Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte

Dissertation

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
Mathematik

vorgelegt

von Dipl.-Math. Mike Espig

Leipzig, den
21. Dezember 2007

Vorwort

Die vorliegende Arbeit wurde in der Arbeitsgruppe Wissenschaftliches Rechnen des Max-Planck-Instituts für Mathematik in den Naturwissenschaften in der Zeit von März 2005 bis Dezember 2007 angefertigt.

Ich danke Herrn Professor Dr. Dr. h.c. Wolfgang Hackbusch dafür, dass er mir das Thema dieser Arbeit zur selbständigen Bearbeitung überlassen hat, und für die wertvolle Unterstützung, die ich dabei stets erfahren habe.

Herr Dr. Lars Grasedyck und die anderen Mitarbeiter des Instituts waren jederzeit zu anregenden Diskussionen bereit. Bei ihnen möchte ich mich an dieser Stelle gleichfalls bedanken.

Mein liebevollster Dank gilt meiner Familie, Louisa und Manjana Espig sowie meinen Eltern Petra und Rainer Espig, für ihren Rückhalt.

Leipzig, im Dezember 2007 Mike Espig

Inhalt

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Konvention

- Codestücke und Schlüsselwörter sind im Zeichensatz `Typewriter` gesetzt.
- Dateinamen sind im Zeichensatz Sans Serif gesetzt.

Einleitung

Die rechnergestützte Darstellung von Funktionen und die Diskretisierung partieller Differential- und Integralgleichungen mit Hilfe konventioneller numerischer Methoden ist infolge von zu hohem Speicher- und Rechenbedarf beschränkt auf drei oder vier Raumdimensionen. Viele hochdimensionale Probleme sind schwierig zu lösen, der Rechenaufwand der üblichen numerischen Verfahren wächst exponentiell mit der Dimension d. Mit N Freiheitsgraden kann nur eine Genauigkeit von $\mathcal{O}(N^{-\frac{r}{d}})$ erreicht werden, wobei r ein zusätzlicher Regularitätsparameter des zugrundeliegenden Problems ist. Diese Tatsache ist seit Bellman (1961) als „Fluch der Dimension“ bekannt.

Nach Zenger (1991) stellen dünne Gitter eine anerkannte und etablierte Technik zur Diskretisierung hochdimensionaler Probleme dar. Die wesentlichen Ideen sind dabei nicht neu und wurden bereits in Arbeiten von Babenko (1960) und Smolyak (1963) beschrieben. Dünne Gitter werden mit Hilfe einer Multiskalen-Tensorproduktbasis definiert, welche aus einer eindimensionalen hierarchischen Basis konstruiert ist. Ferner wird die Multiskalen-Tensorproduktbasis derart ausgedünnt, dass ausschließlich Teilräume einbezogen werden, deren Basisfunktionen einen signifikanten Anteil zur Lösung des Problems beitragen, wobei zusätzliche Glattheitseigenschaften vorausgesetzt sind. Die Anzahl der Freiheitsgrade beträgt hier $\mathcal{O}\left(n(\log n)^{d-1}\right)$, wobei n die Anzahl der Gitterpunkte in einer Koordinatenrichtung bezeichnet. Bei stückweise linearen Basisfunktionen wird z. B. bezüglich der L_2-Norm eine Genauigkeit von $\mathcal{O}\left(n^{-2}(\log n)^{d-1}\right)$ erreicht, sofern die gemischten zweiten Ableitungen der Lösung beschränkt sind. Seit den neunziger Jahren werden dünne Gitter insbesondere zur Lösung von Integral- und Differentialgleichungen in hohen Dimensionen eingesetzt, wie zum Beispiel in den Arbeiten von [48, Zenger (1991)], [14, Griebel (1991)], [4, Bungartz (1992)] und [36, Schwab, Todor (2003)] für stochastische elliptische Probleme.

Neuere vielversprechende Methoden setzen ebenfalls eine Tensorproduktbasis voraus. Hier wird versucht, die gesuchte Lösung möglichst genau als Linearkombination der Basisfunktionen darzustellen. Sei etwa u die gesuchte Lösung, welche mit Hilfe der Produkttensorbasis dargestellt ist:

$$u = \sum_{l_1=1}^{n} \cdots \sum_{l_d=1}^{n} \widehat{u}_{(l_1,\ldots,l_d)} \bigotimes_{\mu=1}^{d} \varphi_{l_\mu \mu}.$$

Um den exponentiellen Speicheraufwand für die Koeffizienten $\widehat{u} \in \mathbb{R}^{n^d} \cong \bigotimes_{\mu=1}^{d} \mathbb{R}^n$ zu umgehen, werden $d \cdot r$ Vektoren $\{u_{j\mu} \in \mathbb{R}^n : j \in \mathbb{N}_{\leq r},\, \mu \in \mathbb{N}_{\leq d}\}$ derart bestimmt, dass für alle Multiindizes folgende Approximation bestmöglich erfüllt ist:

$$\widehat{u}_{(l_1,\ldots,l_d)} \approx \underline{u}_{(l_1,\ldots,l_d)} := \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{l_\mu}.$$

Dies bedeutet für die Lösung

$$u \approx \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \left[\sum_{l_\mu=1}^{n} (u_{j\mu})_{l_\mu} \varphi_{l_\mu \mu} \right].$$

Ferner wird $\underline{u} \in \bigotimes_{\mu=1}^{d} \mathbb{R}^n$ mit Hilfe des Kronecker-Produktes wie folgt dargestellt:

$$\underline{u} = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu}.$$

Der Tensor $\underline{u}$ setzt sich aus der Summe elementarer Tensoren zusammen, deshalb nennt man $\underline{u}$ eine Summe von Elementartensoren oder auch Elementartensor-Summe. Ist die Anzahl der Summanden minimal, dann wird r der Tensorrang von $\underline{u}$ genannt. Jeder Tensor kann als Summe von Elementartensoren geschrieben werden und hat demnach einen Tensorrang. Dieser kann allerdings außerordentlich groß sein, denn es gilt hier:

$$r \leq n^{d-1}.$$

Um Praktikabilität und Umsetzbarkeit der neuen, ambitionierten Methoden zu sichern, darf bei hochdimensionalen Problemen der Tensorrang aller beteiligten oder approximativ dargestellten Tensoren einschließlich der Näherungslösung nicht exponentiell von der Dimension abhängen. Ist diese Prämisse erfüllt, dann haben lineare Operationen mit Summen von Elementartensoren eine günstige Komplexität von $\mathcal{O}(dn^p r^q)$ und der Aufwand zur Speicherung von $\underline{u}$ beträgt $\mathcal{O}(dnr)$, wobei $q,\, p \in \mathbb{N}_{\leq 2}$ sind. Für viele bedeutsame Fälle wurde diese Eigenschaft bereits in den Arbeiten von [30, Beylkin, Mohlenkamp (2002)], [42, 43, Tyrtyshnikov (2003, 2004)], [11, Grasedyck (2004)], [7, Gavrilyuk, Hackbusch, Khoromskij (2005)], [17, Hackbusch, Khoromskij, Tyrtyshnikov (2005)] und [19, 20, 21, Hackbusch, Khoromskij (2006, 2006, 2007)] nachgewiesen. In diesen Arbeiten konnte unter anderem gezeigt werden, dass viele hochdimensionale Probleme näherungsweise mit einer Komplexität der Ordnung $\mathcal{O}(dn^p \log^q(n))$ lösbar sind, wobei $p \in \mathbb{N}_{\leq 2}$ und q unabhängig von d sind.
Beim effizienten Einsatz von Elementartensor-Summen tritt folgendes fundamentales Problem auf. Zu einem gegebenen Tensor v mit Tensorrang R ist eine optimale Niedrigtensorrang-Approximation w^* mit Tensorrang $r \in \mathbb{N}_{<R}$ zu bestimmen. D.h., w^* ist derart zu berechnen, dass für alle Tensoren w mit Tensorrang r folgende Ungleichung erfüllt ist:

$$\|w^* - v\| \leq \|w - v\|.$$

Für d=2 kann die Lösung dieser Teilaufgabe mittels Singulärwertzerlegung berechnet werden, siehe z. B. [29]. Bei hohen Dimensionen, $d \geq 3$, wird eine alternierende Methode der kleinsten Quadrate (ALS)[1] verwendet. Beim ALS-Verfahren wird der zur Verfügung stehendes Suchraum partitioniert und anschließend das Optimierungsproblem alternierend auf den einzelnen Partitionen gelöst. Diese Idee wurde bereits von Yates (1933) benutzt. Die Ersten, die das ALS-Verfahren zum Approximieren von Tensoren einsetzten, waren wohl de Leeuw, Young und Takane (1976), siehe [28]. Hier wird der Suchraum so unterteilt, dass man die Variablen einer Raumrichtung zu einer Partition vereinigt. Bei der Berechnung von Lösungen partieller Differentialgleichungen in hohen Dimensionen wurde das ALS-Verfahren von Beylkin und Mohlenkamp (2002, 2005) eingesetzt. Das ALS-Verfahren zum Approximieren von Elementartensor-Summen konvergiert häufig extrem langsam, dies wurde zum Beispiel von Paatero (1997) in [34, Seite 240] an einem Beispiel demonstriert. Aus diesem Grund verwendet Paatero das Gauß-Newton-Verfahren, welches den gesamten Suchraum mit einbezieht. In der Arbeit von Paatero [34] ist das Gauß-Newton-Verfahren für den Fall d=3 zur Approximation von Tensoren mittels Elementartensor-Summen unter Berücksichtigung starker Nebenbedingungen beschrieben. Bei den dort betrachteten Anwendungen besitzt das Gauß-Newton-Verfahren gute Konvergenzeigenschaften. Im Allgemeinen ist aber gute Konvergenz nicht zu erwarten, denn gemäß [24, Abschnitt 3.7, Sätze 1 bis 3, Seiten 81 ff.] konvergiert das Gauß-Newton-Verfahren zur Minimierung von $h(w) := \frac{1}{2}\|w - v\|^2$ nur unter der Voraussetzung

$$w^* = v$$

superlinear. Diese Voraussetzung widerspricht aber der Forderung $r \in \mathbb{N}_{<R}$. Ist diese Bedingung nicht erfüllt, dann kann das Gauß-Newton-Verfahren sogar divergieren, dies zeigt das Beispiel 10.1.1 in [37, Seite 279].

Ziel dieser Arbeit ist es, ein effizientes und robustes Verfahren zu entwickeln, welches das oben beschriebene Approximationsproblem auch in allgemeinen Tensorprodukt-Prähilberträumen effizient löst. Ferner werden zu diesem Zweck die besten Niedrigtensorrang-Approximationen von Elementartensor-Summen analysiert.

Im ersten Kapitel sind grundlegende Eigenschaften des Tensorprodukts und wichtige Definitionen angegeben. Anschließend werden die besten Approximationen mit Summen von Elementartensoren untersucht. Nach dieser Analyse, wird im dritten Kapitel die zu lösende Approximationsaufgabe präzise formuliert und die Zielfunktion angegeben. Zum besseren Verständnis sind bekannte numerische Verfahren zur Behandlung nichtlinearer Optimierungsaufgaben im vierten Kapitel dargestellt. Im fünften Kapitel steht die Lösung der Approximationsaufgabe im Zentrum der Betrachtung. Mittels repräsentativer Beispiele wird die vorgestellte Methode zur Approximation mit Elementartensor-Summen

[1] Alternating Least Squares

im sechsten Kapitel überprüft. Die datentechnische und algorithmische Implementierung wird nach objektorientierten Gesichtspunkten im siebenten Kapitel kurz diskutiert. Die systematische Vorgehensweise beim objektorientierten Programmentwurf wird erörtert. Klassendiagramme stellen die Ergebnisse der Analyse graphisch dar und auf Besonderheiten bei der Realisierung wird kurz eingegangen.

1 Tensorprodukte

1.1 Tensorprodukt von Vektorräumen

Das Wesen des Tensorprodukts besteht darin, dass man zu gegebenen Vektorräumen $A_1, \ldots, A_d$ einen Raum $\mathcal{T}$ und eine multilineare Abbildung t derart konstruiert, dass jede multilineare Abbildung g in einen beliebigen Vektorraum B über das Tensorprodukt faktorisiert wird, zur Anschauung dient Abbildung 1.1; hierbei bezeichnet f_g eine lineare Abbildung. Tatsächlich gibt es immer ein Tensorprodukt mit dieser universellen Eigenschaft.

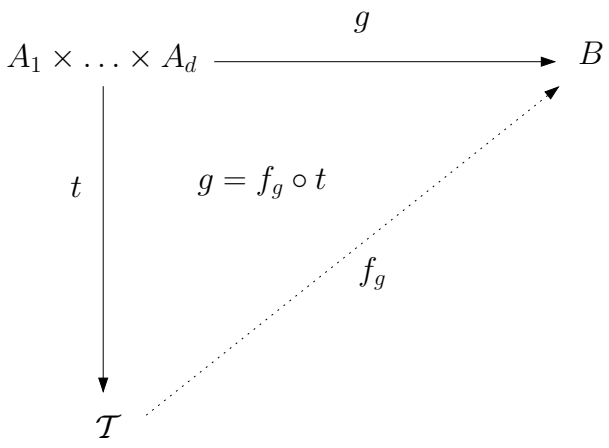


Abbildung 1.1: Das Tensorprodukt der Vektorräume $A_1, \ldots, A_d$.

Im Folgenden seien stets $d \in \mathbb{N}_{\geq 2}$ und $A_1, \ldots, A_d$ reelle Vektorräume.

Definition 1.1.1 (Tensorprodukt)**.** *Seien $\mathcal{T}$ ein Vektorraum und $t : A_1 \times \cdots \times A_d \longrightarrow \mathcal{T}$ eine multilineare Abbildung. Dann heißt das Paar $(\mathcal{T}, t)$ ein* Tensorprodukt *von $A_1, \ldots, A_d$, falls:*

(T1) Das Bild von t ist ein Erzeugendensystem von $\mathcal{T}$.

(T2) Ist B ein weiterer Vektorraum und ist $g : A_1 \times \cdots \times A_d \longrightarrow B$ eine beliebige multilineare Abbildung, so gibt es eine lineare Abbildung $f_g : \mathcal{T} \longrightarrow B$, so dass $g = f_g \circ t$ ist.

Diese Definition sichert noch nicht die Existenz eines Tensorprodukts, sie ergibt sich allerdings aus dem folgenden Satz 1.1.5.

Notation 1.1.2 (Tensorprodukt)**.** *Für ein Tensorprodukt $(\mathcal{T}, t)$ von $A_1, \ldots, A_d$ werden folgende Bezeichnungen verwendet:*

- *Die Vektorräume $A_1, \ldots, A_d$ werden* Ansatzräume *des Tensorprodukts $\mathcal{T}$ genannt.*
- *Man nennt d die* Ordnung *des Tensorprodukts und die Elemente aus $\mathcal{T}$ werden Tensoren der* Ordnung d *genannt.*
- *Seien $a_1 \in A_1, \ldots, a_d \in A_d$. Zur abkürzenden Schreibweise setzt man:*

$$\bigotimes_{\mu=1}^{d} A_\mu := \mathcal{T}, \tag{1.1}$$

$$\bigotimes_{\mu=1}^{d} a_\mu := t(a_1, \ldots, a_d), \tag{1.2}$$

$$\bigotimes_{\mu\in\emptyset} a_\mu := 1 \in \mathbb{R}. \tag{1.3}$$

- *Sei A ein reeller Vektorraum, dann schreibt man vereinfacht:*

$$\otimes^0 A := \mathbb{R}, \tag{1.4}$$

$$\otimes^1 A := A, \tag{1.5}$$

$$\otimes^d A := \otimes_{\mu=1}^{d} A. \tag{1.6}$$

Lemma 1.1.3. *Seien $\mathcal{T}$ ein Vektorraum und $t : A_1 \times \cdots \times A_d \longrightarrow \mathcal{T}$ eine multilineare Abbildung. Dann sind äquivalent:*

(i) Das Paar $(\mathcal{T}, t)$ ist ein Tensorprodukt von $A_1, \ldots, A_d$.

