Theorie und Anwendungen Hierarchischer Matrizen

Dissertation
zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität
to 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)

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 $R_k$-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 $R_k$-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 $H$-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 Spektal- 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 \( H \)-Matrizen in der Programmiersprache C 131
A.1 Vorwort .............................................................. 131
A.2 Full-Matrix und \( R_k \)-Matrix .................................. 132
A.3 \( H \)-Matrix ......................................................... 139
A.4 Newton-Iteration zur Berechnung von \( \text{sign}(M) \) ............... 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 \( \varepsilon \) 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


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

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

\[ u(x) = \int_{\Omega} g(x, y) u(y) \, dy + f(x) \quad \forall x \in \Omega \]  

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 \((I - K)u = f\), 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 \( K \), 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 \( K \) aus, die ähnliche Eigenschaften wie ein lokaler Operator hat. Wir fixieren für die Einführung die Kernfunktion

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

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

\[ |\partial_\iota^j g(x, y)| \leq j! \frac{1}{|x - y|^j}, \quad j \in \mathbb{N}. \]
1.2 Galerkinverfahren

Bei der Diskretisierung mit dem Galerkinverfahren macht man den Ansatz

\[ u(y) = \sum_{i=1}^{n} \phi_i(y)u_i \]

\[ \phi_i = \text{i-te Basisfunktion} \]

für die Lösung \( u \) und löst die schwache Formulierung

\[ (\phi_l, (I - K)u)_{L^2(\Omega)} = (\phi_l, f)_{L^2(\Omega)} \]

\[ \forall l = 1, \ldots, n \]

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

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

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

\[ g_e(x,y) := \sum_{j=0}^{\mu-1} \frac{1}{j!} \partial_x^j g(x_0,y)(x-x_0)^j \]

bzgl. der ersten Variable in \( x_0 \) bis zur Ordnung \( \mu \), so entsteht ein Fehler

\[ |g(x,y) - g_e(x,y)| \leq \frac{1}{\mu!} |x-x_0|^{\mu} \max\{|\partial_x^\mu g(\xi,y)| \mid \xi \in [x_0,x]\} \]

\[ \leq \frac{1}{\mu!} |x-x_0|^{\mu} \max\left\{ \frac{1}{|\xi-y|^{\mu}} \mid \xi \in [x_0,x]\right\} \]

\[ = \max\left\{ \left( \frac{|x-x_0|}{|\xi-y|} \right)^{\mu} \mid \xi \in [x_0,x] \right\}. \]

Um eine mit der Entwicklungsordnung \( \mu \) besser werdende Approximation zu erreichen, muß \( \frac{|x-x_0|}{|\xi-y|} < 1 \) gelten.
1.4 Zulässigkeitsbedingung

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

\[
\min \{ \text{diam}(\sigma), \text{diam}(\tau) \} \leq 2\eta \text{dist}(\sigma, \tau)
\]

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 \( \sigma \). Ist \( x_0 \) der Mittelpunkt von \( \sigma \) (Mittelpunkt des Intervalls) und \( (x, y) \in \sigma \times \tau \), so gilt für alle \( \xi \in [x_0, x] \):

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

Setzen wir \( \eta < 1 \) voraus, so erhalten wir eine bzgl. der Entwicklungsortung \( \mu \) exponentielle Konvergenz des Taylorpolynoms gegen die Kernfunktion. Wir erhalten also direkt aus der Zulässigkeitsbedingung die Approximationseigenschaft, welche mit kleiner werdendem \( \eta \) besser wird. Ein kleineres \( \eta \) 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 \( \eta \)-zulässiges Produkt \( \sigma \times \tau \) sowie eine Entwicklungsortung \( \mu \) fixiert. Der entartete Kern \( g \) hat auf \( \sigma \times \tau \) die Darstellung (4). Für Basisfunktionen mit \( \text{supp} \phi_l \subset \sigma \) und \( \text{supp} \phi_i \subset \tau \) kann man \( g \) durch den entarteten Kern (4) approximieren und erhält mit den Bezeichnungen

\[
a_l^{(j, \sigma, \tau)} := \int_{\sigma} (x - x_0)^j \phi_l(x) \, dx
\]

\[
b_i^{(j, \sigma, \tau)} := \int_{\tau} \frac{1}{j!} \partial^j g(x_0, y) \phi_i(y) \, dy
\]

die Rang-\( \mu \)-Darstellung

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

von \( K \) auf der Produkt-Indexmenge \( I_\sigma \times I_\tau := \{ (l, i) \mid \text{supp} \phi_l \subset \sigma, \text{supp} \phi_i \subset \tau \} \).

Eine Rang-\( \mu \)-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 Partitionierung zu erzeugen, ist zum Beispiel die sukzessive Bisektion von $\Omega$: 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

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

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

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

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.

Abbildung 1: Partitionierungen für $p = 2, 3, 4, 5$. Der Punkt (0,0) liegt jeweils oben links, (1,1) unten rechts.
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 komplexe 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 $\Omega$ und $\Omega \times \Omega$ in Form von Baumstrukturen $T_\Omega$ und $T_{\Omega \times \Omega}$ an.

$T_\Omega$ ist durch die folgenden beiden Bedingungen charakterisiert:

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

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 $\Omega_{2i-1}^{(l+1)} \times \Omega_{2i-1}^{(l+1)}$, $\Omega_{2i}^{(l+1)} \times \Omega_{2i}^{(l+1)}$, $\Omega_{2i}^{(l+1)} \times \Omega_{2i-1}^{(l+1)}$ und $\Omega_{2i-1}^{(l+1)} \times \Omega_{2i}^{(l+1)}$.

Es ist zu beachten, daß in $T_\Omega$ 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

$$I_i^{(l)} := \left\{ i \in \{1, \ldots, n\} \mid \supp \phi_i \subset \Omega_i^{(l)} \right\} = \left\{ (i-1) \frac{n}{2^l} + 1, \ldots, (i-1) \frac{n}{2^l} + \frac{n}{2} \right\}$$
Abbildung 3: Hierarchische Partitionierung von $[0,1] \times [0,1]$ für $p = 4$
der Indizes von Basisfunktionen, deren Träger in $\Omega_{i}^{(0)}$ enthalten ist. Analog zu den Bäumen $T_{1}$ und $T_{1 \times 1}$ erhalten wir die Bäume $T_{1}$ und $T_{1 \times 1}$ zu den Indexmengen $I = \{1, \ldots, n\}$ und $I \times I$.

Die Menge der Hierarchischen Matrizen ($H$-Matrizen) zu dem Baum $T_{1 \times I}$ und einem vorgegebenen Rang $k \in \mathbb{N}_{0}$ ist definiert als

$$\mathcal{M}_{H,k}(T_{1 \times I}) := \{ M \in \mathbb{K}^{I \times I} \mid \text{rang}(M_{b}) \leq k \ \forall b \in \mathcal{L}^{+}(T_{1 \times I}) \},$$

wobei $\mathcal{L}^{+}(T_{1 \times I})$ 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 $|I_{b}| = 2^{p-1}$ definieren. Wir nehmen an, daß für alle zulässigen Blätter des Baumes $T_{1 \times I}$ bereits die arithmetischen Operationen etabliert sind (Rang-$k$-Arithmetik, $k$ der maximale Rang in den zulässigen Blöcken), sowie für die nicht zulässigen Blätter ($1 \times 1$-Matrizen) die herkömmliche Arithmetik verwendet wird. Seien also

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

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

\[
\sum_{l=0}^{p} \kappa 2^l N_{Rk,\oplus}(2^{p-l}).
\]

Wie wir später sehen werden, ist \( N_{Rk,\oplus}(j) \leq C k^2 j \), also

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

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 Inversio n bei festgehaltenem Rang wird in Abschnitt 5 abgeschätzt.


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 Rk-Matrizen und werden dabei auf die Singulärwertzerlegung einer Rk-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 \((R_{\leq k}\text{-Matrix})\)**

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

**Bemerkung 2.2 \((\text{Idealeigenschaft von } R_{\leq k}(n,m))\)**

\( (R_{\leq k}(n,m),+) \) ist keine Untergruppe von \( (\mathbb{R}^{n,m},+) \), da sie bezüglich der Addition nicht abgeschlossen ist, aber \( R_{\leq k}(n,m) \) 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 \( R_{\leq k}\)-Matrizen sind die \( R_k\)-Matrizen.

**Definition 2.3 \((R_k\text{-Matrix})\)**

Eine Matrix \( M \in \mathbb{R}^{n,m} \) bezeichnen wir als eine \( R_k\)-Matrix, wenn sie in der Darstellung

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

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

**Bemerkung 2.4 \((\text{Indexmengen und Körper})\)**

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 \( R_k(n,m)\)-Matrizen betrachtet werden.

Auf die Behandlung komplexwertiger Matrizen wird in Abschnitt 2.7 eingegangen, im folgenden behandeln wir nur reellwertige Matrizen.

**Bemerkung 2.5 \((R_k(V,W)\text{-Matrizen})\)**

Sei \( V \) ein \( k\)-dimensionaler Unterraum des \( \mathbb{R}^n \) und \( W \) ein \( k\)-dimensionaler Unterraum von \( \mathbb{R}^m \). Die Menge der \( R_k(V,W)\)-Matrizen ist dann definiert als die Menge der \( R_k\)-Matrizen, deren Vektoren aus der Darstellung (6) die Bedingung \( a^i \in V, b^i \in W \) für \( i = 1, \ldots, k \) erfüllen.
1, . . . , k erfüllen. \( R^k(V, W) \) ist bzgl. der Addition abgeschlossen und damit (im Gegensatz zu \( R^k(n, m) \)) eine Untergruppe von \( (\mathbb{R}^{n,m}, +) \).

**Bemerkung 2.6 (Singulärwertzerlegung)**
Eine Matrix \( M \in \mathbb{R}^{n,m}, l := \text{rang}(M) \), welche in Singulärwertzerlegung
\[ M = U \Sigma V^T \]
mit \( \Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_l, 0, \ldots, 0\} \in \mathbb{R}^{n,m} \) und unitären \( U \in \mathbb{R}^{n,n}, V \in \mathbb{R}^{m,m} \) gegeben ist, ist insbesondere eine \( R^l(n, m) \)-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 (\( R^k \)-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 \( R(l,n,m) \)-Matrix
\[ M = \sum_{i=1}^{l} a^i (e^i)^T \quad \text{bzw.} \quad M = \sum_{i=1}^{l} e^i (b^i)^T \]
mit den Spaltenvektoren \( a^i \) bzw. Zeilenvektoren \( b^i \) von \( M \) darstellen.

### 2.2 Auswertung

Die Matrix-Vektor-Multiplikation einer \( R^k(n, m) \)-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 \( N_{R^k,V}(n,m) \) (bzw. \( N_{V,R^k}(n,m) \)) bezeichnet und ist
\[ N_{R^k,V}(n,m) = 2k(n+m) - k - n \]
\[ N_{V,R^k}(n,m) = 2k(n+m) - k - m. \]
2.3 Multiplikation

Zur Multiplikation einer $R_k(n, m)$-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:

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

$$FM = F (\sum_{i=1}^{k} a^i (b^i)^T) = \sum_{i=1}^{k} (Fa^i)(b^i)^T.$$

Der Aufwand für die Multiplikation einer $R_k(n, m')$- mit einer $R_k(m', m)$-Matrix wird mit $N_{R_k,R_k}(n, m', m) = 2k^2(m' + \min\{n, m\}) - k^2 - k \min\{n, m\}$ bezeichnet und liegt bei

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

2.4 Darstellungswechsel und Konvertierungen

Wir haben bereits gesehen, daß die $R_{\leq k}$-Matrizen als $R_k$-Matrizen dargestellt werden können. Es gilt nun, für eine beliebige Matrix $M$ eine Approximation (und Darstellung) in $R_k$ 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 $R_{\leq k}$-Matrizen liefert.

2.4.1 Gekürzte Singulärwertzerlegung

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

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

von $M$.

**Definition 2.8 (Gekürzte Singulärwertzerlegung)**

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

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

17
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 \( R_{\tilde{k}} \)-Matrizen:

\[
\min_{\text{rang}(B) \leq \tilde{k}} \|M - B\|_2 = \|M - \tilde{M}\|_2 = \sigma_{k+1},
\]

\[
\min_{\text{rang}(B) \leq \tilde{k}} \|M - B\|_F = \|M - \tilde{M}\|_F = \sqrt{\sum_{i=\tilde{k}+1}^{k'} \sigma_i^2}.
\]

**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

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

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

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

\[
\|M - B\|_2 \geq \|(M - B)z^{\tilde{k}+1}\|_2 = \|Mz^{\tilde{k}+1}\|_2 = \sqrt{\sum_{i=1}^{\tilde{k}+1} \sigma_i (v^i)^Tz^{\tilde{k}+1)^2} \geq \sigma_{k+1} \sqrt{\sum_{i=1}^{\tilde{k}+1} ((v^i)^Tz^{\tilde{k}+1)^2} = \sigma_{k+1}.
\]

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
\[ z^j \in \text{span}\{y^{\tilde{k}+1}, \ldots, y^m\} \cap \text{span}\{v^1, \ldots, v^j\} \text{ und } \|(M - B)z^j\|_2 \geq \sigma_j \text{ für } j = \tilde{k} + 1, \ldots, k'. \]

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

\[ \dim(W_j) \geq j - \tilde{k}, \quad z^j \in W_j, \quad i = \tilde{k} + 1, \ldots, j - 1. \]

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

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

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:

\[ \|M - B\|_F = \|M - B\|_F = \sqrt{\sum_{i=1}^{n} \|M - B Ze\|_2^2} \geq \sqrt{\sum_{i=k+1}^{k'} \sigma_i^2}. \]

\[ \text{Bemerkung 2.11 (Aufwand der (gekürzten) Singulärwertzerlegung)} \]

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 \( M \in \mathbb{R}^{n \times m} \) auf \( \mathbb{R}^k(n, m) \) 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 \( \mathbb{R}^k \)-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 $R_k$-Matrizen)

Gegeben sei eine $R_k(n,m)$-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

$$A = Q_AR_A, \quad Q_A \in \mathbb{R}^{n,k}, R_A \in \mathbb{R}^{k,k}$$
$$B = Q_BR_B, \quad Q_B \in \mathbb{R}^{m,k}, R_B \in \mathbb{R}^{k,k}$$

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])

<table>
<thead>
<tr>
<th>Operation</th>
<th>Komplexität</th>
</tr>
</thead>
<tbody>
<tr>
<td>QR-Zerlegung von $A$</td>
<td>$4nk^2$</td>
</tr>
<tr>
<td>QR-Zerlegung von $B$</td>
<td>$4mk^2$</td>
</tr>
<tr>
<td>Multiplikation von $R_A R_B^T$</td>
<td>$2k^3$</td>
</tr>
<tr>
<td>SVD von $R_A R_B^T$</td>
<td>$21k^3$</td>
</tr>
<tr>
<td>Multiplikation von $Q_A U, Q_B V$</td>
<td>$nk^2 + mk^2$</td>
</tr>
</tbody>
</table>

Gesamt: $N_{R_k, SVD}(n,m) = \frac{5(n+m)k^2}{2} + 23k^3$  

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 $v^1, \ldots, v^k$ 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 $M v^i$ 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 $R_k$-Matrix

$$M = \sum_{i=1}^{k} a^i (b^i)^T \in \mathbb{R}^{n,m}$$
und $k' := \text{rang}(M)$. Gesucht ist eine QR-Zerlegung

$$M = QRII = \sum_{i=1}^{k'} q_i^j (r_i^j)^T \Pi$$

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

$$\tilde{M} := \sum_{i=1}^{k} q_i^j ((r_i^j)^T \Pi).$$

Einen Algorithmus zur Berechnung der gekürzten QR-Zerlegung findet man in [29] (Gram-Schmidt-Version). Um den Aufwand für $R_k$-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 $\Pi$ gespeichert, zu Beginn sind $R := 0$ und $Q := 0$. Zuerst werden die Normen $v_j := \|M_j\|_2^2$, $j = 1, \ldots, m$, der Spalten von $M$ berechnet und wir setzen $p_i := i$, $i = 1, \ldots, m$. Nacheinander werden nun die Vektoren $q_i^j$ und $r_i^j$ 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 $M - \sum_{i=1}^{k'-1} q_i^\nu ((r_i^\nu)^T \Pi)$ 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 $r_i^j$ und $r_i^j$.
3. (Q um eine Spalte erweitern) Setze $q_i^j := M_{p_i} - \sum_{\nu=1}^{i-1} q_i^\nu r_i^\nu$.
4. (Diagonalelement von R berechnen) Berechne $r_i^i := \|q_i^j\|_2$. Abbruch, falls $r_i^i \approx 0$.
5. (Q normieren) Normalisiere $q_i^j := q_i^j / r_i^i$.
6. (Rest von R berechnen) Setze $r_i^\nu := (q_i^\nu)^T M_{p_\nu}$ 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 $R_k(n, m)$-Matrix bei $O(k^2(n + m))$:

<table>
<thead>
<tr>
<th>Aufwand</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2k^2(n + m) - k - m + (2k - 1)m$</td>
<td>Spaltennormen berechnen</td>
</tr>
<tr>
<td>$\frac{1}{2}k^2$</td>
<td>Pivotindex bestimmen</td>
</tr>
</tbody>
</table>
$k^2$  
$3k^2n$  
$2kn$  
$kn$  
$k^2(n + m) - \frac{1}{2}(k^2 + kn)$  
$k^2$  

$R$ pivotieren  
Spalte von $Q$ berechnen  
Diagonale von $R$  
$Q$ normieren  
Rest von $R$ berechnen  
Spaltennorm aktualisieren

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

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

$$\|M - M^{QR}\|_2 = n^2,$$
$$\|M - M^{SVD}\|_2 = n.$$
Folgerung 2.15 (Zeilenpivotwahl zum Kürzen)
In Beispiel 2.14 sieht die komplette QR-Zerlegung von $M$ folgendermaßen aus:

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

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 $R_{2(n,n)}$-Matrix

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

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

$$M^{QR} = \begin{bmatrix}
\frac{1}{\sqrt{3}} & \frac{-2}{\sqrt{6}} \\
\frac{1}{\sqrt{3}} & \frac{4}{\sqrt{6}} \\
\frac{1}{\sqrt{3}} & \frac{4}{\sqrt{6}} \\
\frac{1}{\sqrt{3}} & \frac{4}{\sqrt{6}} \\
\vdots & \vdots \\
0 & 0
\end{bmatrix} \begin{bmatrix}
\sqrt{3} & 2\sqrt{3} & 0 & \cdots & 0 \\
\frac{2}{3}\sqrt{3} & \frac{2}{3}\sqrt{3} & 0 & \cdots & 0
\end{bmatrix},$$

so daß die erste Zeile zur Approximation genommen wird. Der Approximationsfehler ist dann $\sqrt{5}/3 \approx 0.816$, 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 $R_{k(n,m)}$-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 $R_{k}$-Matrix approximiert werden. In diesem Fall kann die Approximation jedoch sehr ungenau sein. Um eine $R_{k}$-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 \( s \mathbb{R}^{k}(n,m) \)-Matrizen

\[
M^{(\nu)} = \sum_{i=1}^{k_{\nu}} a_{(\nu)}^{i}(b_{(\nu)}^{i})^{T}, \quad \nu = 1, \ldots, s.
\]

ergibt im allgemeinen eine \( \mathbb{R}^{k}(n,m) \)-Matrix mit Rang \( k = \sum_{i=1}^{s} k_{\nu} \). Dies erfordert lediglich ein Kopieren der Vektoren \( a, b \) in die gewünschte Zielmatrix. Mit \( \oplus : \mathbb{R}^{k}(n,m) \times \mathbb{R}^{k}(n,m) \rightarrow \mathbb{R}^{k}(n,m) \) bezeichnen wir die formatierte Addition, welche das Ergebnis der Addition einer \( \mathbb{R}^{k'}(n,m) \)- und einer \( \mathbb{R}^{k''}(n,m) \)-Matrix auf eine \( \mathbb{R}^{k}(n,m) \)-Matrix kürzt. Implizit wird hier die Wahl einer Konvertierung (zum Kurzen) 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 \( \mathbb{R}^{k}(n,m) \)-Matrizen. Der Aufwand der formatierten Addition innerhalb der \( \mathbb{R}^{k}(n,m) \)-Matrizen ist gemäß Algorithmus 2.12 beschränkt durch

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

2.6 Spektral- und Frobeniusnorm

Von einer \( \mathbb{R}^{k}(n,m) \)-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

\[
N_{\mathbb{R}^{k},\|\|}(n,m) = 4(n + m)k^{2} + 23k^{3}.
\]

2.7 Komplexwertige \( \mathbb{R}^{k} \)-Matrizen

Die Arithmetik komplexwertiger \( \mathbb{R}^{k} \)-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:
  \[
  N_{\mathbb{C}^{k},V} = 8k(n + m) - 2k - 2n, \quad N_{V,\mathbb{C}^{k}} = 8k(n + m) - 2k - 2m.
  \]

- Multiplikation:
  \[
  N_{\mathbb{C}^{k},\mathbb{C}^{k}}(n,m') = 8k^{2}(m' + \min\{n,m\}) - 2k^{2} - 2k \min\{n,m\}.
  \]
• 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 $\mathbb{R}^k$-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 $H$-Arithmetik, die wir definieren wollen. Ihre Erzeugung gliedert sich in zwei Teile:


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

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

2. Zu jedem Knoten $v \in V \setminus \{\text{root}(T)\}$ gibt es einen Weg $(v_i)_{i=0}^n$ der Länge $n \in \mathbb{N}$ von der Wurzel $v_0 = \text{root}(T)$ 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 $q \in T$.
- $T^{(0)} := \{\text{root}(T)\}$ 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 $T^{(i)}$.
- $L(T) := \{q \in T \mid \forall v \in V : (q,v) \notin E\}$ sind die Blätter des Baumes.
- $L(T,i) := L(T) \cap T^{(i)}$ sind die Blätter des Baumes auf der Stufe $i = 0, \ldots, p_T$.
- $L(T, \leq i) := L(T) \cap (T^{(i)} \cup \cdots \cup T^{(0)})$ 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 \( V \subseteq \mathcal{P}(I) \setminus \{\emptyset\} \) (\( \mathcal{P} \) = Potenzmenge) und \( \text{root}(T) = I \) einen hierarchischen Partitionsbaum, oder kurz \( \mathcal{H} \)-Baum, der Menge \( I \), falls für alle \( t \in T \setminus L(T) \) gilt:
\[
t = \bigcup_{s \in S(t)} s.
\]
Mit \( L_T := \{ l \in \mathbb{N}_0 \mid L(T, l) \neq \emptyset \} \) bezeichnen wir die Menge der Stufen von \( T \), auf denen sich Blätter befinden.

Bemerkung 3.4 (\( \mathcal{H} \)-Baum bildet stufenweise Partition)
Set \( T \) ein \( \mathcal{H} \)-Baum von \( I \) und setze \( P^{(i)} := T^{(i)} \cup L(T, \leq i - 1) \) für \( i = 0, \ldots, p_T \). Dann gilt für jede Stufe \( i \in \{0, \ldots, p_T\} \) des Baumes
\[
I = \bigcup_{t \in P^{(i)}} t,
\]
d. h. jede Stufe definiert eine Partition \( P^{(i)} \) der Indexmenge. Allgemeiner gilt für jeden Teilbaum \( T' \) von \( T \), der die Wurzel \( I \) enthält, \( I = \bigcup_{t \in L(T')} t \).

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 \( p = \min\{p_T, p_J\} \) 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 \( i = 0, \ldots, \min\{p_T, p_J\} \)
\[
(T_I \otimes T_J)^{(i)} = T_I^{(i)} \times T_J^{(i)}.
\]

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.
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 Kolokationspunkte 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 \( I_1^{(1)}, J_2^{(1)} \)) 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 \( I_1^{(p)} \) 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 $j \in I_1^{(p)}$ 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)
   \[ \{ j \in I_1^{(p)} \mid \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle \} \approx \{ j \in I_1^{(p)} \mid \langle e, m_j \rangle > \langle e, m_{j^*} \rangle \} \]
   
   - (geometrisch balanciert)
   \[ \text{diam}(\{ m_j \mid j \in I_1^{(p)}, \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle \}) \]
   \[ \approx \text{diam}(\{ m_j \mid j \in I_1^{(p)}, \langle e, m_j \rangle > \langle e, m_{j^*} \rangle \}) \]

   ist und setze
   
   \[ I_2^{(p+1)} := \{ j \in I_1^{(p)} \mid \langle e, m_j \rangle \leq \langle e, m_{j^*} \rangle \}, \quad I_2^{(p+1)} := \{ j \in I_1^{(p)} \mid \langle e, m_j \rangle > \langle e, m_{j^*} \rangle \} \]
   
   (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 gleichmäßig 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 $\rho_j$ von $\Omega_1^{(p)}$ projiziert auf die $j$-te Koordinatenachse zu bestimmen und $e := e_j$ korrespondierend zum größten $\rho_j$ zu setzen. Einlelementige 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 Indezmenge $I$ sei mit dem BSP-Algorithmus kardinalitätsbalanciert erzeugt worden. Dann gilt:

29
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.


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

$$I_1^{(0)} = \{1, \ldots, n\},$$
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 \(I^{(0)}_1 = \{1, \ldots, n\}\). Auf der ersten Stufe wird die Punktmenge in die mit den kleineren und die mit den größeren \(x\)-Koordinaten aufgeteilt:

\(I^{(1)}_1 = \{(i,j) \mid j \in \{1, \ldots, 2^{p_T}\}, i \in \{1, \ldots, 2^{p_T} - 1\}\},\)

\(I^{(1)}_2 = \{(i,j) \mid j \in \{1, \ldots, 2^{p_T}\}, i \in \{1 + 2^{p_T} - 1, \ldots, 2^{p_T/2}\}\}.

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:

\[
I^{(p)}_1 = \left\{(i,j) \mid i \in \sum_{\nu=0}^{\left\lfloor \frac{p-1}{2} \right\rfloor} 2^{p_T - \nu} \beta_2 + \left[1, 2^{p_T - p}\right], j \in \sum_{\nu=0}^{\left\lfloor \frac{p-2}{2} \right\rfloor} 2^{p_T - 1 - \nu} \beta_{2\nu+1} + \left[1, 2^{p_T - p+1}\right]\right\}.
\]

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

\[
I^{(p)}_1 = \left\{(i,j,r) \mid i \in \sum_{\nu=0}^{\left\lfloor \frac{p-4}{3} \right\rfloor} 2^{p_T - \nu} \beta_3 + \left[1, 2^{p_T - p}\right], j \in \sum_{\nu=0}^{\left\lfloor \frac{p-2}{3} \right\rfloor} 2^{p_T - 1 - \nu} \beta_{3\nu+1} + \left[1, 2^{p_T - p+1}\right], r \in \sum_{\nu=0}^{\left\lfloor \frac{p-1}{3} \right\rfloor} 2^{p_T - 1 - \nu} \beta_{3\nu+2} + \left[1, 2^{p_T - p+2}\right]\right\}.
\]
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.


3. In [26] wird der \( \mathcal{H} \)-Baum aus einer vorhandenen Gitterhierarchie extrahiert, indem für jedes Element \( \tau \) einer Diskretisierungsstufe die Indizes des entsprechenden Knotens des \( \mathcal{H} \)-Baumes aus den Söhnen von \( \tau \) 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 \( \mathbb{R}^k \)-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 \( \mathbb{R}^k \)-Darstellung des korrespondierenden Blockes der Matrix möglich ist, wird über die sogenannte Zulässigkeitsbedingung

\[
Z : T_{I \times J} \rightarrow \{ \text{„zulässig“, „nicht zulässig“} \}
\]

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).

