j' < j$.
   2.a) $\tau^{(j')} \times \tilde{\tau}_2 \in \mathcal{L}(T)$: Es ist

   $$\emptyset \ne \tau \subset \tau^{(j')} \cap \tau^{(j)} \tag{16}$$

   Weil $\mathcal{L}(T)$ eine Partition von $I \times J$ ist, gilt $\tau^{(j')} \times \tilde{\tau}_2 \cap \tau^{(j)} \times \tilde{\tau}_1 = \emptyset$ oder $\tau^{(j')} \times \tilde{\tau}_2 = \tau^{(j)} \times \tilde{\tau}_1$. Letzteres ist wegen $j \ne j'$ nur für $\tilde{\tau}_1 \cap \tilde{\tau}_2 = \emptyset$ möglich, aus ersterem folgt nach Formel (16) $\tilde{\tau}_1 \cap \tilde{\tau}_2 = \emptyset$.
   2.b) $\tau^{(j')} \times \tilde{\tau}_2 \notin \mathcal{L}(T)$. Nach Definition von U_j ist dann $\tilde{\tau}_2 \times \tau'^{(j')} \in \mathcal{L}(T')$. Da $T'^{(j)} \cup \mathcal{L}(T', \le j-1)$ eine Partition von $J \times I'$ bildet (Bemerkung 3.4), folgt $\tilde{\tau}_2 \times \tau'^{(j')} \cap \tilde{\tau}_1 \times \tau'^{(j)} = \emptyset$ und insbesondere $\tilde{\tau}_2 \times \tau' \cap \tilde{\tau}_1 \times \tau' = \emptyset$, also $\tilde{\tau}_2 \cap \tilde{\tau}_1 = \emptyset$.

Überdeckung: Sei $q \in J$ und $\tilde{\tau} \in \mathcal{L}(T_J, i)$ mit $q \in \tilde{\tau}$. Es gilt $\tau^{(0)} \times \tilde{\tau}^{(0)} = I \times J \in T$ und $\tilde{\tau}^{(0)} \times \tau'^{(0)} = J \times I' \in T'$. Sind beide Kreuzprodukte keine Blätter von T bzw. T', so sind $\tau^{(1)} \times \tilde{\tau}^{(1)}$ und $\tilde{\tau}^{(1)} \times \tau'^{(1)}$ Söhne der Knoten in T bzw. T'. Sei $j \in \{0, \ldots, i\}$ der erste Index, für den $\tau^{(j)} \times \tilde{\tau}^{(j)} \in T$ oder $\tilde{\tau}^{(j)} \times \tau'^{(j)} \in T'$ ein Blatt von T bzw. T' sind. Dann ist $q \in \tilde{\tau} \subset \tilde{\tau}^{(j)} \in U_j$.

Wir erhalten

$$\begin{aligned}(M \cdot M')|_{\tau\times\tau'} &= M|_{\tau\times J} \cdot M'|_{J\times\tau'} = \sum_{j=0}^{i} \sum_{\tilde{\tau}\in U_j} M|_{\tau\times\tilde{\tau}} \cdot M'|_{\tilde{\tau}\times\tau'} \\ &= \sum_{j=0}^{i} \sum_{\tilde{\tau}\in U_j} \left(M|_{\tau^{(j)}\times\tilde{\tau}} \cdot M'|_{\tilde{\tau}\times\tau'^{(j)}}\right)|_{\tau\times\tau'}.\end{aligned}$$

Ist $\tau \times \tau'$ bzgl. $Z \cdot Z'$ zulässig, so besteht nach Definition jeder der Summanden aus einem Produkt einer $\mathbf{R}k$ mit einer beliebigen oder einer beliebigen mit einer $\mathbf{R}k'$-Matrix, d.h. der Rang der Summe ist beschränkt durch die Summe der Minima der Ränge k, k' der Faktoren. ■

Die *formatierte* Multiplikation

$$\odot : \mathcal{M}_{\mathcal{H},k}(T, Z) \times \mathcal{M}_{\mathcal{H},k'}(T', Z') \to \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z}),$$

besteht aus der Hintereinanderausführung der Multiplikation und der Konvertierung des Ergebnisses $M \cdot M'$ nach $\mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$. Ist T ein aus T_I, T_J und T' ein aus $T_J, T_{I'}$ gebildeter $\mathcal{H}_\times$-Baum, so beschreibt Lemma 4.11 das (eindeutige) Zwischenergebnis $M \cdot M'$. Die Konvertierung (Bestapproximation oder Approximation) liefert dann ein (nicht eindeutiges) Ergebnis in $\mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$.

Bemerkung 4.12 *(Berechnung der formatierten Multiplikation)*
Wir beschränken uns hier auf die Situation, daß T ein aus T_I, T_J, T' ein aus $T_J, T_{I'}$ und $\tilde{T}$ ein aus $T_I, T_{I'}$ gebildeter $\mathcal{H}_\times$-Baum ist, d.h. die Bäume „passen“: Es gilt $T \cdot T' \subset T_I \otimes T_{I'}$ und $\tilde{T} \subset T_I \otimes T_{I'}$, so daß zwei Knoten $\tau \in T \cdot T'$ und $\tilde{\tau} \in \tilde{T}$ mit $\tau \cap \tilde{\tau} \neq \emptyset$ die Bedingung „τ ist Vorfahr von $\tilde{\tau}$“ oder „$\tilde{\tau}$ ist Vorfahr von τ“ erfüllen (weil dies in $T_I \otimes T_{I'}$ gilt).
Sei $M \in \mathcal{M}_{\mathcal{H},k}(T, Z), M' \in \mathcal{M}_{\mathcal{H},k'}(T', Z')$ und $\tau \times \tau' \in \mathcal{L}(\tilde{T})$.

1. *Fall: $\tau \times \tau' \in \mathcal{L}(T \cdot T')$. In diesem Fall erhält man aus (15) die Darstellung des exakten Produktes in dem Block als eine Rang-$(k \cdot k')$-Matrix.*

2. *Fall: $\tau \times \tau' \in T \cdot T' \setminus \mathcal{L}(T \cdot T')$. Die Untermatrix $(M \cdot M')|_{\tau\times\tau'}$ liegt nun in hierarchischer Form mit Blättern wie in (15) vor.*

3. *Fall: $\tau \times \tau' \notin T$. Sei $\tau^{(j)} \times \tau'^{(j)}$ der erste Vorfahr, der in T liegt. Dieser Vorfahr besitzt die Darstellung (15), so daß die gesuchte Untermatrix die Einschränkung auf $\tau \times \tau'$ ist.*

Die Bestapproximation besteht darin, für die oben beschriebenen Darstellungen der Blätter von $M \cdot M'$ eine Rang-$\tilde{k}$-Bestapproximation mit der gekürzten Singulärwertzerlegung zu berechnen, problematisch kann dies für die im zweiten Fall entstehende hierarchische Struktur sein, die möglicherweise nicht von niedrigem Rang ist.
Eine Approximation des Ergebnisses erzielt man, indem im ersten und letzten Fall sukzessive die Summanden aus der Darstellung (15) aufaddiert werden (formatierte $\mathbf{R}k$-Addition) und im zweiten Fall eine hierarchische Approximation (Abschnitt 4.2.2) bestimmt wird.

4.5. Inversion

Sowohl die Summe als auch das Produkt zweier $\mathcal{H}$-Matrizen ließ sich exakt als $\mathcal{H}$-Matrix darstellen, wobei die Partition, Zulässigkeitsbedingung und Rangverteilung in überschaubarer expliziter Form angegeben werden konnte. Dies ist für die Inversion nicht mehr möglich, da selbst die Inverse einer schwachbesetzten Matrix im allgemeinen keine Blöcke mit niedrigem Rang aufweist, d.h. die Ergebnismatrix besitzt entweder keine zulässigen Blätter oder der Rang ist gleich der Größe des Blattes (triviale $\mathbf{R}k$-Darstellung).
Wir werden im folgenden zwei Algorithmen zur approximativen Bestimmung der Inversen einer Matrix angeben. Die erste Methode ist die Block-Gauß-Elimination zur direkten Berechnung einer approximativen Inversen mittels formatierter Arithmetik und die zweite Methode ist ein iterativer Multilevel-Ansatz.

4.5.1. Block-Gauß-Elimination

Gegeben sei ein $\mathcal{H}$-Baum T_I von I, ein aus T_I, T_I gebildeter $\mathcal{H}_\times$-Baum T und eine $\mathcal{H}$-Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$. Die blockweise Gauß-Elimination ist von der Anordnung der Indizes abhängig, daher benötigen wir für jeden Knoten $\tau \in T_I$ eine Anordnung seiner Söhne. Zusammen mit einer Anordnung der Indizes in den Blättern von T_I wird so eine Ordnung auf I definiert.

Bemerkung 4.13 *(Block-Gauß-Elimination zur Inversion in einem Knoten)*
Wir fixieren einen Knoten $\tau \times \tau \in T$ mit $S_{T_I}(\tau) = \{\sigma_1, \ldots, \sigma_s\}$. Dies definiert eine Blockstruktur $M_{ij} := M|_{\sigma_i \times \sigma_j}$, $i, j \in \{1, \ldots, s\}$, der Matrizen auf $\tau \times \tau$. Die Berechnung der Inversen $(M|_{\tau\times\tau})^{-1}$ erfolgt in $2s - 1$ Schritten, in den ersten s Schritten wird die Zerlegung $R = LM|_{\tau\times\tau}$ mit oberer Block-Dreiecksmatrix R und unterer normierter (Diagonalblöcke sind Einheitsmatrix) Block-Dreiecksmatrix L erzeugt, in den letzten $s - 1$ Schritten die Inverse $L^{-1}R$ ermittelt.
Start:

$$\begin{aligned} R_{ij}^{(0)} &:= M_{ij} \qquad i, j \in \{1, \ldots, s\} \\ L^{(0)} &:= I. \end{aligned}$$

Für $\nu = 1, \ldots, s$ setze

$$\begin{aligned}
R_{\nu\nu}^{(\nu)} &:= I \\
R_{\nu j}^{(\nu)} &:= (R_{\nu\nu}^{(\nu-1)})^{-1} R_{\nu j}^{(\nu-1)} && j = \nu+1, \ldots, s \\
R_{i\nu}^{(\nu)} &:= 0 && i = \nu+1, \ldots, s \\
R_{ij}^{(\nu)} &:= R_{ij}^{(\nu-1)} - R_{i\nu}^{(\nu-1)} R_{\nu j}^{(\nu)} && i,j = \nu+1, \ldots, s \\
L_{\nu j}^{(\nu)} &:= (R_{\nu\nu}^{(\nu-1)})^{-1} L_{\nu j}^{(\nu-1)} && j = 1, \ldots, \nu \\
L_{ij}^{(\nu)} &:= L_{ij}^{(\nu-1)} - R_{i\nu}^{(\nu-1)} L_{\nu j}^{(\nu)} && i = \nu+1, \ldots, s, \quad j = 1, \ldots, \nu.
\end{aligned}$$

Für $\nu = s, \ldots, 2$ setze

$$L_{ij}^{(2s-\nu+1)} := L_{ij}^{(2s-\nu)} - R_{i\nu}^{(s)} L_{\nu j}^{(2s-\nu)} \quad i = 1, \ldots, \nu-1, \quad j = 1, \ldots, s.$$

Beispiel 4.14 *(Block-Gauß-Elimination bei Binärbäumen)*
Sei T_I ein Binärbaum. Dann ist die Inverse eines Blockes

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

durch

$$M^{-1} = \begin{bmatrix} M_{11}^{-1} + M_{11}^{-1} M_{12} S^{-1} M_{21} M_{11}^{-1} & -M_{11}^{-1} M_{12} S^{-1} \\ -S^{-1} M_{21} M_{11}^{-1} & S^{-1} \end{bmatrix}$$

mit $S := M_{22} - M_{21} M_{11}^{-1} M_{12}$ gegeben.

Bemerkung 4.15 *(Hierarchische Block-Gauß-Elimination)*
Die Inversion der $p+1$-stufigen Matrix M läßt sich mit Hilfe von Bemerkung 4.18 auf die Multiplikation, Addition und Inversion der p-stufigen Untermatrizen $R_{ij}^{(\nu)}, L_{ij}^{(\nu)}$ zurückführen. Rekursiv erhält man so einen Algorithmus zur Inversion von M, der auf die Addition und Multiplikation von $\mathcal{H}$-Matrizen sowie die Inversion in den Blättern der Matrix aufbaut. Zur Durchführbarkeit muß gewährleistet sein, daß in allen Blättern $\tau \times \tau \in T$ die korrespondierenden Matrizen $(R_{\nu\nu}^{(\nu-1)})_{\nu=1}^{s}$ aus Bemerkung 4.18 invertierbar sind. Eine hinreichende Bedingung hierfür ist, daß die Matrix M positiv definit ist. Für die praktische Durchführbarkeit ist außerdem wichtig, wie weit die Matrizen $(R_{\nu\nu}^{(\nu-1)})_{\nu=1}^{s}$ von einer nicht invertierbaren Matrix entfernt sind ($\rightarrow$Rundungsfehler). Später werden wir die exakten Multiplikationen und Additionen durch die formatierten Versionen ersetzen, so daß der Abstand eine Schranke für den maximal zulässigen Fehler angibt.

Bemerkung 4.16 *(Zusammenhang zwischen M und $R_{\nu\nu}^{(\nu-1)}$)*
Seien $L, R \in \mathbb{R}^{I \times I}$ mit $R = LM$, R eine obere Dreiecksmatrix, L eine untere Blockdreiecksmatrix (Blockung $(T_I \otimes T_I)^{(i)}$) und $R|_b = I$ (Identität) für alle Diagonalblöcke. Aus der blockweisen Darstellung bzgl. der Partitionierung ($n := |(T_I)^{(i)}|$)

$$\begin{bmatrix} I & * & \cdots & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & I \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ & \cdots & * & L_{nn} \end{bmatrix} \cdot \begin{bmatrix} M_{11} & \cdots & \cdots & M_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ M_{n1} & \cdots & \cdots & M_{nn} \end{bmatrix}$$

liest man für die Hauptuntermatrix $M^{\underline{j}}$ (Hauptuntermatrix zu den Indizes aus den ersten j Elementen aus $(T_I)^{(i)}$)

$$\begin{bmatrix} I & * & \cdots & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & I \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ & \cdots & * & L_{jj} \end{bmatrix} \cdot \begin{bmatrix} M_{11} & \cdots & \cdots & M_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ M_{j1} & \cdots & \cdots & M_{jj} \end{bmatrix}$$

ab. Multipliziert mit der Inversen von $M^{\underline{j}}$ erhält man

$$\begin{bmatrix} I & * & \cdots & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & I \end{bmatrix} \cdot \begin{bmatrix} ((M^{\underline{j}})^{-1})_{11} & \cdots & \cdots & ((M^{\underline{j}})^{-1})_{1j} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ ((M^{\underline{j}})^{-1})_{j1} & \cdots & \cdots & ((M^{\underline{j}})^{-1})_{jj} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ & \cdots & * & L_{jj} \end{bmatrix}$$

und erhält so aus der letzten Zeile

$$((M^{\underline{j}})^{-1})_{jj} = L_{jj}.$$

Die Elemente L_{jj} sind gerade die in der Block-Gauß-Elimination auftretenden Matrizen $(R_{\nu\nu}^{(\nu-1)})^{-1}$, so daß wir schließlich die gewünschte Darstellung erreichen:

$$(R_{\nu\nu}^{(\nu-1)})^{-1} = ((M^{\underline{j}})^{-1})_{jj} \tag{17}$$

Lemma 4.17 *(Robustheit der Block-Gauß-Elimination)*
Mit $\lambda_{min}(A)$ und $\lambda_{max}(A)$ bezeichnen wir den kleinsten bzw. größten Eigenwert einer Matrix A. Sei M symmetrisch positiv definit. Dann gilt:

$$\lambda_{min}(R_{\nu\nu}^{(\nu-1)}) \geq \lambda_{min}(M), \quad \lambda_{max}(R_{\nu\nu}^{(\nu-1)}) \leq \lambda_{max}(M)$$

für alle in der Block-Gauß-Elimination zu invertierenden Diagonalblöcke $R_{\nu\nu}^{(\nu-1)}$ (Notation aus Bemerkung 4.18).

Beweis: Sei $R_{\nu\nu}^{(\nu-1)}$ ein in der Block-Gauß-Elimination auftretender zu invertierender Block in der Darstellung $((M^{\underline{j}})^{-1})_{jj}^{-1}$ (siehe (17)). Die Eigenwerte der Untermatrix $M^{\underline{j}}$ von M liegen, da M symmetrisch ist, in $[\lambda_{min}(M), \lambda_{max}(M)]$. Also liegen die Eigenwerte von $(M^{\underline{j}})^{-1}$ in $[\lambda_{max}(M)^{-1}, \lambda_{min}(M)^{-1}]$. Die Eigenwerte der Untermatrix $((M^{\underline{j}})^{-1})_{jj}$ liegen ebenfalls in diesem Intervall und die Eigenwerte der Inversen $((M^{\underline{j}})^{-1})_{jj}^{-1}$ somit wieder in $[\lambda_{min}(M), \lambda_{max}(M)]$. ■

Algorithmus 4.18 *(Formatierte Inversion $()^{\textcircled{-1}}$)*
Die formatierte *Inversion*

$$()^{\textcircled{-1}} : \mathcal{M}_{\mathcal{H},k}(T, Z) \to \mathcal{M}_{\mathcal{H},k}(T, Z)$$

wird folgendermaßen durchgeführt:

(Vorbereitung)

In $R \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ ist die zu invertierende Matrix gespeichert. Nach der Inversion soll $L \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ die approximative Inverse speichern. $H \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ ist eine Hilfsmatrix, die zur Berechnung als Zwischenspeicher benötigt wird. Der Inhalt von R wird bei der Inversion überschrieben, L und H sind zu Beginn auf Null initialisiert.

(Hierarchie)

Die Inversion wird rekursiv über den Baum T definiert. Für ein Blatt wird ein üblicher Algorithmus zur Inversion von vollbesetzten Matrizen verwendet, für andere Knoten die nachfolgende Inversion aufgerufen.

(Inversion von $R|_{\tau\times\tau}, \tau\times\tau \in T \setminus \mathcal{L}(T)$)

Sei $S_{T_I}(\tau) = \{\sigma_1,\ldots,\sigma_s\}$ und $A_{ij} := A|_{\sigma_i\times\sigma_j}$, $i,j \in \{1,\ldots,s\}$ für $A \in \{R,L,H\}$. Für $\nu = 1,\ldots,s$ setze

$$
\begin{array}{lcll}
L_{\nu\nu} & := & R_{\nu\nu}^{-1} & (H_{\nu\nu} \text{ als Zwischenspeicher}) \\
H_{\nu j} & := & L_{\nu\nu} \odot R_{\nu j} & \\
R_{\nu j} & := & H_{\nu j} & j \in \{\nu+1,\ldots,s\} \\
H_{ij} & := & -R_{i\nu} \odot R_{\nu j} & \\
R_{ij} & := & R_{ij} \oplus H_{ij} & i,j \in \{\nu+1,\ldots,s\} \\
L_{i\nu} & := & R_{i\nu} \odot L_{\nu\nu} & i \in \{\nu+1,\ldots,s\} \\
L_{\nu j} & := & -L_{\nu\nu} \odot L_{\nu j} & j \in \{1,\ldots,\nu-1\} \\
H_{ij} & := & -R_{i\nu} \odot L_{\nu j} & \\
L_{ij} & := & L_{ij} \oplus H_{ij} & i \in \{\nu+1,\ldots,s\}, j \in \{1,\ldots,\nu-1\},
\end{array}
$$

für $\nu = s,\ldots,2$ setze

$$L_{ij} := -R_{i\nu} \odot L_{\nu j} \qquad i \in \{1,\ldots,\nu-1\}, \quad j \in \{1,\ldots,\nu\}.$$

Die Wahl der in der Addition und Multiplikation enthaltenen Konvertierungen (Bestapproximation, Approximation, hierarchische Approximation) definiert verschiedene Varianten der formatierten Inversion. Die Inversion ist sowohl von der Wahl dieser Konvertierungen als auch von der Anordnung der Söhne der Knoten von T abhängig. Standardmäßig bezeichnen wir als die formatierte Inversion die Variante mit der Approximation zur Konvertierung.

Bemerkung 4.19 *(Notwendige Bedingungen für die Block-Gauß-Elimination)*
Sei $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ symmetrisch positiv definit mit kleinstem Eigenwert λ_{min}, größtem Eigenwert λ_{max}, T ein p-stufiger Baum und $Z(\tau\times\tau) =$ „nicht zulässig“ für alle $\tau\times\tau \in \mathcal{L}(T)$. Nach Lemma 4.17 liegen die Eigenwerte der in der Block-Gauß-Elimination zu invertierenden Blöcke $R|_{\tau\times\tau}$ in $[\lambda_{min},\lambda_{max}]$. Die sogenannte Approximationseigenschaft

$$\|R|_{\tau\times\tau} - R_{\mathcal{H}}\|_F \leq \varepsilon_A \lambda_{max}, \tag{18}$$

$$\|(R|_{\tau\times\tau})^{-1} - R_{\mathcal{H}^{-1}}\|_F \leq \varepsilon_A \lambda_{min}^{-1} \tag{19}$$

($R_{\mathcal{H}}$ ist eine Bestapproximation von $R|_{\tau\times\tau}$ in $\mathcal{M}_{\mathcal{H},k}(T|_{\tau\times\tau}, Z)$ und $R_{\mathcal{H}^{-1}}$ eine Bestapproximation von $(R|_{\tau\times\tau})^{-1}$ in $\mathcal{M}_{\mathcal{H},k}(T|_{\tau\times\tau}, Z)$) garantiert, daß die vor und nach der Inversion eines Blockes entstehenden Matrizen auf das Zielformat konvertiert werden können. Die so entstehenden Approximationsfehler lassen sich als Rundungsfehler der „Division" auffassen, so daß für die Fehleranalyse die bekannte Rundungsfehleranalyse der Gauß-Elimination herangezogen werden kann. Generell ist bislang nur bewiesen (vgl. [30]), daß die Fehlerverstärkung durch den Faktor $n2^n$ beschränkt ist. Es wird aber stets angemerkt, daß (noch) keine reelle Matrix bekannt ist, für die die Fehlerverstärkung bei Totalpivotsuche nicht durch n^2 beschränkt ist ([30] Seite 169 und [32] Seite 213). In dem hier beschriebenen hierarchischen Block-Gauß-Eliminationsverfahren treten also folgende Probleme auf:

1. *Es wird nicht pivotiert. Die Durchführbarkeit wird aber dennoch durch Lemma 4.17 gesichert.*
2. *Vorhandene Rundungsfehler könnten durch das Kürzen auf $\mathcal{H}$-Matrix-Formate zu höheren neuen Rundungsfehlern führen ($\rightarrow$ Lemma 4.5).*
3. *Die Fehlerverstärkung ist nur durch $n2^n$ beschränkt.*

Unter der Annahme, daß nicht pivotiert werden muß, die vorhandenen Rundungsfehler beim Konvertieren nicht verstärkt werden, die Addition und Multiplikation exakt durchgeführt wird und die Fehlerverstärkung durch ein Polynom vom Grad c in n beschränkt ist, erhalten wir:

$$\|M^{-1} - M^{\ominus 1}\|_F = O(n^c)\varepsilon_A$$

Der Fehleranalyse aus Bemerkung 4.19 liegt zugrunde, daß die Operationen $\oplus, \odot$ exakt ausgeführt werden, und nur vor und nach der Inversion eines Blockes eine Kürzung auf die Zielstruktur vorgenommen wird. Die so erhaltene Inverse nennen wir eine *Bestapproximationsinverse*. In der Praxis zeigt sich, daß kaum ein Unterschied zu der kostengünstigsten Variante mit formatierter Addition (Bestapproximation) und Multiplikation (Approximation) festzustellen ist. Für die numerischen Ergebnisse in dieser Arbeit wird daher immer diese Variante gewählt und wir sprechen von der $\mathcal{H}$-Inversen (obwohl sie nicht eindeutig ist). Für die Inversion eines nicht weiter bekannten Operators M empfiehlt sich folgendes Vorgehen: Zuerst wird die $\mathcal{H}$-Inverse bestimmt und der Fehler

$$\|I - M^{\ominus 1} \cdot M\|_2$$

(vgl. Satz 4.31) ermittelt (die $\mathcal{H}$-Inverse ist wesentlich schneller als die Bestapproximationsinverse zu berechnen). Ist der Fehler etwas zu groß, so kann man die Struktur durch Rangerhöhung anreichern und wieder eine $\mathcal{H}$-Inverse berechnen. Ist der Fehler wesentlich größer als erhofft, dann berechnet man die Bestapproximationsinverse und erhöht auch hier gegebenenfalls den Rang in den zulässigen Blättern. Diesen Prozess kann man automatisieren und so zu einer adaptiven $\mathcal{H}$-Arithmetik kommen, die in Abschnitt 6 weiter ausgeführt wird.

4.5.2. Newton-Iteration

Das Newton-Verfahren zur Lösung der Gleichung

$$X^{-1} - M = 0$$

für eine gegebene Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ führt zu der Iterationsvorschrift

$$X^{(i+1)} := 2X^{(i)} - X^{(i)} \cdot M \cdot X^{(i)}, \tag{20}$$

mit einer geeigneten Startmatrix $X^{(0)}$. Dieses Verfahren wurde bereits in [14] für die Berechnung der Inversen einer $\mathcal{H}$-Matrix vorgeschlagen und soll hier etwas genauer untersucht werden.

Lemma 4.20 *(Konvergenz bei exakter Arithmetik)*
Sei $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ eine invertierbare Matrix, $\|\cdot\|$ eine beliebige Matrixnorm und $X^{(0)} \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ eine Näherung für M^{-1} mit $\|M\|\|X^{(0)} - M^{-1}\| = q < 1$. Dann konvergiert die Newton-Iteration (20) quadratisch gegen M^{-1} und für die Iterierten gilt:

$$\|X^{(i)} - M^{-1}\| \leq \|M\|^{-1} q^{2^i}$$

Beweis: *(vgl. [14], Lemma 6.4)*
Wir beweisen die Behauptung per Induktion über i. Der Induktionsanfang ist in den Voraussetzungen an q enthalten. Es folgt mit der Bezeichnung $E^{(i)} := M^{-1} - X^{(i)}$:

$$\begin{aligned}
X^{(i+1)} &= 2X^{(i)} - X^{(i)} \cdot M \cdot X^{(i)} \\
&= 2M^{-1} - 2E^{(i)} - M^{-1} - E^{(i)} M E^{(i)} + E^{(i)} + E^{(i)} \\
&= M^{-1} - E^{(i)} M E^{(i)}, \\
\|X^{(i+1)} - M^{-1}\| &\leq \|M\|^{-1} q^{2^i} \|M\| \|M\|^{-1} q^{2^i} \\
&= \|M\|^{-1} q^{2^{i+1}}.
\end{aligned}$$

■

Beispiel 4.21 *(Keine Konvergenz bei formatierter Arithmetik)*
Ersetzt man in der Iterationsvorschrift (20) die exakten arithmetischen Operationen $+, \cdot$ durch die formatierten $\oplus$ (Bestapproximation) und $\odot$ (Bestapproximation), so konvergiert die Newton-Iteration im allgemeinen nicht gegen die Bestapproximation von M^{-1} in $\mathcal{M}_{\mathcal{H},k}(T,Z)$. In numerischen Tests zeigt sich sogar, daß sich die Approximationsgüte der $\mathcal{H}$-Inversen nicht durch eine Newton-Iteration mit $M^{\ominus 1}$ als Startwert verbessern läßt. Der Grund dafür ist, daß das Produkt $X^{(i)}M$ in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ nicht hinreichend genau dargestellt werden kann (Störung der Identität mit einer unstrukturierten Matrix).

Lemma 4.22 *(Konvergenz bei formatierter Arithmetik)*
Sei $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ invertierbar, $Y^{(0)} \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ mit

$$\|M\|_F \|Y^{(0)} - M^{-1}\|_F = q < \frac{1}{9} \tag{21}$$

und für die Inverse M^{-1} gelte die Approximationseigenschaft

$$\|M^{-1} - M_{\mathcal{H}}^{-1}\|_F \leq \varepsilon_A \leq \frac{1}{16}\|M\|_F^{-1}, \tag{22}$$

($M_{\mathcal{H}}^{-1}$ ist eine Bestapproximation von M^{-1} in $\mathcal{M}_{\mathcal{H},k}(T,Z)$). Dann gilt für die Iterierten der formatierten Newton-Iteration

$$Y^{(i+1)} := (2Y^{(i)} - Y^{(i)} \cdot M \cdot Y^{(i)})_{\mathcal{H}} \tag{23}$$

mit $X^{(0)} := Y^{(0)}$ als Startwert die Abschätzung

$$\|M^{-1} - Y^{(i)}\|_F \leq q^{2^i}\|M\|_F^{-1} + \frac{1}{4}q^{2^{i-1}}\|M\|_F^{-1} + 2\varepsilon_A. \tag{24}$$

Beweis: Wir zeigen per Induktion, daß für die Iterierten der (unformatierten) Newton-Iteration die Abschätzung

$$\begin{aligned} \|X^{(i)} - Y^{(i)}\|_F &\leq \varepsilon_i, \\ \varepsilon_0 &:= 0, \\ \varepsilon_i &:= 2\varepsilon_A + \frac{1}{4}q^{2^{i-1}}\|M\|_F^{-1}, \qquad i \in \mathbb{N} \end{aligned}$$

erfüllt ist. Der Induktionsanfang $i = 0$ ist durch die Voraussetzungen abgedeckt. Es gilt mit der Bezeichnung $\delta^{(i)} := X^{(i)} - Y^{(i)}$ und $E^{(i)} := M^{-1} - X^{(i)}$:

$$\begin{aligned} Y^{(i+1)} &= (2Y^{(i)} - Y^{(i)} \cdot M \cdot Y^{(i)})_{\mathcal{H}} \\ &= \Big(2X^{(i)} - 2\delta^{(i)} - X^{(i)} \cdot M \cdot X^{(i)} \\ &\quad + X^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot X^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)}\Big)_{\mathcal{H}} \\ &= \Big(X^{(i+1)} - 2\delta^{(i)} + X^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot X^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)}\Big)_{\mathcal{H}} \\ &= \Big(\underbrace{X^{(i+1)} - E^{(i)} \cdot M \cdot \delta^{(i)} - \delta^{(i)} \cdot M \cdot E^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)}}_{=:Y'}\Big)_{\mathcal{H}}, \end{aligned}$$

$$\begin{aligned} \|Y' - X^{(i+1)}\|_F &= \|E^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot E^{(i)} + \delta^{(i)} \cdot M \cdot \delta^{(i)}\|_F \\ &\leq 2\|E^{(i)}\|_F\|M\|_F\|\delta^{(i)}\|_F + \|\delta^{(i)}\|_F^2\|M\|_F \\ &\leq 2q^{2^i}\varepsilon_i + \|M\|_F\varepsilon_i^2 \end{aligned} \tag{25}$$

Für $i = 0$ folgt

$$\begin{aligned} \|Y^{(1)} - X^{(1)}\|_F &= \|X_{\mathcal{H}}^{(1)} - X^{(1)}\|_F \\ &\overset{L.4.5}{\leq} \|M^{-1} - X^{(1)}\|_F + \|M^{-1} - M_{\mathcal{H}}^{-1}\|_F \\ &\leq q^2\|M\|_F^{-1} + \varepsilon_A \\ &\leq \frac{1}{4}q\|M\|_F^{-1} + \varepsilon_A \end{aligned}$$

und im Fall $i \geq 1$ ist

$$\begin{aligned}
\|Y^{(i+1)} - X^{(i+1)}\|_F &\leq \|Y' - Y^{(i+1)}\|_F + \|Y' - X^{(i+1)}\|_F \\
&\leq \|Y' - M_{\mathcal{H}}^{-1}\|_F + \|Y' - X^{(i+1)}\|_F \\
&\overset{\text{Dreiecksungl.}}{\leq} \|M^{-1} - M_{\mathcal{H}}^{-1}\|_F + \|M^{-1} - X^{(i+1)}\|_F + 2\|Y' - X^{(i+1)}\|_F \\
&\overset{(25)}{\leq} 4q^{2^i}\varepsilon_i + 2\|M\|_F\varepsilon_i^2 + q^{2^{i+1}}\|M\|_F^{-1} + \varepsilon_A \\
&= 8q^{2^i}\varepsilon_A + q^{2^i}q^{2^{i-1}}\|M\|_F^{-1} \\
&\quad +8\|M\|_F\varepsilon_A^2 + 2\varepsilon_A q^{2^{i-1}} + \frac{1}{8}q^{2^{i-1}}q^{2^{i-1}}\|M\|_F^{-1} \\
&\quad +q^{2^{i+1}}\|M\|_F^{-1} + \varepsilon_A \\
&\overset{(21)(22)}{\leq} \frac{1}{8}\varepsilon_A + \frac{1}{9}q^{2^i}\|M\|_F^{-1} + \frac{1}{2}\varepsilon_A + \frac{1}{4}\varepsilon_A + \frac{1}{8}q^{2^i}\|M\|_F^{-1} \\
&\quad +\frac{1}{81}q^{2^i}\|M\|_F^{-1} + \varepsilon_A \\
&\leq 2\varepsilon_A + (\frac{1}{9} + \frac{1}{8} + \frac{1}{81})q^{2^i}\|M\|_F^{-1} \\
&= 2\varepsilon_A + \frac{161}{648}q^{2^i}\|M\|_F^{-1}.
\end{aligned}$$

■

Die formatierte Newton-Iteration (23) konvergiert nach Lemma 4.22 für hinreichend gute Startwerte und darstellbare Inverse bis auf einen Faktor von 2 gegen die Bestapproximation der Inversen in $\mathcal{M}_{\mathcal{H},k}(T,Z)$. Die Approximationsgüte ε_A der Inversen M^{-1} kann üblicherweise durch geeignete Rangwahl k die Voraussetzung (22) erfüllen. Die Startwerte $Y^{(0)}$ der Newton-Iteration hingegen sind im allgemeinen nicht gegeben. Diese kann man mit Hilfe der Block-Gauß-Elimination ($Y^{(0)} := M^{\textcircled{-1}}$) erhalten.
Die quadratische Konvergenz der Newton-Iteration scheint zu einem schnellen Inversionsalgorithmus zu führen. Die Komplexität der exakten $\mathcal{H}$-Multiplikation übersteigt jedoch die der $\mathcal{H}$-Inversion, so daß es günstiger ist, die $\mathcal{H}$-Inverse $M^{\textcircled{-1}}$ in einer angereicherten $\mathcal{H}$-Struktur zu berechnen, als die formatierte Newton-Iteration in dieser durchzuführen. Die Newton-Iteration wird dann sinnvoll, wenn die Berechnung der $\mathcal{H}$-Inversen mit der Block-Gauß-Elimination nicht mehr bis zu einer gewünschten Genauigkeit oder überhaupt nicht möglich ist. Eine alternative Methode zur Berechnung einer Startmatrix für die Newton-Iteration wird in Abschnitt 4.5.3 vorgestellt.

4.5.3. Geschachtelte Iteration

Die geschachtelte Iteration basiert auf der Idee, die Startmatrix der Newton-Iteration durch Prolongation einer approximativen Inversen auf einer gröberen Diskretisierungsstufe zu erhalten. Die approximative Inverse auf der gröberen Stufe erhält man wiederum durch die Prolongation der Inversen einer noch gröberen Stufe. Ausgehend von einer hinreichend groben Stufe i_0, auf der die Inverse exakt berechnet wird, kann man so zu der Startmatrix $Y^{(0)}$ auf der Stufe i_n kommen. Benötigt werden

- Matrizen $(M_i)_{i=i_0}^{i_n}$,
- $\mathcal{H}$-Matrix-Klassen $\mathcal{M}_{\mathcal{H},k_i}(T_i, Z_i)$, $i = i_0, \ldots, i_n$,
- Prolongationen $P^{i,i+1} : \mathcal{M}_{\mathcal{H},k_i}(T_i, Z_i) \to \mathcal{M}_{\mathcal{H},k_{i+1}}(T_{i+1}, Z_{i+1})$, $i = i_0, \ldots, i_n - 1$ und
- ein Inversionsalgorithmus auf jeder Stufe, z.B. die formatierte Newton-Iteration.

Hinreichende Voraussetzungen für das Gelingen der geschachtelten Iteration (mit der formatierten Newton-Iteration zur Inversion auf jeder Stufe) leiten sich direkt aus Lemma 4.22 ab. Die Approximationseigenschaft (22) der Räume $\mathcal{M}_{\mathcal{H},k_i}(T_i, Z_i)$ wird zu

$$\|M_i^{-1} - (M_i^{-1})_{\mathcal{H}}\|_F \leq \frac{1}{16}\|M_i\|_F^{-1} \tag{26}$$

und die Approximationsgüte (21) der Startnäherung zu

$$\|P^{i,i+1}((M_i^{-1})_{\mathcal{H}}) - M_{i+1}^{-1}\|_F \leq \frac{1}{18}\|M_{i+1}\|_F^{-1}. \tag{27}$$

Die Konstante $\frac{1}{9}$ aus (21) verringert sich auf $\frac{1}{18}$, da in der formatierten Newton-Iteration die Bestapproximation nur bis auf einen Faktor 2 erreicht wird.
Im Folgenden werden wir für ein Beispiel die Klassen $\mathcal{M}_{\mathcal{H},k_i}(T_i, Z_i)$ definieren und passende Prolongationen $P^{i,i+1}$ angeben. Diese Konstruktion läßt sich dann auf eine größere Klasse von Problemen verallgemeinern.

Problem 4.23 *(Modellproblem)*
Auf dem Gebiet $\Omega := [0,1]^2$ *soll die Poisson-Gleichung*

$$-\Delta u = f \qquad \left(\Delta := \frac{\partial}{\partial_x^2} + \frac{\partial}{\partial_y^2}\right)$$

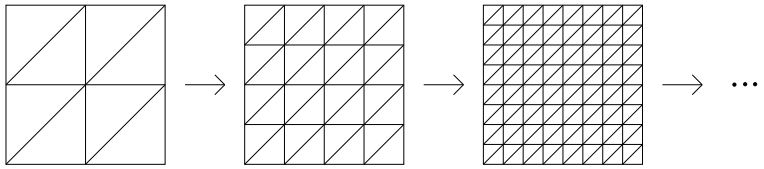
mit Dirichlet-Bedingung

$$u|_{\partial\Omega} = 0$$

für gegebenes f *gelöst werden. In der schwachen Formulierung suchen wir zu gegebener rechter Seite* $f \in H^0(\Omega)$ *ein* $u \in H_0^1(\Omega)$ *so, daß gilt:*

$$\forall v \in H_0^1(\Omega): \quad \int_\Omega < \nabla u, \nabla v > = \int_\Omega fv.$$

Problem 4.24 *(Hierarchisch diskretisiertes Modellproblem)*
Den unendlichdimensionalen Raum $H_0^1(\Omega)$ *approximieren wir durch eine Folge* $(X_i)_{i=i_0}^{i_n}$ *von endlichdimensionalen Räumen. Hier wählen wir für den* n_i*-dimensionalen Raum* X_i *die Basis* $\{\phi_1^{(i)}, \ldots, \phi_{n_i}^{(i)}\}$ *aus stetigen und stückweise (auf Dreiecken) affinen Funktionen (→ konforme Finite-Elemente-Diskretisierung). Der Träger einer Basisfunktion* $\phi_j^{(i)}$ *erstreckt sich jeweils über die an einem Gitterpunkt* $\theta_j^{(i)}$ *liegenden Dreiecke und es gelte* $\phi_j^{(i)}(\theta_k^{(i)}) = \delta_{jk}$ *(→ Lagrange-Basis). Die Triangulation erhält man durch regelmäßige Verfeinerung der Starttriangulation:*

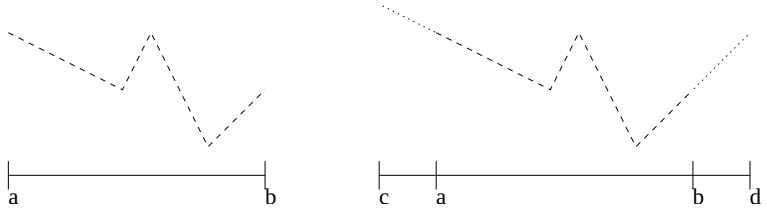


Jede Verfeinerungsstufe $i = i_0, \ldots, i_n$ definiert einen Teilraum X_i von $H_0^1(\Omega)$, und es gilt $X_i \subset X_j$ für $i_0 \leq i < j \leq i_n$. Auf jeder Stufe $i \in \{i_0, \ldots, i_n\}$ erhalten wir ein diskretes Problem

$$M_i x = f_i$$

mit der Matrix $(M_i)_{j\nu} := \int_\Omega < \nabla\phi_\nu, \nabla\phi_j >$ und Vektoren $x, f_i \in \mathbb{R}^{n_i}$. Gesucht ist eine Approximation der Inversen M_i^{-1} auf der feinsten Stufe.

Die Bedingung (26) entspricht der Approximierbarkeit der Inversen des diskreten Laplace-Operators Δ auf allen Stufen. Diese Approximierbarkeit wird in Abschnitt 7.3 behandelt. Die Voraussetzung (27) ist wesentlich schwieriger zu erfüllen und erfordert die Konstruktion einer geeigneten Prolongation, die mit möglichst geringem Aufwand durchführbar sein sollte.
Die Schachtelung der Räume $X_i \subset X_j$ erlaubt eine exakte Darstellung einer Gitterfunktion $x_i \in X_i$ in X_j; diese kanonische Prolongation nennen wir $\Pi_{j \leftarrow i}$. Ist eine Gitterfunktion nur auf einem Teil $\Omega' \subset \Omega$ definiert, so läßt sie sich linear fortsetzen:



Dies ermöglicht die Prolongation des Abschnittes einer Gitterfunktion auf einen größeren Abschnitt im nächstfeineren Gitter.

Algorithmus 4.25 *(Prolongation von Vektoren)*
Auf der Stufe i sei eine Indexteilmenge $I \subset \{1, \ldots, n_i\}$ und auf der Stufe $i+1$ eine Indexteilmenge $J \subset \{1, \ldots, n_{i+1}\}$ gegeben. Definiere zu jedem Index $j \in J$ ein nächstgelegenes Dreieck $\tau_j^{(i)} := (\theta_x^{(i)}, \theta_y^{(i)}, \theta_z^{(i)}), x, y, z \in I$. Wir fordern, daß wenigstens ein Dreieck $\tau_j^{(i)}$ die Bedingung $x, y, z \in I$ erfüllt (daß das Dreieck „in I enthalten“ ist). Die lineare Interpolation von Stützwerten v_x, v_y, v_z zu den Gitterpunkten $(\theta_x^{(i)}, \theta_y^{(i)}, \theta_z^{(i)})$ bezeichnen wir mit $\Pi_{\tau_j^{(i)}}[v]$. Die Prolongation

$$P_V^{i,i+1} : \mathbb{R}^I \to \mathbb{R}^J$$

ist für $v \in \mathbb{R}^I$ definiert durch

$$(P_V^{i,i+1}[v])_j := \Pi_{\tau_j^{(i)}}[v](\theta_j^{(i+1)})$$

Ist $I = \{1, \ldots, n_i\}$, so stimmt $P_V^{i,i+1}$ mit $\Pi_{i+1 \leftarrow i}$ überein.

Algorithmus 4.26 *(Prolongation von $\mathbf{R}k$-Matrizen)*
Sei $R' \in \mathbf{R}k(I', J'), I' \subset \{1, \ldots, n_i\}, J' \subset \{1, \ldots, n_i\}$, in der Darstellung

$$R = \sum_{\nu=1}^{k'} a^\nu b^{\nu T}$$

gegeben. Dann definieren wir die Prolongierte $R \in \mathbf{R}k(I, J), I \subset \{1, \ldots, n_{i+1}\}, J \subset \{1, \ldots, n_{i+1}\}$, von R' durch

$$R := P^{i,i+1}_{\mathbf{R}k}[R'] := \sum_{\nu=1}^{k} P^{i,i+1}_V[a^\nu] \cdot P^{i,i+1}_V[b^\nu]^T.$$

Algorithmus 4.27 *(Prolongation von $\mathcal{H}$-Matrizen)*
Gegeben sei $M' \in \mathcal{M}_{\mathcal{H},k'}(T', Z')$ und wir fixieren eine Zuordnung $\mathcal{Z} : \mathcal{L}(T) \to T'$. Gesucht ist eine Approximation $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$ an M'. Definiere für $b = r \times s \in \mathcal{L}(T), r \neq s$,

$$M|_b := \left(P^{i,i+1}_{\mathbf{R}k}[M'|_{\mathcal{Z}(t)}]\right)_{\mathcal{H},k},$$

wobei in nicht zulässigen Blättern b die kanonische $\mathbf{R}k$-Darstellung benutzt wird und, falls b zulässig und $\mathcal{Z}(b)$ nicht zulässig ist, eine Konvertierung (Bestapproximation) vorgenommen wird. Setze

$$M|_{r\times r} := 0$$

für $b = r \times r \in \mathcal{L}(T)$.