(ii) Ist B ein Vektorraum und ist $g : A_1 \times \cdots \times A_d \longrightarrow B$ eine beliebige multilineare Abbildung, so gibt es genau *eine lineare Abbildung $f_g : \mathcal{T} \longrightarrow B$, so dass $g = f_g \circ t$ ist.*

Beweis. Analog zu [45, Theorem 1.1., Seiten 5f.] ■

Bemerkung 1.1.4. *Die Aussage (ii) wird in der Literatur oft unter dem Namen „universelle Eigenschaft des Tensorprodukts“ erwähnt.*

Satz 1.1.5. *Seien $d \in \mathbb{N}$ und $A_1, \ldots, A_d$ reelle Vektorräume. Dann gibt es ein Tensorprodukt $(\mathcal{T}, t)$ von $A_1, \ldots, A_d$. Dieses ist bis auf Isomorphie eindeutig bestimmt, d.h., ist $(\tilde{\mathcal{T}}, \tilde{t})$ ein weiteres Tensorprodukt, so gibt es einen Isomorphismus $i : \mathcal{T} \longrightarrow \tilde{\mathcal{T}}$, so dass das Diagramm in Abbildung 1.2 kommutiert.*

Beweis. [13, 1.6., 1.7., Seiten 9 ff. und 1.20. Seiten 27 ff.] ■

Lemma 1.1.6. *Die folgenden reellen Vektorräume sind isomorph:*

$$A_1 \otimes A_2 \cong A_2 \otimes A_1, \tag{1.7}$$

$$A_1 \otimes A_2 \otimes A_3 \cong A_1 \otimes (A_2 \otimes A_3), \tag{1.8}$$

$$A_1 \otimes A_2 \otimes A_3 \cong (A_1 \otimes A_2) \otimes A_3. \tag{1.9}$$

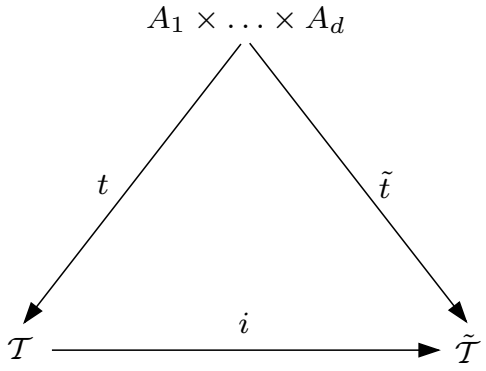


Abbildung 1.2: Das Tensorprodukt ist bis auf Isomorphie eindeutig bestimmt.

Beweis. [45, Proposition 1.5, Seite 10 und Proposition 1.7., Seiten 12 f.] ■

Lemma 1.1.7. *Seien $\mu \in \mathbb{N}_{\leq d}$, $A'_\mu \leq A_\mu$ ein Untervektorraum von A_μ und $(\mathcal{T}, t)$ ein Tensorprodukt von $A_1, \ldots, A_d$. Dann ist das Erzeugnis von $t(A'_1 \times \cdots \times A'_d)$ mit der Restriktion von t auf $A'_1 \times \cdots \times A'_d$ ein Tensorprodukt von $A'_1, \ldots, A'_d$.*

Beweis. Analog zu [45, Proposition 1.6, Seiten 10f.]. ■

Lemma 1.1.8. *Seien $\mu \in \mathbb{N}_{\leq d}$ und $\{a_{\mu,i_\mu} : i_\mu \in I_\mu\}$ Basis von A_μ. Dann gilt:*

- *$\{\otimes_{\mu=1}^d a_{\mu,i_\mu} : (i_1, \ldots, i_d) \in \times_{\mu=1}^d I_\mu\}$ ist eine Basis von $\bigotimes_{\mu=1}^d A_\mu$,*
- $\dim\left(\bigotimes_{\mu=1}^d A_\mu\right) = \prod_{\mu=1}^d \dim(A_\mu)$.

Beweis. Analog zu [45, Proposition 1.2., Seite 8.] ■

Beispiel 1.1.9 (Kroneckerprodukt von Vektoren und Matrizen)**.** *Es seien die Vektorräume $\mathbb{R}^m$ und $\mathbb{R}^n$ mit ihren Standardbasen $\{e_i : i \in \mathbb{N}_{\leq m}\}$ und $\{e_j : j \in \mathbb{N}_{\leq n}\}$ gegeben. Der Vektorraum $\mathbb{R}^{mn}$ habe die Standardbasis $\{e_k : k \in \mathbb{N}_{\leq mn}\}$. Zu jedem $k \in \mathbb{N}_{\leq mn}$ gibt es nach dem euklidischen Algorithmus eindeutig bestimmte $i \in \mathbb{N}_{\leq m}$ und $j \in \mathbb{N}_{\leq n}$ mit $k = (i-1)m + j$. Man setzt nun $t_{ij} := e_k$, falls $k = (i-1)m + j$. $\{t_{ij} : i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}\}$ ist die Basis $\{e_k : k \in \mathbb{N}_{\leq mn}\}$ von $\mathbb{R}^{mn}$. Dann ist $\mathbb{R}^{mn}$ mit folgender bilinearer Abbildung*

$$\otimes : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^{mn} \tag{1.10}$$

$$(x, y) \mapsto \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j t_{ij} \tag{1.11}$$

ein Tensorprodukt von $\mathbb{R}^m$ und $\mathbb{R}^n$.

Beweis.

(T1) Klar ist, dass $\{t_{ij} : i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}\}$ den Vektorraum $\mathbb{R}^{mn}$ erzeugt.

(T2) Sei $g : \mathbb{R}^m \times \mathbb{R}^n \to B$ eine bilineare Abbildung in einen beliebigen reellen Vektorraum B. Bekanntlich ist eine lineare Abbildung durch ihre Werte auf einer Basis festgelegt. Man definiere daher $f_g : \mathbb{R}^{mn} \to B$ durch

$$f_g(t_{ij}) := g(e_i, e_j), \tag{1.12}$$

für alle $i \in \mathbb{N}_{\leq m}$ und alle $j \in \mathbb{N}_{\leq n}$. Dann gilt:

$$\begin{aligned}
(f_g \circ \otimes)(x, y) &= f_g\left(\sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j t_{ij}\right) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j f_g(t_{ij}) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j g(e_i, e_j) \\
&= g(\sum_{i=1}^{m} x_i e_i, \sum_{j=1}^{n} y_j e_j) \\
&= g(x, y),
\end{aligned}$$

für alle $(x, y) \in \mathbb{R}^m \times \mathbb{R}^n$.

$$\Rightarrow \quad g = f_g \circ \otimes.$$

■

In vollkommen analoger Weise kann man zeigen, dass $\mathbb{R}^{rs\times mn}$ *mittels*

$$\otimes : \mathbb{R}^{r\times m} \times \mathbb{R}^{s\times n} \to \mathbb{R}^{rs\times mn} \tag{1.13}$$

$$(X, Y) \mapsto \sum_{i_1=1}^{r}\sum_{i_2=1}^{m}\sum_{j_1=1}^{s}\sum_{j_2=1}^{n} x_{i_1 i_2} y_{j_1 j_2} T_{i_1 i_2 j_1 j_2} \tag{1.14}$$

ein Tensorprodukt von $\mathbb{R}^{r\times m}$ *und* $\mathbb{R}^{s\times n}$ *ist, siehe [45, Seiten 15f.] Die Abbildung (1.10) wird* Kroneckerprodukt[1] *von* $\mathbb{R}^m$ *und* $\mathbb{R}^n$ *genannt. Gleichfalls nennt man Abbildung (1.13) Kroneckerprodukt von* $\mathbb{R}^{r\times m}$ *und* $\mathbb{R}^{s\times n}$*. Ordnet man die mehrfach indizierten Basen lexikographisch an, so schreiben sich die Kroneckerprodukte komponentenweise als Block-Vektor*

$$x \otimes y = \begin{pmatrix} x_1 y \\ \vdots \\ x_m y \end{pmatrix},$$

bzw. als Block-Matrix

$$X \otimes Y = \begin{pmatrix} x_{11}Y & \dots & x_{1m}Y \\ \vdots & \ddots & \vdots \\ x_{r1}Y & \dots & x_{rm}Y \end{pmatrix} \in \mathbb{R}^{rs\times mn}.$$

[1] Leopold Kronecker, deutscher Mathematiker, ∗ 7. Dezember 1823 in Liegnitz, † 29. Dezember 1891 in Berlin.

Lemma 1.1.10. *Sei $x \in A_1 \otimes A_2$. Ist $x = \sum_{j=1}^{r} a_{1,j} \otimes a_{2,j}$ eine Darstellung von x mit minimalem $r \in \mathbb{N}$, dann sind $\{a_{1,j} \in A_1 : j \in \mathbb{N}_{\leq r}\}$ und $\{a_{2,j} \in A_2 : j \in \mathbb{N}_{\leq r}\}$ linear unabhängig.*

Beweis. [13, 1.5., Seite 7.] ■

Lemma 1.1.11. *Seien $r \in \mathbb{N}$, $U_1 := \{a_{1,j} \in A_1 : j \in \mathbb{N}_{\leq r}\}$ und $U_2 := \{a_{2,j} \in A_2 : j \in \mathbb{N}_{\leq r}\}$.*
Ist U_1 linear unabhängig, dann gilt

$$\sum_{j=1}^{r} a_{1,j} \otimes a_{2,j} = 0 \quad \Rightarrow \quad \forall j \in \mathbb{N}_{\leq r} : a_{2,j} = 0. \tag{1.15}$$

Ist U_2 linear unabhängig, dann gilt

$$\sum_{j=1}^{r} a_{1,j} \otimes a_{2,j} = 0 \quad \Rightarrow \quad \forall j \in \mathbb{N}_{\leq r} : a_{1,j} = 0. \tag{1.16}$$

Beweis. [13, 1.5., Seite 7.] ■

Korollar 1.1.12. *Seien $r, r' \in \mathbb{N}$, $U_1 := \{u_{1,j} \in A_1 \setminus \{0\} : j \in \mathbb{N}_{\leq r}\}$, $U_2 := \{u_{2,j} \in A_2 \setminus \{0\} : j \in \mathbb{N}_{\leq r}\}$, $U_1' := \{u_{1,j}' \in A_1 \setminus \{0\} : j \in \mathbb{N}_{\leq r'}\}$, $U_2' := \{u_{2,j}' \in A_2 \setminus \{0\} : j \in \mathbb{N}_{\leq r'}\}$ mit*

$$\sum_{j=1}^{r} u_{1,j} \otimes u_{2,j} = \sum_{j=1}^{r'} u_{1,j}' \otimes u_{2,j}'. \tag{1.17}$$

Dann sind $U_1 \cup U_1'$ und $U_2 \cup U_2'$ linear abhängig.

Beweis. Angenommen $U_1 \cup U_1'$ oder $U_2 \cup U_2'$ seien linear unabhängig, dann folgt wegen Lemma 1.1.11 und

$$\sum_{j=1}^{r} u_{1,j} \otimes u_{2,j} - \sum_{j=1}^{r'} u_{1,j}' \otimes u_{2,j}' = 0, \tag{1.18}$$

dass $U_1 \cup U_1' = \{0\}$ bzw. $U_2 \cup U_2' = \{0\}$. Widerspruch! ■

Korollar 1.1.13. *Seien $a_\mu, b_\mu \in A_\mu \setminus \{0\}$ für alle $\mu \in \mathbb{N}_{\leq d}$. Dann sind folgende Aussagen äquivalent:*

(i) $\bigotimes_{\mu=1}^{d} a_\mu = \bigotimes_{\mu=1}^{d} b_\mu$.

(ii) Es existieren $\lambda_1, \ldots, \lambda_d \in \mathbb{R}$ mit $\prod_{\mu=1}^{d} \lambda_\mu = 1$ und $a_\mu = \lambda_\mu b_\mu$ für alle $\mu \in \mathbb{N}_{\leq d}$.

Beweis.
(ii) ⇒ (i): Klar.

(i) ⇒ (ii): Mit vollständiger Induktion über d.
Induktionsanfang: ($d = 2$)

Seien $a_1, b_1, \in A_1 \setminus \{0\}$ und $a_2, b_2, \in A_2 \setminus \{0\}$ mit $\otimes_{\mu=1}^{2} a_\mu = \otimes_{\mu=1}^{2} b_\mu$. Dann existieren wegen Korollar 1.1.12 $\lambda_1, \lambda_2 \in \mathbb{R}$ mit $a_1 = \lambda_1 b_1$ und $a_2 = \lambda_2 b_2$. Ferner gilt:

$$\begin{aligned} a_1 \otimes a_2 &= (\lambda_1\lambda_2)\,(b_1 \otimes b_2) \\ &= (\lambda_1\lambda_2)\,(a_1 \otimes a_2) \\ \\ \Rightarrow 0 &= (1 - \lambda_1\lambda_2)\,\underbrace{a_1 \otimes a_2}_{\neq 0}, \end{aligned}$$

woraus $1 = \lambda_1\lambda_2$ folgt.
Induktionsschluss:
Seien $\mu \in \mathbb{N}_{\leq d+1}$ und $a_\mu, b_\mu \in A_\mu$ mit $\bigotimes_{\mu=1}^{d+1} a_\mu = \bigotimes_{\mu=1}^{d+1} b_\mu$. Nach zweimaliger Anwendung der Induktionsvoraussetzung existieren $\lambda_1, \ldots, \lambda_d, \lambda'_{d+1}, \lambda \in \mathbb{R}$ mit $1 = \lambda_1\lambda$, $1 = \left(\prod_{\mu=2}^{d} \lambda_\mu\right) \lambda'_{d+1}$ und

$$\forall \mu \in \mathbb{N}_{\leq d} : a_\mu = \lambda_\mu b_\mu, \tag{1.19}$$

$$a_{d+1} = \lambda'_{d+1}\lambda b_{d+1}. \tag{1.20}$$

Man setze daher $\lambda_{d+1} := \lambda'_{d+1}\lambda \in \mathbb{R}$. Dann gilt

$$a_\mu = \lambda_\mu b_\mu$$

für alle $\mu \in \mathbb{N}_{\leq d+1}$ und $\prod_{\mu=1}^{d+1} \lambda_\mu = \lambda_1 \left(\prod_{\mu=2}^{d} \lambda_\mu\right) \lambda'_{d+1}\lambda = \lambda_1\lambda = 1$. ■

1.2 Tensorprodukt linearer Abbildungen

Im Folgenden seien stets $d \in \mathbb{N}$, $A_1, \ldots, A_d$, $B_1, \ldots, B_d$, $C_1, \ldots, C_d$ reelle Vektorräume, $\varphi_\mu \in \mathcal{H}om\,(A_\mu, B_\mu)$[2] und $\psi_\mu \in \mathcal{H}om\,(B_\mu, C_\mu)$ für alle $\mu \in \mathbb{N}_{\leq d}$.
Das Tensorprodukt von Vektorräumen erlaubt die „Linearisierung“ von multilinearen Abbildungen zwischen Vektorräumen. Mit Hilfe der Homomorphismen $\varphi_1, \ldots, \varphi_d$ kann man folgende Abbildung komponentenweise definieren:

$$\bigtimes_{\mu=1}^{d} \varphi_\mu : \bigtimes_{\mu=1}^{d} A_\mu \to \bigtimes_{\mu=1}^{d} B_\mu \tag{1.21}$$

$$(a_1, \ldots, a_d) \mapsto (\varphi_1(a_1), \ldots, \varphi_d(a_d)). \tag{1.22}$$

Die Verkettung von $\times_{\mu=1}^{d} \varphi_\mu$ mit der Tensorabbildung $\otimes_B$ (von $\bigotimes_{\mu=1}^{d} B_\mu$) ist eine multilineare Abbildung. Diese induziert durch $\bigotimes_{\mu=1}^{d} A_\mu$ genau eine lineare Abbildung $\tilde{T}_{(\varphi_1,\ldots,\varphi_d)} \in \mathcal{H}om\left(\bigotimes_{\mu=1}^{d} A_\mu, \bigotimes_{\mu=1}^{d} A_\mu\right)$, siehe zur Anschauung Abbildung 1.3.

[2] $\mathcal{H}om\,(A_\mu, B_\mu) := \{\varphi : A_\mu \to B_\mu : \varphi \text{ ist ein Homomorphismus}\}$

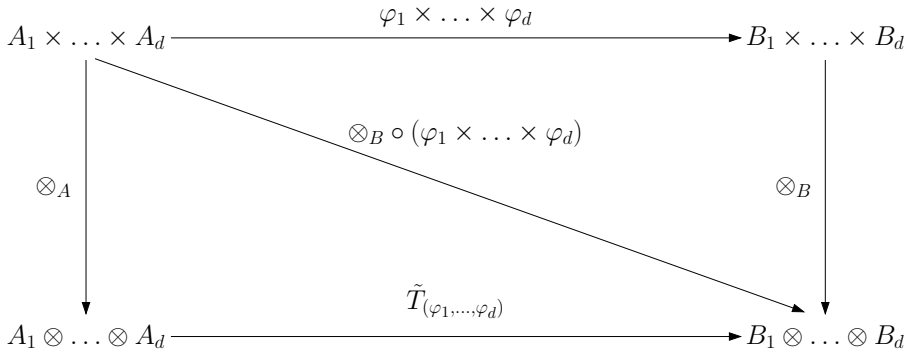


Abbildung 1.3: Das Tensorprodukt linearer Abbildungen.