33
Definition 3.16 (η-zulässig, $Z_\eta$)
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 $\sigma := \bigcup_{j \in s} \sigma_j$
\[
\min \{ \text{diam} (\tau), \text{diam} (\sigma) \} \leq 2 \eta \text{dist} (\tau, \sigma) \tag{9}
\]
ist. Entsprechend nennen wir ein Mengenprodukt $\tau \times \sigma$ zulässig, wenn (9) gilt. Die Standard-Zulässigkeitsbedingung $Z_\eta$ (zum Parameter $\eta$) ist definiert als
\[
Z_\eta(r \times s) := \begin{cases} 
"zulässig" & \text{falls } r \times s \text{ η-zulässig ist}, \\
"nicht zulässig" & \text{sonst.}\end{cases} \tag{10}
\]
Hier bezeichnet diam($M$) das Infimum der Durchmesser aller Kugeln, die die Menge $M$ enthalten, und dist($\tau, \sigma$) das Infimum der Abstände aller Punkte aus $\tau$ zu Punkten aus $\sigma$. Die Wahl des Parameters $\eta$ 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 $\tilde{\text{diam}}$ und $\tilde{\text{dist}}$ 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 $\eta \in \mathbb{R}_{>0}$
\[
\tilde{Z}_\eta^{-1}(\text{"zulässig"}) \subseteq Z_\eta^{-1}(\text{"zulässig"})
\]
ist, also unter der Standard-Zulässigkeitsbedingung $Z_\eta$ 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
\[
\tilde{Z}_{\tilde{\eta}}^{-1}(\text{"zulässig"}) \supset Z_\eta^{-1}(\text{"zulässig"})
\]
extistiert. Diese Eigenschaft stellt sicher, daß ein unter $\tilde{Z}_\eta$ nicht zulässiger Knoten auch unter $Z_\eta$ 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
\[
Q(\tau) := \left\{ x \in \mathbb{R}^d \mid \inf_{y \in \tau} y_i \leq x_i \leq \sup_{y \in \tau} y_i \right\}.
\]

Definition 3.19 (Einfache Zulässigkeitsbedingung $Z_\eta^{\text{simple}}$)
Die approximativen Distanzen und Durchmesser definieren wir für beschränkte Mengen $\tau, \sigma \subset \mathbb{R}^d$ durch
\[
diam^{\text{simple}}(\tau) := \text{diam}(Q(\tau)),
\]
\[
dist^{\text{simple}}(\tau, \sigma) := \text{dist}(Q(\tau), Q(\sigma))
\]
und die approximative Zulässigkeitsbedingung $Z_\eta^{\text{simple}} : T_I \otimes T_J \rightarrow \{\text{"zulässig"}, \text{"nicht zulässig"}\}$ für $\eta \in \mathbb{R}_{>0}$ durch (9),(10) mit $\text{diam}^{\text{simple}}, \text{dist}^{\text{simple}}$ anstelle von $\text{diam}, \text{dist}$. 

34
Bemerkung 3.20 (Ineffizienz von $Z_{\eta}^{\text{simple}}$)
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 $\tau$ unter $Z_{\eta}$ zulässig, aber für kein $\eta \in \mathbb{R}_{>0}$ unter der zu scharfen Zulässigkeitsbedingung $Z_{\eta}^{\text{simple}}$ zulässig.

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

$$
diam^{\text{eff}}(\tau) := \text{diam}(Q(\tau)),
$$
$$
dist^{\text{eff}}(\tau, \sigma) := \begin{cases} 
\text{diam}(Q(\tau), \sigma) & \text{falls } diam^{\text{eff}}(\tau) \leq diam^{\text{eff}}(\sigma) \\
\text{diam}(\tau, Q(\sigma)) & \text{falls } diam^{\text{eff}}(\tau) > diam^{\text{eff}}(\sigma)
\end{cases}
$$

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

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

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

d.h. $Z_{\eta}^{\text{eff}}$ 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

$$
\min \{\text{diam}(\tau), \text{diam}(\sigma)\} \leq 2\tilde{\eta}\text{dist}(\tau, \sigma)
$$

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

$$
\text{diam}^{\text{eff}}(\tau) = \sqrt[2]{\sum_{i=1}^{d} \left( \max_{x \in \tau} x_i - \min_{x \in \tau} x_i \right)^2} \leq \sqrt{d}\text{diam}(\tau)
$$

und

$$
\text{dist}^{\text{eff}}(\tau, \sigma) = \text{dist}(Q(\tau), \sigma) \geq \text{dist}(\tau, \sigma) - \text{diam}^{\text{eff}}(\tau)
$$

folgt
\[
\frac{1}{\sqrt{d}} \text{diam}^{\text{eff}}(\tau) = \min \left\{ \frac{1}{\sqrt{d}} \text{diam}^{\text{eff}}(\tau), \frac{1}{\sqrt{d}} \text{diam}^{\text{eff}}(\sigma) \right\} \\
\leq \min \{ \text{diam}(\tau), \text{diam}(\sigma) \} \\
\leq 2\eta \text{dist}(\tau, \sigma) \\
\leq 2\eta (\text{dist}^{\text{eff}}(\tau, \sigma) + \text{diam}^{\text{eff}}(\tau)), \\
\left(1 - \frac{2\eta}{1 + 2\eta}\right) \text{diam}^{\text{eff}}(\tau) \leq \frac{2\eta}{1 + 2\eta} \text{dist}^{\text{eff}}(\tau, \sigma), \\
\min \{ \text{diam}^{\text{eff}}(\tau), \text{diam}^{\text{eff}}(\sigma) \} \leq 2\eta \text{dist}^{\text{eff}}(\tau, \sigma).
\]

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 $\mathcal{H}_x$-Baum $T_{I \times J}$ mit logarithmis-clinear- 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 $\sigma, \tau$ haben die spezielle Struktur

\[
\tau = \bigcup_{i \in r} \tau_i, \quad \sigma = \bigcup_{j \in s} \sigma_j.
\]

Die Elemente $\tau_i, \sigma_j$ 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 $\sigma$ zu $\tau$ läßt sich dann mit quadratischem Aufwand $O(|r| + |s|)^2$ durchführen. Zur Verringerung der Kosten kann man die approximative Zulässigkeitsbedingung $Z_\eta^{\text{eff}}$ verwenden, ohne das Risiko einzugehen, nicht zulässige Knoten als zulässig anzuerkennen, und ohne die Effizienz zu verlieren. Die Auswertung von $Z_\eta^{\text{eff}}$ läßt sich mit $O(|r| + |s|)$ Berechnungen des Abstandes der Elemente aus $\tau, \sigma$ 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 $Z_\eta^{\text{simple}}$)

Für das Paneel-Clusterungs-Verfahren wird jeweils die Zulässigkeit einer Menge $\tau_i$ (Kolkationspunkt oder Träger einer Basisfunktion) zu einem Cluster $\sigma = \bigcup_{j \in s} \sigma_j$ 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 $Z_\eta^{\text{simple}}$ 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 \( \mathcal{H} \)-Baum von \( I \times J \) mit möglichst wenig Blättern, die alle zulässig sind oder eine vorgegebene Mächtigkeit unterschreiten. Einen solchen \( \mathcal{H} \)-Baum nennen wir \textit{minimal zulässig}. Die Erzeugung eines minimal zulässigen \( \mathcal{H} \)-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.

\textbf{Definition 3.25} \textit{(Aus \( T_I, T_J \) gebildeter \( \mathcal{H} \)-Baum, zulässiger \( \mathcal{H} \)-Baum)}

Sei \( T = (V', E') \) ein \( \mathcal{H} \)-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:

\[ V' \subset V, \quad E' \subset E. \]

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

\[ Z(\sigma \times \tau) = \text{"zulässig" oder } |\sigma \times \tau| \leq b_{\text{min}}. \quad (11) \]

\textbf{Beispiel 3.26} \textit{(Berechnung des minimal zulässigen aus \( T_I, T_J \) gebildeten \( \mathcal{H} \)-Baumes)}

Gegeben sei eine Zulässigkeitsbedingung \( Z \) und eine Blockgröße \( b_{\text{min}} \in \mathbb{N} \). \( T \) heißt minimal zulässig aus \( T_I, T_J \) gebildet (bzgl. \( Z \) und \( b_{\text{min}} \)), falls \( T \) unter allen zulässigen aus \( T_I, T_J \) gebildeten \( \mathcal{H} \)-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

\[ Z(I \times J) = \text{"zulässig" oder } |I \times J| \leq b_{\text{min}}, \]

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 \( \mathcal{H} \)-Baum. Dies kann durch die Voraussetzung \( p_{T_I} = p_{T_J} =: p \) (alle Bäume besitzen dieselbe Tiefe) und

\[ \forall \tau \in \mathcal{L}(T_I) \cup \mathcal{L}(T_J) : \quad |\tau| \leq \sqrt{b_{\text{min}}} \quad \land \quad \tau \in T_I^{(p)} \cup T_J^{(p)} \]

ausgeschlossen werden.

Sei \( T \) der minimal zulässige aus \( T_I, T_J \) gebildete \( \mathcal{H} \)-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 $\mathcal{H}_0$-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 $\mathcal{H}_0$-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 $\mathcal{H}_0$-Bäumen kann es nützlich sein zu wissen, in welcher Struktur das (exakte) Ergebnis liegt. Diese Information kann man direkt aus den $\mathcal{H}_0$-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 $i \in \mathbb{N}_0$

$$B_i := \{\tau \cap \tau' \mid \tau \in T^{(i)} \cup \mathcal{L}(T, \leq i - 1), \tau' \in T'^{(i)} \cup \mathcal{L}(T', \leq i - 1)\}.$$
Die Summe \( T + T' \) der \( \mathcal{H} \)-Bäume mit \((T + T')^{(i)} \subseteq B_i \) für alle \( i \in \mathbb{N}_0 \) wird von der Wurzel an aufsteigend definiert durch \( \text{root}(T + T') := I \) und
\[
S_{T+T'}(\tau \cap \tau') := \begin{cases}
\{ \sigma \cap \sigma' \mid \sigma \in S_T(\tau), \sigma' \in S_{T'}(\tau') \} \setminus \{ \emptyset \} & \tau \notin \mathcal{L}(T) \land \tau' \notin \mathcal{L}(T') \\
\{ \sigma \cap \tau' \mid \sigma \in S_T(\tau) \} \setminus \{ \emptyset \} & \tau \notin \mathcal{L}(T) \land \tau' \in \mathcal{L}(T') \\
\{ \sigma' \cap \tau \mid \sigma' \in S_{T'}(\tau') \} \setminus \{ \emptyset \} & \tau \in \mathcal{L}(T) \land \tau' \notin \mathcal{L}(T') \\
\emptyset & \tau \in \mathcal{L}(T) \land \tau' \in \mathcal{L}(T')
\end{cases}
\]

Beweis der Wohlderdefiniertheit: 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 \( \tau \cap \tau' = \tau \cap \tilde{\tau}' \cap \tilde{\tau} \cap \tilde{\tau}' \subseteq \tau \cap \tilde{\tau} \) folgt \( \tau \cap \tilde{\tau} \neq \emptyset \) und aus \( \tau \cap \tau' = \tau \cap \tilde{\tau}' \cap \tilde{\tau} \cap \tilde{\tau}' \subseteq \tau' \cap \tilde{\tau}' \) 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 \( \mathcal{H}_\kappa \)-Bäumen (Tiefe 1)

Seien \( T, T' \) zwei \( \mathcal{H}_\kappa \)-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 \( \mathbb{R}_k \)-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 \( T_I, T_J \) gebildet wurde und \( T_I, T_J, T_J \) binärer Bäume der Tiefe \( \geq 1 \) sind. Das Produkt läßt sich dann wie folgt blockweise beschreiben:

In einem Block \( \Box+\Box \) ist das Ergebnis eine \( \mathbb{R}_k \)-Matrix, falls bei jedem der beiden Summanden \( \Box \) und \( \Box \) einer der Faktoren eine \( \mathbb{R}_k \)-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 $H_\sigma$-Bäume jeweils den $H$-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 $H_\sigma$-Bäumen)**

Seien $T_I, T_J, T_J'$ $H$-Bäume, $T'$ ein aus $T_I, T_J$ gebildeter $H_\sigma$-Baum. Das Produkt $T \cdot T'$ der $H$-Bäume wird von der Wurzel an aufsteigend definiert durch \( (T \cdot T')_0 := I \times I' \) und

\[
S_{T,T'}(\tau \times \tau') := \{ \sigma \times \sigma' \mid \exists \tilde{\tau}, \tilde{\sigma} \in T : \sigma \times \tilde{\sigma} \in S_T(\tau \times \tilde{\tau}) \land \tilde{\sigma} \times \sigma' \in S_{T'}(\tilde{\tau} \times \tau') \}.
\]

**Bemerkung 3.30 (Produkt ist $H_\sigma$-Baum)**

Das Produkt aus Definition 3.29 ist ein aus $T_I, T_J'$ gebildeter $H_\sigma$-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 $\tau' \in T_J^{(i)}$, also $T \cdot T'$ aus $T_I, T_J'$ gebildet. Zu zeigen bleibt, daß $T \cdot T'$ ein $H$-Baum ist. Sei also $\tau \times \tau'$ kein Blatt von $T \cdot T'$. Dann gibt es $\tilde{\tau} \in T_J$ derart, daß $\tau \times \tilde{\tau}$ Knoten von $T$ und $\tilde{\tau} \times \tau'$ Knoten von $T$ und $T'$ sind, die beide keine Blätter sind. Entsprechend gehören alle Söhne $T$ bzw. $T'$ und somit $\sigma \times \sigma'$ zu $T \cdot T'$ für alle $\sigma \in S_{T_J}(\tau), \sigma' \in S_{T'_J}(\tau')$, d.h.

\[
\tau \times \tau' = \bigcup_{\sigma \times \sigma' \in S_{T,T'}(\tau \times \tau')} \sigma \times \sigma'.
\]

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 $H_\sigma$-Bäumen)**

Die $H$-Bäume $T_I, T_J, T_J'$ 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 $H_\sigma$-Bäume $T, T'$ sind die minimal zulässigen aus $T_I, T_J$ und $T_J, T_J'$ gebildeten $H_\sigma$-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.

40
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, H-Matrix, darstellbar)
Seien $I, J$ zwei Mengen, $T$ ein $\mathcal{H}_k$-Baum von $I \times J$, $Z$ eine Zulässigkeitsbedingung auf $T$ und $k : \mathcal{L}(T) \rightarrow \mathbb{N}_0$ (Rangverteilung).
Eine Matrix $M \in \mathbb{R}^{I \times J}$ heißt 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 $R_k(b)$-Matrix ist (für eine $R_k(b)(b)$-Matrix schreiben wir kurz $R_k(b)$-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 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 $R_{\leq k(b)}$-Matrix ist.

Notation 4.2 (Hierarchische Matrizen, $\mathcal{M}_{\mathcal{H}, k}(T, Z)$)
Die Menge aller H-Matrizen bzgl. eines $\mathcal{H}_k$-Baumes $T$ von $I \times J$, einer Zulässigkeitsbedingung $Z$ auf $T$ und einer Rangverteilung $k : \mathcal{L}(T) \rightarrow \mathbb{N}_0$ 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 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 H-Matrizen in einer anderen Klasse von 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 $R_k$-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 $\mathcal{H}_k$-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 $\mathcal{M}_{H,k'}(T', Z')$ einen minimalen Abstand zu $M$ in der Frobenius-Norm hat (diese nennen wir eine Bestapproximation bezüglich der Frobeniusnorm):

$$
\|M - M'\|_F = \min_{\tilde{M} \in \mathcal{M}_{H,k'}(T', Z')} \|M - \tilde{M}\|_F
$$

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 Singulärwertauflosung (Definition 2.8) für vollbesetzte Matrizen eine Bestapproximation berechnet.

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

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

   Ergänzt man die $b_\nu$-Matrizen $R_\nu$ 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 Singulärwertauflosung (Algorithmus 2.12) für $Rk$-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 $M_{H'}$)**

Mit $M_{H'}$ bezeichnen wir eine Bestapproximation (bezüglich der Frobeniusnorm) von $M$ in $\mathcal{M}_{H,k'}(T', Z')$. Genau genommen müßte man die Bezeichnung $M_{H,k',T',Z'}$ für die Menge aller Bestapproximierenden verwenden, da aber aus dem Zusammenhang klar ist, welche Klasse von $H$-Matrizen gemeint ist, und die Mehrdeutigkeit von $M_H$ 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 $H$-Matrix-Klassen keine Unterräume der Matrizen sind ($\rightarrow$ iterative Verfahren ungünstig). In Abschnitt 6 werden wir für spezielle $H$-Matrizen die Abschätzung

$$
\max_{b \in T'} \|M|_b\|_2 \leq \|M\|_2 \leq C_{sp} p_T \max_{b \in T'} \|M|_b\|_2
$$

zeigen, so daß bis auf den Faktor $p_T$ (Baumtiefe) die Bestapproximation in der Spektrenorm äquivalent zur blockweisen Bestapproximation in der Spektrennorm 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.

Lieg ein 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

\[
\|X - X_\mathcal{H}\|_F \leq \|X - M\|_F + \|M - M_\mathcal{H}\|_F,
\]

\[
\|M - X_\mathcal{H}\|_F \leq 2\|X - M\|_F + \|M - M_\mathcal{H}\|_F.
\]

**Beweis:** \( \|X - X_\mathcal{H}\|_F \leq \|X - M\|_F \leq \|X - M\|_F + \|M - M_\mathcal{H}\|_F. \)


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 \( \mathcal{R}k' \)-Matrix zu approximieren.

Wir fixieren einen \( \mathcal{H} \)-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 \( R \in \mathcal{R}k'(I \times J) \), 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 \( M|_{b_i} \) \( i = 1, \ldots, |S(b)| \) berechnet, so definieren wir die Approximation \( R_b \) von \( M|_b \) als die auf Rang \( k' \) gekürzte Singularwerterlegung der Matrix \( \sum_{i=1}^{|S(b)|} R_{b_i} |b_i \) \( \), 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 \( M \in \mathcal{M}_{\mathcal{H},1}(T, Z) \)

44
Abbildung 11: Im ersten Schritt werden alle vollbesetzten Blöcke \((F)\) in \(Rk'\)-Matrizen \((Rk')\) konvertiert, danach jeweils die Söhne eines Knotens in einer \(Rk'\)-Matrix zusammengefaßt.

(Z ≡ „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
\[
\|R - M\| \leq (2p_T^{i+1} + 1)\|R_{\text{best}} - M\|,
\]
wobei \(R_{\text{best}}\) die Bestapproximation von \(M\) in \(R_{\leq k'}\) und \(\|\cdot\|\) die Frobeniusnorm ist.

Beweis: Mit \(R_i\) bezeichnen wir die durch die Matrizen \(R_{ib}, b \in T^{(i)}\), 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
\[
\|R_i - R_{i-1}\| \leq \|R_i - \tilde{R}\| \quad (12)
\]
für alle \(\tilde{R} \in \mathbb{R}^{I \times J}\) mit \(\tilde{R}_{ib} \in R_{\leq k'}(b)\) für alle \(b \in T^{(i-1)}\) gilt. Insbesondere sind auch
\[
\|R_i - R_{i-1}\| \leq \|R_i - R_{\text{best}}\|, \quad (13)
\]
\[
\|M - R_{p_T}\| \leq \|M - R_{\text{best}}\| \quad (14)
\]
45
Abbildung 12: Die $\mathcal{H}$-Bäume $T_J$ und $T_t$.

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

$$\|R_i - R^{\text{best}}\| \leq \|R_{i+1} - R_i\| + \|R_{i+1} - R^{\text{best}}\|$$

schließt man den Induktionsschritt. Es folgt

$$\|R - M\| = \|\sum_{i=0}^{p_T-1} R_i - R_{i+1} + R_{p_T} - M\|$$

$$\leq \sum_{i=0}^{p_T-1} \|R_i - R_{i+1}\| + \|R_{p_T} - M\|$$

$$\leq \sum_{i=0}^{p_T-1} 2^{p_T-i-1}\|R_{p_T} - R^{\text{best}}\| + \|R^{\text{best}} - M\|$$

$$= 2^{p_T} \sum_{i=0}^{p_T-1} 2^{-i}\|R_{p_T} - R^{\text{best}}\| + \|R^{\text{best}} - M\|$$

$$\leq 2^{p_T}\|R_{p_T} - R^{\text{best}}\| + \|R^{\text{best}} - M\|$$
\[ \leq 2^{pt} \left( \| M - R_p \| + \| M - R_{\text{best}} \| \right) + \| R_{\text{best}} - M \| \]

(14)

\[ \leq (2^{pt+1} + 1) \| M - R_{\text{best}} \|. \]

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 \( \mathcal{H}_k \)-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 \( Z + Z' : T + T' \to \{ \text{zulässig}, \text{nicht zulässig} \} \) durch

\[
\sigma := \begin{cases} 
\text{zulässig} & \text{Zulässigkeit} = \text{Zulässigkeit} = Z'('), \\
\text{nicht zulässig} & \text{sonst}
\end{cases}
\]

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

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

Von besonderem Interesse ist die formatierte Addition

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

die das Ergebnis \( M + M' \) der Addition in eine Matrix \( \tilde{M} \in \mathcal{M}_{\mathcal{H},k'}(\tilde{T}, \tilde{Z}) \) 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 $b \in \mathcal{L}(\tilde{T})$ einzeln durchgeführt. Dazu werden alle Blätter $b \in \mathcal{L}(T)$ und $b' \in \mathcal{L}(T')$ mit $b \cap b' \neq \emptyset$ und $b' \cap b \neq \emptyset$ ermittelt (maximal $p_T, p_{T'}$ Stufen). Mit Hilfe der Fortsetzung erhält man die Darstellung

$$(M + M')|_b = \sum_{b \in \mathcal{L}(T)} M|_{b \cap b} + \sum_{b' \in \mathcal{L}(T') \setminus \mathcal{L}(T)} M'|_{b \cap b'}. $$

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 zu viele Blätter in der Summation nicht zulässig sind, ist es effizienter, erst eine vollbesetzte Summation und anschließend eine Singularwertzerlegung (der vollbesetzten Matrix) zum Kürzen durchzuführen.

2. Zweistufige Approximation: Im ersten Schritt wird die Matrix $M$ nach $\tilde{M} \in \mathcal{M}_{H,k}(\tilde{T}, \tilde{Z})$ konvertiert. Danach wird zu $\tilde{M}$ die Matrix $M'$ addiert und das Ergebnis nach $\mathcal{M}_{H,k}(\tilde{T}, \tilde{Z})$ 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 $s^25(n + m)k^2 + s^323k^3$. 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 $M|_b$ die $\mathbf{R}_k$-Approximationen $R_b|_b, R_{b'}|_b$ 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 $\tau$ enthält. $\tau^{(j)}$ bezeichnen wir als den Vorfahren von $\tau$ auf der $j$-ten Stufe.

**Lemma 4.11** (Ergebnis der Multiplikation)

Gegeben seien zwei $\mathcal{H}$-Matrizen $M, M'$ aus unterschiedlichen Klassen $\mathcal{M}_{H,k}(T, Z)$, $\mathcal{M}_{H,k'}(T', Z')$. $T$ sei ein aus $T_I, T_J$ und $T'$ ein aus $T_J, T_I$ gebildeter $\mathcal{H}_k$-Baum. Für ein Blatt $\tau \times \tau' \in \mathcal{L}(T \cdot T', i)$ (Baumprodukt aus Abschnitt 3.4) definieren wir die Rangverteilung $k \cdot k'$ : $\mathcal{L}(T \cdot T') \rightarrow \mathbb{N}_0$ und Zulässigkeitsbedingung
Z · Z' : \mathcal{L}(T · T') \to \{ \text{"zulässig"}, \text{"nicht zulässig"} \} wie folgt:

\[ U_j := \{ \tilde{\tau} \in T_j^{(j)} \mid \tau^{(j)} \times \tilde{\tau} \in T \land \tilde{\tau} \times \tau^{(j)} \in T' \land (\tau^{(j)} \times \tilde{\tau} \in \mathcal{L}(T) \lor \tilde{\tau} \times \tau^{(j)} \in \mathcal{L}(T')) \}, \quad 0 \leq j \leq i \]

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

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

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

\[ (M \cdot M')|_{\tau \times \tau'} = \sum_{j=0}^{i} \sum_{\tilde{\tau} \in U_j} (M|_{\tau^{(j)} \times \tilde{\tau}} \cdot M'|_{\tilde{\tau} \times \tau^{(j)}})|_{\tau \times \tau'} \quad (15) \]

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

Zwischenbehauptung:

\[ J = \bigcup_{0 \leq j \leq i} U_j \]

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 \( j = j' \): \( T_j^{(j)} \cup \mathcal{L}(T_j, \leq j) \) 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 \( j' \neq j \): O.B.d.A. sei \( j' < j \).

2.a) \( \tau^{(j')} \times \tilde{\tau}_2 \in \mathcal{L}(T) \): Es ist

\[ \emptyset \neq \tau \subset \tau^{(j')} \cap \tau^{(j)} \quad (16) \]

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 \( j \neq j' \) 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 \( T_j^{(j)} \cup \mathcal{L}(T_j, \leq j - 1) \) 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 \( \tilde{\tau}_2 \cap \tilde{\tau}_1 = \emptyset \).

49
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 \( \tau^{(0)} \times \tau^{(0)} = J \times I' \in T' \). 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

\[
(M \cdot M')|_{\tau \times \tau'} = M|_{\tau \times J} \cdot M'|_{J \times \tau'} = \sum_{j=0}^{i} \sum_{\tilde{\tau} \in U_{j}} M|_{\tau \times \tilde{\tau}} \cdot M'|_{\tilde{\tau} \times \tau'}
\]

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

Die formierte Multiplikation

\[
\odot : \mathcal{M}_{H,k}(T, Z) \times \mathcal{M}_{H,k'}(T', Z') \rightarrow \mathcal{M}_{H,k}(\tilde{T}, \tilde{Z})
\]

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

**Bemerkung 4.12 (Berechnung der formierten 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 \( H_{k} \)-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 „\( \tau \) ist Vorfahr von \( \tilde{\tau} \)“ oder „\( \tilde{\tau} \) ist Vorfahr von \( \tau \)“ erfüllen (weil dies in \( T_{I} \otimes T_{I'} \) gilt).