Bemerkung 4.28 *(Zur Prolongation von $\mathcal{H}$-Matrizen)*
In Algorithmus 4.27 wird die Prolongierte überall außer auf der Diagonalen definiert, auf der Diagonalen stehen Nullen. Dies hat den Grund, daß bei der Approximation von Operatoren wie aus dem Einführungsbeispiel ein singuläres (nicht glattes) Verhalten auf der Diagonalen üblich ist. Die Prolongation ist aber über die Interpolation definiert, die in diesem Fall also nicht angebracht ist. Hat man die prolongierte Matrix $P^{i,i+1}((M_i^{-1})_{\mathcal{H}})$ berechnet, so lassen sich aus

$$I = M_{i+1}M_{i+1}^{-1} \approx M_{i+1}P^{i,i+1}((M_i^{-1})_{\mathcal{H}})$$

die Diagonalelemente mittels

$$P^{i,i+1}((M_i^{-1})_{\mathcal{H}})|_t := (M_{i+1}|_t)^{-1}(I - (M_{i+1}P^{i,i+1}((M_i^{-1})_{\mathcal{H}}))|_t)$$

berechnen. Ein ähnliches Problem tritt bei der Prolongation in nicht zulässigen Blöcken $t = r \times s$ auf, diese lassen sich allerdings nicht wie die Diagonalblöcke auf einfache Weise indirekt bestimmen, deshalb muß hier die Interpolation bzw. Extrapolation genügen.

Bislang wurde nur gesagt, wie die Prolongation von einer Stufe i zur nächsten Stufe $i+1$ bei fixierten Bäumen T_i, T_{i+1} und Zuordnungen $\mathcal{Z}$ definiert ist. Die Zulässigkeitsbedingungen Z_i, Z_{i+1} sind hier, beim Laplaceoperator, die Standard-Zulässigkeitsbedingung Z_η. Die Bäume T_i werden im folgenden Algorithmus konstruiert.

Algorithmus 4.29 *(Geschachtelte Konstruktion der Bäume)*
Gegeben sei der $\mathcal{H}$-Baum $T^1_{i_0}$ für das Gitter zum Raum X_{i_0}. Sukzessive wird der Baum T^1_{i+1} aus T^1_i konstruiert.

Identische Gitterpunkte einsortieren:
Definiere den Baum $T^1_{i+\frac{1}{2}}$ als den Baum T^1_i, wobei jeder Index j, der zu einem Gitterpunkt $\theta^{(i)}_j$ gehört, durch den entsprechenden Index j' auf der nächsten Stufe zum selben Gitterpunkt $\theta^{(i+1)}_{j'} = \theta^{(i)}_j$ ersetzt wird.

Neue Gitterpunkte einsortieren:
Neue Gitterpunkte entstehen jeweils in den Mittelpunkten der Kanten der Dreiecke. Wähle zu jedem auf Stufe $i+1$ neu hinzugekommenen Gitterpunkt j' einen Gitterpunkt j der Kante, auf der j' entstanden ist. Füge j' allen Knoten von $T^1_{i+\frac{1}{2}}$ hinzu, in denen auch j vorkommt.

Große Blätter verfeinern:
Der Baum T^1_{i+1} entsteht aus $T^1_{i+\frac{1}{2}}$, indem die zu groß gewordenen Blätter weiter unterteilt werden (z.B. mit dem BSP-Algorithmus).

Zwischenergebnis: Der Cluster-Baum T^1_{i+1} ist konstruiert. Der Baum T_{i+1} ist nun der minimal aus T^1_{i+1} erzeugte $\mathcal{H}_\times$-Baum. Einem Blatt $r \times s \in T_{i+1}$ wird das eindeutig bestimmte kleinste Element $r' \times s' \in T_i$ zugeordnet, welches die Indizes zu den groben Gitterpunkten enthält.

Bemerkung 4.30 *(Offene Probleme)*
Die Rangwahl k wurde bislang ausgeklammert, da sie von dem konkreten diskretisierten Problem abhängig ist. Sie kann z.B. adaptiv wie in Abschnitt 6 erfolgen, insbesondere kann man auf diese Weise stets die Approximationseigenschaft (26) erfüllen.
Die beschriebene Prolongation $P^{i,i+1}$ besitzt eine gewisse Approximationsgüte $\|P^{i,i+1}((M_i^{-1})_{\mathcal{H}}) - M_{i+1}^{-1}\|_F$, die allerdings nicht durch eine geeignete Parameterwahl (und Inkaufnahme höherer Komplexität) beliebig verbessert werden kann.

4.6. Normen

Die Spektralnorm läßt sich im allgemeinen nicht mit vertretbarem Aufwand (proportional zum Speicheraufwand) direkt berechnen. In der Praxis genügt es allerdings, die Norm bis auf einen relativen Fehler von $\varepsilon \in [0.1, 0.01]$ zu berechnen ($\rightarrow$ Satz 4.31). Die Frobeniusnorm läßt sich über die Summe der Frobeniusnormen in den Blättern ermitteln ($\rightarrow$ Bemerkung 4.36).

Satz 4.31 *(Approximative Spektralnorm)*
Gegeben sei eine beliebige Matrix M der Dimension $n \times m$. Dann läßt sich die Spektralnorm von M bis auf einen relativen Fehler von ε mit $O(\varepsilon^{-1}(log(l) - log(\varepsilon))$ ($l := \min\{n, m\}$) Matrix-Vektor-Multiplikationen (für M) durch eine Vektoriteration berechnen. Vorausgesetzt wird hierbei, daß der „zufällige" Startvektor in der Vektoriteration

nicht im Senkrechtraum zum größten Eigenwert von M^TM liegt, sondern gleichmäßig aus allen Eigenvektoren zusammengesetzt ist (siehe auch Bemerkung 4.32). Die geschätzte Spektralnorm $\|M\|_{2,apx}$ erfüllt $\|M\|_{2,apx} \leq \|M\|_2$.

Beweis: Wir führen eine Vektoriteration für die symmetrische positiv semidefinite $l \times l$-Matrix

$$A := \begin{cases} M^TM & n > m \\ MM^T & n \leq m \end{cases}$$

durch:

$$\begin{aligned} x^{(0)} &:= random \\ x^{(i+1)} &:= \frac{Ax^{(i)}}{\|Ax^{(i)}\|_2} \\ \lambda^{(i)} &:= (x^{(i)})^T Ax^{(i)}, \qquad i = 1, 2, \ldots \end{aligned}$$

Sei dazu eine orthonormale Eigenvektorbasis $(e_j)_{j=1}^l$ mit zugehörigen Eigenwerten $(\lambda_j)_{j=1}^l$ von A fixiert. Wir nehmen ferner an, daß die λ_j absteigend sortiert seien. Es gilt dann die Darstellung

$$\begin{aligned} x^{(0)} &=: \sum_{j=1}^{l} \mu_j e^j, \quad \|x^{(0)}\|_2 = 1 \\ x^{(i)} &= \frac{\sum_{j=1}^l \mu_j \lambda_j^i e^j}{(\sum_{j=1}^l \mu_j^2 \lambda_j^{2i})^{\frac{1}{2}}} \\ \lambda^{(i)} &= \frac{\sum_{j=1}^l \mu_j^2 \lambda_j^{2i+1}}{\sum_{j=1}^l \mu_j^2 \lambda_j^{2i}}, \qquad i = 1, 2, \ldots \end{aligned}$$

Der Koeffizient μ_1 zum ersten Eigenvektor in der Darstellung von $x^{(0)}$ erfüllt für ein C_μ die Abschätzung $\mu_1^2 \geq (C_\mu l)^{-1}$ (gleichmäßige Zusammensetzung von $x^{(0)}$ aus allen Eigenvektoren). Die Approximationsgüte ε ist vorgegeben, und wir setzen

$$k := \max\{j \in \{1, \ldots, l\} \mid \lambda_j \geq (1 - \frac{\varepsilon}{2})\lambda_1\}.$$

Zwischenbehauptung: $\lambda^{(i)} \leq \lambda^{(i+1)}$ für alle $i \in \mathbb{N}$.
Beweis: Es gelten die Umformungen

$$\begin{aligned} & & \lambda^{(i)} &\leq \lambda^{(i+1)} \\ \Leftrightarrow & \left(\sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i+1}\right)\left(\sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i+2}\right) &\leq & \left(\sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i+3}\right)\left(\sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i}\right) \\ \Leftrightarrow & \sum_{j=1}^{l}\sum_{\nu=1}^{l} \mu_j^2 \mu_\nu^2 \lambda_j^{2i+1} \lambda_\nu^{2i+2} &\leq & \sum_{j=1}^{l}\sum_{\nu=1}^{l} \mu_j^2 \mu_\nu^2 \lambda_j^{2i} \lambda_\nu^{2i+3} \end{aligned}$$

$$
\begin{array}{lll}
\Leftrightarrow & \displaystyle\sum_{j=1}^{l}\sum_{\nu=j+1}^{l} \mu_j^2\mu_\nu^2(\lambda_j^{2i+1}\lambda_\nu^{2i+2}+\lambda_j^{2i+2}\lambda_\nu^{2i+1}) & \leq \displaystyle\sum_{j=1}^{l}\sum_{\nu=j+1}^{l} \mu_j^2\mu_\nu^2(\lambda_j^{2i}\lambda_\nu^{2i+3}+\lambda_j^{2i+3}\lambda_\nu^{2i}) \\
\Leftarrow \quad \forall j<\nu: & \mu_j^2\mu_\nu^2(\lambda_j^{2i+1}\lambda_\nu^{2i+2}+\lambda_j^{2i+2}\lambda_\nu^{2i+1}) & \leq \mu_j^2\mu_\nu^2(\lambda_j^{2i}\lambda_\nu^{2i+3}+\lambda_j^{2i+3}\lambda_\nu^{2i}) \\
\Leftarrow \quad \forall j<\nu: & \lambda_j\lambda_\nu^2+\lambda_j^2\lambda_\nu & \leq \lambda_\nu^3+\lambda_j^3 \\
\Leftarrow \quad \forall j<\nu: & (\lambda_j-\lambda_\nu)\lambda_\nu^2 & \leq (\lambda_j-\lambda_\nu)\lambda_j^2
\end{array}
$$

Damit ist die Zwischenbehauptung bewiesen, aus ihr folgt $\|M\|_{2,apx} \leq \|M\|_2$. Wir wollen den ersten Index i finden, ab dem $\lambda^{(i)} \geq (1-\varepsilon)\lambda_1$ erfüllt ist, d.h.

$$
\lambda_1 - \lambda^{(i)} = \frac{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}(\lambda_1-\lambda_j)}{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}} \leq \varepsilon\lambda_1.
$$

Aus den Ungleichungen

$$
\frac{\sum_{j=1}^{k}\mu_j^2\lambda_j^{2i}(\lambda_1-\lambda_j)}{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}} \overset{\text{Def. } k}{\leq} \frac{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}(\frac{\varepsilon}{2}\lambda_1)}{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}} \leq \frac{\varepsilon}{2}\lambda_1
$$

und

$$
\begin{array}{lll}
\dfrac{\sum_{j=k+1}^{l}\mu_j^2\lambda_j^{2i}(\lambda_1-\lambda_j)}{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}} & \leq & \lambda_1\dfrac{\sum_{j=k+1}^{l}\mu_j^2\lambda_j^{2i}}{\sum_{j=1}^{l}\mu_j^2\lambda_j^{2i}} \\
& \overset{\text{Def. } k}{\leq} & \lambda_1\dfrac{\sum_{j=k+1}^{l}\mu_j^2((1-\frac{\varepsilon}{2})\lambda_1)^{2i}}{\mu_1^2\lambda_1^{2i}} \\
& \leq & \lambda_1(1-\dfrac{\varepsilon}{2})^{2i}\dfrac{\sum_{j=1}^{l}\mu_j^2}{\mu_1^2} \\
& \leq & \lambda_1(1-\dfrac{\varepsilon}{2})^{2i}\mu_1^{-2} \\
& \leq & \lambda_1(1-\dfrac{\varepsilon}{2})^{2i}C_\mu l
\end{array}
$$

ergeben sich die äquivalenten Bedingungen

$$
\begin{array}{rcl}
(1-\dfrac{\varepsilon}{2})^{2i}C_\mu l & \leq & \dfrac{\varepsilon}{2}, \\
i & \geq & \dfrac{\log(\varepsilon/2)-\log(C_\mu l)}{2\log(1-\varepsilon/2)},
\end{array}
$$

welche für $\varepsilon \to 0$, wegen $\log(1-\varepsilon/2) \to -\varepsilon/2$, äquivalent sind zu

$$
i \geq \varepsilon^{-1}(\log(l)+\log(C_\mu)-\log(\varepsilon/2)).
$$

■

Bemerkung 4.32 *(Startwerte in der Vektoriteration)*
Die Konvergenz der Vektoriteration aus Satz 4.31 hängt von den zufälligen Startwerten ab. Ist

$$x^{(0)} = \sum_{j=1}^{l} \mu_j e^j \qquad \left(\{e^1, \ldots, e^l\} \text{ Orthonormalbasis}, \quad \sum_{j=1}^{l} \mu_j^2 = 1 \right)$$

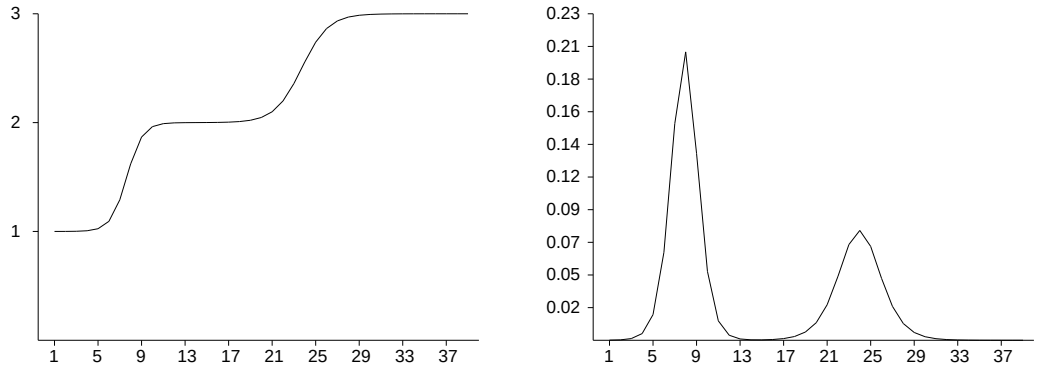
der Startvektor der Iteration für die $l \times l$-Matrix A, so ist der Aufwand der Iteration bis zu einer relativen Genauigkeit von ε mit $1 - \varepsilon \in [0, 1[$ durch

$$N_{2norm} := \frac{\log(\varepsilon/2) - \log(\mu_1^{-2})}{2\log(1 - \varepsilon/2)}$$

Matrix-Vektor-Multiplikationen beschränkt. Die stochastische Größe μ_1^{-2} geht nur logarithmisch in diese Abschätzung ein

Bemerkung 4.33 *(Adaptive Schätzung der Spektralnorm)*
In der Praxis zeigt sich, daß die Approximation der Spektralnorm unabhängig von der Größe der Matrix oft mit sehr wenigen (< 5) Schritten der Vektoriteration hinreichend genau (10% rel. Fehler) wird (siehe Beispiel 4.35). Es stellt sich daher die Frage, ob man anhand der Folge der geschätzten Eigenwerte $\lambda^{(i)}$ eine Prognose über die Nähe zum Eigenwert λ_1 erhalten kann. Bislang wissen wir nur, daß die Konvergenz monoton ist und nach einer bestimmten Anzahl von Iterationsschritten die Schätzung für die Spektralnorm hinreichend genau ist. Da die Konvergenz unabhängig von der Fehlertoleranz ε ist, wird man kaum ein geeignetes scharfes Abbruchkriterium finden. Oft wird der Ausweg vorgeschlagen, eine maximale Zahl von Iterationen vorzugeben und vorher abzubrechen, falls der Zuwachs $(\lambda^{(i+1)} - \lambda^{(i)})/\lambda^{(i+1)}$ sehr klein ist. Diese Strategie versagt jedoch, falls der Startvektor ungünstig gewählt wurde (siehe Beispiel 4.34).

Beispiel 4.34 *(Konvergenzverhalten der Vektoriteration)*
Führt man die Vektoriteration zur Bestimmung des größten Eigenwertes der Matrix $A := \mathrm{diag}(1, 2, 3)$ mit den Startwerten $x^{(0)} := (10^4, 10^2, 10^{-2})/\sqrt{10^8 + 10^4 + 10^{-4}}$ durch, so erhält man nach 22 Schritten eine bis auf einen relativen Fehler von $\varepsilon = 0.3$ genaue Schätzung für den größten Eigenwert. Die geschätzte nötige Iterationszahl ist $N_{2norm} = 49$ ($C_\mu = 10^6/3$). In der linken Abbildung sind die Rayleigh-Quotienten und in der rechten Abbildung die Zuwächse nach den einzelnen Iterationsschritten dargestellt.



Ein Abbruch bei „kleinen“ Zuwächsen hätte eine Schätzung von 1 *für die Norm der Matrix zur Folge gehabt, und wenn man erst nach Verringerung der Zuwächse abbricht eine Schätzung von* 2.

Beispiel 4.35 *(Konvergenz der Vektoriteration für die Spektralnorm)*
Die Matrix M_n sei die Finite-Elemente-Diskretisierung des Laplace-Operators auf dem Einheitsquadrat mit elementweise affinen Basisfunktionen bei regelmäßiger Gitterstruktur und n Freiheitsgraden (Problem 4.23 und Problem 4.24). Die Eigenwerte von M_n liegen in $[0,8]$. Die Anzahl der Vektoriterationsschritte i, die benötigt werden, um eine relative Genauigkeit von ε zu erreichen, ist in der folgenden Tabelle für zunehmende n aufgelistet:

ε $n=$	1024	4096	16384	65536	262144	$\frac{\lambda^{(i+1)}-\lambda^{(i)}}{\lambda^{(i+1)}}$
1/2	1	1	1	1	1	
1/4	1	1	1	1	1	
1/8	1	1	1	1	1	0.7
1/16	2	2	2	2	2	0.09
1/32	5	4	4	4	4	0.02
1/64	8	8	8	8	8	0.004
1/128	17	17	17	16	16	0.0009
1/256	41	33	34	33	32	0.0002
1/512	66	57	67	67	63	0.00006
1/1024	97	101	129	130	123	0.00002

Man sieht sehr deutlich die Abhängigkeit $O(\varepsilon^{-1})$, allerdings führt die gleichmäßige Verteilung der Eigenwerte dazu, daß die Konvergenz unabhängig von der Dimension n ist. Die Eigenwerte liegen für $n \to \infty$ dicht in $[0,8]$, ihr Verhältnis zueinander wirkt sich aber offenbar nicht auf die Konvergenz aus.

Bemerkung 4.36 *(Berechnung der Frobeniusnorm)*
Für die Frobeniusnorm einer Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$, $\operatorname{root}(T) = I \times J$, gilt

$$\|M\|_F = \sqrt{\sum_{i\in I}\sum_{j\in J} M_{ij}^2} = \sqrt{\sum_{t\in\mathcal{L}(T)}\sum_{(i,j)\in t} M_{ij}^2} = \sqrt{\sum_{t\in\mathcal{L}(T)} \|M|_t\|_F^2}.$$

In den nicht zulässigen Blättern ist $\|M|_t\|_F^2 = \sum_{(i,j)\in t} M_{ij}^2$ zu berechnen, in den zulässigen Blättern erhält man die Frobeniusnorm wie in Abschnitt 2.6.

5. Komplexität für allgemeine Hierarchien

Die Komplexität arithmetischer Operationen von $\mathcal{H}$-Matrizen läßt sich unter geringen Voraussetzungen auf die Komplexität zur Speicherung der Matrizen zurückführen. Zur Analyse der Komplexität für konkrete Probleme ist also lediglich die Größe der aufzustellenden Matrix zu bestimmen.
Die Komplexität werden wir für $\mathcal{H}$-Matrizen mit *schwachbesetzter Blockstruktur* abschätzen. Im Fall der Multiplikation und Inversion wird zusätzlich noch die *Fast-Idempotenz* der Blockstruktur benötigt. Für eine große Klasse von Partitionierungen werden wir diese beiden Eigenschaften nachweisen.

Notation 5.1 *($\mathcal{L}^+, \mathcal{L}^-$)*
Sei T eine $\mathcal{H}$-Partition und Z eine zugehörige Zulässigkeitsbedingung. Dann definieren wir

$$\begin{aligned}
\mathcal{L}^+(T) &:= Z^{-1}(\text{„zulässig“}) \cap \mathcal{L}(T) \\
\mathcal{L}^-(T) &:= Z^{-1}(\text{„nicht zulässig“}) \cap \mathcal{L}(T) \\
\mathcal{L}^+(T,i) &:= Z^{-1}(\text{„zulässig“}) \cap \mathcal{L}(T,i), \qquad i = 0,\dots,p_T \\
\mathcal{L}^-(T,i) &:= Z^{-1}(\text{„nicht zulässig“}) \cap \mathcal{L}(T,i), \quad i = 0,\dots,p_T.
\end{aligned}$$

Notation 5.2 *(Aufwandskonstanten N_*)*
Der Aufwand einer Operation O in einer Klasse X zu den Parametern $P_1,\dots$ wird stets mit $N_{X,O}(P_1,\dots)$ bezeichnet. Den Aufwand zur Speicherung einer $n\times m$-$\mathbb{R}k$-Matrix bzw. unstrukturierten vollbesetzten Matrix bezeichnen wir entsprechend mit $N_{\mathbb{R}k,St}(n,m)$ bzw. $N_{F,St}(n,m)$ (F steht für full*):*

$$\begin{aligned}
N_{\mathbb{R}k,St}(n,m) &= k(n+m), \\
N_{F,St}(n,m) &= nm.
\end{aligned}$$

Für die Matrix-Vektor-Multiplikation gilt nach Abschnitt 2.2

$$\begin{aligned}
k(n+m) \le\ & N_{\mathbb{R}k\cdot V}(n,m) \le 2k(n+m), \\
nm \le\ & N_{F\cdot V}(n,m) \le 2nm.
\end{aligned}$$

5.1. Speicherbedarf

Definition 5.3 *(Schwachbesetzte Blockstruktur)*
Ein aus T_I, T_J gebildeter $\mathcal{H}_\times$-Baum T heißt schwachbesetzt *zur Konstante C_{sp} („sp“ steht für sparse), falls*

$$\begin{aligned}
\forall i \in \{0,\dots,p_{T_I}\} \quad \forall \tau \in T_I^{(i)} :\ & |\{\tau' \in T_J^{(i)} \mid \tau\times\tau' \in T^{(i)}\}| \le C_{sp}, \\
\forall i \in \{0,\dots,p_{T_J}\} \quad \forall \tau' \in T_J^{(i)} :\ & |\{\tau \in T_I^{(i)} \mid \tau\times\tau' \in T^{(i)}\}| \le C_{sp}.
\end{aligned}$$

Bemerkung 5.4 *(Zur Schwachbesetztheit)*
Die Bedingungen aus Definition 5.3 lassen sich für die Standard-Zulässigkeitsbedingung Z_η vereinfachen. Sei dazu $D(\tau)$ die die Geometrie der Basisfunktionen zu den Indizes aus τ charakterisierende Menge (siehe Abschnitt 3.2.1). Ist $\sigma \times \sigma'$ ein Knoten von T, so kann der Vater-Knoten $\tau \times \tau'$ nicht zulässig sein (für minimal zulässige $\mathcal{H}_\times$-Bäume T). Ein minimal (bzgl. Z_η) zulässiger $\mathcal{H}_\times$-Baum T ist also schwachbesetzt, falls
$\forall i \in \{0, \ldots, p_{T_I}\} \forall \tau \in T_I^{(i)}$:

$$\left| \left\{ \tau' \in T_J^{(i)} \;\middle|\; \operatorname{dist}(D(\tau), D(\tau')) \leq \frac{\min(\operatorname{diam}(D(\tau)), \operatorname{diam}(D(\tau')))}{2\eta} \right\} \right| \leq \frac{1}{s} C_{sp},$$

s = maximale Zahl der Söhne eines Knotens $\tau' \in T_J$, analog für $\tau \in T_J^{(i)}$.

Beispiel 5.5 *(Modellproblem)*
In dem Modellproblem aus Beispiel 3.13 bestehen die Cluster gerade aus den Indizes zu Punkten in einem Teilrechteck des Gebietes. Bis auf die Gitterweite $h = 1/(n+1)$ liegen die Träger von Knoten-Basisfunktionen (linear/bilinear) zu den Punkten eines Clusters in der Bounding-Box des Clusters. Für den Parameter $\eta = \sqrt{2}(\frac{1}{2} + h)$ sind bzgl. der Standard-Zulässigkeitsbedingung Z_η auf einer geraden Stufe genau die Paare τ, τ' von Clustern, die sich berühren, nicht zulässig, auf einer ungeraden Stufe kommt in vertikaler Richtung eine Schicht hinzu. In Abbildung 13 ist die Situation für einen Cluster τ dargestellt. Die Schwachbesetztheit soll bzgl. des Clusters σ getestet werden. Von einer ungeraden zu einer geraden Stufe sind neun Vatercluster mit je zwei Söhnen nicht zulässig, also $C_{sp} \geq 18$ hinreichend, andernfalls sind 15 Vatercluster nicht zulässig, so daß $C_{sp} \geq 30$ hinreichend ist.

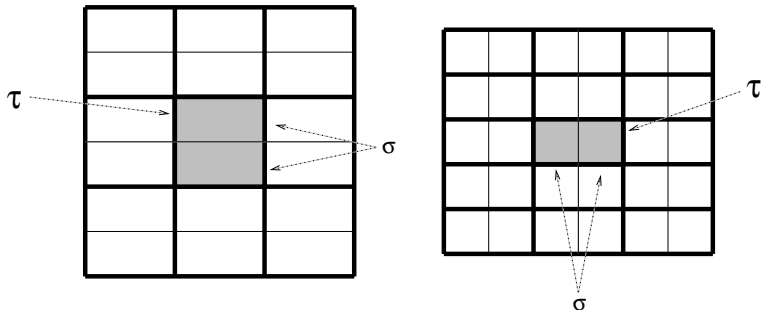


Abbildung 13: In der Mitte liegt der (dunkle) Cluster τ, um ihn herum die anderen nicht zulässigen Cluster derselben Stufe. Die dünnen Linien gehören zu den Söhnen. Die Zulässigkeit soll für einen der beiden Söhne σ von τ getestet werden.

Lemma 5.6 *(Global geometrisch balancierter Fall)*
Wir fixieren eine zu partitionierende Indexmenge I mit dazu gehörenden die Geometrie

beschreibenden Mengen $\tau_i \subset \mathbb{R}^d$, $i \in I$ (siehe Abschnitt 3.2.1). Diese Mengen könnten z.B. Dreiecke aus einer Oberflächentriangulation in $\mathbb{R}^{d=3}$ sein. Setze $\Omega := \bigcup_{i\in I} \tau_i$. Ω sei in dem Würfel $m_\Omega + [0,H)^d$ enthalten, $m_\Omega \in \mathbb{R}^d$. Die Mengen τ_i seien lokal, d.h. es gibt ein $\bar{h} \in \mathbb{R}$ mit

$$\tau_i \subset m_i + [0,\bar{h})^d,$$

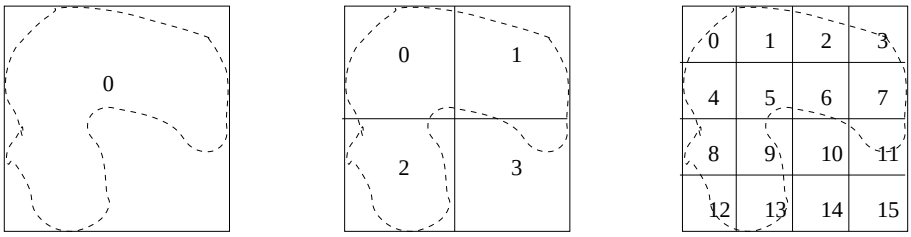
wobei $m_i \in m_\Omega + [0,H)^d$. Ferner seien die Mengen τ_i separierbar in dem Sinne, daß es ein $\underline{h} \in \mathbb{R}$ gibt, so daß für alle $x \in \mathbb{R}^d$ gilt

$$\left| \left\{ i \in I \mid \tau_i \cap (x + [0,\underline{h})^d) \neq \emptyset \right\} \right| \leq C_{sep}.$$

Dann läßt sich ein schwachbesetzter bzgl. Z_η zulässiger $\mathcal{H}$-Baum von I durch geometrisch balancierte BSP erzeugen: Setze für jede Stufe $l = 0,1,...$ und jeden Index $j = \sum_{i=1}^d 2^{l(d-i)} x_i$, $x_i \in \{0,\ldots,2^l-1\}$,

$$\begin{aligned} y_i &:= x_i 2^{-l} H, \quad i = 1,\ldots,d, \quad y = (y_1,\ldots,y_d), \\ I_j^{(l)} &:= \{ i \in I \mid m_i - m_\Omega \in y + [0,2^{-l}H)^d \}. \end{aligned}$$

Erklärung: *Auf der Stufe l wird der Würfel $m_\Omega + [0,H)^d$, der das Gebiet Ω enthält, in $(2^l)^d$ gleichgroße Würfel aufgeteilt. Ein Index $i \in I$ gehört zu $I_j^{(l)}$, falls der Mittelpunkt m_i in dem j-ten Würfel der Stufe l liegt. Die Indizes j werden in der Form $j = \sum_{i=1}^d 2^{l(d-i)} x_i$ geschrieben, so daß x_i angibt, der wievielte Würfel in der i-ten Koordinatenrichtung gemeint ist. In der folgenden Abbildung sind die Indizes j zu den Würfeln der Stufen $0,1,2$ eingetragen:*



Die Wurzel des $\mathcal{H}$-Baumes T_I ist $I = I_0^{(0)}$ und die 2^d Söhne eines Knotens $I_j^{(l)}$ für $j = \sum_{i=1}^d 2^{l(d-i)} x_i$ sind

$$S_{T_I}(I_j^{(l)}) = \left\{ I_{j'}^{(l+1)} \;\middle|\; j' = \sum_{i=1}^d 2^{(l+1)(d-i)} x_i',\ x_i' \in \{2x_i, 2x_i+1\} \right\}.$$

Der $\mathcal{H}_\times$-Baum T sei der wie in Beispiel 3.26 minimal zulässige aus T_I gebildete $\mathcal{H}_\times$-Baum. Dann ist T schwachbesetzt mit der Konstante

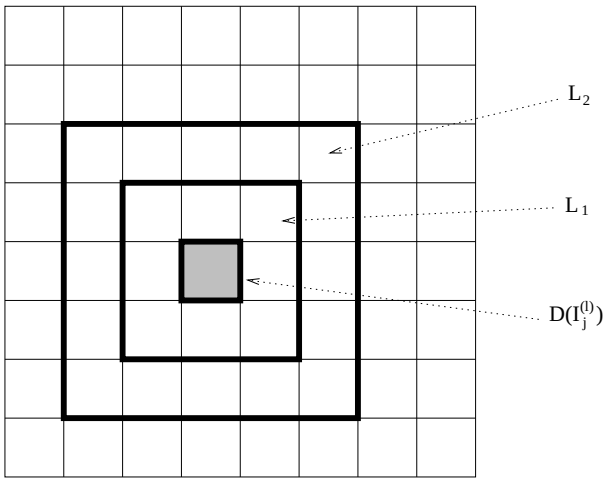
$$C_{sp} := 2^d (2L+1)^d, \qquad L := \left\lceil \frac{\sqrt{d}}{2} \eta^{-1} + (\sqrt{d} + 4\eta) \bar{h} 2^{l-1} H^{-1} \eta^{-1} \right\rceil = O(\eta^{-1}).$$

Die Anzahl der Stufen von T ist für $b_{min} \geq C_{sep}$ beschränkt durch

$$p_T \leq \left\lceil \log_2(\frac{H}{\underline{h}}) \right\rceil .$$

Beweis: Baumtiefe p_T: Nach Definition der $I_j^{(l)}$ und der Voraussetzung an die Separabilität der Mengen τ_i gilt $|I_j^{(l)}| \leq C_{sep}$, falls $2^{-l}H \leq \underline{h}$, also falls $l \geq \log_2(\frac{H}{\underline{h}})$.
Schwachbesetztheit C_{sp}:
Der Durchmesser von $D(I_j^{(l)})$ ist wegen der Lokalität der τ_i beschränkt durch $\sqrt{d}(H2^{-l} + \bar{h})$. Der Abstand zweier Mengen $D(I_j^{(l)}), D(I_{j'}^{(l)})$ wird schichtweise betrachtet:

$$\begin{aligned} L_1 &:= \{D(I_{j'}^{(l)}) \mid \mathrm{dist}(D(I_{j'}^{(l)}), D(I_j^{(l)})) = 0\}, \\ L_i &:= \{D(I_{j'}^{(l)}) \mid \mathrm{dist}(D(I_{j'}^{(l)}), L_{i-1}) = 0\} \setminus L_{i-1}. \end{aligned}$$



Die Elemente der $(i+1)$-ten Schicht haben einen Abstand von mindestens $H2^{-l}i - 2\bar{h}$ zu $D(I_j^{(l)})$. Es folgt für $D(I_{j'}^{(l)}) \in L_{i+1}$:

$$\begin{aligned} \min(\mathrm{diam}(D(I_j^{(l)})), \mathrm{diam}(D(I_{j'}^{(l)}))) &\leq \sqrt{d}(H2^{-l} + \bar{h}), \\ 2\eta \mathrm{dist}(D(I_j^{(l)}), D(I_{j'}^{(l)})) &\geq 2\eta(H2^{-l}i - 2\bar{h}), \end{aligned}$$

$$\begin{aligned} & \min(\mathrm{diam}(D(I_j^{(l)})), \mathrm{diam}(D(I_{j'}^{(l)}))) \leq 2\eta \mathrm{dist}(D(I_j^{(l)}), D(I_{j'}^{(l)})) \\ \Leftarrow\ & \sqrt{d}(H2^{-l} + \bar{h}) \leq 2\eta(H2^{-l}i - 2\bar{h}) \\ \Leftarrow\ & \sqrt{d}H2^{-l} + (\sqrt{d} + 4\eta)\bar{h} \leq 2\eta H 2^{-l} i \\ \Leftarrow\ & \frac{\sqrt{d}}{2}\eta^{-1} + (\sqrt{d} + 4\eta)\bar{h}H^{-1}2^{l-1}\eta^{-1} \leq i. \end{aligned}$$

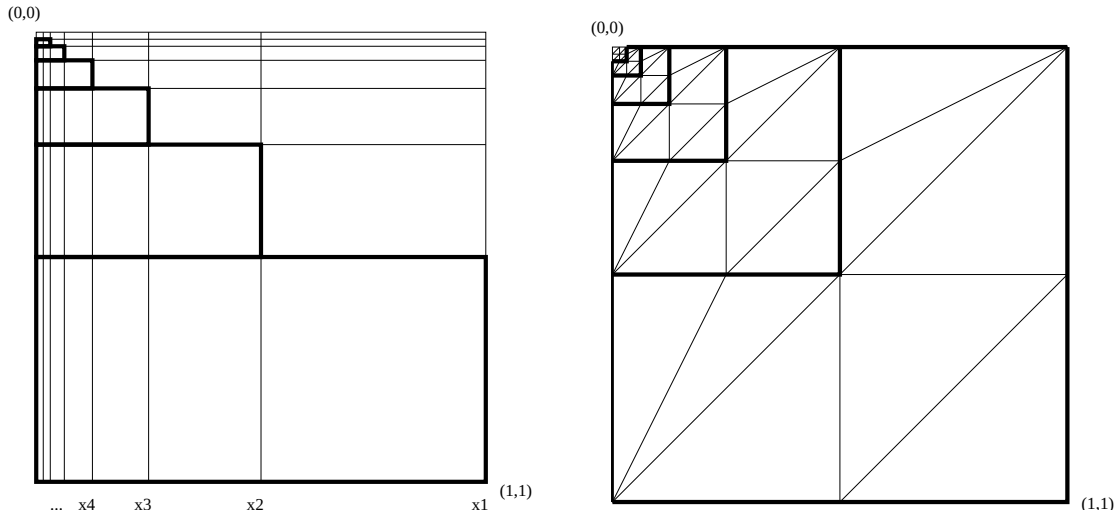


Abbildung 14: Das Gitter links ist geometrisch verfeinert und für eine Clusterung nicht geeignet, das rechte Gitter ist lokal uniform und führt zu einer schwachbesetzten $\mathcal{H}$-Matrix.

Setzt man

$$L := \left\lceil \frac{\sqrt{d}}{2}\eta^{-1} + (\sqrt{d}+4\eta)\bar{h}2^{l-1}H^{-1}\eta^{-1} \right\rceil,$$

so sind die Elemente bis zur $L+1$-ten Schicht die einzigen nicht zulässigen Cluster, also $(2L+1)^d$ nicht zulässige Cluster. Nach Bemerkung 5.4 ist $C_{sp} \leq 2^d(2L+1)^d$. ■

Beispiel 5.7 *(Adaptive Gitter)*
Für regelmäßige Gitter ist es leicht, einen schwachbesetzten $\mathcal{H}_\times$-Baum T zu generieren (siehe [11], [13] für eine explizite Konstruktion). Lokal uniforme Gitter sind (z.B. zur Adaption an den Rand des Gebietes) in vielen Fällen angebracht und sollen durch die Voraussetzungen für die Komplexitätsabschätzungen nicht ausgeschlossen werden. Die beiden Triangulationen in Abbildung 14 sind zum Ursprung $(0,0)$ hin verfeinert. Das rechte Gitter besteht aus nicht entarteten Elementen, während das linke Tensorgitter Elemente enthält, deren Fläche in Relation zum Durchmesser immer kleiner wird.

Der entartete Fall:

Das linke Gitter ist geometrisch graduiert, d.h. die Gitterpunkte $((x_i,x_j))_{i,j=0}^{\sqrt{n}}$ haben die Koordinaten $x_i := 2^{1-i}$ für $i = 1,\ldots,\sqrt{n}-1$ und $x_{\sqrt{n}} := 0$. Die Paneele in der unteren Hälfte $[0,1]\times[0,\frac{1}{2}]$ des Gebietes sind zu keinem anderen Paneel in der unteren Hälfte zulässig ($\operatorname{diam} > \frac{1}{2}$, $\operatorname{dist} < \frac{1}{2}$, $\eta \leq \frac{1}{2}$*) und führen zu $\sqrt{n}\cdot\sqrt{n}$ Matrixeinträgen in vollbesetzten Blöcken. Die Paneele in den darüberliegenden Streifen $[2^{-i-1},2^{-i}]\times[0,2^{-i}]$, $i = 1,\ldots,\sqrt{n}-1$, führen zu $(\sqrt{n}-i)\cdot(\sqrt{n}-i)$ Matrixeinträgen*

in vollbesetzten Blöcken. Insgesamt sind $O(n^{\frac{3}{2}})$ Einträge in vollbesetzten Blöcken zu speichern, so daß der Speicheraufwand nicht mehr logarithmisch-linear ist.

Der gutartige Fall:

In der rechten Triangulation sind die 8 Elemente eines L-Streifens $[0,2^{-i}]\times[0,2^{-i}]\setminus[0,2^{-i-1}]\times[0,2^{-i-1}]$ zu jedem Element in $[0,2^{-i-3}]\times[0,2^{-i-3}]$ zulässig ($\eta=\frac{1}{2}$), so daß hier ein logarithmisch-linearer Speicheraufwand möglich ist.

Lemma 5.8 *(Lokal geometrisch balancierter Fall)*
Sei T ein aus T_I, T_J gebildeter bzgl. der Standard-Zulässigkeitsbedingung Z_η minimal zulässiger $\mathcal{H}_\times$-Baum. Die $\mathcal{H}$-Bäume T_I, T_J seien lokal geometrisch balanciert in dem Sinne, daß es eine Konstante C_{bal} gibt, so daß für alle $i \in \{0,\ldots,p_T\}, \tau \in T_I^{(i)}, \tau' \in T_J^{(i)}$

$$\begin{aligned}
\operatorname{dist}(D(\tau),D(\tau')) \le \frac{1}{2\eta}\operatorname{diam}(D(\tau')) \quad &\Rightarrow \quad \operatorname{diam}(D(\tau)) \le C_{bal}\operatorname{diam}(D(\tau')),\\
\operatorname{dist}(D(\tau),D(\tau')) \le \frac{1}{2\eta}\operatorname{diam}(D(\tau)) \quad &\Rightarrow \quad \operatorname{diam}(D(\tau')) \le C_{bal}\operatorname{diam}(D(\tau))
\end{aligned}$$

gilt (siehe Abbildung 15), und die Cluster seien nicht entartet, d.h. es gibt eine Konstante

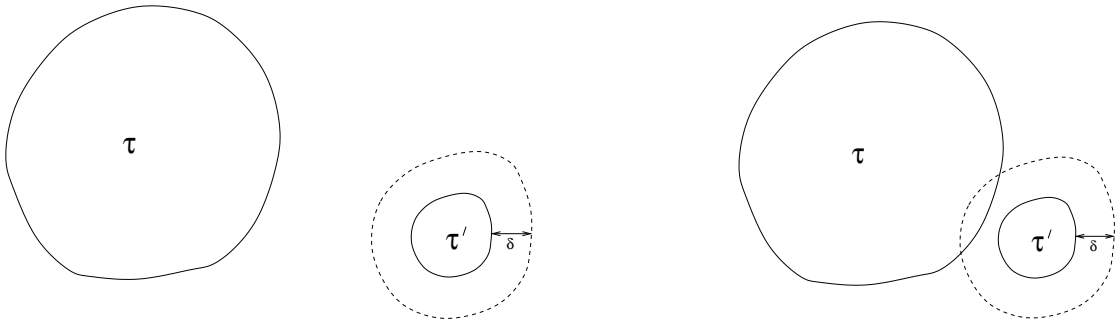


Abbildung 15: In der linken Abbildung ist der Abstand von τ zu τ' im Vergleich zu $\delta = \frac{1}{2\eta}$ mal dem Durchmesser des kleineren Clusters τ' groß genug, also darf τ beliebig groß sein. In der rechten Abbildung ist der Abstand nicht groß genug, also ist τ höchstens C_{bal}-mal so groß wie τ'.

E, so daß für alle $i \in \{0,\ldots,p_T\}, \tau \in T_I^{(i)}, \tau' \in T_J^{(i)}$ gilt

$$\begin{aligned}
E|D(\tau)| &\ge \operatorname{diam}(D(\tau))^d,\\
E|D(\tau')| &\ge \operatorname{diam}(D(\tau'))^d.
\end{aligned}$$

Die Anzahl der Söhne eines Knotens sei durch s beschränkt und von den Mengen τ_i, σ_i zur Charakterisierung der Geometrie (definiert in Abschnitt 3.2.1) haben jeweils höchstens s' einen Durchschnitt mit positivem Maß. Dann ist T mit einer Konstante $C_{sp} = O(\eta^{-d})$ schwachbesetzt.

Beweis: Sei $i \in \{0, \ldots, p_T\}$ und $\tau \in T_I^{(i)}$ (wir beweisen die erste Bedingung von Definition 5.3, die zweite folgt analog). Nach Bemerkung 5.4 genügt es, für

$$M := \left\{ \tau' \in T_J^{(i)} \mid \mathrm{dist}(D(\tau), D(\tau')) \leq \frac{\min(\mathrm{diam}(D(\tau)), \mathrm{diam}(D(\tau')))}{2\eta} \right\}$$

$|M| \leq \frac{1}{s} C_{sp}$ zu zeigen. Die Menge teilen wir auf in

$$\begin{aligned} M_1 &:= \left\{ \tau' \in M \mid \mathrm{diam}(D(\tau)) > \mathrm{diam}(D(\tau')) \right\}, \\ M_2 &:= \left\{ \tau' \in M \mid \mathrm{diam}(D(\tau)) \leq \mathrm{diam}(D(\tau')) \right\}. \end{aligned}$$

Sei x das Čebyšëv-Zentrum von $D(\tau)$ (Mittelpunkt der kleinsten $D(\tau)$ enthaltenden Kugel). Mit $K(x, r)$ bezeichnen wir die abgeschlossene Kugel um x mit Radius r.