Die lineare Abbildung $\tilde{T}_{(\varphi_1,\ldots,\varphi_d)}$ ist eindeutig durch

$$\tilde{T}_{(\varphi_1,\ldots,\varphi_d)} \circ \otimes_A = \otimes_B \circ \bigtimes_{\mu=1}^{d} \varphi_\mu \tag{1.23}$$

definiert. D.h. für alle $(a_1, \ldots, a_d) \in \times_{\mu=1}^{d} A_\mu$ gilt

$$\tilde{T}_{(\varphi_1,\ldots,\varphi_d)}(a_1 \otimes \cdots \otimes a_d) = \varphi_1(a_1) \otimes \cdots \otimes \varphi_d(a_d). \tag{1.24}$$

Ferner definiert man folgende Abbildung:

$$\tilde{T} : \bigtimes_{\mu=1}^{d} \mathcal{H}om\,(A_\mu, B_\mu) \;\; \rightarrow \;\; \mathcal{H}om\left(\bigotimes_{\mu=1}^{d} A_\mu, \bigotimes_{\mu=1}^{d} B_\mu\right) \tag{1.25}$$

$$(\varphi_1, \ldots, \varphi_d) \;\; \mapsto \;\; \tilde{T}_{(\varphi_1,\ldots,\varphi_d)}. \tag{1.26}$$

Anhand der definierenden Eigenschaft (1.24) erhält man nun für alle $\mu \in \mathbb{N}$, $\varphi'_\mu \in \mathcal{H}om\,(A_\mu, B_\mu)$ und alle $\alpha, \beta \in \mathbb{R}$:

$$\tilde{T}_{(\varphi_1,\ldots,\alpha\varphi_\mu+\beta\varphi'_\mu,\ldots,\varphi_d)} \;\; = \;\; \alpha\tilde{T}_{(\varphi_1,\ldots,\varphi_\mu,\ldots,\varphi_d)} + \beta\tilde{T}_{(\varphi_1,\ldots,\varphi'_\mu,\ldots,\varphi_d)}. \tag{1.27}$$

Damit ist $\tilde{T}$ eine multilineare Abbildung. Diese induziert wegen der universellen Eigenschaft von $\bigotimes_{\mu=1}^{d} \mathcal{H}om\,(A_\mu, B_\mu)$ einen Homomorphismus

$$T : \bigotimes_{\mu=1}^{d} \mathcal{H}om\,(A_\mu, B_\mu) \;\; \rightarrow \;\; \mathcal{H}om\left(\bigotimes_{\mu=1}^{d} A_\mu, \bigotimes_{\mu=1}^{d} B_\mu\right), \tag{1.28}$$

welcher durch

$$\tilde{T} = T \circ \otimes \tag{1.29}$$

eindeutig bestimmt ist, zur Anschauung siehe Abbildung 1.4.
Im Folgenden identifiziert man die Elemente aus $\otimes_{\mu=1}^{d} \mathcal{H}om\,(A_\mu, B_\mu)$ mit den Elementen aus $\mathcal{H}om\left(\otimes_{\mu=1}^{d} A_\mu, \otimes_{\mu=1}^{d} B_\mu\right)$. Der nachstehende Satz 1.2.1 rechtfertigt diese Identifikation.

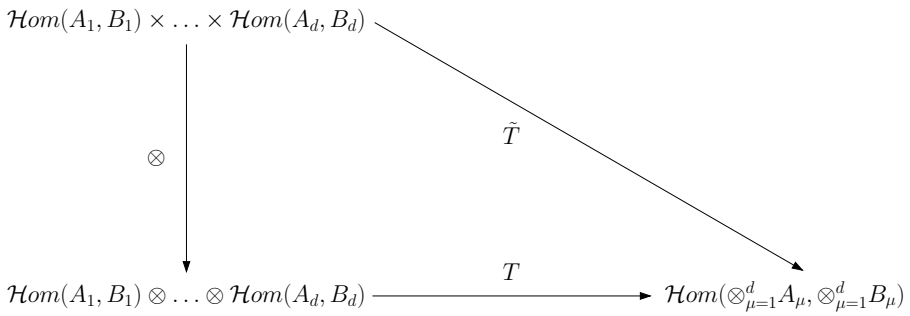


Abbildung 1.4: Der kanonische Homomorphismus T.

Satz 1.2.1. *Der kanonische Homomorphismus (1.28) ist ein Monomorphismus.*

Beweis. [13, 1.16., Seiten 22 f. und 1.20. Seiten 27 ff.] ■

Korollar 1.2.2. *Sind die Räume $A_1, \ldots, A_d$ und $B_1, \ldots, B_d$ endlich dimensional, dann ist der kanonische Homomorphismus (1.28) ein Isomorphismus.*

Beweis. Der Homomorphismus ist injektiv und die beiden Vektorräume haben die gleiche Dimension. ■

Bemerkung 1.2.3. *Sei $(a_1, \ldots, a_d) \in \times_{\mu=1}^{d} A_\mu$. Im Hinblick auf den Satz 1.2.1 wird man die Abbildung $\tilde{T}_{(\varphi_1,\ldots,\varphi_d)}$ mit $\varphi_1 \otimes \cdots \otimes \varphi_d$ identifizieren; dementsprechend schreibt man Gleichung (1.24) wie folgt:*

$$(\varphi_1 \otimes \cdots \otimes \varphi_d)(a_1 \otimes \cdots \otimes a_d) = \varphi_1(a_1) \otimes \cdots \otimes \varphi_d(a_d). \tag{1.30}$$

Daneben gilt nachstehende Kompositionsformel

$$\left(\bigotimes_{\mu=1}^{d} \psi_\mu\right) \circ \left(\bigotimes_{\mu=1}^{d} \varphi_\mu\right) = \bigotimes_{\mu=1}^{d} (\psi_\mu \circ \varphi_\mu). \tag{1.31}$$

Sei $\mu \in \mathbb{N}_{\leq d}$. Ist φ_μ ein Isomorphismus, d.h., existiert $\varphi_\mu^{-1} \in \mathcal{H}om\,(B_\mu, A_\mu)$ mit $\varphi_\mu \circ \varphi_\mu^{-1} = id_{B_\mu}$ und $\varphi_\mu^{-1} \circ \varphi_\mu = id_{A_\mu}$, dann gilt außerdem

$$\left(\bigotimes_{\mu=1}^{d} \varphi_\mu\right)^{-1} = \bigotimes_{\mu=1}^{d} \varphi_\mu^{-1}. \tag{1.32}$$

Ferner ist

$$\mathbf{Id}_{\mathcal{H}om\left(\otimes_{\mu=1}^{d} A_\mu, \otimes_{\mu=1}^{d} A_\mu\right)} = \bigotimes_{\mu=1}^{d} \mathbf{Id}_{A_\mu}. \tag{1.33}$$

Beweis. Gleichung (1.32) folgt unmittelbar aus (1.31). Zu zeigen bleiben (1.31) und (1.33). Wegen (T1) aus Definition 1.1.1 genügt es, die Gleichungen auf

$\{\otimes_{\mu=1}^{d} a_\mu : a_\mu \in A_\mu, \text{ für alle } \mu \in \mathbb{N}_{\leq d}\}$ zu überprüfen. Sei daher $(a_1, \ldots, a_d) \in \times_{\mu=1}^{d} A_\mu$. Es gilt dann:

$$\begin{aligned}\left(\bigotimes_{\mu=1}^{d} \psi_\mu\right) \circ \left(\bigotimes_{\mu=1}^{d} \varphi_\mu\right)\left(\bigotimes_{\mu=1}^{d} a_\mu\right) &= \left(\bigotimes_{\mu=1}^{d} \psi_\mu\right)\left(\bigotimes_{\mu=1}^{d} \varphi_\mu(a_\mu)\right) \\ &= \bigotimes_{\mu=1}^{d} \psi_\mu(\varphi_\mu(a_\mu))\end{aligned}$$

$$\begin{aligned}\bigotimes_{\mu=1}^{d} \psi_\mu(\varphi_\mu(a_\mu)) &= \bigotimes_{\mu=1}^{d} \psi_\mu(\varphi_\mu(a_\mu)) \\ &= \bigotimes_{\mu=1}^{d} (\psi_\mu \circ \varphi_\mu)(a_\mu) \\ &= \left(\bigotimes_{\mu=1}^{d} (\psi_\mu \circ \varphi_\mu)\right)\left(\bigotimes_{\mu=1}^{d} a_\mu\right).\end{aligned}$$

$$\begin{aligned}T\left(\bigotimes_{\mu=1}^{d} \mathbf{Id}_{A_\mu}\right)\left(\bigotimes_{\mu=1}^{d} a_\mu\right) &= \tilde{T}\left(\bigtimes_{\mu=1}^{d} \mathbf{Id}_{A_\mu}\right)\left(\bigotimes_{\mu=1}^{d} a_\mu\right) \\ &= \left(\bigotimes_{\mu=1}^{d} \mathbf{Id}_{A_\mu}\right)\left(\bigotimes_{\mu=1}^{d} a_\mu\right) \\ &= \bigotimes_{\mu=1}^{d} a_\mu.\end{aligned}$$

■

Lemma 1.2.4. *Seien $d \in \mathbb{N}$ und $\varphi_\mu \in \mathcal{H}om\,(A_\mu, B_\mu)$ für alle $\mu \in \mathbb{N}_{\leq d}$. Dann gilt:*

$$\text{Bild}\left(\bigotimes_{\mu=1}^{d} \varphi_\mu\right) = \bigotimes_{\mu=1}^{d} \text{Bild}\,(\varphi_\mu), \tag{1.34}$$

$$\text{Kern}\left(\bigotimes_{\mu=1}^{d} \varphi_\mu\right) = \sum_{\mu=1}^{d} A_1 \otimes \ldots \otimes \text{Kern}\,(\varphi_\mu) \otimes \ldots \otimes A_d. \tag{1.35}$$

Beweis. [13, 1.20. Seiten 29 f.] ■

Beispiel 1.2.5. *Seien $(e_1, \ldots, e_m)$ und $(e_1, \ldots, e_n)$ die geordneten Standardbasen von $\mathbb{R}^m$ und $\mathbb{R}^n$. Für das Tensorprodukt von $\mathbb{R}^m$ und $\mathbb{R}^n$ wählt man $(e_i \otimes e_j : i \in \mathbb{N}_{\leq m},\ j \in \mathbb{N}_{\leq n})$ in lexikographischer Ordnung und verfährt*

analog mit $\mathbb{R}^r$, $\mathbb{R}^s$ *und* $\mathbb{R}^r \otimes \mathbb{R}^s$. *Die linearen Abbildungen* $\varphi : \mathbb{R}^m \to \mathbb{R}^r$, $\psi : \mathbb{R}^n \to \mathbb{R}^s$ *seien bezüglich dieser Basen beschrieben durch die Matrizen* $A \in \mathbb{R}^{r\times m}$, $B \in \mathbb{R}^{s\times n}$. *Sind ferner* $i \in \mathbb{N}_{\leq m}$, $j \in \mathbb{N}_{\leq n}$. *Es gilt*

$$\begin{aligned}(\varphi \otimes \psi)(e_i \otimes e_j) &= \varphi(e_i) \otimes \psi(e_j) \\ &= \left(\sum_{k=1}^{r} a_{ki} e_k\right) \otimes \left(\sum_{l=1}^{s} b_{lj} e_l\right) \\ &= \sum_{k=1}^{r}\sum_{l=1}^{s} a_{ki} b_{lj} (e_k \otimes e_l).\end{aligned}$$

Nun erkennt man leicht, dass die Abbildung $\varphi \otimes \psi$ *bezüglich der lexikographisch angeordneten Basen beschrieben wird durch das Kroneckerprodukt der Matrizen* A *und* B,

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{r1}B & \dots & a_{rm}B \end{pmatrix} \in \mathbb{R}^{rs\times mn}.$$

1.3 Tensorprodukt bilinearer Abbildungen

Im Folgenden seien stets $d \in \mathbb{N}$, $A_1, \dots, A_d$, $B_1, \dots, B_d$, $C_1, \dots, C_d$ reelle Vektorräume.

Lemma 1.3.1. *Seien* $\beta_\mu : A_\mu \times B_\mu \to C_\mu$ *bilineare Abbildungen für* $\mu \in \mathbb{N}_{\leq d}$. *Dann gibt es genau eine bilineare Abbildung*

$$\bigotimes_{\mu=1}^{d} \beta_\mu : \bigotimes_{\mu=1}^{d} A_\mu \times \bigotimes_{\mu=1}^{d} B_\mu \to \bigotimes_{\mu=1}^{d} C_\mu \tag{1.36}$$

mit

$$\left(\bigotimes_{\mu=1}^{d} \beta_\mu\right)\left(\bigotimes_{\mu=1}^{d} a_\mu, \bigotimes_{\mu=1}^{d} b_\mu\right) = \bigotimes_{\mu=1}^{d} \beta_\mu(a_\mu, b_\mu) \tag{1.37}$$

für alle $a_1 \in A_1, \dots, a_d \in A_d$ *und* $b_1 \in B_1, \dots, b_d \in B_d$.

Beweis. Analog zu [13, 1.21., Seiten 30 f.] ■

Lemma 1.3.2. *Seien* $(A_1, \langle , \rangle_1), \dots, (A_d, \langle , \rangle_d)$ *Prähilberträume*[3]. *Dann ist die durch Lemma 1.3.1 definierte bilineare Abbildung*

$$\langle , \rangle := \bigotimes_{\mu=1}^{d} \langle , \rangle_\mu : \bigotimes_{\mu=1}^{d} A_\mu \times \bigotimes_{\mu=1}^{d} A_\mu \to \mathbb{R} \tag{1.38}$$

[3]David Hilbert, deutscher Mathematiker, ∗ 23. Januar 1862, Königsberg (Ostpreußen), † 14. Februar 1943 in Göttingen.

mit

$$\left\langle \bigotimes_{\mu=1}^{d} x_\mu, \bigotimes_{\mu=1}^{d} y_\mu \right\rangle = \prod_{\mu=1}^{d} \langle x_\mu, y_\mu \rangle_\mu \tag{1.39}$$

für alle $x_1 \in A_1, \ldots, x_d \in A_d$ *und* $y_1 \in A_1, \ldots, y_d \in A_d$*, ein Skalarprodukt auf* $\bigotimes_{\mu=1}^{d} A_\mu$*.*

Beweis. Mittels (T1) aus Definition 1.1.1 und Gleichung (1.39) folgt unmittelbar die Symmetrie von $\langle , \rangle$.
Sei $a \in \otimes_{\mu=1}^{d} A_\mu$, etwa $a = \sum_{i=1}^{r} \otimes_{\mu=1}^{d} a_{i\mu}$. Seien ferner $\mu \in \mathbb{N}_{\leq d}$ und $\{z_{l_\mu \mu} \in A_\mu : l_\mu \in \mathbb{N}_{t_\mu}\}$ eine Orthonormalbasis von span $\{a_{i\mu} : i \in \mathbb{N}_{\leq r}\}$. Dann existieren $\alpha_{i\mu} \in \mathbb{R}^{t_\mu}$ mit

$$a_{i\mu} = \sum_{l_\mu=1}^{t_\mu} (\alpha_{i\mu})_{l_\mu} z_{l_\mu \mu}$$

für alle $i \in \mathbb{N}_{\leq r}$. Weiter gilt

$$\begin{aligned} a &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} a_{i\mu} \\ &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \left(\sum_{l_\mu=1}^{t_\mu} (\alpha_{i\mu})_{l_\mu} z_{l_\mu \mu} \right) \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \underbrace{\sum_{i=1}^{r} \prod_{\mu=1}^{d} (\alpha_{i\mu})_{l_\mu}}_{\alpha_{(l_1,\ldots,l_d)} :=} \bigotimes_{\mu=1}^{d} z_{l_\mu \mu} \end{aligned}$$

und damit

$$\begin{aligned} \langle a, a \rangle &= \left\langle \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \alpha_{(l_1,\ldots,l_d)} \bigotimes_{\mu=1}^{d} z_{l_\mu \mu}, \sum_{l'_1=1}^{t_1} \cdots \sum_{l'_d=1}^{t_d} \alpha_{(l'_1,\ldots,l'_d)} \bigotimes_{\mu=1}^{d} z_{l'_\mu \mu} \right\rangle \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \sum_{l'_1=1}^{t_1} \cdots \sum_{l'_d=1}^{t_d} \alpha_{(l_1,\ldots,l_d)} \alpha_{(l'_1,\ldots,l'_d)} \left\langle \bigotimes_{\mu=1}^{d} z_{l_\mu \mu}, \bigotimes_{\mu=1}^{d} z_{l'_\mu \mu} \right\rangle \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \sum_{l'_1=1}^{t_1} \cdots \sum_{l'_d=1}^{t_d} \alpha_{(l_1,\ldots,l_d)} \alpha_{(l'_1,\ldots,l'_d)} \prod_{\mu=1}^{d} \left\langle z_{l_\mu \mu}, z_{l'_\mu \mu} \right\rangle \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \sum_{l'_1=1}^{t_1} \cdots \sum_{l'_d=1}^{t_d} \alpha_{(l_1,\ldots,l_d)} \alpha_{(l'_1,\ldots,l'_d)} \prod_{\mu=1}^{d} \delta_{l_\mu l'_\mu} \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \alpha^2_{(l_1,\ldots,l_d)} \\ &\geq 0 \end{aligned}$$

und

$$\langle a, a \rangle = 0 \Rightarrow a = 0.$$

Hiermit ist $\langle , \rangle$ ein Skalarprodukt auf $\bigotimes_{\mu=1}^{d} A_\mu$. ■

1.4 Darstellung von Tensoren

Im Folgenden seien $d \in \mathbb{N}_{\geq 2}$, $A_1, \ldots, A_d$ reelle Vektorräume und $(t, \mathcal{T})$ ein Tensorprodukt von $A_1, \ldots, A_d$.