Sei \( M \in \mathcal{M}_{H,k}(T, Z), M' \in \mathcal{M}_{H,k'}(T', Z') \) 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 Singulärwertezerlegung 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 $\mathbb{R}^k$-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 $\mathbb{R}^k$-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 $\mathcal{H}_k$-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 $ mit $S_T(\tau) = \{\sigma_1,\ldots,\sigma_s\}$. 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:

$$R_{ij}^{(0)} := M_{ij}, \quad i,j \in \{1,\ldots,s\}$$

$$L^{(0)} := I.$$
Für $\nu = 1, \ldots, s$ setze

\begin{align*}
R^{(\nu)}_{\nu} &:= I \\
R^{(\nu)}_{\nu j} &:= (R^{(\nu-1)}_{\nu})^{-1} R^{(\nu-1)}_{\nu j} \quad j = \nu + 1, \ldots, s \\
R^{(\nu)}_{\nu i} &:= 0 \quad i = \nu + 1, \ldots, s \\
R^{(\nu)}_{ij} &:= R^{(\nu-1)}_{ij} - R^{(\nu-1)}_{\nu} R^{(\nu)}_{\nu j} \\
L^{(\nu)}_{ij} &:= (R^{(\nu-1)}_{\nu})^{-1} R^{(\nu-1)}_{\nu j} \quad j = 1, \ldots, \nu \\
L^{(\nu)}_{ij} &:= L^{(\nu-1)}_{ij} - R^{(\nu-1)}_{\nu} L^{(\nu)}_{\nu j} \\
&\quad i = \nu + 1, \ldots, s, \quad j = 1, \ldots, \nu.
\end{align*}

Für $\nu = s, \ldots, 2$ setze

\begin{align*}
L^{(2s-\nu+1)}_{ij} &:= L^{(2s-\nu)}_{ij} - R^{(s)}_{ij} L^{(2s-\nu)}_{\nu j} \\
&\quad i = 1, \ldots, \nu - 1, \quad j = 1, \ldots, s.
\end{align*}

**Beispiel 4.14** (Block-Gauß-Elimination bei Binärbäumen)

Set $T_1$ 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 $R^{(\nu)}_{ij}, L^{(\nu)}_{ij}$ zurückführen. Rekursiv erhält man so einen Algorithmus zur Inversion von $M$, der auf die Addition und Multiplikation von $H$-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 $(R^{(\nu)}_{\nu})^{s}_{\nu = 1}$ 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 $(R^{(\nu-1)}_{\nu})^{s}_{\nu = 1}$ 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 $R^{(\nu-1)}_{\nu}$)

Seien $L, R \in \mathbb{R}^{I \times I}$ mit $R = LM$, $R$ eine obere Dreiecksmatrix, $L$ eine untere Blockdreiecksmatrix (Blockung $(T_1 \otimes T_1)^{(i)}$) und $R|_b = I$ (Identität) für alle Diagonalblöcke. Aus der blockweisen Darstellung bzgl. der Partitionierung ($n := |(T_1)^{(i)}|$)

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

52
liest man für die Hauptuntermatrix $M^\sharp$ (Hauptuntermatrix zu den Indizes aus den ersten $j$ Elementen aus $(T_1)^{(i)}$)

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

ab. Multipliziert mit der Inversen von $M^\sharp$ erhält man

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

und erhält so aus der letzten Zeile

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

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

$$(R_{vv}^{(v-1)})^{-1} = ((M^\sharp)^{-1})_{jj} \tag{17}$$

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:

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

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

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

Algorithmus 4.18 (Formatierte Inversion $(\odot)$)
Die formatierte Inversion

$$
(\odot) : \mathcal{M}_{\mathcal{H}, k}(T, Z) \rightarrow \mathcal{M}_{\mathcal{H}, k}(T, Z)
$$

wird folgendermaßen durchgeführt:
In \( R \in \mathcal{M}_{H,k}(T, Z) \) ist die zu invertierende Matrix gespeichert. Nach der Inversion soll \( L \in \mathcal{M}_{H,k}(T, Z) \) die approximative Inverse speichern. \( H \in \mathcal{M}_{H,k}(T, Z) \) 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.

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.

Sei \( S_T(\tau) = \{ \sigma_1, \ldots, \sigma_s \} \) 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

\[
\begin{align*}
L_{\nu \nu} & := R_{\nu \nu} \otimes (H_{\nu \nu} \text{ als Zwischenspeicher}) \\
H_{\nu j} & := L_{\nu \nu} \odot R_{\nu j} \\
R_{\nu j} & := H_{\nu j} \quad j \in \{ \nu + 1, \ldots, s \} \\
H_{ij} & := -R_{\nu \nu} \odot R_{\nu j} \\
R_{ij} & := R_{ij} \oplus R_{\nu j} \quad i, j \in \{ \nu + 1, \ldots, s \} \\
L_{\nu \nu} & := R_{\nu \nu} \odot L_{\nu \nu} \quad i \in \{ \nu + 1, \ldots, s \} \\
L_{\nu j} & := -L_{\nu \nu} \odot L_{\nu j} \quad j \in \{ 1, \ldots, \nu - 1 \} \\
H_{ij} & := -R_{\nu \nu} \odot L_{\nu j} \\
L_{ij} & := L_{ij} \oplus H_{ij} \quad i \in \{ \nu + 1, \ldots, s \}, j \in \{ 1, \ldots, \nu - 1 \},
\end{align*}
\]

Für \( \nu = s, \ldots, 2 \) setze

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


**Bemerkung 4.19 (Notwendige Bedingungen für die Block-Gauß-Elimination)**

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

\[
\begin{align*}
\| R|_{\tau \times \tau} - R_H \|_F & \leq \varepsilon A \lambda_{\text{max}}, \\
\| (R|_{\tau \times \tau})^{-1} - R_{H^{-1}} \|_F & \leq \varepsilon A \lambda_{\text{min}}^{-1}
\end{align*}
\]

54
(\(R_H\) ist eine Bestapproximation von \(R_{|\tau \times \tau}\) in \(M_{H,k}(T_{|\tau \times \tau}, Z)\) und \(R_H^{-1}\) eine Bestapproximation von \((R_{|\tau \times \tau})^{-1}\) in \(M_{H,k}(T_{|\tau \times \tau}, Z)\)) 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 \(n^2\) beschränkt ist. Es wird aber stets angemerkt, daß (noch) keine reelle Matrix bekannt ist, für die die Fehlerverstärkung bei Total pivotsuche 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 \(H\)-Matrix-Formate zu höheren neuen Rundungsfehlern führen (→ Lemma 4.5).

3. Die Fehlerverstärkung ist nur durch \(n^2\) 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:

\[
\|M^{-1} - M^{\oplus}\|_F = O(n^c)\varepsilon_A
\]

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 \(H\)-Inverse (obwohl sie nicht eindeutig ist). Für die Inversion eines nicht weiter bekannten Operators \(M\) empfiehlt sich folgendes Vorgehen: Zuerst wird die \(H\)-Inverse bestimmt und der Fehler

\[
\|I - M^{\oplus} \cdot M\|_2
\]

(vgl. Satz 4.31) ermittelt (die \(H\)-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 \(H\)-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 \(H\)-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 \( M \in \mathcal{M}_{H,k}(T, Z) \) führt zu der Iterationsvorschrift

\[ X^{(i+1)} := 2X^{(i)} - X^{(i)} \cdot M \cdot X^{(i)}, \quad (20) \]

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

**Lemma 4.20 (Konvergenz bei exakter Arithmetik)**

Sei \( M \in \mathcal{M}_{H,k}(T, Z) \) eine invertierbare Matrix, \( \| \cdot \| \) eine beliebige Matrixnorm und \( X^{(0)} \in \mathcal{M}_{H,k}(T, Z) \) 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 \( E^{(i)} := M^{-1} - X^{(i)} \):

\[
X^{(i+1)} &= 2X^{(i)} - X^{(i)} \cdot M \cdot X^{(i)} \\
&= 2M^{-1} - 2E^{(i)} - M^{-1} - E^{(i)}ME^{(i)} + E^{(i)} + E^{(i)} \\
&= M^{-1} - E^{(i)}ME^{(i)}, \\
\| X^{(i+1)} - M^{-1} \| &\leq \| M \|^{-1} q^{2^i} \| M \| \| M \|^{-1} q^{2^i} \\
&= \| M \|^{-1} q^{2^i+1}.
\]

**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 \( \mathcal{M}_{H,k}(T, Z) \). In numerischen Tests zeigt sich sogar, daß sich die Approximationsgüte der \( H \)-Inversen nicht durch eine Newton-Iteration mit \( M^{-1} \) als Startwert verbessern läßt. Der Grund dafür ist, daß das Produkt \( X^{(i)} M \) in \( \mathcal{M}_{H,k}(T, Z) \) nicht hinreichend genau dargestellt werden kann (Störung der Identität mit einer unstrukturierten Matrix).

**Lemma 4.22 (Konvergenz bei formatierter Arithmetik)**

Sei \( M \in \mathcal{M}_{H,k}(T, Z) \) invertierbar, \( Y^{(0)} \in \mathcal{M}_{H,k}(T, Z) \) mit

\[ \| M \|_F \| Y^{(0)} - M^{-1} \|_F = q < \frac{1}{9} \quad (21) \]

56
und für die inverse $M^{-1}$ gelte die Approximationseigenschaft

$$\|M^{-1} - M^{-1}_H\|_F \leq \varepsilon_A \leq \frac{1}{16}\|M\|^{-1}_F,$$

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

$$Y^{(i+1)} := (2Y^{(i)} - Y^{(i)} \cdot M \cdot Y^{(i)})_H$$

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

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

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

$$\|X^{(i)} - Y^{(i)}\|_F \leq \varepsilon_i,$$

$$\varepsilon_0 := 0,$$

$$\varepsilon_i := 2\varepsilon_A + \frac{1}{4}q^{2^{i-1}}\|M\|^{-1}_F, \quad i \in \mathbb{N}$$

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

$$Y^{(i+1)} = (2Y^{(i)} - Y^{(i)} \cdot M \cdot Y^{(i)})_H$$

$$= \left(2X^{(i)} - 2\delta^{(i)} - X^{(i)} \cdot M \cdot X^{(i)}\right)_H$$

$$+ X^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot X^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)}\right)_H$$

$$= \left((X^{(i+1)} - 2\delta^{(i)} + X^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot X^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)})_Hight.$$

$$\left. = (X^{(i+1)} - E^{(i)} \cdot M \cdot \delta^{(i)} - \delta^{(i)} \cdot M \cdot E^{(i)} - \delta^{(i)} \cdot M \cdot \delta^{(i)})_H\right)$$

$$= \left(\underbrace{Y' \cdot \ldots \cdot Y'}_{\text{definiert}}\right)_H,$$

$$\|Y' - X^{(i+1)}\|_F = \|E^{(i)} \cdot M \cdot \delta^{(i)} + \delta^{(i)} \cdot M \cdot E^{(i)} + \delta^{(i)} \cdot M \cdot \delta^{(i)}\|_F$$

$$\leq 2\|E^{(i)}\|_F\|M\|_F\|\delta^{(i)}\|_F + \|\delta^{(i)}\|_F^2\|M\|_F$$

$$\leq 2q^2\varepsilon_i + \|M\|_F\varepsilon_i^2.$$

Für $i = 0$ folgt

$$\|Y^{(1)} - X^{(1)}\|_F = \|X^{(1)}_H - X^{(1)}\|_F$$

$$\leq \frac{L_A^4}{5} \|M^{-1} - X^{(1)}\|_F + \|M^{-1} - M^{-1}_H\|_F$$

$$\leq q^2\|M\|^{-1}_F + \varepsilon_A$$

$$\leq \frac{1}{4}q\|M\|^{-1}_F + \varepsilon_A$$

57
und im Fall \( i \geq 1 \) ist
\[
\| Y^{(i+1)} - X^{(i+1)} \|_F \leq \| Y' - Y^{(i+1)} \|_F + \| Y' - X^{(i+1)} \|_F \\
\leq \| Y' - M^{-1} \|_F + \| Y' - X^{(i+1)} \|_F \\
\leq \| M^{-1} - M^{-1} \|_F + \| M^{-1} - X^{(i+1)} \|_F + 2 \| Y' - X^{(i+1)} \|_F
\]

\[\text{Dreiecksung} \]
\[
\leq 4q^2 \varepsilon_i + 2 \| M \|_F \varepsilon^2_i + q^2q^2q^{-1} \| M \|^{-1}_F + \varepsilon_A
\]
\[
= 8q^2 \varepsilon_A + q^2q^2q^{-1} \| M \|^{-1}_F \\
+ 8 \| M \|_F \varepsilon^2_A + 2 \varepsilon_A q^2q^{-1} + \frac{1}{8}q^2q^2q^{-1} \| M \|^{-1}_F
\]
\[
+ q^2q^2q^{-1} \| M \|^{-1}_F + \varepsilon_A
\]
\[\leq \frac{1}{8} \varepsilon_A + \frac{1}{9}q^2 \| M \|^{-1}_F + \varepsilon_A + \frac{1}{8} \varepsilon_A + \frac{1}{8}q^2 \| M \|^{-1}_F
\]
\[
+ \frac{1}{8}q^2 \| M \|^{-1}_F + \varepsilon_A
\]
\[
\leq 2 \varepsilon_A + \left( \frac{1}{9} + \frac{1}{8} + \frac{1}{8} \right)q^2 \| M \|^{-1}_F
\]
\[
= 2 \varepsilon_A + \frac{161}{648}q^2 \| M \|^{-1}_F.
\]

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 \( M_{H,k}(T, Z) \). Die Approximationsgüte \( \varepsilon_A \) 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 (\( Y^{(0)} := M^{(0)} \)) erhalten.

Die quadratische Konvergenz der Newton-Iteration scheint zu einem schnellen Inversionsalgorithmus zu führen. Die Komplexität der exakten \( H \)-Multiplikation übersteigt jedoch die der \( H \)-Inversion, so daß es günstiger ist, die \( H \)-Inverse \( M^{(0)} \) in einer angereicherten \( H \)-Struktur zu berechnen, als die formatierte Newton-Iteration in dieser durchzuführen. Die Newton-Iteration wird dann sinnvoll, wenn die Berechnung der \( H \)-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 größeren Diskretisierungsstufe zu erhalten. Die approximative Inverse auf der größeren Stufe erhält man wiederum durch die Prolongation der Inversen einer noch größeren 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 $(M_i)_{i=i_0}^{i_n}$,

• $H$-Matrix-Klassen $M_{H,k_i}(T_i,Z_i), i = i_0,\ldots,i_n$,

• Prolongationen $P^{i,i+1}: M_{H,k_i}(T_i,Z_i) \rightarrow M_{H,k_{i+1}}(T_{i+1},Z_{i+1}), i = i_0,\ldots,i_n - 1$ 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 $M_{H,k_i}(T_i,Z_i)$ wird zu

$$\|M_i^{-1} - (M_i^{-1})_H\|_F \leq \frac{1}{16}\|M_i\|_F^{-1}$$

(26)

und die Approximationsgüte (21) der Startnäherung zu

$$\|P^{i,i+1}(M_i^{-1}_H) - M_{i+1}^{-1}\|_F \leq \frac{1}{18}\|M_{i+1}\|_F^{-1}.$$ 

(27)

Die Konstante $\frac{1}{16}$ 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 $M_{H,k_i}(T_i,Z_i)$ 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

$$-\Delta u = f$$

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 $u \in H^1_0(\Omega)$ so, daß gilt:

$$\forall v \in H^1_0(\Omega): \int_{\Omega} \langle \nabla u, \nabla v \rangle = \int_{\Omega} f v.$$

Problem 4.24 (Hierarchisch diskretisiertes Modellproblem)

Den unendlichdimensionalen Raum $H^1_0(\Omega)$ 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 $\{\phi_{i,j}^{(k)}\}_{j=1}^{n_i}$ aus stetigen und stückweise (auf Dreiecken) affinen Funktionen ($\rightarrow$ 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

$$\phi_{j}^{(i)}(\theta_{k}^{(i)}) = \delta_{jk}$$

($\rightarrow$ 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 $H^1_0(\Omega)$, 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

$$M_ix = f_i$$

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


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 $\tau^{(i)}_j := (\theta^{(i)}_x, \theta^{(i)}_y, \theta^{(i)}_z), x, y, z \in I$. Wir fordern, daß wenigstens ein Dreieck $\tau^{(i)}_j$ die Bedingung $x, y, z \in I$ erfüllt (daß das Dreieck „in I enthalten“ ist). Die lineare Interpolation von Stützpunkten $v_x, v_y, v_z$ zu den Gitterpunkten $(\theta^{(i)}_x, \theta^{(i)}_y, \theta^{(i)}_z)$ bezeichnen wir mit $\Pi_{\tau^{(i)}_j}[v]$. Die Prolongation

$$P^{i+1}_V : \mathbb{R}^I \to \mathbb{R}^J$$

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

$$(P^{i+1}_V[v])_j := \Pi_{\tau^{(i)}_j}[v](\theta^{(i+1)}_j)$$

Ist $I = \{1, \ldots, n_i\}$, so stimmt $P^{i+1}_V$ mit $\Pi_{i+1 \leftarrow i}$ überein.*
Algorithmus 4.26 (Prolongation von $\mathbb{R}^k$-Matrizen)
Sei $R' \in \mathbb{R}^k(I', J')$, $I' \subset \{1, \ldots, n_i\}$, $J' \subset \{1, \ldots, n_i\}$, in der Darstellung
\[
R = \sum_{\nu=1}^{k'} a^\nu b^\nu T
\]
gegeben. Dann definieren wir die Prolongierte $R \in \mathbb{R}^k(I, J)$, $I \subset \{1, \ldots, n_{i+1}\}$, $J \subset \{1, \ldots, n_{i+1}\}$, von $R'$ durch
\[
R := P_{\mathbb{R}^k}^{i,i+1}[R'] := \sum_{\nu=1}^{k} P_{\mathbb{R}^k}^{i,i+1}[a^\nu] \cdot P_{\mathbb{R}^k}^{i,i+1}[b^\nu]^T.
\]

Algorithmus 4.27 (Prolongation von $\mathcal{H}$-Matrizen)
Gegeben sei $M' \in \mathcal{M}_{\mathcal{H},k}(T', Z')$ und wir fixieren eine Zuordnung $Z : \mathcal{L}(T) \rightarrow T'$. Gesucht ist eine Approximation $M \in \mathcal{M}_{\mathcal{H},k}(T, Z)$ an $M'$. Definiere für $b = r \times s \in \mathcal{L}(T), r \neq s,$
\[
M|_b := \left(P_{\mathbb{R}^k}^{i,i+1}[M'|_Z(t)]\right)_{\mathcal{H},k},
\]
wobei in nicht zulässigen Blättern $b$ die kanonische $\mathbb{R}^k$-Darstellung benutzt wird und, falls $b$ zulässig und $Z(b)$ 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 $P_{\mathcal{H}}^{i,i+1}((M_i^{-1})_{\mathcal{H}})$ berechnet, so lassen sich aus
\[
I = M_{i+1}M_i^{-1} \approx M_{i+1}P_{\mathcal{H}}^{i,i+1}((M_i^{-1})_{\mathcal{H}})
\]
die Diagonalelemente mittels
\[
P_{\mathcal{H}}^{i,i+1}((M_i^{-1})_{\mathcal{H}})|_t := (M_{i+1}|_t)^{-1}(I - (M_{i+1}P_{\mathcal{H}}^{i,i+1}((M_i^{-1})_{\mathcal{H}}))|_t)
\]
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 $Z$ definiert ist. Die Zulässigkeitsbedingungen $Z_i, Z_{i+1}$ sind hier, beim Laplaceoperator, die Standard-Zulässigkeitsbedingung $Z_\eta$. Die Bäume $T_i$ werden im folgenden Algorithmus konstruiert.
Algorithmus 4.29 (Geschachtelte Konstruktion der Bäume)
Gegeben sei der $H$-Baum $T^1_{i0}$ für das Gitter zum Raum $X_{i0}$. Sukzessive wird der Baum $T^1_{i+1}$ aus $T^1_i$ konstruiert.

Identische Gitterpunkte einsortieren:
Definiere den Baum $T^{1+1}_{i+1}$ als den Baum $T^1_i$, wobei jeder Index $j$, der zu einem Gitterpunkt $\theta_j^{(i)}$ gehört, durch den entsprechenden Index $j'$ auf der nächsten Stufe zum selben Gitterpunkt $\theta_j^{(i+1)} = \theta_j^{(i)}$ 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 $T^1_{i+1}$ ist nun der minimal aus $T^1_{i+\frac{1}{2}}$ erzeugte $H_x$-Baum. Einem Blatt $r \times s \in T_{i+1}$ wird das eindeutig bestimmte kleinste Element $r' \times s' \in T_1$ 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 $\|P^{i,i+1}(P^{1,i-1}H) - M^{-1}_{i+1}\|_F$, 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 $\varepsilon$ 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 die "zufällige" Startvektor in der Vektoriteration
nicht im Senkrechtraum zum größten Eigenwert von $M^T M$ 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

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

durch:

$$x(0) := \text{random}$$

$$x(i+1) := \frac{Ax(i)}{\|Ax(i)\|_2}$$

$$\lambda(i) := (x(i)^T Ax(i))^i, \quad i = 1, 2, \ldots$$

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 $\lambda_j$ absteigend sortiert seien. Es gilt dann die Darstellung

$$x(0) =: \sum_{j=1}^l \mu_j e_j, \quad \|x(0)\|_2 = 1$$

$$x(i) = \sum_{j=1}^l \frac{\mu_j \lambda_j^i e_j}{\sum_{j=1}^l \mu_j^2 \lambda_j^{2i}}$$

$$\lambda(i) = \frac{\sum_{j=1}^l \mu_j^2 \lambda_j^{2i+1}}{\sum_{j=1}^l \mu_j^2 \lambda_j^{2i}}, \quad i = 1, 2, \ldots$$

Der Koeffizient $\mu_1$ zum ersten Eigenvektor in der Darstellung von $x(0)$ erfüllt für ein $C_\mu$ die Abschätzung $\mu_1^2 \geq (C_\mu l)^{-1}$ (gleichmäßige Zusammensetzung von $x(0)$ aus allen Eigenvektoren). Die Approximationsgüte $\varepsilon$ 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

$$\Leftrightarrow \begin{pmatrix} \sum_{j=1}^l \mu_j^2 \lambda_j^{2i+1} \\ \sum_{j=1}^l \mu_j^2 \lambda_j^{2i+2} \end{pmatrix} \leq \begin{pmatrix} \sum_{j=1}^l \mu_j^2 \lambda_j^{2i+3} \\ \sum_{j=1}^l \mu_j^2 \lambda_j^{2i} \end{pmatrix}$$

$$\Leftrightarrow \sum_{j=1}^l \sum_{\nu=1}^l \mu_j^2 \mu_\nu^2 \lambda_j^{2i+1} \lambda_\nu^{2i+2} \leq \sum_{j=1}^l \sum_{\nu=1}^l \mu_j^2 \mu_\nu^2 \lambda_j^{2i+3} \lambda_\nu^{2i+3}$$

63
\[
\sum_{j=1}^{l} \sum_{\nu=1}^{l} \mu_j \mu_\nu (\lambda_j^{2i+1} \lambda_\nu + \lambda_j^{2i+2} \lambda_\nu) \leq \sum_{j=1}^{l} \sum_{\nu=1}^{l} \mu_j \mu_\nu (\lambda_j^{2i+2} \lambda_\nu + \lambda_j^{2i+3} \lambda_\nu)
\]

\[
\sum_{j=-\nu}^{\nu} \mu_j \mu_\nu (\lambda_j^{2i+1} \lambda_\nu + \lambda_j^{2i+2} \lambda_\nu) \leq \mu_j \mu_\nu (\lambda_j^{2i+2} \lambda_\nu + \lambda_j^{2i+3} \lambda_\nu)
\]

\[
\sum_{j=-\nu}^{\nu} \mu_j \mu_\nu (\lambda_j^{2i+1} \lambda_\nu + \lambda_j^{2i+2} \lambda_\nu) \leq \mu_j \mu_\nu (\lambda_j^{2i+2} \lambda_\nu + \lambda_j^{2i+3} \lambda_\nu)
\]

\[
\sum_{j=-\nu}^{\nu} \mu_j \mu_\nu (\lambda_j^{2i+1} \lambda_\nu + \lambda_j^{2i+2} \lambda_\nu) \leq \mu_j \mu_\nu (\lambda_j^{2i+2} \lambda_\nu + \lambda_j^{2i+3} \lambda_\nu)
\]

Damit ist die Zwischenbehauptung bewiesen, aus ihr folgt \(\|M\|_2 \approx \|M\|_2\). 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

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

und

\[
\sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i} (\lambda_1 - \lambda_j) \leq \lambda_1 \sum_{j=1}^{l} \mu_j^2 \lambda_j^{2i}
\]

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

ergeben sich die äquivalenten Bedingungen

\[
(1 - \frac{\varepsilon}{2})^2 C_{\mu} l \leq \frac{\varepsilon}{2},
\]

\[
i \geq \frac{\log(\varepsilon/2) - \log(C_{\mu} l)}{2 \log(1 - \varepsilon/2)},
\]

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

\[ x^{(0)} = \sum_{j=1}^{l} \mu_j e^j \]

\(\{e^1, \ldots, e^l\}\) Orthonormalbasis, \(\sum_{j=1}^{l} \mu_j^2 = 1\)

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

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

Matrix-Vektor-Multiplikationen beschränkt. Die stochastische Größe \(\mu_1^{-2}\) 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 \(\lambda_1\) 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 \(\varepsilon\) 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 \(A := \text{diag}(1, 2, 3)\) 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 \(N_{2\text{norm}} = 49\) \((C_{\mu} = 10^6/3)\). 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 Basiskurven 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 $\varepsilon$ zu erreichen, ist in der folgenden Tabelle für zunehmende $n$ aufgelistet:

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$n$</th>
<th>1024</th>
<th>4096</th>
<th>16384</th>
<th>65536</th>
<th>262144</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1/4</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1/8</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.7</td>
</tr>
<tr>
<td>1/16</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.09</td>
</tr>
<tr>
<td>1/32</td>
<td></td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0.02</td>
</tr>
<tr>
<td>1/64</td>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>0.004</td>
</tr>
<tr>
<td>1/128</td>
<td></td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>16</td>
<td>0.0009</td>
</tr>
<tr>
<td>1/256</td>
<td></td>
<td>41</td>
<td>33</td>
<td>34</td>
<td>33</td>
<td>0.0002</td>
</tr>
<tr>
<td>1/512</td>
<td></td>
<td>66</td>
<td>57</td>
<td>67</td>
<td>67</td>
<td>0.0006</td>
</tr>
<tr>
<td>1/1024</td>
<td></td>
<td>97</td>
<td>101</td>
<td>129</td>
<td>130</td>
<td>123</td>
</tr>
</tbody>
</table>

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 $n \rightarrow \infty$ 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 $M \in M_{H,k}(T,Z)$, $\text{root}(T) = I \times J$, 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 $\|M|_k\|_F^2 = \sum_{(i,j) \in t} M_{ij}^2$ 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 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 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 (ℒ⁺, ℒ⁻)
Sei T eine ℋ-Partition und Z eine zugehörige Zulässigkeitsbedingung. Dann definieren wir

\[ \mathcal{L}^+(T) := Z^{-1}(\text{"zulässig"}) \cap \mathcal{L}(T) \]
\[ \mathcal{L}^-(T) := Z^{-1}(\text{"nicht zulässig"}) \cap \mathcal{L}(T) \]
\[ \mathcal{L}^+(T,i) := Z^{-1}(\text{"zulässig"}) \cap \mathcal{L}(T,i), \quad i = 0, \ldots, p_T \]
\[ \mathcal{L}^-(T,i) := Z^{-1}(\text{"nicht zulässig"}) \cap \mathcal{L}(T,i), \quad i = 0, \ldots, p_T. \]

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

\[ N_{Rk,St}(n, m) = k(n + m), \]
\[ N_{F,St}(n, m) = nm. \]

Für die Matrix-Vektor-Multiplikation gilt nach Abschnitt 2.2

\[ k(n + m) \leq N_{Rk,V}(n, m) \leq 2k(n + m), \]
\[ nm \leq N_{F,V}(n, m) \leq 2nm. \]

5.1 Speicherbedarf

Definition 5.3 (Schwachbesetzte Blockstruktur)
Ein aus \(T_I, T_J\) gebildeter ℋₜ-Baum \(T\) heißt schwachbesetzt zur Konstante \(C_{sp}\) (\(\ast\)sp" steht für sparse), falls

\[ \forall i \in \{0, \ldots, p_T\} \quad \forall \tau \in T_I^{(i)} : \quad |\{\tau' \in T_J^{(i)} \mid \tau \times \tau' \in T^{(i)}\}| \leq C_{sp}, \]
\[ \forall i \in \{0, \ldots, p_T\} \quad \forall \tau' \in T_J^{(i)} : \quad |\{\tau \in T_I^{(i)} \mid \tau \times \tau' \in T^{(i)}\}| \leq C_{sp}. \]
Bemerkung 5.4 (Zur Schwachbesetztheit)
Die Bedingungen aus Definition 5.3 lassen sich für die Standard-Zulässigkeitsbedingung $Z_\eta$ vereinfachen. Sei dazu $D(\tau)$ die die Geometrie der Basisfunktionen zu den Indizes aus $\tau$ charakterisierende Menge (siehe Abschnitt 3.2.1). Ist $\tau \times \tau'$ ein Knoten von $T$, so kann der Vater-Knoten $\tau \times \tau'$ nicht zulässig sein (für minimal zulässige $H_\infty$-Bäume $T$).
Ein minimal (bzgl. $Z_\eta$) zulässiger $H_\infty$-Baum $T$ ist also schwachbesetzt, falls
\[ \forall i \in \{0, \ldots, p_T\} \forall \tau \in T^{(i)}_J : \left| \left\{ \tau' \in T^{(i)}_J \mid \text{dist}(D(\tau), D(\tau')) \leq \frac{\min(\text{diam}(D(\tau)), \text{diam}(D(\tau')))}{2\eta} \right\} \right| \leq \frac{1}{s} C_{sp}, \]
s = maximale Zahl der Söhne eines Knotens $\tau' \in T_J$, analog für $\tau \in T^{(i)}_J$.