$\underline{M_1}$: Sei $\tau' \in M_1$. Der Abstand von $D(\tau)$ zu $D(\tau')$ ist nach Definition von M beschränkt durch $\frac{1}{2\eta}\mathrm{diam}(D(\tau'))$, also folgt

$$D(\tau') \subset K := K\left(x, \underbrace{\frac{1}{2}\mathrm{diam}(D(\tau))}_{\text{Abstand in } \tau \text{ zu } x} + \underbrace{\frac{1}{2\eta}\mathrm{diam}(D(\tau))}_{\text{Abstand } \tau \to \tau'} + \underbrace{\mathrm{diam}(D(\tau))}_{\text{Abstand in } \tau'} \right).$$

Das Volumen der Kugel läßt sich nach oben abschätzen durch

$$|K| \leq ((3 + \frac{1}{\eta})\mathrm{diam}(D(\tau)))^d.$$

Weil T_I, T_J lokal geometrisch balanciert sind, ist $\mathrm{diam}(D(\tau')) \geq C_{bal}^{-1}\mathrm{diam}(D(\tau))$. Es folgt

$$|D(\tau')| \geq E^{-1}\mathrm{diam}(D(\tau'))^d \geq E^{-1}C_{bal}^{-d}\mathrm{diam}(D(\tau))^d,$$

also

$$|D(\tau')| \geq E^{-1}C_{bal}^{-d}(3 + 1/\eta)^{-d}|K|,$$

so daß sich in K nur $EC_{bal}^d(3 + \frac{1}{\eta})^d$ disjunkte Elemente der Größe der Elemente von M_1 befinden können. Nach Voraussetzung ist dann $|M_1| \leq s'EC_{bal}^d(3 + \frac{1}{\eta})^d$.

$\underline{M_2}$: Sei $\tau' \in M_2$. Weil T_I, T_J lokal geometrisch balanciert sind, ist $\mathrm{diam}(D(\tau')) \leq C_{bal}\mathrm{diam}(D(\tau))$. Es folgt

$$D(\tau') \subset K := K\left(x, \underbrace{\frac{1}{2}\mathrm{diam}(D(\tau))}_{\text{Abstand in } \tau \text{ zu } x} + \underbrace{\frac{1}{2\eta}\mathrm{diam}(D(\tau))}_{\text{Abstand } \tau \to \tau'} + \underbrace{C_{bal}\mathrm{diam}(D(\tau))}_{\text{Abstand in } \tau'} \right).$$

Das Volumen der Kugel läßt sich nach oben abschätzen durch

$$|K| \leq ((1 + \frac{1}{\eta} + 2C_{bal})\mathrm{diam}(D(\tau)))^d$$

und das von $D(\tau')$ nach unten durch

$$|D(\tau')| \geq E^{-1}\mathrm{diam}(D(\tau'))^d \geq E^{-1}\mathrm{diam}(D(\tau))^d,$$

so daß sich in K nur $E(1+\frac{1}{\eta}+2C_{bal})^d$ disjunkte Elemente der Größe der Elemente von M_2 befinden können. Nach Voraussetzung ist dann $|M_2| \leq s'E(1+\frac{1}{\eta}+2C_{bal})^d$.

■

Bemerkung 5.9 *(Voraussetzungen von Lemma 5.8)*
Die Voraussetzungen von Lemma 5.8 sind auf Paneel-Ebene (vgl. Beispiel 5.7) notwendig. Für allgemeinere Clusterungen sind die dritte und vierte Voraussetzung des Lemmas allerdings nicht ohne weiteres erfüllbar (z.B. bei Oberflächentriangulationen in der BEM). Für den Beweis genügt es, daß sich in den entsprechenden Kugeln nur $O(\eta^{-d})$ Cluster derselben Stufe befinden können. Eine vertiefte Analyse würde den Rahmen dieser Arbeit sprengen und wird kaum die Vielzahl der in der Praxis auftretenden Triangulationen abdecken. Die Folgerung aus Lemma 5.8 besteht darin, daß es sinnvoll ist, lokal geometrisch balanciert zu clustern, und daß ein nur lokal uniformes Gitter durchaus zu einem schwachbesetzten $\mathcal{H}_\times$-Baum führen kann.

Beispiel 5.10 *(Ein adaptives Gitter)*
Wir betrachten wieder das in Beispiel 5.7 gegebene und in Abbildung 14 rechts dargestellte Gitter. Das Gebiet $\Omega = [0,1]^2$ teilen wir in L-Streifen $L_j := [0,2^{-j}]^2 \setminus [0,2^{-j-1}]^2$, $j = 0,1,\ldots,n-2$, auf und setzen $L_n := [0,2^{-n+1}]^2$. Gegeben sei ein Knoten τ eines beliebigen $\mathcal{H}$-Baumes T der Dreiecke $(\tau_i)_{i=1}^{8n}$ in Ω, $I = \{1,\ldots,8n\}$.
<u>τ ist nicht entartet:</u>
Sei $i \in \tau$ derart, daß $\min\{j \mid \tau_i \subset L_j\} \leq \min\{j \mid \tau_{i'} \subset L_j\}$ für alle $i' \in \tau$ gilt. Dann ist $D(\tau) \subset [0,2^{-j}]^2$, also $\mathrm{diam}(D(\tau)) \leq \sqrt{2}\cdot 2^{-j}$, und $|D(\tau)| \geq |D(\tau_i)| \geq \frac{1}{16}2^{-2j}$. Zusammen ergibt sich

$$\mathrm{diam}(D(\tau))^2 \leq 2\cdot 2^{-2j} \leq 32|D(\tau)|.$$

<u>T ist lokal geometrisch balanciert:</u> Sei $\tau' \in T$ und $i \in \tau'$ derart, daß $\min\{j \mid \tau_i \subset L_j\} \leq \min\{j \mid \tau_{i'} \subset L_j\}$ für alle $i' \in \tau'$ gilt. Dann ist $\mathrm{diam}(\tau') \geq 2^{-j-1}$. Der Einfachheit halber sei vorerst $\eta := 0.5$. Ist $\mathrm{dist}(D(\tau),D(\tau')) \leq \mathrm{diam}(D(\tau'))$, so muß $D(\tau) \subset [0,2^{-j+1}]^2$ gelten, also $\mathrm{diam}(D(\tau)) \leq \sqrt{2}\cdot 2^{-j+1} \leq 4\sqrt{2}\mathrm{diam}(\tau')$. Im allgemeineren Fall $\eta < 0.5$ folgt aus $\mathrm{dist}(D(\tau),D(\tau')) \leq \frac{1}{2\eta}\mathrm{diam}(D(\tau'))$ nur $D(\tau) \subset [0,2^{-j+1-\log_2(2\eta)}]^2$ und somit $\mathrm{diam}(D(\tau)) \leq \sqrt{2}\cdot 2^{-j+1-\log_2(2\eta)} \leq 4\sqrt{2}(2\eta)^{-1}\mathrm{diam}(\tau')$.
Offenbar erfüllt jeder bzgl. Z_η minimal zulässige $\mathcal{H}_\times$-Baum T von $I\times I$, in dem die Zahl der Söhne eines Knotens beschränkt ist, die Voraussetzungen von Lemma 5.8, so daß die Wahl der Clusterung (kardinalitätsbalanciert, geometrisch balanciert) unabhängig von der Schwachbesetztheit des $\mathcal{H}_\times$-Baumes ist. Daher scheint es hier vorteilhaft, kardinalitätsbalanciert zu partitionieren, um die Baumtiefe zu minimieren.

Lemma 5.11 *(Speicheraufwand im schwachbesetzten Fall)*
Für den Speicheraufwand $N_{\mathcal{H},St}(k,T,Z)$ einer Matrix aus $\mathcal{M}_{\mathcal{H},k}(T,Z)$ mit schwachbesetztem aus T_I, T_J gebildetem $\mathcal{H}_\times$-Baum T und konstantem Rang k gilt:

$$N_{\mathcal{H},St}(k,T,Z) \leq |L_T| C_{sp} \max(k, \frac{1}{2} b_{min})(|I| + |J|).$$

Beweis:

$$\begin{aligned}
N_{\mathcal{H},St}(k,T,Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} N_{\mathbf{R}k,St}(|\tau|,|\sigma|) \quad + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} N_{F,St}(|\tau|,|\sigma|) \\
&\leq \sum_{i\in L_T} \sum_{\tau\times\sigma\in\mathcal{L}^+(T,i)} k|\tau| + \sum_{i\in L_T} \sum_{\tau\times\sigma\in\mathcal{L}^+(T,i)} k|\sigma| + \sum_{i\in L_T} \sum_{\tau\times\sigma\in\mathcal{L}^-(T,i)} |\tau||\sigma| \\
&\leq \sum_{i\in L_T} \sum_{\tau\times\sigma\in\mathcal{L}(T,i)} |\tau| \max(k, \frac{1}{2} b_{min}) + \sum_{i\in L_T} \sum_{\tau\times\sigma\in\mathcal{L}(T,i)} |\sigma| \max(k, \frac{1}{2} b_{min}) \\
&\leq |L_T| C_{sp} \max(k, \frac{1}{2} b_{min})(|I| + |J|).
\end{aligned}$$

■

Bemerkung 5.12 *(Was passiert bei unbalancierten $\mathcal{H}$-Bäumen ?)*
Unbalancierte $\mathcal{H}$-Bäume zu einer Indexmenge I zeichnen sich dadurch aus, daß ihre Tiefe nicht proportional zu $\log(|I|)$ ist, daher sind die bisherigen Voraussetzungen und Abschätzungen für diese $\mathcal{H}$-Bäume nicht relevant. Solche $\mathcal{H}$-Bäume entstehen zum Beispiel bei adaptiven Verfahren durch lokale Verfeinerung. Ein zu Anfang kardinalitätsbalancierter $\mathcal{H}$-Baum wird nach mehreren lokalen Verfeinerungen unbalanciert. Um nicht nach jedem einzelnen Verfeinerungsschritt (z.B. eines Elementes der Triangulation) den $\mathcal{H}$-Baum neu aufzustellen, sucht man nach Kriterien für die Unbalanciertheit bzw. Ausgeglichenheit des aktuellen $\mathcal{H}$-Baumes. Prinzipiell müssen hier zwei Anwendungsfelder unterschieden werden:

- *Für die Diskretisierung, Speicherung und Matrix-Vektor-Multiplikation wird in [7] eine Möglichkeit zur Messung der Balanciertheit in einem adaptiven Schema angegeben. Dadurch lassen sich neue Freiheitsgrade mit einem Aufwand von $O(\log(|I|))$ in die Matrix einfügen.*
- *Für die aufwendigeren Operationen, wie etwa die formatierte Multiplikation und Inversion, ist der Aufwand der Clusterung und Diskretisierung im Verhältnis eher gering, daher kann hier stets (unmittelbar vor der Inversion bzw. Multiplikation) der Baum neu erzeugt werden.*

5.2. Auswertung

Lemma 5.13 *(Aufwand der Matrix-Vektor-Multiplikation)*
Der Aufwand $N_{\mathcal{H}\cdot V}(k,T,Z)$ der Matrix-Vektor-Multiplikation in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ läßt sich abschätzen durch

$$N_{\mathcal{H},St}(k,T,Z) \leq \quad N_{\mathcal{H}\cdot V}(k,T,Z) \quad \leq 2N_{\mathcal{H},St}(k,T,Z).$$

Beweis: Sei $\text{root}(T) = I \times J$, $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ und $v \in \mathbb{R}^{|J|}$. Im Folgenden kürzen wir $k := k(\tau,\sigma)$ ab. Dann gilt

$$\begin{aligned}
Mv &= \Big(\sum_{\tau\times\sigma\in\mathcal{L}(T)} (M|_{\tau\times\sigma})|^{I\times J}\Big)v \\
&= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} (M|_{\tau\times\sigma})|^{I\times J}v \quad + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} (M|_{\tau\times\sigma})|^{I\times J}v, \\
N_{\mathcal{H}\cdot V}(k,T,Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} N_{\mathbf{R}k\cdot V}(|\tau|,|\sigma|) \quad + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} N_{F\cdot V}(|\tau|,|\sigma|) \\
&= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} (2k(|\tau|+|\sigma|) - k - |\tau|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} (2|\tau|\cdot|\sigma| - |\tau|) \\
&\geq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} k(|\tau|+|\sigma|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} |\tau|\cdot|\sigma| \\
&= N_{\mathcal{H},St}(k,T,Z), \\
N_{\mathcal{H}\cdot V}(k,T,Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} (2k(|\tau|+|\sigma|) - k - |\tau|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} (2|\tau|\cdot|\sigma| - |\tau|) \\
&\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 2k(|\tau|+|\sigma|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} 2|\tau|\cdot|\sigma| \\
&= 2N_{\mathcal{H},St}(k,T,Z).
\end{aligned}$$

■

Bemerkung 5.14 *(Durchführung der Auswertung)*
Die Komplexitätsbetrachtungen aus Lemma 5.13 setzen voraus, daß die Blöcke einer Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ zu Blättern aus T direkt vorliegen (z.B. in Form einer Liste oder eines Arrays). Der zusätzliche Aufwand durch Suchen in dem Baum T und komplizierte Speicherung der Blöcke wird hier nicht berücksichtigt.

5.3. Bestapproximation, Approximation und hierarchische Approximation

Bemerkung 5.15 *($|T|$ und $|\mathcal{L}(T)|$)*
Die Mächtigkeit eines schwachbesetzten aus T_I, T_J gebildeten $\mathcal{H}_\times$-Baumes T läßt sich abschätzen durch

$$|T| = \sum_{i=0}^{p_T} |T^{(i)}| \leq \sum_{i=0}^{p_T} C_{sp} \min\{|I|,|J|\} \leq C_{sp}(1+p_T)\min\{|I|,|J|\}.$$

Nehmen wir für die Bäume T_I, T_J die Bedingung

$$\forall t \in T_I \cup T_J : \quad |S(t)| \neq 1$$

an, so ist $|T_I| \leq 2|I|$ und $|T_J| \leq 2|J|$. In diesem Fall gilt

$$|\mathcal{L}(T)| = \sum_{\tau\times\sigma\in\mathcal{L}(T)} 1 \leq \min\{\sum_{\tau\in T_I} C_{sp}, \sum_{\sigma\in T_J} C_{sp}\} \leq 2C_{sp}\min\{|I|,|J|\}.$$

Lemma 5.16 *(Aufwand der Bestapproximation)*
Der Aufwand $N_{\mathcal{H}\to\mathcal{H}}(k,k',T,Z)$ der Bestapproximation einer Matrix aus $\mathcal{M}_{\mathcal{H},k}(T,Z)$ durch eine Matrix aus $\mathcal{M}_{\mathcal{H},k'}(T,Z)$, $k' \leq k$, läßt sich abschätzen durch

$$N_{\mathcal{H}\to\mathcal{H}}(k,k',T,Z) \leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 5(|\tau|+|\sigma|)k(\tau\times\sigma)^2 + 23k(\tau\times\sigma)^3.$$

Für konstanten Rang $k \geq 1$ gilt

$$N_{\mathcal{H}\to\mathcal{H}}(k,k',T,Z) \leq 5kN_{\mathcal{H},St}(k,T,Z) + 23k^3|\mathcal{L}^+(T)|.$$

Beweis: Es gilt nach Algorithmus 2.12

$$\begin{aligned} N_{\mathcal{H}\to\mathcal{H}}(k,k',T,Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} N_{\mathbf{R}k,SVD}(|\tau|,|\sigma|) \\ &\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 5k(\tau\times\sigma)^2(|\tau|+|\sigma|) + 23k(\tau\times\sigma)^3 \\ &\leq 5kN_{\mathcal{H},St}(k,T,Z) + 23k^3|\mathcal{L}^+(T)|. \end{aligned}$$

■

Folgerung 5.17 *(Aufwand der Approximation)*
Der Aufwand $N_{\mathcal{H}\to\mathcal{H}}(s\cdot k,k,T,Z)$ der Approximation einer Matrix aus $\mathcal{M}_{\mathcal{H},s\cdot k}(T,Z)$ durch eine Matrix aus $\mathcal{M}_{\mathcal{H},k}(T,Z)$ ($s-1$ mal Kürzen von Rang $2k$ auf Rang k, siehe auch Abschnitt 2.5) läßt sich abschätzen durch

$$N_{\mathcal{H}\to\mathcal{H}}(s\cdot k,k,T,Z) \leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} (s-1)20(|\tau|+|\sigma|)k(\tau\times\sigma)^2 + (s-1)184k(\tau\times\sigma)^3.$$

Für konstanten Rang $k \geq 1$ gilt

$$N_{\mathcal{H}\to\mathcal{H}}(k,k',T,Z) \leq (s-1)20kN_{\mathcal{H},St}(k,T,Z) + (s-1)184k^3|\mathcal{L}^+(T)|.$$

Lemma 5.18 *(Aufwand der hierarchischen Approximation)*
Der Aufwand $N_{\mathcal{H}\to\mathbf{R}k}(k',T,Z)$ der hierarchischen Approximation einer Matrix aus $\mathcal{M}_{\mathcal{H},k'}(T,Z)$ durch eine $\mathbf{R}k$-Matrix für einen schwachbesetzten aus T_I, T_J gebildeten $\mathcal{H}$-Baum T und konstanten Rang k' läßt sich mit $s := \max_{t\in T}|S(t)|$ und $\kappa := \max\{k', sk, b_{min}\}$ abschätzen durch

$$N_{\mathcal{H}\to\mathbf{R}k}(k',T,Z) \leq 5C_{sp}(1+p_T)\kappa^2(|I|+|J|) + 23|T|\kappa^3.$$

Beweis: Für jeden Knoten $\tau\times\sigma \in T\setminus\mathcal{L}(T)$ ist die Summe aus s $\mathbf{R}k$-Matrizen auf Rang k zu kürzen. Das Kürzen erfolgt als Bestapproximation in $\mathbf{R}k(\tau\times\sigma)$ (gekürzte SVD, siehe Algorithmus 2.12) und hat eine Komplexität $N_{\mathbf{R}sk\to\mathbf{R}k}(\tau\times\sigma)$ von

$$N_{\mathbf{R}sk\to\mathbf{R}k}(\tau\times\sigma) \leq 5s^2k^2(|\tau|+|\sigma|) + 23s^3k^3.$$

Für zulässige Blätter $\tau \times \sigma \in \mathcal{L}^+(T)$ (Rang k') wird die Bestapproximation in $\mathbf{R}k(\tau \times \sigma)$ ebenfalls mit der gekürzten SVD bestimmt:

$$N_{\mathbf{R}k' \to \mathbf{R}k}(\tau \times \sigma) \leq 5k'^2(|\tau| + |\sigma|) + 23k'^3.$$

Zusammen mit dem Aufwand $N_{F \to \mathbf{R}k}(\tau \times \sigma) \leq 11(|\tau|^3 + |\sigma|^3)$ der Bestapproximation (Golub-Reinsch SVD, siehe [6]) einer allgemeinen vollbesetzten Matrix in $\mathbf{R}k(\tau \times \sigma)$ ergibt sich

$$\begin{aligned}
N_{\mathcal{H} \to \mathbf{R}k}(k', T, Z) &= \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} N_{\mathbf{R}k' \to \mathbf{R}k}(\tau \times \sigma) + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} N_{F \to \mathbf{R}k}(\tau \times \sigma) \\
&\quad + \sum_{\tau \times \sigma \in T \setminus \mathcal{L}(T)} N_{\mathbf{R}sk \to \mathbf{R}k}(\tau \times \sigma) \\
&\leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 5k'^2(|\tau| + |\sigma|) + 23k'^3 + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 11(|\tau|^3 + |\sigma|^3) \\
&\quad + \sum_{\tau \times \sigma \in T \setminus \mathcal{L}(T)} 5s^2k^2(|\tau| + |\sigma|) + 23s^3k^3 \\
&\leq \sum_{\tau \times \sigma \in T} 5 \max\{k'^2, (sk)^2\}(|\tau| + |\sigma|) \\
&\quad + 23 \max\{k'^3, (sk)^3\}|T \setminus \mathcal{L}^-(T)| + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 11(|\tau|^3 + |\sigma|^3) \\
&\leq \sum_{i=0}^{p_T} \sum_{\tau \times \sigma \in T^{(i)}} 5 \max\{k'^2, (sk)^2\}(|\tau| + |\sigma|) \\
&\quad + 23 \max\{k'^3, (sk)^3\}|T \setminus \mathcal{L}^-(T)| + 22b_{min}^3|\mathcal{L}^-(T)| \\
&\leq \sum_{i=0}^{p_T} \sum_{\tau \in T_I^{(i)}} 5C_{sp} \max\{k'^2, (sk)^2\}|\tau| \\
&\quad + \sum_{i=0}^{p_T} \sum_{\sigma \in T_J^{(i)}} 5C_{sp} \max\{k'^2, (sk)^2\}|\sigma| \\
&\quad + 23|T| \max\{k'^3, (sk)^3, b_{min}^3\} \\
&\leq (p_T + 1)5C_{sp}\kappa^2(|I| + |J|) + 23|T|\kappa^3.
\end{aligned}$$

■

5.4. Addition

Lemma 5.19 *(Aufwand der Matrix-Addition)*
Der Aufwand $N_{\mathcal{H},\oplus}(k, T, Z)$ der formatierten Addition in $\mathcal{M}_{\mathcal{H},k}(T, Z)$ läßt sich abschätzen durch

$$N_{\mathcal{H},\oplus}(k, T, Z) \leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 20(|\tau| + |\sigma|)k(\tau \times \sigma)^2 + 184k(\tau \times \sigma)^3 \quad + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} |\tau| \cdot |\sigma|.$$

Für konstanten Rang $k \geq 1$ gilt

$$N_{\mathcal{H},\oplus}(k,T,Z) \leq 20kN_{\mathcal{H},St}(k,T,Z) + 184k^3|\mathcal{L}^+(T)|.$$

Beweis: Es gilt nach Abschnitt 2.5

$$\begin{aligned}
N_{\mathcal{H},\oplus}(k,T,Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} N_{\mathbf{R}k,\oplus}(|\tau|,|\sigma|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} N_{F,+}(|\tau|,|\sigma|) \\
&\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 20k(\tau\times\sigma)^2(|\tau|+|\sigma|) + 184k(\tau\times\sigma)^3 + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} |\tau|\cdot|\sigma| \\
&\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 20k^2(|\tau|+|\sigma|) + 184k^3 + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} 20k|\tau|\cdot|\sigma| \\
&= 20kN_{\mathcal{H},St}(k,T,Z) + 184k^3|\mathcal{L}^+(T)|.
\end{aligned}$$

■

Bemerkung 5.20 *(Variabler Rang)*
Die Addition ist für konstanten Rang im wesentlichen mit dem Faktor $20k$ proportional zum Speicherbedarf. Wird der Rang in den einzelnen Blöcken unterschiedlich gewählt, so wird bei der Addition der Aufwand in den Blöcken mit hohem Rang noch verstärkt. Ist T schwachbesetzt, die Rangverteilung k konstant und $S(t) \neq 1$ für alle $t \in T$, so erhalten wir aus Lemma 5.11, Bemerkung 5.15 und Lemma 5.19 die Abschätzung

$$N_{\mathcal{H},\oplus}(k,T,Z) = O((k^2|L_T| + k^3)(|I|+|J|)).$$

Ist auf einer Stufe l der Rang $k = k(l)$ und auf allen anderen Stufen k konstant, so erhält man die Abschätzung

$$N_{\mathcal{H},\oplus}(k,T,Z) = O((k^2|L_T| + k^3)(|I|+|J|)) + O(k(l)^3(|I|+|J|)),$$

so daß $k(l)^3 \leq k^2|L_T|$ zu derselben Komplexitätsordnung führt. Ein lineares Anwachsen $k(l) = b\cdot(p_T - l)$ führt zu einem Aufwand von

$$N_{\mathcal{H},\oplus}(k,T,Z) = O(b^2p_T^3(|I|+|J|))$$

und somit nicht *zu einer wesentlichen Ersparnis gegenüber konstantem Rang $k = b\cdot p_T$.*

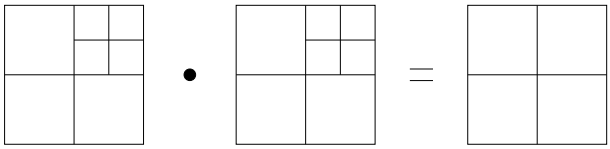
5.5. Multiplikation

Der Aufwand der Addition und Auswertung ließ sich unmittelbar auf den Speicherbedarf zurückführen, da alle Blöcke der Matrix unabhängig voneinander betrachtet werden konnten. In der Multiplikation findet eine Verknüpfung der Blöcke einer Zeile mit den Blöcken einer Spalte statt, so daß hier die Verteilung der Blöcke innerhalb der Matrix eine Rolle spielt. Setzt man wieder die Schwachbesetztheit des $\mathcal{H}$-Baumes und zusätzlich die Fast-Idempotenz voraus, so läßt sich auch für die Multiplikation der Aufwand auf den Speicheraufwand zurückführen.

Definition 5.21 *(Fast idempotent)*
Ein aus T_I gebildeter bzgl. Z zulässiger $\mathcal{H}_\times$-Baum T heißt fast idempotent *zur Konstante C_{id}, falls für jedes Blatt $\tau \times \tau' \in \mathcal{L}(T)$ gilt:*

$$\left| \left\{ \sigma \times \sigma' \in T \cdot T \;\mid\; \sigma \subset \tau \wedge \sigma' \subset \tau' \right\} \right| \leq C_{id}.$$

Bemerkung 5.22 *(Zur Fast-Idempotenz)*
Im Fall $C_{id} = 1$ stimmt die Definition der Fast-Idempotenz nicht *mit der Idempotenz überein. Es gilt T idempotent $\Rightarrow C_{id} = 1$, aber nicht umgekehrt:*



Im folgenden verwenden wir die Notation $\tau^{(i)}$ aus Definition 4.10 für die Vorfahren von τ auf der i-ten Stufe.

Lemma 5.23 *(Fast-Idempotenz bei Standard-Zulässigkeitsbedingung)*
Sei T ein aus T_I gebildeter bzgl. der Standard-Zulässigkeitsbedingung Z_η zulässiger $\mathcal{H}_\times$-Baum, $s := \max_{t \in T} |S(t)|$. Der $\mathcal{H}$-Baum T_I sei geometrisch verfeinert, d.h. es gebe Konstanten C_1, C_2 mit $C_1 C_2 < 1$, $C_2 < 1$ und

$$\forall i \in \{0, \ldots, p_{T_I}\} \forall \tau \in T_I^{(i)} \forall j \in \{0, \ldots, i-1\} : \operatorname{diam}(D(\tau)) \leq C_1 C_2^j \operatorname{diam}(D(\tau^{(i-j)}))$$

und er sei lokal geometrisch balanciert, d.h. für alle $i \in \{0, \ldots, p_T\}, \tau, \tau' \in T_I^{(i)}$ gelte

$$\operatorname{dist}(D(\tau), D(\tau')) \leq \frac{1}{2\eta} \operatorname{diam}(D(\tau)) \quad \Rightarrow \quad \operatorname{diam}(D(\tau)) \leq C_{bal} \cdot \operatorname{diam}(D(\tau')).$$

Dann ist T fast idempotent mit der Konstante $C_{id} = \max\{2b_{min}, s3C_1 C_{bal} s^{-\log_s(C_2)^{-1}}\}$.

Beweis: Sei $\tau \times \tau' \in \mathcal{L}(T, i)$.

1.Fall: $\tau \times \tau'$ ist nicht zulässig, also $|\tau \times \tau'| \leq b_{min}$. Dann besitzt $\tau \times \tau'$ in $T \cdot T$ wegen $C_1 C_2 < 1$ höchstens $2b_{min}$ Nachfahren (b_{min} Blätter und wegen $C_1 C_2 < 1$ gehören mindestens zwei Knoten zu einem Vater), also genügt in diesem Fall $C_{id} \geq 2b_{min}$.

2.Fall: $\tau \times \tau'$ ist zulässig, also

$$\min\{\operatorname{diam}(D(\tau)), \operatorname{diam}(D(\tau'))\} \leq 2\eta \operatorname{dist}(D(\tau), D(\tau')).$$

Die Mächtigkeit der Menge $M := \{\sigma \times \sigma' \in T \cdot T \;\mid\; \sigma \subset \tau \wedge \sigma' \subset \tau'\}$ gilt es abzuschätzen. Sei $\sigma \times \sigma' \in M$ und $\sigma \times \sigma' \neq \tau \times \tau'$. Nach Definition von $T \cdot T$ gibt es ein $\tilde{\sigma} \in T_I$ mit $\sigma \times \tilde{\sigma} \in T, \tilde{\sigma} \times \sigma' \in T$ und weder $\tau \times \tilde{\sigma}^{(i)}$ noch $\tilde{\sigma}^{(i)} \times \tau'$ sind zulässig. Weil T_I lokal geometrisch balanciert ist, folgt daraus

$$\begin{aligned} \operatorname{diam}(D(\tilde{\sigma}^{(i)})) &\leq C_{bal} \operatorname{diam}(D(\tau)), \\ \operatorname{diam}(D(\tilde{\sigma}^{(i)})) &\leq C_{bal} \operatorname{diam}(D(\tau')). \end{aligned}$$

Sei $j \in \mathbb{N}$ mit $\sigma \times \sigma' \in (T \cdot T)^{(i+j)}$. Da T_I geometrisch verfeinert wurde, ist

$$\begin{aligned} \mathrm{diam}(D(\tilde{\sigma})) &\leq C_1 C_2^j \mathrm{diam}(D(\tilde{\sigma}^{(i)})) \\ &\leq C_1 C_2^j C_{bal} \min\{\mathrm{diam}(D(\tau)), \mathrm{diam}(D(\tau'))\} \\ &\leq C_1 C_2^j C_{bal} 2\eta \mathrm{dist}(D(\tau), D(\tau')). \end{aligned} \tag{28}$$

Wir definieren die Umgebungen

$$\begin{aligned} U_\tau &:= \{x \in \mathbb{R}^d \mid \mathrm{dist}(x, D(\tau)) \leq \frac{1}{3}\mathrm{dist}(D(\tau), D(\tau'))\} \quad \text{und} \\ U_{\tau'} &:= \{x \in \mathbb{R}^d \mid \mathrm{dist}(x, D(\tau')) \leq \frac{1}{3}\mathrm{dist}(D(\tau), D(\tau'))\}. \end{aligned}$$

Annahme: $C_1 C_2^j C_{bal} \leq \frac{1}{3}\min\{1, \frac{1}{2\eta}\}$. Dann ist $\mathrm{diam}(D(\tilde{\sigma})) \leq \frac{1}{3}\mathrm{dist}(D(\tau), D(\tau'))$, also $U_\tau \cap D(\tilde{\sigma}) = \emptyset$ oder $U_{\tau'} \cap D(\tilde{\sigma}) = \emptyset$. Sei o.B.d.A. $U_\tau \cap D(\tilde{\sigma}) = \emptyset$. Dann gilt

$$\begin{aligned} \min\{\mathrm{diam}(D(\sigma)), \mathrm{diam}(D(\tilde{\sigma}))\} &\leq \mathrm{diam}(D(\tilde{\sigma})) \\ &\leq \frac{1}{3}\min\{\mathrm{diam}(D(\tau)), \mathrm{diam}(D(\tau'))\} \\ &\overset{(28)}{\leq} \frac{2}{3}\eta \mathrm{dist}(D(\tau), D(\tau')) \\ &\overset{U_\tau \cap D(\tilde{\sigma}) = \emptyset}{\leq} 2\eta \mathrm{dist}(D(\tilde{\sigma}), D(\tau)) \\ &\leq 2\eta \mathrm{dist}(D(\tilde{\sigma}), D(\sigma)). \end{aligned}$$

Es folgt, daß $\sigma \times \tilde{\sigma}$ zulässig ist. (Ende der Annahme)

Wir erhalten also die Aussage

$$C_1 C_2^j C_{bal} \leq \frac{1}{3}\min\{1, \frac{1}{2\eta}\} \Rightarrow \sigma \times \tilde{\sigma} \text{ zulässig}$$

und ihre Kontraposition

$$\sigma \times \tilde{\sigma} \text{ nicht zulässig } \Rightarrow C_1 C_2^j C_{bal} > \frac{1}{3}\min\{1, \frac{1}{2\eta}\}.$$

Da die Vorfahren von $\sigma \times \tilde{\sigma}$ nicht zulässig sind (keine Blätter nach Def. von $T \cdot T$), gilt für $\eta \leq 0.5$:

$$\begin{aligned} C_1 C_2^{j-1} C_{bal} &> \frac{1}{3}, \\ C_2^j &> \frac{C_2}{3 C_1 C_{bal}}, \\ j \log_s(C_2) &> \log_s(C_2) - \log_s(3 C_1 C_{bal}) \\ j &< 1 + \log_s(3 C_1 C_{bal}) \log_s(C_2^{-1})^{-1}. \end{aligned}$$

Die Zahl der Nachfahren $\sigma \times \sigma'$ von $\tau \times \tau'$ auf j Stufen ist beschränkt durch s^j, so daß $C_{id} \geq s(3 C_1 C_{bal})^{\log_s(C_2^{-1})^{-1}}$ für die Fast-Idempotenz genügend ist.

■

Bemerkung 5.24 *(Voraussetzungen für Idempotenz)*
Die zweite Voraussetzung von Lemma 5.23 (lokal geometrisch balanciert) ist wieder die gleiche wie in Lemma 5.8 zur Schwachbesetztheit und im allgemeinen leicht erfüllbar. Die erste Bedingung (geometrisch verfeinert) läßt sich beim BSP-Algorithmus (Beispiel 3.10) dadurch erreichen, daß jeweils nach einigen kardinalitätsbalancierten Teilungen eine geometrisch balancierte Aufteilung vorgenommen wird.

Folgerung 5.25 *Ist T schwachbesetzt (Konstante $C_{sp}(T)$) und fast idempotent (Konstante $C_{id}(T)$) , so ist $T \cdot T$ schwachbesetzt mit der Konstante $C_{sp}(T \cdot T) = C_{id}(T)^2 C_{sp}(T)$.*

Beweis: Sei $i \in \{0, \ldots, p_{T_I}\}$ und $\tau \in T_I^{(i)}$.

Zwischenbehauptung: Zu jedem $\tau' \in T_I$ mit $\tau \times \tau' \in (T \cdot T)^{(i)}$ existiert ein $j \in \{i - C_{id}(T), \ldots, i\}$ mit $\tau^{(j)} \times \tau'^{(j)} \in T^{(j)}$.

Beweis der Zwischenbehauptung: Sei $\tau \times \tau' \in (T \cdot T)^{(i)}$. Ist in der Folge $(\tau^{(j)} \times \tau'^{(j)})_{j=0}^{i}$ ein Blatt von T enthalten, so folgt die Behauptung aus der Fast-Idempotenz von T. Ist in der Folge kein Blatt enthalten, so ist $\tau \times \tau' \in T$.

Rest des Beweises: Es gilt
$|\{\tau \times \tau' \in (T \cdot T)^{(i)}\}| \leq \sum_{j=i}^{i-C_{id}(T)} C_{id}(T) |\{\tau^{(j)} \times \tau' \in T^{(j)}\}| \leq C_{id}(T)^2 C_{sp}(T)$. ■

Die Aufwandsabschätzung für die Multiplikation zweier hierarchischer Matrizen teilt sich in drei Abschnitte auf. Zuerst untersuchen wir die exakte Multiplikation zweier Matrizen zum selben Baum $T_{I \times I}$ der Indexmenge $I \times I$. Das Ergebnis liegt nicht mehr in derselben Klasse von $\mathcal{H}$-Matrizen, kann aber in derselben Struktur mit höherem Rang $p_{T_{I \times I}} k$ dargestellt werden. Danach gilt es, das Ergebnis wieder zur ursprünglichen Klasse mit Rang k zu konvertieren, so daß man die bzgl. der Frobeniusnorm bestmögliche Approximation in der Klasse erhält. Zuletzt schätzen wir den Aufwand zur Berechnung bei approximativer Konvertierung ab, wobei hier das Aufstellen der exakten Produktmatrix vermieden werden kann (siehe Bemerkung 4.12).

Satz 5.26 *(Aufwand der exakten Matrix-Matrix-Multiplikation)*
Sei T_I ein $\mathcal{H}$-Baum der Indexmenge I und T ein aus T_I gebildeter bzgl. Z zulässiger $\mathcal{H}_\times$-Baum. T sei schwachbesetzt und fast idempotent. Die Berechnung des Produktes zweier Matrizen $M, M' \in \mathcal{M}_{\mathcal{H},k}(T, Z)$ zu konstantem Rang k in der Darstellung $M \cdot M' \in \mathcal{M}_{\mathcal{H},\tilde{k}}(T, Z)$ hat einen Aufwand von

$$\begin{aligned} N_{\mathcal{H},\odot}^{exakt}(k, T, Z) &\leq 4(p_T + 1) C_{sp} \max(k, b_{min}) N_{\mathcal{H},St}(k, T, Z) \\ &\quad + |\mathcal{L}^-(T)| 2(p_T + 1)(C_{id} + 1) C_{sp} k b_{min}^2 \\ &= O(p_T k) N_{\mathcal{H},St}(k, T, Z) \end{aligned}$$

und der zur Darstellung nötige Rang ist beschränkt durch

$$\tilde{k} \leq C_{id} \max((p_T + 1) C_{sp} k, b_{min}).$$

Beweis: Nach Lemma 4.11, Formel (15), gilt für jedes Blatt $\tau \times \tau' \in \mathcal{L}(T \cdot T, i)$:

$$(M \cdot M')|_{\tau\times\tau'} = \sum_{j=0}^{i} \sum_{\tilde{\tau}\in U_j} \left(M|_{\tau^{(j)}\times\tilde{\tau}} \cdot M'|_{\tilde{\tau}\times\tau'^{(j)}} \right)|_{\tau\times\tau'}$$

Der Aufwand $N^{exakt}_{\mathcal{H},\odot}(k,T,Z)$ wird getrennt abgeschätzt für die Berechnung der zulässigen Blätter (Aufwand N^+) in der Darstellung von $\mathcal{M}_{\mathcal{H},k\cdot k}(T\cdot T, Z\cdot Z)$, der nicht zulässigen Blätter (Aufwand N^-) in der Darstellung von $\mathcal{M}_{\mathcal{H},k\cdot k}(T\cdot T, Z\cdot Z)$ und für den Darstellungswechsel von $\mathcal{M}_{\mathcal{H},k\cdot k}(T\cdot T, Z\cdot Z)$ zu $\mathcal{M}_{\mathcal{H},\tilde{k}}(T,Z)$.

1. Zulässige Blätter $\tau \times \tau'$:

Nach Definition der U_j und $Z \cdot Z$ ist jeweils $\tau^{(j)} \times \tilde{\tau}$ oder $\tilde{\tau} \times \tau'^{(j)}$ in (15) ein zulässiges Blatt von T bzw. T'. Nach Abschnitt 2.3 sind für jeden Summanden höchstens k Matrix-Vektor-Multiplikationen mit $M|_{\tau^{(j)}\times\tilde{\tau}}$ oder $(M'|_{\tilde{\tau}\times\tau'^{(j)}})^T$ durchzuführen. Summiert über die Partition $\dot{\cup}_{j=0}^{i} U_j$ von J sind höchstens k Matrix-Vektor-Multiplikationen mit $M|_{\tau\times J}$ und $(M'|_{J\times\tau'})^T$ nötig, deren Aufwand nach Lemma 5.13 durch $2k$-mal den Speicheraufwand $N_{M|_{\tau\times J},St}$ von $M|_{\tau\times J}$ bzw. $N_{M'|_{J\times\tau'},St}$ von $(M'|_{J\times\tau'})^T$ beschränkt ist. Es folgt

$$\begin{aligned} N^+ &\leq \sum_{\tau\times\tau'\in\mathcal{L}^+(T\cdot T)} 2kN_{M|_{\tau\times J},St} \quad + \sum_{\tau\times\tau'\in\mathcal{L}^+(T\cdot T)} 2kN_{M'|_{J\times\tau'},St} \\ &= \sum_{i=0}^{p_T} \sum_{\tau\times\tau'\in\mathcal{L}^+(T\cdot T,i)} 2kN_{M|_{\tau\times J},St} \quad + \sum_{i=0}^{p_T} \sum_{\tau\times\tau'\in\mathcal{L}^+(T\cdot T,i)} 2kN_{M'|_{J\times\tau'},St}. \end{aligned}$$

2. Nicht zulässige Blätter $\tau \times \tau'$:

Nach Definition der U_j ist jeweils $\tau^{(j)} \times \tilde{\tau}$ oder $\tilde{\tau} \times \tau'^{(j)}$ in (15) ein Blatt von T bzw. T'. Ist das Blatt bzgl. Z zulässig, so sind k Matrix-Vektor-Multiplikationen durchzuführen und das Ergebnis aufzuaddieren. Ist das Blatt nicht zulässig, so sind höchstens b_{min} Matrix-Vektor-Multiplikationen durchzuführen und das Ergebnis aufzuaddieren. Da bei wenigstens einem Summanden beide Faktoren nicht zulässig sind (siehe Definition von $Z\cdot Z$), folgt $|\tau\times\tau'| \leq b^2_{min}$. Der Aufwand für das Aufaddieren ist pro Blatt also beschränkt durch $(p_T+1)|U_j|b^2_{min} \leq (p_T+1)C_{sp}b^2_{min}$. Zusammen erhält man mit Lemma 5.13

$$\begin{aligned} N^- &\leq \sum_{\tau\times\tau'\in\mathcal{L}^-(T\cdot T)} 2\max(k,b_{min})N_{M|_{\tau\times J},St} \quad + \\ &\quad \sum_{\tau\times\tau'\in\mathcal{L}^-(T\cdot T)} 2\max(k,b_{min})N_{M'|_{J\times\tau'},St} + \sum_{\tau\times\tau'\in\mathcal{L}^-(T\cdot T)} (p_T+1)C_{sp}b^2_{min} \\ &\leq \sum_{i=0}^{p_T} \sum_{\tau\times\tau'\in\mathcal{L}^-(T\cdot T,i)} 2\max(k,b_{min})N_{M|_{\tau\times J},St} \quad + \\ &\quad \sum_{i=0}^{p_T} \sum_{\tau\times\tau'\in\mathcal{L}^-(T\cdot T,i)} 2\max(k,b_{min})N_{M'|_{J\times\tau'},St} + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b^2_{min}. \end{aligned}$$

Der Aufwand zur Berechnung von $M \cdot M'$ in $\mathcal{M}_{\mathcal{H},k\cdot k}(T \cdot T, Z \cdot Z)$ ergibt sich aus der Summe von N^+ und N^-:

$$\begin{aligned}
N^+ + N^- &\leq \sum_{i=0}^{p_T} \sum_{\tau\times\tau' \in \mathcal{L}(T\cdot T,i)} 2\max(k, b_{min}) N_{M|_{\tau\times J},St} \quad + \\
&\quad \sum_{i=0}^{p_T} \sum_{\tau\times\tau' \in \mathcal{L}(T\cdot T,i)} 2\max(k, b_{min}) N_{M'|_{J\times\tau'},St} + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b_{min}^2 \\
&\leq \sum_{i=0}^{p_T} \sum_{\tau\in T^{(i)}} 2C_{sp}\max(k, b_{min}) N_{M|_{\tau\times J},St} \quad + \\
&\quad \sum_{i=0}^{p_T} \sum_{\tau'\in T^{(i)}} 2C_{sp}\max(k, b_{min}) N_{M'|_{J\times\tau'},St} \quad + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b_{min}^2 \\
&\leq (p_T+1)2C_{sp}\max(k, b_{min}) N_{\mathcal{H},St}(k,T,Z) \quad + \\
&\quad (p_T+1)2C_{sp}\max(k, b_{min}) N_{\mathcal{H},St}(k,T,Z) + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b_{min}^2 \\
&\leq 4(p_T+1)C_{sp}\max(k, b_{min}) N_{\mathcal{H},St}(k,T,Z) + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b_{min}^2.
\end{aligned}$$

3. Darstellungswechsel: Der Darstellungswechsel von $\mathcal{M}_{\mathcal{H},k\cdot k}(T \cdot T, Z \cdot Z)$ nach $\mathcal{M}_{\mathcal{H},\tilde{k}}(T, Z)$ erfolgt in den nicht zulässigen Blättern von T durch Kopieren (vollbesetzt $\rightarrow$ vollbesetzt) bzw. Ausmultiplizieren ($\mathbf{R}k\rightarrow$ vollbesetzt) und in den zulässigen Blättern von T durch Kopieren ($\sum\mathbf{R}k\rightarrow \mathbf{R}k$) bzw. kanonische $\mathbf{R}k$-Darstellung (siehe 2.7). Der Rang von $M \cdot M'$ in den zulässigen Blättern von $\mathcal{M}_{\mathcal{H},k\cdot k}(T \cdot T, Z \cdot Z)$ ist, da höchstens $(p_T+1)C_{sp}$ Summanden auftreten, durch $(p_T+1)C_{sp}k$ beschränkt.
Der Aufwand zum Ausmultiplizieren ($\mathbf{R}k\rightarrow$ vollbesetzt) wird durch

$$\begin{aligned}
N^{ausmul} &\leq \sum_{\tau\times\tau' \in \mathcal{L}^-(T)} 2(p_T+1)C_{sp}kb_{min}^2 \\
&\leq |\mathcal{L}^-(T)|2(p_T+1)C_{sp}kb_{min}^2
\end{aligned}$$

abgeschätzt und der Gesamtaufwand erfüllt schließlich

$$\begin{aligned}
N^{exakt}_{\mathcal{H},\odot}(k,T,Z) &= N^+ + N^- + N^{ausmul} \\
&\leq 4(p_T+1)C_{sp}\max(k, b_{min}) N_{\mathcal{H},St}(k,T,Z) \\
&\quad + |\mathcal{L}^-(T\cdot T)|(p_T+1)C_{sp}b_{min}^2 \\
&\quad + |\mathcal{L}^-(T)|2(p_T+1)C_{sp}kb_{min}^2 \\
&\leq 4(p_T+1)C_{sp}\max(k, b_{min}) N_{\mathcal{H},St}(k,T,Z) \\
&\quad + |\mathcal{L}^-(T)|2(p_T+1)(C_{id}+1)C_{sp}kb_{min}^2.
\end{aligned}$$

Bestimmung des Ranges $\tilde{k}$: Wir wollen nun den Rang $\tilde{k}$, der zur Darstellung von $M \cdot M'$ in $\mathcal{M}_{\mathcal{H},\tilde{k}}(T, Z)$ benötigt wird, bestimmen. Aufgrund der Fast-Idempotenz von T wissen wir bereits, daß ein Blatt von T aus höchstens C_{id} Blättern von $T \cdot T$ besteht oder in einem Blatt von $T \cdot T$ enthalten ist. Nicht zulässige Blätter $\tau \times \tau' \in T \cdot T$ erfüllen (s.o.)