Definition 1.4.1 (Elementartensor, Repräsentantensystem)**.** *Man nennt einen Tensor $v \in \mathcal{T}$ einen* Elementartensor *genau dann, wenn $v \in \mathrm{Bild}\,(t)$ ist, d.h.*

$$\forall \mu \in \mathbb{N}_{\leq d} : \exists v_\mu \in A_\mu : v = \bigotimes_{\mu=1}^{d} v_\mu. \tag{1.40}$$

Man nennt das d-Tupel der Vektoren $(v_1, \ldots, v_d)$ aus Gleichung (1.40) ein Repräsentantensystem *von v.*

Bemerkung 1.4.2. *Für einen Elementartensor ist ein Repräsentantensystem nicht eindeutig bestimmt, denn wegen der Multilinearität von t kann man durch Skalieren ein neues System erhalten, welches den selben Elementartensor darstellt, d.h., seien $v = \otimes_{\mu=1}^{d} v_\mu$ und $\lambda_1, \ldots, \lambda_d \in \mathbb{R} \setminus \{1\}$ mit $1 = \prod_{\mu=1}^{d} \lambda_\mu$, dann ist $v = \otimes_{\mu=1}^{d} (\underbrace{\lambda_\mu v_\mu}_{\neq v_\mu})$.*

Da das Bild von t ein Erzeugendensystem von $\mathcal{T}$ ist, kann man jedes $v \in \mathcal{T}$ als Summe von Elementartensoren schreiben. Daher stellt sich die Frage nach der kleinsten Anzahl von erzeugenden Elementartensoren. Diese Fragestellung führt zur Definition des Tensorrangs.

Definition 1.4.3 (Tensorrang, Repräsentantensystem, Summe von Elementartensoren)**.** *Sei $v \in \mathcal{T}$. Unter dem* Tensorrang *von v versteht man folgende natürliche Zahl*

$$\mathrm{rang}_{\mathcal{T}}(v) := \min \left\{ r \in \mathbb{N}_{\geq 0} : v = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu},\ v_{i\mu} \in A_\mu \right\}. \tag{1.41}$$

Ferner ist für $r \in \mathbb{N}$ die Menge der Tensoren mit Tensorrang r bzw. Tensorrang kleiner gleich r wie folgt definiert:

$$\mathcal{T}_r := \{ v \in \mathcal{T} : \mathrm{rang}_{\mathcal{T}}(v) = r \}, \tag{1.42}$$

$$\mathcal{T}_{\leq r} := \{ v \in \mathcal{T} : \mathrm{rang}_{\mathcal{T}}(v) \leq r \}. \tag{1.43}$$

Ist $v \in \mathcal{T}_r$, etwa $v = \sum_{i=1}^{r} \otimes_{\mu=1}^{d} v_{i\mu}$, dann nennt man v eine Summe von Elementartensoren *mit Tensorrang r. Daneben bezeichnet man das System der Vektoren $(v_{i\mu} : i \in \mathbb{N}_{\leq r},\ \mu \in \mathbb{N}_{\leq d})$ als ein* Repräsentantensystem *von v.*

Bemerkung 1.4.4. *Seien $r \in \mathbb{N}$ und $v \in \mathcal{T}_r$.*

- *Als Synonym für Summe von Elementartensoren verwendet man auch Elementartensor-Summe.*

- *In Definition 1.4.3 ist der Tensorrang des Nulltensors gleich* 0 *gesetzt.*

- *Offenbar ist $\mathcal{T}_r$ ein Kegel,*[4] *aber $\mathcal{T}_r$ ist kein Untervektorraum von $\mathcal{T}$, denn seien $v, w \in \mathcal{T}_r$, so ist $v + w \in \mathcal{T}_{\leq 2r} \neq \mathcal{T}_r$.*

- *Analog zu Elementartensoren erkennt man auch hier, dass ein Repräsentantensystem von v nicht eindeutig ist.*

- *Sei $n_\mu := \dim A_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}$. Zum Abspeichern eines Repräsentantensystem s von v werden $r \sum_{\mu=1}^{d} n_\mu$ Einträge benötigt, d.h. der Speicherbedarf wächst nur linear in d.*

- *Ferner gilt im endlich-dimensionalen Fall, $n_\mu := \dim A_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}$, folgende Abschätzung:*

$$r_{\max} := \max\{\operatorname{rang}_{\mathcal{T}}(v) : v \in \mathcal{T}\} \leq \frac{\prod_{\mu=1}^{d} n_\mu}{\max\{n_\mu : \mu \in \mathbb{N}_{\leq d}\}}. \tag{1.44}$$

Beweis. Seien $\mu \in \mathbb{N}_{\leq d}$, $\{v_{i\mu} : i \in \mathbb{N}_{n_\mu}\}$ eine Basis von A_μ und ohne Beschränkung der Allgemeinheit $n_d = \max\{n_\mu : \mu \in \mathbb{N}_{\leq d}\}$. Wegen Lemma 1.1.8 gilt dann für jedes $v \in \mathcal{T}$

$$\begin{aligned} v &= \sum_{l_1=1}^{n_1} \cdots \sum_{l_d=1}^{n_d} a_{(l_1,\ldots,l_d)} \bigotimes_{\mu=1}^{d} v_{l_\mu \mu} \\ &= \sum_{l_1=1}^{n_1} \cdots \sum_{l_{d-1}=1}^{n_{d-1}} \bigotimes_{\mu=1}^{d-1} v_{l_\mu \mu} \bigotimes \left(\sum_{l_d=1}^{n_d} a_{(l_1,\ldots,l_d)} v_{l_d d} \right). \end{aligned}$$

Woraus

$$\operatorname{rang}_{\mathcal{T}}(v) \leq \frac{\prod_{\mu=1}^{d} n_\mu}{\max\{n_\mu : \mu \in \mathbb{N}_{\leq d}\}}$$

folgt. ∎

Bemerkung 1.4.5. *Seien $B_1, \ldots, B_d$ reelle Vektorräume, $(t_B, \mathcal{B})$ ein Tensorprodukt von $B_1, \ldots, B_d$ und $\mathcal{L} := \bigotimes_{\mu=1}^{d} \mathcal{H}om\,(A_\mu, B_\mu)$. Ferner seien $r_1, r_2 \in \mathbb{N}$, $v \in \mathcal{T}_{r_1}$, etwa $v := \sum_{i_1=1}^{r_1} \otimes_{\mu=1}^{d} v_{i_1\mu}$ und $\varphi \in \mathcal{L}_{r_2}$, etwa $\varphi := \sum_{i_2=1}^{r_2} \otimes_{\mu=1}^{d} \varphi_{i_2\mu}$. Dann ist*

$$\varphi(v) \in \mathcal{B}_{\leq r_1 r_2}, \tag{1.45}$$

[4] $A \subseteq X$ ist ein Kegel $:\Leftrightarrow \forall \lambda \in \mathbb{R}_{>0} : \forall a \in A : \lambda a \in A$

denn

$$\begin{aligned}\varphi(v) &= \left(\sum_{i_2=1}^{r_2}\bigotimes_{\mu=1}^{d}\varphi_{i_2\mu}\right)\left(\sum_{i_1=1}^{r_1}\bigotimes_{\mu=1}^{d}v_{i_1\mu}\right)\\ &= \sum_{i_2=1}^{r_2}\sum_{i_1=1}^{r_1}\left(\bigotimes_{\mu=1}^{d}\varphi_{i_2\mu}\right)\left(\bigotimes_{\mu=1}^{d}v_{i_1\mu}\right)\\ &= \sum_{i_2=1}^{r_2}\sum_{i_1=1}^{r_1}\bigotimes_{\mu=1}^{d}\varphi_{i_2\mu}\left(v_{i_1\mu}\right).\end{aligned}$$

In dieser Arbeit ist das Tensorprodukt von Untervektorräumen endlicher Dimension von großer Bedeutung. Sei daher U_μ ein Untervektorraum von A_μ, $t_\mu := \dim U_\mu \in \mathbb{N}$ und $\mathcal{U}_\mu := \{u_{i\mu} : i \in \mathbb{N}_{\le t_\mu}\}$ eine Basis von U_μ für alle $\mu \in \mathbb{N}_{\le d}$. Jeder Tensor $v \in \bigotimes_{\mu=1}^{d} U_\mu$ hat dann gemäß Lemma 1.1.8 die eindeutige Darstellung

$$v = \sum_{l_1=1}^{t_1}\cdots\sum_{l_d=1}^{t_d} a_{(l_1,\ldots,l_d)}\bigotimes_{\mu=1}^{d}u_{l_\mu\mu}. \tag{1.46}$$

Definition 1.4.6 (Koeffiziententensor)**.** *Die Koeffizienten $a_{(l_1,\ldots,l_d)}$ aus Gleichung (1.46) werden zu einem Tensor $a \in \bigotimes_{\mu=1}^{d}\mathbb{R}^{t_\mu}$ zusammengefasst. Dieser Tensor a wird der* Koeffiziententensor *von v bezüglich der Basis $\mathcal{U} := \{\otimes_{\mu=1}^{d}u_{l_\mu\mu} : (l_1,\ldots,l_d) \in \times_{\mu=1}^{d}\mathbb{N}_{\le t_\mu}\}$ genannt. Ist aus dem Zusammenhang klar, welche Basis zugrunde liegt, dann sagt man kürzer: a ist der Koeffiziententensor von v.*

Bemerkung 1.4.7 (Tucker-Zerlegung, Tucker-Rang)**.** *Sind für gegebenes v alle Unterräume U_μ minimal gewählt und $\mathcal{U}_\mu$ eine orthonormale Basis von U_μ ($\mu \in \mathbb{N}_{\le d}$), dann wird die Darstellung (1.46)* Tucker-Zerlegung *von v genannt. Hierbei bezeichnet man $t := (t_1,\ldots,t_d)$ als* Tucker-Rang *von v.*

Um den Koeffiziententensor von v abzuspeichern, benötigt man $\prod_{\mu=1}^{d} t_\mu$ Einträge, d.h. der Speicherbedarf wächst exponentiell in d. Dieser Umstand wird im Fall $v \in \mathcal{T}_r$ nicht eintreten. Diese Tatsache verdeutlicht folgendes Lemma 1.4.8.

Lemma 1.4.8. *Seien $r \in \mathbb{N}$ und $v \in \mathcal{T}_r$, etwa $v = \sum_{i=1}^{r}\otimes_{\mu=1}^{d}v_{i\mu}$. Ferner sei U_μ ein Untervektorraum von A_μ mit $V_\mu := \{v_{i\mu} : i \in \mathbb{N}_{\le r}\} \subset U_\mu$ und $t_\mu := \dim U_\mu$ für alle $\mu \in \mathbb{N}_{\le d}$. Weiter seien $\mathcal{U}_\mu := \{u_{i\mu} : i \in \mathbb{N}_{\le t_\mu}\}$ eine Basis von U_μ, $\mathcal{S} := \bigotimes_{\mu=1}^{d}\mathbb{R}^{t_\mu}$ und $a \in \mathcal{S}$ der eindeutig bestimmte Koeffiziententensor von v. Dann gilt $a \in \mathcal{S}_r$.*

Beweis. Seien $\mu \in \mathbb{N}_{\le d}$ und $i \in \mathbb{N}_{\le r}$. Nach Voraussetzung existiert $a_{i\mu} \in \mathbb{R}^{t_\mu}$ mit

$$v_{i\mu} = \sum_{l_\mu=1}^{t_\mu}(a_{i\mu})_{l_\mu}u_{l_\mu\mu}. \tag{1.47}$$

Es gilt

$$\begin{aligned} v &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \\ &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \left(\sum_{l_\mu=1}^{t_\mu} (a_{i\mu})_{l_\mu} u_{l_\mu \mu} \right) \\ &= \sum_{i=1}^{r} \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \bigotimes_{\mu=1}^{d} \left((a_{i\mu})_{l_\mu} u_{l_\mu \mu} \right) \\ &= \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} \underbrace{\sum_{i=1}^{r} \prod_{\mu=1}^{d} (a_{i\mu})_{l_\mu}}_{a_{(l_1,\ldots,l_d)}=} \bigotimes_{\mu=1}^{d} u_{l_\mu \mu}. \end{aligned}$$

Womit

$$a = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} a_{i\mu} \in \mathcal{S}_{\leq r}$$

gezeigt wurde.
Angenommen, es existieren $r' \in \mathbb{N}_{<r}$ und Vektoren $a'_{i\mu} \in \mathbb{R}^{t_\mu}$ mit

$$a = \sum_{i=1}^{r'} \bigotimes_{\mu=1}^{d} a'_{i\mu}.$$

Gleichung (1.47) induziert eine lineare Abbildung $L_\mu : \mathbb{R}^{t_\mu} \to A_\mu$ mit

$$v_{i\mu} = L_\mu a_{i\mu}.$$

Es folgt

$$\begin{aligned} v &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \\ &= \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} L_\mu a_{i\mu} \\ &= \sum_{i=1}^{r} \left(\bigotimes_{\mu=1}^{d} L_\mu \right) \left(\bigotimes_{\mu=1}^{d} a_{i\mu} \right) \\ &= \left(\bigotimes_{\mu=1}^{d} L_\mu \right) \left(\sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} a_{i\mu} \right) \end{aligned}$$

$$\Rightarrow \quad v = La, \tag{1.48}$$

wobei

$$L := \bigotimes_{\mu=1}^{d} L_\mu : \mathcal{S} \to \mathcal{T} \tag{1.49}$$

gesetzt ist. Andererseits würde aber mit $v'_{i\mu} := L_\mu a'_{i\mu}$ ein Widerspruch zu $v \in \mathcal{T}_r$ folgen, denn

$$\begin{aligned} v &= La \\ &= \left(\bigotimes_{\mu=1}^{d} L_\mu \right) \left(\sum_{i=1}^{r'} \bigotimes_{\mu=1}^{d} a'_{i\mu} \right) \\ &= \sum_{i=1}^{r'} \bigotimes_{\mu=1}^{d} v'_{i\mu}. \end{aligned}$$

■

Bemerkung 1.4.9. *Die Prämissen von Lemma 1.4.8 seien vorausgesetzt und ferner seien $\mu \in \mathbb{N}_{\le d}$ und $n_\mu := \dim A_\mu \in \mathbb{N}$. Der Speicherbedarf des Koeffiziententensors von $v \in \mathcal{T}_r$ beträgt dann $r \sum_{\mu=1}^{d} t_\mu$ und zusätzlich $\sum_{\mu=1}^{d} t_\mu n_\mu$ für die Basen $\mathcal{U}_\mu$. Insgesamt werden zum Abspeichern $\sum_{\mu=1}^{d} t_\mu (r + n_\mu)$ Einträge benötigt, im Gegensatz zur konservativen Speicherung von $r \sum_{\mu=1}^{d} n_\mu$, aus Bemerkung 1.4.4.*

Notation 1.4.10. *Seien $\mu_1 \in \mathbb{N}_{\le d}$ und $(v_1, \ldots, v_d) \in \times_{\mu=1}^{d} A_\mu$. Dann sind $\mathcal{T}^{\mu_1}$ und $v^{\mu_1} \in \mathcal{T}^{\mu_1}$ folgendermaßen definiert:*

$$\mathcal{T}^{\mu_1} := \bigotimes_{\mu \in \mathbb{N}_{\le d} \setminus \{\mu_1\}} A_\mu \tag{1.50}$$

und

$$v^{\mu_1} := \bigotimes_{\mu \in \mathbb{N}_{\le d} \setminus \{\mu_1\}} v_\mu. \tag{1.51}$$

Daneben definiert man für $w \in A_{\mu_1}$ folgende Substitution $\mathfrak{S}_{\mu_1,w}$:

$$\mathfrak{S}_{\mu_1,w} \quad : \quad \bigtimes_{\mu=1}^{d} A_\mu \longrightarrow \mathcal{T}_1 \tag{1.52}$$

$$\hat{v} := (v_1, \ldots, v_d) \quad \mapsto \quad \mathfrak{S}_{\mu_1,w}(\hat{v}) := \left(\bigotimes_{\mu=1}^{\mu_1-1} v_\mu \right) \otimes w \otimes \left(\bigotimes_{\mu=\mu_1+1}^{d} v_\mu \right). \tag{1.53}$$

Ist $v \in \mathcal{T}_1$ und $(v_1, \ldots, v_d)$ ein fest gewähltes Repräsentantensystem von v, dann wird folgende vereinfachte Schreibweise definiert:

$$v^{\mu_1}(w) := \mathfrak{S}_{\mu_1,w}\left((v_1, \ldots, v_d)\right). \tag{1.54}$$

Beispiel 1.4.11. *Seien $n \in \mathbb{N}$ und $\{a, b\} \subset \mathbb{R}^n$ linear unabhängig, $\mathcal{S} := \otimes^d \mathbb{R}^n$ und $i \in \mathbb{N}_{\le d}$. Ferner seien folgende Tensoren definiert:*

$$v := \bigotimes_{\mu=1}^{d} b \tag{1.55}$$

und

$$t := \sum_{i=1}^{d} v^i(a) \in \mathcal{S}_{\le d}. \tag{1.56}$$

Tensoren dieser Struktur treten bei praktischen Anwendungen häufig auf, so ist z. B. der Laplace[5]-Operator in d Dimensionen von solcher Gestalt. In der folgenden Betrachtung soll die Symmetrie von t in den einzelnen Faktoren des Tensorproduktes nicht berücksichtigt werden. Der Speicherbedarf beträgt in der konventionellen Weise

$$d^2 n.$$

Stellt man dagegen t in dem Unterraum $\bigotimes^d \operatorname{span}\{a, b\}$ dar, dann werden bei der alternativen Speicherung $2d^2$ Einträge für den Koeffiziententensor und $2dn$ Einträge für die Basisvektoren benötigt. Daraus resultiert ein Gesamtbedarf von

$$2d(n + d).$$

Man überprüft jetzt leicht, dass

$$2d(n + d) \le d^2 n$$

gilt, wobei $d \ge 3$ und $n \ge 6$ vorausgesetzt sind, d.h. die alternative Darstellung von t bietet in diesem Fall einen Vorteil.