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 $Z_\eta$ auf einer geraden Stufe genau die Paare $\tau, \tau'$ 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 $\tau$ dargestellt. Die Schwachbesetztheit soll bzgl. des Clusters $\sigma$ 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 $\tau$, 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 $\sigma$ von $\tau$ 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 \( \Omega := \bigcup_{i \in I} \tau_i \). \( \Omega \) sei in dem Würfel \( m_\Omega + [0,H]^d \) enthalten, \( m_\Omega \in \mathbb{R}^d \). Die Mengen \( \tau_i \) 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 \( \tau_i \) separierbar in dem Sinne, daß es ein \( h \in \mathbb{R} \) gibt, so daß für alle \( x \in \mathbb{R}^d \) gilt

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

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

\[
y_i := x_i 2^{-l} H, \quad i = 1, \ldots, d, \quad y = (y_1, \ldots, y_d),
\]

\[
I_j^{(l)} := \{ i \in I \mid m_i - m_\Omega \in y + [0,2^{-l} H]^d \}.
\]

Erklärung: Auf der Stufe \( l \) wird der Würfel \( m_\Omega + [0,H]^d \), der das Gebiet \( \Omega \) enthält, in \((2^d)^d\) 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 \( j = \sum_{i=1}^d 2^{(d-i)} x_i \) 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 \( H \)-Baumes \( T_I \) ist \( I = I_0^{(0)} \) und die \( 2^d \) Söhne eines Knotens \( I_j^{(l)} \) für \( j = \sum_{i=1}^d 2^{(d-i)} x_i \) sind

\[
S_{T_I}(I_j^{(l)}) = \left\{ I_j^{(l+1)} \mid j' = \sum_{i=1}^d 2^{(l+1)-(d-i)} x_i', \ x_i' \in \{2x_i, 2x_i + 1\} \right\}.
\]

Der \( H_{\bar{h}} \)-Baum \( T \) sei der wie in Beispiel 3.26 minimal zulässige aus \( T_I \) gebildete \( H_{\bar{h}} \)-Baum. Dann ist \( T \) schwachbesetzt mit der Konstante

\[
C_{\text{sp}} := 2^d (2L + 1)^d, \quad L := \left[ \frac{\sqrt{d}}{2} \eta^{-1} + (\sqrt{d} + 4\eta)\bar{h} 2^{l-1} H^{-1} \eta^{-1} \right] = O(\eta^{-1}).
\]
Die Anzahl der Stufen von $T$ ist für $b_{\text{min}} \geq C_{\text{sep}}$ beschränkt durch

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

**Beweis:** Baumtiefe $p_T$: Nach Definition der $I_{j(i)}$ und der Voraussetzung an die Separabilität der Mengen $\tau_i$ gilt $|I_{j(i)}| \leq C_{\text{sep}}$, falls $2^{-l}H \leq h$, also falls $l \geq \log_2(\frac{H}{h})$.

Schwachbesetztheit $C_{\text{sp}}$: Der Durchmesser von $D(I_{j(i)})$ ist wegen der Lokalität der $\tau_i$ beschränkt durch $\sqrt{d}(H2^{-l} + \bar{h})$. Der Abstand zweier Mengen $D(I_{j(i)}), D(I'_{j(i)})$ wird schichtweise betrachtet:

$$L_1 := \{D(I_{j(i)}') \mid \text{dist}(D(I_{j(i)}'), D(I_{j(i)})) = 0\},$$
$$L_i := \{D(I_{j(i)}') \mid \text{dist}(D(I_{j(i)}'), L_{i-1}) = 0\} \setminus L_{i-1}.$$  

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

$$\min(\text{diam}(D(I_{j(i)}'), \text{diam}(D(I'_{j(i)}))) \leq \sqrt{d}(H2^{-l} + \bar{h}) ,$$
$$2\eta \text{dist}(D(I_{j(i)}'), D(I'_{j(i)})) \geq 2\eta(H2^{-l}i - 2\bar{h}) ,$$

$$\min(\text{diam}(D(I_{j(i)}'), \text{diam}(D(I'_{j(i)}))) \leq 2\eta \text{dist}(D(I_{j(i)}'), D(I'_{j(i)}))$$

$\Leftrightarrow \sqrt{d}(H2^{-l} + \bar{h}) \leq 2\eta(H2^{-l}i - 2\bar{h})$

$\Leftrightarrow \sqrt{d}H2^{-l} + (\sqrt{d} + 4\eta)\bar{h} \leq 2\eta H2^{-l}i$

$\Leftrightarrow \sqrt{d}2^{-l-1}\eta^{-1} + (\sqrt{d} + 4\eta)\bar{h}H^{-1}2^{i-1}\eta^{-1} \leq i.$

70
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 $H$-Matrix.

Setzt man

$$L := \left\lceil \frac{\sqrt{d}}{2} \eta^{-1} + (\sqrt{d} + 4\eta)h2^{l-1}H^{-1}\eta^{-1} \right\rceil,$$

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)**


**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 (diam $> \frac{1}{2}$, 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 Blöcke. Insgesamt sind \( O(n^{1/2}) \) 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 \([0, 2^{-i}] \times [0, 2^{-i}] \backslash [0, 2^{-i-1}] \times [0, 2^{-i-1}]\) 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 bezüglich der Standard-Zulässigkeitsbedingung \( Z_\eta \) minimal zulässiger \( \mathcal{H}_\eta \)-Baum. Die \( \mathcal{H} \)-Bäume \( T_I, T_J \) seien lokal geometrisch balanciert in dem Sinne, daß es eine Konstante \( C_{\text{bal}} \) gibt, so daß für alle \( i \in \{0, \ldots, p_T\}, \tau \in T_I^{(i)}, \tau' \in T_J^{(i)} \)

\[
\text{dist}(D(\tau), D(\tau')) \leq \frac{1}{2\eta} \text{diam}(D(\tau')) \Rightarrow \text{diam}(D(\tau)) \leq C_{\text{bal}} \text{diam}(D(\tau'))
\]

\[
\text{dist}(D(\tau), D(\tau')) \leq \frac{1}{2\eta} \text{diam}(D(\tau)) \Rightarrow \text{diam}(D(\tau')) \leq C_{\text{bal}} \text{diam}(D(\tau))
\]

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

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

\[
E|D(\tau)| \geq \text{diam}(D(\tau))^d, \quad E|D(\tau')| \geq \text{diam}(D(\tau'))^d.
\]

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

Die Anzahl der Söhne eines Knotens sei durch \( s \) beschränkt und von den Mengen \( \tau_i, \sigma_i \) 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 \( C_{\text{sp}} = O(\eta^{-d}) \) 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

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

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

$$M_1 := \{ \tau' \in M \mid \text{diam}(D(\tau)) > \text{diam}(D(\tau')) \},$$
$$M_2 := \{ \tau' \in M \mid \text{diam}(D(\tau)) \leq \text{diam}(D(\tau')) \}.$$

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

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

$$D(\tau') \subset K := K \left( x, \frac{1}{2} \text{diam}(D(\tau)) + \frac{1}{2\eta} \text{diam}(D(\tau)) + \frac{\text{diam}(D(\tau))}{\eta} \right).$$

Das Volumen der Kugel läßt sich nach oben abschätzen durch

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

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

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

also

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

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

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

$$D(\tau') \subset K := K \left( x, \frac{1}{2} \text{diam}(D(\tau)) + \frac{1}{2\eta} \text{diam}(D(\tau)) + C_{bal} \text{diam}(D(\tau)) \right).$$

Das Volumen der Kugel läßt sich nach oben abschätzen durch

$$|K| \leq ((1 + 1 + 2C_{bal} \text{diam}(D(\tau)))^d.$$

73
und das von $D(\tau')$ nach unten durch

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

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

\[\blacksquare\]

\textbf{Bemerkung 5.9 (Voraussetzungen von Lemma 5.8)}


\textbf{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 $L_j := [0,2^{-j}]^2 \setminus [0,2^{-j-1}]^2$, $j = 0, 1, \ldots, n-2$, und setzen $L_n := [0,2^{-n+1}]^2$. Gegeben sei ein Knoten $\tau$ eines beliebigen $H$-Baumes $T$ der Dreiecke $(\tau_i)_{i=1}^n$ in $\Omega$, $I = \{1, \ldots, 8n\}$.

$\tau$ ist nicht entartet:

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 $\text{diam}(D(\tau)) \leq \sqrt{2} \cdot 2^{-j}$, und $|D(\tau)| \geq |D(\tau_i)| \geq \frac{1}{16} 2^{-2j}$.

Zusammen ergibt sich

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

$T$ ist lokal geometrisch balanciert: 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 $\text{diam}(\tau') \geq 2^{-j-1}$. Der Einfachheit halber sei vorerst $\eta := 0.5$. Ist $\text{dist}(D(\tau), D(\tau')) \leq \text{diam}(D(\tau'))$, so muß $D(\tau) \subset [0,2^{-j+1}]^2$ gelten, also $\text{diam}(D(\tau)) \leq \sqrt{2} \cdot 2^{-j+1} \leq 4\sqrt{2}\text{diam}(\tau')$. Im allgemeineren Fall $\eta < 0.5$ folgt aus $\text{dist}(D(\tau'), D(\tau')) \leq \frac{1}{20}\text{diam}(D(\tau'))$ nur $D(\tau) \subset [0,2^{-j+1-\log_2(2\eta)}]^2$ und somit

$$\text{diam}(D(\tau)) \leq \sqrt{2} \cdot 2^{-j+1-\log_2(2\eta)} \leq 4\sqrt{2}(2\eta)^{-1}\text{diam}(\tau').$$

Offenbar erfüllt jeder bzl. $Z_\eta$ minimal zulässige $H_\infty$-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 $H_\infty$-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 \( N_{H,St}(k, T, Z) \) einer Matrix aus \( M_{H,k}(T, Z) \) mit schwachbesetztem aus \( T_I, T_J \) gebildetem \( H \)-Baum \( T \) und konstantem Rang \( k \) gilt:

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

Beweis:
\[
N_{H,St}(k, T, Z) = \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} N_{Rk,St}(|\tau|, |\sigma|) + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} N_{F,St}(|\tau|, |\sigma|)
\]
\[
\leq \sum_{i \in L_T} \sum_{\tau \times \sigma \in \mathcal{L}^+(T,i)} k|\tau| + \sum_{i \in L_T} \sum_{\tau \times \sigma \in \mathcal{L}^+(T,i)} k|\sigma| + \sum_{i \in L_T} \sum_{\tau \times \sigma \in \mathcal{L}^-(T,i)} |\tau||\sigma|
\]
\[
\leq \sum_{i \in L_T} \sum_{\tau \times \sigma \in \mathcal{L}(T,i)} |\tau| \max(k, \frac{1}{2} b_{min}) + \sum_{i \in L_T} \sum_{\tau \times \sigma \in \mathcal{L}(T,i)} |\sigma| \max(k, \frac{1}{2} b_{min})
\]
\[
\leq |L_T| C_{sp} \max(k, \frac{1}{2} b_{min})(|I| + |J|).
\]

Bemerkung 5.12 (Was passiert bei unbalancierten \( H \)-Bäumen ?)
Unbalancierte \( H \)-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 \( H \)-Bäume nicht relevant. Solche \( H \)-Bäume entstehen zum Beispiel bei adaptiven Verfahren durch lokale Verfeinerung. Ein zu Anfang kardinalitätsbalancierter \( H \)-Baum wird nach mehreren lokalen Verfeinerungen unbalanciert. Um nicht nach jedem einzelnen Verfeinerungsschritt (z.B. eines Elementes der Triangulation) den \( H \)-Baum neu aufzustellen, sucht man nach Kriterien für die Unbalanciertheit bzw. Ausgeglichenheit des aktuellen \( H \)-Baumes. Prinzipiell müssen hier zwei Anwendungsfelder unterschieden werden:


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 \( N_{H,V}(k, T, Z) \) der Matrix-Vektor-Multiplikation in \( M_{H,k}(T, Z) \) läßt sich abschätzen durch

\[
N_{H,St}(k, T, Z) \leq N_{H,V}(k, T, Z) \leq 2N_{H,St}(k, T, Z).
\]
Beweis: Sei root\((T) = I × J\), \(M ∈ M_{H,k}(T, Z)\) und \(v ∈ ℝ^{|J|}\). Im Folgenden kürzen wir \(k := k(τ, σ)\) ab. Dann gilt
\[
Mv = ( \sum_{τ×σ∈L(T)} (M|_{τ×σ})|_{I×J} )v
= \sum_{τ×σ∈L^+(T)} (M|_{τ×σ})|_{I×J}v + \sum_{τ×σ∈L^-(T)} (M|_{τ×σ})|_{I×J}v,
\]
\[
N_{H,V}(k, T, Z) = \sum_{τ×σ∈L^+(T)} N_{R_{K,V}}(|τ|, |σ|) + \sum_{τ×σ∈L^-(T)} N_{F,V}(|τ|, |σ|)
≥ \sum_{τ×σ∈L^+(T)} k(|τ| + |σ|) + \sum_{τ×σ∈L^-(T)} |τ| · |σ|
= N_{H,St}(k, T, Z),
\]
\[
N_{H,V}(k, T, Z) = \sum_{τ×σ∈L^+(T)} (2k(|τ| + |σ|) - k - |τ|) + \sum_{τ×σ∈L^-(T)} (2|τ| · |σ| - |τ|)
≤ \sum_{τ×σ∈L^+(T)} 2k(|τ| + |σ|) + \sum_{τ×σ∈L^-(T)} 2|τ| · |σ|
= 2N_{H,St}(k, T, Z).
\]

Bemerkung 5.14  (Durchführung der Auswertung)
Die Komplexitätsbetrachtungen aus Lemma 5.13 setzen voraus, daß die Blöcke einer Matrix \(M ∈ M_{H,k}(T, Z)\) 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 |\(L(T)|\))
Die Mächtigkeit eines schwachbesetzten aus \(T_I, T_J\) gebildeten \(H_k\)-Baumes \(T\) läßt sich abschätzen durch
\[
|T| = \sum_{i=0}^{pr} |T^{(i)}| ≤ \sum_{i=0}^{pr} C_{sp} \min\{|I|, |J|\} ≤ C_{sp}(1 + pr) \min\{|I|, |J|\}.
\]
Nehmen wir für die Bäume \(T_I, T_J\) die Bedingung
\[∀t ∈ T_I ∪ T_J : |S(t)| ≠ 1\]
an, so ist \(|T_I| ≤ 2|I|\) und \(|T_J| ≤ 2|J|\). In diesem Fall gilt
\[
|L(T)| = \sum_{τ×σ∈L(T)} 1 ≤ \min\{\sum_{τ∈T_I} C_{sp}, \sum_{σ∈T_J} C_{sp}\} ≤ 2C_{sp} \min\{|I|, |J|\}.
\]

76
Lemma 5.16 (Aufwand der Bestapproximation)
Der Aufwand \( N_{H\rightarrow \mathcal{H}}(k,k',T,Z) \) der Bestapproximation einer Matrix aus \( \mathcal{M}_{H,k}(T,Z) \) durch eine Matrix aus \( \mathcal{M}_{H,k'}(T,Z) \), \( k' \leq k \), läßt sich abschätzen durch
\[
N_{H\rightarrow \mathcal{H}}(k,k',T,Z) \leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 5(|\tau| + |\sigma|)k(\tau \times \sigma)^2 + 23k(\tau \times \sigma)^3.
\]
Für konstanten Rang \( k \geq 1 \) gilt
\[
N_{H\rightarrow \mathcal{H}}(k,k',T,Z) \leq 5kN_{H,St}(k,T,Z) + 23k^3|\mathcal{L}^+(T)|.
\]
Beweis: Es gilt nach Algorithmus 2.12
\[
N_{H\rightarrow \mathcal{H}}(k,k',T,Z) = \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} N_{Rk,SVD}(|\tau|,|\sigma|)
\]
\[
\leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 5k(\tau \times \sigma)^2(|\tau| + |\sigma|) + 23k(\tau \times \sigma)^3
\]
\[
\leq 5kN_{H,St}(k,T,Z) + 23k^3|\mathcal{L}^+(T)|.
\]

Folgerung 5.17 (Aufwand der Approximation)
Der Aufwand \( N_{H\rightarrow \mathcal{H}}(s \cdot k,k,T,Z) \) der Approximation einer Matrix aus \( \mathcal{M}_{H,s,k}(T,Z) \) durch eine Matrix aus \( \mathcal{M}_{H,k}(T,Z) \) (s – 1 mal Kürzen von Rang 2k auf Rang k, siehe auch Abschnitt 2.5) läßt sich abschätzen durch
\[
N_{H\rightarrow \mathcal{H}}(s \cdot k,k,T,Z) \leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} (s - 1)20(|\tau| + |\sigma|)k(\tau \times \sigma)^2 + (s - 1)184k(\tau \times \sigma)^3.
\]
Für konstanten Rang \( k \geq 1 \) gilt
\[
N_{H\rightarrow \mathcal{H}}(k,k',T,Z) \leq (s - 1)20kN_{H,St}(k,T,Z) + (s - 1)184k^3|\mathcal{L}^+(T)|.
\]

Lemma 5.18 (Aufwand der hierarchischen Approximation)
Der Aufwand \( N_{H\rightarrow Rk}(k',T,Z) \) der hierarchischen Approximation einer Matrix aus \( \mathcal{M}_{H,k'}(T,Z) \) durch eine \( Rk \)-Matrix für einen schwachbesetzten aus \( T_1,T_2 \) 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
\[
N_{H\rightarrow Rk}(k',T,Z) \leq 5C_{sp}(1 + p_T)k^2(|I| + |J|) + 23|T|\kappa^3.
\]
Beweis: Für jeden Knoten \( \tau \times \sigma \in T \setminus \mathcal{L}(T) \) ist die Summe aus \( s Rk \)-Matrizen auf Rang \( k \) zu kürzen. Das Kürzen erfolgt als Bestapproximation in \( Rk(\tau \times \sigma) \) (gekürzte SVD, siehe Algorithmus 2.12) und hat eine Komplexität \( N_{Rsk \rightarrow Rk}(\tau \times \sigma) \) von
\[
N_{Rsk \rightarrow Rk}(\tau \times \sigma) \leq 5s^2k^2(|\tau| + |\sigma|) + 23s^3k^3.
\]
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:

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

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

\[
N_{\mathcal{H}^k} - \mathbf{R}^k(k', T, Z) = \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} N_{\mathbf{R}^k}^{k'} - \mathbf{R}^k(\tau \times \sigma) + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} N_{\mathcal{F}^k} - \mathbf{R}^k(\tau \times \sigma)
\]

\[
\leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 5k'^2(|\tau| + |\sigma|) + 23k'^3 + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 11(|\tau|^3 + |\sigma|^3)
\]

\[
+ \sum_{\tau \times \sigma \in \mathcal{L}^+(T) \setminus \mathcal{L}(T)} 5s^2k^2(|\tau| + |\sigma|) + 23s^3k^3
\]

\[
\leq \sum_{\tau \times \sigma \in \mathcal{T}} 5 \max\{k'^2, (sk)^2\}(|\tau| + |\sigma|)
\]

\[
+ 23 \max\{k^3, (sk)^3\} |T \setminus \mathcal{L}^{-}(T)| + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 11(|\tau|^3 + |\sigma|^3)
\]

\[
\leq \sum_{i=0}^{p_T} \sum_{\tau \times \sigma \in \mathcal{T}_i} 5 \max\{k'^2, (sk)^2\}(|\tau|)
\]

\[
+ \sum_{i=0}^{p_T} 5C_{sp} \max\{k'^2, (sk)^2\} |\sigma|
\]

\[
+ 23|T| \max\{k^3, (sk)^3, b_{\min}^3\}
\]

\[
\leq (p_T + 1)5C_{sp}k'^2(|I| + |J|) + 23|T|k^3.
\]

\[\blacksquare\]

5.4 Addition

**Lemma 5.19** (Aufwand der Matrix-Addition)

Der Aufwand \( N_{\mathcal{H}^k}(k, T, Z) \) der formatierten Addition in \( \mathcal{M}_{\mathcal{H},k}(T, Z) \) läßt sich abschätzen durch

\[
N_{\mathcal{H}^k}(k, T, Z) \leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 20(|\tau| + |\sigma|)k(\tau \times \sigma)^2 + 184k(\tau \times \sigma)^3 + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} |\tau| \cdot |\sigma|.
\]
Für konstanten Rang \( k \geq 1 \) gilt
\[
N_{H,\oplus}(k, T, Z) \leq 20kN_{H,St}(k, T, Z) + 184k^3|L^+(T)|.
\]

**Beweis:** Es gilt nach Abschnitt 2.5
\[
N_{H,\oplus}(k, T, Z) = \sum_{\tau \times \sigma \in L^+(T)} N_{R,\oplus}(|\tau|, |\sigma|) + \sum_{\tau \times \sigma \in L^-(T)} N_{F,\oplus}(|\tau|, |\sigma|)
\]
\[
\leq \sum_{\tau \times \sigma \in L^+(T)} 20k(\tau \times \sigma)^2(|\tau| + |\sigma|) + 184k(\tau \times \sigma)^3 + \sum_{\tau \times \sigma \in L^-(T)} |\tau| \cdot |\sigma|
\]
\[
\leq \sum_{\tau \times \sigma \in L^+(T)} 20k^2(|\tau| + |\sigma|) + 184k^3 + \sum_{\tau \times \sigma \in L^-(T)} 20k|\tau| \cdot |\sigma|
\]
\[
= 20kN_{H,St}(k, T, Z) + 184k^3|L^+(T)|.
\]

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

Ist auf einer Stufe \( l \) der Rang \( k = k(l) \) und auf allen anderen Stufen \( k \) konstant, so erhält man die Abschätzung
\[
N_{H,\oplus}(k, T, Z) = O((k^2|L_T| + k^3)(|I| + |J|)) + O(k(l)^3(|I| + |J|)),
\]
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
\[
N_{H,\oplus}(k, T, Z) = O(b^2p_T^3(|I| + |J|))
\]
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 \( H \)-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 $T_1$ gebildeter bzgl. $Z$ zulässiger $H_\infty$-Baum $T$ heißt fast idempotent zur Konstante $C_{id}$, falls für jedes Blatt $\tau \times \tau' \in L(T)$ gilt:

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

**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:

![Diagram of two graphs](image)

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

**Lemma 5.23** (Fast-Idempotenz bei Standard-Zulässigkeitsbedingung)
Sei $T$ ein aus $T_1$ gebildeter bzgl. der Standard-Zulässigkeitsbedingung $Z_\eta$ zulässiger $H_\infty$-Baum, $s := \max_{t \in T} |S(t)|$. Der $H$-Baum $T_1$ sei geometrisch verfeinert, d.h. es gebe Konstanten $C_1, C_2$ mit $C_1 C_2 < 1$, $C_2 < 1$ und

$$\forall i \in \{0, \ldots, p_{T_1}\} \forall \tau \in T_1^{(i)} \forall j \in \{0, \ldots, i - 1\}: \text{diam}(D(\tau)) \leq C_1 C_2^j \text{diam}(D(\tau^{(i-j)}))$$

und er sei lokal geometrisch balanciert, d.h. für alle $i \in \{0, \ldots, p_T\}$, $\tau, \tau' \in T_1^{(i)}$ gelte

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

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 $\tau \times \tau' \in L(T, i)$.