$|\tau \times \tau'| \leq b_{min}^2$ und somit $\text{rang}((M \cdot M')|_{\tau \times \tau'}) \leq b_{min}$. Damit läßt sich der zur Darstellung nötige Rang abschätzen durch

$$\tilde{k} \leq C_{id} \max((p_T + 1) C_{sp} k, b_{min}).$$

■

Folgerung 5.27 *(Bestapproximation und Approximation)*
Sei T_I ein $\mathcal{H}$-Baum der Indexmenge I und T ein aus T_I gebildeter bzgl. Z zulässiger $\mathcal{H}_\times$-Baum. T sei schwachbesetzt und fast idempotent. Dann läßt sich der Aufwand N^{best}, N^{apx} der formatierten Multiplikation (Bestapproximation/Approximation) in $\mathcal{M}_{\mathcal{H},k}(T, Z)$ für konstanten Rang k abschätzen durch

$$\begin{aligned} N^{best}_{\mathcal{H},\odot}(k, T, Z) &= O(p_T^2 k) N_{\mathcal{H},St}(k, T, Z) + O(p_T^3 k^3) |\mathcal{L}^+(T)|, \\ N^{apx}_{\mathcal{H},\odot}(k, T, Z) &= O(p_T k) N_{\mathcal{H},St}(k, T, Z) + O(p_T k^3) |\mathcal{L}^+(T)| \end{aligned}$$

Beweis: Satz 5.26, Lemma 5.16 und Folgerung 5.17. ■

Bemerkung 5.28 *(Multiplikation bei verschiedenen Bäumen)*
Seien $T_I, T_J, T_{I'}$ $\mathcal{H}$-Bäume der Indexmengen I, J, I', T ein aus T_I, T_J gebildeter bzgl. Z zulässiger, T' ein aus $T_J, T_{I'}$ gebildeter bzgl. Z' zulässiger und $\tilde{T}$ ein aus $T_I, T_{I'}$ gebildeter bzgl. $\tilde{Z}$ zulässiger $\mathcal{H}_\times$-Baum. T und T' erfüllen die Schwachbesetztheitsbedingung

$$\begin{aligned} \forall i \in \{0, \ldots, p_{T_I}\} \quad \forall \tau \in T_I^{(i)} : \quad |\{\tau \times \tau' \in T^{(i)}\}| \leq C_{sp} \\ \forall i \in \{0, \ldots, p_{T_{I'}}\} \quad \forall \tau' \in T_{I'}^{(i)} : \quad |\{\tau \times \tau' \in T'^{(i)}\}| \leq C_{sp} \end{aligned}$$

und die Kompatiblitätsbedingung (anlog zur Fast-Idempotenz)

$$\forall \tau \times \tau' \in \mathcal{L}(\tilde{T}) : \quad \left| \left\{ \sigma \times \sigma' \in T \cdot T' \;\middle|\; \sigma \subset \tau \wedge \sigma' \subset \tau' \right\} \right| \leq C_{id}.$$

Dann ist der Aufwand für die exakte Multiplikation

$$\cdot : \mathcal{M}_{\mathcal{H},k}(T, Z) \times \mathcal{M}_{\mathcal{H},k}(T', Z') \to \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$$

beschränkt durch

$$\begin{aligned} N^{exakt}_{\mathcal{H},\cdot}(k, T, T', \tilde{T}, Z, Z', \tilde{Z}) &\leq 2(p_T + 1) C_{sp} \max(k, b_{min}) (N_{\mathcal{H},St}(k, T, Z) \\ &\quad + N_{\mathcal{H},St}(k, T', Z')) \\ &\quad + |\mathcal{L}^-(\tilde{T})| 2(p_T + 1)(C_{id} + 1) C_{sp} k b_{min}^2 \\ &= O(p_T k)(N_{\mathcal{H},St}(k, T, Z) + N_{\mathcal{H},St}(k, T', Z')) \end{aligned}$$

und der zur Darstellung nötige Rang ist beschränkt durch

$$\tilde{k} \leq C_{id} \max((p_T + 1) C_{sp} k, b_{min}).$$

Satz 5.29 *(Aufwand der formatierten Inversion)*
Sei T_I ein $\mathcal{H}$-Baum der Indexmenge I und T ein aus T_I gebildeter bzgl. Z zulässiger $\mathcal{H}_\times$-Baum. Die Berechnung der $\mathcal{H}$-Inversen (siehe Abschnitt 4.5.1 bzw. Algorithmus 4.18) einer Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ hat einen Aufwand von

$$N_{\mathcal{H}^{\ominus^{-1}}}(k,T,Z) \leq N^{apx}_{\mathcal{H},\odot}(k,T,Z).$$

Beweis: Zur Inversion einer Untermatrix zu einem Block $\tau \times \tau$, $S(\tau) = \{\sigma_1, \ldots, \sigma_s\}$, der zu invertierenden Matrix werden formatierte Multiplikationen und Additionen für die Untermatrizen zu $\sigma_i \times \sigma_j$ durchgeführt. Wir werden zeigen, daß diese Operationen ebenfalls bei der formatierten Multiplikation durchgeführt werden. Um den Beweis einfach zu gestalten, wird hier nur der Fall eines Binärbaumes T_I behandelt. Ferner sei $|\tau \times \tau| = 1$ für alle $\tau \times \tau \in \mathcal{L}(T)$. Die Behauptung wird per Induktion über die Baumtiefe p_T bewiesen.
Induktionsanfang: $p_T = 0$. Nach Voraussetzung ist eine 1×1 Matrix zu invertieren, und wir nehmen an, daß der Aufwand derselbe wie zur Multiplikation zweier Zahlen ist.
Induktionsschritt: Zur Multiplikation zweier Matrizen der Struktur

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

wird

$$A \odot A = \begin{bmatrix} A_{11} \odot A_{11} \oplus A_{12} \odot A_{21} & A_{12} \odot A_{22} \oplus A_{11} \odot A_{12} \\ A_{21} \odot A_{11} \oplus A_{22} \odot A_{21} & A_{22} \odot A_{22} \oplus A_{21} \odot A_{12} \end{bmatrix}$$

berechnet, also alle Kombinationen $A_{i\nu} \odot A_{\nu j}$ für $i,j,\nu = 1,2$. Für die Inversion werden (vgl. Algorithmus 4.18) die Matrizen

$$\begin{aligned} L_{11} &:= R_{11}^{\ominus^{-1}} \\ H_{12} &:= L_{11} \odot R_{12} \\ R_{12} &:= H_{12} \\ H_{22} &:= -R_{21} \odot R_{12} \\ R_{22} &:= R_{22} \oplus H_{22} \\ L_{21} &:= R_{21} \odot L_{11} \end{aligned}$$

$$\begin{aligned} L_{22} &:= R_{22}^{\ominus^{-1}} \\ L_{21} &:= -L_{22} \odot L_{21} \\ H_{11} &:= -R_{12} \odot L_{21} \\ L_{11} &:= L_{11} \oplus H_{11} \\ L_{12} &:= -R_{12} \odot L_{22} \end{aligned}$$

berechnet. Auch hier treten die Kombinationen $A_{i\nu} \odot A_{\nu j}$ für $i,j,\nu = 1,2$ auf, wobei anstelle der Diagonal-Multiplikationen $i = j = \nu$ die Inversion für die Untermatrizen durchgeführt wird, die nach Induktionsvoraussetzung höchstens so aufwendig wie die Multiplikation ist.

Im allgemeinen Fall ist die Inversion in den Blättern nicht notwendigerweise proportional zur Multiplikation (siehe Strassen-Multiplikation in [6]), allerdings wird in den Abschätzungen für die Multiplikation ein kubischer Aufwand angenommen, so daß die Abschätzungen für die Multiplikation und die Inversion dieselben sind. Ist der Baum T_I kein Binärbaum, so überzeugt man sich schnell, daß in Algorithmus 4.18 jeweils genau einmal die Kombinationen $A_{i\nu} \odot A_{\nu j}$ für $i, j, \nu = 1, \ldots, s$ auftreten, wobei anstelle der Diagonal-Multiplikationen $i = j = \nu$ die Inversionen für die Untermatrizen durchgeführt werden. ■

5.6. Spektral- und Frobeniusnorm

Lemma 5.30 *(Aufwand zur Berechnung der Frobeniusnorm)*
Der Aufwand $N_{\mathcal{H},\|\cdot\|_F}(k, T, Z)$ zur Berechnung der Frobeniusnorm einer Matrix aus $\mathcal{M}_{\mathcal{H},k}(T, Z)$ läßt sich abschätzen durch

$$N_{\mathcal{H},\|\cdot\|_F}(k, T, Z) \leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 4(|\tau| + |\sigma|)k(\tau \times \sigma)^2 + 27k(\tau \times \sigma)^3 \quad + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} 2|\tau| \cdot |\sigma|.$$

Für konstanten Rang $k \geq 1$ gilt

$$N_{\mathcal{H},\|\cdot\|_F}(k, T, Z) \leq 4kN_{\mathcal{H},St}(k, T, Z) + 27k^3|\mathcal{L}^+(T)|.$$

Beweis: Es gilt nach Abschnitt 2.6

$$\begin{aligned}
N_{\mathcal{H},\|\cdot\|_F}(k, T, Z) &= \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} N_{\mathbf{R}k,\|\cdot\|_F}(|\tau|, |\sigma|) + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} N_{F,\|\cdot\|_F}(|\tau|, |\sigma|) \\
&\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 4k(\tau \times \sigma)^2(|\tau| + |\sigma|) + 23k(\tau \times \sigma)^3 \\
&\quad + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} 2|\tau| \cdot |\sigma| \\
&\leq \sum_{\tau\times\sigma\in\mathcal{L}^+(T)} 4k^2(|\tau| + |\sigma|) + 23k(\tau \times \sigma)^3 + \sum_{\tau\times\sigma\in\mathcal{L}^-(T)} 4k|\tau| \cdot |\sigma| \\
&= 4kN_{\mathcal{H},St}(k, T, Z) + 23k^3|\mathcal{L}^+(T)|.
\end{aligned}$$

■

Folgerung 5.31 *(Aufwand zur Approximation der Spektralnorm)*
Der Aufwand zur Approximation der Spektralnorm einer Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T_{I\times J}, Z)$ bis auf einen relativen Fehler von ε mit der Vektoriteration aus 4.31 hat einen Aufwand von

$$N_{\mathcal{H},\|\cdot\|_{2,apx}}(k, T, Z, \varepsilon) = O(\varepsilon^{-1}(log(\min\{|I|, |J|\}) - log(\varepsilon))N_{\mathcal{H},St}(k, T_{I\times J}, Z).$$

6. Adaptive Arithmetik

Die Betrachtungen aus den vorangegangenen Abschnitten haben sich auf den Fall beschränkt, in dem $\mathcal{H}$-Bäume $T_I, T_J, T_{I\times J}$ und Rangverteilungen k vorgegeben sind und innerhalb von $\mathcal{H}$-Matrix-Strukturen arithmetische Operationen durchzuführen sind. Die Erzeugung des Baumes $T_{I\times J}$ läßt sich kanonisch durchführen, wenn die Bäume T_I, T_J und eine Zulässigkeitsbedingung gegeben sind. Die Bäume T_I, T_J sind die mit dem BSP-Algorithmus erzeugten und liefern im allgemeinen gute Kandidaten zur Konstruktion der Matrix-Blöcke. Die richtige Wahl der Zulässigkeitsbedingung setzt erheblich mehr Wissen über das zugrunde liegende Problem voraus. Im Falle der Diskretisierung von Integralgleichungen leitet sich diese aus der explizit gegebenen Kernfunktion ab, im Falle der Lösung partieller Differentialgleichungen hängt sie von der (unbekannten) Greenschen Funktion ab. Das wesentliche Verhalten der Greenschen Funktion wird durch die Singularitätenfunktion (Fundamentallösung, Kernfunktion bei der Integralgleichungsmethode) beschrieben, welche als bekannt angenommen wird. Zur Steuerung der Genauigkeit der approximativen $\mathcal{H}$-Arithmetik bleibt schließlich die Rangverteilung k. Sie soll im Folgenden so bestimmt werden, daß das Ergebnis der durchzuführenden (approximativen) arithmetischen Operation eine gewisse vorgegebene Genauigkeit erreicht. Bei der Approximation einer Matrix aus der Diskretisierung einer Integralgleichung werden in [2] Methoden zur adaptiven (kanonischen) Rangwahl vorgestellt. Wir wollen nun untersuchen, wie die Rangwahl durchgeführt werden kann, so daß auch bei den Operationen $\oplus, \odot, \ominus^{-1}$ die Rangwahl automatisiert ist.

6.1. Grundlagen

Definition 6.1 *(Stufenweise Schwachbesetztheit)*
Sei T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum. Dann definieren wir die stufenweisen Schwachbesetztheitskonstanten als

$$C_{sp}^{(l)} := \max\left(\max_{\tau\in T_I^{(l)}} |\{\tau' \in T_J \mid \tau\times\tau' \in \mathcal{L}(T)\}|, \max_{\tau'\in T_J^{(l)}} |\{\tau \in T_I \mid \tau\times\tau' \in \mathcal{L}(T)\}|\right).$$

und setzen $\bar{C}_{sp} := \sum_{l\in L_T} C_{sp}^{(l)}$.

Satz 6.2 *(Schranken für die Spektralnorm)*
Sei T ein aus T_I, T_J gebildeter schwachbesetzter $\mathcal{H}$-Baum und $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$. Dann gilt

$$\max_{t\in T} \|M|_t\|_2 \leq \quad \|M\|_2 \quad \leq C_{sp} \sum_{l\in L_T} \max_{t\in\mathcal{L}(T,l)} \|M|_t\|_2.$$

Mit den Schwachbesetztheitskonstanten aus Definition 6.1 gilt die schärfere Abschätzung

$$\|M\|_2 \quad \leq \sum_{l\in L_T} C_{sp}^{(l)} \max_{t\in\mathcal{L}(T,l)} \|M|_t\|_2.$$

Beweis: Es gilt für eine Matrix $A \in \mathbb{R}^{n,m}$ mit höchstens C Einträgen in jeder Spalte und Zeile und $\bar{a} := \max_{i,j} |a_{ij}|$

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \leq \sqrt{C\bar{a}C\bar{a}} \leq C\bar{a}.$$

Für Blockmatrizen $A = (A_{ij})_{i,j=1}^{n,m}$, $A_{ij} \in \mathbb{R}^{n_i,m_j}$, mit höchstens C von Null verschiedenen Blöcken in jeder Block-Zeile und Block-Spalte erhalten wir das entsprechende Resultat

$$\|A\|_2 \leq C \max_{i,j} \|A_{ij}\|_2, \tag{29}$$

das im nachfolgenden Hilfssatz bewiesen wird. Wir definieren für $l \in L_T$

$$M^{(l)}|_t := \begin{cases} M|_t & \text{falls } t \in \mathcal{L}(T,l) \\ 0 & \text{sonst} \end{cases}, \qquad t \in \mathcal{L}(T).$$

Dann gilt

$$\begin{aligned} \|M\|_2 &= \|\sum_{l \in L_T} M^{(l)}\|_2 \leq \sum_{l \in L_T} \|M^{(l)}\|_2 \\ &\overset{(29)}{\leq} \sum_{l \in L_T} C_{sp}^{(l)} \max_{t \in \mathcal{L}(T,l)} \|M|_t\|_2 \\ &\leq C_{sp} \sum_{l \in L_T} \max_{t \in \mathcal{L}(T,l)} \|M|_t\|_2. \end{aligned}$$

■

Beispiel 6.3 *(Schärfe der Schranke)*
Definiere für alle $n, m \in \mathbb{N}$ die $n \times m$-Matrix

$$R_{1,n,m} := ab^T, \quad a_i := \frac{1}{\sqrt{n}}, b_j := \frac{1}{\sqrt{m}}.$$

Dann ist $\|R_{1,n,m}\|_2 = \|a\|_2 \|b\|_2 = 1$. Sei $n = 2^p$, $p \in \mathbb{N}$ und $I := \{1, \ldots, n\}$. Der Baum T_I sei der in Beispiel 3.12 definierte $\mathcal{H}$-Baum, die Zulässigkeitsbedingung $Z : T_I \times T_I \to \{$„zulässig“, „nicht zulässig“$\}$ sei definiert durch

$$Z(r,s) := \begin{cases} \text{„zulässig“} & \text{falls } t \cap s = \emptyset \\ \text{„nicht zulässig“} & \text{sonst} \end{cases}$$

und der $\mathcal{H}_\times$-Baum $T_{I \times I}$ sei der aus T_I, T_I gebildete minimal zulässige $\mathcal{H}_\times$-Baum (siehe Beispiel 3.26). Dann ist $T_{I \times I}$ genau der in [11, Abschnitt 2.2.2] definierte Baum T und besitzt die Schwachbesetztheitskonstante $C_{sp} = 1$.
Sei $\varepsilon \in \mathbb{R}_{>0}$ vorgegeben. Wir definieren die Matrix $M \in \mathcal{M}_{\mathcal{H},1}(T,Z)$ durch

$$M|_{r \times s} := \varepsilon R_{1,|r|,|s|}$$

für alle Blätter $r \times s \in \mathcal{L}(T)$. Dann gilt nach Satz 6.2 $\|M\|_2 \leq p\varepsilon$. Für den Vektor $v \equiv 1$ rechnet man leicht $Mv = p\varepsilon v$ nach, so daß $\|M\|_2 = p\varepsilon$ gilt. In diesem Fall ist die Schranke für die Spektralnorm aus Satz 6.2 also die exakte Spektralnorm.

Hilfssatz 6.4 *(Spektralnorm von schwachbesetzten Blockmatrizen)*
Seien I, J Mengen, $I = \dot\cup_{i=1,\ldots,n} I_i$, $J = \dot\cup_{j=1,\ldots,m} J_j$ und $A \in \mathbb{R}^{I\times J}$ mit der Eigenschaft, daß

$$\begin{aligned}\forall i = 1,\ldots,n: &\quad |\{j \in \{1,\ldots,m\} \mid A_{I_i\times J_j} \neq 0\}| \leq C,\\ \forall j = 1,\ldots,m: &\quad |\{i \in \{1,\ldots,n\} \mid A_{I_i\times J_j} \neq 0\}| \leq C.\end{aligned}$$

Dann läßt sich die Spektralnorm von A abschätzen durch

$$\|A\|_2 \leq C \max_{\substack{i=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_i\times J_j}\|_2.$$

Beweis: Wir definieren die Block-∞-Norm für $x \in \mathbb{R}^J$ und $y \in \mathbb{R}^I$ durch

$$\begin{aligned}\|x\|_{\infty,b,m} &:= \max_{j=1,\ldots,m} \|x|_{J_j}\|_2,\\ \|y\|_{\infty,b,n} &:= \max_{i=1,\ldots,n} \|y|_{I_i}\|_2\end{aligned}$$

und die entsprechende Matrixnorm

$$\|A\|_{\infty,b} := \max_{\|x\|_{\infty,b,m}=1} \|Ax\|_{\infty,b,n}.$$

Sei $z \in \mathbb{R}^J$ mit $A^TAz = \|A\|_2^2 z$. Dann gilt

$$\begin{aligned}\|A\|_{\infty,b} &= \max_{\|x\|_{\infty,b,m}=1} \|Ax\|_{\infty,b,n}\\ &= \max_{\|x\|_{\infty,b,m}=1} \max_{i=1,\ldots,n} \|(Ax)|_{I_i}\|_2\\ &= \max_{\|x\|_{\infty,b,m}=1} \max_{i=1,\ldots,n} \|\sum_{j=1}^m A|_{I_i\times J_j} x|_{J_j}\|_2\\ &\leq \max_{\|x\|_{\infty,b,m}=1} \max_{i=1,\ldots,n} C \max_{\substack{i'=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_{i'}\times J_j}\|_2 \|x|_{J_j}\|_2\\ &\leq C \left(\max_{\substack{i'=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_{i'}\times J_j}\|_2\right)\left(\max_{\|x\|_{\infty,b,m}=1} \max_{j=1,\ldots,m} \|x|_{J_j}\|_2\right)\\ &= C \max_{\substack{i'=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_{i'}\times J_j}\|_2.\end{aligned}$$

Analog folgt $\|A^T\|_{\infty,b} \leq C \max_{\substack{i'=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_{i'}\times J_j}\|_2$ und damit

$$\begin{aligned}\|A\|_2^2\|z\|_{\infty,b} &= \|A^TAz\|_{\infty,b}\\ &\leq \|A^T\|_{\infty,b}\|A\|_{\infty,b}\|z\|_{\infty,b}\\ &\leq C^2 \left(\max_{\substack{i=1,\ldots,n\\ j=1,\ldots,m}} \|A|_{I_i\times J_j}\|_2^2\right)\|z\|_{\infty,b}.\end{aligned}$$

■

6.2. Konvertierung

Gegeben sei eine Matrix $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$, T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum und $\varepsilon \in \mathbb{R}_{>0}$. Gesucht ist eine Rangverteilung $\tilde{k} \leq k$ und eine Matrix $\tilde{M} \in \mathcal{M}_{\mathcal{H},\tilde{k}}(T,Z)$, so daß die Approximationseigenschaft

$$\|M - \tilde{M}\|_2 \leq \varepsilon \tag{30}$$

gilt. Offenbar ist dies für die triviale Wahl $\tilde{k} = k$ erfüllt, wir suchen also unter allen $\tilde{k}$, welche die Bedingung (30) erfüllen, diejenige, welche den Aufwand $N_{\mathcal{H},St}(\tilde{k},T,Z)$ möglichst minimiert. Aus Satz 6.2 wissen wir bereits, daß $\|(M-\tilde{M})|_{I_i \times J_j}\|_2 \leq \frac{\varepsilon}{C_{sp}|L_T|}$ eine hinreichende Bedingung ist, so daß der Rang in einem Block $I_i \times J_j$ so gewählt werden kann (mit Hilfe der gekürzten Singulärwertzerlegung 2.12), daß diese Bedingung erfüllt ist.

Folgerung 6.5 *(Rangwahl bei der Konvertierung)*
Sei T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum, $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$. Definiere die Rangverteilung in den zulässigen Blättern $\tau \times \tau' \in \mathcal{L}^+(T)$ durch

$$\tilde{k}(\tau \times \tau') := \min\left\{ i \in \{0,\ldots,k(\tau\times\tau')\} \;\middle|\; \|(M|_{\tau\times\tau'}) - (M|_{\tau\times\tau'})_{\mathcal{H},i}\|_2 \leq \frac{\varepsilon}{\bar{C}_{sp}} \right\}.$$

Dann gilt für jede Bestapproximierende $M_{\mathcal{H}} \in \mathcal{M}_{\mathcal{H},\tilde{k}}(T,Z)$ von M die Approximationseigenschaft (30).

Folgerung 6.6 *(Diskretisierung)*
Sei T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum, $A \in \mathbb{R}^{I\times J}$. Definiere die Rangverteilung in den zulässigen Blättern $\tau \times \tau' \in \mathcal{L}^+(T)$ durch

$$k(\tau \times \tau') := \min\left\{ i \in \{0,\ldots,\min\{|\tau|,|\tau'|\}\} \;\middle|\; \|A|_{\tau\times\tau'} - (A|_{\tau\times\tau'})_{\mathcal{H},i}\|_2 \leq \frac{\varepsilon}{\bar{C}_{sp}} \right\}.$$

Dann gilt für jede Bestapproximierende $M_{\mathcal{H}} \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ von M die Approximationseigenschaft (30). Eine $\mathbf{R}k$-Approximation von $A|_{\tau\times\tau'}$ kann mit den Methoden aus [2] erfolgen, die blockweisen Abschätzungen werden dort allerdings nur für die Frobeniusnorm getroffen, so daß sie zwar für die Spektralnorm gültig, aber nicht scharf sind.

6.3. Addition und Multiplikation

Folgerung 6.7 *(Rangwahl bei der Addition)*
Sei T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum, $M \in \mathcal{M}_{\mathcal{H},k}(T,Z)$, $M' \in \mathcal{M}_{\mathcal{H},k'}(T,Z)$. Definiere die Rangverteilung in den zulässigen Blättern $\tau \times \tau' \in \mathcal{L}^+(T)$ durch

$$\tilde{k}(\tau \times \tau') := \max\left\{ i \in \{0,\ldots,k(\tau\times\tau') + k'(\tau\times\tau')\} \;\middle|\; \sigma_i((A+B)|_{\tau\times\tau'}) > \frac{\varepsilon}{\bar{C}_{sp}} \vee i = 0 \right\}$$

(σ_i ist der i-te Singulärwert). Dann gilt für die formatierte Addition $\oplus : \mathcal{M}_{\mathcal{H},k}(T,Z) \times \mathcal{M}_{\mathcal{H},k'}(T,Z) \to \mathcal{M}_{\mathcal{H},\tilde{k}}(T,Z)$

$$\|(A+B) - (A \oplus B)\|_2 \leq \varepsilon.$$

Folgerung 6.8 *(Rangwahl bei der Multiplikation)*
Sei T ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum, T' ein aus $T_J, T_{I'}$ gebildeter $\mathcal{H}$-Baum, $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$ und $M' \in \mathcal{M}_{\mathcal{H},k'}(T', Z')$.
Dann gilt für die formatierte Multiplikation (Bestapproximation) $\odot : \mathcal{M}_{\mathcal{H},k}(T, Z) \times \mathcal{M}_{\mathcal{H},k'}(T', Z') \to \mathcal{M}_{\mathcal{H},\tilde{k}}(\tilde{T}, \tilde{Z})$

$$\|(A \cdot B) - (A \odot B)\|_2 \leq \varepsilon,$$

wobei der Rang $\tilde{k}$ definiert wird durch

$$\tilde{k}(\tau \times \tau') := \max\left\{ i \in \{0, \ldots, (k \cdot k')(\tau \times \tau')\} \;\middle|\; \sigma_i((A \cdot B)|_{\tau \times \tau'}) > \frac{\varepsilon}{\tilde{C}_{sp}} \vee i = 0 \right\}.$$

Im Falle der Approximation (siehe Bemerkung 4.12) kann man die Berechnung von $(A \cdot B)|_{\tau \times \tau'}$ vermeiden, indem vorab die Anzahl der Summanden aus der Darstellung (15) bestimmt und der zulässige Gesamtfehler $\frac{\varepsilon}{\tilde{C}_{sp}}$ gleichmäßig auf die Summationen verteilt wird.

6.4. Inversion

Anders als bei der Addition und Multiplikation kann die Rangwahl bei der Inversion nicht a posteriori, d.h. nach Berechnung des exakten Ergebnisses, erfolgen. Möchte man hingegen den nötigen Rang zum Erreichen des relativen Fehlers $\frac{\|A^{-1} - \tilde{A^{-1}}\|_2}{\|A^{-1}\|_2} < \varepsilon$ a priori schätzen, so benötigt man Abschätzungen für die Norm der Matrix und ihrer Inversen, die noch nicht bekannt ist.

Folgerung 6.9 *(Schätzung der Spektralnorm der Inversen)*
Sei $A \in \mathbb{R}^{n,n}$ invertierbar. Mit Hilfe von Satz 4.31 ermitteln wir eine hinreichend gute Schätzung $\tilde{\Theta}$ für $\Theta := \|A\|_2^{-2}$. Die positiv semidefinite und symmetrische Iterationsmatrix des Richardson-Verfahrens (siehe [8]) ist $M_\Theta^{Rich} := I - \Theta A^T A$. Die Matrix $A^T A$ bzw. M_Θ^{Rich} muß nicht aufgestellt werden, es genügt die Matrix-Vektor-Multiplikation durchzuführen. Mit Hilfe von Satz 4.31 bestimmen wir eine hinreichend gute Schätzung $\tilde{\rho}$ für den größten Eigenwert ρ von M_Θ^{Rich}. Dann gilt

$$\|A^{-1}\|_2^2 = (1 - \rho)^{-1}\Theta \approx (1 - \tilde{\rho})^{-1}\tilde{\Theta}.$$

Beweis: Gemäß [8, Lemma 4.4.1] gilt

$$\begin{aligned} \rho &= \max\{|1 - \Theta\lambda_{min}(A^T A)|, |1 - \Theta\lambda_{max}(A^T A)|\} \\ &= 1 - \Theta\lambda_{min}(A^T A) \\ &= 1 - \Theta\lambda_{max}(A^{-1}A^{-T})^{-1} \\ &= 1 - \Theta\|A^{-1}\|_2^{-2}. \end{aligned}$$

■

Bemerkung 6.10 *(Güte der Schätzung der Spektralnorm der Inversen)*
Die Bestimmung von $\tilde{\Theta}$ genügt für vorgegebenes ε der Abschätzung

$$\frac{\Theta^{-1} - \tilde{\Theta}^{-1}}{\Theta^{-1}} \leq \varepsilon,$$

so daß man den relativen Fehler der Berechnung von $\|A^{-1}\|_2^2$ abschätzen kann durch

$$|\Theta - \tilde{\Theta}| \leq \frac{\varepsilon}{1-\varepsilon}\Theta.$$

Ein wesentlich größeres Problem stellt die Approximation von $(1-\rho)^{-1}$ dar. Hier erhält man aus $(\rho - \tilde{\rho})/\rho \leq \varepsilon$ nur die Abschätzung

$$\frac{|(1-\rho)^{-1} - (1-\tilde{\rho})^{-1}|}{(1-\rho)^{-1}} \leq \varepsilon \frac{\rho}{(1-\tilde{\rho})} \leq \varepsilon \mathrm{cond}_2(A)^2,$$

so daß bei schlecht konditionierter Matrix A eine erhebliche Fehlerverstärkung auftritt. Denselben Effekt erzielt man, wenn man etwa aus der Konvergenzrate des Gradientenverfahrens Rückschlüsse auf die Eigenwerte ziehen möchte.
Schlußfolgerung: Stellt sich bei der Berechnung von ρ das Verhalten $\tilde{\rho} \to 1$ ein, so ist die Schätzung für die Spektralnorm von A^{-1} fehlgeschlagen und A schlecht konditioniert. In diesem Fall kann man, z.B. mittels der $\mathcal{H}$-Inversion, die Inverse approximieren und deren Norm mit Hilfe von Satz 4.31 ausrechnen.

In Bemerkung 4.19 wurde bereits darauf hingewiesen, daß in der Gauß-Elimination eine Fehlerverstärkung um den Faktor $n2^n$ möglich wäre. Tritt dieser Fall bei der zu invertierenden Matrix A ein, so wird die Inversion unabhängig von der Rangwahl fehlschlagen. Nimmt man hingegen an, daß die Fehlerverstärkung nur um einen Faktor δ_A unabhängig von der Höhe des Kürzungsfehlers erfolgt, so läßt sich die Approximation mit dem folgenden Algorithmus berechnen.

Algorithmus 6.11 *(Adaptive zweistufige Inversion)*

Gegeben: Eine $\mathcal{H}$-Matrix $A \in \mathcal{M}_{\mathcal{H},k}T, Z$, ein aus T_I, T_J gebildeter $\mathcal{H}$-Baum T und eine Fehlertoleranz ε.

Gesucht: Eine approximative Inverse $A^{\ominus 1}$ von A mit $\|I - A^{\ominus 1}A\|_2 \leq \varepsilon$.

1. *Stufe: Bestimme die stufenweisen Schwachbesetztheitskonstanten $C_{sp}^{(l)}$ und $\bar{C}_{sp} := \sum_{l \in L} C_{sp}^{(l)}$ sowie die Spektralnorm $\|A\|_2$ von A mit Hilfe von Satz 4.31. Setze $\varepsilon_{local} := \varepsilon/(\bar{C}_{sp}\|A\|_2)$.*

   Führe die formatierte Inversion nach Algorithmus 4.18 in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ durch, wobei die Rangverteilung k für die Matrizen R, L, H gesondert dadurch bestimmt wird, daß bei jeder Kürzung in einem zulässigen Blatt der lokale Kürzungsfehler ε_{local} nicht überschritten wird.

Ermittle anschließend den Faktor $\delta_A := \|I - A^{\ominus}A\|_2/\varepsilon$ *der Fehlerverstärkung.*

2. *Stufe: Führe wie in (1.) die Rang-adaptive formatierte Inversion für* $\varepsilon_{local} := \varepsilon/(\bar{C}_{sp}\|A\|_2\delta_A)$ *durch (dann hebt sich der zusätzliche Faktor* $1/\delta_A$ *mit der Fehlerverstärkung um den Faktor* δ_A *gerade weg).*

Bemerkung 6.12 *(Zur adaptiven Inversion)*
Die Voraussetzung, daß die Fehlerverstärkung in allen Blöcken gleichmäßig stattfindet, ist ausgesprochen unrealistisch. Die in der Gauß-Elimination zuerst bearbeiteten Blöcke werden Auswirkungen auf alle nachfolgenden Blöcke haben, während in den zuletzt bearbeiteten Blöcken keine Fehlerverstärkung möglich ist. Die genaue Gewichtung ist jedoch sehr von der Matrix A abhängig und soll hier nicht weiter untersucht werden.

7. Approximationseigenschaft

Für die Arithmetik hierarchischer Matrizen haben wir vorausgesetzt, daß sich die zugrundeliegenden Matrizen sowie das Ergebnis und etwaige Zwischenergebnisse als $\mathcal{H}$-Matrizen darstellen lassen. Für Differentialoperatoren ist die Darstellbarkeit einfach zu zeigen, für Integraloperatoren wurde sie bereits hinreichend in der Literatur untersucht. Zum Beweis der Darstellbarkeit der Inversen beschränken wir uns auf stark elliptische Differentialoperatoren und glatte Ränder.

7.1. Notwendige und Hinreichende Bedingungen

Sei T ein $\mathcal{H}_\times$-Baum, Z eine Zulässigkeitsbedingung und k eine Rangverteilung. Zu approximieren ist eine Matrix $M \in \mathbb{R}^{n,m}$ durch eine Matrix $M_\mathcal{H} \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ bis auf einen Fehler von ε:

$$\|M - M_\mathcal{H}\| \leq \varepsilon. \tag{31}$$

Hier sei $\|\cdot\|$ die Spektral- oder Frobeniusnorm.

Bemerkung 7.1 *(Notwendige Bedingung)*
Für eine Bestapproximation $M_\mathcal{H}$ von M in der Frobeniusnorm gilt

$$\begin{aligned}\|M - M_\mathcal{H}\|_F^2 &= \sum_{t\times s\in\mathcal{L}^+(T)} \|M|_{t\times s} - (M|_{t\times s})_\mathcal{H}\|_F^2 \\ &= \sum_{t\times s\in\mathcal{L}^+(T)} \sum_{i=k(t\times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t\times s})^2\end{aligned}$$

und in der Spektralnorm gilt

$$\begin{aligned}\|M - M_\mathcal{H}\|_2 &\geq \max_{t\times s\in\mathcal{L}^+(T)} \|M|_{t\times s} - (M|_{t\times s})_\mathcal{H}\|_2 \\ &= \max_{t\times s\in\mathcal{L}^+(T)} \sigma_{k(t\times s)+1}(M|_{t\times s}).\end{aligned}$$

Eine notwendige Bedingung an die Matrix M für (31) in der Spektral- oder Frobeniusnorm ist demnach

$$\forall t\times s\in\mathcal{L}^+(T): \qquad \sigma_{k(t\times s)+1}(M|_{t\times s}) \leq \varepsilon. \tag{32}$$

Bemerkung 7.2 *(Hinreichende Bedingung)*
Für eine Bestapproximation $M_\mathcal{H}$ von M in der Frobeniusnorm gilt

$$\begin{aligned}\|M - M_\mathcal{H}\|_F^2 &= \sum_{t\times s\in\mathcal{L}^+(T)} \sum_{i=k(t\times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t\times s})^2 \\ &\leq |\mathcal{L}^+(T)| \max_{t\times s\in\mathcal{L}^+(T)} \sum_{i=k(t\times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t\times s})^2.\end{aligned}$$

Eine hinreichende Bedingung für (31) an M wäre

$$\forall t \times s \in \mathcal{L}^+(T): \quad \sum_{i=k(t\times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t\times s}) \leq \frac{\varepsilon}{\sqrt{|\mathcal{L}^+(T)|}}. \tag{33}$$

In der Spektralnorm gilt nach Satz 6.2

$$\|M - M_{\mathcal{H}}\|_2 \leq C_{sp}|L_T| \max_{t\times s\in\mathcal{L}^+(T)} \|M|_{t\times s} - (M|_{t\times s})_{\mathcal{H}}\|_2.$$

Hier ist die Bedingung

$$\forall t \times s \in \mathcal{L}^+(T): \quad \sigma_{k(t\times s)+1}(M|_{t\times s}) \leq \frac{\varepsilon}{C_{sp}|L_T|} \tag{34}$$

hinreichend für (31).

Bemerkung 7.3 *(Lokalität der Approximationseigenschaft)*
Für die Spektralnorm ist die globale Approximationseigenschaft (31) bis auf den Faktor $C_{sp}|L_T|$ äquivalent zur lokalen Approximationseigenschaft (34). Nimmt man ein gleichartiges Verhalten der Singulärwerte in allen zulässigen Blöcken der Matrix an, so ist (33) äquivalent zu (31). Im folgenden werden wir daher nur noch die lokale Approximationseigenschaft (32) in den zulässigen Blöcken der Matrix M untersuchen.

7.2. Fredholmsche Integraloperatoren

Wir fixieren den Operator

$$\mathcal{K}: H^{\alpha_\mathcal{K}+\beta_\mathcal{K}}(\Omega_y) \to H^{\beta_\mathcal{K}}(\Omega_x),\ f \mapsto u,\ \ u(x) := \int_{\Omega_y} g(x,y)f(y)\,\mathrm{d}y$$

der Ordnung $\alpha_\mathcal{K}$ und eine Diskretisierung K von $\mathcal{K}$, z.B. eine Galerkin-Diskretisierung mit integrierbaren Basisfunktionen $(b_i)_{i\in r}, (b_j)_{j\in s}$:

$$K_{ij} := \int_{\Omega_y}\int_{\Omega_x} b_i(x)g(x,y)b_j(y)\,\mathrm{d}x\,\mathrm{d}y, \quad (i,j) \in r \times s,$$

eine Kollokationsmethode mit integrierbaren Basisfunktionen $(b_j)_{j\in s}$ und Kollokationspunkten $(x_i)_{i\in r}$:

$$K_{ij} := \int_{\Omega_x} g(x_i,y)b_j(y)\,\mathrm{d}x, \quad (i,j) \in r \times s,$$

oder eine Nyström-Methode mit Gewichten $(\omega_j)_{j\in s}$ und Stützstellen $(x_i)_{i\in r}, (y_j)_{j\in s}$:

$$K_{ij} := \omega_j g(x_i,y_j), \quad (i,j) \in r \times s.$$

Die (lokale) Approximationseigenschaft für K ist gleichbedeutend damit, daß man eine $\mathbf{R}k$-Matrix $R \in \mathbb{R}^{r\times s}$ mit

$$\|K - R\|_2 \leq \varepsilon \quad \text{bzw.} \quad \|K - R\|_F \leq \varepsilon$$

findet.

Lemma 7.4 *(Approximation entarteter Kerne)*
Ist die Kernfunktion $g(x,y)$ auf $\Omega_x \times \Omega_y$ durch eine entartete Kernfunktion (ausgeartete Kernfunktion, Rang-k-Funktion, Funktional-Skeleton)

$$\tilde{g}(x,y) = \sum_{i=1}^{k} \tilde{g}_{1,i}(x)\tilde{g}_{2,i}(y)$$

bis auf einen relativen Fehler von ε approximierbar, d.h.

$$|\tilde{g}(x,y) - g(x,y)| \leq \varepsilon |g(x,y)|, \tag{35}$$

so gilt für jede $\mathbb{R}k$-Bestapproximation R von K im Fall der Nyström-Methode

$$\|K - R\|_F \leq \varepsilon \|K\|_F.$$

Sind die Basisfunktionen b_i nichtnegativ (oder nichtpositiv) und ist die Kernfunktion g auf den Trägern der Basisfunktionen nichtnegativ (oder nichtpositiv), so gilt die Abschätzung ebenfalls für die Kollokationsmethode und das Galerkinverfahren. Sind nur die Basisfunktionen nichtnegativ (oder nichtpositiv), so gilt

$$\|K - R\|_F \leq \varepsilon \|\bar{K}\|_F,$$

wobei $\bar{K}$ die entsprechende Diskretisierung von $\bar{\mathcal{K}}[f](x) := \int_{\Omega_y} |g(x,y)| f(y)\,\mathrm{dy}$ ist.