Lemma 1.4.12. *Seien $r \in \mathbb{N}$, $\mu \in \mathbb{N}_{\le d}$, $i \in \mathbb{N}_{\le r}$, $v_{i\mu} \in A_\mu$ und $v_i := \bigotimes_{\mu=1}^{d} v_{i\mu}$. Ferner sei*

$$\sum_{i=1}^{r} v_i = 0. \tag{1.57}$$

(a) Es existiert ein $\nu \in \mathbb{N}_{\le d}$, für das $\{v_i^\nu : i \in \mathbb{N}_{\le r}\}$ linear unabhängig sei. Dann gilt

$$\forall i \in \mathbb{N}_{\le r} : v_{i\nu} = 0. \tag{1.58}$$

(b) Es existiert ein $\nu \in \mathbb{N}_{\le d}$, für das $\{v_{i\nu} : i \in \mathbb{N}_{\le r}\}$ linear unabhängig sei. Dann gilt

$$\forall i \in \mathbb{N}_{\le r} : v_i^\nu = 0. \tag{1.59}$$

[5] Pierre-Simon (Marquis de) Laplace, französischer Mathematiker und Astronom, $*$ 28. März 1749, Beaumont-en-Auge (Normandie), † 5. März 1827 in Paris.

Beweis. (Ähnlich wie für Lemma 1.1.11)
(a) Seien $t_\nu \in \mathbb{N}_{\leq r}$ und $\{e_{l\nu} : l \in \mathbb{N}_{\leq t_\nu}\}$ eine Basis von $\operatorname{span}\{v_{i\nu} : i \in \mathbb{N}_{\leq r}\}$. Dann ist $v_{i\nu} = \sum_{l=1}^{t_\nu} \lambda_{i,l} e_{l\nu}$ für alle $i \in \mathbb{N}_{\leq r}$, mit $\lambda_{i,l} \in \mathbb{R}$. Es gilt

$$\begin{aligned} 0 &= \sum_{i=1}^{r} v_i \\ &= \sum_{i=1}^{r} v_i^\nu(v_{i\nu}) \\ &= \sum_{i=1}^{r} v_i^\nu \left(\sum_{l=1}^{t_\nu} \lambda_{i,l} e_{l\nu} \right) \\ &= \sum_{i=1}^{r} \sum_{l=1}^{t_\nu} \lambda_{i,l} v_i^\nu (e_{l\nu}) . \end{aligned}$$

Gemäß Lemma 1.1.8 ist $\{v_i^\nu(e_{l\nu}) : i \in \mathbb{N}_{\leq r},\ l \in t_\nu\}$ linear unabhängig. Es folgt damit $\lambda_{i,l} = 0$ für alle $i \in \mathbb{N}_{\leq r}$ und $l \in \mathbb{N}_{\leq t_\nu}$, also $v_{i\nu} = 0$ für alle $i \in \mathbb{N}_{\leq r}$.
(b) Der Beweis erfolgt analog zu (a). ■

Korollar 1.4.13. *Seien $r \in \mathbb{N}$, $\mu \in \mathbb{N}_{\leq d}$, $i \in \mathbb{N}_{\leq r}$, $v_{i\mu} \in A_\mu \setminus \{0\}$ und $v_i := \bigotimes_{\mu=1}^{d} v_{i\mu}$. Ferner existiere ein $\nu \in \mathbb{N}_{\leq d}$ mit $\{v_{i\nu} : i \in \mathbb{N}_{\leq r}\}$ linear unabhängig. Dann ist $\{v_i : i \in \mathbb{N}_{\leq r}\}$ linear unabhängig.*

Beweis. Seien $\lambda_i \in \mathbb{R}$ für alle $i \in \mathbb{N}_{\leq r}$ und gelte

$$\sum_{i=1}^{r} \lambda_i v_i = 0.$$

Mit Lemma 1.4.12 folgt dann $\lambda_i = 0$ für alle $i \in \mathbb{N}_{\leq r}$, denn $v_i^\nu \neq 0$. ■

Bemerkung 1.4.14. *Die Rückrichtung von Korollar 1.4.13 gilt im Allgemeinen nicht. Dies folgt aus dem nachstehenden Gegenbeispiel:*

$$V := \{w_1 \otimes w_1,\ w_1 \otimes w_2,\ w_2 \otimes w_1\},$$

wobei $\{w_1, w_2\}$ linear unabhängig ist.

Lemma 1.4.15. *Seien $r \in \mathbb{N}$ und $v \in \mathcal{T}_r$, etwa $v := \sum_{i=1}^{r} v_i$, wobei $v_i := \bigotimes_{\mu=1}^{d} v_{i\mu}$, $v_{i\mu} \in A_\mu \setminus \{0\}$, ist. Dann ist für alle $\mu \in \mathbb{N}_{\leq d}$ $\{v_i^\mu : i \in \mathbb{N}_{\leq r}\}$ linear unabhängig.*

Beweis. Angenommen, es existiert ein $\mu_0 \in \mathbb{N}_{\leq d}$ mit $\{v_i^{\mu_0} : i \in \mathbb{N}_{\leq r}\}$ linear abhängig. Dann existieren $\lambda_1, \ldots, \lambda_r \in \mathbb{R}$ und ein $i_0 \in \mathbb{N}_{\leq r}$ mit $\lambda_{i_0} \neq 0$ und $\sum_{i=1}^{r} \lambda_i v_i^{\mu_0} = 0$. Ohne Beschränkung der Allgemeinheit sei $i_0 = r$. Dann gilt

$$v_r^{\mu_0} = \sum_{i=1}^{r-1} \underbrace{\frac{-\lambda_i}{\lambda_r}}_{\tilde{\lambda}_i :=} v_i^{\mu_0},$$

weiter folgt

$$\begin{aligned} v &= \sum_{i=1}^{r} v_i \\ &= \left(\sum_{i=1}^{r-1} v_i^{\mu_0}(v_{i\mu_0})\right) + v_r^{\mu_0}(v_{r\mu_0}) \\ &= \sum_{i=1}^{r-1} \left(v_i^{\mu_0}(v_{i\mu_0}) + \tilde{\lambda}_i v_i^{\mu_0}(v_{r\mu_0})\right) \\ &= \sum_{i=1}^{r-1} \underbrace{v_i^{\mu_0}(v_{i\mu_0} + \tilde{\lambda}_i v_{r\mu_0})}_{\tilde{v}_i :=} \\ &= \sum_{i=1}^{r-1} \tilde{v}_i, \end{aligned}$$

im Widerspruch zu $\text{rang}_{\mathcal{T}}(v) = r$. ■

2 Analyse von besten Approximationen mit Elementartensor-Summen

2.1 Elemente der Optimierungstheorie

Der nachstehende Abschnitt bereitet grundlegende Definitionen und Eigenschaften aus der Optimierungstheorie auf, wobei nur wesentliche, für die Arbeit relevante Punkte beschrieben sind.

2.1.1 Minimallösung und Tangentialkegel

Im Folgenden seien $n \in \mathbb{N}$, $\emptyset \neq K \subseteq \mathbb{R}^n$ und $f : \mathbb{R}^n \to \mathbb{R}$ eine Funktion.
Bei einem restringierten Minimierungsproblem sucht man ein $x^* \in K$ mit

$$\forall x \in K : f(x^*) \leq f(x). \tag{2.1}$$

Ist die Funktion f nicht konvex, so kann im Allgemeinen nicht entschieden werden, ob x^* ein globales Minimum von f auf K ist, wenigstens nicht durch Berechnung von endlich vielen Funktions- bzw. Ableitungswerten, siehe [37, Abschnitt 3.4, Seiten 65ff.]. In diesem Fall sollte die Aufgabenstellung (2.1) abgeschwächt werden, so dass $f(x^*)$ nur mit Funktionswerten in einer Umgebung von x^* verglichen wird. Dies führt in natürlicher Weise auf den Begriff des lokalen Minimums, d.h., es existiert eine Umgebung $U \subseteq K$ von x^* mit

$$\forall x \in K \cap U : f(x^*) \leq f(x). \tag{2.2}$$

Definition 2.1.1 (Minimallösung, Minimalwert und Niveaumenge)**.** *Die* Menge der Minimallösungen *von f auf K wird mit*

$$\mathfrak{M}(f, K) := \{x \in K : f(x) = \inf f(K)\} \tag{2.3}$$

bezeichnet.
Die Zahl $\inf f(K) \in [-\infty, \infty)$ *heißt* Minimalwert der Minimierungsaufgabe (f, K).
Für ein $r \in \mathbb{R}$ bezeichne

$$S_f(r) := \{x \in K : f(x) \leq r\} \tag{2.4}$$

die zugehörige Niveaumenge von f. *Für ein* $x_0 \in K$ *benutzt man folgende abkürzende Schreibweise:*

$$S_f(x_0) := S_f(f(x_0)). \tag{2.5}$$

Definition 2.1.2 (Tangentialkegel)**.** *Ein Vektor* $d \in \mathbb{R}^n$ *heißt* tangential *zu* K *im Punkte* $x \in K$*, wenn Folgen* $(x_k)_{k\in\mathbb{N}} \subseteq K$ *und* $(t_k)_{k\in\mathbb{N}} \subseteq \mathbb{R}$ *existieren mit:*

- $x_k \xrightarrow[k\to\infty]{} x$,
- $t_k \searrow 0$ *für* $k \to \infty$,
- $\frac{x_k - x}{t_k} \xrightarrow[k\to\infty]{} d$.

Die Menge aller dieser Richtungen heißt Tangentialkegel *von* K *in* $x \in K$ *und wird mit* $\mathfrak{T}_{\mathfrak{K}}(x)$ *bezeichnet,*

$$\mathfrak{T}_{\mathfrak{K}}(x) := \left\{ d \in \mathbb{R}^n : \ \exists (x_k)_{k\in\mathbb{N}} \subseteq K : \exists t_k \searrow 0 : x_k \to x \wedge \frac{x_k - x}{t_k} \to d \right\}. \tag{2.6}$$

Man verifiziert leicht, dass die Menge $\mathfrak{T}_{\mathfrak{K}}(x)$ stets ein Kegel ist, womit insbesondere die Namensgebung gerechtfertigt ist.

Lemma 2.1.3. *Seien* K *ein Kegel und* $x \in K$*. Dann ist* $-x \in \mathfrak{T}_{\mathfrak{K}}(x)$.

Beweis. Setze $t_k := \frac{1}{k+1}$ und $x_k := (1 - t_k)\, x$ für alle $k \in \mathbb{N}$. Dann ist $(x_k)_{k\in\mathbb{N}} \subseteq K$, denn K ist ein Kegel. Ferner gilt offenbar $t_k \searrow 0$, $x_k \to x$ und $\frac{x_k - x}{t_k} = -x$ für $k \to \infty$. ■

Lemma 2.1.4. *Sei* $x \in K$*. Dann ist der Tangentialkegel* $\mathfrak{T}_{\mathfrak{K}}(x)$ *abgeschlossen.*

Beweis. [9, Lemma 2.29, Seite 42]. ■

Lemma 2.1.5. *Seien* $x^* \in K$ *ein lokales Minimum des restringierten Minimierungsproblems (2.1) und* f *stetig differenzierbar. Dann gilt für die Richtungsableitung im Punkte* x^*

$$\forall d \in \mathfrak{T}_{\mathfrak{K}}(x^*) : f'(x^*, d) \geq 0. \tag{2.7}$$

Beweis. [9, Lemma 2.30, Seiten 42 f.]. ■

2.1.2 Beste Approximation

Im Folgenden seien $(X, \langle\cdot,\cdot\rangle)$ ein reeller Hilbertraum und $\emptyset \neq K \subseteq X$. Ferner bezeichne $\|\cdot\|$ die durch das Skalarprodukt $\langle\cdot,\cdot\rangle$ induzierte Norm.

Definition 2.1.6 (Beste Approximation)**.** *Sei* $a \in X$*. Ein Element* $x^* \in K$ *heißt* beste Approximation *von* a *bezüglich* K*, wenn für alle* $x \in K$

$$\|a - x^*\| \leq \|a - x\| \tag{2.8}$$

gilt.

Satz 2.1.7 (Approximationssatz)**.** *Seien K abgeschlossen und konvex und ferner sei $a \in X$. Dann existiert genau eine beste Approximation von a bezüglich K.*

Beweis. [44, Satz V.3.2, Seite 219]. ■

Satz 2.1.8 (Existenz im endlich dimensionalen Fall)**.** *Sei X endlich dimensional und K abgeschlossen. Dann existiert eine beste Approximation von a bezüglich K.*

Beweis. Sei $a \in X$ und ein $k \in K \neq \emptyset$ gewählt. Die abgeschlossene Kugel $\overline{B}(a, \|a-k\|) := \{y \in X : \|a-y\| \leq \|a-k\|\}$ ist kompakt, da X endlich dimensional ist. Daneben ist auch $K \cap \overline{B}(a, \|a-k\|)$ kompakt und nicht leer, denn K ist nach Voraussetzung insbesondere abgeschlossen. Nach dem Satz von Weierstraß[1] existiert nun eine beste Approximation von a bezüglich K, denn die Norm ist eine stetige Abbildung. ■

Lemma 2.1.9. *Seien $a \in X$ und*

$$\begin{aligned} \tilde{f} : X &\to \mathbb{R} \\ x &\mapsto \tilde{f}(x) := \frac{1}{2}\|a-x\|^2. \end{aligned}$$

Dann ist $\tilde{f}$ Gâteaux[2]-differenzierbar in jedem Punkt $x \in X$. Für die Richtungsableitung in x in Richtung d gilt

$$\tilde{f}'(x^*, d) = -\langle a-x, d\rangle . \tag{2.9}$$

Beweis. Seien $x, d \in X$. Für alle $\alpha \in \mathbb{R} \setminus \{0\}$ gilt

$$\begin{aligned} \frac{\tilde{f}(x+\alpha d) - \tilde{f}(x)}{\alpha} &= \frac{\tilde{f}(x) - \alpha\langle a-x, d\rangle + \frac{\alpha^2}{2}\langle d, d\rangle - \tilde{f}(x)}{\alpha} \\ &= -\langle a-x, d\rangle + \frac{\alpha}{2}\langle d, d\rangle \end{aligned}$$

$$\Rightarrow \quad \frac{\tilde{f}(x+\alpha d) - \tilde{f}(x)}{\alpha} \xrightarrow[\alpha\to 0]{} -\langle a-x, d\rangle .$$

■

Lemma 2.1.10. *Seien $n \in \mathbb{N}$, $\emptyset \neq K \subset \mathbb{R}^n$ ein Kegel, $a \in \mathbb{R}^n$ und $x^* \in K$ lokale beste Approximation von a bezüglich K, d.h., x^* ist ein lokales Minimum der Funktion $\tilde{f}(x) := \frac{1}{2}\|a-x\|^2$ in K. Dann gilt*

$$\|a - x^*\| \leq \|a\|. \tag{2.10}$$

[1] Karl Theodor Wilhelm Weierstraß, deutscher Mathematiker, $*$ 31. Oktober 1815 in Ostenfelde, † 19. Februar 1897 in Berlin.

[2] René Eugène Gâteaux, französischer Mathematiker, $*$ 1889, gefallen 3. Oktober 1914.