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

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

Die Mächtigkeit der Menge $M := \{ \sigma \times \sigma' \in T \cdot T \mid \sigma \subset \tau \land \sigma' \subset \tau' \}$ gilt es abschätzen. Sei $\sigma \times \sigma' \in M$ und $\sigma \times \sigma' \neq \tau \times \tau'$. Nach Definition von $T \cdot T$ gibt es ein $\tilde{\sigma} \in T_1$ 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 $T_1$ lokal geometrisch balanciert ist, folgt daraus

$$\text{diam}(D(\tilde{\sigma}^{(i)})) \leq C_{bal} \text{diam}(D(\tau)), \quad \text{diam}(D(\tilde{\sigma}^{(i)})) \leq C_{bal} \text{diam}(D(\tau')).$$
Sei $j \in \mathbb{N}$ mit $\sigma \times \sigma' \in (T \cdot T)^{((i+j))}$. Da $T_1$ geometrisch verfeinert wurde, ist
\[
diam(D(\tilde{\sigma})) \leq C_1 C_2^j diam(D(\tilde{\sigma}^{(i)})) \\
\leq C_1 C_2^j C_{bal} \min\{diam(D(\tau)), diam(D(\tau'))\} \\
\leq C_1 C_2^j C_{bal} 2\eta \text{dist}(D(\tau), D(\tau')).
\] (28)

Wir definieren die Umgebungen
\[
U_\tau := \{ x \in \mathbb{R}^d \mid \text{dist}(x, D(\tau)) \leq \frac{1}{3} \text{dist}(D(\tau), D(\tau')) \} \quad \text{und}
\]
\[
U_{\tau'} := \{ x \in \mathbb{R}^d \mid \text{dist}(x, D(\tau')) \leq \frac{1}{3} \text{dist}(D(\tau), D(\tau')) \}.
\]

Annahme: $C_1 C_2^j C_{bal} \leq \frac{1}{3} \min\{1, \frac{1}{2\eta}\}$. Dann ist $diam(D(\tilde{\sigma})) \leq \frac{1}{3} \text{dist}(D(\tilde{\sigma}), D(\tilde{\tau}'))$,
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
\[
\min\{diam(D(\sigma)), diam(D(\tilde{\sigma}))\} \leq diam(D(\tilde{\sigma})) \\
\leq \frac{1}{3} \min\{diam(D(\tau)), diam(D(\tau'))\} \\
\leq 2\eta \text{dist}(D(\tilde{\sigma}), D(\tau)) \\
\leq 2\eta \text{dist}(D(\tilde{\sigma}), D(\sigma)).
\] (28)

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

Da die Vorfahren von $\sigma \times \tilde{\sigma}$ nicht zulässig sind (keine Blätter nach Def. von $T \cdot T$), gilt für $\eta \leq 0.5$:
\[
C_1 C_2^j C_{bal} > \frac{1}{3},
\]
\[
C_2^j > \frac{C_2}{3C_1 C_{bal}},
\]
\[
j \log_s(C_2) > \log_s(C_2) - \log_s(3C_1 C_{bal})
\]
\[
j < 1 + \log_s(3C_1 C_{bal}) \log_s(C_2^{-1})^{-1}.
\]

Die Zahl der Nachfahren $\sigma \times \sigma'$ von $\tau \times \tau'$ auf $j$ Stufen ist beschränkt durch $s^j$, so daß $C_{id} \geq s(3C_1 C_{bal})^{\log_s(C_2^{-1})^{-1}}$ 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 \( (\tau^{(j)} \times \tau'^{(j)})_{j=0}^{\infty} \) 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
\[
|\{\tau \times \tau' \in (T \cdot T)^{(i)}\}| \leq \sum_{j=i}^{-C_{id}(T)} C_{id}(T) \left|\{\tau^{(j)} \times \tau'^{(j)} \in T^{(j)}\}\right| \leq C_{id}(T)^2 C_{sp}(T). \]

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 bezgl. 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 \( \mathcal{H} \times \mathcal{H} \)-Baum. \( T \) sei schwachbesetzt und fast idempotent. Die Berechnung des Produktes zweier Matrizen \( M, M' \in M_{\mathcal{H}, k}(T, Z) \) zu konstantem Rang \( k \) in der Darstellung \( M \cdot M' \in M_{\mathcal{H}, k, \tilde{k}}(T, Z) \) hat einen Aufwand von
\[
N_{\mathcal{H}, \odot}^{\text{exakt}}(k, T, Z) \leq 4(p_T + 1)C_{sp} \max(k, b_{\min}) N_{\mathcal{H}, \odot}(k, T, Z) + |\mathcal{L}^{-}(T)| 2(p_T + 1)(C_{id} + 1) C_{sp} k b_{\min}^2 \quad = \quad O(p_T k) N_{\mathcal{H}, \odot}(k, T, Z)
\]
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 $\tau \times \tau' \in \mathcal{L}(T \cdot T, i)$:

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

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

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 $M|_{\tau \times j}$ oder $(M'|_{\tau \times j})T$ durchzuführen. Summiert über die Partition $U_j \cup j=0 U_j$ von $J$ sind höchstens $k$ Matrix-Vektor-Multiplikationen mit $M|_{\tau \times J}$ und $(M'|_{\tau \times J})T$ nötig, deren Aufwand nach Lemma 5.13 durch $2k$-mal den Speicheraufwand $N_{M|_{\tau \times J},St}^+$ von $M|_{\tau \times J}$ bzw. $N_{M'|_{\tau \times J},St}^+$ von $(M'|_{\tau \times J})T$ beschränkt ist. Es folgt

$$N^+ \leq \sum_{\tau \times \tau' \in \mathcal{L}^+(T \cdot T)} \sum_{\tau \times \tau' \in \mathcal{L}^+(T \cdot T)} 2kN_{M|_{\tau \times J},St}^+ + 2kN_{M'|_{\tau \times J},St}^+$$

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

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

$$N^- \leq \sum_{\tau \times \tau' \in \mathcal{L}^-(T \cdot T)} 2\max(k, b_{\min})N_{M|_{\tau \times J},St}^- + \sum_{\tau \times \tau' \in \mathcal{L}^-(T \cdot T)} 2\max(k, b_{\min})N_{M'|_{\tau \times J},St}^- + \sum_{\tau \times \tau' \in \mathcal{L}^-(T \cdot T)} (p_T+1)C_{sp}b_{\min}^2$$

$$\leq \sum_{i=0}^{p_T} \sum_{\tau \times \tau' \in \mathcal{L}^-(T \cdot T,i)} 2\max(k, b_{\min})N_{M|_{\tau \times J},St}^- + \sum_{i=0}^{p_T} \sum_{\tau \times \tau' \in \mathcal{L}^-(T \cdot T,i)} 2\max(k, b_{\min})N_{M'|_{\tau \times J},St}^- + (p_T+1)C_{sp}b_{\min}^2.$$
Der Aufwand zur Berechnung von $M \cdot M'$ in $M_{H,k}(T \cdot T, Z \cdot Z)$ ergibt sich aus der Summe von $N^+$ und $N^-:

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

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

Der Aufwand zum Ausmultiplizieren ($\mathcal{R}k$ $\rightarrow$ vollbesetzt) wird durch

\[
N^{\text{ausmul}} \leq \sum_{\tau \times \tau' \in \mathcal{L}^-(T)} 2(pt + 1)C_{sp}kb_{min}^2 \\
\leq |\mathcal{L}^-(T)|2(pt + 1)C_{sp}kb_{min}^2
\]

abgeschätzt und der Gesamtaufwand erfüllt schließlich

\[
N^{\text{exakt}}_{H,k}(k, T, Z) = N^+ + N^- + N^{\text{ausmul}} \\
\leq 4(pt + 1)C_{sp} \max(k, b_{min}) N_{H,St}(k, T, Z) + |\mathcal{L}^-(T \cdot T)|(pt + 1)C_{sp}b_{min}^2 \\
+ |\mathcal{L}^-(T)|2(pt + 1)C_{sp}kb_{min}^2 \\
\leq 4(pt + 1)C_{sp} \max(k, b_{min}) N_{H,St}(k, T, Z) + |\mathcal{L}^-(T)|2(pt + 1)(C_{id} + 1)C_{sp}kb_{min}^2.
\]

Bestimmung des Ranges $\tilde{k}$: Wir wollen nun den Rang $\tilde{k}$, der zur Darstellung von $M \cdot M'$ in $M_{H,k}(T, Z)$ 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.)
\[ |\tau \times \tau'| \leq b_{\text{min}}^2 \] und somit \( \text{rang}((M \cdot M')|_{\tau \times \tau'}) \leq b_{\text{min}} \). Damit läßt sich der zur Darstellung nötige Rang abschätzen durch
\[
\hat{k} \leq C_{id} \max((p_T + 1)C_{sp}k, b_{\text{min}}).
\]

\[ \text{Folgerung 5.27} \ (\text{Bestapproximation und Approximation}) \]

Sei \( T_I \) ein \( H \)-Baum der Indexmenge \( I \) und \( T \) ein aus \( T_I \) gebildeter bzgl. \( Z \) zulässiger \( H \)-Baum. \( T \) sei schwachbesetzt und fast idempotent. Dann läßt sich der Aufwand \( N_{\text{best}}, N_{\text{apx}} \) der formatierten Multiplikation (Bestapproximation/Approximation) in \( M_{H,k}(T, Z) \) für konstanten Rang \( k \) abschätzen durch
\[
N_{\text{best}}^\circ(k, T, Z) = O(p_T^2 k)N_{H,Sl}(k, T, Z) + O(p_T^2 k^3)|L^+(T)|,
N_{\text{apx}}^\circ(k, T, Z) = O(p_T k)N_{H,Sl}(k, T, Z) + O(p_T k^3)|L^+(T)|
\]

\[ \text{Beweis:} \ \text{Satz 5.26, Lemma 5.16 und Folgerung 5.17.} \]

\[ \text{Bemerkung 5.28} \ (\text{Multiplikation bei verschiedenen Bäumen}) \]

Seien \( T_I, T_J, T_I' \) \( H \)-Bäume der Indexmengen \( I, J, I' \), \( T \) ein aus \( T_I, T_J \) gebildeter bzgl. \( Z \) zulässiger, \( T' \) ein aus \( T_I, T_J \) gebildeter bzgl. \( Z' \) zulässiger und \( \tilde{T} \) ein aus \( T_I', T_J \) gebildeter bzgl. \( Z \) zulässiger \( H \)-Baum. \( T \) und \( T' \) erfüllen die Schwachbesetztheitsbedingung
\[
\forall i \in \{0, \ldots, p_T\} \ \forall \tau \in T_I^{(i)}: \ |\{\tau \times \tau' \in T_I^{(i)}\}| \leq C_{sp}
\]
\[
\forall i \in \{0, \ldots, p_{T'}\} \ \forall \tau' \in T_J^{(i)}: \ |\{\tau \times \tau' \in T_J^{(i)}\}| \leq C_{sp}
\]
und die Kompatibilitätsbedingung (analog zur Fast-Idempotenz)
\[
\forall \tau \times \tau' \in L(\tilde{T}): \ |\{\sigma \times \sigma' \in T \cdot T' \ | \ \sigma \subset \tau \land \sigma' \subset \tau'\}| \leq C_{id}.
\]

Dann ist der Aufwand für die exakte Multiplikation
\[
\cdots : M_{H,k}(T, Z) \times M_{H,k}(T', Z') \to M_{H,k}(\tilde{T}, \tilde{Z})
\]
beschränkt durch
\[
N_{H,k}^{\text{exakt}}(k, T, T', \tilde{T}, Z, Z', \tilde{Z}) \leq 2(p_T + 1)C_{sp} \max(k, b_{\text{min}})(N_{H,Sl}(k, T, Z) + N_{H,Sl}(k, T', Z')) + |L^-(\tilde{T})|2(p_T + 1)(C_{id} + 1)C_{sp}k b_{\text{min}}^2
= O(p_T k)(N_{H,Sl}(k, T, Z) + N_{H,Sl}(k, T', Z'))
\]
und der zur Darstellung nötige Rang ist beschränkt durch
\[
\hat{k} \leq C_{id} \max((p_T + 1)C_{sp}k, b_{\text{min}}).
\]
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 $\mathcal{H}_k$-Baum. Die Berechnung der $\mathcal{H}$-Inversen (siehe Abschnitt 4.5.1 bzw. Algorithmus 4.18) einer Matrix $M \in M_{\mathcal{H},k}(T, Z)$ hat einen Aufwand von

$$N_{\mathcal{H} \odot}(k, T, Z) \leq N_{\mathcal{H}, \ominus}^{\text{apx}}(k, T, Z).$$

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 Binärbäumes $T_I$ behandelt. Ferner sei $|\tau \times \tau| = 1$ für alle $\tau \times \tau \in \mathbb{L}(T)$. Die Behauptung wird per Induktion über die Baumtiefe $p_T$ bewiesen.

Induktionsanfang: $p_T = 0$. Nach Voraussetzung ist eine $1 \times 1$ Matrix zu invertieren, und wir nehmen an, daß der Aufwand dieselbe 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

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

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

$$L_{11} := R_{11} \odot$$
$$H_{12} := L_{11} \odot R_{12}$$
$$R_{12} := H_{12}$$
$$H_{22} := -R_{21} \odot R_{12}$$
$$R_{22} := R_{22} \odot H_{22}$$
$$L_{21} := R_{21} \odot L_{11}$$

$$L_{22} := R_{22} \odot$$
$$L_{21} := -L_{22} \odot L_{21}$$
$$H_{11} := -R_{12} \odot L_{21}$$
$$L_{11} := L_{11} \odot H_{11}$$
$$L_{12} := -R_{12} \odot L_{22}$$

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 $N_{H,\|\cdot\|_F}(k, T, Z)$ zur Berechnung der Frobeniusnorm einer Matrix aus $M_{H,k}(T, Z)$ läßt sich abschätzen durch

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

Für konstanten Rang $k \geq 1$ gilt

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

Beweis: Es gilt nach Abschnitt 2.6

$$N_{H,\|\cdot\|_F}(k, T, Z) = \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} N_{R_{\|\cdot\|_F}}(|\tau|, |\sigma|) + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} N_{F_{\|\cdot\|_F}}(|\tau|, |\sigma|)$$

$$\leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 4k(\tau \times \sigma)^2(|\tau| + |\sigma|) + 23k(\tau \times \sigma)^3$$

$$+ \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 2|\tau| \cdot |\sigma|$$

$$\leq \sum_{\tau \times \sigma \in \mathcal{L}^+(T)} 4k^2(|\tau| + |\sigma|) + 23k(\tau \times \sigma)^3 + \sum_{\tau \times \sigma \in \mathcal{L}^-(T)} 4k|\tau| \cdot |\sigma|$$

$$= 4kN_{H,St}(k, T, Z) + 23k^3|\mathcal{L}^+(T)|.$$  

Beweisende Gleichung

Folgerung 5.31 (Aufwand zur Approximation der Spektralnorm)

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

$$N_{H,\|\cdot\|_2,apx}(k, T, Z, \varepsilon) = O(\varepsilon^{-1}(\log(\min\{|I|, |J|\}) - \log(\varepsilon))N_{H,St}(k, T_{I \times J}, Z)).$$
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 V erfahren der Greenschen Funktion wird durch die Singularitätenfunktion (Fundamentallösung, Kernfunktion bei der Integralgleichungs methode) 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 $\oplus, \odot, \ominus$ 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

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

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

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

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

Mit den Schwachbesetztheitskonstanten aus Definition 6.1 gilt die schärfere Abschätzung

$$
\|M\|_2 \leq \sum_{l \in L_T} C^{(l)}_{sp} \max_{t \in \mathcal{L}(T,l)} \|M|_t\|_2.
$$
Beweis: Es gilt für eine Matrix $A \in \mathbb{R}^{n,m}$ mit höchstens $C$ Einträgen in jeder Spalte und Zeile und $\bar{a} := \max_{i,j} |a_{ij}|$

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

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

$$\|A\|_2 \leq C \max_{i,j} \|A_{ij}\|_2,$$  \hspace{1cm} (29)

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

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

Dann gilt

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

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 $Z : T_I \times T_I \rightarrow \{\text{„zulässig“}, \text{„nicht zulässig“}\}$ 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 $\mathcal{H}$-Baum $T_{I\times I}$ sei der aus $T_I, T_I$ gebildete minimal zulässige $\mathcal{H}$-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 $M \in M_{\mathcal{H},1}(T, Z)$ 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, $I = \bigcup_{i=1, \ldots, n} I_i$, $J = \bigcup_{j=1, \ldots, m} J_j$ und $A \in \mathbb{R}^{I \times J}$ mit der Eigenschaft, daß

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

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

Dann läßt sich die Spektralnorm von $A$ abschätzen durch

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

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

$$\|x\|_{\infty, b, m} := \max_{j=1, \ldots, m} \|x|_{J_j}\|_2,$$

$$\|y\|_{\infty, b, n} := \max_{i=1, \ldots, n} \|y|_{I_i}\|_2$$

und die entsprechende Matrixnorm

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

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

$$\|A\|_{\infty, b} = \max_{\|x\|_{\infty, b, m} = 1} \|Ax\|_{\infty, b, n} = \max_{\|x\|_{\infty, b, m} = 1} \max_{i=1, \ldots, n} \|(Ax)|_{I_i}\|_2$$

$$\leq \max_{\|x\|_{\infty, b, m} = 1} \max_{i=1, \ldots, n} \|\sum_{j=1}^m A_{I_i \times J_j}x|_{J_j}\|_2$$

$$\leq \max_{\|x\|_{\infty, b, m} = 1} \max_{i=1, \ldots, n} \max_{j=1, \ldots, m} \|A_{I_i \times J_j}\|_2 \|x|_{J_j}\|_2 \leq C \left( \max_{i' = 1, \ldots, n} \|A_{I_{i'} \times J_j}\|_2 \right) \left( \max_{j=1, \ldots, m} \max_{i=1, \ldots, n} \|x|_{J_j}\|_2 \right)$$

$$= C \max_{i' = 1, \ldots, n} \|A_{I_{i'} \times J_j}\|_2 \|z\|_{\infty, b}.$$

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

$$\|A\|_2^2 \|z\|_{\infty, b} = \|A^Tz\|_{\infty, b} \leq \|A^T\|_{\infty, b} \|A\|_{\infty, b} \|z\|_{\infty, b} \leq C^2 \left( \max_{i=1, \ldots, n} \|A_{I_i \times J_j}\|_2^2 \right) \|z\|_{\infty, b}. \]
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

$$\|M - \tilde{M}\|_2 \leq \varepsilon$$  (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 $N_{\mathcal{H}, \text{ST}}(\tilde{k}, T, Z)$ möglichst minimiert. Aus Satz 6.2 wissen wir bereits, daß $\|M - \tilde{M}\|_{I_i \times J_j} \leq \frac{\varepsilon}{C_{sp}}$ eine hinreichende Bedingung ist, so daß der Rang in einem Block $(I_i \times J_j)$ so gewählt werden kann (mit Hilfe der gekürzten Singularwertzerlegung 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

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

Dann gilt für jede Bestapproximierende $M_{\tilde{h}} \in \mathcal{M}_{\mathcal{H},\tilde{k}}(T, Z)$ 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

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

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

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

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

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

91
Folgerung 6.8 (Rangwahl bei der Multiplikation)
Sei $T$ ein aus $T_1, T_2$ gebildeter $\mathcal{H}$-Baum, $T'$ ein aus $T_1', T_2'$ 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

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

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 $\frac{\|A^{-1} - \Theta^{-1}\|_2}{\|A^{-1}\|_2} < \varepsilon$ 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 $\Theta := \|A\|^{-2}$. Die positiv semidefinite und symmetrische Iterationsmatrix des Richardson-Verfahrens (siehe [8]) ist $M_{\Theta}^{\text{Rich}} := I - \Theta A^T A$. Die Matrix $A^T A$ bzw. $M_{\Theta}^{\text{Rich}}$ 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 $\rho$ von $M_{\Theta}^{\text{Rich}}$. 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

$$\rho = \max \{ |1 - \Theta \lambda_{\min}(A^T A)|, |1 - \Theta \lambda_{\max}(A^T A)| \} = 1 - \Theta \lambda_{\min}(A^T A) = 1 - \Theta \lambda_{\max}(A^{-1} A^{-T})^{-1} = 1 - \Theta \|A^{-1}\|_2^{-2}.$$
Bemerkung 6.10 (Güte der Schätzung der Spektralnorm der Inversen)

Die Bestimmung von $\tilde{\Theta}$ genügt für vorgegebenes $\varepsilon$ 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

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

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

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

so daß bei schlecht konditionierter Matrix $A$ eine erhebliche Fehlerverstärkung auftritt. Den gleichen Effekt erzielt man, wenn man etwa aus der Konvergenzrate des Gradientenverfahrens Rückschlüsse auf die Eigenwerte ziehen möchte.

Schlußfolgerung: Stellt sich bei der Berechnung von $\rho$ das Verhalten $\tilde{\rho} \rightarrow 1$ ein, so ist die Schätzung für die Spektralnorm von $A^{-1}$ fehlergeschlagen und $A$ schlecht konditioniert. In diesem Fall kann man, z.B. mittels der $H$-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 $\delta_A$ 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 $H$-Matrix $A \in \mathcal{M}_{H,k}(T, Z)$, ein aus $T_1, T_J$ gebildeter $H$-Baum $T$ und eine Fehlertoleranz $\varepsilon$.

Gesucht: Eine approximative Inverse $A^\odot$ von $A$ mit $\|I - A^\odot A\|_2 \leq \varepsilon$.

1. Stufe: Bestimme die stufenweisen Schwachbesetztheitskonstanten $C_{sp}^{(l)}$ und $C_{sp} := \sum_{l \in L} C_{sp}^{(l)}$ sowie die Spektralnorm $\|A\|_2$ von $A$ mit Hilfe von Satz 4.31. Setze $\varepsilon_{local} := \varepsilon/(C_{sp}\|A\|_2)$.

Führe die formatierte Inversion nach Algorithmus 4.18 in $\mathcal{M}_{H,k}(T, Z)$ 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 $\varepsilon_{local}$ nicht überschritten wird.

93
Ermittle anschließend den Faktor $\delta_A := \| I - A^\circ A \|_2 / \varepsilon$ der Fehlerverstärkung.

2. Stufe: Führe wie in (1.) die Rang-adaptive formatierte Inversion für $\varepsilon_{\text{local}} := \varepsilon / (\tilde{C}_{\text{sp}} \| A \|_2 \delta_A)$ durch (dann hebt sich der zusätzliche Faktor $1/\delta_A$ mit der Fehlerverstärkung um den Faktor $\delta_A$ gerade weg).

Bemerkung 6.12 (Zur adaptiven Inversion)
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 $\mathcal{H}_2$-Baum, $Z$ eine Zulässigkeitsbedingung und $k$ eine Rangverteilung. Zu approximieren ist eine Matrix $M \in \mathbb{R}^{n,m}$ durch eine Matrix $M_{\mathcal{H}} \in M_{\mathcal{H},k}(T,Z)$ bis auf einen Fehler von $\varepsilon$:

$$\|M - M_{\mathcal{H}}\| \leq \varepsilon.$$ (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

$$\|M - M_{\mathcal{H}}\|_F^2 = \sum_{t \times s \in \mathcal{L}^+(T)} \|M|_{t \times s} - (M|_{t \times s})_{\mathcal{H}}\|_F^2$$

$$= \sum_{t \times s \in \mathcal{L}^+(T)} \sum_{i=k(t\times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t \times s})^2$$

und in der Spektralnorm gilt

$$\|M - M_{\mathcal{H}}\|_2 \geq \max_{t \times s \in \mathcal{L}^+(T)} \|M|_{t \times s} - (M|_{t \times s})_{\mathcal{H}}\|_2$$

$$= \max_{t \times s \in \mathcal{L}^+(T)} \sigma_{k(t \times s)+1}(M|_{t \times s}).$$

Eine notwendige Bedingung an die Matrix $M$ für (31) in der Spektral- oder Frobeniusnorm ist demnach

$$\forall t \times s \in \mathcal{L}^+(T) : \sigma_{k(t \times s)+1}(M|_{t \times s}) \leq \varepsilon.$$ (32)

**Bemerkung 7.2 (Hinreichende Bedingung)**

Für eine Bestapproximation $M_{\mathcal{H}}$ von $M$ in der Frobeniusnorm gilt

$$\|M - M_{\mathcal{H}}\|_F^2 = \sum_{t \times s \in \mathcal{L}^+(T)} \sum_{i=k(t \times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t \times s})^2$$

$$\leq |\mathcal{L}^+(T)| \max_{t \times s \in \mathcal{L}^+(T)} \sum_{i=k(t \times s)+1}^{\min(|t|,|s|)} \sigma_i(M|_{t \times s})^2.$$
Eine hinreichende Bedingung für (31) an $M$ wäre
\[ \forall t \times s \in \mathcal{L}^+(T) : \min_{i=k(t \times s)+1} \sigma_1(M_{t \times s}) \leq \frac{\varepsilon}{\sqrt{|\mathcal{L}^+(T)|}}. \] (33)

In der Spektralnorm gilt nach Satz 6.2
\[ \|M - M_K\|_2 \leq C_{sp}|LT| \max_{t \times s \in \mathcal{L}^+(T)} \|M_{t \times s} - (M_{t \times s})_N\|_2. \]

Hier ist die Bedingung
\[ \forall t \times s \in \mathcal{L}^+(T) : \sigma_{k(t \times s)+1}(M_{t \times s}) \leq \frac{\varepsilon}{C_{sp}|LT|} \] (34)
hinreichend für (31).

**Bemerkung 7.3** (Lokalität der Approximationseigenschaft)
Für die Spektralnorm ist die globale Approximationseigenschaft (31) bis auf den Faktor $C_{sp}|LT|$ 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
\[ K : H^{\alpha_K + \beta_K}(\Omega_y) \rightarrow H^{\beta_K}(\Omega_x), \quad f \mapsto u, \quad u(x) := \int_{\Omega_y} g(x, y) f(y) \, dy \]
der der Ordnung $\alpha_K$ und eine Diskretisierung $K$ von $\mathcal{K}$, z.B. eine Galerkin-Diskretisierung mit integrierbaren Basisfunktionen $(b_i)_{i \in r}, (b_j)_{j \in s}$:
\[ K_{ij} := \int_{\Omega_y} \int_{\Omega_x} b_i(x) g(x, y) b_j(y) \, dx \, dy, \quad (i, j) \in r \times s, \]
eine Kollokationsmethode mit integrierbaren Basisfunktionen $(b_j)_{j \in s}$ und Kollokationspunkten $(x_i)_{i \in r}$:
\[ K_{ij} := \int_{\Omega_x} g(x_i, y) b_j(y) \, dx, \quad (i, j) \in r \times s, \]
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 $\mathbb{R}^k$-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) approximierbar, d.h.

\[
\tilde{g}(x,y) = \sum_{i=1}^{k} \tilde{g}_{1,i}(x)\tilde{g}_{2,i}(y)
\]

bis auf einen relativen Fehler von \( \varepsilon \) approximierbar, d.h.

\[
|\tilde{g}(x,y) - g(x,y)| \leq \varepsilon |g(x,y)|, \quad (35)
\]

so gilt für jede \( Rk \)-Bestapproximation \( R \) von \( K \) im Fall der Nyström-Methode

\[
\|K - R\|_F \leq \varepsilon \|K\|_F,
\]

wobei \( K \) die entsprechende Diskretisierung von \( \tilde{K}[f](x) := \int_{\Omega_y} g(x,y)|f(y)| \, dy \) ist.

**Beweis:** Definiere den Operator \( \tilde{K}[f](x) := \int_{\Omega_y} \tilde{g}(x,y)f(y) \, dy \) 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

\[
|K_{ij} - \tilde{K}_{ij}| = \int_{\Omega_y} b_j(y)(g(x_i, y) - \tilde{g}(x, y)) \, dy \leq \varepsilon \int_{\Omega_y} |b_j(y)||g(x_i, y)| \, dy = \varepsilon |K_{ij}|
\]

sowie für das Galerkin-Verfahren

\[
|K_{ij} - \tilde{K}_{ij}| = \left| \int_{\Omega_x} \int_{\Omega_y} b_i(x)(g(x,y) - \tilde{g}(x,y))b_j(y) \, dx \, dy \right| \leq \varepsilon \int_{\Omega_x} \int_{\Omega_y} |b_i(x)||g(x,y)||b_j(y)| \, dx \, dy = \varepsilon |K_{ij}|
\]

Besitzen nur die Basisfunktionen \( b_i \) keine Vorzeichenwechsel, so ist im Fall der Kollokationsmethode

\[
|K_{ij} - \tilde{K}_{ij}| = \int_{\Omega_y} b_j(y)(g(x_i, y) - \tilde{g}(x_i, y)) \, dy \leq \varepsilon \int_{\Omega_y} |b_j(y)||g(x_i, y)| \, dy = \varepsilon \tilde{K}_{ij}
\]
und für das Galerkin-Verfahren

\[
|K_{ij} - \tilde{K}_{ij}| = \left| \int_{\Omega_x} \int_{\Omega_y} b_i(x)(g(x, y) - \tilde{g}(x, y))b_j(y) \, dx \, dy \right|
\leq \varepsilon \int_{\Omega_x} \int_{\Omega_y} |b_i(x)||g(x, y)||b_j(y)| \, dx \, dy = \varepsilon \tilde{K}_{ij}.
\]

\[\blacksquare\]

**Beispiel 7.5 (Keine lokale Approximationseigenschaft)**


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

\[\tilde{K}_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x)g(x, y)b_1(y) \, dx \, dy = 0.\]

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

\[K_{11} - \tilde{K}_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x)(g(x, y) - \tilde{g}(x, y))b_1(y) \, dx \, dy = \int_{[0,1]} \int_{[0, \frac{1}{2}]} b_1(x)\varepsilon b_1(y) \, dx \, dy = \frac{1}{2} \varepsilon.\]

In diesem Fall ist die Abschätzung \(\|K - R\|_F \leq \varepsilon\|K\|_F\) (\(= 0\)) nicht erfüllt.