Beweis: Definiere den Operator $\tilde{\mathcal{K}}[f](x) := \int_{\Omega_y} \tilde{g}(x,y) f(y)\,\mathrm{dy}$ und die Diskretisierung $\tilde{K}$ analog zu K. Im Fall der Nyström-Methode gilt

$$|K_{ij} - \tilde{K}_{ij}| = |\omega_j||g(x_i,y_j) - \tilde{g}(x_i,y_j)| \leq \varepsilon |\omega_j||g(x_i,y_j)| = \varepsilon |K_{ij}|.$$

Für Basisfunktionen b_i und Kernfunktionen g ohne Vorzeichenwechsel auf den jeweiligen Trägern der Basisfunktionen gilt im Fall der Kollokationsmethode

$$|K_{ij} - \tilde{K}_{ij}| = \left| \int_{\Omega_y} b_j(y)(g(x_i,y) - \tilde{g}(x_i,y))\,\mathrm{dy} \right| \leq \varepsilon \int_{\Omega_y} |b_j(y)||g(x_i,y)|\,\mathrm{dy} = \varepsilon |K_{ij}|$$

sowie für das Galerkin-Verfahren

$$\begin{aligned} |K_{ij} - \tilde{K}_{ij}| &= \left| \int_{\Omega_x} \int_{\Omega_y} b_i(x)(g(x,y) - \tilde{g}(x,y)) b_j(y)\,\mathrm{dx}\,\mathrm{dy} \right| \\ &\leq \varepsilon \int_{\Omega_x} \int_{\Omega_y} |b_i(x)||g(x,y)||b_j(y)|\,\mathrm{dx}\,\mathrm{dy} \quad = \varepsilon |K_{ij}|. \end{aligned}$$

Besitzen nur die Basisfunktionen b_i keine Vorzeichenwechsel, so ist im Fall der Kollokationsmethode

$$|K_{ij} - \tilde{K}_{ij}| = \left| \int_{\Omega_y} b_j(y)(g(x_i,y) - \tilde{g}(x_i,y))\,\mathrm{dy} \right| \leq \varepsilon \int_{\Omega_y} |b_j(y)||g(x_i,y)|\,\mathrm{dy} = \varepsilon \bar{K}_{ij}$$

und für das Galerkin-Verfahren

$$\begin{aligned} |K_{ij} - \tilde{K}_{ij}| &= \left| \int_{\Omega_x} \int_{\Omega_y} b_i(x)(g(x,y) - \tilde{g}(x,y))b_j(y)\,\mathrm{dx}\,\mathrm{dy} \right| \\ &\leq \varepsilon \int_{\Omega_x} \int_{\Omega_y} |b_i(x)||g(x,y)||b_j(y)|\,\mathrm{dx}\,\mathrm{dy} \quad = \varepsilon \bar{K}_{ij}. \end{aligned}$$

■

Beispiel 7.5 *(Keine lokale Approximationseigenschaft)*
1. Beispiel: Vorzeichenwechsel bei der Kernfunktion.
Wir betrachten die Kernfunktion

$$g : [0,1] \times [0,1] \to \mathbb{R}, (x,y) \mapsto \begin{cases} 1 & x < \frac{1}{2} \\ -1 & x \geq \frac{1}{2} \end{cases}$$

und die Basisfunktion

$$b_1 : [0,1] \to \mathbb{R}, x \mapsto 1.$$

Es gilt für die Galerkin-Diskretisierung

$$K_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x) g(x,y) b_1(y)\,\mathrm{dx}\,\mathrm{dy} = 0.$$

Setzt man

$$\tilde{g} : [0,1] \times [0,1] \to \mathbb{R}, (x,y) \mapsto \begin{cases} 1-\varepsilon & x < \frac{1}{2} \\ -1 & x \geq \frac{1}{2} \end{cases},$$

so gilt $|g(x,y) - \tilde{g}(x,y)| \leq \varepsilon = \varepsilon |g(x,y)|$ *und*

$$K_{11} - \tilde{K}_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x)(g(x,y) - \tilde{g}(x,y)) b_1(y)\,\mathrm{dx}\,\mathrm{dy} = \int_{[0,1]} \int_{[0,\frac{1}{2}]} b_1(x) \varepsilon b_1(y)\,\mathrm{dx}\,\mathrm{dy} = \frac{1}{2}\varepsilon.$$

In diesem Fall ist die Abschätzung $\|K - R\|_F \leq \varepsilon \|K\|_F$ $(= 0)$ *nicht erfüllt.*
2. Beispiel: Vorzeichenwechsel bei der Basisfunktion.
Wir betrachten die Kernfunktion

$$g : [0,1] \times [0,1] \to \mathbb{R}, (x,y) \mapsto 1$$

und die Basisfunktion

$$b_1 : [0,1] \to \mathbb{R}, x \mapsto \begin{cases} 1 & x < \frac{1}{2} \\ -1 & x \geq \frac{1}{2} \end{cases}.$$

Es gilt für die Galerkin-Diskretisierung

$$\bar{K}_{11} = K_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x) g(x,y) b_1(y)\,\mathrm{dx}\,\mathrm{dy} = 0.$$

Setzt man

$$\tilde{g} : [0,1] \times [0,1] \to \mathbb{R}, (x,y) \mapsto \begin{cases} 1-\varepsilon & x,y < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \vee y \geq \frac{1}{2} \end{cases},$$

so gilt $|g(x,y) - \tilde{g}(x,y)| \leq \varepsilon$ *und*

$$K_{11} - \tilde{K}_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x)(g(x,y) - \tilde{g}(x,y)) b_1(y) \,\mathrm{dx}\,\mathrm{dy} = \int_{[0,\frac{1}{2}]} \int_{[0,\frac{1}{2}]} b_1(x) \varepsilon b_1(y) \,\mathrm{dx}\,\mathrm{dy} = \frac{1}{4}\varepsilon.$$

In diesem Fall ist weder $\|K - R\|_F \lesssim \varepsilon \|K\|_F$ *noch* $\|K - R\|_F \lesssim \varepsilon \|\bar{K}\|_F$ *erfüllt.*

Bemerkung 7.6 *(Lokale Approximationsaussagen in der Literatur)*
Abschätzungen der Form $\|K - R\| \leq \varepsilon \|K\|$ *sind für allgemeine Kernfunktionen und Ansatzräume in der Literatur nicht zu finden. Stattdessen wird häufig der globale Approximationsfehler unter speziellen Voraussetzungen (z.B. explizite Darstellung von* $\tilde{g}$*) abgeschätzt. In [2, Satz 1.3.8] wird die Abschätzung*

$$\|K - R\|_F \leq \varepsilon \left(\sum_{i \in r} \|l_i\|^2 \right)^{\frac{1}{2}} \left(\sum_{j \in s} \|\tilde{l}_j\|^2 \right)^{\frac{1}{2}} \sup_{(x,y) \in \Omega_x \times \Omega_y} |g(x,y)|$$

hergeleitet (dort wird die Diskretisierung mittels der Funktionale $l_i, \tilde{l}_j$ *verallgemeinert), in [14, Lemma 3.2] wird*

$$\|u - \tilde{u}\|_W \lesssim \inf_{v \in V} \|u - v\|_W + \frac{c(0,m)}{m!} \eta^m \|\bar{\mathcal{K}}\|_{W \leftarrow W} \|u\|_W$$

bewiesen, wobei hier der Operator $\bar{\mathcal{K}}$ *durch*

$$\bar{\mathcal{K}}[f](x) := \int_{\Omega_y} \max_{z \in \Omega_y} |g(x,z)| f(y) \,\mathrm{dy}$$

definiert ist und $\|\bar{\mathcal{K}}\|_{W \leftarrow W}$ *für spezielle Kernfunktionen weiter untersucht wird.*

Lemma 7.7 *(Approximation durch entartete Kerne)*
Die Kernfunktion g *erfülle die asymptotische Glattheitsbedingung*

$$\forall (x,y) \in \Omega_x \times \Omega_y : \quad |\partial_x^\alpha \partial_y^\beta g(x,y)| \leq c_{\alpha,\beta} \|x - y\|^{-|\alpha+\beta|} |g(x,y)|. \tag{36}$$

Das Gebiet $\Omega_x \times \Omega_y$ *sei* η*-zulässig,* x^* *das Čebyšëv-Zentrum von* Ω_x*,* y^* *das Čebyšëv-Zentrum von* Ω_y *und* $(x,y) \in \Omega_x \times \Omega_y$*. Dann gilt für die Taylorentwicklung bis zur Ordnung* m

$$\text{in } x^*, \text{ falls } \operatorname{diam}(\Omega_x) \leq \operatorname{diam}(\Omega_y): \qquad \tilde{g}(x,y) := \sum_{|\nu|=0}^{m-1} \frac{1}{\nu!} (x^* - x)^\nu \partial_x^\nu g(x^*, y)$$

$$\text{in } y^*, \text{ falls } \operatorname{diam}(\Omega_x) > \operatorname{diam}(\Omega_y): \qquad \tilde{g}(x,y) := \sum_{|\nu|=0}^{m-1} \frac{1}{\nu!} (y^* - y)^\nu \partial_y^\nu g(x, y^*)$$

die Abschätzung

$$|g(x,y) - \tilde{g}(x,y)| \leq \max\left\{ \max_{|\alpha|\leq m} c_{\alpha,0} \ , \ \max_{|\beta|\leq m} c_{0,\beta} \right\} \eta^m |g(x,y)|.$$

Beweis: *[13, Lemma 3.15]*

Bemerkung 7.8 *(Darstellung bei Addition, Multiplikation und Inversion)*
Addition:
Sind für zwei Matrizen $K_1, K_2 \in \mathbb{R}^{r\times s}$ $\mathbf{R}k$*-Approximationen* R_1, R_2 *mit* $\|K_1 - R_1\| \leq \varepsilon\|K_1\|$, $\|K_2 - R_2\| \leq \varepsilon\|K_2\|$ *gegeben, so ist* $R_1 + R_2$ *eine* $\mathbf{R}2k$*-Approximation von* $K_1 + K_2$ *mit* $\|(K_1 + K_2) - (R_1 + R_2)\| \leq \varepsilon(\|K_1\| + \|K_2\|)$. *Diese Abschätzung ist allerdings pessimistisch: Gilt für* g_1, g_2 *die asymptotische Glattheit (36), so erfüllt auch* $g_1 + g_2$ *diese, so daß für* $g_1 + g_2$ *mit demselben Rang die gleiche Approximationsgüte erzielt werden kann.*
Multiplikation:
Sind für zwei Matrizen $K_1 \in \mathbb{R}^{r\times s}, K_2 \in \mathbb{R}^{s\times t}$ $\mathcal{H}$*-Approximationen* H_1, H_2 *mit* $\|K_1 - H_1\| \leq \varepsilon\|K_1\|$, $\|K_2 - H_2\| \leq \varepsilon\|K_2\|$ *gegeben, so ist* H_1H_2 *eine* $\mathcal{H}$*-Approximation von* K_1K_2 *mit*

$$\|K_1K_2 - H_1H_2\| \leq \varepsilon\|K_1\|\|K_2\| + \varepsilon\|H_1\|\|K_2\| \approx 2\varepsilon\|K_1\|\|K_2\|.$$

Hier erhöht sich der Rang entsprechend Satz 5.26 und der relative Fehler ist für $\|K_1K_2\| \approx \|K_1\|\|K_2\|$ *gleich gut. Sind die zugrundeliegenden Kernfunktionen* g_1, g_2 *auf zulässigen Clustern asymptotisch glatt, so erhält man aus*

$$\mathcal{K}_1[\mathcal{K}_2[f]](x) = \int_{\Omega_z} \underbrace{\int_{\Omega_y} g_1(x,y)g_2(y,z)\,\mathrm{d}y}_{=:\hat{g}(x,z)} f(z)\,\mathrm{d}z,$$

allerdings nicht *die asymptotische Glattheit von* $\hat{g}$.
Inversion:
Für die Inversion würde man Aussagen brauchen, ob sich die inverse Matrix blockweise durch niedrigen Rang approximieren läßt. Dies wäre der Fall, wenn sich die Einträge der Inversen wieder als Diskretisierung eines Integraloperators mit asymptotisch glattem Kern interpretieren ließen. Bislang ist (mir) weder bekannt, ob und unter welchen Bedingungen sich die Inverse wieder als Integraloperator auffassen läßt, noch ob die entsprechende Kernfunktion geeignete Glattheitsannahmen erfüllt oder durch Rang-k-Funktionen angenähert werden kann.

7.3. Differentialoperatoren

Sei Ω ein beschränktes Gebiet in $\mathbb{R}^d$, $\Gamma := \partial\Omega$ eine C^∞-Oberfläche und Ω liege jeweils nur auf einer Seite von Γ. Wir fixieren einen stark elliptischen linearen Differentialoperator

$\mathcal{A}$ der Ordnung $2\alpha_{\mathcal{A}}$ mit C^∞-Koeffizienten auf einer offenen Obermenge $\tilde{\Omega}$ von $\bar{\Omega}$ und setzen Existenz und Eindeutigkeit von Lösungen des Randwertproblems

$$\mathcal{A}u(x) = f(x) \qquad x \in \Omega, \tag{37}$$
$$u(x) = 0 \qquad x \in \Gamma \tag{38}$$

voraus:

$$\forall f \in L^2(\Omega)\ \exists u \in H^{2\alpha_{\mathcal{A}}}(\Omega): \qquad \mathcal{A}u = f, \quad u|_\Gamma = 0,$$
$$\forall u \in H^{2\alpha_{\mathcal{A}}}(\Omega): \qquad \mathcal{A}u = 0 \ \wedge\ u|_\Gamma = 0 \ \Rightarrow u = 0.$$

Ferner sei die Singularitätenfunktion $s(x,y)$ (Fundamentallösung, Elementarlösung, in älterer Literatur manchmal Greensche Funktion) mit

$$\mathcal{A}\int_{\tilde{\Omega}} s(x,y)f(y)\,\mathrm{dy} = f(x) \qquad \forall x \in \tilde{\Omega}, f \in C_0^\infty(\tilde{\Omega}),$$
$$\int_{\tilde{\Omega}} s(x,y)(\mathcal{A}u)(y)\,\mathrm{dy} = u(x) \qquad \forall x \in \tilde{\Omega}, u \in C_0^\infty(\tilde{\Omega}).$$

bekannt.

Bemerkung 7.9 *(Darstellbarkeit des diskreten Differentialoperators)*
Die Lokalität des Differentialoperators $\mathcal{A}$ bewirkt, daß zur Darstellung einer Galerkin-Diskretisierung A in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ ein zulässiger Knoten $r \times s$ lediglich die Bedingung $D(r) \cap D(s)$ für die Träger $D(X) := \cup_{i\in X} \operatorname{supp}\phi_i$ der Basisfunktionen erfüllen muß. Diese Bedingung ist bei der Standard-Zulässigkeitsbedingung immer erfüllt, allgemeine Zulässigkeitsbedingungen sollte man entsprechend ergänzen. Für alle zulässigen Blätter $r \times s$ gilt

$$A|_{r\times s} = 0,$$

also $A \in \mathcal{M}_{\mathcal{H},0}(T,Z)$.

Bemerkung 7.10 *(Addition und Multiplikation)*
Die Addition zweier Matrizen $A \in \mathcal{M}_{\mathcal{H},0}(T,Z), B \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ liegt wieder in derselben Klasse $\mathcal{M}_{\mathcal{H},k}(T,Z)$. Das Produkt aus zwei Matrizen $A \in \mathcal{M}_{\mathcal{H},0}(T,Z)$, $B \in \mathcal{M}_{\mathcal{H},k}(T',Z')$ mit aus T_I, T_J bzw. $T_J, T_{I'}$ gebildeten $\mathcal{H}$-Bäumen T, T' liegt in $\mathcal{M}_{\mathcal{H},k}(T\cdot T', Z)$ (siehe Lemma 4.11). Für $T = T'$ benötigt man die Idempotenz der Partitionsmultiplikation, damit das Produkt der Matrizen wieder zu $\mathcal{M}_{\mathcal{H},k}(T,Z)$ gehört.

Satz 7.11 *(Lösungsdarstellung durch Greensche Funktion)*
Es existieren Konstanten $C_{\alpha,\beta}$ und eine kompensierende Kernfunktion g_c, so daß für $p := 2\alpha_{\mathcal{A}} - |\alpha| - |\beta|$, $(x,y) \in \Omega\times\Omega$ und $l(x,y) := \operatorname{dist}(x,\Gamma) + \operatorname{dist}(y,\Gamma)$

$$|\partial_x^\alpha \partial_y^\beta g_c(x,y)| \le C_{\alpha,\beta} \begin{cases} l(x,y)^{\min(p-d,0)} & p \neq d \\ 1 + \log(\frac{1}{l(x,y)}) & p = d. \end{cases} \tag{39}$$

und für eine Lösung u von (37,38), $f \in L^2(\Omega)$ und $x \in \Omega$

$$u(x) = \int_\Omega (s(x,y) - g_c(x,y)) f(y)\,\mathrm{dy}$$

gilt.
Beweis: *[28, VI., Theorem 4.2]*

Definition 7.12 *(Erweiterte Standard-Zulässigkeitsbedingung)*
Gegeben seien Indexmengen I und J, $\mathcal{H}$-Bäume T_I von I und T_J von J sowie Mengen $\{\tau_i \mid i \in I\}, \{\sigma_j \mid j \in J\}$, deren Elemente Teilmengen des $\mathbb{R}^d$ sind und die zugrundeliegende Geometrie charakterisieren (zum Beispiel Träger von Basisfunktionen oder Kollokationspunkte). Der Rand $\Gamma := \partial\Omega$ des Gebietes Ω sei vorgegeben. Ein Knoten $r \times s \in T_I \otimes T_J$ heißt zum Parameter η absolut zulässig, *falls für $\tau := \bigcup_{i \in r} \tau_i \subset \Omega$ und $\sigma := \bigcup_{j \in s} \sigma_j \subset \Omega$*

$$\min\{\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\} \leq 2\eta \min \begin{cases} \operatorname{dist}(\tau,\Gamma) + \operatorname{dist}(\Gamma,\sigma) \\ \operatorname{dist}(\tau,\sigma) \end{cases} \tag{40}$$

ist. Die erweiterte Standard-Zulässigkeitsbedingung $\bar{Z}_\eta$ (zum Parameter η) ist definiert als

$$\bar{Z}_\eta(r \times s) := \begin{cases} \text{„zulässig“} & \text{falls } r \times s \text{ absolut zulässig ist,} \\ \text{„nicht zulässig“} & \text{sonst.} \end{cases} \tag{41}$$

Satz 7.13 *(Approximation durch entartete Kerne)*
Gegeben sei eine asymptotisch glatte Singularitätenfunktion $s(x,y)$ von $\mathcal{A}$:

$$\forall (x,y) \in \tau \times \sigma : \quad |\partial_x^\alpha \partial_y^\beta s(x,y)| \leq C_{\alpha,\beta} \|x-y\|^{-|\alpha|-|\beta|} |s(x,y)|. \tag{42}$$

Das Gebiet $\tau \times \sigma$ sei absolut zulässig, x^* das Čebyšëv-Zentrum von τ, y^* das Čebyšëv-Zentrum von σ und $(x,y) \in \tau \times \sigma$. Dann gilt für die Taylorentwicklung von $g(x,y) := s(x,y) - g_c(x,y)$ bis zur Ordnung m

$$\text{in } x^*, \text{ falls } \operatorname{diam}(\tau) \leq \operatorname{diam}(\sigma): \qquad \tilde{g}(x,y) := \sum_{|\nu|=0}^{m-1} \frac{1}{\nu!} (x^* - x)^\nu \partial_x^\nu g(x^*, y)$$

$$\text{in } y^*, \text{ falls } \operatorname{diam}(\tau) > \operatorname{diam}(\sigma): \qquad \tilde{g}(x,y) := \sum_{|\nu|=0}^{m-1} \frac{1}{\nu!} (y^* - y)^\nu \partial_y^\nu g(x, y^*)$$

die Abschätzung

$$|g(x,y) - \tilde{g}(x,y)| \leq C\eta^m |g(x,y)| + C'\eta^m.$$

Der Faktor C hängt von m ab, ist allerdings beschränkt, falls η hinreichend klein ist und die Faktoren $C_{\alpha,0}, C_{0,\beta}$ aus der asymptotischen Glattheitsbedingung $C_{\alpha,0} = O(|\alpha|!)$ und $C_{0,\beta} = O(|\beta|!)$ erfüllen (für $C_{\alpha,0} = O(|\alpha|! C^{|\alpha|})$ erhält man die Abschätzung entsprechend mit $(C\eta)^m$ statt η^m).
Der Faktor C' hängt vom Gebiet Ω, der Entwicklungsordnung m und der kompensierenden Kernfunktion g_c ab.

Beweis: O.B.d.A. sei $\mathrm{diam}(\tau) \leq \mathrm{diam}(\sigma)$. Der Restterm der Taylorentwicklung bis zur Ordnung m erfüllt für ein $\zeta \in \tau$

$$\begin{aligned}
|g(x,y) - \tilde{g}(x,y)| &\leq \frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha g(\zeta, y)| \\
&= \frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha s(\zeta, y) + \partial_x^\alpha g_c(\zeta, y)| \\
&\leq \frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha s(\zeta, y)| + \frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha g_c(\zeta, y)|.
\end{aligned}$$

1. Term: Aus der asymptotischen Glattheit von s folgt

$$|\partial_x^\alpha s(\zeta, y)| \leq C_m \|\zeta - y\|^{-m} |s(\zeta, y)|. \tag{43}$$

Sei $\zeta \in \tau$ so gewählt, daß es $|s(\zeta, y)|$ maximiert. Mit Hilfe von

$$\begin{aligned}
|s(\zeta,y)| - |s(x,y)| &= ||s(\zeta,y)| - |s(x,y)|| \\
&\leq |s(\zeta,y) - s(x,y)| \\
&\overset{\text{Mittelw.-Ungl.}}{\leq} \|\zeta - x\| \max_{\xi \in [\zeta,x]} \partial_x s(\xi, y) \\
&\overset{(42)}{\leq} C_2 \|\zeta - x\| \max_{\xi \in [\zeta,x]} \|\xi - y\|^{-1} |s(\zeta,y)| \\
&\leq C_2 \mathrm{diam}(\tau) \mathrm{dist}(\tau,\sigma)^{-1} |s(\zeta,y)| \\
&\overset{(40)}{\leq} 2C_2\eta |s(\zeta,y)|
\end{aligned}$$

erhalten wir (für hinreichend kleines η) $|s(\zeta,y)| \leq \frac{1}{1-2C_2\eta}|s(x,y)|$ und damit die Abschätzung für den relativen Fehler:

$$\begin{aligned}
\frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha s(\zeta, y)| &\leq \frac{1}{m!}\|x^* - x\|^m C_m \|\zeta - y\|^{-m} \frac{1}{1 - 2C_2\eta} |s(\zeta,y)| \\
&\leq C_m \frac{1}{1-2C_2\eta} \frac{1}{m!} (\frac{1}{2}\mathrm{diam}(\tau))^m \mathrm{dist}(\tau,\sigma)^{-m} |s(x,y)| \\
&\overset{(40)}{\leq} \frac{C_m}{m!} \frac{1}{1-2C_2\eta} \eta^m |s(x,y)|.
\end{aligned}$$

2. Term: Wir nehmen an, daß m so groß ist, daß $p \neq d$ gilt (siehe Satz 7.11). Dann folgt

$$\begin{aligned}
\frac{1}{m!}\|x^* - x\|^m \max_{|\alpha|=m} |\partial_x^\alpha g_c(\zeta, y)| &\overset{(39)}{\leq} \frac{C_m}{m!}\|x^* - x\|^m l(\zeta,y)^{2\alpha_{\mathcal{A}} - d - m} \\
&\leq \frac{C_m}{m!}(\frac{1}{2}\mathrm{diam}(\tau))^m l(\zeta,y)^{2\alpha_{\mathcal{A}} - d - m} \\
&\overset{(40)}{\leq} \frac{C_m}{m!}\eta^m l(\zeta,y)^{2\alpha_{\mathcal{A}} - d}.
\end{aligned}$$

Gesamt:

$$|g(x,y) - \tilde{g}(x,y)| \leq \frac{C_m}{m!}\frac{1}{1-2C_2\eta}\eta^m |s(x,y)| + \frac{C_m}{m!}\eta^m l(\zeta,y)^{2\alpha_{\mathcal{A}}-d}$$

■

Lemma 7.14 *(Darstellung der diskreten Inversen)*
Die Inverse des Operators $\mathcal{A}$ sei in der Darstellung

$$\mathcal{A}^{-1}[f](x) = \int_\Omega g(x,y) f(y)\,\mathrm{d}y, \quad f \in L^2(\Omega),$$

gegeben. A sei eine Galerkin-Diskretisierung von $\mathcal{A}$ mit Lagrange-Basisfunktionen $(\phi_i)_{i=1}^n$ zu Knoten $(x_i)_{i=1}^n$:

$$\begin{aligned} M_{ij} &:= \int_\Omega \phi_i(y)\phi_j(y)\,\mathrm{d}y, \\ A_{ij} &:= (\phi_i, \mathcal{A}\phi_j) := \int_\Omega \phi_i(y)\mathcal{A}\phi_j(y)\,\mathrm{d}y\,. \end{aligned}$$

Dann gilt für die Einträge der diskreten Inversen

$$\begin{aligned} (A^{-1})_{ij} &= \Pi\mathcal{A}^{-1}[\tilde{\phi}_j](x_i), \\ \tilde{\phi}_j &:= \sum_{\nu=1}^n (M^{-1})_{\nu j}\phi_\nu, \end{aligned}$$

wobei $\Pi : H^{2\alpha_{\mathcal{A}}}(\Omega) \to \langle \phi_1, \ldots, \phi_n\rangle$ die Galerkin-Projektion

$$\forall i \in \{1,\ldots,n\}: \quad (\phi_i, \mathcal{A}\Pi u) = (\phi_i, \mathcal{A}u)$$

für Funktionen $u \in H^{2\alpha_{\mathcal{A}}}(\Omega)$ ist.

Beweis: Wir setzen $\tilde{u}^{(j)} := \mathcal{A}^{-1}[\tilde{\phi}_j]$. Der Vektor $u^{(j)}$ der Basisdarstellung von $\Pi\tilde{u}^{(j)}$ ist $u_i^{(j)} = (\Pi\tilde{u}^{(j)})[x_i]$. Wir zeigen $Au^{(j)} = e_j$ für alle $j = 1,\ldots,n$. Es gilt für alle $i \in \{1,\ldots,n\}$:

$$\begin{aligned} e_i^T A u^{(j)} &= \sum_{\nu=1}^n e_i^T A e_\nu u_\nu^{(j)} = \sum_{\nu=1}^n (\phi_i, \mathcal{A}\phi_\nu) u_\nu^{(j)} \\ &= (\phi_i, \mathcal{A}\Pi\tilde{u}^{(j)}) = (\phi_i, \mathcal{A}\tilde{u}^{(j)}) \\ &= (\phi_i, \tilde{\phi}_j) \\ &= \sum_{\nu=1}^n (M^{-1})_{\nu j}(\phi_i, \phi_\nu) \\ &= \sum_{\nu=1}^n (M^{-1})_{\nu j} M_{i\nu} \\ &= (M \cdot M^{-1})_{ij} = \delta_{ij}. \end{aligned}$$

Es folgt $(A^{-1})_{ij} = (A^{-1}e_j)_i = u_i^{(j)} = (\Pi\mathcal{A}^{-1}[\tilde{\phi}_j])(x_i)$. ∎

Bemerkung 7.15 *(Inversion)*
Die diskrete Inverse A^{-1} eines stark elliptischen Differentialoperators $\mathcal{A}$ auf einem Gebiet mit glattem Rand entspricht nach Satz 7.11 und Lemma 7.14 bis auf den Dikretisierungsfehler (durch Π) der Diskretisierung eines Integraloperators mit der Kollokationsmethode: Die Kollokationspunkte sind die Knoten x_i der Lagrange-Basisfunktionen ϕ_i und die Basisfunktionen sind $(\tilde{\phi}_i)_{i=1}^n$. Es ist zu beachten, daß die Basisfunktionen $\tilde{\phi}_i$ nicht lokal sind, so daß die für die Integraloperatoren gültigen Approximationsaussagen hier nicht gelten. Zudem sind die Konstanten für die asymptotische Glattheit von dem Gebiet abhängig und die kompensierende Kernfunktion g_c erlaubt keine relativen Fehlerabschätzungen bei der Approximation durch einen entarteten Kern. Die Gebietsabhängigkeit und die erweiterte Standard-Zulässigkeitsbedingung $\bar{Z}_\eta$ sind für die Block-Gauß-Elimination zur Inversion zu restriktiv: Nach Bemerkung 4.16, Formel (17), sind die im Laufe des Verfahrens zu approximierenden Inversen

$$(R_{\nu\nu}^{(\nu-1)})^{-1} = ((A^{\underline{j}})^{-1})_{jj}$$

die entsprechenden Inversen von $\mathcal{A}$ auf einem Teilgebiet. Eine feinere Clusterung zum Rand, wie es die erweiterte Zulässigkeitsbedingung $\bar{Z}_\eta$ fordert, ist hier nur bedingt möglich, und gebietsabhängige Konstanten wären von allen auftretenden Teilgebieten abhängig.
In der Praxis zeigt sich ein gegenteiliges Verhalten: Auch bei Gebieten mit vielen einspringenden Ecken ist die Standard-Zulässigkeitsbedingung ausreichend und die Block-Gauß-Elimination zur Inversion liefert annähernd die Bestapproximation der Inversen. Selbst springende Koeffizienten im Differentialoperator $\mathcal{A}$ erlauben eine $\mathcal{H}$-Approximation der Inversen (siehe hierzu Abschnitte 8.2.2 und 8.2.3). Hier ist es entscheidend, daß lediglich eine Approximation durch niedrigen Rang in den zulässigen Blöcken der Matrix nötig ist. Der Umweg über die asymptotische Glattheit der Greenschen Funktion ist nur zum Beweis der Approximierbarkeit durch eine gekürzte Taylorentwicklung erforderlich.
Für konkrete Probleme ist eine bessere Kenntnis der Konstanten in der Greenschen Funktion relevant. Differentialoperatoren, deren Koeffizienten gebietsabhängig sind, erfordern eine spezielle Diskretisierung und Clusterung. Da der Einfluß des Randes und somit der kompensierenden Kernfunktion g_c vernachlässigbar zu sein scheint, genügt es, eine Näherung für die Singularitätenfunktion zu bestimmen und aus ihr die Kriterien für die Clusterung (Zulässigkeit) abzuleiten.

Beispiel 7.16 *(Greensche Funktion auf der Einheitskugel)*
Auf der Einheitskugel $\Omega := \{x \in \mathbb{R}^3 \mid \|x\|_2 \leq 1\}$ ist die Greensche Funktion des Laplace-Operators durch

$$g(x,y) = \frac{1}{4\pi}\|x-y\|_2^{-1} - \frac{1}{4\pi}\left\|x\|y\|_2 - \frac{y}{\|y\|_2}\right\|_2^{-1}$$

gegeben ([9, Satz 3.3.1]). Die Singularitätenfunktion $s(x,y) = \frac{1}{4\pi}\|x-y\|_2^{-1}$ *erfüllt offenbar die asymptotische Glattheitsbedingung. Der Punkt* $y/\|y\|_2$ *liegt auf dem Rand* Γ *des Gebietes und gemäß Satz 7.11 erwarten wir ein singuläres Verhalten, falls* $x, y \to \Gamma$. *Für die Ableitungen der Greenschen Funktion gilt*

$$\forall m \in \mathbb{N}: \lim_{x,y\to\Gamma} \partial_{x_i}^m g(x,y) \quad = \quad 0,$$

so daß am Rand ein besonders „günstiges" Verhalten zu beobachten ist.

Beweis: Gegeben seien zwei Funktionen $f, h : \Omega \times \Omega \to \mathbb{R}$, deren Ableitungen nach x_i durch

$$\begin{aligned} \partial_{x_i} f(x,y) &= -(4\pi)^2 f(x,y)^3 h(x,y), && (44)\\ \partial_{x_i} h(x,y) &= c(y) && (45) \end{aligned}$$

gegeben sind. Per Induktion zeigt man leicht, daß es dann ein Polynom $p_m : \mathbb{R}\times\mathbb{R}\times\mathbb{R} \to \mathbb{R}$ gibt, welches

$$\partial_{x_i}^m f(x,y) = p_m(f(x,y), h(h,y), c(y)) \qquad (46)$$

erfüllt. Wir zeigen nun, daß sowohl die Singularitätenfunktion s als auch die kompensierende Kernfunktion g_c die Bedingungen (44) und (45) erfüllen, also für dasselbe Polynom p_m der Ableitungsvorschrift (46) genügen. Für die Singularitätenfunktion s gilt

$$\begin{aligned} \partial_{x_i} s(x,y) &= \partial_{x_i} \frac{1}{4\pi}\|x-y\|_2^{-1} \\ &= -\frac{1}{4\pi}\|x-y\|_2^{-3}(x_i - y_i) \\ &= -(4\pi)^2 s(x,y)^3 (x_i - y_i). \end{aligned}$$

Die Funktion $(x,y) \mapsto (x_i - y_i)$ erfüllt (45) mit $c(y) \equiv 1$. Für die kompensierende Kernfunktion g_c gilt

$$\begin{aligned} \partial_{x_i} g_c(x,y) &= \partial_{x_i} \frac{1}{4\pi}\left\|x\|y\|_2 - y\|y\|_2^{-1}\right\|_2^{-1} \\ &= -\frac{1}{4\pi}\left\|x\|y\|_2 - y\|y\|_2^{-1}\right\|_2^{-3} \|y\|_2 (x_i\|y\|_2 - y_i\|y\|_2^{-1}), \end{aligned}$$

und die Funktion $(x,y) \mapsto \|y\|_2(x_i\|y\|_2 - y_i\|y\|_2^{-1})$ erfüllt (45) mit $c(y) = \|y\|_2^2$. Damit erhalten wir

$$\begin{aligned} \lim_{y\to\Gamma} \partial_{x_i}^m g(x,y) &= \lim_{y\to\Gamma} \partial_{x_i}^m s(x,y) - \lim_{y\to\Gamma} \partial_{x_i}^m g_c(x,y) \\ &= \lim_{y\to\Gamma} p_m(s(x,y), x_i - y_i, 1) - \lim_{y\to\Gamma} p_m(g_c(x,y), \|y\|_2(x_i\|y\|_2 - y_i\|y\|_2^{-1}), \|y\|_2^2) \\ &= \lim_{y\to\Gamma} p_m(s(x,y), x_i - y_i, 1) - \lim_{y\to\Gamma} p_m(s(x,y), x_i - y_i, 1) \\ &= 0. \end{aligned}$$

■

8. Anwendungen

Die Anwendungen und numerischen Tests zu hierarchischen Matrizen gliedern sich in drei Teile: Zuerst führen wir für ein Referenzproblem, das Einfachschichtpotential, eine Approximation durch eine $\mathcal{H}$-Matrix durch. Die Kernfunktion und ihre Ableitungen sind explizit gegeben, entsprechend sind die Konstanten aus den Abschätzungen für die Approximationseigenschaft bekannt. Die Auswirkungen der Parameterwahl für die Standard-Zulässigkeitsbedingung (Z_η) und für die Rangverteilung (k) sind hier zu sehen. Als zweites wird die Inversion von Matrizen, die aus der Diskretisierung partieller Differentialgleichungen stammen, untersucht. Die Singularitätenfunktion s und die kompensierende Kernfunktion g_c werden nicht als bekannt vorausgesetzt.
Zuletzt werden wir die $\mathcal{H}$-Matrix-Arithmetik zur Auflösung von Matrixgleichungen verwenden und den Einfluß der approximativen Arithmetik auf die Lösungsverfahren studieren.

8.1. Referenzproblem: Einfachschichtpotential

Sei $\Omega := \{x \in \mathbb{R}^3 \mid \|x\|_2 < 1\}$ und $\Gamma := \partial\Omega$. Der Operator zum Einfachschichtpotential

$$\begin{aligned} \mathcal{K} &: L^\infty(\Gamma) \to C(\Gamma), \\ \mathcal{K}[f](x) &:= \frac{1}{4\pi} \int_\Gamma \frac{f(y)}{\|x-y\|_2} \, \mathrm{d}\Gamma_\mathrm{y} \end{aligned}$$

wird mit der Kollokationsmethode diskretisiert:

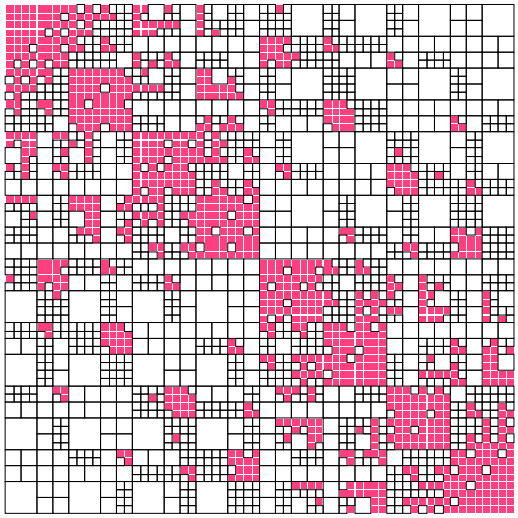
$$\begin{aligned} K &: \mathbb{R}^n \to \mathbb{R}^n, \\ K_{ij} &:= \frac{1}{4\pi} \int_\Gamma \frac{b_j(y)}{\|x_i-y\|_2} \, \mathrm{d}\Gamma_\mathrm{y} \, . \end{aligned}$$

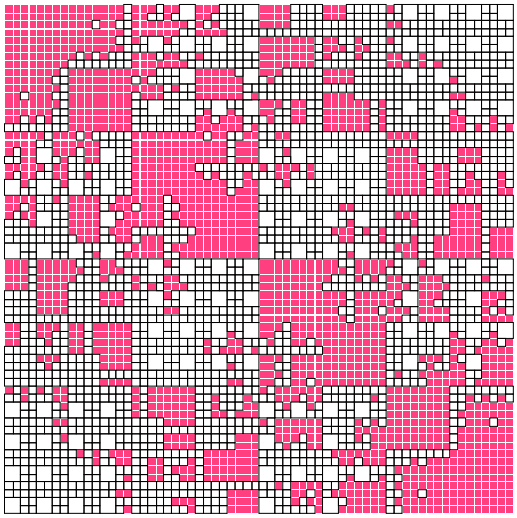
Als Basis $(b_j)_{j=1}^n$ wählen wir die charakteristischen Funktionen auf Dreiecken (regelmäßige Oberflächentriangulation von Γ) und als Kollokationspunkte $(x_i)_{i=1}^n$ die Mittelpunkte der Dreiecke. Der Cluster-Baum T_I zur Indexmenge $I = \{1, \ldots, n\}$ wird mit dem BSP-Algorithmus kardinalitätsbalanciert erzeugt und T ist der minimal aus T_I, T_I gebildete $\mathcal{H}_\times$-Baum.

8.1.1. Approximation der Kernfunktion durch eine Taylorentwicklung

Die Kernfunktion $s(x,y) = \|x_i - y\|_2^{-1}$ wird auf bzgl. der Standard-Zulässigkeitsbedingung Z_η zulässigen Mengenprodukten $\tau \times \sigma$ durch ihre Taylorentwicklung bis zur Ordnung $m = 1,2,3,4$ ersetzt, was einem Rang von $k = 1,4,10,20$ entspricht. In Abbildung 16 sind die relativen Approximationsfehler der Bestapproximation $K_\mathcal{H} \in \mathcal{M}_{\mathcal{H},k}(T, Z_\eta)$ und der Taylor-Approximation $K_\mathrm{T} \in \mathcal{M}_{\mathcal{H},k}(T, Z_\eta)$ für $\eta \in \{0.5, 0.8\}$ und $n \in \{2048, 8192\}$ dargestellt. Die Abhängigkeit von η^m (siehe Lemma 7.7) ist nur schwach zu erkennen: Die Approximation ist wesentlich besser als es die Theorie vorhersagt, es scheint so, als sei η wesentlich kleiner gewählt. Im Fall der Taylor-Approximation könnte

man $\eta \approx 0.25$ vermuten. Der Grund dafür, daß η kleiner zu sein scheint ist der, daß nicht alle zulässigen Blöcke die Zulässigkeitsbedingung scharf erfüllen.
Die Approximation durch Ersetzen der Kernfunktion durch ihre Taylorentwicklung bis zur Ordnung m ist sehr viel schlechter als die Bestapproximation. Eine alternative Vorgehensweise zur Bestimmung der $\mathbf{R}k$-Matrizen in den zulässigen Blöcken wird in [2] vorgestellt.





n=2048 $\boldsymbol{\eta = .8}$	$\frac{\|K-K_{\mathcal{H}}\|_2}{\|K\|_2}$	η?	$\frac{\|K-K_{\mathrm{T}}\|_2}{\|K\|_2}$	η?
$m=1$	$4.9\cdot10^{-3}$	–	$1.2\cdot10^{-2}$	–
$m=2$	$8.5\cdot10^{-5}$	.017	$3.5\cdot10^{-3}$	.29
$m=3$	$2.7\cdot10^{-7}$	.003	$7.5\cdot10^{-4}$	.21
$m=4$	$3.7\cdot10^{-10}$	.001	$1.7\cdot10^{-4}$	.23

n=2048 $\boldsymbol{\eta = .5}$	$\frac{\|K-K_{\mathcal{H}}\|_2}{\|K\|_2}$	η?	$\frac{\|K-K_{\mathrm{T}}\|_2}{\|K\|_2}$	η?
$m=1$	$2.2\cdot10^{-3}$	–	$1.3\cdot10^{-2}$	–
$m=2$	$1.5\cdot10^{-5}$	.0068	$1.7\cdot10^{-3}$	.13
$m=3$	$9.8\cdot10^{-9}$	.0007	$2.6\cdot10^{-4}$	.15
$m=4$	$5.4\cdot10^{-12}$	.0006	$3.9\cdot10^{-5}$	.15

n=8192 $\boldsymbol{\eta = .8}$	$\frac{\|K-K_{\mathcal{H}}\|_2}{\|K\|_2}$	η?	$\frac{\|K-K_{\mathrm{T}}\|_2}{\|K\|_2}$	η?
$m=1$	$5.1\cdot10^{-3}$	–	$3.5\cdot10^{-2}$	–
$m=2$	$9.0\cdot10^{-5}$	.018	$3.5\cdot10^{-3}$	.10
$m=3$	$2.6\cdot10^{-7}$	.003	$7.8\cdot10^{-4}$	.22
$m=4$	$4.2\cdot10^{-10}$	.002	$1.8\cdot10^{-4}$	.23

n=8192 $\boldsymbol{\eta = .5}$	$\frac{\|K-K_{\mathcal{H}}\|_2}{\|K\|_2}$	η?	$\frac{\|K-K_{\mathrm{T}}\|_2}{\|K\|_2}$	η?
$m=1$	$2.5\cdot10^{-3}$	–	$2.3\cdot10^{-2}$	–
$m=2$	$1.8\cdot10^{-5}$	.0072	$1.7\cdot10^{-3}$	.07
$m=3$	$1.1\cdot10^{-8}$	.0006	$3.0\cdot10^{-4}$	.18
$m=4$	$6.5\cdot10^{-12}$	.0006	$4.9\cdot10^{-5}$	.16

Abbildung 16: Approximation der Kollokationsmatrix K des Einfachschichtpotentials durch eine $\mathcal{H}$-Matrix $K_{\mathcal{H}}$ (Bestapproximation) und durch die mit der Taylorentwicklung der Kernfunktion entstehende $\mathcal{H}$-Matrix K_{T}. Über den Tabellen sind die $\mathcal{H}$-Matrix-Strukturen für $\eta = 0.8$ bzw. $\eta = 0.5$ und $n = 2048$ Freiheitsgrade abgebildet (nicht zulässige Blöcke sind rot/dunkel).