Beweis. Wegen Lemma 2.1.3 ist $-x^* \in \mathfrak{T}_{\mathfrak{K}}(x^*)$. Mit Lemma 2.1.5 und Gleichung (2.9) folgt dann

$$0 \leq \langle a - x^*, x^* \rangle .$$

Ferner gilt

$$\begin{aligned} \Rightarrow \quad & \frac{1}{2} \langle x^*, x^* \rangle \leq \langle x^*, x^* \rangle \leq \langle a, x^* \rangle \\ \Rightarrow \quad & \frac{1}{2}\|a\|^2 - \langle a, x^* \rangle + \frac{1}{2}\|x^*\|^2 \leq \frac{1}{2}\|a\|^2 \\ \Rightarrow \quad & \|a - x^*\| \leq \|a\|. \end{aligned}$$

■

2.2 Kennzeichen von besten Approximationen mit Elementartensor-Summen

In diesem Abschnitt seien $d \in \mathbb{N}_{\geq 2}$, $(A_1, \langle \cdot, \cdot \rangle_1), \ldots, (A_d, \langle \cdot, \cdot \rangle_d)$ reelle Prähilberträume und $(t, \mathcal{T})$ ein Tensorprodukt von $A_1, \ldots, A_d$. Daneben bezeichne $\langle \cdot, \cdot \rangle_{\mathcal{T}}$ das durch Lemma 1.3.2 induzierte Skalarprodukt von $\mathcal{T}$ und ferner sei $\| \cdot \|_{\mathcal{T}} := \sqrt{\langle \cdot, \cdot \rangle_{\mathcal{T}}}$ gesetzt. Ist aus dem Zusammenhang klar, welches Skalarprodukt gemeint ist, so wird auf eine genaue Kennzeichnung verzichtet.

Lemma 2.2.1. *Seien $\mu \in \mathbb{N}_{\leq d}$, $U_\mu \leq A_\mu$ ein Untervektorraum von A_μ, $P_\mu : A_\mu \to U_\mu$ die orthogonale Projektion von A_μ auf U_μ und ferner sei $\mathcal{U} := \bigotimes_{\mu=1}^{d} U_\mu$ gesetzt. Dann ist $P := \bigotimes_{\mu=1}^{d} P_\mu : \mathcal{T} \to \mathcal{U}$ die orthogonale Projektion von $\mathcal{T}$ auf $\mathcal{U}$.*

Beweis. Gemäß Bemerkung 1.2.3 gilt

$$\begin{aligned} P^2 &= \left(\bigotimes_{\mu=1}^{d} P_\mu \right) \circ \left(\bigotimes_{\mu=1}^{d} P_\mu \right) \\ &= \bigotimes_{\mu=1}^{d} P_\mu^2 = \bigotimes_{\mu=1}^{d} P_\mu \\ &= P. \end{aligned}$$

Seien $v, w \in \mathcal{T}$, etwa $v = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu}$ und $w = \sum_{j=1}^{s} \bigotimes_{\mu=1}^{d} w_{j\mu}$, wobei

$r, s \in \mathbb{N}$ sind. Es folgt

$$\begin{aligned}\langle Pv, w\rangle &= \sum_{i=1}^{r}\sum_{j=1}^{s}\prod_{\mu=1}^{d}\langle P_\mu v_{i\mu}, w_{j\mu}\rangle \\ &= \sum_{i=1}^{r}\sum_{j=1}^{s}\prod_{\mu=1}^{d}\langle v_{i\mu}, P_\mu w_{j\mu}\rangle \\ &= \langle v, Pw\rangle\,, \\ \Rightarrow P &= P^t.\end{aligned}$$

■

Bemerkung 2.2.2. *Insbesondere folgt aus Lemma 2.2.1, dass der Tensorrang der orthogonalen Projektion von $\mathcal{T}$ auf $\mathcal{U}$ gleich eins ist.*

Lemma 2.2.3. *Seien $r \in \mathbb{N}$, $\mu \in \mathbb{N}_{\le d}$, $v \in \mathcal{T}$, etwa $v = \sum_{i=1}^{r}\bigotimes_{\mu=1}^{d} v_{i\mu}$, $U_\mu \le A_\mu$ ein Untervektorraum von A_μ und gelte $\{v_{i\mu} : i \in \mathbb{N}_{\le r}\} \subset U_\mu$. Daneben sei $P_\mu : A_\mu \to U_\mu$ die orthogonale Projektion von A_μ auf U_μ und $P := \bigotimes_{\mu=1}^{d} P_\mu$. Dann gilt für alle $x \in \mathcal{T}$*

$$\|v - Px\| \le \|v - x\|. \tag{2.11}$$

Ist zusätzlich $x \in \mathcal{T} \setminus \mathcal{U}$, dann gilt

$$\|v - Px\| < \|v - x\|. \tag{2.12}$$

Beweis. Es gilt $v = Pv$, denn $v \in \bigotimes_{\mu=1}^{d} U_\mu$ ist vorausgesetzt. Sei ferner $x \in \mathcal{T}$. Es folgt

$$\begin{aligned}\|v - x\|^2 &= \|v - Px\|^2 - 2\langle v - Px, x - Px\rangle + \|x - Px\|^2 \\ &= \|v - Px\|^2 - 2\langle v - x, Px - P^2x\rangle + \|x - Px\|^2 \\ &= \|v - Px\|^2 + \underbrace{\|x - Px\|^2}_{\ge 0}, \\ \Rightarrow \|v - Px\| &\le \|v - x\|.\end{aligned}$$

Ist $x \notin \mathcal{U}$, dann ist $\|x - Px\| > 0$ und damit $\|v - Px\| < \|v - x\|$. ■

Der nachstehende Satz 2.2.4 analysiert das Repräsentantensystem von besten Approximationen mit Summen von Elementartensoren und ist für diese Arbeit von zentraler Bedeutung.
Sei im Folgenden

$$v = \sum_{i=1}^{R}\bigotimes_{\mu=1}^{d} v_{i\mu}$$

eine Summe von Elementartensoren mit Tensorrang R und

$$x^* = \sum_{i=1}^{r}\bigotimes_{\mu=1}^{d} x^*_{i\mu}$$

eine beste Approximation von v mit Tensorrang $r < R$. Satz 2.2.4 besagt, dass für jedes $\mu \in \mathbb{N}_{\leq d}$ die Vektoren $x^*_{j\mu}$ in $\operatorname{span}\{v_{i\mu} : i \in \mathbb{N}_{\leq R}\}$ aufzufinden sind, wobei $j \leq r$ beliebig ist. Die konsequente Anwendung dieses Resultats hat weitreichende Konsequenzen, die in den darauf aufbauenden Ergebnissen beschrieben werden.

Satz 2.2.4. *Seien $R \in \mathbb{N}$ und $r \in \mathbb{N}_{<R}$. Ferner seien $v \in \mathcal{T}_R$, etwa $v = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} v_{i\mu}$, $U_\mu := \operatorname{span}\{v_{i\mu} : i \in \mathbb{N}_{\leq R}\}$ für $\mu \in \mathbb{N}_{\leq d}$ und*

$$\begin{aligned} f : \mathcal{T}_{\leq r} &\to \mathbb{R}_{\geq 0} \\ x &\mapsto f(x) := \|v - x\|. \end{aligned}$$

Darüber hinaus sei $x^* \in \mathfrak{M}(f, \mathcal{T}_{\leq r})$, etwa $x^* = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} x^*_{j\mu}$, wobei $\mathfrak{M}(f, \mathcal{T}_{\leq r})$ die Menge der Minimallösungen von f auf $\mathcal{T}_{\leq r}$ ist, siehe Definition 2.1.1. Dann gilt

$$\forall \mu \in \mathbb{N}_{\leq d} \forall j \in \mathbb{N}_{\leq r} : x^*_{j\mu} \in U_\mu, \tag{2.13}$$

d.h. $x^* \in \mathcal{U} := \bigotimes_{\mu=1}^{d} U_\mu$.

Beweis. Seien $\mu \in \mathbb{N}_{\leq d}$, $P_\mu : A_\mu \to U_\mu$ die orthogonale Projektion von A_μ auf U_μ und $P := \bigotimes_{\mu=1}^{d} P_\mu : \mathcal{T} \to \mathcal{U}$. Angenommen, es existierten $\mu' \in \mathbb{N}_{\leq d}$ und $j \in \mathbb{N}_{\leq r}$ mit $x^*_{j\mu'} \notin U_{\mu'}$, d.h. $x^* \notin \mathcal{U}$. Dann ist

$$\begin{aligned} \hat{x} := Px^* &= \left(\bigotimes_{\mu=1}^{d} P_\mu \right) \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} x^*_{j\mu} \\ &= \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \underbrace{P_\mu x^*_{j\mu}}_{\in U_\mu} \in \mathcal{U}_{\leq r}. \end{aligned}$$

Mit Lemma 2.2.3 folgt dann

$$\|v - \hat{x}\| < \|v - x^*\|.$$

Dies steht aber im Widerspruch zu $x^* \in \mathfrak{M}(f, \mathcal{T}_{\leq r})$. ■

Korollar 2.2.5. *Es gelten die Bezeichnungen und Voraussetzungen von Satz 2.2.4. Ferner sei U'_μ ein weiterer Untervektorraum von A_μ, wobei $U_\mu \leq U'_\mu$ für $\mu \in \mathbb{N}_{\leq d}$ erfüllt sei. Dann gilt*

$$\forall \mu \in \mathbb{N}_{\leq d} \forall j \in \mathbb{N}_{\leq r} : x^*_{j\mu} \in U'_\mu. \tag{2.14}$$

Beispiel 2.2.6. *Seien $U \leq \mathbb{R}^{n \times n}$ der Vektorraum der unteren Dreiecksmatrizen und $O \leq \mathbb{R}^{n \times n}$ der Vektorraum der oberen Dreiecksmatrizen, wobei $n \in \mathbb{N}_{\geq 2}$ ist. Ferner sei $\mathcal{M} := U \otimes O$. Dann ist gemäß Korollar 2.2.5 jede beste Niedrigtensorrang-Approximation $X^* \in \mathcal{M}_{\leq r}$ von $V \in \mathcal{M}_R$ der Gestalt, dass jeder Summand von X^* in der ersten Komponente nur aus einer unteren Dreiecksmatrix und in der zweiten Komponente nur aus einer oberen Dreiecksmatrix besteht.*

Bemerkung 2.2.7. *Es gelten die Bezeichnungen und Voraussetzungen von Satz 2.2.4. Ferner seien $i \in \mathbb{N}_{\leq R}$, $j \in \mathbb{N}_{\leq r}$ und $\{z_{l\mu} \in U_\mu : l \in \mathbb{N}_{\leq t_\mu}\}$ eine orthonormale Basis von U_μ für alle $\mu \in \mathbb{N}_{\leq d}$, wobei $t_\mu = \dim U_\mu \in \mathbb{N}$ ist. Wegen Satz 2.2.4 kann man die Suche nach einer besten Approximation von v in $\mathcal{T}_{\leq r}$ auf $\mathcal{U}_{\leq r}$ einschränken. Sei $x \in \mathcal{U}_{\leq r}$, etwa $x = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} x_{j\mu}$, es existieren dann $\alpha_{i\mu}, \xi_{j\mu} \in \mathbb{R}^{t_\mu}$ mit*

$$\begin{aligned} v_{i\mu} &= \sum_{l_\mu=1}^{t_\mu} (\alpha_{i\mu})_{l_\mu} z_{l_\mu \mu}, \\ x_{j\mu} &= \sum_{l_\mu=1}^{t_\mu} (\xi_{j\mu})_{l_\mu} z_{l_\mu \mu}. \end{aligned}$$

Diese Gleichungen induzieren eine lineare Abbildung $Z_\mu : \mathbb{R}^{t_\mu} \to U_\mu$ mit

$$\begin{aligned} v_{i\mu} &= Z_\mu \alpha_{i\mu}, \\ x_{j\mu} &= Z_\mu \xi_{j\mu}. \end{aligned}$$

Setzt man $\mathcal{S} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}$, $\alpha := \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} \alpha_{i\mu} \in \mathcal{S}_{\leq R}$, $\xi := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{S}_{\leq r}$ und

$$\begin{aligned} Z &: \mathcal{S}_{\leq r} \to \mathcal{U}_{\leq r} \\ Z &:= \bigotimes_{\mu=1}^{d} Z_\mu, \end{aligned}$$

dann gilt

$$\begin{aligned} v &= \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} v_{i\mu} = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} Z_\mu \alpha_{i\mu} = \sum_{i=1}^{R} \left(\bigotimes_{\mu=1}^{d} Z_\mu \right) \left(\bigotimes_{\mu=1}^{d} \alpha_{i\mu} \right) = Z\alpha, \\ x &= \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} x_{j\mu} = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} Z_\mu \xi_{j\mu} = \sum_{j=1}^{r} \left(\bigotimes_{\mu=1}^{d} Z_\mu \right) \left(\bigotimes_{\mu=1}^{d} \xi_{j\mu} \right) = Z\xi. \end{aligned}$$

Weiter folgt dann

$$\|v - x\|_{\mathcal{T}}^2 = \langle Z(\alpha - \xi), Z(\alpha - \xi) \rangle_{\mathcal{T}} = \left\langle \alpha - \xi, Z^t Z(\alpha - \xi) \right\rangle_{\mathcal{S}} = \|\alpha - \xi\|_{\mathcal{S}}^2.$$

Korollar 2.2.8. *Es gelten die Bezeichnungen und Voraussetzungen von Bemerkung 2.2.7. Ferner seien $\xi^* \in \mathcal{S}_{\leq r}$ mit $x^* := Z\xi^* \in \mathcal{T}_{\leq r}$ und $\tilde{f} : \mathcal{S}_{\leq r} \to \mathbb{R}_{\geq 0}$, $\xi \mapsto \tilde{f}(\xi) := \|\alpha - \xi\|_{\mathcal{S}}$. Dann gilt*

$$x^* \in \mathfrak{M}(f, \mathcal{T}_{\leq r}) \Leftrightarrow \xi^* \in \mathfrak{M}(\hat{f}, \mathcal{S}_{\leq r}). \tag{2.15}$$

Aus diesem Grund genügt es, die Approximation von Elementartensor-Summen auf $\mathcal{S}_{\leq r}$ zu betrachten; dabei wird aus Gründen der praktischen Umsetzung vorausgesetzt, dass die Berechnung des Koeffiziententensors und einer orthonormalen Basis von U_μ einfach ist.
Betrachtet man das Tensorprodukt über die reellen Vektorräume $\mathbb{R}^{n_\mu}$, wobei $n_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}_{\leq d}$, dann ist es in vielen Fällen vorteilhaft, eine Unterraumdarstellung zu bestimmen, wie im Beispiel 1.4.11 oder, falls der Tensorrang viel kleiner ist als $\max\{n_1, \ldots, n_d\}$.

Bemerkung 2.2.9 (Berechnung einer Orthonormalbasis im reellen Standardfall)**.** *Seien $R \in \mathbb{N}$, $\mu \in \mathbb{N}_{\leq d}$, $\mathcal{R} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{n_\mu}$ sowie $v := \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{R}_R$ und $n_\mu \in \mathbb{N}$. Ferner seien*

$$V_\mu := (v_{1\mu}, v_{2\mu}, \ldots, v_{R\mu}) \in \mathbb{R}^{n_\mu \times R}$$

und $t_\mu := \mathrm{Rang}(V_\mu) \in \mathbb{N}_{\leq R} \ll n_\mu$.
Durch QR-Zerlegung mit Spaltenpivotisierung, siehe [10, 5.4.1, Seiten 248ff.], berechnet man nun

$$Q_\mu^t V_\mu P_\mu = \begin{bmatrix} R_1^{(\mu)} & R_2^{(\mu)} \\ 0 & 0 \end{bmatrix},$$

wobei $Q_\mu := (q_{1\mu}, \ldots, q_{n_\mu \mu}) \in \mathbb{R}^{n_\mu \times n_\mu}$ eine orthogonale Matrix, $P_\mu \in \mathbb{R}^{R \times R}$ eine Permutationsmatrix, $R_1^{(\mu)} \in \mathbb{R}^{t_\mu \times t_\mu}$ eine reguläre obere Dreiecksmatrix und $R_2^{(\mu)} \in \mathbb{R}^{t_\mu \times (R - t_\mu)}$ sind. Diese Zerlegung impliziert

$$\forall i \in \mathbb{N}_{\leq R} : v_{i\mu} \in \mathrm{span}\{q_{1\mu}, \ldots, q_{t_\mu \mu}\} =: U_\mu.$$

Der Rechenaufwand beträgt dann für eine Faktorisierung, gemäß [10, 5.4.1, Seite 250],

$$4 n_\mu t_\mu R - 2 t_\mu^2 (n_\mu + R) + \frac{4}{3} t_\mu^3,$$

so dass insgesamt

$$\sum_{\mu=1}^{d} 4 n_\mu t_\mu R - 2 t_\mu^2 (n_\mu + R) + \frac{4}{3} t_\mu^3$$

Rechenoperationen benötigt werden. Wie man sieht, wächst die Komplexität nur linear mit d.
Benutzt man die in [10, 5.4.1, Seiten 248 ff.] vorgestellte Methode mit realen Daten, dann wird man nach t_μ Schritten eine Zerlegung der Form

$$Q_\mu^t V_\mu P_\mu = \begin{bmatrix} R_1^{(\mu)} & R_2^{(\mu)} \\ 0 & R_3^{(\mu)} \end{bmatrix}$$

erhalten, wobei $R_3^{(\mu)} \in \mathbb{R}^{(n_\mu - t_\mu) \times (R - t_\mu)} \setminus \{0\}$ ist, siehe [10, 5.5.6, Seiten 258 ff.], denn in Gegenwart von Rundungsfehlern sind die Einträge der Matrix $R_3^{(\mu)}$ nicht exakt null. Erfüllt $R_3^{(\mu)}$ aber folgende Ungleichung:

$$\|R_3^{(\mu)}\|_2 \leq \epsilon \|V_\mu\|_2,$$

dann kann man t_μ als numerischen Rang von V_μ ansehen, hierbei ist $\epsilon \in \mathbb{R}_{>0}$ eine kleine maschinenabhängige Konstante. Darüber hinaus muss man noch unerwünschte Skalierungseffekte vermeiden, daher werden alle Spaltenvektoren von V_μ zuvor normiert.