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

\[\tilde{K}_{11} = K_{11} = \int_{[0,1]} \int_{[0,1]} b_1(x)g(x, y)b_1(y) \, dx \, dy = 0.\]
Setzt man
\[ \tilde{g} : [0, 1] \times [0, 1] \to \mathbb{R}, (x, y) \mapsto \begin{cases} 1 - \varepsilon & x, y < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \lor y \geq \frac{1}{2} \end{cases}, \]
so gilt \(|g(x, y) - \tilde{g}(x, y)| \leq \varepsilon\) und
\[
\begin{align*}
K_{11} - \tilde{K}_{11} &= \int_{[0, 1]} \int_{[0, 1]} b_1(x)(g(x, y) - \tilde{g}(x, y))b_1(y) \, dx \, dy = \int_{[0, \frac{1}{2}]} \int_{[0, \frac{1}{2}]} b_1(x)\varepsilon b_1(y) \, dx \, dy = \frac{1}{4}\varepsilon.
\end{align*}
\]
In diesem Fall ist weder \(\|K - R\|_F \lesssim \varepsilon \|K\|_F\) noch \(\|K - R\|_F \lesssim \varepsilon \|\tilde{K}\|_F\) 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
\[
\|K - R\|_F \leq \varepsilon \left( \sum_{i \in I} \|l_i\|^2 \right)^{\frac{1}{2}} \left( \sum_{j \in J} \|\tilde{l}_j\|^2 \right)^{\frac{1}{2}} \sup_{(x, y) \in \Omega_x \times \Omega_y} |g(x, y)|
\]
hergeleitet (dort wird die Diskretisierung mittels der Funktionale \(l_i, \tilde{l}_j\) verallgemeinert), in [14, Lemma 3.2] wird
\[
\|u - \tilde{u}\|_W \lesssim \inf_{v \in V} \|u - v\|_W + \frac{c(0, m)}{m!} \eta^m \|\tilde{K}\|_{W \rightarrow W} \|u\|_W
\]
bewiesen, wobei hier der Operator \(\tilde{K}\) durch
\[
\tilde{K}[f](x) := \int_{\Omega_y} \max_{z \in \Omega_y} |g(x, z)| f(y) \, dy
\]
definiert ist und \(\|\tilde{K}\|_{W \rightarrow W}\) für spezielle Kernfunktionen weiter untersucht wird.

Lemma 7.7 (Approximation durch entartete Kerne)
Die Kernfunktion \(g\) erfüllt die asymptotische Glattheitsbedingung
\[
\forall (x, y) \in \Omega_x \times \Omega_y : \left| \partial_x^\alpha \partial_y^\beta g(x, y) \right| \leq c_{\alpha, \beta} \|x - y\|^{-|\alpha + \beta|} |g(x, y)|. \tag{36}
\]
Das Gebiet \(\Omega_x \times \Omega_y\) sei \(\eta\)-zulässig, \(x^*\) das \(\varepsilon\)-Zentrum von \(\Omega_x\), \(y^*\) das \(\varepsilon\)-Zentrum von \(\Omega_y\) und \((x, y) \in \Omega_x \times \Omega_y\). Dann gilt für die Taylorentwicklung bis zur Ordnung \(m\)
\[
\begin{align*}
in x^* , \text{ falls diam}(\Omega_x) \leq \text{diam}(\Omega_y) : \quad \tilde{g}(x, y) := \sum_{|\nu| = 0}^{m-1} \frac{1}{\nu!} (x^* - x)^\nu \partial_x^\nu g(x^*, y) \\
in y^* , \text{ falls diam}(\Omega_x) > \text{diam}(\Omega_y) : \quad \tilde{g}(x, y) := \sum_{|\nu| = 0}^{m-1} \frac{1}{\nu!} (y^* - y)^\nu \partial_y^\nu g(x, y^*)
\end{align*}
\]
99
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 \( K_1, K_2 \in \mathbb{R}^{r \times s} \) \( R \)-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 \( R \)-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 \( K_1 \in \mathbb{R}^{r \times s}, K_2 \in \mathbb{R}^{s \times t} \) \( H \)-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 \( H_1 H_2 \) eine \( H \)-Approximation von \( K_1 K_2 \) mit

\[\|K_1 K_2 - H_1 H_2\| \leq \varepsilon \|K_1\| \|K_2\| + \varepsilon \|H_1\| \|H_2\| \approx 2 \varepsilon \|K_1\| \|K_2\|.

Hier erhöht sich der Rang entsprechend Satz 5.26 und der relative Fehler ist für \( \|K_1 K_2\| \approx \|K_1\| \|K_2\| \) gleich gut. Sind die zugrundeliegenden Kernfunktionen \( g_1, g_2 \) auf zulässigen Clustern asymptotisch glatt, so erhält man aus

\[ K_1[K_2[f]](x) = \int_{\Omega_y} \int_{\Omega_y} g_1(x,y) g_2(y,z) \, dy \, f(z) \, dz, \]

allerdings nicht die asymptotische Glattheit von \( \tilde{g} \).

**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 \( \Omega \) ein beschränktes Gebiet in \( \mathbb{R}^d \), \( \Gamma := \partial \Omega \) eine \( C^\infty \)-Oberfläche und \( \Omega \) liege jeweils nur auf einer Seite von \( \Gamma \). Wir fixieren einen stark elliptischen linearen Differentialoperator
\( A \) der Ordnung \( 2\alpha_A \) mit \( C^\infty \)-Koeffizienten auf einer offenen Obermenge \( \tilde{\Omega} \) von \( \Omega \) und

setzen Existenz und Eindeutigkeit von Lösungen des Randwertproblems

\[
\begin{align*}
A u(x) &= f(x) \quad x \in \Omega, \quad \text{(37)} \\
u(x) &= 0 \quad x \in \Gamma \quad \text{(38)}
\end{align*}
\]

voraus:

\[
\forall f \in L^2(\Omega) \ \exists u \in H^{2\alpha_A}(\Omega) : \quad A u = f, \quad u|_\Gamma = 0,
\]

\[
\forall u \in H^{2\alpha_A}(\Omega) : \quad A u = 0 \land u|_\Gamma = 0 \Rightarrow u = 0.
\]

Ferner sei die Singularitätenfunktion \( s(x,y) \) (Fundamentallösung, Elementarlösung, in älterer Literatur manchmal Greensche Funktion) mit

\[
\begin{align*}
A \int_{\tilde{\Omega}} s(x,y)f(y) \, dy &= f(x) \quad \forall x \in \tilde{\Omega}, f \in C_0^\infty(\tilde{\Omega}), \\
\int_{\tilde{\Omega}} s(x,y)(Au)(y) \, dy &= u(x) \quad \forall x \in \tilde{\Omega}, u \in C_0^\infty(\tilde{\Omega}).
\end{align*}
\]

bekannt.

**Bemerkung 7.9** (Darstellbarkeit des diskreten Differentialoperators)

Die Lokalität des Differentialoperators \( A \) 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 \( D(X) := \bigcup_{i \in X} \text{supp} \phi_i \) 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 \( T_j, T_j' \) bzw. \( T_j, T_j' \), 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 \( p := 2\alpha_A - |\alpha| - |\beta|, (x, y) \in \Omega \times \Omega \) und \( l(x,y) := \text{dist}(x,\Gamma) + \text{dist}(y,\Gamma) \)

\[
|\partial^\alpha_x \partial^\beta_y g_c(x,y)| \leq C_{\alpha,\beta} \begin{cases} l(x,y)^{\min(p-d,0)} & p \neq d \\ 1 + \log(l(x,y)) & p = d. \end{cases}
\]

(39)
und für eine Lösung $u$ von \((37,38)\), $f \in L^2(\Omega)$ und $x \in \Omega$

$$u(x) = \int_{\Omega} (s(x,y) - g_c(x,y))f(y) \, dy$$
gilt.

**Beweis:** [28, VI., Theorem 4.2]

**Definition 7.12** (Erweiterte Standard-Zulässigkeitsbedingung)
Gegeben seien Indexmengen $I$ und $J$, $H$-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 $\Omega$ sei vorgegeben. Ein Knoten $r \times s \in T_I \otimes T_J$ heißt zum Parameter $\eta$ absolut zulässig, falls für $T := \bigcup_{i \in r} \tau_i \subset \Omega$ und $\sigma := \bigcup_{j \in s} \sigma_j \subset \Omega$

$$\min \{\text{diam}(\tau), \text{diam}(\sigma)\} \leq 2\eta \min \left\{ \frac{\text{dist}(\tau, \Gamma)}{\text{diam}(\tau)}, \frac{\text{dist}(\Gamma, \sigma)}{\text{diam}(\tau, \sigma)} \right\},$$

ist. Die erweiterte Standard-Zulässigkeitsbedingung $\tilde{Z}_\eta$ (zum Parameter $\eta$) ist definiert als

$$\tilde{Z}_\eta(r \times s) := \begin{cases} \text{„zulässig“} & \text{falls } r \times s \text{ absolut zulässig ist,} \\ \text{„nicht zulässig“} & \text{sonst.} \end{cases}$$

**Satz 7.13** (Approximation durch entartete Kerne)
Gegeben sei eine asymptotisch glatte Singularitätenfunktion $s(x,y)$ von $A$:

$$\forall (x,y) \in T \times \sigma: \ |\partial_x^\alpha \partial_y^\beta s(x,y)| \leq C_{\alpha,\beta} \|x - y\|^{-|\alpha|-|\beta|} |s(x,y)|. \tag{42}$$

Das Gebiet $T \times \sigma$ sei absolut zulässig, $x^*$ das Čebyšev-Zentrum von $T$, $y^*$ das Čebyšev-Zentrum von $\sigma$ und $(x,y) \in T \times \sigma$. Dann gilt für die Taylorentwicklung von $g(x,y) := s(x,y) - g_c(x,y)$ bis zur Ordnung $m$

\begin{align*}
\text{in } x^*, \text{ falls } \text{diam}(\tau) \leq \text{diam}(\sigma): & \quad \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 } \text{diam}(\tau) > \text{diam}(\sigma): & \quad \tilde{g}(x,y) := \sum_{|\nu| = 0}^{m-1} \frac{1}{\nu!} (y^* - y)^\nu \partial_y^\nu g(x, y^*)
\end{align*}

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 $\eta$ hinreichend klein ist und die Faktoren $C_{0,0}, C_{0,\beta}$ aus der asymptotischen Glattheitsbedingung $C_{0,0} = O(|\alpha|!)$ und $C_{0,\beta} = O(|\beta|!)$ erfüllen (für $C_{0,0} = O(|\alpha|!C^{[\alpha]})$ erhält man die Abschätzung entsprechend mit $(C\eta)^m$ statt $\eta^m$).

Der Faktor $C'$ hängt vom Gebiet $\Omega$, der Entwicklungsordnung $m$ und der kompensierenden Kernfunktion $g_c$ ab.
Beweis: O.B.d.A. sei $\text{diam}(\tau) \leq \text{diam}(\sigma)$. Der Restterm der Taylorentwicklung bis zur Ordnung $m$ erfüllt für ein $\zeta \in \tau$

$$|g(x, y) - \bar{g}(x, y)| \leq \frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x g(\zeta, y)|$$

$$= \frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x s(\zeta, y) + \partial^\alpha_x g_c(\zeta, y)|$$

$$\leq \frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x s(\zeta, y)| + \frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x g_c(\zeta, y)|.$$

1. Term: Aus der asymptotischen Glattheit von $s$ folgt

$$|\partial^\alpha_x s(\zeta, y)| \leq C_m ||\zeta - y||^{-m} |s(\zeta, y)|.$$ (43)

Sei $\zeta \in \tau$ so gewählt, dass es $|s(\zeta, y)|$ maximiert. Mit Hilfe von

$$|s(\zeta, y)| - |s(x, y)| \leq |s(\zeta, y)| - |s(x, y)|$$

$$\leq |s(\zeta, y) - s(x, y)|$$

$$\leq \|\zeta - x\| \max_{\xi \in [\zeta], x} \partial_\xi s(\xi, y)$$

$$\leq C_2 \|\zeta - x\| \max_{\xi \in [\zeta], x} ||\xi - y||^{-1} |s(\zeta, y)|$$

$$\leq C_2 \text{diam}(\tau) \text{dist}(\tau, \sigma)^{-1} |s(\zeta, y)|$$

erhalten wir (für hinreichend kleines $\eta$) $|s(\zeta, y)| \leq \frac{1}{1 - 2C_2 \eta} |s(x, y)|$ und damit die Abschätzung für den relativen Fehler:

$$\frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x s(\zeta, y)| \leq \frac{1}{m!}||x^* - x||^m C_m ||\zeta - y||^{-m} \frac{1}{1 - 2C_2 \eta} |s(\zeta, y)|$$

$$\leq C_m \frac{1}{1 - 2C_2 \eta} \frac{1}{m!} \left( \frac{1}{2} \text{diam}(\tau) \right)^m \text{dist}(\tau, \sigma)^{-m} |s(x, y)|$$

$$\leq C_m \frac{1}{m!} \frac{1}{1 - 2C_2 \eta} \eta^m |s(x, y)|.$$ (40)

2. Term: Wir nehmen an, dass $m$ so groß ist, dass $p \neq d$ gilt (siehe Satz 7.11). Dann folgt

$$\frac{1}{m!}||x^* - x||^m \max_{|\alpha|=m} |\partial^\alpha_x g_c(\zeta, y)| \leq \frac{C_m}{m!} ||x^* - x||^m l(\zeta, y)^{2\alpha_A - d - m}$$

$$\leq \frac{C_m}{m!} \left( \frac{1}{2} \text{diam}(\tau) \right)^m l(\zeta, y)^{2\alpha_A - d - m}$$

$$\leq C_m \frac{1}{m!} \eta^m l(\zeta, y)^{2\alpha_A - d}.$$
Gesamt:
\[ |g(x, y) - \tilde{g}(x, y)| \leq C_m \frac{1}{m!} \eta_m |s(x, y)| + C_m \eta_m \lambda^{2\alpha-\delta} \]

\[ \text{Lemma 7.14 (Darstellung der diskreten Inversen)} \]
Die Inverse des Operators \( A \) sei in der Darstellung
\[ A^{-1}[f](x) = \int_{\Omega} g(x, y) f(y) \, dy, \quad f \in L^2(\Omega), \]
gegeben. \( A \) sei eine Galerkin-Diskretisierung von \( A \) mit Lagrange-Basisfunktionen \( (\phi_i)_{i=1}^n \) zu Knoten \( (x_i)_{i=1}^n \):
\[ M_{ij} := \int_{\Omega} \phi_i(y) \phi_j(y) \, dy, \]
\[ A_{ij} := (\phi_i, A \phi_j) := \int_{\Omega} \phi_i(y) A \phi_j(y) \, dy. \]

Dann gilt für die Einträge der diskreten Inversen
\[ (A^{-1})_{ij} = \Pi A^{-1}[\tilde{\phi}_j](x_i), \]
\[ \tilde{\phi}_j := \sum_{\nu=1}^n (M^{-1})_{\nu j} \phi_\nu, \]
wobei \( \Pi : H^{2\alpha, A}(\Omega) \to \langle \phi_1, \ldots, \phi_n \rangle \) die Galerkin-Projektion
\[ \forall i \in \{1, \ldots, n\} : (\phi_i, A \Pi u) = (\phi_i, A u) \]
für Funktionen \( u \in H^{2\alpha, A}(\Omega) \) ist.

**Beweis:** Wir setzen \( \tilde{u}^{(j)} := A^{-1}[\tilde{\phi}_j] \). Der Vektor \( u^{(j)} \) der Basisdarstellung von \( \Pi \tilde{u}^{(j)} \) ist \( u_i^{(j)} = (\Pi \tilde{u}^{(j)})[x_i] \). Wir zeigen \( A u^{(j)} = e_j \) für alle \( j = 1, \ldots, n \). Es gilt für alle \( i \in \{1, \ldots, n\} \):
\[ e_i^T A u^{(j)} = \sum_{\nu=1}^n e_i^T A e_\nu u_\nu^{(j)} = \sum_{\nu=1}^n (\phi_i, A e_\nu) u_\nu^{(j)} \]
\[ = (\phi_i, A \Pi \tilde{u}^{(j)}) = (\phi_i, A \tilde{u}^{(j)}) \]
\[ = (\phi_i, \tilde{\phi}_j) \]
\[ = \sum_{\nu=1}^n (M^{-1})_{\nu j} (\phi_i, \phi_\nu) \]
\[ = \sum_{\nu=1}^n (M^{-1})_{\nu j} M_{ij} \]
\[ = (M \cdot M^{-1})_{ij} = \delta_{ij}. \]
Es folgt $(A^{-1})_{ij} = (A^{-1}e_j)_i = u^{(j)}_i = (IL^{-1}[\tilde{\phi}_j])(x_i).$

**Bemerkung 7.15 (Inversion)**

Die diskrete Inverse $A^{-1}$ eines stark elliptischen Differentialoperators $A$ auf einem Gebiet mit glattem Rand entspricht nach Satz 7.11 und Lemma 7.14 bis auf den Discretisierungsfehler (durch $\Pi$) der Diskretisierung eines Integraloperators mit der Kollokationsmethode: Die Kollokationspunkte sind die Knoten $x_i$ der Lagrange-Basisfunktionen $\phi_i$ und die Basisfunktionen sind $(\tilde{\phi}_i)^{n}_{i=1}$. 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 Fehlerschätzungen bei der Approximation durch einen entarteten Kern. Die Gebietsabhängigkeit und die erweiterte Standard-Zulässigkeitsbedingung $\tilde{Z}_\eta$ 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

$$(R^{(\nu-1)}_{\nu
u})^{-1} = ((A^{\nu})^{-1})_{jj}$$

die entsprechenden Inversen von $A$ auf einem Teilgebiet. Eine feinere Clusterung zum Rand, wie es die erweiterte Zulässigkeitsbedingung $\tilde{Z}_\eta$ 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 $A$ 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 Singularitätsfunktion 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

$$g(x, y) = \frac{1}{4\pi} \frac{1}{\|x - y\|_2^{-1}} - \frac{1}{4\pi} \left\| x \right\|_2 \left\| y \right\|_2 \left( y \right\|_2^{-1}$$

105
gegeben ([9, Satz 3.3.1]). Die Singularitätenfunktion \( s(x, y) = \frac{1}{4\pi} \|x - y\|_2^{-1} \) erfüllt offenbar die asymptotische Glättetheitsbedingung. Der Punkt \( y/\|y\|_2 \) liegt auf dem Rand \( \Gamma \) des Gebietes und gemäß Satz 7.11 erwarten wir ein singuläres Verhalten, falls \( x, y \to \Gamma \).

F"ur die Ableitungen der Greenschen Funktion gilt

\[ \forall m \in \mathbb{N} : \lim_{x, y \to \Gamma} \partial^m_{x_i} g(x, y) = 0, \]

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

\[
\begin{align*}
\partial_{x_i} f(x, y) &= -(4\pi)^2 f(x, y)^3 h(x, y), \quad (44) \\
\partial_{x_i} h(x, y) &= c(y) \quad (45)
\end{align*}
\]

gegeben sind. Per Induktion zeigt man leicht, daß es dann ein Polynom \( p_m : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) gibt, welches

\[
\partial^m_{x_i} f(x, y) = p_m(f(x, y), h(h, y), c(y)) \quad (46)
\]

erfüllt. Wir zeigen nun, daß sowohl die Singularitätenfunktion \( s \) als auch die kompensierende Kernfunktion \( g \) die Bedingungen (44) und (45) erfüllen, also für dasselbe Polynom \( p_m \) der Ableitungsvorschrift (46) genügen. Für die Singularitätenfunktion \( s \) gilt

\[
\partial_{x_i} s(x, y) = \partial_{x_i} \frac{1}{4\pi} \|x - y\|_2^{-1} s(x, y) = -(4\pi)^2 s(x, y)^3 (x_i - y_i) = -(4\pi)^2 s(x, y)^3 (x_i - y_i).
\]

Die Funktion \( (x, y) \mapsto (x_i - y_i) \) erfüllt (45) mit \( c(y) \equiv 1 \). Für die kompensierende Kernfunktion \( g_c \) gilt

\[
\partial_{x_i} g_c(x, y) = \partial_{x_i} \frac{1}{4\pi} \|x\|_2^{-1} \|y\|_2^{-1} = -(4\pi)^2 \|x\|_2^{-1} \|y\|_2^{-1},
\]

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

\[
\lim_{y \to \Gamma} \partial^m_{x_i} g(x, y) = \lim_{y \to \Gamma} \partial^m_{x_i} s(x, y) - \lim_{y \to \Gamma} \partial^m_{x_i} g_c(x, y)
\]

\[
\begin{align*}
&= \lim_{y \to \Gamma} p_m(s(x, y), x_i - y_i, 1) - \lim_{y \to \Gamma} p_m(g_c(x, y), \|y\|_2(x_i \|y\|_2 - y_i \|y\|_2^{-1}), \|y\|_2^2) \\
&= \lim_{y \to \Gamma} p_m(s(x, y), x_i - y_i, 1) - \lim_{y \to \Gamma} p_m(s(x, y), x_i - y_i, 1) \\
&= 0.
\end{align*}
\]
8 Anwendungen

Die Anwendungen und numerischen Tests zu hierarchischen Matrizen gliedern sich in drei Teile: Zuerst führten wir für ein Referenzproblem, das Einfachschichtpotential, eine Approximation durch eine \( H \)-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 (\( Z_\eta \)) und für die Rangverteilung (\( k \)) sind hier zu sehen. Als zweites wird die Inversion von Matrizen, die aus der Diskretisierung partieller Differenzialgleichungen stammen, untersucht. Die Singularitätenfunktion \( s \) und die kompensierende Kernfunktion \( g_c \) werden nicht als bekannt vorausgesetzt. Zuletzt werden wir die \( H \)-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

\[
K : L^\infty(\Gamma) \to C(\Gamma),
K[f](x) := \frac{1}{4\pi} \int_{\Gamma} \frac{f(y)}{\| x - y \|_2^2} \, d\Gamma_y
\]

wird mit der Kollokationsmethode diskretisiert:

\[
K : \mathbb{R}^n \to \mathbb{R}^n,
K_{ij} := \frac{1}{4\pi} \int_{\Gamma} \frac{b_j(y)}{\| x_i - y \|_2^2} \, d\Gamma_y.
\]

Als Basis \( (b_j)_{j=1}^n \) wählen wir die charakteristischen Funktionen auf Dreiecken (regelmäßige Oberflächentriangulation von \( \Gamma \)) 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 \( H \times \)-Baum.

8.1.1 Approximation der Kernfunktion durch eine Taylorentwicklung

Die Kernfunktion \( s(x, y) = \| x - y \|_2^{-1} \) wird auf bzgl. der Standard-Zulässigkeitsbedingung \( Z_\eta \) 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 \( K_H \in \mathcal{M}_H \) und der Taylor-Approximation \( K_T \in \mathcal{M}_{H,k}(T, Z_\eta) \) für \( \eta \in \{0.5, 0.8\} \) und \( n \in \{2048, 8192\} \) dargestellt. Die Abhängigkeit von \( \eta^m \) (siehe Lemma 7.7) ist nur schwach zu erkennen: Die Approximation ist wesentlich besser als die Theorie vorhersagt, es scheint so, als sei \( \eta \) wesentlich kleiner gewählt. Im Fall der Taylor-Approximation könnte
man \( \eta \approx 0.25 \) vermuten. Der Grund dafür, daß \( \eta \) 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 \( R_k \)-Matrizen in den zulässigen Blöcken wird in [2] vorgestellt.

Abbildung 16: Approximation der Kollokationsmatrix \( K \) des Einfachschichtpotentials durch eine \( \mathcal{H} \)-Matrix \( K_H \) (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 \( K_\Omega[f] \) mit

\[
K_\Omega : L^\infty(\Gamma) \to C(\Omega),
\]

\[
K_\Omega[f](x) := \frac{1}{4\pi} \int_\Gamma \frac{f(y)}{\|x-y\|_2} \, d\Gamma_y,
\]

nach [10, Lemma 8.1.3] der Potentialgleichung (Laplace-Gleichung)

\[
- \Delta K_\Omega[f] := -(\partial_x^2 + \partial_y^2 + \partial_z^2)K_\Omega[f] = 0
\]

in \( \Omega \). Unter allen Belegungen \( f \), deren Einfachschichtpotential \( K_\Omega[f] \) die Laplace-Gleichung lös't, suchen wir diejenige, welche die Dirichlet-Randbedingung

\[
K[f](x) = \phi(x), \quad x \in \Gamma,
\]

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 \( H \)-Inverse gewonnene Näherungslösung

\[
\tilde{f}(x) := \sum_{i=1}^n (K^\infty v)_i b_i(x), \quad v_i := \phi(x_i),
\]

kann in die \( H \)-Approximation \( K_{\Omega,H} \) des Operators \( K_\Omega \) eingesetzt werden und liefert eine Approximation \( K_{\Omega,H}[\tilde{f}] \) für die Dirichlet-Randwertaufgabe (47,48).

Die Approximation von \( K \) durch eine Matrix \( K_{apx} \in M_{H,k}(T,Z) \) wurde im vorigen Abschnitt untersucht. Die approximative Inverse \( K_{apx}^\infty \) von \( K_{apx} \) erfüllt die Abschätzung

\[
\|I - K_{apx}^\infty \cdot K\|_2 = \|I - K_{apx}^\infty K_{apx} + K_{apx}^\infty (K_{apx} - K)\|_2
\leq \|I - K_{apx}^\infty \cdot K_{apx}\|_2 + \|K_{apx}^\infty (K_{apx} - K)\|_2
\leq \|I - K_{apx}^\infty \cdot K_{apx}\|_2 + \|K_{apx} - K\|_2 \|K_{apx}^\infty\|_2.
\]

Der Inversionsfehler \( \|I - K_{apx}^\infty \cdot K_{apx}\|_2 \) sollte also in der Größenordnung von \( \|K_{apx} - K\|_2 \|K_{apx}^\infty\|_2 \) liegen. Wir berechnen die Inverse von \( K_T \) und \( K_H \) aus dem vorigen Abschnitt mit der Block-Gauß-Elimination (Abschnitt 4.5.1) in \( M_{H,k}(T,Z) \) 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 \( \delta_{apx} := \|K_{apx} - K\|_2 \|K_{apx}^\infty\|_2 \) lassen sich die Inversen offenbar als \( H \)-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.
Das erste Modellproblem wird die Poisson-Gleichung auf dem Einheitsquadrat sein. Für von linearen partiellen Differentialgleichungen.

Die Tests dienen als Referenz für die in diesem Abschnitt zu untersuchenden Inversen. riskisch getestet und ist durch die Theorie in Abschnitt 7.2 bereits hinreichend fundiert. Die Approximation von Matrizen, die aus der Diskretisierung von Integralgleichungen 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 $Z_\eta$ 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.