8.1.2. Dirichlet-Randwertaufgabe als Integralgleichung 1. Art für das Einfachschichtpotential

Ist f integrierbar, so genügt $\mathcal{K}_\Omega[f]$ mit

$$\begin{aligned} \mathcal{K}_\Omega &: L^\infty(\Gamma) \to C(\Omega), \\ \mathcal{K}_\Omega[f](x) &:= \frac{1}{4\pi} \int_\Gamma \frac{f(y)}{\|x-y\|_2} \, \mathrm{d}\Gamma_\mathrm{y}, \end{aligned}$$

nach [10, Lemma 8.1.3] der Potentialgleichung (Laplace-Gleichung)

$$-\triangle \mathcal{K}_\Omega[f] := -(\partial_x^2 + \partial_y^2 + \partial_z^2)\mathcal{K}_\Omega[f] = 0 \tag{47}$$

in Ω. Unter allen Belegungen f, deren Einfachschichtpotential $\mathcal{K}_\Omega[f]$ die Laplace-Gleichung löst, suchen wir diejenige, welche die Dirichlet-Randbedingung

$$\mathcal{K}[f](x) = \phi(x), \quad x \in \Gamma, \tag{48}$$

für ein vorgegebenes $\phi \in C^2(\Gamma)$ erfüllt. Gemäß [10, Satz 8.1.22] existiert eine Lösung $f \in C^1(\Gamma)$ von (48), die nach [10, Satz 8.1.20] außerdem eindeutig ist. Die über die $\mathcal{H}$-Inverse gewonnene Näherungslösung

$$\tilde{f}(x) := \sum_{i=1}^{n} (K^{\ominus 1} v)_i b_i(x), \qquad v_i := \phi(x_i),$$

kann in die $\mathcal{H}$-Approximation $\mathcal{K}_{\Omega,\mathcal{H}}$ des Operators $\mathcal{K}_\Omega$ eingesetzt werden und liefert eine Approximation $\mathcal{K}_{\Omega,\mathcal{H}}[\tilde{f}]$ für die Dirichlet-Randwertaufgabe (47,48).
Die Approximation von K durch eine Matrix $K_{apx} \in \mathcal{M}_{\mathcal{H},k}(T,Z)$ wurde im vorigen Abschnitt untersucht. Die approximative Inverse $K_{apx}^{\ominus 1}$ von K_{apx} erfüllt die Abschätzung

$$\begin{aligned} \|I - K_{apx}^{\ominus 1} \cdot K\|_2 &= \|I - K_{apx}^{\ominus 1} K_{apx} + K_{apx}^{\ominus 1}(K_{apx} - K)\|_2 \\ &\leq \|I - K_{apx}^{\ominus 1} K_{apx}\|_2 + \|K_{apx}^{\ominus 1}(K_{apx} - K)\|_2 \\ &\leq \|I - K_{apx}^{\ominus 1} K_{apx}\|_2 + \|K_{apx} - K\|_2 \|K_{apx}^{\ominus 1}\|_2. \end{aligned}$$

Der Inversionsfehler $\|I - K_{apx}^{\ominus 1} K_{apx}\|_2$ sollte also in der Größenordnung von $\|K_{apx} - K\|_2 \|K_{apx}^{\ominus 1}\|_2$ liegen. Wir berechnen die Inverse von K_T und $K_\mathcal{H}$ aus dem vorigen Abschnitt mit der Block-Gauß-Elimination (Abschnitt 4.5.1) in $\mathcal{M}_{\mathcal{H},k}$(T,Z) für zunehmenden Rang k, jeweils beginnend mit dem zur Approximation von K benutzten Rang. Die Ergebnisse sind in Abbildung 17 für $n \in \{2048, 8192\}$ und $\eta = 0.8$ zu sehen. Für die Bestapproximation sollte der Rang zur Inversion um eins höher als zur Diskretisierung gewählt werden, im Fall der Taylor-Approximation kann derselbe Rang verwendet werden. Bis zu dem durch die Approximation der Matrix K entstandenen Fehler $\delta_{apx} := \|K_{apx} - K\|_2 \|K_{apx}^{\ominus 1}\|_2$ lassen sich die Inversen offenbar als $\mathcal{H}$-Matrizen darstellen.
Um das Verhalten der Singulärwerte in den zulässigen Blöcken genauer zu sehen, sind für $n = 2048, m = 3, k = 20$ und $\eta = 0.8$ die Singulärwerte des Blockes mit dem größten k-ten Singulärwert in Abbildung 18 dargestellt.

n=2048 $\varepsilon_{\mathcal{H}}$	k	$k+1$	$k+2$	$\delta_{\mathcal{H}}$
m=1	$1.2{\cdot}10^{-0}$	$5.7{\cdot}10^{-1}$	$1.7{\cdot}10^{-1}$	$5.4{\cdot}10^{-1}$
m=2	$1.2{\cdot}10^{-2}$	$4.5{\cdot}10^{-3}$	$2.4{\cdot}10^{-3}$	$9.2{\cdot}10^{-3}$
m=3	$1.1{\cdot}10^{-5}$	$1.2{\cdot}10^{-5}$	$1.2{\cdot}10^{-5}$	$2.9{\cdot}10^{-5}$
m=4	$3.0{\cdot}10^{-8}$	$1.9{\cdot}10^{-8}$	$1.9{\cdot}10^{-8}$	$4.0{\cdot}10^{-8}$

n=2048 $\varepsilon_{\mathbf{T}}$	k	$k+1$	$k+2$	δ_{T}
m=1	$1.2{\cdot}10^{-0}$	$9.3{\cdot}10^{-1}$	$8.6{\cdot}10^{-1}$	$2.9{\cdot}10^{-0}$
m=2	$1.3{\cdot}10^{-1}$	$1.3{\cdot}10^{-1}$	$1.3{\cdot}10^{-1}$	$3.9{\cdot}10^{-1}$
m=3	$2.5{\cdot}10^{-2}$	$2.8{\cdot}10^{-2}$	$2.8{\cdot}10^{-2}$	$8.0{\cdot}10^{-2}$
m=4	$8.2{\cdot}10^{-3}$	$8.7{\cdot}10^{-3}$	$8.7{\cdot}10^{-3}$	$1.8{\cdot}10^{-2}$

n=8192 $\varepsilon_{\mathcal{H}}$	k	$k+1$	$k+2$	$\delta_{\mathcal{H}}$
m=1	$3.7{\cdot}10^{-0}$	$2.1{\cdot}10^{-0}$	$5.7{\cdot}10^{-1}$	$1.1{\cdot}10^{-0}$
m=2	$3.3{\cdot}10^{-2}$	$1.5{\cdot}10^{-2}$	$4.9{\cdot}10^{-3}$	$2.0{\cdot}10^{-2}$
m=3	$3.1{\cdot}10^{-5}$	$2.3{\cdot}10^{-5}$	$1.8{\cdot}10^{-5}$	$5.8{\cdot}10^{-5}$
m=4	$5.2{\cdot}10^{-8}$	$3.2{\cdot}10^{-8}$	$3.0{\cdot}10^{-8}$	$9.2{\cdot}10^{-8}$

n=8192 $\varepsilon_{\mathbf{T}}$	k	$k+1$	$k+2$	δ_{T}
m=1	$3.4{\cdot}10^{-0}$	$2.0{\cdot}10^{-0}$	$1.9{\cdot}10^{-0}$	$7.7{\cdot}10^{-0}$
m=2	$2.7{\cdot}10^{-1}$	$2.7{\cdot}10^{-1}$	$2.6{\cdot}10^{-1}$	$7.7{\cdot}10^{-1}$
m=3	$3.9{\cdot}10^{-2}$	$4.3{\cdot}10^{-2}$	$3.8{\cdot}10^{-2}$	$1.7{\cdot}10^{-1}$
m=4	$1.4{\cdot}10^{-2}$	$1.3{\cdot}10^{-2}$	$1.3{\cdot}10^{-2}$	$4.0{\cdot}10^{-2}$

Abbildung 17: Approximation der Inversen der Kollokationsmatrix K des Einfachschichtpotentials durch eine $\mathcal{H}$-Matrix $K_{apx}^{\ominus 1}$. Der durch die Diskretisierung entstandene Fehler ist $\delta_{apx} := \|K - K_{apx}\|_2 \|K_{apx}^{\ominus 1}\|_2$, der Inversionsfehler ist $\varepsilon_{apx} := \|I - K_{apx}^{\ominus 1} \cdot K\|_2$.

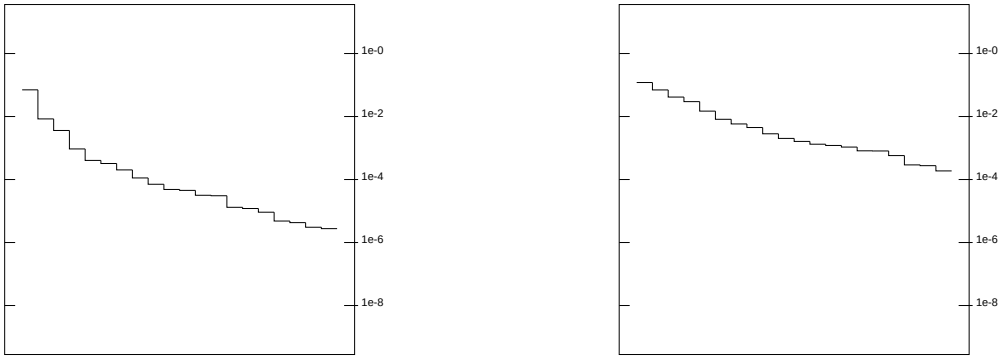


Abbildung 18: Ein Block aus der Approximation der Inversen der Kollokationsmatrix K des Einfachschichtpotentials durch eine $\mathcal{H}$-Matrix $K_{apx}^{\ominus 1}$, links für die Bestapproximation und rechts für die Taylor-Approximation. Aufgetragen sind die 20 Singulärwerte des Blockes in logarithmischer Skala, die Parameter sind $\eta = 0.8, m = 3, k = 20$ und $n = 2048$.

8.2. Partielle Differentialgleichungen

Die Approximation von Matrizen, die aus der Diskretisierung von Integralgleichungen mit asymptotisch glatten Kernfunktionen stammen, wurde im letzten Abschnitt numerisch getestet und ist durch die Theorie in Abschnitt 7.2 bereits hinreichend fundiert. Die Tests dienen als Referenz für die in diesem Abschnitt zu untersuchenden Inversen von linearen partiellen Differentialgleichungen.
Das erste Modellproblem wird die Poisson-Gleichung auf dem Einheitsquadrat sein. Für

die Singularitätenfunktion s ist die Standard-Zulässigkeitsbedingung Z_η zwar geeignet, aus Abschnitt 7.3 erhalten wir allerdings nur Aussagen für die erweiterte Standard-Zulässigkeitsbedingung $\bar{Z}_\eta$, die sich nicht mit der Block-Gauß-Elimination zur Inversion vereinbaren läßt. Die numerischen Tests werden zeigen, daß die Standard-Zulässigkeitsbedingung für die Approximationseigenschaft ausreicht.
Das Modellproblem wird anschließend in zwei Richtungen verallgemeinert:

- Das Einheitsquadrat wird durch einen Stern ersetzt und die Triangulation wird zu den Ecken hin stark verfeinert. Dadurch zeigt sich, welchen Einfluß die durch den Rand erzeugten Greenschen Funktionen auf die Darstellbarkeit haben.
- Der Laplace-Operator wird durch einen Operator mit nicht konstanten (springenden) Koeffizienten ersetzt. Die Singularitätenfunktion ist nicht bekannt, so daß auch die passende Zulässigkeitsbedingung nicht angegeben werden kann. Hier werden wir die adaptive Arithmetik aus Abschnitt 6 benutzen, um die „falsche“ Zulässigkeitsbedingung Z_η des Laplace-Operators durch lokal höheren Rang zu kompensieren.

8.2.1. Das Modellproblem: Poisson-Gleichung

Die Poisson-Gleichung

$$\begin{aligned} -\triangle u &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{auf } \Gamma := \partial\Omega \end{aligned}$$

soll für verschiedene (hinreichend glatte) rechte Seiten f auf dem Einheitsquadrat $\Omega := [0,1]^2$ gelöst werden (vgl. Problem 4.23). Eine Finite-Elemente-Diskretisierung wie in Problem 4.24 durch lokale Basisfunktionen zu einer regelmäßigen Triangulation von Ω führt zu der Aufgabe, die (schwachbesetzte) Matrix

$$A_{ij} := \int_\Omega \langle \nabla\phi_j, \nabla\phi_i \rangle, \qquad i,j \in I,$$

zu invertieren. Die Clusterung der Indexmenge $I = \{1, \ldots n\}$ erfolgt kardinalitätsbalanciert mit dem BSP-Algorithmus (Beispiel 3.10). Der $\mathcal{H}_\times$-Baum T ist der minimal aus dem Clusterbaum erzeugte $\mathcal{H}_\times$-Baum (Beispiel 3.26), wobei wir hier die Standard-Zulässigkeitsbedingung $Z_\eta, \eta = 0.8$, (Abschnitt 3.2.1) und $b_{min} := 32^2$ zugrundelegen. In Abbildung 19 (links) sind die relativen Fehler $\|I - A^{\ominus 1} A\|_2$ für die formatierte Inversion (mit der Block-Gauß-Elimination aus Abschnitt 4.5.1) in $\mathcal{M}_{\mathcal{H},k}(T, Z_\eta)$ für zunehmenden Rang k und Problemgröße n zu sehen. In den Tabellen 21 (links) und 22 (rechts) sind die Zeiten zur Inversion der (schwachbesetzten) Matrix und für die Durchführung einer Matrix-Vektor-Multiplikation (mit der nicht schwachbesetzten Inversen) angegeben. Der Speicherverbrauch für eine einzelne $\mathcal{H}$-Matrix ist in Tabelle 20 (rechts) wiedergegeben.
Zur Auflösung eines linearen Gleichungssystems der Form $Ax = b$ gibt es zwei unterschiedliche Ansätze. Zum einen kann die $\mathcal{H}$-Inverse $A^{\ominus 1}$ so genau bestimmt werden, daß

	Anzahl Freiheitsgrade n					
k	32^2	64^2	128^2	256^2	512^2	Quotient
1	$5.2\cdot10^{-1}$	$2.1\cdot10^{-0}$	$7.6\cdot10^{-0}$	$2.4\cdot10^{+1}$	$4.9\cdot10^{+1}$	2.0
2	$3.5\cdot10^{-2}$	$3.5\cdot10^{-1}$	$2.0\cdot10^{-0}$	$8.2\cdot10^{-0}$	$2.4\cdot10^{+1}$	2.9
3	$4.9\cdot10^{-3}$	$3.1\cdot10^{-2}$	$2.0\cdot10^{-1}$	$1.1\cdot10^{-0}$	$5.1\cdot10^{-0}$	4.6
4	$1.1\cdot10^{-3}$	$9.1\cdot10^{-3}$	$5.1\cdot10^{-2}$	$2.7\cdot10^{-1}$	$1.2\cdot10^{-0}$	4.4
5	$2.2\cdot10^{-4}$	$7.2\cdot10^{-4}$	$4.5\cdot10^{-3}$	$2.3\cdot10^{-2}$	$1.0\cdot10^{-1}$	4.3
6	$4.5\cdot10^{-5}$	$2.7\cdot10^{-4}$	$1.5\cdot10^{-3}$	$7.9\cdot10^{-3}$		
7	$6.4\cdot10^{-6}$	$3.1\cdot10^{-5}$	$1.7\cdot10^{-4}$	$1.1\cdot10^{-3}$		
8	$2.4\cdot10^{-6}$	$2.1\cdot10^{-5}$	$1.2\cdot10^{-4}$	$6.4\cdot10^{-4}$		
9	$6.8\cdot10^{-7}$	$4.6\cdot10^{-6}$	$2.0\cdot10^{-5}$	$9.6\cdot10^{-5}$		
10	$2.9\cdot10^{-7}$	$4.1\cdot10^{-6}$	$1.9\cdot10^{-5}$	$8.0\cdot10^{-5}$		
15	$8.4\cdot10^{-13}$	$5.4\cdot10^{-10}$	$2.2\cdot10^{-9}$	$2.8\cdot10^{-8}$		

Abbildung 19: Relativer Fehler $\|I-A^{\textcircled{-1}}A\|_2$ zur Approximation der Inversen des Laplace-Operators mit Rang k und n Freiheitsgraden auf dem Einheitsquadrat.

	Anzahl Freiheitsgrade n					
k	32^2	64^2	128^2	256^2	512^2	Quotient
1	$3.0\cdot10^{+0}$	$1.6\cdot10^{+1}$	$7.4\cdot10^{+1}$	$3.3\cdot10^{+2}$	$1.5\cdot10^{+3}$	4.5
2	$3.2\cdot10^{+0}$	$1.8\cdot10^{+1}$	$9.1\cdot10^{+1}$	$4.2\cdot10^{+2}$	$1.9\cdot10^{+3}$	4.5
3	$3.4\cdot10^{+0}$	$2.1\cdot10^{+1}$	$1.1\cdot10^{+2}$	$5.1\cdot10^{+2}$	$2.4\cdot10^{+3}$	4.7
4	$3.7\cdot10^{+0}$	$2.3\cdot10^{+1}$	$1.2\cdot10^{+2}$	$6.0\cdot10^{+2}$	$2.8\cdot10^{+3}$	4.7
5	$4.0\cdot10^{+0}$	$2.6\cdot10^{+1}$	$1.4\cdot10^{+2}$	$6.9\cdot10^{+2}$	$3.3\cdot10^{+3}$	4.8
6	$4.2\cdot10^{+0}$	$2.8\cdot10^{+1}$	$1.6\cdot10^{+2}$	$7.8\cdot10^{+2}$		
7	$4.5\cdot10^{+0}$	$3.1\cdot10^{+1}$	$1.7\cdot10^{+2}$	$8.7\cdot10^{+2}$		
8	$4.7\cdot10^{+0}$	$3.3\cdot10^{+1}$	$1.9\cdot10^{+2}$	$9.6\cdot10^{+2}$		
9	$5.0\cdot10^{+0}$	$3.6\cdot10^{+1}$	$2.0\cdot10^{+2}$	$1.0\cdot10^{+3}$		
10	$5.2\cdot10^{+0}$	$3.8\cdot10^{+1}$	$2.2\cdot10^{+2}$	$1.1\cdot10^{+3}$		
15	$6.4\cdot10^{+0}$	$5.1\cdot10^{+1}$	$3.0\cdot10^{+2}$	$1.6\cdot10^{+3}$		

Abbildung 20: Speicherverbrauch (in 1024^2 Byte) einer $n \times n$-$\mathcal{H}$-Matrix mit Rang k.

die Differenz der approximativen Lösung $A^{\textcircled{-1}}b$ zur Lösung $A^{-1}b$ in der Größenordnung des durch die Finite-Elemente-Methode entstandenen Diskretisierungsfehlers liegt. Zum anderen kann eine approximative Inverse $A^{\textcircled{-1}}$ als Iterationsmatrix eines linearen Iterationsverfahrens (siehe [8])

$$x^{i+1} := x^i - A^{\textcircled{-1}}(Ax^i - b), \qquad x^0 := 0,$$

verwendet werden. Die Konvergenzrate des Verfahrens ist mindestens $\|I - A^{\textcircled{-1}}A\|_2$, so daß im Fall $\|I - A^{\textcircled{-1}}A\|_2 < 0.1$ bereits nach wenigen Schritten die Maschinengenauigkeit

	Anzahl Freiheitsgrade n					
k	32^2	64^2	128^2	256^2	512^2	Quotient
1	$9.3 \cdot 10^{+0}$	$6.8 \cdot 10^{+1}$	$4.3 \cdot 10^{+2}$	$1.9 \cdot 10^{+3}$	$9.3 \cdot 10^{+3}$	4.9
2	$9.7 \cdot 10^{+0}$	$8.0 \cdot 10^{+1}$	$5.0 \cdot 10^{+2}$	$2.7 \cdot 10^{+3}$	$1.4 \cdot 10^{+4}$	5.2
3	$1.1 \cdot 10^{+1}$	$9.7 \cdot 10^{+1}$	$6.4 \cdot 10^{+2}$	$3.7 \cdot 10^{+3}$	$2.0 \cdot 10^{+4}$	5.4
4	$1.2 \cdot 10^{+1}$	$1.2 \cdot 10^{+2}$	$8.3 \cdot 10^{+2}$	$5.1 \cdot 10^{+3}$	$2.6 \cdot 10^{+4}$	5.1
5	$1.3 \cdot 10^{+1}$	$1.4 \cdot 10^{+2}$	$1.1 \cdot 10^{+3}$	$6.6 \cdot 10^{+3}$	$3.5 \cdot 10^{+4}$	5.3
6	$1.4 \cdot 10^{+1}$	$1.7 \cdot 10^{+2}$	$1.3 \cdot 10^{+3}$	$8.2 \cdot 10^{+3}$		
7	$1.6 \cdot 10^{+1}$	$2.0 \cdot 10^{+2}$	$1.6 \cdot 10^{+3}$	$9.9 \cdot 10^{+3}$		
8	$1.7 \cdot 10^{+1}$	$2.2 \cdot 10^{+2}$	$1.8 \cdot 10^{+3}$	$1.2 \cdot 10^{+4}$		
9	$1.8 \cdot 10^{+1}$	$2.4 \cdot 10^{+2}$	$2.0 \cdot 10^{+3}$	$1.3 \cdot 10^{+4}$		
10	$1.9 \cdot 10^{+1}$	$2.6 \cdot 10^{+2}$	$2.4 \cdot 10^{+3}$	$1.5 \cdot 10^{+4}$		
15	$2.1 \cdot 10^{+1}$	$3.2 \cdot 10^{+2}$	$3.0 \cdot 10^{+3}$	$2.1 \cdot 10^{+4}$		

Abbildung 21: Benötigte Zeit (in Sekunden) zur Berechnung der $\mathcal{H}$-Inversen des Laplace-Operators mit Rang k und n Freiheitsgraden auf dem Einheitsquadrat (CPU: UltraSPARC 250 MHz).

	Anzahl Freiheitsgrade n					
k	32^2	64^2	128^2	256^2	512^2	Quotient
1	$2.7 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	$6.9 \cdot 10^{-1}$	$3.0 \cdot 10^{+0}$	$1.4 \cdot 10^{+1}$	4.7
2	$2.9 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$	$8.1 \cdot 10^{-1}$	$3.9 \cdot 10^{+0}$	$1.7 \cdot 10^{+1}$	4.4
3	$3.2 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$9.7 \cdot 10^{-1}$	$4.7 \cdot 10^{+0}$	$2.1 \cdot 10^{+1}$	4.5
4	$3.4 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$1.1 \cdot 10^{-0}$	$5.7 \cdot 10^{+0}$	$2.4 \cdot 10^{+1}$	4.2
5	$3.5 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	$1.2 \cdot 10^{-0}$	$6.4 \cdot 10^{+0}$	$3.4 \cdot 10^{+1}$	5.3
6	$3.7 \cdot 10^{-2}$	$2.6 \cdot 10^{-1}$	$1.3 \cdot 10^{-0}$	$6.6 \cdot 10^{+0}$		
7	$3.9 \cdot 10^{-2}$	$2.7 \cdot 10^{-1}$	$1.4 \cdot 10^{-0}$	$7.3 \cdot 10^{+0}$		
8	$4.1 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$1.6 \cdot 10^{-0}$	$8.2 \cdot 10^{+0}$		
9	$4.3 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$	$1.7 \cdot 10^{-0}$	$8.4 \cdot 10^{+0}$		
10	$4.5 \cdot 10^{-2}$	$3.3 \cdot 10^{-1}$	$1.8 \cdot 10^{-0}$	$8.9 \cdot 10^{+0}$		
15	$5.0 \cdot 10^{-2}$	$3.9 \cdot 10^{-1}$	$2.3 \cdot 10^{-0}$	$1.2 \cdot 10^{+1}$		

Abbildung 22: Benötigte Zeit (in Sekunden) zur Durchführung einer Matrix-Vektor-Multiplikation für eine $n \times n$-$\mathcal{H}$-Matrix mit Rang k (CPU: UltraSPARC 250 MHz).

erreicht wird.

8.2.2. Ein nicht uniformes Gitter

Die Poisson-Gleichung aus dem vorigen Abschnitt wollen wir jetzt auf dem in Abbildung 23 dargestellten Gebiet lösen. Die Gittergenerierung erfolgt durch Verfeinerung zu den Eckpunkten und regelmäßige Verfeinerung. Der erste Verfeinerungsschritt ist in Abbil-

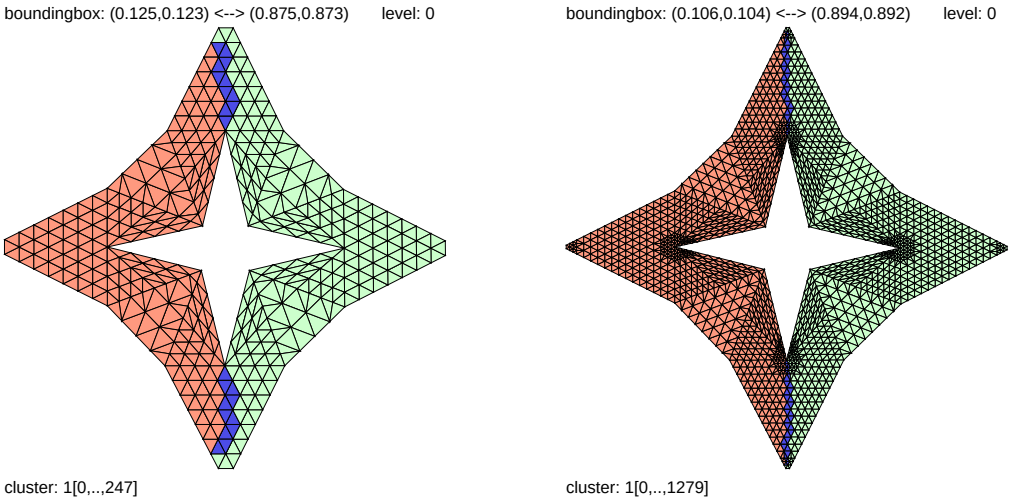


Abbildung 23: Die Triangulation des Sterngebietes ist links zu sehen. Das einmal lokal zu den 8 spitzen Ecken und anschließend global verfeinerte Gitter ist rechts zu sehen.

dung 23 (rechts) zu sehen. $\mathcal{H}$-Baum und $\mathcal{H}_\times$-Baum werden wie im vorigen Abschnitt erzeugt, der Parameter für die Zulässigkeit ist $\eta = 0.8$ und die minimale Blockgröße ist $b_{min} = 32^2$. in Abbildung 24 ist die Matrixpartition zu dem minimal aus dem BSP-$\mathcal{H}$-Baum der Indexmenge $I = \{1, \ldots, n\}$ erzeugten $\mathcal{H}_\times$-Baum zu sehen. Für $n \in \{1280, 5456, 22256, 89648\}$ Freiheitsgrade und Rang $k \in \{1, 3, 5, 10\}$ erhalten wir die folgenden relativen Fehler $\|I - A^{\ominus 1} A\|_2$ bei der Inversion der Matrix A:

	$n = 1280$	$n = 5456$	$n = 22256$	$n = 89648$
k=1	$2.6 \cdot 10^{-2}$	$3.3 \cdot 10^{-1}$	$2.1 \cdot 10^{-0}$	$8.0 \cdot 10^{-0}$
k=3	$4.5 \cdot 10^{-4}$	$9.6 \cdot 10^{-3}$	$3.2 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$
k=5	$1.4 \cdot 10^{-6}$	$1.8 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$
k=10	$8.7 \cdot 10^{-15}$	$2.8 \cdot 10^{-8}$	$9.7 \cdot 10^{-7}$	$1.1 \cdot 10^{-5}$

Offenbar läßt sich im Vergleich zum Problem auf dem Einheitsquadrat eine wesentlich bessere Approximation erzielen (vgl. Tabelle 19 (links)). Der Aufwand zur Inversion liegt für $n = 89648$ Freiheitsgrade und Rang $k = 1$ bei $3.1 \cdot 10^{+3}$ Sekunden, also nur geringfügig über dem entsprechenden Aufwand für das Problem auf dem Einheitsquadrat.

8.2.3. Eine Bilinearform mit nicht konstanten Koeffizienten

Wir betrachten die Differentialgleichung

$$\begin{aligned} -\operatorname{div} \sigma(x) \nabla u(x) &= f(x), \qquad x \in \Omega, \\ u &= 0 \qquad \text{auf } \Gamma := \partial\Omega, \end{aligned}$$

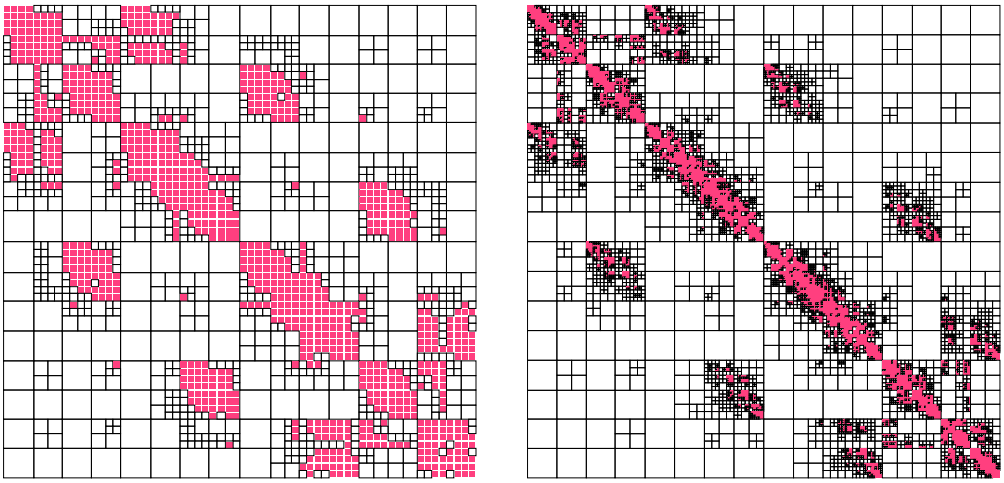
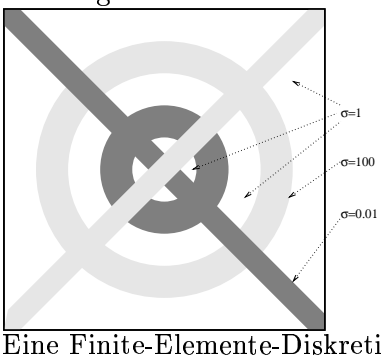


Abbildung 24: Die Matrixpartition für $n = 1280$ und $n = 5456$ Freiheitsgrade zum Sterngebiet mit lokaler Verfeinerung.

mit einer Funktion $\sigma : \mathbb{R}^2 \to \mathbb{R}_{>0}$ auf dem Gebiet $\Omega := [0,1]^2$, deren Funktionswerte dem folgenden Bild zu entnehmen sind:



$$\sigma(x,y) = \begin{cases} 0.01 & |x+y-1| < 0.05 \vee \\ & (0.1 \le \|(x,y)\| < 0.2) \wedge (|x-y| \ge 0.05) \\ 100 & |x-y| < 0.05 \vee \\ & (0.3 \le \|(x,y)\| < 0.4) \wedge (|x+y-1| \ge 0.05) \\ 1 & \text{sonst.} \end{cases}$$

Eine Finite-Elemente-Diskretisierung wie in Problem 4.24 durch lokale Basisfunktionen zu einer regelmäßigen Triangulation von Ω führt zu der Aufgabe, die (schwachbesetzte) Matrix

$$A_{ij} := \int_\Omega \sigma(x) \langle \nabla \phi_j(x), \nabla \phi_i(x) \rangle \, \mathrm{dx}, \qquad i,j \in I,$$

zu invertieren. Die Clusterung der Indexmenge $I = \{1, \ldots n\}$ und Partitionierung der Matrix erfolgt wie in Abschnitt 8.2.1 zur Standard-Zulässigkeitsbedingung Z_η. Die relativen Fehler $\|I - A^{\ominus 1} A\|_2$ zur Approximation der Inversen A^{-1} von A mit konstantem Rang k in jedem zulässigen Block sind in Abbildung 25 für $n \in \{32^2, 64^2, 128^2, 256^2\}$ aufgeführt. Offenbar erhält man eine Approximationsgüte von $\|I - A^{\ominus 1} A\|_2 \approx \frac{n}{10} 0.26^k$, also dasselbe Ergebnis wie im Fall eines Differentialoperators mit glatten Koeffizienten. Dieses Ergebnis beruht nicht darauf, daß σ in großen Teilen des Gebietes konstant ist, denn auch für stochastisch verteiltes σ erhält man (in diesem Modellfall) ähnliche Resultate.

	Anzahl Freiheitsgrade n			
k	32^2	64^2	128^2	256^2
1	$3.5 \cdot 10^{+1}$	$1.1 \cdot 10^{+2}$	$3.1 \cdot 10^{+2}$	$9.5 \cdot 10^{+2}$
2	$2.4 \cdot 10^{0}$	$1.7 \cdot 10^{+1}$	$1.3 \cdot 10^{+2}$	$4.3 \cdot 10^{+2}$
3	$6.0 \cdot 10^{-1}$	$3.9 \cdot 10^{0}$	$1.3 \cdot 10^{+1}$	$5.4 \cdot 10^{+1}$
4	$9.4 \cdot 10^{-2}$	$1.0 \cdot 10^{0}$	$3.4 \cdot 10^{0}$	$1.0 \cdot 10^{+1}$
5	$2.6 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$	$6.6 \cdot 10^{0}$
6	$1.1 \cdot 10^{-3}$	$7.7 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$1.3 \cdot 10^{0}$
7	$3.9 \cdot 10^{-5}$	$2.1 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$
8	$9.6 \cdot 10^{-6}$	$1.3 \cdot 10^{-3}$	$1.6 \cdot 10^{-2}$	$4.2 \cdot 10^{-2}$
9	$7.8 \cdot 10^{-6}$	$4.5 \cdot 10^{-4}$	$3.4 \cdot 10^{-3}$	$6.2 \cdot 10^{-3}$
10	$7.0 \cdot 10^{-7}$	$2.9 \cdot 10^{-4}$	$9.7 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$
15	$5.1 \cdot 10^{-12}$	$7.9 \cdot 10^{-9}$	$8.3 \cdot 10^{-7}$	$1.6 \cdot 10^{-6}$
20	$5.9 \cdot 10^{-12}$	$2.5 \cdot 10^{-11}$	$4.5 \cdot 10^{-9}$	$6.3 \cdot 10^{-9}$

Abbildung 25: Relativer Fehler $\|I - A^{\textcircled{-1}}A\|_2$ zur Approximation der Inversen mit n Freiheitsgraden und Rang k auf dem Einheitsquadrat mit springenden Koeffizienten $\sigma : [0,1]^2 \to \mathbb{R}_{>0}$.

Wir wollen nun die adaptive Inversion aus Abschnitt 6 verwenden, um die Inverse bis auf einen relativen Fehler $\|I - A^{\textcircled{-1}}A\|_2$ von ε zu bestimmen. Die Inversion führen wir wie in Algorithmus 6.11 durch, allerdings nur die erste Stufe. Das Ergebnis weist dann einen Fehler $\varepsilon_{real} := \|I - A^{\textcircled{-1}}A\|_2$ auf, da die Fehlerverstärkung in der ersten Stufe noch nicht berücksichtigt wurde. Die benötigte Zeit t zur adaptiven Inversion bis auf den Fehler ε_{real} vergleichen wir in Tabelle 26 mit der Zeit für die Inversion bei konstantem Rang k, wobei k zu einem gleichhohen Inversionsfehler führt. Die adaptive Inversion erreicht schon nach der ersten Stufe annähernd die vorgegebene Fehlerschranke und ist schneller als die Inversion mit konstantem Rang. Man beachte, daß im Laufe der Inversion ständig neuer Speicher für zunehmenden Rang angelegt werden muß und daher eine effiziente Speicherverwaltung erforderlich ist.

		Anzahl Freiheitsgrade							
		$n = 32^2$		$n = 64^2$		$n = 128^2$		$n = 256^2$	
ε		adaptiv	konst.	adaptiv	konst.	adaptiv	konst.	adaptiv	konst.
10^{-1}	t=	$1.1 \cdot 10^{+1}$	$1.1 \cdot 10^{+1}$	$9.7 \cdot 10^{+1}$	$1.4 \cdot 10^{+2}$	$7.6 \cdot 10^{+2}$	$1.1 \cdot 10^{+3}$	$5.9 \cdot 10^{+3}$	$5.1 \cdot 10^{+3}$
	ε_{real}=	$2.0 \cdot 10^{-1}$	k=3	$2.3 \cdot 10^{-1}$	k=5	$1.0 \cdot 10^{-0}$	k=5	$2.7 \cdot 10^{-0}$	k=4
10^{-3}	t=	$1.1 \cdot 10^{+1}$	$1.4 \cdot 10^{+1}$	$1.2 \cdot 10^{+2}$	$2.2 \cdot 10^{+2}$	$1.0 \cdot 10^{+3}$	$2.0 \cdot 10^{+3}$	$8.5 \cdot 10^{+3}$	$1.2 \cdot 10^{+4}$
	ε_{real}=	$1.1 \cdot 10^{-3}$	k=6	$3.2 \cdot 10^{-3}$	k=8	$3.3 \cdot 10^{-3}$	k=9	$8.8 \cdot 10^{-3}$	k=8
10^{-6}	t=	$1.3 \cdot 10^{+1}$	$1.8 \cdot 10^{+1}$	$1.7 \cdot 10^{+2}$	$3.1 \cdot 10^{+2}$	$1.6 \cdot 10^{+3}$	$2.7 \cdot 10^{+3}$	$1.4 \cdot 10^{+4}$	$3.5 \cdot 10^{+4}$
	ε_{real}=	$1.4 \cdot 10^{-6}$	k=9	$1.4 \cdot 10^{-6}$	k=13	$4.1 \cdot 10^{-6}$	k=13	$8.7 \cdot 10^{-6}$	k=14

Abbildung 26: Vergleich zwischen adaptiver Rangwahl und konstantem Rang.

8.3. Matrixgleichungen

In den Abschnitten 8.1 und 8.2 wurden Matrizen von spezieller Struktur durch $\mathcal{H}$-Matrizen approximiert. Bei den dort zugrundeliegenden Problemen ist man im wesentlichen an der Durchführbarkeit der Matrix-Vektor-Multiplikation interessiert, so daß zum Beispiel zur Inversion schnelle Gleichungslöser wie Mehrgitterverfahren mit linearem Aufwand der $\mathcal{H}$-Inversion überlegen sind. Da sich die $\mathcal{H}$-Inversion als äußerst robust erwiesen hat und keine spezielle Gitterstruktur (zum Beispiel eine Gitterhierarchie) voraussetzt, kann man sie als eine neue Methode zur Auflösung von Gleichungssystemen ansehen. Die Tatsache, daß die gesamte Inverse explizit vorliegt, wurde bisher kaum beachtet.
In diesem Abschnitt werden wir die $\mathcal{H}$-Arithmetik in einem Verfahren zur Auflösung von Matrixgleichungen verwenden. $\mathcal{H}$-Matrizen treten hier sowohl als Lösungen der Gleichungen als auch als Approximationen der Matrizen, die das Problem definieren, auf. Die Ljapunov-Gleichung ist ein Spezialfall der Matrix-Riccati-Gleichung. Sie besitzt zwar eine einfachere Struktur, wird allerdings nicht gesondert behandelt.

8.3.1. Linear-quadratisches Kontrollproblem

Das linear-quadratische Kontrollproblem (mit unendlichem Zeithorizont) besteht darin, ein $u \in L^2(0,\infty;\mathbb{R}^{n_u})$ zu finden, welches das Funktional

$$J(u,x_0) = \int_0^\infty y(t)^T y(t) + u(t)^T u(t)\,\mathrm{dt} \tag{49}$$

minimiert, wobei y durch die (schwache) Lösung $x \in L^2(0,\infty;\mathbb{R}^n)$ der Differentialgleichung

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \qquad t \in (0,\infty), \\ y(t) &= Cx(t), \\ x(0) &= x_0 \in \mathbb{R}^n \end{aligned}$$

definiert ist. Die Matrizen $B \in \mathbb{R}^{n,n_u}$ und $C \in \mathbb{R}^{n_y,n}$ sind beliebig vorgegeben, die Matrix $A \in \mathbb{R}^{n,n}$ sei eine Stabilitätsmatrix, d.h. das Spektrum von A sei in der negativen Halbachse $\{a+ib \in \mathbb{C} \mid a < 0\}$ enthalten.

Satz 8.1 *(Lösung durch Rückkopplungssteuerung)*
Die optimale (minimierende) Kontrolle u von (49) existiert und läßt sich in Rückkopplungsform

$$u(t) = -B^T X x(t), \quad t \in (0,\infty),$$

realisieren, wobei $X \in \mathbb{R}^{n,n}$ die in der Menge der symmetrischen positiv semidefiniten Matrizen eindeutige Lösung von

$$A^T X + XA - XFX + G = 0 \tag{50}$$

für die Matrizen $F := BB^T$ und $G := C^T C$ ist (zum Beweis siehe [20] und benutze: A negativ $\Rightarrow$ Stabilisierbarkeit und Entdeckbarkeit).

8.3.2. Modellproblem: Wärmeleitungsgleichung

Wir betrachten das linear-quadratische (ortsabhängige) Kontrollproblem der eindimensionalen Wärmeleitungsgleichung

$$\begin{array}{lcll} \frac{\partial}{\partial t}x(t,\xi) & = & \frac{\partial^2}{\partial \xi^2}x(t,\xi) + b(\xi)u(t), & \xi \in (0,1), t \in (0,\infty), \\ x(t,s) & = & 0, & s \in \{0,1\}, t \in (0,\infty), \\ x(0,\xi) & = & x_0(\xi), & \xi \in (0,1), \\ y(t) & = & \int_0^1 c(\xi)x(t,\xi)d\xi, & t \in (0,\infty), \end{array} \tag{51}$$

wobei $x_0, b, c \in L^2(0,1)$ und $n_u = n_y = 1$ ist (Kostenfunktional (49)). Die Funktionen b und c sind durch

$$b(\xi) := \begin{cases} 1 & \xi \in (0.2, 0.3) \\ 0 & \text{sonst} \end{cases},$$

$$c(\xi) := \begin{cases} 1 & \xi \in (0.2, 0.3) \\ 0 & \text{sonst} \end{cases}$$

gegeben.

Problem 8.2 *(FEM-Modellproblem)*
Für die Differentialgleichung (51) führen wir eine Finite-Elemente-Diskretisierung bzgl. der Ortsvariable ξ mit stückweise affinen Basisfunktionen $(\phi_i)_{i=1,\dots,n}$ auf einem regelmäßigen Gitter von $(0,1)$ mit n Freiheitsgraden (inneren Gitterpunkten) durch. Die entsprechenden diskreten Operatoren sind

$$\begin{array}{lcll} \tilde{A}_{ij}^{FEM} & := & -\int_0^1 D\phi_i(\xi)D\phi_j(\xi)d\xi & i,j = 1,\dots,n, \\ \tilde{B}_{i1}^{FEM} & := & \int_{0.2}^{0.3} \phi_i(\xi)d\xi, & i = 1,\dots,n, \\ E_{ij}^{FEM} & := & \int_0^1 \phi_i(\xi)\phi_j(\xi)d\xi & i,j = 1,\dots,n, \\ A^{FEM} & := & (E^{FEM})^{-1}\tilde{A}^{FEM}, & \\ B^{FEM} & := & (E^{FEM})^{-1}\tilde{B}^{FEM}, & \\ C_{1j}^{FEM} & := & \int_{0.2}^{0.3} \phi_j(\xi)d\xi, & i = 1,\dots,n. \end{array}$$

Mit den Matrizen $A^{FEM}, B^{FEM}, C^{FEM}$ liegt die (ortsdiskretisierte) Differentialgleichung (51) in der Form

$$\begin{array}{lcll} \dot{x}(t) & = & A^{FEM}x(t) + B^{FEM}u(t), & t \in (0,\infty), \\ y(t) & = & C^{FEM}x(t), & \\ x(0) & = & x_0 \in \mathbb{R}^n, & \end{array}$$

also wie im vorigen Abschnitt, vor (dieses Modellproblem ist [24] entnommen).

Problem 8.3 *(FD-Modellproblem)*
Für die Differentialgleichung (51) führen wir eine Finite-Differenzen-Diskretisierung bzgl. der Ortsvariable ξ mit stückweise affinen Basisfunktionen $(\phi_i)_{i=1,\ldots,n}$ auf einem regelmäßigen Gitter von $(0,1)$ mit n Freiheitsgraden (inneren Gitterpunkten) d.h. der Gitterweite $h := (n+1)^{-1}$ durch. Die entsprechenden diskreten Operatoren sind

$$\begin{aligned} A_{ij}^{FD} &:= \begin{cases} 2h^{-2} & i = j \\ -h^{-2} & |i-j| = 1 \\ 0 & \textit{sonst.} \end{cases} \qquad i,j = 1,\ldots,n, \\ B_{i1}^{FD} &:= \begin{cases} 1 & i \cdot h \in [0.2, 0.3] \\ 0 & \textit{sonst.} \end{cases} \qquad i = 1,\ldots,n, \\ C_{1j}^{FD} &:= \int_{0.2}^{0.3} \phi_j(\xi) d\xi, \qquad i = 1,\ldots,n. \end{aligned}$$

Die (ortsdiskretisierte) Differentialgleichung (51) ist in diesem Fall

$$\begin{aligned} \dot{x}(t) &= A^{FD}x(t) + B^{FD}u(t), \qquad t \in (0,\infty), \\ y(t) &= C^{FD}x(t), \\ x(0) &= x_0 \in \mathbb{R}^n. \end{aligned}$$

Der Vorteil gegenüber der Finite-Elemente-Diskretisierung ist der, daß die Inversion der Massematrix E^{FEM} entfällt.

Zur Lösung des Optimierungsproblems (49) ist die Gleichung (50) zu lösen, die im folgenden Abschnitt genauer untersucht wird.

8.3.3. Algebraische Matrix-Riccati-Gleichung

Die algebraische Matrix-Riccati-Gleichung

$$A^TX + XA - XFX + G = 0, \qquad A,F,G \in \mathbb{R}^{n,n},\ A < 0,\ F,G \geq 0, \tag{52}$$

wollen wir lösen. X^* sei eine symmetrisch positiv semidefinite Lösung von (52).

Satz 8.4 *(Lösungsdarstellung)*
Wir definieren

$$\begin{bmatrix} M & N \end{bmatrix} := \operatorname{sign}\left(\begin{bmatrix} A^T & G \\ F & -A \end{bmatrix} \right) - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

und

$$X := -(M^TM)^{-1}M^TN.$$

Dann ist X eine Lösung von (52).