2.3 Summen von Elementartensoren mit beschränkten Summanden

Im Folgenden seien $d \in \mathbb{N}_{\geq 2}$, $S := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}$ und $t_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}_{\leq d}$. Nach dem Approximationssatz ist die Existenz der besten Approximation nur auf abgeschlossenen Mengen gesichert. Leider ist die Menge der Elementartensor-Summen im Allgemeinen nicht abgeschlossen.

Lemma 2.3.1. *Für $r \geq 2$ und $d \geq 3$ ist $\mathcal{S}'_{\leq r}$ nicht abgeschlossen, wobei $\mathcal{S}' := \otimes^d \mathbb{R}^n$ und $n \in \mathbb{N}$.*

Beweis. Sei $\{a, b\} \subset \mathbb{R}^n$ linear unabhängig. Gemäß [38, Proposition 4.6, Seiten 15 f.] hat

$$t := a \otimes b \otimes b + b \otimes a \otimes b + b \otimes b \otimes a$$

einen Tensorrang von 3. Sei $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}'_{\leq 2}$ mit

$$t^k := (\frac{1}{k}a + b) \otimes (b + \frac{1}{k}a) \otimes kb + b \otimes b \otimes (a - kb)$$

für alle $k \in \mathbb{N}$. Mittels elementarer Umformung folgt

$$\begin{aligned} t^k &= a \otimes b \otimes b + b \otimes a \otimes b + b \otimes b \otimes a + \frac{1}{k} a \otimes a \otimes b \\ &= t + \frac{1}{k} a \otimes a \otimes b. \\ \Rightarrow \quad & \|t - t^k\| = \|a \otimes a \otimes b\| \frac{1}{k} \xrightarrow[k\to\infty]{} 0. \end{aligned}$$

■

Bemerkung 2.3.2. *Das in Lemma 2.3.1 angegebene Gegenbeispiel ist von besonderer Bedeutung und darf nicht ignoriert werden, denn der Laplace-Operator in 3 Dimensionen ist von dieser Gestalt.*
Der Artikel von de Silva und Lim, siehe [38], analysiert allgemeinere Klassen von Gegenbeispielen für Tensoren der Ordnung 3. Es existieren sogar Gegenbeispiele, welche mehrere Tensorränge überspringen können.

Eine einfache Analyse des Gegenbeispiels aus Lemma 2.3.1 zeigt, dass für $k \to \infty$ die Summanden von t^k unbeschränkt sind, d.h. es gilt

$$\begin{aligned} \left\| (\frac{1}{k}a + b) \otimes (b + \frac{1}{k}a) \otimes kb \right\| &\xrightarrow[k\to\infty]{} \infty, \\ \|b \otimes b \otimes (a - kb)\| &\xrightarrow[k\to\infty]{} \infty. \end{aligned}$$

Zusammenfassend kann man sagen, dass die Existenz der besten Approximation nicht gesichert und die numerische Behandlung dieses Minimierungsproblems wegen der Unbeschränktheit der Summanden nicht durchführbar ist. Einen nahe liegenden Ansatz zur Behebung dieses Problems liefert folgende Definition.

Definition 2.3.3 (Elementartensor-Summen mit beschränkten Summanden)**.** *Seien $c \in \mathbb{R}_{>0}$ und $r \in \mathbb{N}$. Die Menge der* Elementartensor-Summen mit beschränkten Summanden *ist per Definition Teilmenge von $\mathcal{S}_r$ bzw. $\mathcal{S}_{\leq r}$ und wie folgt definiert:*

$$\mathcal{S}_r^c := \left\{ v = \sum_{i=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} v_{i\mu}}_{v_i :=} \in \mathcal{S}_r : \|v_i\| \leq c \right\} \tag{2.16}$$

bzw.

$$\mathcal{S}_{\leq r}^c := \left\{ v = \sum_{i=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} v_{i\mu}}_{v_i :=} \in \mathcal{S}_{\leq r} : \|v_i\| \leq c \right\}. \tag{2.17}$$

Lemma 2.3.4. *Seien $c \in \mathbb{R}_{>0}$ und $r \in \mathbb{N}$. Dann ist $S_{\leq r}^c$ abgeschlossen.*

Beweis. Seien $c \in \mathbb{R}_{>0}$ und $r \in \mathbb{N}$. Ferner sei $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}_{\leq r}^c$ und $t \in \mathcal{S}$ mit

$$\lim_{k\to\infty} t^k = t,$$

wobei $t^k := \sum_{i=1}^{r} \otimes_{\mu=1}^{d} t_{i\mu}^k$ für alle $k \in \mathbb{N}$ ist. Seien $i \in \mathbb{N}_{\leq r}$ und $\mu \in \mathbb{N}_{\leq d}$. Wegen der Multilinearität von $\otimes$ kann man ohne Beschränkung der Allgemeinheit annehmen, dass $\|t_{i1}^k\| = \cdots = \|t_{id}^k\|$ erfüllt ist, daneben ist $(t_{i\mu}^k)_{k\in\mathbb{N}}$ eine durch $\sqrt[d]{c}$ beschränkte Folge. Daher existiert eine gegen $\tilde{t}_{i\mu} \in \mathbb{R}^{t_\mu}$ konvergente Teilfolge $(t_{i\mu}^{k(l)})_{l\in\mathbb{N}}$, wobei $\|\tilde{t}_{i\mu}\| \leq \sqrt[d]{c}$ erfüllt ist. Setzt man nun $\tilde{t} := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \tilde{t}_{i\mu} \in \mathcal{S}_{\leq r}^c$, dann folgt $t^{k(l)} \xrightarrow[l\to\infty]{} \tilde{t}$ und damit $\tilde{t} = t$, denn die folgende Abbildung

$$\begin{aligned} \mathfrak{E} : \bigtimes_{\mu=1}^{d} \bigtimes_{i=1}^{r} \mathbb{R}^{t_\mu} &\to \mathcal{S}_{\leq r} \\ \hat{t} := (t_{i\mu} : i \in \mathbb{N}_{\leq r},\ \mu \in \mathbb{N}_{\leq d}) &\mapsto \mathfrak{E}(\hat{t}) := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} t_{i\mu} \end{aligned}$$

ist insbesondere stetig. ∎

Korollar 2.3.5. *Seien $r' \in \mathbb{N}$, $r \in \mathbb{N}_{<r'}$, $t \in \mathcal{S}_{r'}$ und $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}_{\leq r}$ mit*

$$\lim_{k\to\infty} t^k = t,$$

wobei $t^k := \sum_{i=1}^{r} t_i^k$, $t_i^k \in \mathcal{S}_1$, für alle $k \in \mathbb{N}$ gesetzt sei. Dann ist für mindestens ein $i \in \mathbb{N}_{\leq r}$ die Folge der zugehörigen Summanden $(t_i^k)_{k\in\mathbb{N}}$ nicht beschränkt.

Beweis. Angenommen $(t_i^k)_{k\in\mathbb{N}}$ ist für alle $i \in \mathbb{N}_{\leq r}$ durch $c \in \mathbb{R}_{>0}$ beschränkt. Gemäß Lemma 2.3.4 ist dann $t \in \mathcal{S}^c_{\leq r} \subset \mathcal{S}_{\leq r}$. Widerspruch, denn $\operatorname{rang}_{\mathcal{S}}(t) = r' > r$. ■

Korollar 2.3.6. *$\mathcal{S}_{\leq 1}$ ist abgeschlossen.*

Beweis. Sei $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}_{\leq 1}$ mit

$$\lim_{k\to\infty} t^k = t \in \mathcal{S}.$$

Dann ist für alle $k \in \mathbb{N}$ der Summand von t^k durch $\|t\|$ beschränkt. Der Rest des Beweises folgt analog zum Beweis von Lemma 2.3.4. ■

3 Approximationsaufgabe und Zielfunktion

In diesem Kapitel werden die beiden bedeutsamen Approximationsaufgaben formuliert und anschließend die Zielfunktion definiert. Des Weiteren sind die für die numerische Optimierung wichtigen ersten und zweiten Ableitungen der Zielfunktion angegeben.
Im Folgenden seien $d \in \mathbb{N}_{\geq 3}$, $c \in \mathbb{R}_{\geq 0}$, $\mathcal{S} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}$ und $t_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}_{\leq d}$.

3.1 Formulierung der Approximationsaufgabe

In Kapitel 5 wird die numerische Lösung der nachstehenden Approximationsaufgaben diskutiert; um in diesem folgenden Teil der Arbeit stetig wiederholende Begriffserklärungen zu vermeiden, werden die anstehenden Definitionen auch als Notationen verwendet.

Definition 3.1.1 (Approximationsaufgabe)**.**
Im Folgenden seien $R \in \mathbb{N}$,

$$\alpha := \sum_{i=1}^{R} \alpha_i = \sum_{i=1}^{R} \underbrace{\bigotimes_{\mu=1}^{d} \alpha_{i\mu}}_{\alpha_i :=} \in \mathcal{S}_R \tag{3.1}$$

gegeben und zunächst $r \in \mathbb{N}_{<R}$ fest gewählt. Gesucht wird ein

$$\xi^* := \sum_{i=1}^{r} \xi_i^* = \sum_{i=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} \xi_{i\mu}^*}_{\xi_i^* :=} \in \mathcal{S}_{\leq r}, \tag{3.2}$$

welches die Approximationsaufgabe

$$\|\alpha - \xi^*\| = \min_{\xi \in \mathcal{S}_{\leq r}^c \cap U(\xi^*)} \|\alpha - \xi\| \tag{3.3}$$

löst, wobei $U(\xi^*)$ eine Umgebung von ξ^* ist, unter der Nebenbedingung:

$$\forall j \in \mathbb{N}_{\leq r} \forall \mu \in \mathbb{N}_{\leq d} \forall \nu \in \mathbb{N}_{\leq d} \setminus \{\mu\} : \|\xi_{j\mu}^*\| = \|\xi_{j\nu}^*\|. \tag{3.4}$$

Wie in Abschnitt 2.1.1 bereits erwähnt, ist es für nichtkonvexe Zielfunktionen schwierig, ein globales Minimum zu bestimmen; daher wird die Suche nach einem Minimum auf eine Umgebung $U(\xi^*)$ von ξ^* eingeschränkt.
In den künftigen Anwendungen ist a priori nicht klar, wie groß der Zielrang gewählt werden muss, vielmehr ist eine gewünschte Approximationsgenauigkeit vorgegeben, welche sich bei kleinstmöglichem Tensorrang einstellen soll. Aus diesem Grund wird folgende erweiterte Approximationsaufgabe gestellt.

Definition 3.1.2 (Erweiterte Approximationsaufgabe)**.**
Seien $\alpha \in \mathcal{S}_R$, $R \in \mathbb{N}$ und $\varepsilon \in \mathbb{R}_{\geq 0}$ gegeben. Zu bestimmen ist ein $\xi_\varepsilon \in \mathcal{S}_{\leq r_\varepsilon}$ derart, dass

$$\begin{aligned} \|\alpha - \xi_\varepsilon\| &\leq \varepsilon, && (3.5)\\ \|\alpha - \xi_\varepsilon\| &= \min_{\xi \in \mathcal{S}^c_{\leq r_\varepsilon} \cap U(\xi_\varepsilon)} \|\alpha - \xi\|, && (3.6) \end{aligned}$$

unter der Nebenbedingung:

$$\forall j \in \mathbb{N}_{\leq r_\varepsilon} \forall \mu \in \mathbb{N}_{\leq d} \forall \nu \in \mathbb{N}_{\leq d} \setminus \{\mu\} : \|\xi_{\varepsilon\, j\mu}\| = \|\xi_{\varepsilon\, j\nu}\|, \tag{3.7}$$

erfüllt sind, wobei $r_\varepsilon \in \mathbb{N}_{\leq R}$ kleinstmöglich sein soll.

3.2 Definition der Zielfunktion

Im Folgenden wird die Zielfunktion der Approximationsaufgabe definiert. Daneben wird vereinbart, dass die hier definierten Terme in den folgenden Teilabschnitten als Notation dienen.
Die Minimierung des Abstandes findet auf dem Repräsentantensystem bezüglich der Funktion

$$\frac{1}{2}\|\cdot\|^2 \circ \mathfrak{E} : \bigtimes_{\mu=1}^{d} \bigtimes_{j=1}^{r} \mathbb{R}^{t_\mu} \to \mathbb{R}_{\geq 0}$$

statt, wobei die Abbildung $\mathfrak{E}$ wie folgt definiert ist:

$$\begin{aligned} \mathfrak{E} : \bigtimes_{\mu=1}^{d} \bigtimes_{j=1}^{r} \mathbb{R}^{t_\mu} &\to \mathcal{S}_{\leq r} \\ \hat{\xi} := (\xi_{j\mu} : \mu \in \mathbb{N}_{\leq d},\ j \in \mathbb{N}_{\leq r}) &\mapsto \mathfrak{E}(\hat{\xi}) := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{i\mu}. \end{aligned}$$

Notation 3.2.1. *Ist $\underline{t} := (t_1, \ldots, t_d)$, dann ist zur abkürzenden Schreibweise $\mathfrak{R}_{d,r,\underline{t}} := \times_{\mu=1}^{d} \times_{j=1}^{r} \mathbb{R}^{t_\mu}$ gesetzt. Ferner wird ein Repräsentantensystem von $\xi \in \mathcal{S}_{\leq r}$ kürzer mit $\hat{\xi}$ bezeichnet, d.h. sei $\hat{\xi} := (\xi_{j\mu} : \mu \in \mathbb{N}_{\leq d},\ j \in \mathbb{N}_{\leq r})$ derart, dass*

$$\xi = \mathfrak{E}(\hat{\xi}) = \sum_{j=1}^{r} \xi_j = \sum_{j=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} \xi_{j\mu}}_{=:\xi_j}$$

erfüllt ist.