8.2.1 Das Modellproblem: Poisson-Gleichung

Die Poisson-Gleichung

\[-\triangle u = f \quad \text{in } \Omega,\]
\[u = 0 \quad \text{auf } \Gamma := \partial \Omega\]

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 $\Omega$ führt zu der Aufgabe, die (schwachbesetzte) Matrix

\[A_{ij} := \int_{\Omega} \langle \nabla \phi_j, \nabla \phi_i \rangle, \quad i, j \in I,\]

to invertieren. Die Clusterung der Indexmenge $I = \{1, \ldots, n\}$ erfolgt kardinalitätsbalanciert mit dem BSP-Algorithmus (Beispiel 3.10). Der $H_s$-Baum $T$ ist der minimal aus dem Clusterbaum erzeugte $H_s$-Baum (Beispiel 3.26), wobei wir hier die Standard-Zulässigkeitsbedingung $Z_\eta, \eta = 0.8$, (Abschnitt 3.2.1) und $b_{\text{min}} := 32^2$ zugrundelegen. In Abbildung 19 (links) sind die relativen Fehler $\|I - A^\Theta A\|_2$ für die formatierte Inversion (mit der Block-Gauß-Elimination aus Abschnitt 4.5.1) in $M_{H,k}(T, Z_\eta)$ für zunehmenden Rang $k$ und Problemgöße $n$ 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 $H$-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 $H$-Inverse $A^\Theta$ so genau bestimmt werden, daß
Abbildung 19: Relativer Fehler $\|I - A^{\ominus}A\|_2$ zur Approximation der Inversen des Laplace-Operators mit Rang $k$ und $n$ Freiheitsgraden auf dem Einheitsquadrat.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$32^2$</th>
<th>$64^2$</th>
<th>$128^2$</th>
<th>$256^2$</th>
<th>$512^2$</th>
<th>Quotient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$5.2_{-1}^{10}$</td>
<td>$2.1_{-0}^{10}$</td>
<td>$7.6_{-0}^{10}$</td>
<td>$2.4_{+1}^{10}$</td>
<td>$4.9_{+1}^{10}$</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>$3.5_{-2}^{10}$</td>
<td>$3.5_{-1}^{10}$</td>
<td>$2.0_{-0}^{10}$</td>
<td>$8.2_{-0}^{10}$</td>
<td>$2.4_{+1}^{10}$</td>
<td>2.9</td>
</tr>
<tr>
<td>3</td>
<td>$4.9_{-3}^{10}$</td>
<td>$3.1_{-2}^{10}$</td>
<td>$2.0_{-1}^{10}$</td>
<td>$1.1_{-0}^{10}$</td>
<td>$5.1_{-0}^{10}$</td>
<td>4.6</td>
</tr>
<tr>
<td>4</td>
<td>$1.1_{-3}^{10}$</td>
<td>$9.1_{-1}^{10}$</td>
<td>$5.1_{-2}^{10}$</td>
<td>$2.7_{-1}^{10}$</td>
<td>$1.2_{-0}^{10}$</td>
<td>4.4</td>
</tr>
<tr>
<td>5</td>
<td>$2.2_{-4}^{10}$</td>
<td>$7.2_{-4}^{10}$</td>
<td>$4.5_{-3}^{10}$</td>
<td>$2.3_{-2}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>6</td>
<td>$4.5_{-5}^{10}$</td>
<td>$2.7_{-4}^{10}$</td>
<td>$1.5_{-3}^{10}$</td>
<td>$7.9_{-3}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>7</td>
<td>$6.4_{-6}^{10}$</td>
<td>$3.1_{-5}^{10}$</td>
<td>$1.7_{-4}^{10}$</td>
<td>$1.1_{-3}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>8</td>
<td>$2.4_{-6}^{10}$</td>
<td>$2.1_{-5}^{10}$</td>
<td>$1.2_{-4}^{10}$</td>
<td>$6.4_{-4}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>9</td>
<td>$6.8_{-7}^{10}$</td>
<td>$4.6_{-6}^{10}$</td>
<td>$2.0_{-5}^{10}$</td>
<td>$9.6_{-5}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>10</td>
<td>$2.9_{-7}^{10}$</td>
<td>$4.1_{-6}^{10}$</td>
<td>$1.9_{-5}^{10}$</td>
<td>$8.0_{-5}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
<tr>
<td>15</td>
<td>$8.4_{-13}^{10}$</td>
<td>$5.4_{-10}^{10}$</td>
<td>$2.2_{-9}^{10}$</td>
<td>$2.8_{-8}^{10}$</td>
<td>$1.0_{-1}^{10}$</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Abbildung 20: Speicherverbrauch (in 1024² Byte) einer $n \times n$-H-Matrix mit Rang $k$.

die Differenz der approximativen Lösung $A^{\ominus}b$ zur Lösung $A^{-1}b$ in der Größenordnung des durch die Finite-Elemente-Methode entstandenen Diskretisierungsfehlers liegt. Zum anderen kann eine approximative Inverse $A^{\ominus}$ als Iterationsmatrix eines linearen Iterationsverfahrens (siehe [8])

$$x^{i+1} := x^i - A^{\ominus}(Ax^i - b), \quad x^0 := 0,$$

verwendet werden. Die Konvergenzrate des Verfahrens ist mindestens $\|I - A^{\ominus}A\|_2$, so daß im Fall $\|I - A^{\ominus}A\|_2 < 0.1$ bereits nach wenigen Schritten die Maschinengenauigkeit
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. $H$-Baum und $\mathcal{H}$-Baum werden wie im vorigen Abschnitt erzeugt, der Parameter für die Zulässigkeit ist $\eta = 0.8$ und die minimale Blockgröße ist $b_{\text{min}} = 32^2$. In Abbildung 24 ist die Matrixpartition zu dem minimal aus dem BSP-$H$-Baum der Indexmenge $I = \{1, \ldots , n\}$ erzeugten $\mathcal{H}$-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 $\| I - A^{\ominus}A \|_2$ bei der Inversion der Matrix $A$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k=1$</th>
<th>$k=3$</th>
<th>$k=5$</th>
<th>$k=10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 1280$</td>
<td>$2.6 \cdot 10^{-2}$</td>
<td>$4.5 \cdot 10^{-4}$</td>
<td>$1.4 \cdot 10^{-6}$</td>
<td>$8.7 \cdot 10^{-15}$</td>
</tr>
<tr>
<td>$n = 5456$</td>
<td>$3.3 \cdot 10^{-1}$</td>
<td>$9.6 \cdot 10^{-3}$</td>
<td>$1.8 \cdot 10^{-4}$</td>
<td>$2.8 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>$n = 22256$</td>
<td>$2.1 \cdot 10^{-6}$</td>
<td>$3.2 \cdot 10^{-2}$</td>
<td>$1.3 \cdot 10^{-3}$</td>
<td>$9.7 \cdot 10^{-7}$</td>
</tr>
<tr>
<td>$n = 89648$</td>
<td>$8.0 \cdot 10^{-0}$</td>
<td>$2.2 \cdot 10^{-1}$</td>
<td>$1.1 \cdot 10^{-2}$</td>
<td>$1.1 \cdot 10^{-5}$</td>
</tr>
</tbody>
</table>

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

$$- \text{div} \sigma(x) \nabla u(x) = f(x), \quad x \in \Omega,$$

$$u = 0 \quad \text{auf} \; \Gamma := \partial \Omega,$$
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:

Eine Finite-Elemente-Diskretisierung wie in Problem 4.24 durch lokale Basisfunktionen zu einer regelmäßigen Triangulation von $\Omega$ führt zu der Aufgabe, die (schwachbesetzte) Matrix

$$A_{ij} := \int_{\Omega} \sigma(x) \langle \nabla \phi_j(x), \nabla \phi_i(x) \rangle \, dx, \quad i,j \in I,$$

zu invertieren. Die Clusterung der Indexmenge $I = \{1, \ldots, n\}$ und Partitionierung der Matrix erfolgt wie in Abschnitt 8.2.1 zur Standard-Zulässigkeitsbedingung $Z_n$. Die relativen Fehler $\|I - A^{\ominus} A\|_2$ 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 $\|I - A^{\ominus} A\|_2 \approx \frac{\sqrt{n}}{k}0.26^k$, also dasselbe Ergebnis wie im Fall eines Differentialoperators mit glatten Koeffizienten. Dieses Ergebnis beruht nicht darauf, daß $\sigma$ in großen Teilen des Gebietes konstant ist, denn auch für stochastisch verteiltes $\sigma$ erhält man (in diesem Modellfall) ähnliche Resultate.
Wir wollen nun die adaptive Inversion aus Abschnitt 6 verwenden, um die Inverse bis auf einen relativen Fehler \( \| I - A^\ominus A \|_2 \) von \( \varepsilon \) zu bestimmen. Die Inversion führen wir wie in Algorithmus 6.11 durch, allerdings nur die erste Stufe. Das Ergebnis weist dann einen Fehler \( \varepsilon_{\text{real}} := \| I - A^\ominus A \|_2 \) auf, da die Fehlerverstärkung in der ersten Stufe noch nicht berücksichtigt wurde. Die benötigte Zeit \( t \) zur adaptiven Inversion bis auf den Fehler \( \varepsilon_{\text{real}} \) 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.

Abbildung 25: Relativer Fehler \( \| I - A^\ominus A \|_2 \) zur Approximation der Inversen mit \( n \) Freiheitsgraden und Rang \( k \) auf dem Einheitsquadrat mit springenden Koeffizienten \( \sigma : [0, 1]^2 \rightarrow \mathbb{R}_{>0} \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( 32^2 )</th>
<th>( 64^2 )</th>
<th>( 128^2 )</th>
<th>( 256^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 3.5 \times 10^{-11} )</td>
<td>( 1.1 \times 10^{-11} )</td>
<td>( 3.1 \times 10^{-12} )</td>
<td>( 9.5 \times 10^{-12} )</td>
</tr>
<tr>
<td>2</td>
<td>( 2.4 \times 10^{-10} )</td>
<td>( 1.7 \times 10^{-11} )</td>
<td>( 1.3 \times 10^{-11} )</td>
<td>( 4.3 \times 10^{-12} )</td>
</tr>
<tr>
<td>3</td>
<td>( 6.0 \times 10^{-10} )</td>
<td>( 3.9 \times 10^{-11} )</td>
<td>( 1.3 \times 10^{-10} )</td>
<td>( 5.4 \times 10^{-11} )</td>
</tr>
<tr>
<td>4</td>
<td>( 9.4 \times 10^{-10} )</td>
<td>( 1.0 \times 10^{-11} )</td>
<td>( 3.4 \times 10^{-10} )</td>
<td>( 1.0 \times 10^{-11} )</td>
</tr>
<tr>
<td>5</td>
<td>( 2.6 \times 10^{-9} )</td>
<td>( 2.8 \times 10^{-10} )</td>
<td>( 7.6 \times 10^{-9} )</td>
<td>( 6.6 \times 10^{-9} )</td>
</tr>
<tr>
<td>6</td>
<td>( 1.1 \times 10^{-9} )</td>
<td>( 3.7 \times 10^{-10} )</td>
<td>( 3.1 \times 10^{-9} )</td>
<td>( 1.3 \times 10^{-9} )</td>
</tr>
<tr>
<td>7</td>
<td>( 3.9 \times 10^{-9} )</td>
<td>( 2.1 \times 10^{-9} )</td>
<td>( 4.8 \times 10^{-9} )</td>
<td>( 2.3 \times 10^{-9} )</td>
</tr>
<tr>
<td>8</td>
<td>( 9.6 \times 10^{-9} )</td>
<td>( 6 \times 10^{-9} )</td>
<td>( 1.6 \times 10^{-8} )</td>
<td>( 4.2 \times 10^{-9} )</td>
</tr>
<tr>
<td>9</td>
<td>( 7.8 \times 10^{-9} )</td>
<td>( 4.5 \times 10^{-9} )</td>
<td>( 3.4 \times 10^{-8} )</td>
<td>( 6.2 \times 10^{-9} )</td>
</tr>
<tr>
<td>10</td>
<td>( 7.0 \times 10^{-7} )</td>
<td>( 2.9 \times 10^{-4} )</td>
<td>( 9.7 \times 10^{-4} )</td>
<td>( 2.5 \times 10^{-3} )</td>
</tr>
<tr>
<td>11</td>
<td>( 5.1 \times 10^{-12} )</td>
<td>( 7.9 \times 10^{-9} )</td>
<td>( 8.3 \times 10^{-9} )</td>
<td>( 1.6 \times 10^{-6} )</td>
</tr>
<tr>
<td>12</td>
<td>( 5.9 \times 10^{-12} )</td>
<td>( 2.5 \times 10^{-11} )</td>
<td>( 4.5 \times 10^{-9} )</td>
<td>( 6.3 \times 10^{-9} )</td>
</tr>
</tbody>
</table>

Abbildung 26: Vergleich zwischen adaptiver Rangwahl und konstantem Rang.
8.3 Matrixgleichungen


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

$$J(u, x_0) = \int_0^\infty y(t)^T y(t) + u(t)^T u(t) \, dt \quad (49)$$

minimiert, wobei $y$ durch die (schwache) Lösung $x \in L^2(0, \infty; \mathbb{R}^n)$ der Differentialgleichung

$$\dot{x}(t) = Ax(t) + Bu(t), \quad t \in (0, \infty),$$
$$y(t) = Cx(t),$$
$$x(0) = x_0 \in \mathbb{R}^n$$

definiert ist. Die Matrizen $B \in \mathbb{R}^{n \times n_u}$ und $C \in \mathbb{R}^{n_y \times n}$ sind beliebig vorgegeben, die Matrix $A \in \mathbb{R}^{n \times n}$ 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 $X \in \mathbb{R}^{n \times n}$ die in der Menge der symmetrischen positiv semidefiniten Matrizen eindeutige Lösung von

$$A^T X + X A - X F X + G = 0 \quad (50)$$

8.3.2 Modellproblem: Wärmeleitungsgleichung

Wir betrachten das lineare-quadratische (ortsabhängige) Kontrollproblem der eindimensionalen Wärmeleitungsgleichung

\[ \frac{\partial}{\partial t} x(t, \xi) = \frac{\partial^2}{\partial \xi^2} x(t, \xi) + b(\xi) u(t), \quad \xi \in (0, 1), t \in (0, \infty), \]

\[ x(t, s) = 0, \quad s \in \{0, 1\}, t \in (0, \infty), \]

\[ x(0, \xi) = x_0(\xi), \quad \xi \in (0, 1), \]

\[ y(t) = \int_0^1 c(\xi) x(t, \xi) \, d\xi, \quad t \in (0, \infty), \]

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

\[ b(\xi) := \begin{cases} 1 & \xi \in (0.2, 0.3) \\ 0 & \text{sonst} \end{cases}, \]

\[ c(\xi) := \begin{cases} 1 & \xi \in (0.2, 0.3) \\ 0 & \text{sonst} \end{cases} \]

gegben.

Problem 8.2 (FEM-Modellproblem)

Für die Differentialgleichung (51) führen wir eine Finite-Elemente-Diskretisierung bzgl. der Ortsvariable \( \xi \) mit stückweise affinen Basisfunktionen \( (\phi_i)_{i=1,\ldots,n} \) auf einem regelmäßigen Gitter von \((0, 1)\) mit \( n \) Freiheitsgraden (inneren Gitterpunkten) durch. Die entsprechenden diskreten Operatoren sind

\[ \tilde{A}_{ij}^{\text{FEM}} := - \int_0^1 D\phi_i(\xi) D\phi_j(\xi) \, d\xi \quad i, j = 1, \ldots, n, \]

\[ \tilde{B}_{i1}^{\text{FEM}} := \int_{0.2}^{0.3} \phi_i(\xi) \, d\xi, \quad i = 1, \ldots, n, \]

\[ \tilde{E}_{ij}^{\text{FEM}} := \int_0^1 \phi_i(\xi) \phi_j(\xi) \, d\xi \quad i, j = 1, \ldots, n, \]

\[ A^{\text{FEM}} := (E^{\text{FEM}})^{-1} \tilde{A}^{\text{FEM}}, \]

\[ B^{\text{FEM}} := (E^{\text{FEM}})^{-1} \tilde{B}^{\text{FEM}}, \]

\[ C_{ij}^{\text{FEM}} := \int_{0.2}^{0.3} \phi_j(\xi) \, d\xi \quad i = 1, \ldots, n. \]

Mit den Matrizen \( A^{\text{FEM}}, B^{\text{FEM}}, C^{\text{FEM}} \) liegt die (ortsdiskretisierte) Differentialgleichung (51) in der Form

\[ \dot{x}(t) = A^{\text{FEM}} x(t) + B^{\text{FEM}} u(t), \quad t \in (0, \infty), \]

\[ y(t) = C^{\text{FEM}} x(t), \]

\[ x(0) = x_0 \in \mathbb{R}^n, \]

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 \( \xi \) 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

\[
A_{ij}^{FD} := \begin{cases} 
2h^2 & \text{if } i = j \\
-h^2 & \text{if } |i - j| = 1 \\
0 & \text{otherwise} 
\end{cases} \text{ für } i, j = 1, \ldots, n,
\]

\[
B_{1i}^{FD} := \begin{cases} 
1 & \text{if } i \cdot h \in [0.2, 0.3] \\
0 & \text{otherwise} 
\end{cases} \text{ für } i = 1, \ldots, n,
\]

\[
C_{1j}^{FD} := \int_{0.2}^{0.3} \phi_j(\xi) d\xi, \quad i = 1, \ldots, n.
\]

Die (ortsdiskretisierte) Differentialgleichung (51) ist in diesem Fall

\[
\begin{aligned}
\dot{x}(t) &= A^{FD}x(t) + B^{FD}u(t), \quad t \in (0, \infty), \\
y(t) &= C^{FD}x(t), \\
x(0) &= x_0 \in \mathbb{R}^n.
\end{aligned}
\]

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

\[
A^T X + X A - X F X + G = 0, \quad A, F, G \in \mathbb{R}^{n \times n}, \quad A < 0, \quad F, G \geq 0, \quad (52)
\]

wollen wir lösen. \( X^* \) sei eine symmetrisch positiv semidefinite Lösung von (52).

Satz 8.4 (Lösungsdarstellung)

Wir definieren

\[
\begin{bmatrix} M & N \end{bmatrix} := \text{sign} \left( \begin{bmatrix} A^T & G \\ F & -A \end{bmatrix} \right) - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}
\]

und

\[
X := -(M^T M)^{-1}M^T N.
\]

Dann ist \( X \) eine Lösung von (52).

Beweis: [23]

Die Matrix-Signum-Funktion ist die zur Funktion \( a + ib \mapsto \text{sign}(a) \) 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

\[
S^{(0)} := S, \\
S^{(i+1)} := \frac{1}{2}(S^{(i)} + (S^{(i)})^{-1}), \quad i = 0, \ldots
\]
zur Auflösung von $f(X) := X^2 - I = 0$ gegen $\text{sign}(S)$, 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

\[
S^{(i)} = \frac{1}{2}(S + S^{-1}) = Q\frac{1}{2}(D + D^{-1})Q^T, \\
(S^{(i)})^{-1} = Q2(D + D^{-1})^{-1}Q^T.
\]
Es folgt

\[
\|(S^{(i)})^{-1}\|_2 = \|Q2(D + D^{-1})^{-1}Q^T\|_2 = 2\|(D + D^{-1})^{-1}\|_2 = 2 \max_{i=1,\ldots,n} \frac{1}{|D_{ii} + D_{ii}^{-1}|}.
\]
Da $D_{ii} \in \mathbb{R}$ vorausgesetzt war, ist $|D_{ii} + D_{ii}^{-1}| \geq 2$ und somit $\max_{i=1,\ldots,n} \frac{1}{|D_{ii} + D_{ii}^{-1}|} \leq 1$.

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 $\mathcal{H}_k$-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 $S \in \mathbb{R}^{n \times n}$ und das Spektrum von $S$ berühre die imaginäre Achse nicht. Für die iterierten der formatierten Newton-Iteration
\[
\begin{align*}
\tilde{S}^{(0)} &:= (S)_{\mathcal{H}}, \\
\tilde{S}^{(i+1)} &:= \frac{1}{2} \left( \tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \ominus \right), \quad i = 0, \ldots, i_{\text{max}}
\end{align*}
\]
gelte
\[
\| (\tilde{S}^{(i)})^{-1} - \tilde{S}^{(i)} \|_2 \leq \delta,
\]
\[
\| (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \ominus) - (\tilde{S}^{(i)} + \tilde{S}^{(i)} \ominus) \|_2 \leq \rho,
\]
wobei wir für die Folge
\[
\begin{align*}
c_0 &:= \| \tilde{S}^{(0)} - S \|_2 (\rho + \delta)^{-1}, \\
c_{i+1} &:= \frac{1}{2} (1 + c_i + c_i \frac{\| (S^{(i)})^{-1} \|_2^2}{1 - c_i (\rho + \delta) \| (S^{(i)})^{-1} \|_2})
\end{align*}
\]
die Bedingung
\[
c_i (\rho + \delta) \| (S^{(i)})^{-1} \|_2 < 1, \quad i = 0, \ldots, i_{\text{max}},
\]
(\text{S}^{(i)} \text{ wie in Satz 8.5) annehmen. Dann gilt die Abschätzung}
\[
\| \tilde{S}^{(i)} - S^{(i)} \|_2 \leq c_i (\rho + \delta), \quad i = 0, \ldots, i_{\text{max}}.
\]

Beweis: Wir beweisen die Behauptung per Induktion. Der Induktionsanfang ist nach Voraussetzung an $c_0$ erfüllt. Wir definieren
\[
\begin{align*}
E^{(i)} &:= S^{(i)} - \tilde{S}^{(i)}, \\
D^{(i)} &:= \tilde{S}^{(i)} \ominus - (\tilde{S}^{(i)})^{-1}, \\
R^{(i)} &:= (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \ominus) - (\tilde{S}^{(i)} + \tilde{S}^{(i)} \ominus).
\end{align*}
\]
Es folgt
\[
\begin{align*}
\tilde{S}^{(i+1)} &= \frac{1}{2} (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \ominus) \\
&= \frac{1}{2} (\tilde{S}^{(i)} + \tilde{S}^{(i)} \ominus) + \frac{1}{2} R^{(i)} \\
&= \frac{1}{2} (S^{(i)} - E^{(i)} + (S^{(i)} - E^{(i)})^{-1} + D^{(i)}) + \frac{1}{2} R^{(i)} \\
&= \frac{1}{2} (S^{(i)} - E^{(i)} + (S^{(i)})^{-1} \sum_{\nu=0}^{\infty} (E^{(i)} (S^{(i)})^{-1})^\nu + D^{(i)}) + \frac{1}{2} R^{(i)} \\
&= S^{(i+1)} - \frac{1}{2} E^{(i)} + \frac{1}{2} (S^{(i)})^{-1} \sum_{\nu=1}^{\infty} (E^{(i)} (S^{(i)})^{-1})^\nu + \frac{1}{2} D^{(i)} + \frac{1}{2} R^{(i)}.
\end{align*}
\]
Ab dem zweiten Schritt nutzen wir die Definition von \( c_{i+1} \) aus:

\[
\| \tilde{S}(i+1) - S(i+1) \|_2 \leq c_i \frac{(\rho + \delta)}{2} + \frac{1}{2}(\rho + \delta) \\
+ \frac{c_i}{2} \| (S(i))^{-1} \|_2 (\rho + \delta) \| (S(i))^{-1} \|_2 \sum_{\nu=0}^{\infty} (c_i (\rho + \delta) \| (S(i))^{-1} \|_2)^\nu \\
\leq \frac{c_i}{2} (\rho + \delta) + \frac{1}{2} (\rho + \delta) \\
+ \frac{c_i}{2} \| (S(i))^{-1} \|_2 (\rho + \delta) \| (S(i))^{-1} \|_2 \frac{1}{1 - c_i (\rho + \delta) \| (S(i))^{-1} \|_2} \\
= \frac{1}{2} (1 + c_i + c_i \frac{\| (S(i))^{-1} \|_2^2}{1 - c_i (\rho + \delta) \| (S(i))^{-1} \|_2}) (\rho + \delta) \\
= c_{i+1} (\rho + \delta).
\]

\[\square\]

**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 \( \rho \) bzw. \( \delta \) aufweisen. Dies kann man im Fall der adaptiven \( H \)-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 \( \| \text{sign}(S)^{-1} \|_2 = \| \text{sign}(S) \|_2 \geq 1 \)), 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 \( \rho, \delta \) wie \( c_i \approx O(\| (S(i))^{-1} \|_2) \), 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 \( \alpha S \) 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 \( \alpha^{-1} \) kleinere \( \rho, \delta \) erfüllt ist und außerdem die Faktoren \( c_i \) schneller exponentiell anwachsen.
Für den Fall, daß \( S \) in \( \mathcal{M}_{H,k}(T, Z) \) 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 \( \pm 1 \) heranzubringen. Die Norm des betragsmäßig kleinsten Spektralwertes \( \lambda_{\text{min}} \) schätzen wir durch
\[
\lambda_{\text{min}} := \|S^{-1}\|_2^{-1},
\]
die Norm des betragsmäßig größten Spektralwertes durch
\[
\lambda_{\text{max}} := \|S\|_2.
\]
Der Faktor \( \alpha \), für den die geschätzten Extrema \( \lambda_{\text{min}}(\alpha S) \cdot \lambda_{\text{max}}(\alpha S) = 1 \) erfüllen, ist
\[
\alpha := \sqrt{\frac{\|S^{-1}\|_2}{\|S\|_2}} = \sqrt{\text{cond}_2(S)}\|S\|_2^{-1}.
\]

**Lemma 8.10** *(Lokale Konvergenz bei formatierter Arithmetik)*

Sei \( T \) ein \( H_x \)-Baum, \( Z \) eine Zulässigkeitsbedingung und \( k \) eine Rangverteilung auf \( T \). Sämtliche Operationen der formatierten \( H \)-Arithmetik beziehen sich auf \( \mathcal{M}_{H,k}(T, Z) \). Sei \( S \in \mathbb{R}^{n,n} \) und das Spektrum von \( S \) berühre die imaginäre Achse nicht. Für die Iterierten der formatierten Newton-Iteration
\[
\tilde{S}^{(0)} := (S)_{H},
\]
\[
\tilde{S}^{(i+1)} := \frac{1}{2} \left( \tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \odot \right), \quad i = 0, \ldots
\]
gelte
\[
\|(\tilde{S}^{(i)})^{-1} - \tilde{S}^{(i)} \odot\|_2 \leq \delta,
\]
\[
\|(\tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \odot) - (\tilde{S}^{(i)} + \tilde{S}^{(i)} \odot)\|_2 \leq \rho,
\]
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
\[
\|(\tilde{S}^{(0)})^2 - I\|_2 =: q \leq \frac{1}{4},
\]
\[
\rho + \delta \leq \frac{1}{4} \sigma^{-1},
\]
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
\[
E^{(i)} := I - (\tilde{S}^{(i)})^2, \\
D^{(i)} := \tilde{S}^{(i)} \odot (\tilde{S}^{(i)})^{-1}, \\
R^{(i)} := (\tilde{S}^{(i)} \oplus \tilde{S}^{(i)} \odot) - (\tilde{S}^{(i)} + \tilde{S}^{(i)} \odot).
\]