Beweis: [23] ■

Die Matrix-Signum-Funktion ist die zur Funktion $a+ib \mapsto \operatorname{sign}(a)$ gehörende Matrixfunktion für Matrizen, deren Spektrum nicht die imaginäre Achse berührt. Die Berechnung der Matrix-Signum-Funktion im Zusammenhang formatierter $\mathcal{H}$-Arithmetik ist Gegenstand der folgenden Sätze.

Satz 8.5 *(Konvergenz der Newton-Iteration)*
Sei $S \in \mathbb{R}^{n,n}$ und das Spektrum von S berühre die imaginäre Achse nicht. Dann konvergiert die Newton-Iteration

$$\begin{aligned} S^{(0)} &:= S, \\ S^{(i+1)} &:= \frac{1}{2}(S^{(i)} + (S^{(i)})^{-1}), \quad i = 0, \ldots \end{aligned}$$

zur Auflösung von $f(X) := X^2 - I = 0$ gegen $\operatorname{sign}(S)$, und die Konvergenz ist lokal quadratisch.

Beweis: [23] ■

Lemma 8.6 *(Zur Fehlerverstärkung)*
Für die Iterierten $S^{(i)}$, $i \in \mathbb{N}$, der Newton-Iteration aus Satz 8.5 gilt, falls $S^{(i-1)}$ symmetrisch ist,

$$\|(S^{(i)})^{-1}\|_2 \leq 1.$$

Beweis: Sei $S := S^{(i-1)} \in \mathbb{R}^{n,n}$ in der Schur-Form

$$S = QDQ^T$$

mit Diagonalmatrix D und unitärem Q gegeben. Dann ist

$$S^{-1} = QD^{-1}Q^T$$

und somit

$$\begin{aligned} S^{(i)} &= \frac{1}{2}(S + S^{-1}) = Q\frac{1}{2}(D + D^{-1})Q^T, \\ (S^{(i)})^{-1} &= Q2(D + D^{-1})^{-1}Q^T. \end{aligned}$$

Es folgt

$$\begin{aligned} \|(S^{(i)})^{-1}\|_2 &= \|Q2(D + D^{-1})^{-1}Q^T\|_2 \\ &= 2\|(D + D^{-1})^{-1}\|_2 \\ &= 2 \max_{i=1,\ldots,n} \frac{1}{|D_{ii} + D_{ii}^{-1}|}. \end{aligned}$$

Da $D_{ii} \in \mathbb{R}$ vorausgesetzt war, ist $|D_{ii} + D_{ii}^{-1}| \geq 2$ und somit $2 \max_{i=1,\ldots,n} \frac{1}{|D_{ii}+D_{ii}^{-1}|} \leq 1$.
■

Bemerkung 8.7 *(Zur Fehlerverstärkung)*
In Lemma 8.6 haben wir für reell diagonalisierbare Matrizen gezeigt, daß die Norm der Inversen der Iterierten ab dem ersten Schritt höchstens 1 ist, so daß bei der Inversion keine Fehlerverstärkung auftritt. Im Allgemeinen erwarten wir lediglich, daß das Spektrum von S die imaginäre Achse nicht berührt. Je näher es an der imaginären Achse liegt, umso größer kann $\|(S^{(0)})^{-1}\|_2$ werden.

Satz 8.8 *(Bedingte Konvergenz bei formatierter Arithmetik)*
Sei T ein $\mathcal{H}_\times$-Baum, Z eine Zulässigkeitsbedingung und k eine Rangverteilung auf T. Sämtliche Operationen der formatierten $\mathcal{H}$-Arithmetik beziehen sich auf $\mathcal{M}_{\mathcal{H},k}(T,Z)$. Sei $S \in \mathbb{R}^{n,n}$ und das Spektrum von S berühre die imaginäre Achse nicht. Für die Iterierten der formatierten Newton-Iteration

$$\begin{aligned} \tilde{S}^{(0)} &:= (S)_{\mathcal{H}}, \\ \tilde{S}^{(i+1)} &:= \frac{1}{2}\left(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^1}\right), \quad i = 0, \ldots, i_{max} \end{aligned}$$

gelte

$$\begin{aligned} \|(\tilde{S}^{(i)})^{-1} - \tilde{S}^{(i)\ominus^1}\|_2 &\leq \delta, \\ \|(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^1}) - (\tilde{S}^{(i)} + \tilde{S}^{(i)\ominus^1})\|_2 &\leq \rho, \end{aligned}$$

wobei wir für die Folge

$$\begin{aligned} c_0 &:= \|\tilde{S}^{(0)} - S\|_2(\rho+\delta)^{-1}, \\ c_{i+1} &:= \frac{1}{2}(1 + c_i + c_i \frac{\|(S^{(i)})^{-1}\|_2^2}{1 - c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2}) \end{aligned}$$

die Bedingung

$$c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2 < 1, \qquad i = 0, \ldots, i_{max}, \tag{53}$$

($S^{(i)}$ wie in Satz 8.5) annehmen. Dann gilt die Abschätzung

$$\|\tilde{S}^{(i)} - S^{(i)}\|_2 \leq c_i(\rho+\delta), \qquad i = 0, \ldots, i_{max}.$$

Beweis: Wir beweisen die Behauptung per Induktion. Der Induktionsanfang ist nach Voraussetzung an c_0 erfüllt. Wir definieren

$$\begin{aligned} E^{(i)} &:= S^{(i)} - \tilde{S}^{(i)}, \\ D^{(i)} &:= \tilde{S}^{(i)\ominus^1} - (\tilde{S}^{(i)})^{-1}, \\ R^{(i)} &:= (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^1}) - (\tilde{S}^{(i)} + \tilde{S}^{(i)\ominus^1}). \end{aligned}$$

Es folgt

$$\begin{aligned} \tilde{S}^{(i+1)} &= \frac{1}{2}(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^1}) \\ &= \frac{1}{2}(\tilde{S}^{(i)} + \tilde{S}^{(i)\ominus^1}) + \frac{1}{2}R^{(i)} \\ &= \frac{1}{2}(S^{(i)} - E^{(i)} + (S^{(i)} - E^{(i)})^{-1} + D^{(i)}) + \frac{1}{2}R^{(i)} \\ &= \frac{1}{2}(S^{(i)} - E^{(i)} + (S^{(i)})^{-1}\sum_{\nu=0}^{\infty}(E^{(i)}(S^{(i)})^{-1})^\nu + D^{(i)}) + \frac{1}{2}R^{(i)} \\ &= S^{(i+1)} - \frac{1}{2}E^{(i)} + \frac{1}{2}(S^{(i)})^{-1}\sum_{\nu=1}^{\infty}(E^{(i)}(S^{(i)})^{-1})^\nu + \frac{1}{2}D^{(i)} + \frac{1}{2}R^{(i)}. \end{aligned}$$

Ab dem zweiten Schritt nutzen wir die Definition von c_{i+1} aus:

$$\begin{aligned}
\|\tilde{S}^{(i+1)} - S^{(i+1)}\|_2 &\overset{Ind.}{\leq} \frac{c_i}{2}(\rho+\delta) + \frac{1}{2}(\rho+\delta) \\
&\quad + \frac{c_i}{2}\|(S^{(i)})^{-1}\|_2(\rho+\delta)\|(S^{(i)})^{-1}\|_2 \sum_{\nu=0}^{\infty}(c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2)^{\nu} \\
&\leq \frac{c_i}{2}(\rho+\delta) + \frac{1}{2}(\rho+\delta) \\
&\quad + \frac{c_i}{2}\|(S^{(i)})^{-1}\|_2(\rho+\delta)\|(S^{(i)})^{-1}\|_2 \frac{1}{1-c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2} \\
&= \frac{1}{2}(1 + c_i + c_i \frac{\|(S^{(i)})^{-1}\|_2^2}{1-c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2})(\rho+\delta) \\
&= c_{i+1}(\rho+\delta).
\end{aligned}$$

■

Bemerkung 8.9 *(Voraussetzungen für die Konvergenz)*
In Satz 8.8 benötigen wir für die Konvergenz der Newton-Iteration die Darstellbarkeit der Iterierten, d.h. die formatierte Addition bzw. Inversion darf nur einen Fehler von ρ bzw. δ aufweisen. Dies kann man im Fall der adaptiven $\mathcal{H}$-Arithmetik aus Abschnitt 6 durch eine geeignete (automatische) Rangverteilung k erzielen. Die dritte Voraussetzung (53),

$$c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2 < 1,$$

ist schon wesentlich restriktiver. Solange die Norm $\|(S^{(i)})^{-1}\|_2$ der Iterierten wie in Lemma 8.6 klein ist (im Grenzfall ist $\|\mathrm{sign}(S)^{-1}\|_2 = \|\mathrm{sign}(S)\|_2 \geq 1$), so ist die Bedingung leichter zu erfüllen.
Die Folge der c_i verhält sich für den Fall

$$\frac{\|(S^{(i)})^{-1}\|_2^2}{1-c_i(\rho+\delta)\|(S^{(i)})^{-1}\|_2} \approx 1$$

wie $c_i = i/2$. Potentiell ist die Iteration also immer divergent (mehr zum Abbruchkriterium in Lemma 8.10 und Bemerkung 8.11). Im Fall $\|(S^{(i)})^{-1}\|_2 \gg 1$ verhalten sich die c_i auch bei extrem kleinen ρ, δ wie $c_i \approx O(\|(S^{(i)})^{-1}\|_2^i)$, d.h. schon nach wenigen Schritten entarten die Iterierten.
Eine Strategie zur Beschleunigung der Konvergenz der Newton-Iteration ist die Skalierung. Anstelle das Signum von S zu berechnen, wird das Signum von αS berechnet, da es unter Skalierung invariant ist. Die Iteration lautet nun

$$S^{(i+1)} := \frac{1}{2}(\alpha S^{(i)} + \alpha^{-1}(S^{(i)})^{-1}),$$

und die Norm der Inversen ist nur noch durch $\alpha^{-1}\|(S^{(i)})^{-1}\|_2$ beschränkt, so daß (53) nur für um den Faktor α^{-1} kleinere ρ, δ erfüllt ist und außerdem die Faktoren c_i schneller exponentiell anwachsen.

Für den Fall, daß S in $\mathcal{M}_{\mathcal{H},k}(T,Z)$ liegt, ist der Startfehler $c_0 = 0$, so daß die Fehlerverstärkung im ersten Schritt nicht relevant ist. Es empfiehlt sich also, im ersten Schritt die Skalierung zu verwenden, mit dem Ziel, die Extrema des Spektrums von S möglichst nahe an ± 1 heranzubringen. Die Norm des betragsmäßig kleinsten Spektralwertes λ_{min} schätzen wir durch

$$\lambda_{min} := \|S^{-1}\|_2^{-1},$$

die Norm des betragsmäßig größten Spektralwertes durch

$$\lambda_{max} := \|S\|_2.$$

Der Faktor α, für den die geschätzten Extrema $\lambda_{min}(\alpha S) \cdot \lambda_{max}(\alpha S) = 1$ erfüllen, ist

$$\alpha := \sqrt{\frac{\|S^{-1}\|_2}{\|S\|_2}} = \sqrt{\mathrm{cond}_2(S)}\|S\|_2^{-1}.$$

Lemma 8.10 *(Lokale Konvergenz bei formatierter Arithmetik)*
Sei T ein $\mathcal{H}_\times$-Baum, Z eine Zulässigkeitsbedingung und k eine Rangverteilung auf T. Sämtliche Operationen der formatierten $\mathcal{H}$-Arithmetik beziehen sich auf $\mathcal{M}_{\mathcal{H},k}(T,Z)$. Sei $S \in \mathbb{R}^{n,n}$ und das Spektrum von S berühre die imaginäre Achse nicht. Für die Iterierten der formatierten Newton-Iteration

$$\begin{aligned} \tilde{S}^{(0)} &:= (S)_{\mathcal{H}}, \\ \tilde{S}^{(i+1)} &:= \frac{1}{2}\left(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^{-1}}\right), \quad i = 0, \ldots \end{aligned}$$

gelte

$$\begin{aligned} \|(\tilde{S}^{(i)})^{-1} - \tilde{S}^{(i)\ominus^{-1}}\|_2 &\leq \delta, \\ \|(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus^{-1}}) - (\tilde{S}^{(i)} + \tilde{S}^{(i)\ominus^{-1}})\|_2 &\leq \rho, \end{aligned}$$

wobei wir für die Folge die Invertierbarkeit aller $\|\tilde{S}^{(i)}\|_2$ annehmen. Ferner seien für

$$\sigma := \max\left\{ 2\ ,\ \max_{i \in \mathbb{N}_0} \|\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1}\|_2 \right\}$$

die Abschätzungen

$$\begin{aligned} \|(\tilde{S}^{(0)})^2 - I\|_2 \quad =: q &\leq \frac{1}{4}, \\ \rho + \delta &\leq \frac{1}{4}\sigma^{-1}, \end{aligned}$$

erfüllt. Dann konvergiert die formatierte Newton-Iteration bis auf $\sigma(\rho + \delta)$ quadratisch gegen eine Wurzel der Identität:

$$\|(\tilde{S}^{(i)})^2 - I\|_2 \leq q^{2^{i+1}} + \sigma(\rho + \delta).$$

Beweis: Wir setzen

$$\begin{aligned} E^{(i)} &:= I - (\tilde{S}^{(i)})^2, \\ D^{(i)} &:= \tilde{S}^{(i)\ominus} - (\tilde{S}^{(i)})^{-1}, \\ R^{(i)} &:= (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)\ominus}) - (\tilde{S}^{(i)} + \tilde{S}^{(i)\ominus}). \end{aligned}$$

Die Behauptung wird per Induktion bewiesen, der Induktionsanfang ist nach Definition von q erfüllt. Es gilt

$$\begin{aligned} (\tilde{S}^{(i+1)})^2 &= (\frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1}) + \frac{1}{2}D^{(i)} + \frac{1}{2}R^{(i)})^2 \\ &= \frac{1}{4}(\tilde{S}^{(i)})^2 + \frac{1}{2}I + \frac{1}{4}(\tilde{S}^{(i)})^{-2} + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\ &\quad + \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}) \\ &= \frac{3}{4}I - \frac{1}{4}E^{(i)} + \frac{1}{4}\sum_{\nu=0}^{\infty}(E^{(i)})^\nu + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\ &\quad + \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}) \\ &= I + \frac{1}{4}\sum_{\nu=2}^{\infty}(E^{(i)})^\nu + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\ &\quad + \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}), \end{aligned}$$

$$\|(\tilde{S}^{(i+1)})^2 - I\|_2 \leq \frac{1}{4}\frac{\|E^{(i)}\|_2^2}{1 - \|E^{(i)}\|_2} + \frac{1}{4}(\rho + \delta)^2 + \frac{1}{2}\sigma(\rho + \delta).$$

Setzt man für $\|E^{(i)}\|_2$ die Induktionsvoraussetzung ein, so bleibt zu zeigen:

$$\frac{(q^{2^i+1} + \sigma(\rho + \delta))^2}{1 - q^{2^i+1} - \sigma(\rho + \delta)} \leq q^{2^{i+1}+1} + (\sigma - \frac{1}{4}(\rho + \delta) - \frac{1}{2}\sigma)(\rho + \delta).$$

Nach dem Ausmultiplizieren genügt es, die folgenden beiden Aussagen zu zeigen:

$$\begin{aligned} q^{2^{i+1}+2} &\leq q^{2^{i+1}+1} - q^{2^{i+1}+1}q^{2^i+1}, \\ (2q^{2^i+1} + \sigma^2(\rho + \delta))(\rho + \delta) &\leq (1 - q^{2^i+1} - \sigma(\rho + \delta))(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta))(\rho + \delta). \end{aligned}$$

Die erste Aussage ist für $q \leq \frac{1}{4}$ erfüllt, die zweite Aussage folgt der Reihe nach:

$$\begin{aligned} & (2q^{2^i+1} + \sigma^2(\rho + \delta))(\rho + \delta) \leq (1 - q^{2^i+1} - \sigma(\rho + \delta))(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta))(\rho + \delta) \\ \overset{q \leq \frac{1}{4}}{\Leftarrow} \quad & \frac{1}{8} + \sigma^2(\rho + \delta) \leq (\frac{15}{16} - \sigma(\rho + \delta))(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta)) \end{aligned}$$

$$
\begin{aligned}
&\Leftarrow \quad \frac{1}{8} + \sigma^2(\rho+\delta) \leq \frac{15}{32}\sigma - \frac{1}{2}\sigma^2(\rho+\delta) - \frac{15}{64}(\rho+\delta) + \frac{1}{4}\sigma(\rho+\delta)^2 \\
&\Leftarrow \quad \frac{1}{8} + \frac{3}{2}\sigma^2(\rho+\delta) + \frac{15}{64}(\rho+\delta) \leq \frac{15}{32}\sigma \\
&\overset{\sigma\geq 2}{\Leftarrow} \quad \frac{1}{16}\sigma + \frac{3}{2}\sigma^2(\rho+\delta) + \frac{15}{256}(\rho+\delta)\sigma^2 \leq \frac{15}{32}\sigma \\
&\Leftarrow \quad \frac{1}{16} + \frac{399}{256}\sigma(\rho+\delta) \leq \frac{15}{32} \\
&\Leftarrow \quad (\rho+\delta) \leq \frac{256}{399}\frac{13}{32}\sigma^{-1} \\
&\Leftarrow \quad (\rho+\delta) \leq \frac{1}{4}\sigma^{-1}.
\end{aligned}
$$

■

Bemerkung 8.11 *(Abbruchkriterium in der Newton-Iteration)*
Ein Abbruchkriterium für die formatierte Newton-Iteration (Satz 8.8) ist

$$\|I - (\tilde{S}^{(i)})^2\|_2 \leq \texttt{MACH\_EPS},$$

wobei `MACH_EPS` *die für die $\mathcal{H}$-Arithmetik relevante Maschinengenauigkeit angibt. Die Maschinengenauigkeit muß jedoch nicht vorab bekannt sein: Die Iteration ist nach Lemma 8.10 lokal quadratisch konvergent, so daß ein deutliches Abnehmen der Konvergenzrate ein zuverlässiges Zeichen für das Erreichen der $\mathcal{H}$-Arithmetik-Maschinengenauigkeit ist.*

Folgerung 8.12 *(Lösung der Matrix-Riccati-Gleichung)*
Die Lösung der algebraischen Matrix-Riccati-Gleichung (52) gliedert sich in drei Schritte:

1. *Für die beteiligten Matrizen A, F, G sind geeignete $\mathcal{H}$-Matrix-Strukturen zu finden, die sich aus dem zugrundeliegenden Problem ableiten. Für rein algebraische Probleme, die nicht aus der Diskretisierung einer (elliptischen) partiellen Differentialgleichung oder Integralgleichung stammen, läßt sich die $\mathcal{H}$-Matrix-Arithmetik im Allgemeinen nicht anwenden. Die $\mathcal{H}$-Arithmetik zielt jedoch speziell auf die bei Finite-Elemente-Diskretisierungen entstehenden großen Gleichungssysteme ab, bei kleinen Dimensionen $n < 1000$ lassen sich problemlos vollbesetzte Matrizen verwenden.*

2. *Sind die $\mathcal{H}$-Matrix-Strukturen $\mathcal{M}_{\mathcal{H},k_A}(T_A, Z_A)$ für A, $\mathcal{M}_{\mathcal{H},k_F}(T_F, Z_F)$ für F und $\mathcal{M}_{\mathcal{H},k_G}(T_G, Z_G)$ für G gewählt (die Rangverteilung kann offen bleiben und später adaptiv bestimmt werden), so ist die Matrix*

$$S := \begin{bmatrix} A^T & G \\ F & -A \end{bmatrix} \in \mathcal{M}_{\mathcal{H},k}(T, Z)$$

mit den entsprechenden aus $k_A, k_F, k_G, T_A, T_F, T_G, Z_A, Z_F, Z_G$ in kanonischer Weise zusammengesetzten Strukturen k, T, Z aufzustellen. $\operatorname{sign}(S)$ *wird mit der*

Newton-Iteration aus Satz 8.8 (Skalierung nur im ersten Schritt) berechnet, wobei die vorzugebenden Werte δ, ρ mindestens so klein gewählt werden, daß die Bedingung (53) erfüllt ist (hierfür benötigt man die Norm der Inversen $(S^{(i)})^{-1}$, die erst im Laufe der Iteration bekannt wird, gegebenenfalls muß man die Iteration ein zweites Mal mit korrigierten ρ, δ starten).

3. *Gemäß Satz 8.4 wird die Lösung X ausgerechnet. Die dabei auftretende Inversion $(M^T M)^{-1}$ kann wieder adaptiv erfolgen.*

Die Güte einer approximativen Lösung $\tilde{X}$ der Gleichung (52) läßt sich nicht ohne weiteres überprüfen. Bei der Inversion von Matrizen haben wir stets $\|I - A^{\textcircled{-1}}A\|_2$ als Güte für die approximative Inverse $A^{\textcircled{-1}}$ gewählt (siehe Abschnitt 8.2.1). Hier ist

$$\frac{\|A^{-1} - A^{\textcircled{-1}}\|_2}{\|A^{-1}\|_2} = \frac{\|(I - A^{\textcircled{-1}}A)A^{-1}\|_2}{\|A^{-1}\|_2} \leq \|I - A^{\textcircled{-1}}A\|_2,$$

also $\|I - A^{\textcircled{-1}}A\|_2$ eine obere Schranke für den relativen Fehler. Dadurch konnte die Berechnung von A^{-1} umgangen und die Güte mit fast-linearem Aufwand bestimmt werden. Für die Riccati-Gleichung wäre die Norm des Residuums

$$R(\tilde{X}) := A\tilde{X} + \tilde{X}A^T - \tilde{X}F\tilde{X} + G$$

eine leicht zu berechnende Größe. Wegen $R(X) = 0$ gilt

$$\begin{aligned}
R(\tilde{X}) &= R(\tilde{X}) - R(X) \\
&= A^T(\tilde{X} - X) + (\tilde{X} - X)A - \tilde{X}F\tilde{X} + XFX \\
&\approx A^T(\tilde{X} - X) + (\tilde{X} - X)A, \\
\|R(\tilde{X})\|_2 &\approx 2\|A\|_2\|\tilde{X} - X\|_2, \\
\frac{\|\tilde{X} - X\|_2}{\|X\|_2} &\approx \frac{\|\tilde{X} - X\|_2}{\|\tilde{X}\|_2} \\
&\approx \frac{\|R(\tilde{X})\|_2}{2\|A\|_2\|\tilde{X}\|_2}.
\end{aligned}$$

In der Praxis zeigt sich allerdings, daß für zunehmende Problemgröße n die Schätzung für den relativen Fehler bei kleinem Rang ($\rightarrow$ grobe Approximation) ungenau wird (die Terme höherer Ordnung sind dann nicht vernachlässigbar).

Folgerung 8.13 *(Lösung der Lyapunov-Gleichung)*
Die Lösung der Lyapunov-Gleichung

$$AX + XA^T + G = 0$$

für negativ definites A läßt sich unmittelbar auf die Lösung der algebraischen Matrix-Riccati-Gleichung (52) zurückführen. Die Struktur der Matrix

$$S = \begin{bmatrix} A^T & G \\ 0 & -A \end{bmatrix}$$

erlaubt jedoch eine vereinfachte Durchführung: Es gilt

$$\frac{1}{2}(S+S^{-1}) = \begin{bmatrix} \frac{1}{2}(A+A^{-1})^T & \frac{1}{2}(G+A^{-T}GA^{-1}) \\ 0 & -\frac{1}{2}(A+A^{-1}) \end{bmatrix},$$

so daß die Signum-Iteration zu

$$\begin{aligned} A^{(0)} &:= A, \\ G^{(0)} &:= G, \\ A^{(i+1)} &:= \frac{1}{2}(A^{(i)} \oplus A^{(i)\ominus}), \\ G^{(i+1)} &:= \frac{1}{2}(G^{(i)} \oplus (A^{(i)\ominus})^T G^{(i)} A^{(i)\ominus}) \end{aligned}$$

wird und in jedem Schritt nur noch eine Inversion von $A^{(i)}$ durchzuführen ist. Die Berechnung der Lösung X wird wegen

$$\operatorname{sign}(S) = \begin{bmatrix} -I & G^{(\infty)} \\ 0 & I \end{bmatrix}$$

zu $X := \frac{1}{2}G^{(\infty)}$. Die Güte einer approximativen Lösung $\tilde{X}$ kann man wie in Folgerung 8.12 durch

$$\frac{\|X-\tilde{X}\|_2}{\|X\|_2} \approx \frac{\|A\tilde{X}+\tilde{X}A^T+G\|_2}{2\|A\|_2\|\tilde{X}\|_2}$$

schätzen.

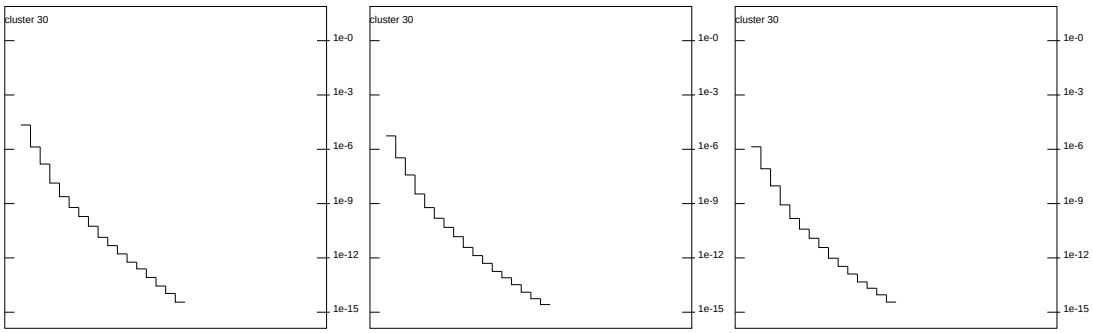


Abbildung 27: Singulärwerte der Lösung X für $n = 256, 1024, 4096$ Freiheitsgrade in logarithmischer Skala von $1.0 \cdot_{10} -^{15}$ bis $1.0 \cdot_{10} 0$.

Beispiel 8.14 *(Finite-Elemente-Lösung des Modellproblems)*
Zur Lösung von Problem 8.2 gehen wir wie in Folgerung 8.12 (Lösung der Matrix-Riccati-Gleichung) vor. Die Lösung X besitzt die in Abbildung 27 zu sehenden exponentiell abfallenden Singulärwerte. Wir wählen daher für F, G, X die Struktur der **R**k*-Matrix.*

Die Matrix $A = (E^{FEM})^{-1}\tilde{A}^{FEM}$ läßt sich als $\mathcal{H}$-Matrix in der Struktur zur Standard-Zulässigkeitsbedingung Z_η, $\eta = 0.8$, darstellen (siehe Abschnitt 8.2.1). Die benötigte Zeit zur Berechnung von $\tilde{X}$ und des Residuums $R(\tilde{X}) = A\tilde{X} + \tilde{X}A^T - \tilde{X}F\tilde{X} + G$ ist in der Abbildung 28 zu sehen. Verwendet wurde die adaptive $\mathcal{H}$-Arithmetik und $\delta = \rho = \varepsilon/\|A\|_2$, wobei die Inversion nicht zweistufig erfolgte, sondern noch den Faktor der Fehlerverstärkung aufweist. Dieses in [24] vorgeschlagene Modellproblem weist ganz erhebliche Skalierungsprobleme auf: Die Norm von A^{FEM} liegt in der Größenordnung $\|A^{FEM}\|_2 \approx 400n^2$, so daß $\|R(\tilde{X})\|_2$ deutlich größer als die Differenz $\|\tilde{X} - X\|_2$ zur tatsächlichen Lösung X ist.

		[24]	Anzahl Freiheitsgrade n						
ε		101	101	200	400	1000	2000	4000	10000
10^{-1}	t=	$2.1_{\cdot 10}{}^{+3}$	$1.6_{\cdot 10}{}^{+0}$	$4.9_{\cdot 10}{}^{+0}$	$2.1_{\cdot 10}{}^{+1}$	$9.2_{\cdot 10}{}^{+1}$	$3.2_{\cdot 10}{}^{+2}$	$1.2_{\cdot 10}{}^{+3}$	$1.3._{\cdot 10}{}^{+4}$
	ε^*	-	$5.7_{\cdot 10}{}^{-2}$	$6.1_{\cdot 10}{}^{-2}$	$2.9_{\cdot 10}{}^{-2}$	$2.6_{\cdot 10}{}^{-2}$	$2.2_{\cdot 10}{}^{-2}$	$1.5_{\cdot 10}{}^{-2}$	$1.1_{\cdot 10}{}^{-2}$
10^{-3}	t=	$2.1_{\cdot 10}{}^{+3}$	$2.2_{\cdot 10}{}^{+0}$	$6.3_{\cdot 10}{}^{+0}$	$2.9_{\cdot 10}{}^{+1}$	$1.2_{\cdot 10}{}^{+2}$	$4.3_{\cdot 10}{}^{+2}$	$1.5_{\cdot 10}{}^{+3}$	$1.6_{\cdot 10}{}^{+4}$
	ε^*	-	$4.5_{\cdot 10}{}^{-3}$	$4.4_{\cdot 10}{}^{-3}$	$6.0_{\cdot 10}{}^{-3}$	$4.2_{\cdot 10}{}^{-3}$	$2.9_{\cdot 10}{}^{-3}$	$2.4_{\cdot 10}{}^{-3}$	$1.7._{\cdot 10}{}^{-3}$
10^{-6}	t=	$2.1_{\cdot 10}{}^{+3}$	$3.3_{\cdot 10}{}^{+0}$	$1.2_{\cdot 10}{}^{+1}$	$4.3_{\cdot 10}{}^{+1}$	$1.8_{\cdot 10}{}^{+2}$	$5.9_{\cdot 10}{}^{+2}$	$2.1_{\cdot 10}{}^{+3}$	$2.1_{\cdot 10}{}^{+4}$
	ε^*	-	$6.5_{\cdot 10}{}^{-6}$	$1.3_{\cdot 10}{}^{-5}$	$3.5_{\cdot 10}{}^{-5}$	$1.2_{\cdot 10}{}^{-4}$	$6.2_{\cdot 10}{}^{-5}$	$6.1_{\cdot 10}{}^{-5}$	$1.1_{\cdot 10}{}^{-4}$

Abbildung 28: Die Zeit t zur Berechnung der approximativen Lösung der Riccati-Gleichung und der Fehler $\varepsilon^* := \|R(\tilde{X})\|_2$. Die Rechnungen wurden auf einer Sun Quasar mit 450 MHz durchgeführt. Zum Vergleich sind links die Zeiten zur Berechnung der Lösung mit dem Mehrgitter-Algorithmus aus [24] auf einer Sun Sparc 600 mit 40 MHz (ca. 20 mal langsamer) aufgeführt.

Beispiel 8.15 *(Finite-Differenzen-Lösung des Modellproblems)*
Zur Lösung von Problem 8.3 gehen wir wie im vorigen Beispiel 8.14 vor, d.h. für F, G, X wählen wir die Struktur der $\mathbf{R}k$-Matrix, für A eine $\mathcal{H}$-Matrix-Struktur zur Standard-Zulässigkeitsbedingung Z_η, $\eta = 0.8$. Wir wollen nun nicht mehr das Residuum $R(\tilde{X})$, sondern den relativen Fehler

$$\varepsilon := \|\tilde{X} - X\|_2/\|X\|_2$$

bestimmen.
Für größere Problemdimensionen n ist es kaum möglich, eine (bis auf Maschinengenauigkeit) exakte Lösung mit klassischen Verfahren und vollbesetzten Matrizen zu berechnen. Die Lösung der Matrix-Riccati-Gleichung (52) läßt sich mit dem Newton-Kleinman-Verfahren [19] auf wiederholtes Lösen einer verallgemeinerten Lyapunov-Gleichung zurückführen. Diese lineare Gleichung kann mit Mehrgitterverfahren wie in [24] gelöst

werden. Die Lösungsmatrix X muß in diesem Fall als vollbesetzte Matrix mit n^2 Unbekannten aufgefaßt werden, würde also für $n = 65536$ etwa 35 Gigabyte Speicher benötigen, der zur Zeit nicht zur Verfügung steht. Verwendet man statt eines Mehrgitterverfahrens einen einfacheren iterativen Löser, so verliert man die levelunabhängige Konvergenz, d.h. für große n wird hier der Aufwand zur Lösung zu groß.
Wir berechnen eine Referenzlösung X deshalb mit Hilfe der $\mathcal{H}$-Arithmetik für konstanten Rang $k_A = 13$. Auf den gröberen Stufen $n \leq 1024$ stimmt die Referenzlösung bis auf einen relativen Fehler von 10^{-12} mit der tatsächlichen (diskreten) Lösung überein.
Die Ergebnisse für konstanten Rang $k = k_A$ (keine adaptive Arithmetik) für die Struktur von A und Rang $k_F = k_G = k_X = 20$ für die $\mathbf{R}k$-Strukturen von F, G, X sind in Abbildung 29 zu sehen. In der Newton-Iteration zur Berechnung der sign*-Funktion werden ca. $\frac{3}{2} log_2(n)$ Schritte benötigt und in jedem Schritt ist eine Inversion mit Aufwand $O(n log_2(n))$ durchzuführen. Der Gesamtaufwand liegt entsprechend bei $O(n \log_2(n)^3)$, ist also wieder linear bis auf logarithmische Terme der Ordnung* 3.

Rang	Anzahl Freiheitsgrade n					
	101	256	1024	4096	16384	65536
k=1	$8.8 \cdot 10^{-3}$	$1.5 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	$2.5 \cdot 10^{-0}$	divergent	divergent
k=2	$2.4 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	$4.2 \cdot 10^{-4}$	$1.2 \cdot 10^{-3}$	$5.6 \cdot 10^{-4}$	$6.7 \cdot 10^{-4}$
k=3	$6.8 \cdot 10^{-6}$	$1.2 \cdot 10^{-5}$	$1.3 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$2.3 \cdot 10^{-5}$	$3.9 \cdot 10^{-5}$
k=4	$7.7 \cdot 10^{-8}$	$9.1 \cdot 10^{-8}$	$1.1 \cdot 10^{-7}$	$1.0 \cdot 10^{-6}$	$1.8 \cdot 10^{-6}$	$6.2 \cdot 10^{-7}$
k=5	$4.6 \cdot 10^{-9}$	$4.6 \cdot 10^{-9}$	$1.1 \cdot 10^{-8}$	$1.5 \cdot 10^{-8}$	$3.0 \cdot 10^{-8}$	$3.1 \cdot 10^{-8}$
k=6	$1.9 \cdot 10^{-10}$	$3.7 \cdot 10^{-10}$	$2.4 \cdot 10^{-10}$	$4.9 \cdot 10^{-10}$	$5.9 \cdot 10^{-10}$	$1.7 \cdot 10^{-9}$
Iterat.	12	14	17	20	23	26
Zeit(k=2)	2.2	8.5	67	462	3033	18263

Abbildung 29: Der relative Fehler $\varepsilon := \|\tilde{X} - X\|_2 / \|X\|_2$ für zunehmenden Rang k und n Freiheitsgrade. In der letzten Zeile ist die Zeit in Sekunden zur Berechnung der Lösung im Fall $k = 2$ auf einer Sun Quasar mit 450 MHz angegeben, in der vorletzten Zeile die Anzahl der Iterationsschritte in der Newton-Iteration.

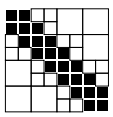
Fazit

Die Untersuchungen zur Durchführung der $\mathcal{H}$-Arithmetik sind für die grundlegenden Operationen „Addition“ und „Multiplikation“ im Fall von $\mathcal{H}_\times$-Bäumen, die aus zueinander passenden $\mathcal{H}$-Bäumen gebildet wurden, abgeschlossen. Die formatierten Operationen lassen sich in beiden Fällen als Hintereinanderausführung der exakten Operation und einer orthogonalen Projektion auf die Menge der $\mathcal{H}$-Matrizen darstellen.
Für die Inverse einer $\mathcal{H}$-Matrix läßt sich keine brauchbare Darstellungsformel angeben. Die über die Block-Gauß-Elimination gewonnene $\mathcal{H}$-Inverse wird rekursiv definiert und benötigt zusätzliche Voraussetzungen an die zu invertierende Matrix, erweist sich aber in der Praxis als sehr robust.
Die Abschätzung der Komplexität der $\mathcal{H}$-Arithmetik teilt sich in zwei Abschnitte auf:

1. Für die $\mathcal{H}_\times$-Bäume werden die Eigenschaften *schwachbesetzt* und *fast idempotent* nachgewiesen. Hier sind die Beweise auf die Standard-Zulässigkeitsbedingung beschränkt und müssen für andere konkrete Zulässigkeitsbedingungen neu geführt werden. Insbesondere die Clusterung der Indexmenge, die hier mit dem BSP-Algorithmus erfolgte, ist für allgemeine Zulässigkeitsbedingungen anders durchzuführen.

2. Für $\mathcal{H}$-Matrizen zu *schwachbesetzten* und *fast idempotenten* $\mathcal{H}$-Bäumen werden die Komplexitätsabschätzungen bewiesen. Der Einfluß der Konstanten aus der Schwachbesetztheit und Fast-Idempotenz ist hier explizit zu sehen.

Für die Anwendbarkeit der $\mathcal{H}$-Arithmetik benötigt man, wegen der durch die approximative Arithmetik entstehenden Fehler, eine genaue Fehleranalyse. Für einige Verfahren liegt diese bereits vor oder kann entsprechend ergänzt werden, für andere Verfahren sollte man die Fehleranalyse unter Berücksichtigung der Eigenschaften der $\mathcal{H}$-Approximation führen.
Ein offenes und sehr schwieriges Problem ist die Frage nach der Approximierbarkeit einer Matrix durch eine $\mathcal{H}$-Matrix. Eine erste Analyse wurde für die Inversen von stark elliptischen Differentialoperatoren bei glatten Rändern präsentiert. Selbst in diesem sehr restriktiven Fall konnte die Approximierbarkeit nicht in der Weise gezeigt werden, wie man es für Integraloperatoren mit asymptotisch glatten Kernfunktionen kennt. Die ‘Konstanten’ in den Abschätzungen zur asymptotischen Glattheit der kompensierenden Kernfunktion (bzw. Greenschen Funktion) sind von der Ableitungsordnung abhängig, so daß man keine Aussagen über die Asymptotik für zunehmenden Rang (Entwicklungsordnung) erhält.
Die numerischen Tests geben Anlaß zu der Hoffnung, daß sich die Approximationseigenschaft wenigstens in Modellfällen beweisen läßt.



A. Implementierung von $\mathcal{H}$-Matrizen in der Programmiersprache C

A.1. Vorwort

Die Umsetzung mathematisch formulierter Algorithmen auf Rechnern ist eine nicht wohldefinierte Funktion. Sehr viele wichtige Details sind von konkreten Rechnerarchitekturen, dem aktuellen Stand einer Programmiersprache oder zur Verfügung stehenden Bibliotheken abhängig. Eine Möglichkeit, diesen Problemen aus dem Weg zu gehen, ist die Verwendung sogenannten „Pseudocodes“, der eine nicht existierende Programmiersprache darstellt, die sich nicht um Details kümmert. Baut ein Algorithmus auf andere Algorithmen auf, so kann man auf Literatur verweisen, in denen diese Algorithmen in „Pseudocode“ beschrieben werden. Diese Form der Notation ist lediglich eine andere Methode, Algorithmen mathematisch zu formulieren. Die Umsetzung auf Rechnern wird dabei nicht berücksichtigt.
Wir wählen die konkrete Programmiersprache C zur Implementierung, da sie für die meisten Rechnerarchitekturen vorhanden ist und dem American National Standards Institute sowie International Organization for Standardization (ANSI/ISO) C Standard genügt (siehe z.B. [31] für genauere Verweise und [18] für eine Sprachbeschreibung). Rechnerspezifische Techniken (Cache-Ausnutzung, Speicherverwaltung, Parallelisierung) sind dann vom Compiler und etwaigen Bibliotheken abhängig und sollen hier nicht weiter berücksichtigt werden.
Die grundlegenden Methoden der Linearen Algebra werden unter

http://www.netlib.org/lapack/ (LAPACK = Linear Algebra Package)
http://www.netlib.org/blas/ (BLAS = Basic Linear Algebra Subroutines)

als Sourcecode in FORTRAN zusammen mit einer ausführlichen Dokumentation bereitgestellt. Ebenso wie C ist auch FORTRAN eine standardisierte (ANSI/ISO) Sprache und für die meisten Rechnerarchitekturen vorhanden. Die Funktionen aus den BLAS - und LAPACK - Bibliotheken lassen sich von C aus aufrufen. Die hier benutzten Funktionen von BLAS und LAPACK sind

`dcopy_`: Den Inhalt eines Arrays in ein anderes Array kopieren.
`dscal_`: Alle Elemente eines Arrays mit einer Zahl multiplizieren.
`daxpy_`: Zu einem Array ein um einen Faktor skaliertes Array addiern.

`dgemv_`: Matrix-Vektor-Multiplikation.
`dgemm_`: Matrix-Matrix-Multiplikation.
`dgeqrf_`: Berechnung einer QR-Zerlegung, Q in komprimierter Darstellung.
`dorgqr_`: Die komprimierte Darstellung nach `dgeqrf_` auflösen.
`dgetrf_`: Berechnung einer LU-Zerlegung.
`dgetri_`: Berechnung der Inversen nach einer LU-Zerlegung.
`dgesvd_`: Berechnung einer Singulärwertzerlegung.

Die genauen Parameter und Rückgabewerte entnimmt man der Dokumentation von BLAS bzw. LAPACK auf der Webseite oder [1]. Um den Aufruf der Prozeduren von LAPACK und BLAS etwas zu vereinfachen, werden die folgenden Konstanten definiert.

Implementierung A.1 *(Konstanten für* BLAS *und* LAPACK *)*

```
int eins_[1] = {1};
double deins_[1] = {1.0};
double dnull_[1] = {0.0};
double meins_[1] = {-1.0};
char ttrans[1] = {'t'};
char ntrans[1] = {'n'};
```

Eine $\mathcal{H}$-Matrix M basiert auf einem $\mathcal{H}$-Baum T, einer Rangverteilung k, einer Zulässigkeitsbedingung Z und den Daten der Matrix in den Blättern des Baumes T. In den nicht zulässigen Blättern $b \in \mathcal{L}^-(T)$ ist $M|_b$ in Form von $|b|$ Gleitkommazahlen gegeben. Diese werden in der Struktur *Full-Matrix* als ein Array von Zahlen gespeichert ($\rightarrow$ Implementierung A.2). In den zulässigen Blättern $b = r \times s \in \mathcal{L}^+(T)$ ist $M|_b$ in Form einer $\mathbf{R}k(b)$-Matrix $M|_b = AB^T$ mit $A \in \mathbb{R}^{r,k}$, $B \in \mathbb{R}^{s,k}$ gegeben. Die Matrizen A und B werden als Array von Zahlen gespeichert ($\rightarrow$ Implementierung A.4).
Ein Knoten b des Baumes T entspricht der strukturierten Beschreibung einer Teilmatrix $M|_b$ von M. Ist b kein Blatt des Baumes, so beschreiben die Söhne von b eine Aufteilung von $M|_b$ in Untermatrizen und $M|_b$ ist eine „Übermatrix“ (engl.: supermatrix). Wir speichern zu jedem Knoten des Baumes die Verweise auf die Söhne, die Größe der korrespondierenden Matrix und gegebenenfalls Zeiger zu einer $\mathbf{R}k$-Matrix oder Full-Matrix, falls b ein Blatt des Baumes ist, in der Struktur *Supermatrix* ($\rightarrow$ Implementierung A.9). Die Supermatrix zur Wurzel des Baumes ist dann die eigentliche $\mathcal{H}$-Matrix und beinhaltet indirekt die Rangverteilung, Zulässigkeitsbedingung und Baumstruktur von T. Ist T aus zwei $\mathcal{H}$-Bäumen T_I, T_J gebildet, so gehen die Strukturen von T_I, T_J verloren und lassen sich auch nicht notwendig aus der Supermatrix rekonstruieren. Fügt man in jeder Supermatrix einen Zeiger auf die korrespondierenden Knoten von T_I, T_J ein, so bleiben alle Informationen zugänglich.