Die Behauptung wird per Induktion bewiesen, der Induktionsanfang ist nach Definition von \( q \) erfüllt. Es gilt
\[
(\tilde{S}^{(i+1)})^2 = \left( \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1}) + \frac{1}{2}D^{(i)} + \frac{1}{2}R^{(i)} \right)^2 \\
= \frac{1}{4}(\tilde{S}^{(i)})^2 + \frac{1}{2}I + \frac{1}{4}(\tilde{S}^{(i)})^{-2} + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\
+ \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}) \\
= \frac{3}{4}I - \frac{1}{4}E^{(i)} + \frac{3}{4} \sum_{\nu=0}^{\infty} (E^{(i)})^\nu + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\
+ \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}) \\
= I + \frac{1}{4} \sum_{\nu=2}^{\infty} (E^{(i)})^\nu + \frac{1}{4}(D^{(i)} + R^{(i)})^2 \\
+ \frac{1}{2}(\tilde{S}^{(i)} + (\tilde{S}^{(i)})^{-1})(D^{(i)} + R^{(i)}),
\]

\[
\| (\tilde{S}^{(i+1)})^2 - I \|_2 \leq \frac{1}{4} \frac{\| E^{(i)} \|_2^2}{1 - \| E^{(i)} \|_2^2} + \frac{1}{4}(\rho + \delta)^2 + \frac{1}{2}\sigma(\rho + \delta).
\]

Setzt man für \( \| E^{(i)} \|_2 \) die Induktionsvoraussetzung ein, so bleibt zu zeigen:
\[
\frac{(q^{2i+1} + \sigma(\rho + \delta))^2}{1 - q^{2i+1} - \sigma(\rho + \delta)} \leq q^{2i+1} + (\sigma - \frac{1}{4}(\rho + \delta) - \frac{1}{2}\sigma)(\rho + \delta).
\]

Nach dem Ausmultiplizieren genügt es, die folgenden beiden Aussagen zu zeigen:
\[
q^{2i+1} \leq q^{2i+1} - q^{2i+1}q^{2i+1},
\]
\[
(2q^{2i+1} + \sigma^2(\rho + \delta))(\rho + \delta) \leq (1 - q^{2i+1} - \sigma(\rho + \delta))(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta))(\rho + \delta).
\]

Die erste Aussage ist für \( q \leq \frac{1}{4} \) erfüllt, die zweite Aussage folgt der Reihe nach:
\[
(2q^{2i+1} + \sigma^2(\rho + \delta))(\rho + \delta) \leq (1 - q^{2i+1} - \sigma(\rho + \delta))(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta))(\rho + \delta)
\]
\[
\leq \frac{1}{8} + \sigma^2(\rho + \delta) \leq \left( \frac{15}{16} - \sigma(\rho + \delta) \right)(\frac{1}{2}\sigma - \frac{1}{4}(\rho + \delta))
\]

124
\[ \begin{aligned} & \left\langle \frac{1}{8} + \sigma^2 (\rho + \delta) \leq \frac{15}{32}\sigma - \frac{1}{2}\sigma^2 (\rho + \delta) - \frac{15}{64} (\rho + \delta) + \frac{1}{4}\sigma (\rho + \delta)^2 \right. \\
& \left. \left\langle \frac{1}{8} + \frac{3}{2}\sigma^2 (\rho + \delta) + \frac{15}{64} (\rho + \delta) \leq \frac{15}{32}\sigma \right. \\
& \left. \left\langle \sigma \geq 2 \right. \\
& \left. \left\langle \frac{1}{16} + \frac{3}{2}\sigma^2 (\rho + \delta) + \frac{15}{250} (\rho + \delta) \sigma^2 \leq \frac{15}{32}\sigma \right. \\
& \left. \left\langle \frac{1}{16} + \frac{399}{296}\sigma (\rho + \delta) \leq \frac{15}{32}\sigma \right. \\
& \left. \left\langle (\rho + \delta) \leq \frac{256}{399} \cdot 13\sigma^{-1} \right. \\
& \left. \left\langle (\rho + \delta) \leq \frac{1}{4}\sigma^{-1}. \right. \end{aligned} \]

\[ \blacksquare \]

**Bemerkung 8.11 (Abbruchkriterium in der Newton-Iteration)**

Ein Abbruchkriterium für die formatierte Newton-Iteration (Satz 8.8) ist

\[ ||I - (\tilde{S}^{(i)})^2||_2 \leq \text{MACH_EPS}, \]

wobei \text{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. \( \text{sign}(S) \) 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 \((M^TM)^{-1}\) 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 \(\|I - A^\odot A\|_2\) als Güte für die approximative Inverse \(A^\odot\) gewählt (siehe Abschnitt 8.2.1). Hier ist

\[
\frac{\|A^{-1} - A^\odot\|_2}{\|A^{-1}\|_2} = \frac{\|(I - A^\odot A)A^{-1}\|_2}{\|A^{-1}\|_2} \leq \|I - A^\odot A\|_2,
\]

also \(\|I - A^\odot A\|_2\) 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

\[
R(\tilde{X}) := \tilde{A} \tilde{X} + \tilde{X} A^T - \tilde{X} F \tilde{X} + G
\]

eine leicht zu berechnende Größe. Wegen \(R(X) = 0\) gilt

\[
R(\tilde{X}) = R(\tilde{X}) - R(X) = A^T(\tilde{X} - X)A - \tilde{X} F \tilde{X} + X F \tilde{X} \\
\approx A^T(\tilde{X} - X)A, \\
\frac{\|R(\tilde{X})\|_2}{\|X\|_2} \approx \frac{\|	ilde{X} - X\|_2}{\|X\|_2} \\
\frac{\|	ilde{X} - X\|_2}{\|X\|_2} \approx \frac{\|R(\tilde{X})\|_2}{\|X\|_2}.
\]

In der Praxis zeigt sich allerdings, daß für zunehmende Problemgröße \(n\) die Schätzung für den relativen Fehler bei kleinem Rang (→ grobe Approximation) ungenau wird (die Terme höherer Ordnung sind dann nicht vernachlässigbar).

**Folgerung 8.13 (Lösung der Lyapunov-Gleichung)**

Die Lösung der Lyapunov-Gleichung

\[
AX + X A^T + G = 0
\]

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
\[
A^{(0)} := A,
G^{(0)} := G,
A^{(i+1)} := \frac{1}{2}(A^{(i)} \oplus A^{(i)} \odot),
G^{(i+1)} := \frac{1}{2}(G^{(i)} \oplus (A^{(i)} \odot)^T G^{(i)} A^{(i)} \odot)
\]
wird und in jedem Schritt nur noch eine Inversion von \(A^{(i)}\) durchzuführen ist. Die Berechnung der Lösung \(X\) wird wegen
\[
\text{sign}(S) = \begin{bmatrix}
-I & G^{(\infty)} \\
0 & I
\end{bmatrix}
\]
zu \(X := \frac{1}{2}G^{(\infty)}\). Die Güte einer approximativen Lösung \(\tilde{X}\) kann man wie in Folgerung 8.12 durch
\[
\frac{\|X - \tilde{X}\|_2}{\|X\|_2} \approx \frac{\|AX + \tilde{X}A^T + G\|_2}{2\|A\|_2\|X\|_2}
\]
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 $A = (E_{FEM})^{-1} \hat{A}_{FEM}$ läßt sich als $\mathcal{H}$-Matrix in der Struktur zur Standard-Zulässigkeitsbedingung $Z_\eta$, $\eta = 0.8$, darstellen (siehe Abschnitt 8.2.1). Die benötigte Zeit zur Berechnung von $\hat{X}$ und des Residuums $R(\hat{X}) = AX + XAT - \hat{X}F\hat{X} + G$ 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 $A$ 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 $\|R(\hat{X})\|_2$ deutlich größer als die Differenz $\|\hat{X} - X\|_2$ zur tatsächlichen Lösung $X$ ist.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$[24]$</th>
<th>101</th>
<th>200</th>
<th>400</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$ t: $\varepsilon*$</td>
<td>2.1 $10^{+3}$</td>
<td>1.6 $10^{+0}$</td>
<td>4.9 $10^{+0}$</td>
<td>2.1 $10^{+1}$</td>
<td>9.2 $10^{+1}$</td>
<td>3.2 $10^{+2}$</td>
<td>1.2 $10^{+3}$</td>
<td>1.3 $10^{+4}$</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>5.7 $10^{-2}$</td>
<td>6.1 $10^{-2}$</td>
<td>2.9 $10^{-2}$</td>
<td>2.6 $10^{-2}$</td>
<td>2.2 $10^{-2}$</td>
<td>1.5 $10^{-2}$</td>
<td>1.1 $10^{-2}$</td>
</tr>
<tr>
<td>$10^{-3}$ t: $\varepsilon*$</td>
<td>2.1 $10^{+0}$</td>
<td>6.3 $10^{+0}$</td>
<td>2.9 $10^{+1}$</td>
<td>1.2 $10^{+2}$</td>
<td>4.3 $10^{+2}$</td>
<td>1.5 $10^{+3}$</td>
<td>1.6 $10^{+4}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>4.5 $10^{-3}$</td>
<td>4.4 $10^{-3}$</td>
<td>6.0 $10^{-3}$</td>
<td>4.2 $10^{-3}$</td>
<td>2.9 $10^{-3}$</td>
<td>2.4 $10^{-3}$</td>
<td>1.7 $10^{-3}$</td>
</tr>
<tr>
<td>$10^{-6}$ t: $\varepsilon*$</td>
<td>2.1 $10^{+0}$</td>
<td>1.2 $10^{+1}$</td>
<td>4.3 $10^{+1}$</td>
<td>1.8 $10^{+2}$</td>
<td>5.9 $10^{+2}$</td>
<td>2.1 $10^{+3}$</td>
<td>2.1 $10^{+4}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>6.5 $10^{-6}$</td>
<td>1.3 $10^{-5}$</td>
<td>3.5 $10^{-5}$</td>
<td>1.2 $10^{-4}$</td>
<td>6.2 $10^{-5}$</td>
<td>6.1 $10^{-5}$</td>
<td>1.1 $10^{-4}$</td>
</tr>
</tbody>
</table>


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 $R\mathcal{K}$-Matrix, für $A$ eine $\mathcal{H}$-Matrix-Struktur zur Standard-Zulässigkeitsbedingung $Z_\eta$, $\eta = 0.8$. Wir wollen nun nicht mehr das Residuum $R(\hat{X})$, sondern den relativen Fehler

$$\varepsilon := \|\hat{X} - X\|_2 / \|X\|_2$$

bestimmen.

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 $H$-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 $R_k$-Strukturen von $F, G, X$ sind in Abbildung 29 zu sehen. In der Newton-Iteration zur Berechnung der sign-Funktion werden $\frac{3}{2} \log_2(n)$ 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.

<table>
<thead>
<tr>
<th>Rang</th>
<th>Anzahl Freiheitsgrade n</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=1</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>$8.8 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>k=2</td>
<td>$2.4 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>k=3</td>
<td>$6.8 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>k=4</td>
<td>$7.7 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>k=5</td>
<td>$4.6 \cdot 10^{-9}$</td>
</tr>
<tr>
<td>k=6</td>
<td>$1.9 \cdot 10^{-10}$</td>
</tr>
</tbody>
</table>

| Iterat. | 12 | 14 | 17 | 20 | 23 | 26 |
| Zeit(k=2) | 2.2 | 8.5 | 67 | 462 | 3033 | 18263 |

Abbildung 29: Der relative Fehler $\varepsilon := \|X - \tilde{X}\|_2/\|X\|_2$ 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 $\mathcal{H}_\kappa$-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 $\mathcal{H}_\kappa$-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.


Die numerischen Tests geben Anlaß zu der Hoffnung, daß sich die Approximationseigenschaft wenigstens in Modellfällen beweisen läßt.
A Implementierung von \( H \)-Matrizen in der Programmiersprache C

A.1 Vorwort


Die grundlegenden Methoden der Linearen Algebra werden unter

\[
\text{http://www.netlib.org/lapack/} \quad (\text{LAPACK} = \text{Linear Algebra Package})
\]
\[
\text{http://www.netlib.org/blas/} \quad (\text{BLAS} = \text{Basic Linear Algebra Subroutines})
\]


\[
\begin{align*}
dcopy_\_ & : \quad \text{Den Inhalt eines Arrays in ein anderes Array kopieren.} \\
dscal_\_ & : \quad \text{Alle Elemente eines Arrays mit einer Zahl multiplizieren.} \\
daxpy_\_ & : \quad \text{Zu einem Array ein um einen Faktor skaliertes Array addieren.} \\
dgemv_\_ & : \quad \text{Matrix-Vektor-Multiplikation.} \\
dgemm_\_ & : \quad \text{Matrix-Matrix-Multiplikation.} \\
dgeqrf_\_ & : \quad \text{Berechnung einer QR-Zerlegung, Q in komprimierter Darstellung.} \\
dorgqr_\_ & : \quad \text{Die komprimierte Darstellung nach dgeqrf\_ auflösen.} \\
dgetrf_\_ & : \quad \text{Berechnung einer LU-Zerlegung.} \\
dgetri_\_ & : \quad \text{Berechnung der Inversen nach einer LU-Zerlegung.} \\
dgesvd_\_ & : \quad \text{Berechnung einer Singulärwertzerlegung.}
\end{align*}
\]
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)**

```c
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 (→ Implementierung A.2). In den zulässigen Blättern $b = r \times s \in \mathcal{L}^+(T)$ ist $M|_b$ in Form einer $\mathbb{R}^k(b)$-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 (→ Implementierung A.4).


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 $\mathbb{R}^k$-Matrix

**Implementierung A.2 (Datenstruktur $n \times m$-Full-Matrix)**

Die Elemente der $n \times m$-Matrix

$$F = \begin{bmatrix} F_{11} & \cdots & F_{1m} \\ \vdots & \ddots & \vdots \\ F_{n1} & \cdots & F_{nm} \end{bmatrix}$$

werden in dem Array $e$ in der Reihenfolge $F_{11}, \ldots, F_{n1}F_{12}, \ldots, F_{n2}, \ldots, F_{nm}$ gespeichert (dies entspricht der Konvention von LAPACK).
### Implementierung A.3 (Auswertung von Full-Matrizen)

Die Auswertung einer $n \times m$-Full-Matrix $f$ erfolgt durch die LAPACK-Funktion `dgemv_`

```c
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 $f \cdot v$ auf einen Vektor $w$ definiert:

```c
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 $n \times m$-R$^k$-Matrix)

Von der $R^k$-Matrix $R = AB^T$ wird $A \in \mathbb{R}^{n \times k}$ spaltenweise in $a$ und $B \in \mathbb{R}^{m \times k}$ spaltenweise in $b$ gespeichert. $k_t$ ist die Anzahl der Vektoren in der Darstellung (6). Der in diesem Block maximal erlaubte Rang ist $k$.

```c
typedef struct _rkmatrix rkmatrix;
typedef rkmatrix *prkmatrix;
struct _rkmatrix {
    int k;          /* maximaler Rang */
    int kt;         /* tatsächlicher Rang */
    int n;
    int m;
    double* a;
    double* b;      /* R = ab^T */
};
```

Das Anlegen und Freigeben einer $R^k$-Matrix (Speicher anfordern) erfolgt durch

```c
prkmatrix new_rkmatrix(int k_new, int n_new, int m_new);
void del_rkmatrix(prkmatrix r);
```
Implementierung A.5 (Auswertung von Rk-Matizen)
Zur Auswertung einer $n \times m$-Rk-Matrix $r$ wird ein Vektor $v_{\text{tmp}}$ zur Zwischenspeicherung benötigt. Die eigentliche Auswertung von $a$ und $b$ erfolgt durch die LAPACK-Funktion $\text{dgemv}_-$. 

```c
void eval_rkmatrix(prkmatrix r, double* v, double* w){ /* $w = r \cdot v$ */
    if(r->kt>0){
        dgemv_(ttrans,&r->m,&r->kt,deins_,r->b,&r->m,v,eins_,
            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 $r \cdot v$ auf einen Vektor $w$ definiert:

```c
void adddeval_rkmatrix(prkmatrix r, double* v, double* w);
void evaltrans_rkmatrix(prkmatrix r, double* v, double* w);
void adddevaltrans_rkmatrix(prkmatrix r, double* v, double* w);
```

Implementierung A.6 (Konvertieren einer Full-Matrix in eine Rk-Matrix)
Die Konvertierung einer Full-Matrix $F'$ in eine $Rk(n,m)$-Matrix $R$ erfolgt mit Hilfe der Singulärwertzerlegung $\text{dgesvd}_-$ aus LAPACK. Für spätere Zwecke ist es günstig, $R := R \oplus F'$ anstelle von $R := F'_H$ 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:

$$
F = 
\begin{bmatrix}
F_{1,1} & \cdots & F_{1,m_F} \\
\vdots & \ddots & \vdots \\
F_{n_F,1} & \cdots & F_{n_F,m_F}
\end{bmatrix}, \quad
F' = 
\begin{bmatrix}
F_{\text{nof},1,\text{mof}+1} & \cdots & F_{\text{nof},1,\text{mof}+m} \\
\vdots & \ddots & \vdots \\
F_{\text{nof},\text{mof},1} & \cdots & F_{\text{nof},\text{mof},\text{mof}+m}
\end{bmatrix}.
$$

```c
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 $R_k$-Matrizen)
Das Kürzen einer $R_k$-Matrix auf einen niedrigeren Rang $k' \leq k$ erfolgt wie in Algorithmus 2.12. Wir sehen jedoch folgende Verallgemeinerung vor: Zur $R_k$-Matrix $rt$ werden $anz$ $R_{k'}$-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 \( R_k \)-Matrix in einer \( R_{k'} \)-Matrix.

```c
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));

    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,rb, &kmax,sigma,u,&kmax,v,&kmax,
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

```c
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 Rk-Matrizen, die in Implementierung A.7 eingeführt wurde.

Analog zur Konvertierung sind die Additionsroutinen

```c
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 H-Matrix

Implementierung A.9 (Datenstruktur Supermatrix)
Die Supermatrix S besteht aus n×m Untermatrizen $S_{ij}$:

$$
S = \begin{bmatrix}
S_{11} & \cdots & S_{1m} \\
\vdots & \ddots & \vdots \\
S_{n1} & \cdots & S_{nm}
\end{bmatrix}
$$

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 $S_{11}, \ldots, S_{n1}, S_{12}, \ldots, S_{n2}, \ldots, S_{nm}$ 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×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 Rk-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++) {
        eval_supermatrix(s_el[sindex], v, &w[windex]);
        windex += s_el[sindex]->sizen;
        sindex++;
    }
    for(j=1; j<m; j++) {
        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 Auf-
addieren von \( s \cdot v \) 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 aufgeru-
fen. Eine Konvertierung von einer \( \mathcal{H} \)-Matrix-Struktur in eine andere wird nicht imple-
mentiert, da wir für die Inversion und Multiplikation voraussetzen, daß die zugrun-
dieliegenden \( \mathcal{H} \)-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 zurück-
greift.

140
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 $R^k$-Matrizen und Full-Matrizen)
Für die hierarchische Approximation und die Multiplikation benötigen wir eine Funktion, die eine neue Matrix mit dem Inhalt $ab$ anlegt. Dabei beschränken wir uns auf den Fall, daß $a,b$ Full-Matrizen sind (→ liefert eine Full-Matrix zurück), oder daß wenigstens eine von beiden eine $R^k$-Matrix ist (→ liefert eine $R^k$-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 $R^k$-Matrix)
Wir führen gleichzeitig die (formatierte) Multiplikation $a \odot b$, Addition $r \oplus a \odot b$ und hierarchische Approximation des Ergebnisses durch einer $R^k$-Matrix $r$ durch:

Die Matrix $r$ darf Teil einer größeren $R^k$-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} */
                    }
                }
            }
        }
    }
}
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 \( R^k \)-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.

```c
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;

    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);
}
```
prkmatrix r;
pfullmatrix f;
if(a->sizen!=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(a->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){
            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){
$$r = \text{get\_mul\_rkmatrix}(a,b);$$
$$\text{add\_rk2fullmatrix}(r,c\to f,0,0);$$
} else {
  if(a->f!=0x0 && b->f!=0x0){
    \text{dgemm}\_(_n\text{trans},n\text{trans},c\to \text{sizen},c\to \text{sizem},a\to \text{sizem},
    a\to \text{e},a\to \text{sizen},b\to \text{e},b\to \text{sizen},
    c\to \text{e},c\to \text{sizen});
  } else {
    f = \text{get\_mul\_fullmatrix}(a,b);
    \text{add2fullmatrix}(c\to f,f);
  }
}
}
}

\textbf{Implementierung A.16 (Inversion von Supermatrizen)}

Die (formatierte) Inversion erfolgt wie in Algorithmus 4.18, d.h. s, s, s w gehören zur selben $H$-Matrix-Struktur und auf der Diagonalen befinden sich nur Supermatrizen oder Full-Matrizen.

\begin{verbatim}
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);
    }
  }
}
\end{verbatim}

144
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);
}
A.4 Newton-Iteration zur Berechnung von $\text{sign}(M)$

Zur Berechnung von $\text{sign}(s)$ werden vier Matrizen $s$, $s_{\text{sign}}$, $s_{\text{work}}$, $s_{\text{work2}}$ von derselben Struktur wie $s$ benötigt. Der Inhalt von $s$ wird überschrieben und zu Beginn muß in $s_{\text{sign}}$ 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 $\text{norm}_2\_\text{supermatrix}\_\text{prominusid}$ und $\text{norm}_2\_\text{supermatrix}$.

```c
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.,

[2] M. Bebendorf:
Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von
Niedrigrang-Matrizen,

[3] H. Fuchs, Z. M. Kedem, B. F. Naylor:
On Visible Surface Generation by a Priori Tree Structures,

[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,

Matrix Computations,

[7] L. Grasedyck, S. Le Borne:
Adaptive refinement and clustering in the context of $\mathcal{H}$-matrices,

[8] W. Hackbusch:
Iterative Lösung großer schwachbesetzter Gleichungssysteme,

[9] W. Hackbusch:
Theorie und Numerik elliptischer Differentialgleichungen,

[10] W. Hackbusch:
Integralgleichungen,

[11] W. Hackbusch:
A Sparse Matrix Arithmetic based on $\mathcal{H}$-Matrices. Part I: Introduction to $\mathcal{H}$-
Matrices,
[12] W. Hackbusch:  
Multi-Grid Methods and Applications,  

[13] W. Hackbusch, B. Khoromskij:  
A sparse H-matrix arithmetic. Part II: Application to multi-dimensional problems,  
Computing 64 (2000), 1, 21-47.

[14] W. Hackbusch, B. Khoromskij:  
A sparse H-matrix arithmetic: general complexity estimates,  

[15] L. Hörmander:  
The Analysis of Linear Partial Differential Operators I,  

[16] L. Hörmander:  
The Analysis of Linear Partial Differential Operators II,  

[17] L. Hörmander:  
The Analysis of Linear Partial Differential Operators III,  

[18] B. W. Kernighan, D. M. Ritchie:  
The C Programming Language, Second Edition,  

[19] D. L. Kleinman:  
On an iterative technique for Riccati equations computation,  

[20] H. W. Knobloch, H. Kwakernaak:  
Lineare Kontrolltheorie,  

[21] C. Lage:  
Softwareentwicklung zur Randelementmethode: Analyse und Entwurf effizienter Techniken,  

[22] K. Mehlhorn:  
Data Structures and Algorithms 1: Sorting and Searching,  
[23] J. D. Roberts:


[25] D. L. Russell:

[26] S. Sauter:

[27] S. Sauter:
Variable order panel clustering,

[28] N. Shimakura:
Partial Differential Operators of Elliptic Type,

[29] G. W. Stewart:
Four Algorithms for the Efficient Computation of Truncated Pivoted QR Approximations to a Sparse Matrix,

[30] J. Stoer:
Einführung in die Numerische Mathematik I,

[31] S. Summit:
C Programming FAQs,

[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
\( C_{sp}, \text{Schwachbesetztheitskonstante} \) ... 67
\( C_{sp}, \text{Summe der stufenweisen Schwachbesetztheitskonstanten} \) ... 88
\( C_{sp}(i) \), stufenweise Schwachbesetztheitskonstante ... 88
\( L_T \), Menge der Stufen mit Blättern ... 27
\( M_{H,T} \), gekürzte Matrix .................................. 43
\( M_{H,T}^b \), Fortsetzung .................................. 42
\( M_{lb} \), Einschränkung .................................. 42
\( N_{H,T}^{\text{best}}(k, T, Z) \) .................................. 85
\( N_{H,T}^{\text{zul}}(k, T, Z) \) .................................. 82
\( N_F, S, I(n, m) \) .................................. 67
\( N_F, V(n, m) \) .................................. 67
\( N_V, R_k(n, m) \) .................................. 16
\( N_{V, k}(n, m) \) .................................. 16
\( N_{H,T, \otimes}(k, T, Z) \) .................................. 86
\( N_{H,T, \oplus}(k, T, Z) \) .................................. 75
\( N_{R_k, V}(k, T, Z) \) .................................. 75
\( N_{R_k, S, \oplus}(n, m) \) .................................. 20
\( N_{R_k, S, \ominus}(n, m) \) .................................. 24
\( N_{R_k, \otimes}(n, m) \) .................................. 24
\( N_{R_k, \oplus}(n, m) \) .................................. 17
\( S_T(q) \), Söhne von \( q \) im Baum \( T \) ... 26
\( T + T' \), Summe von \( H \)-Bäumen ... 38
\( T \cdot T' \), Produkt von \( H \)-Bäumen ... 40
\( T^{(i)} \), Elemente der \( i \)-ten Stufe von \( T \) ... 26
\( T_I \otimes T_j \) .................................. 27
\( U_j \) .................................. 49
\( Z, \text{Zulässigkeitsbedingung} \) .................................. 33
\( Z + Z' \), Summe von Zulässigkeitsbedingungen ... 47
\( Z \cdot Z' \), Produkt von Zulässigkeitsbedingungen ... 49
\( \eta_{\text{zul}} \), Standard-Zulässigkeitsbedingung 33
\( \eta \)-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
\( \mathcal{M}_{H,k}(T, Z) \), \( H \)-Matrizen ... 42
\( \odot \), formatierte \( H \)-Multiplikation ... 50
\( \oplus \), formatierte \( H \)-Addition ... 47
\( \oplus \), formatierte \( R_k \)-Addition ... 24
\( \tau^{(i)} \), Vorfahr der Stufe \( j \) ... 48
\( b_{\text{min}} \), 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
\( \mathcal{C}_k \)-Matrizen ... 25
\( H \)-Approximation ... 42
\( H \)-Baum ... 27
\( H \)-Matrix ... 42
\( H_{c,B} \)-Baum ... 27
\( \mathcal{L}^- \), nicht zulässige Blätter ... 67
\( \mathcal{L}^+ \), zulässige Blätter ... 67
\( R_k \)-Matrix ... 15
\( R_{\leq k} \)-Matrix ... 15

A

adaptive \( H \)-Addition ... 91
adaptive \( H \)-Inversion ... 93
adaptive \( H \)-Multiplikation ... 92
adaptive Konvertierung ... 91
Addition für \( H \)-Matrizen ... 47
allgemeiner \( H \)-Baum ... 27
Alternative Clusterrungen ... 33
approximative Spektralnorm ... 62
approximative Spektralnorm der Inversion ... 92

Aus \( T_I, T_J \) gebildeter \( H \)-Baum ... 37

B

Baum ... 26
Bestapproximation ... 42
Binäre Raumzerlegung ... 28
Blatt ... 26