A.2. Full-Matrix und Rk-Matrix

Implementierung A.2 *(Datenstruktur $n \times m$-Full-Matrix)*
Die Elemente der $n \times m$-Matrix

$$F = \begin{bmatrix} F_{11} & \dots & F_{1m} \\ \vdots & \ddots & \vdots \\ F_{n1} & \dots & F_{nm} \end{bmatrix}$$

werden in dem Array `e` *in der Reihenfolge* $F_{11}, \dots, F_{n1} F_{12}, \dots, F_{n2}, \dots, F_{nm}$ *gespeichert (dies entspricht der Konvention von* LAPACK *).*

```
typedef struct _fullmatrix fullmatrix;
typedef fullmatrix *pfullmatrix;
struct _fullmatrix {
  int n;
  int m;
  double* e;          /* eintraege spaltenweise */
};
```

Das Anlegen und Freigeben einer Full-Matrix (Speicher anfordern) erfolgt durch

```
pfullmatrix new_fullmatrix(int n_new, int m_new);
void del_fullmatrix(pfullmatrix f);
```

Implementierung A.3 *(Auswertung von Full-Matrizen)*
Die Auswertung einer n×m*-Full-Matrix* f *erfolgt durch die* LAPACK *-Funktion* dgemv_.

```
void eval_fullmatrix(pfullmatrix f, double* v, double* w){
  dgemv_(ntrans,&f->n,&f->m,deins_,f->e,&f->n,v,eins_,dnull_,w,eins_);
}
```

Analog werden die Funktionen zur Auswertung der transponierten Matrix und zum Aufaddieren von f·v *auf einen Vektor* w *definiert:*

```
void addeval_fullmatrix(pfullmatrix f, double* v, double* w);
void evaltrans_fullmatrix(pfullmatrix f, double* v, double* w);
void addevaltrans_fullmatrix(pfullmatrix f, double* v, double* w);
```

Implementierung A.4 *(Datenstruktur $n \times m$-*$\mathbf{R}k$*-Matrix)*
Von der $\mathbf{R}k$*-Matrix* $R = AB^T$ *wird* $A \in \mathbb{R}^{\mathtt{n},\mathtt{kt}}$ *spaltenweise in* a *und* $B \in \mathbb{R}^{\mathtt{m},\mathtt{kt}}$ *spaltenweise in* b *gespeichert.* kt *ist die Anzahl der Vektoren in der Darstellung (6). Der in diesem Block maximal erlaubte Rang ist* k.

```
typedef struct _rkmatrix rkmatrix;
typedef rkmatrix *prkmatrix;
struct _rkmatrix {
  int k;           /* maximaler Rang      */
  int kt;          /* tatsaechlicher Rang */
  int n;
  int m;
  double* a;
  double* b;       /* R = abˆT            */
};
```

Das Anlegen und Freigeben einer $\mathbf{R}k$*-Matrix (Speicher anfordern) erfolgt durch*

```
prkmatrix new_rkmatrix(int k_new, int n_new, int m_new);
void del_rkmatrix(prkmatrix r);
```

Implementierung A.5 *(Auswertung von* $\mathbf{R}k$*-Matrizen)*
Zur Auswertung einer `n`×`m`-$\mathbf{R}k$*-Matrix* `r` *wird ein Vektor* `v_tmp` *zur Zwischenspeicherung benötigt. Die eigentliche Auswertung von* `a` *und* `b` *erfolgt durch die* LAPACK *-Funktion* `dgemv_`.

```
void eval_rkmatrix(prkmatrix r, double* v, double* w){  /* w=r*v */
  if(r->kt>0){
    dgemv_(ttrans,&r->m,&r->kt,deins_,r->b,&r->m,v,eins_,dnull_,
           rkmatrix_v_tmp,eins_);
    dgemv_(ntrans,&r->n,&r->kt,deins_,r->a,&r->n,rkmatrix_v_tmp,eins_,
           dnull_,w,eins_);
  } else {
    dscal_(&r->n,dnull_,w,eins_);
  }
}
```

Analog werden die Funktionen zur Auswertung der transponierten Matrix und zum Aufaddieren von `r`·`v` *auf einen Vektor* `w` *definiert:*

```
void addeval_rkmatrix(prkmatrix r, double* v, double* w);
void evaltrans_rkmatrix(prkmatrix r, double* v, double* w);
void addevaltrans_rkmatrix(prkmatrix r, double* v, double* w);
```

Implementierung A.6 *(Konvertieren einer Full-Matrix in eine* $\mathbf{R}k$*-Matrix)*
Die Konvertierung einer Full-Matrix F' *in eine* $\mathbf{R}k(n,m)$*-Matrix* R *erfolgt mit Hilfe der Singulärwertzerlegung* `dgesvd_` *aus* LAPACK *. Für spätere Zwecke ist es günstig,* $R := R \oplus F'$ *anstelle von* $R := F'_{\mathcal{H}}$ *zu implementieren. Durch vorheriges Initialisieren von* R *mit Null erhält man die Konvertierung. Außerdem sehen wir vor, daß* F' *eine Untermatrix einer größeren Full-Matrix* F *ist:*

$$F = \begin{bmatrix} F_{1,1} & \dots & F_{1,m_F} \\ \vdots & \ddots & \vdots \\ F_{n_F,1} & \dots & F_{n_F,m_F} \end{bmatrix}, \quad F' = \begin{bmatrix} F_{\mathtt{nof}+1,\mathtt{mof}+1} & \dots & F_{\mathtt{nof}+1,\mathtt{mof}+\mathtt{m}} \\ \vdots & \ddots & \vdots \\ F_{\mathtt{nof}+\mathtt{n},\mathtt{mof}+1} & \dots & F_{\mathtt{nof}+n,\mathtt{mof}+\mathtt{m}} \end{bmatrix}.$$

```
void add_full2rkmatrix(pfullmatrix f, prkmatrix rt, int  nof, int mof){
  int n = rt->n, m = rt->m, nm = n*m;
  int i,j,k,info,lwork = 6*n*m;
  double* u = (double*) calloc(n*n,sizeof(double));
  double* v = (double*) calloc(m*m,sizeof(double));
  double* s = (double*) calloc(n,sizeof(double));
  double* tmp = (double*) calloc(nm,sizeof(double));
  double* work = (double*) calloc(lwork,sizeof(double));
  double sing_val;
  if(u==0x0 || v==0x0 || s==0x0 || tmp==0x0 || work==0x0){
    fprintf(stderr,"Speicher voll in add_full2rkmatrix.\n");
```

```
    exit(1);
  }
  if(f->n >= n+nof && f->m >= m+moff){
    if(nof==0 && mof==0 && f->n==n && f->m==m){
      dcopy_(&nm,f->e,eins_,tmp,eins_);
    } else {
      for(i=0; i<m; i++)
        dcopy_(&n,&f->e[(i+mof)*f->n + nof],eins_,&tmp[i*n],eins_);
    }
  } else {
    fprintf(stderr,"Matrizen inkompatibel in add_full2rkmatrix.\n");
    exit(2);
  }
  if(rt->kt>0){
    dgemm_(ntrans,ttrans,&n,&m,&rt->kt,deins_,
           rt->a,&n,rt->b,&m,deins_,
           tmp,&n);
  }
  dgesvd_("S","S",&n,&m,tmp,&n,s,u,&n,v,&m,work,&lwork,&info);
  if(info!=0){
    fprintf(stderr,"Fehler in add_full2rkmatrix, info=%d.\n",info);
    exit(3);
  }
  rt->kt=0;
  nm = n*rt->k; dscal_(&nm,dnull_,rt->a,eins_);
  nm = m*rt->k; dscal_(&nm,dnull_,rt->b,eins_);
  for(i=0; i<rt->k && i<n && i<m; i++){
    sing_val = sqrt(s[i]);
    daxpy_(&n,&sing_val,&u[i*n],eins_,&rt->a[i*n],eins_);
    daxpy_(&m,&sing_val,&v[i],&m,&rt->b[i*m],eins_);
    rt->kt = i+1;
  }
  free(work);
  free(tmp);
  free(s);
  free(v);
  free(u);
}
```

Implementierung A.7 *(Kürzen von* $\mathbf{R}k$*-Matrizen)*
Das Kürzen einer $\mathbf{R}k$*-Matrix auf einen niedrigeren Rang* $k' \leq k$ *erfolgt wie in Algorithmus 2.12. Wir sehen jedoch folgende Verallgemeinerung vor: Zur* $\mathbf{R}k$*-Matrix* `rt` *werden* `anz` $\mathbf{R}k$*-Matrizen* `rk_r[0]`, ..., `rk_r[anz-1]` *addiert, jede der Matrizen* `rk_r[i]` *ist Teil einer größeren Matrix mit dem ersten Index* `(rk_no[i],rk_mo[i])`*. Für den Fall* `anz=1`

entspricht dies der Konvertierung einer $\mathbf{R}k$*-Matrix in einer* $\mathbf{R}k'$*-Matrix.*

```
void addparts2rkmatrix(prkmatrix rt, int anz, int* rk_no,
                       int* rk_mo, prkmatrix* rk_r){
  int i,j,k,lwork,info,kmax=rt->kt;
  int n = rt->n;
  int m = rt->m;
  pfullmatrix f;
  double *atmp,*btmp,*rarb,*u,*v,*qr_work,*tau1,*tau2,*sigma,*rarabwork;
  for(i=0; i<anz; i++) kmax += rk_r[i]->kt;        /* rang der matrix */
  if(kmax==0){
    rt->kt=0;
    return;
  }
  if(kmax>=n || kmax>=m){                 /* rang gross -> fullmatrix */
    f = new_fullmatrix(n,m);
    for(i=0; i<anz; i++){
      dgemm_(ntrans,ttrans,&rk_r[i]->n,&rk_r[i]->m,&rk_r[i]->kt,
             deins_,rk_r[i]->a,&rk_r[i]->n,rk_r[i]->b,&rk_r[i]->m,
             deins_,&f->e[rk_no[i]+rk_mo[i]*f->n],&f->n);
    }
    add_full2rkmatrix(f,rt,0,0);
    del_fullmatrix(f);
    return;
  }
  atmp = (double*) calloc(n*kmax,sizeof(double));
  btmp = (double*) calloc(m*kmax,sizeof(double));
  rarb = (double*) calloc(kmax*kmax,sizeof(double));
  u = (double*) calloc(kmax*kmax,sizeof(double));
  v = (double*) calloc(kmax*kmax,sizeof(double));
  lwork = 10*kmax*kmax;
  rarbwork = (double*) calloc(lwork,sizeof(double));
  lwork = m+n+10;
  qr_work = (double*) calloc(lwork,sizeof(double));
  tau1 = (double*) calloc(n+m,sizeof(double));
  tau2 = (double*) calloc(n+m,sizeof(double));
  sigma = (double*) calloc(n+m,sizeof(double));
  if(atmp==0x0 || btmp==0x0 || rarb==0x0 || u==0x0 || v==0x0 ||
     rarbwork==0x0 || qr_work==0x0 || tau1==0x0 || tau2==0x0 ||
     sigma==0x0){
     fprintf(stderr,"addparts2rkmatrix: Speicher voll\n");
     exit(1);
  }
  for(i=0; i<rt->kt; i++){      /* rkmatrix r=atmp*btmp aufstellen */
```

```
  dcopy_(&n,&rt->a[i*n],eins_,&atmp[i*n],eins_);
  dcopy_(&m,&rt->b[i*m],eins_,&btmp[i*m],eins_);
}
k = rt->kt;
for(j=0; j<anz; j++){          /* rkmatrix r=atmp*btmp aufstellen */
  for(i=0; i<rk_r[j]->kt; i++){
    dcopy_(&rk_r[j]->n,
           &rk_r[j]->a[i*rk_r[j]->n],eins_,
           &atmp[rk_no[j]+(k+i)*n],eins_);
    dcopy_(&rk_r[j]->m,
           &rk_r[j]->b[i*rk_r[j]->m],eins_,
           &btmp[rk_mo[j]+(k+i)*m],eins_);
  }
  k += rk_r[j]->kt;
}
dgeqrf_(&n,&kmax,atmp,&n,tau1,qr_work,&lwork,&info); /* atmp = qa*ra */
if(info!=0){
  fprintf(stderr,"addparts2rkmatrix: info(dgeqrf,a)=%d\n",info);
  exit(2);
}
dgeqrf_(&m,&kmax,btmp,&m,tau2,qr_work,&lwork,&info); /* btmp = qb*rb */
if(info!=0){
  fprintf(stderr,"addparts2rkmatrix: info(dgeqrf,b)=%d\n",info);
  exit(3);
}
for(i=0; i<kmax; i++){                          /* rarb = ra*rb^T */
  for(j=0; j<kmax; j++){
    for(k=0; k<kmax; k++){
      if(k>=i && k>=j) rarb[i+kmax*j] += atmp[i+k*n]*btmp[j + k*m];
    }
  }
}
dorgqr_(&n,&kmax,&kmax,atmp,&n,tau1,qr_work,&lwork,&info);     /* qa */
if(info!=0){
  fprintf(stderr,"addparts2rkmatrix: info(dorgqr,a)=%d\n",info);
  exit(4);
}
dorgqr_(&m,&kmax,&kmax,btmp,&m,tau2,qr_work,&lwork,&info);     /* qb */
if(info!=0){
  fprintf(stderr,"addparts2rkmatrix: info(dorgqr,b)=%d\n",info);
  exit(5);
}
lwork = 10*kmax*kmax;
dgesvd_("A","A",&kmax,&kmax,rarb,&kmax,sigma,u,&kmax,v,&kmax,
```

```
          rarbwork,&lwork,&info);                  /* rarb = u*sigma*v^T*/
  if(info!=0){
    fprintf(stderr,"addparts2rkmatrix: info(dgesvd)=%d\n",info);
    exit(6);
  }
  for(i=0; i<kmax; i++){
    dscal_(&kmax,&sigma[i],&u[i*kmax],eins_);           /* u = u*sigma */
    dcopy_(&kmax,&v[i],&kmax,&rarb[i*kmax],eins_);         /* rarb = v */
  }
  for(k=0; k<kmax && k<rt->k && k<n && k<m; k++);
  rt->kt = k;
  if(k>0){                                  /* rt->a = qa*u, rt-b=qb*v */
    dgemm_(ntrans,ntrans,&n,&k,&kmax,deins_,atmp,&n,
           u,&kmax,dnull_,rt->a,&n);
    dgemm_(ntrans,ntrans,&m,&k,&kmax,deins_,btmp,&m,
           rarb,&kmax,dnull_,rt->b,&m);
  }
  free(sigma);
  free(tau2);
  free(tau1);
  free(qr_work);
  free(rarbwork);
  free(v);
  free(u);
  free(rarb);
  free(btmp);
  free(atmp);
}
```

Implementierung A.8 *(Weitere Konvertierungen und Additionen)*
Für die Konvertierung zwischen den Matrixformaten `rkmatrix` *und* `fullmatrix` *benötigen wir außer Implementierung A.6 noch die Prozeduren*

```
void convert_f2fmatrix(pfullmatrix f, pfullmatrix ft, int nof, int mof);
void convert_rk2fullmatrix(prkmatrix r, pfullmatrix ft, int nof, int mof);
void convert_rk2rkmatrix(prkmatrix r, prkmatrix rt, int nof, int mof);
```

die jeweils eine `rkmatrix` *oder* `fullmatrix` *in eine* `fullmatrix` *oder* `rkmatrix` *umwandeln. Hier ist wieder zugelassen, daß die zu konvertierende Matrix eine Untermatrix einer größeren Matrix mit den ersten Indizes* `(nof,mof)` *ist. Im Fall* `convert_rk2rkmatrix` *braucht man außer der Matrix-Matrix-Multiplikation* `dgemm_` *auch die Kürzung für* $\mathbf{R}k$*-Matrizen, die in Implementierung A.7 eingeführt wurde.*
Analog zur Konvertierung sind die Additionsroutinen

```
void add_full2fullmatrix(pfullmatrix f, pfullmatrix ft, int nof, int mof);
```

```
void add_rk2fullmatrix(prkmatrix r, pfullmatrix ft, int nof, int mof);
void add_rk2rkmatrix(prkmatrix r, prkmatrix rt, int nof, int mof);
```

zu implementieren, die sich im Fall `add_rk2rkmatrix` *wieder auf das Kürzen in Implementierung A.7 zurückführen lassen.*

A.3. $\mathcal{H}$-Matrix

Implementierung A.9 *(Datenstruktur Supermatrix)*
Die Supermatrix S besteht aus `n`$\times$`m` *Untermatrizen* S_{ij}*:*

$$S = \begin{bmatrix} S_{11} & \dots & S_{1m} \\ \vdots & \ddots & \vdots \\ S_{n1} & \dots & S_{nm} \end{bmatrix}$$

Die Untermatrizen S_{ij} *sind wieder von der Struktur* Supermatrix *und werden als ein Array von Zeigern auf die Untermatrizen in* `s` *in der Reihenfolge* $S_{11}, \dots, S_{n1}, S_{12}, \dots, S_{n2}, \dots, S_{nm}$ *gespeichert. Repräsentiert S ein Blatt des zugrundeliegenden Baumes, so wird der Zeiger* `s:=0` *gesetzt und in* `r` *bzw.* `f` *ein Zeiger auf die zu dem Blatt gehörende* `rkmatrix` *bzw.* `fullmatrix` *und* `f:=0x0` *bzw.* `r:=0x0` *gesetzt. S beschreibt eine* `sizen`$\times$`sizem`*-Matrix.*

```
typedef struct _supermatrix supermatrix;
typedef supermatrix *psupermatrix;
struct _supermatrix {
  int n;              /* n mal m Untermatrizen, */
  int m;
  int sizen;          /* Groesse sizen mal sizem, */
  int sizem;
  prkmatrix r;
  pfullmatrix f;
  psupermatrix* s;
};
```

Implementierung A.10 *(Auswertung von Supermatrizen)*
Die Auswertung einer Supermatrix `s` *wird, falls* `s` *kein Blatt des zugrundeliegenden Baumes repräsentiert, auf die Auswertung in den Untermatrizen* `s->s[i]` *zurückgeführt. Repräsentiert* `s` *ein Blatt, so wird für* `s->r` *bzw.* `s->f` *die Auswertung für* $\mathbf{R}k$*-Matrizen bzw. Full-Matrizen aufgerufen.*

```
void eval_supermatrix(psupermatrix s, double* v, double* w){
  int i,j;
  int vindex=0;
  int windex=0;
  int sindex=0;
  int n = s->n;
```

```
  int m = s->m;
  psupermatrix* s_el = s->s;
  if(s->s!=0x0){
   for(i=0; i<n; i++){                          /* 1. spalte auswerten */
      eval_supermatrix(s_el[sindex],v,&w[windex]);
      windex += s_el[sindex]->sizen;
      sindex++;
    }
    for(j=1; j<m; j++){                       /* 2.-m. spalte auswerten */
      vindex += s_el[sindex-1]->sizem;
      windex = 0;
      for(i=0; i<n; i++){
        addeval_supermatrix(s_el[sindex],&v[vindex],&w[windex]);
        windex += s_el[sindex]->sizen;
        sindex++;
      }
    }
  } else {                                          /* s ist ein Blatt */
    if(s->r!=0x0){
      eval_rkmatrix(s->r,v,w);
    } else {
      eval_fullmatrix(s->f,v,w);
    }
  }
}
```

Analog werden die Funktionen zur Auswertung der transponierten Matrix und zum Aufaddieren von `s·v` *auf einen Vektor* `w` *definiert:*

```
void addeval_supermatrix(psupermatrix s, double* v, double* w);
void evaltrans_supermatrix(psupermatrix s, double* v, double* w);
void addevaltrans_supermatrix(psupermatrix s, double* v, double* w);
```

Implementierung A.11 *(Konvertieren von bzw. in Supermatrizen)*
Die Konvertierung einer `fullmatrix` *oder* `rkmatrix` *erfolgt für jeden zu einem Blatt gehörenden Block der Supermatrix einzeln, dort werden dann* `convert_rk2rkmatrix`, `convert_rk2fullmatrix`, `convert_full2rkmatrix` *und* `convert_f2fmatrix` *aufgerufen. Eine Konvertierung von einer $\mathcal{H}$-Matrix-Struktur in eine andere wird nicht implementiert, da wir für die Inversion und Multiplikation voraussetzen, daß die zugrundeliegenden $\mathcal{H}_\times$-Bäume wie in Lemma 4.11 aus den entsprechend passenden $\mathcal{H}$-Bäumen erzeugt wurden. Hier ist lediglich das Kopieren der Daten einer Supermatrix* `s` *in eine Supermatrix* `sc` *mit der Prozedur*

```
void copydata_supermatrix(psupermatrix s, psupermatrix sc);
```

nötig, die in den Blättern auf `convert_rk2rkmatrix` *und* `convert_f2fmatrix` *zurückgreift.*

Implementierung A.12 *(Skalieren von Supermatrizen)*
Die Multiplikation einer Supermatrix `s` *mit einem Skalar* `value` *führt die Prozedur*

```
void scale_supermatrix(psupermatrix s, double value);
```

durch. In den Blättern wird die LAPACK *-Funktion* `dscal_` *aufgerufen.*

Implementierung A.13 *(Multiplikation von* $\mathcal{R}k$*-Matrizen und Full-Matrizen)*
Für die hierarchische Approximation und die Multiplikation benötigen wir eine Funktion, die eine neue Matrix mit dem Inhalt `a`$\cdot$`b` *anlegt. Dabei beschränken wir uns auf den Fall, daß* `a,b` *Full-Matrizen sind (*$\rightarrow$ *liefert eine Full-Matrix zurück), oder daß wenigstens eine von beiden eine* $\mathcal{R}k$*-Matrix ist (*$\rightarrow$ *liefert eine* $\mathcal{R}k$*-Matrix zurück).*

```
pfullmatrix get_mul_fullmatrix(psupermatrix a,psupermatrix b);
prkmatrix get_mul_rkmatrix(psupermatrix a, psupermatrix b);
```

Implementierung A.14 *(Multiplikation und Konvertierung in eine* $\mathcal{R}k$*-Matrix)*
Wir führen gleichzeitig die (formatierte) Multiplikation `a`$\odot$`b`*, Addition* `r`$\oplus$`a`$\odot$`b` *und hierarchische Approximation des Ergebnisses durch einer* $\mathcal{R}k$*-Matrix* `r` *durch:*

$$\mathtt{a} = \begin{bmatrix} a_{1,1} & \dots & a_{1,am} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,am} \end{bmatrix}, \mathtt{b} = \begin{bmatrix} b_{1,1} & \dots & b_{1,m} \\ \vdots & \ddots & \vdots \\ b_{am,1} & \dots & b_{am,m} \end{bmatrix}.$$

Die Matrix r darf Teil einer größeren $\mathcal{R}k$*-Matrix mit erstem Index* `(nof,mof)` *sein.*

```
void add_prod2rkmatrix(prkmatrix r, psupermatrix a, psupermatrix b,
                       int nof, int mof){
  int i,j,k,n=a->n,m=b->m,am=a->m,no,mo;
  prkmatrix r2=0x0;
  pfullmatrix f;
  if(a->s!=0x0 && b->s!=0x0){    /* produkt von supermatrizen */
    if(b->n==am){
        no = nof;
        for(i=0;i<n;i++){
          mo = mof;
          for(j=0; j<m; j++){
            if(r2==0x0 || a->s[i]->sizen!=r2->n ||
               b->s[j*am]->sizem!=r2->m){
              if(r2!=0x0) del_rkmatrix(r2);
              r2 = new_rkmatrix(r->k,a->s[i]->sizen,
                                b->s[j*am]->sizem);
            }else{
              r2->kt=0;
            }
            for(k=0; k<am; k++)        /* r2 = (ab)_{i,j}
```

```
              add_prod2rkmatrix(r2,a->s[i+k*n],
                                b->s[k+j*am],0,0);
            addpart2rkmatrix(r,no,mo,r2,0,0,r2->n,r2->m);
            mo += b->s[j*am]->sizem;

          }
          no += a->s[i]->sizen;
        }
        if(r2!=0x0) del_rkmatrix(r2);
      }
    } else {
      fprintf(stderr,"Matrizen inkompatibel in add_prod2rkmatrix.\n");
      exit(1);
    }
  } else {
    if(a->r!=0x0 || b->r!=0x0){                          /* rkmatrix */
      r2 = get_mul_rkmatrix(a,b);
      addpart2rkmatrix(r,nof,mof,r2,0,0,r2->n,r2->m);
    } else {                                            /* fullmatrix */
      f = get_mul_fullmatrix(a,b);
      r2 = new_rkmatrix(r->k,a->sizen,b->sizem);
      add_full2rkmatrix(f,r2,0,0);
      addpart2rkmatrix(r,nof,mof,r2,0,0,r2->n,r2->m);
      del_rkmatrix(r2);
    }
  }
}
```

Implementierung A.15 *(Multiplikation von Supermatrizen)*
Die Multiplikation zweier Supermatrizen `a,b`*, Addition des Produktes zu einer Supermatrix* `c` *und Konvertierung von* `ab+c` *in die Struktur von* `c` *wird mit der nachfolgenden Prozedur durchgeführt. Das Kürzen wird wie in Bemerkung 4.12 durch sukzessives Aufaddieren (Approximation) vorgenommen, wobei die dabei auftretenden Konvertierungen von Supermatrizen in* $\mathbf{R}k$*-Matrizen mit der hierarchischen Approximation aus Abschnitt 4.2.2 erfolgen (*`add_prod2rkmatrix`*, → Implementierung A.14). Für die Operation* `c`$:=$`a`$\oplus$`b` *initialisiert man* `c` *vorab auf Null. Soll das exakte Produkt berechnet werden, so bestimmt man zuerst die Rangverteilung k der Zielmatrix und ruft für die auf Rang k erweiterte Matrix* `c` *die Multiplikation auf, anschließend kann man z.B. die Bestapproximation (Konvertierung) der erweiterten Matrix in der ursprünglichen Struktur durchführen.*

```
void muladd_supermatrix(psupermatrix c, psupermatrix a, psupermatrix b){
  int i,j,n=c->n,m=c->m,k,nm=a->m;
  psupermatrix a_el,b_el,c_el;
```

```
prkmatrix r;
pfullmatrix f;
if(a->sizem!=b->sizen || a->sizen!=c->sizen || b->sizem!=c->sizem){
  fprintf(stderr,"Matrizen inkompatibel in muladd_supermatrix\n");
  exit(1);
}
if(c->s!=0x0){
  if(a->s!=0x0 && b->s!=0x0){ /* nur supermatrizen */
    for(i=0; i<n; i++)
      for(j=0; j<m; j++){
        c_el = c->s[i+j*n];
        a_el = a->s[i];
        b_el = b->s[j*nm];
        muladd_supermatrix(c_el,a_el,b_el);
        for(k=1; k<nm; k++){
          a_el = a->s[i+k*n];
          b_el = b->s[k+j*nm];
          muladd_supermatrix(c_el,a_el,b_el);
        }
      }
  } else {               /* a oder b ist rk oder fullmatrix */
    if(a->r!=0x0 || b->r!=0x0){
      r = get_mul_rkmatrix(a,b);
      add_rk2supermatrix(r,c,0,0);
    } else {
      f = get_mul_fullmatrix(a,b);
      add_full2supermatrix(f,c,0,0);
    }
  }
} else {
  if(c->r!=0x0){
    if(a->r!=0x0 || b->r!=0x0){
      r = get_mul_rkmatrix(a,b);
      addpart2rkmatrix(c->r,0,0,r,0,0,r->n,r->m);
    } else {
      if(a->f!=0x0 || b->f!=0x0){
        f = get_mul_fullmatrix(a,b);
        add_full2rkmatrix(f,c->r,0,0);
      } else {
        add_prod2rkmatrix(c->r,a,b,0,0);
      }
    }
  } else {
    if(a->r!=0x0 || b->r!=0x0){
```

```
        r = get_mul_rkmatrix(a,b);
        add_rk2fullmatrix(r,c->f,0,0);
      } else {
        if(a->f!=0x0 && b->f!=0x0){
          dgemm_(ntrans,ntrans,&c->sizen,&c->sizem,&a->sizem,deins_,
                 a->f->e,&a->sizen,b->f->e,&b->sizen,deins_,
                 c->f->e,&c->sizen);
        } else {
          f = get_mul_fullmatrix(a,b);
          add2fullmatrix(c->f,f);
        }
      }
    }
  }
}
```

Implementierung A.16 *(Inversion von Supermatrizen)*
Die (formatierte) Inversion erfolgt wie in Algorithmus 4.18, d.h. `si,s,sw` *gehören zur selben $\mathcal{H}$-Matrix-Struktur und auf der Diagonalen befinden sich nur Supermatrizen oder Full-Matrizen.*

```
void invert_supermatrix(psupermatrix si, psupermatrix s,
                        psupermatrix sw){
  int n = s->n;
  int m = s->m;
  int i,j,k,l;
  int info;
  int* ipiv;
  psupermatrix* sw_e = sw->s;
  psupermatrix* s_e = s->s;
  psupermatrix* si_e = si->s;
  if(si->f!=0 && s->f!=0x0 && sw->f!=0x0){    /* fullmatrix Inversion */
    n = si->sizen;
    m = si->sizem;
    ipiv = (int*) malloc((size_t) n*sizeof(int));
    if(ipiv==0x0){
      fprintf(stderr,"Speicher voll in invert_supermatrix.\n");
      exit(1);
    }
    dcopy_(&si->f->nm,s->f->e,eins_,si->f->e,eins_);
    dgetrf_(&n,&m,si->f->e,&n,ipiv,&info);
    if(info!=0){
      fprintf(stderr,
              "Fehler bei LU-Zerlegung in invert_supermatrix, info=%d.\n",
```

```
            info);
    exit(2);
  }
  dgetri_(&n,si->f->e,&n,ipiv,sw->f->e,&sw->f->nm,&info);
  if(info!=0){
    fprintf(stderr,
            "Fehler bei Inversion in invert_supermatrix, info=%d.\n",
            info);
    exit(3);
  }
  free(ipiv);
} else {
  if(si->s==0x0 || s->s==0x0 || sw->s==0x0){
    fprintf(stderr,"Diagonale defekt in invert_supermatrix.\n");
    exit(4);
  }
  for(l=0; l<n; l++){
    invert_supermatrix(si_e[l+n*l],s_e[l+n*l],sw_e[l+n*l]);
    for(j=0; j<l; j++){
      mul_supermatrix(sw_e[l+n*j],si_e[l+n*l],si_e[l+n*j]);
      copydata_supermatrix(sw_e[l+n*j],si_e[l+n*j]);
    }
    for(j=l+1; j<m; j++){
      mul_supermatrix(sw_e[l+n*j],si_e[l+n*l],s_e[l+n*j]);
      copydata_supermatrix(sw_e[l+n*j],s_e[l+n*j]);
    }
    for(i=l+1; i<n; i++){
      for(j=0; j<=l; j++){
        mul_supermatrix(sw_e[i+n*j],s_e[i+n*l],si_e[l+n*j]);
        scale_supermatrix(sw_e[i+n*j],-1.0);
        addto_supermatrix(si_e[i+n*j],sw_e[i+n*j]);
      }
      for(j=l+1; j<m; j++){
        mul_supermatrix(sw_e[i+n*j],s_e[i+n*l],s_e[l+n*j]);
        scale_supermatrix(sw_e[i+n*j],-1.0);
        addto_supermatrix(s_e[i+n*j],sw_e[i+n*j]);
      }
    }
  }
  for(l=n-1; l>=0; l--)
    for(i=l-1; i>=0; i--)
      for(j=0; j<m; j++){
        mul_supermatrix(sw_e[i+n*j],s_e[i+n*l],si_e[l+n*j]);
        scale_supermatrix(sw_e[i+n*j],-1.0);
```

```
          addto_supermatrix(si_e[i+n*j],sw_e[i+n*j]);
        }
  }
}
```

A.4. Newton-Iteration zur Berechnung von $\mathrm{sign}(M)$

Zur Berechnung von sign(s) werden vier Matrizen `s, ssign, swork, swork2` von derselben Struktur wie `s` benötigt. Der Inhalt von `s` wird überschrieben und zu Beginn muß in `ssign` derselbe Inhalt wie in `s` stehen. Die Iteration bricht ab, wenn sich die Konvergenzrate, wie in Bemerkung 8.11 erwähnt, verschlechtert. Zur Berechnung von $\|I-(\tilde{S}^{(i)})^2\|_2$ und $\|S^{(i)}\|_2$ dienen die Funktionen `norm_2_supermatrixprodminusid` und `norm_2_supermatrix`.

```
void sign_supermatrix(psupermatrix ssign, psupermatrix s,
                      psupermatrix swork, psupermatrix swork2){
  int i;
  double norm = 0.0,normalt = 1.0;
  double l1,l2,faktor;
  double rate=0.0,rate_best=1.0;
  for(i=0; i>=0; i++){
    if(i==0) l1 = norm_2_supermatrix(ssign);
    scale_supermatrix(swork2,0.0);
    scale_supermatrix(swork,0.0);
    invert_supermatrix(swork2,ssign,swork);
    if(i==0){
      l2 = norm_2_supermatrix(swork2);
      faktor = sqrt(l2)/sqrt(l1);
      scale_supermatrix(s,0.5*faktor);
      scale_supermatrix(swork2,0.5/faktor);
    }
    add_supermatrix(ssign,s,swork2);
    if(i>0) scale_supermatrix(ssign,0.5);
    if(i==0) scale_supermatrix(s,2.0/faktor);
    norm = norm_2_supermatrixprodminusid(ssign,ssign);
    if(rate < 2.0*rate_best || i<5){
      if(rate<rate_best && i>2) rate_best = rate;
      rate = norm/normalt;
      normalt = norm;
      copycontent_supermatrix(ssign,s);
    }else{
      i=-2;
    }
  }
}
```

Literatur

[1] E. Anderson [et al.]:
LAPACK users' guide - 3rd ed.,
SIAM, 1999.

[2] M. Bebendorf:
Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen,
Dissertation, Saarbrücken (2000).

[3] H. Fuchs, Z. M. Kedem, B. F. Naylor:
On Visible Surface Generation by a Priori Tree Structures,
Computer Graphics 14 (1980) 124-133.

[4] I. P. Gavrilyuk, W. Hackbusch, B. Khoromskij:
$\mathcal{H}$-Matrix Approximation of the Operator Exponent with Applications, Preprint 42 (2000), Max-Planck-Institute for Mathematics in the Sciences, Leipzig (http://www.mis.mpg.de/preprints/2000/).

[5] K. Giebermann:
Multilevel approximation of boundary integral operators,
Computing, to appear.

[6] G. H. Golub, C. F. Van Loan:
Matrix Computations,
Johns Hopkins University Press (1996).

[7] L. Grasedyck, S. Le Borne:
Adaptive refinement and clustering in the context of $\mathcal{H}$-matrices,
Computing, to appear.

[8] W. Hackbusch:
Iterative Lösung großer schwachbesetzter Gleichungssysteme,
B. G. Teubner, Stuttgart (1993).

[9] W. Hackbusch:
Theorie und Numerik elliptischer Differentialgleichungen,
B. G. Teubner, Stuttgart (1986).

[10] W. Hackbusch:
Integralgleichungen,
B. G. Teubner, Stuttgart (1997).

[11] W. Hackbusch:
A Sparse Matrix Arithmetic based on $\mathcal{H}$-Matrices. Part I: Introduction to $\mathcal{H}$-Matrices,
Computing 62 (1999) 89-108.

[12] W. Hackbusch:
Multi-Grid Methods and Applications,
Springer-Verlag Berlin, Heidelberg (1985).

[13] W. Hackbusch, B. Khoromskij:
A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems,
Computing 64 (2000), 1, 21-47.

[14] W. Hackbusch, B. Khoromskij:
A sparse $\mathcal{H}$-matrix arithmetic: general complexity estimates,
J. Comp. Appl. Math. 125 (2000), 479-501.

[15] L. Hörmander:
The Analysis of Linear Partial Differential Operators I,
Grundlehren der mathematischen Wissenschaften, Band 256, Springer-Verlag Berlin, Heidelberg, New York, Tokyo (1983).

[16] L. Hörmander:
The Analysis of Linear Partial Differential Operators II,
Grundlehren der mathematischen Wissenschaften, Band 257, Springer-Verlag Berlin, Heidelberg, New York, Tokyo (1983).

[17] L. Hörmander:
The Analysis of Linear Partial Differential Operators III,
Grundlehren der mathematischen Wissenschaften, Band 274, Springer-Verlag Berlin, Heidelberg, New York, Tokyo (1985).

[18] B. W. Kernighan, D. M. Ritchie:
The C Programming Language, Second Edition,
Prentice Hall (1988).

[19] D. L. Kleinman:
On an iterative technique for Riccati equations computation,
IEEE Trans. Automat. Control, AC-13 (1968), 114-115.

[20] H. W. Knobloch, H. Kwakernaak:
Lineare Kontrolltheorie,
Springer-Verlag Berlin, Heidelberg, New York (1985).

[21] C. Lage:
Softwareentwicklung zur Randelementmethode: Analyse und Entwurf effizienter Techniken,
Dissertation, Kiel (1995).

[22] K. Mehlhorn:
Data Structures and Algorithms 1: Sorting and Searching,
Monographs on Theoretical Computer Science, Springer-Verlag Berlin, Heidelberg (1984).

[23] J. D. Roberts:
Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, Internat. J. Control 32 (1980), 677-687.

[24] J. I. G. Rosen, C. Wang:
A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations, Siam J. Numer. Anal. 32 (1995), 2, 514-541.

[25] D. L. Russell:
Mathematics of Finite Dimensional Control Systems: Theory and Design, Lecture Notes in Pure and Applied Mathematics, 43, Marcel Dekker, New York, 1979.

[26] S. Sauter:
Über die effiziente Verwendung des Galerkinverfahrens zur Lösung Fredholmscher Integralgleichungen,
Dissertation, Kiel (1992).

[27] S. Sauter:
Variable order panel clustering,
Computing 64 (2000) 223-261.

[28] N. Shimakura:
Partial Differential Operators of Elliptic Type,
Translations of Mathematical Monographs, American Mathematical Society, Providence (1992).

[29] G. W. Stewart:
Four Algorithms for the Efficient Computation of Truncated Pivoted QR Approximations to a Sparse Matrix,
Numer. Math. (1999), 83: 313-323.

[30] J. Stoer:
Einführung in die Numerische Mathematik I,
Springer-Verlag (1983).

[31] S. Summit:
C Programming FAQs,
Addison-Wesley Publishing Company, ISBN 0-201-84519-9 (1996).

[32] J. H. Wilkinson:
The Algebraic Eigenvalue Problem,
Oxford University Press (1965).

[33] J. H. Wilkinson:
Convergence of the LR, QR, and Related Algorithms,
Computer Journal 8 (1965), 77-84.

Index

Symbolverzeichnis

C_{bal} 72
C_{id} 80
C_{sp}, Schwachbesetztheitskonstante ... 67
$\bar{C}_{sp}$, Summe der stufenweisen Schwachbesetztheitskonstanten 88
$C_{sp}^{(l)}$, stufenweise Schwachbesetztheitskonstante 88
L_T, Menge der Stufen mit Blättern .. 27
$M_{\mathcal{H}}$, gekürzte Matrix 43
$M|^b$, Fortsetzung 42
$M|_b$, Einschränkung 42
$N_{\mathcal{H},\odot}^{apx}(k,T,Z)$ 85
$N_{\mathcal{H},\odot}^{best}(k,T,Z)$ 85
$N_{\mathcal{H},\odot}^{exakt}(k,T,Z)$ 82
$N_{F,St}(n,m)$ 67
$N_{F\cdot V}(n,m)$ 67
$N_{V\cdot\mathbf{R}k}(n,m)$ 16
$N_{\mathcal{H}^{-1}}(k,T,Z)$ 86
$N_{\mathcal{H},St}(k,T,Z)$ 75
$N_{\mathcal{H},\oplus}(k,T,Z)$ 79
$N_{\mathcal{H}\cdot V}(k,T,Z)$ 75
$N_{\mathbf{R}k,SVD}(n,m)$ 20
$N_{\mathbf{R}k,St}(n,m)$ 67
$N_{\mathbf{R}k,\oplus}(n,m)$ 24
$N_{\mathbf{R}k\cdot V}(n,m)$ 16
$N_{\mathbf{R}k\cdot\mathbf{R}k}(n,m',m)$ 17
$S_T(q)$, Söhne von q im Baum T 26
$T+T'$, Summe von $\mathcal{H}$-Bäumen 38
$T\cdot T'$, Produkt von $\mathcal{H}$-Bäumen 40
$T^{(i)}$, Elemente der i-ten Stufe von T . 26
$T_I\otimes T_J$ 27
U_j 49
Z, Zulässigkeitsbedingung 33
$Z+Z'$, Summe von Zulässigkeitsbedingungen 47
$Z\cdot Z'$, Produkt von Zulässigkeitsbedingungen 49
Z_η, Standard-Zulässigkeitsbedingung 33
η-zulässig 34
$\mathcal{L}(T)$, Blätter 26
$\mathcal{L}(T,\leq i)$, Blätter der Stufen $0-i$ ... 26
$\mathcal{L}(T,i)$, Blätter der Stufe i 26
$\mathcal{M}_{\mathcal{H},k}(T,Z)$, $\mathcal{H}$-Matrizen 42
$\odot$, formatierte $\mathcal{H}$-Multiplikation 50
$\oplus$, formatierte $\mathcal{H}$-Addition 47
$\oplus$, formatierte $\mathbf{R}k$-Addition 24
$\tau^{(j)}$, Vorfahr der Stufe j 48
b_{min}, minimale Blockgröße 37
g_c, kompensierende Kernfunktion ... 101
k, Rangverteilung 42
$k+k'$, Summe von Rangverteilungen 47
$k\cdot k'$, Produkt von Rangverteilungen 49
p_T, Baumtiefe 26
$\mathbb{C}k$-Matrizen 25
$\mathcal{H}$-Approximation 42
$\mathcal{H}$-Baum 27
$\mathcal{H}$-Matrix 42
$\mathcal{H}_\times$-Baum 27
$\mathcal{L}^-$, nicht zulässige Blätter 67
$\mathcal{L}^+$, zulässige Blätter 67
$\mathbf{R}k$-Matrix 15
$\mathbf{R}_{\leq k}$-Matrix 15

A

adaptive $\mathcal{H}$-Addition 91
adaptive $\mathcal{H}$-Inversion 93
adaptive $\mathcal{H}$-Multiplikation 92
adaptive Konvertierung 91
Addition für $\mathcal{H}$-Matrizen 47
allgemeiner $\mathcal{H}$-Baum 27
Alternative Clusterungen 33
approximative Spektralnorm 62
approximative Spektralnorm der Inversen 92
Aus T_I, T_J gebildeter $\mathcal{H}$-Baum 37

B

Baum 26
Bestapproximation 42
Binäre Raumzerlegung 28
Blatt 26

Block-Gauß-Elimination 51
Bounding-Box 34
BSP, Binary Space-Partitioning 28

D
darstellbar als $\mathcal{H}$-Matrix 42
Darstellung der Produktmatrix 49
Dreidimensionale Modell-Clusterung . 32

E
effiziente Zulässigkeitsbedingung 35
Eindimensionale Modell-Clusterung . . 31
einfache Zulässigkeitsbedingung 34

F
fast idempotent 80
formatierte Inversion 53
formatierte Newton-Iteration zur Inversion 56

G
Galerkin-Diskretisierung 96
gekürzte QR-Zerlegung 20
gekürzte Singulärwertzerlegung 17
geometrisch balanciert 28

H
Hierarchische Approximation 44
Hierarchische Matrizen 42
hierarchische Partitionierung 26

I
Inversion von $\mathcal{H}$-Matrizen 51

K
Kürzen . . . . . *siehe* Konvertierung
kardinalitätsbalanciert 28
Knoten 26
Kollokationsmethode 96
Konvertierung 42
Kreuzprodukt von $\mathcal{H}$-Bäumen 27

L
Laplace-Gleichung 109
lokal geometrisch balanciert 72

M
Multiplikation von $\mathcal{H}$-Matrizen 48

N
Newton-Iteration zur Inversion 56
Normen von $\mathbf{R}k$-Matrizen 24
Nyström-Methode 96

O
Orthogonale Iteration 20

P
Poisson-Gleichung 111
Potentialgleichung 109
Produkt von $\mathcal{H}$-Bäumen 40
Produktpartitionsbaum 27

Q
QR-Zerlegung 20

R
Rangverteilung 42

S
schwachbesetzte Blockstruktur 67
Singulärwertzerlegung 16
Standard-Zulässigkeitsbedingung 33
Stufe eines Baumes 26
Summe von $\mathcal{H}$-Bäumen 38

V
Vorfahr 48

W
Wurzel 26

Z
zulässiger $\mathcal{H}$-Baum 37
Zulässigkeitsbedingung 33
Zweidimensionale Modell-Clusterung 32