Da

$$\frac{1}{2}\|\alpha - \xi\|^2 = \underbrace{\frac{1}{2}\|\alpha\|^2}_{=konst.} - \langle \xi, \alpha \rangle + \frac{1}{2}\|\xi\|^2$$

und bei der Minimierung konstante Ausdrücke vernachlässigt werden können sowie eine Normierung aller Teile der Zielfunktion mit der Norm von α angebracht ist, definiert man den Hauptteil der Zielfunktion zu

$$f_1 : \mathfrak{R}_{d,r,\underline{t}} \rightarrow \mathbb{R}_{\geq -\frac{1}{2}} \tag{3.8}$$

$$\hat{\xi} \mapsto f_1(\hat{\xi}) := \frac{1}{\|\alpha\|^2}\left[- \langle \alpha, \xi \rangle + \frac{1}{2}\|\xi\|^2 \right]. \tag{3.9}$$

Ferner gilt

$$\begin{aligned} f_1(\hat{\xi}) &= \frac{1}{\|\alpha\|^2}\left[-\sum_{j=1}^{r}\sum_{i=1}^{R} \langle \alpha_i, \xi_j \rangle + \frac{1}{2}\sum_{j=1}^{r}\sum_{j'=1}^{r} \langle \xi_j, \xi_{j'} \rangle \right] \qquad (3.10) \\ &= \frac{1}{\|\alpha\|^2}\left[-\sum_{j=1}^{r}\sum_{i=1}^{R}\prod_{\mu=1}^{d} \langle \alpha_{i\mu}, \xi_{j\mu} \rangle + \frac{1}{2}\sum_{j=1}^{r}\sum_{j'=1}^{r}\prod_{\mu=1}^{d} \langle \xi_{j\mu}, \xi_{j'\mu} \rangle \right]. \end{aligned}$$

Bei der numerischen Lösung der restringierten Minimierungsaufgabe kann die aufwändige Erfüllung der Kuhn-Tucker-Bedingungen umgangen werden. Um die Nebenbedingung (3.7) zu erfüllen, wird ein Strafterm benutzt. Hierfür wird die nachstehende Funktion g_1 derart konstruiert, dass $g_1(\hat{\xi}) = 0$ ist, falls $\xi \in \mathcal{S}_r$ die Nebenbedingung (3.7) erfüllt. Getreu Kapitel 5, Seiten 63 f., ist die Nebenbedingung (3.7) bei der praktischen Umsetzung einfach zu erfüllen. Die Funktion g_1 ist wie folgt definiert:

$$g_1 : \mathfrak{R}_{d,r,\underline{t}} \rightarrow \mathbb{R}_{\geq 0} \tag{3.11}$$

$$\hat{\xi} \mapsto g_1(\hat{\xi}) := \frac{1}{8\sqrt[d]{\|\alpha\|^4}} \sum_{j=1}^{r}\sum_{\mu=1}^{d}\sum_{\nu=1,\nu\neq\mu}^{d} \left(\|\xi_{j\mu}\|^2 - \|\xi_{j\nu}\|^2\right)^2 \tag{3.12}$$

Diese Nebenbedingung ist sinnvoll, denn nach Korollar 1.1.13, Seite 9, ist ein Repräsentantensystem nicht eindeutig bestimmt. Überdies ist diese Darstellung durch die Nebenbedingung (3.11) noch nicht eindeutig, denn durch Multiplikation einzelner Repräsentantenvektoren mit Faktoren von -1, so dass das Produkt dieser Faktoren 1 ergibt, erhält man ein neues System $\hat{\xi}'$, welches dieselbe Elementartensor-Summe darstellt und die Nebenbedingung (3.7) erfüllt. Nun ist aber dieses neue Repräsentantensystem $\hat{\xi}'$ relativ weit entfernt von $\hat{\xi}$, womit angenommen werden kann, dass in der Umgebung $U(\hat{\xi}^*)$ des Minimums ξ^* in Gleichung (3.3) die Darstellung eindeutig ist.
Gemäß Satz 2.1.8 und Lemma 2.1.8 ist die Approximation nur auf $\mathcal{S}_r^c$ wohldefiniert. Existiert das Minimum der Approximationsaufgabe in $\mathcal{S}_r$ nicht, dann muss gemäß Korollar 2.3.5 eine minimierende Folge unbeschränkte Summanden

haben. Mit Hilfe einer weiteren Nebenbedingung kann die Norm der Summanden beschränkt werden. Hierfür wird folgende Funktion g_2 benutzt:

$$\begin{aligned} g_2 : \mathfrak{R}_{d,r,\underline{t}} &\to \mathbb{R}_{\geq 0} && (3.13)\\ \hat{\xi} &\mapsto g_2(\hat{\xi}) := \frac{1}{2\|\alpha\|^2} \sum_{j=1}^{r} \|\xi_j\|^2 = \frac{1}{2\|\alpha\|^2} \sum_{j=1}^{r} \prod_{\mu=1}^{d} \|\xi_{j\mu}\|^2. && (3.14) \end{aligned}$$

Mittels g_2 wird zusätzlich über die Norm der Summanden minimiert. Diese Herangehensweise hat den Vorteil, dass die Konstante c nicht zuvor gewählt werden muss und eine aufwändige numerische Behandlung der Kuhn-Tucker-Bedingung entfällt.
Damit ist die Zielfunktion beschrieben und man erhält insgesamt

$$\begin{aligned} f : \mathfrak{R}_{d,r,\underline{t}} &\to \mathbb{R}_{\geq -\frac{1}{2}} && (3.15)\\ \hat{\xi} &\mapsto f(\hat{\xi}) := f_1(\hat{\xi}) + \lambda_1 g_1(\hat{\xi}) + \lambda_2 g_2(\hat{\xi}), && (3.16) \end{aligned}$$

wobei $\lambda_1, \lambda_2 \in \mathbb{R}$ sind. Der Parameter λ_2 wird bei der praktischen Umsetzung derart gewählt, dass der Einfluss auf den Hauptteil f_1 nicht signifikant ist.

Bemerkung 3.2.2. *Offenbar ist f aus Gleichung (3.15) ein Polynom in mehreren Unbekannten mit einem Grad von höchstens $2d$. Aus diesem Grund ist die Zielfunktion beliebig oft stetig differenzierbar.*

Wie schon erwähnt, muss die Zielfunktion f normiert sein; infolgedessen ist die Berechnung von $\|\alpha\|^2$ notwendig.

Lemma 3.2.3. *Zur Berechnung von $\|\alpha\|^2$ werden*

$$2 \cdot \binom{R+1}{2} \cdot \sum_{\mu=1}^{d} t_\mu \tag{3.17}$$

Rechenoperationen benötigt.

Beweis. Das Quadrat der Norm von $\alpha = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} \alpha_{i\mu}$ berechnet sich wie folgt:

$$\|\alpha\|^2 = \sum_{i_1=1}^{R} \left(\prod_{\mu=1}^{d} \langle \alpha_{i_1\mu}, \alpha_{i_1\mu} \rangle + 2 \cdot \sum_{i_2=1}^{i_1-1} \prod_{\mu=1}^{d} \langle \alpha_{i_1\mu}, \alpha_{i_2\mu} \rangle \right). \tag{3.18}$$

Der Rechenaufwand zur Berechnung eines Skalarproduktes der Form

$$\prod_{\mu=1}^{d} \langle \alpha_{i_1\mu}, \alpha_{i_2\mu} \rangle$$

beträgt

$$\sum_{\mu=1}^{d} (2 \cdot t_\mu - 1) + d - 1 = 2 \cdot \sum_{\mu=1}^{d} t_\mu - 1.$$

In Gleichung (3.18) müssen $\binom{R+1}{2}$ Terme dieser Art berechnet werden, zusätzlich fallen $\binom{R+1}{2} - 1$ Additionen und eine Multiplikation an. Somit ergibt sich ein Gesamtaufwand von

$$\binom{R+1}{2}\left(2 \cdot \sum_{\mu=1}^{d} t_\mu - 1\right) + \binom{R+1}{2} - 1 + 1 = 2 \cdot \binom{R+1}{2} \cdot \sum_{\mu=1}^{d} t_\mu.$$

■

3.3 Die erste Ableitung der Zielfunktion

Im Folgenden wird die erste Ableitung der Zielfunktion angegeben.

Notation 3.3.1. *Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $\xi := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{S}_{\leq r}$ und $h \in \{f, f_1, g_1, g_2\}$. Die partiellen Ableitungen von h an der Stelle $\hat{\xi}$ werden bei festem μ_1 und j_1 zu einem Vektor $h'_{j_1\mu_1}(\hat{\xi}) \in \mathbb{R}^{t_{\mu_1}}$ zusammengefasst, dessen Einträge sind für alle $l_1 \in \mathbb{N}_{\leq t_{\mu_1}}$ wie folgt definiert:*

$$(h'_{j_1\mu_1}(\hat{\xi}))_{l_1} := \frac{\partial h(\hat{\xi})}{\partial (\xi_{\mu_1 j_1})_{l_1}}.$$

Ferner bezeichnet man $h'_{j_1\mu_1}(\hat{\xi})$ als Komponente *von $h'(\hat{\xi})$ zum Indexpaar (j_1, μ_1) oder kürzer als Komponente, falls das Indexpaar aus dem Zusammenhang klar erkennbar ist.*

Notation 3.3.2 (Skalarprodukt mit einfacher Auslassung)**.** *Sei $\mu_1 \in \mathbb{N}_{\leq d}$, dann ist $\langle \cdot, \cdot \rangle_{\mu_1}$ wie folgt definiert:*

$$\langle \cdot, \cdot \rangle_{\mu_1} \quad : \quad \mathcal{T}_1 \times \mathcal{T}_1 \to \mathbb{R} \tag{3.19}$$

$$(v, w) \mapsto \langle v, w \rangle_{\mu_1} \quad := \quad \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1\}} \langle v_\mu, w_\mu \rangle . \tag{3.20}$$

Lemma 3.3.3. *Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $\xi := \sum_{j=1}^{r} \xi_j = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{T}_r$ und f_1 wie in Gleichung (3.8) definiert. Dann sind die Komponenten von $f'_1(\hat{\xi})$ wie folgt bestimmt:*

$$f'_{1\, j_1\mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2} \left[-\sum_{i=1}^{R} \langle \alpha_i, \xi_{j_1} \rangle_{\mu_1} \alpha_{i\mu_1} + \sum_{j=1}^{r} \langle \xi_j, \xi_{j_1} \rangle_{\mu_1} \xi_{j\mu_1} \right]. \tag{3.21}$$

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$ Es gilt

$$\begin{aligned} f_1(\hat{\xi}) &= \frac{1}{\|\alpha\|^2}\left[-\sum_{j=1}^{r}\sum_{i=1}^{R}\langle\alpha_i,\xi_j\rangle + \frac{1}{2}\sum_{j=1}^{r}\sum_{j'=1}^{r}\langle\xi_j,\xi_{j'}\rangle\right] \\ &= \frac{1}{\|\alpha\|^2}\left[-\sum_{j=1}^{r}\sum_{i=1}^{R}\langle\alpha_i,\xi_j\rangle_{\mu_1}\langle\alpha_{i\mu_1},\xi_{j\mu_1}\rangle\right. \\ &+ \left.\frac{1}{2}\sum_{j=1}^{r}\sum_{j'=1}^{r}\langle\xi_j,\xi_{j'}\rangle_{\mu_1}\langle\xi_{j\mu_1},\xi_{j'\mu_1}\rangle\right], \end{aligned}$$

$$\Rightarrow f'_{1j_1\mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2}\left[-\sum_{i=1}^{R}\langle\alpha_i,\xi_{j_1}\rangle_{\mu_1}\alpha_{i\mu_1} + \sum_{j=1}^{r}\langle\xi_j,\xi_{j_1}\rangle_{\mu_1}\xi_{j\mu_1}\right].$$

■

Lemma 3.3.4. *Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $\xi := \sum_{j=1}^{r}\xi_j = \sum_{j=1}^{r}\bigotimes_{\mu=1}^{d}\xi_{j\mu} \in \mathcal{T}_r$ und g_1 wie in Gleichung (3.11) definiert. Dann berechnen sich die Komponenten von $g'_1(\hat{\xi})$ wie folgt:*

$$g'_{1j_1\mu_1}(\hat{\xi}) = \frac{1}{\sqrt[d]{\|\alpha\|^4}}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2\right)\right]\xi_{j_1\mu_1}. \tag{3.22}$$

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$. Es gilt

$$g_1(\hat{\xi}) = \frac{1}{8\sqrt[d]{\|\alpha\|^4}}\sum_{j=1}^{r}\sum_{\mu=1}^{d}\sum_{\nu=1,\nu\neq\mu}^{d}\left(\|\xi_{j\mu}\|^2 - \|\xi_{j\nu}\|^2\right)^2,$$

$$\begin{aligned} \Rightarrow g'_{1j_1\mu_1}(\hat{\xi}) &= \frac{1}{2\sqrt[d]{\|\alpha\|^4}}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2\right)\right. \\ &+ \left.\sum_{\nu=1,\nu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\nu}\|^2\right)\right]\xi_{j_1\mu_1} \\ &= \frac{1}{\sqrt[d]{\|\alpha\|^4}}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2\right)\right]\xi_{j_1\mu_1}. \end{aligned}$$

■

Lemma 3.3.5. *Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $\xi := \sum_{j=1}^{r}\xi_j = \sum_{j=1}^{r}\bigotimes_{\mu=1}^{d}\xi_{j\mu} \in \mathcal{T}_r$ und g_2 wie in Gleichung (3.13) definiert. Dann gilt für die Komponenten von $g'_2(\hat{\xi})$*

$$g'_{2j_1\mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2}\langle\xi_{j_1},\xi_{j_1}\rangle_{\mu_1}\xi_{j_1\mu_1}. \tag{3.23}$$

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, dann gilt

$$\begin{aligned} g_2(\hat{\xi}) &= \frac{1}{2\|\alpha\|^2} \sum_{j=1}^{r} \|\xi_j\|^2 = \frac{1}{2\|\alpha\|^2} \sum_{j=1}^{r} \langle \xi_j, \xi_j \rangle_{\mu_1} \langle \xi_{j\mu_1}, \xi_{j\mu_1} \rangle , \\ &\Rightarrow g'_{2 j_1 \mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2} \langle \xi_{j_1}, \xi_{j_1} \rangle_{\mu_1} \xi_{j_1 \mu_1}. \end{aligned}$$

■

Korollar 3.3.6. *Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $\xi := \sum_{j=1}^{r} \xi_j = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{T}_r$ und f wie in Gleichung (3.15) definiert. Dann berechnen sich die Komponenten von $f'(\hat{\xi})$ wie folgt:*

$$f'_{j_1 \mu_1}(\hat{\xi}) = f'_{1 j_1 \mu_1}(\hat{\xi}) + \lambda_1 g'_{1 j_1 \mu_1}(\hat{\xi}) + \lambda_2 g'_{2 j_1 \mu_1}(\hat{\xi}), \tag{3.24}$$

wobei $f'_{1 j_1 \mu_1}(\hat{\xi})$, $g'_{1 j_1 \mu_1}(\hat{\xi})$ bzw. $g'_{2 j_1 \mu_1}(\hat{\xi})$ wie in den Gleichungen (3.21), (3.22) bzw. (3.23) definiert sind.

3.4 Die zweite Ableitung der Zielfunktion

Die numerische Behandlung der Approximationsaufgabe wird mit Hilfe von einem Newton-ähnlichen Abstiegsverfahren mit parameterabhängiger Suchrichtung durchgeführt, wie im Abschnitt 4.4 beschrieben. Aus diesem Grund ist die Berechnung der Hesse-Matrix notwendig.

Notation 3.4.1. *Seien $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ sowie $\xi := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{S}_{\leq r}$ und $h \in \{f, f_1, g_1, g_2\}$. Die zweiten partiellen Ableitungen von h an der Stelle $\hat{\xi}$ werden bei festem (μ_1, μ_2) und (j_1, j_2) zu einer Matrix $h''_{\mu_1 \mu_2 j_1 j_2}(\hat{\xi}) \in \mathbb{R}^{t_{\mu_1} \times t_{\mu_2}}$ zusammengefasst, deren Einträge sind für alle $l_1 \in \mathbb{N}_{\leq t_{\mu_1}}$ und $l_2 \in \mathbb{N}_{\leq t_{\mu_2}}$ wie folgt definiert:*

$$(h''_{\mu_1 \mu_2 j_1 j_2}(\hat{\xi}))_{l_1 l_2} := \frac{\partial^2 h(\hat{\xi})}{\partial (\xi_{\mu_1 j_1})_{l_1} \partial (\xi_{\mu_2 j_2})_{l_2}}.$$

Man bezeichnet $h''_{\mu_1 \mu_2 j_1 j_2}(\hat{\xi})$ als Segment *von $h''(\hat{\xi})$ zum Indexpaar (μ_1, μ_2, j_1, j_2), oder kürzer als Segment, falls dies aus dem Zusammenhang klar erkennbar ist.*

Notation 3.4.2 (Skalarprodukt mit zweifacher Auslassung)**.** *Für $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ mit $\mu_1 \neq \mu_2$ ist $\langle \cdot, \cdot \rangle_{\mu_1 \mu_2}$ wie folgt definiert:*

$$\langle \cdot, \cdot \rangle_{\mu_1 \mu_2} : \mathcal{T}_1 \times \mathcal{T}_1 \to \mathbb{R} \tag{3.25}$$

$$(v, w) \mapsto \langle v, w \rangle_{\mu_1 \mu_2} := \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1, \mu_2\}} \langle v_\mu, w_\mu \rangle . \tag{3.26}$$

Daneben ist für $i, j \in \mathbb{N}$

$$\bar{\delta}_{ij} := 1 - \delta_{ij} \tag{3.27}$$

gesetzt, wobei das Kronecker-Delta wie üblich definiert ist, d.h.

$$\delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{3.28}$$

Lemma 3.4.3. *Seien* $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ *sowie* $\xi := \sum_{j=1}^{r} \xi_j = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{T}_r$ *und* f_1 *wie in Gleichung (3.8) definiert. Dann berechnen sich die Segmente von* $f_1''(\hat{\xi})$ *wie folgt:*

$$\begin{aligned} f''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}) &= \frac{1}{\|\alpha\|^2} \Big[A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) \\ &- D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) \Big], \end{aligned} \tag{3.29}$$

wobei

$$A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \delta_{\mu_1\mu_2} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, \tag{3.30}$$

$$B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1\mu_2} \xi_{j_2\mu_1} \xi^t_{j_1\mu_2}, \tag{3.31}$$

$$C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2} \delta_{j_1 j_2} \sum_{j=1}^{r} \langle \xi_j, \xi_{j_1} \rangle_{\mu_1\mu_2} \xi_{j\mu_1} \xi^t_{j\mu_2}, \tag{3.32}$$

$$D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2} \delta_{j_1 j_2} \sum_{i=1}^{R} \langle \alpha_i, \xi_{j_1} \rangle_{\mu_1\mu_2} \alpha_{i\mu_1} \alpha^t_{i\mu_2} \tag{3.33}$$

gesetzt sind.

Beweis. Seien $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ und $l_1 \in \mathbb{N}_{\leq t_{\mu_1}}$, $l_2 \in \mathbb{N}_{\leq t_{\mu_2}}$, es ist dann gemäß Gleichung (3.21)

$$\frac{\partial f_1(\hat{\xi})}{\partial \hat{\xi}_{\mu_1 j_1 l_1}} = \frac{1}{\|\alpha\|^2} \left[\sum_{j=1}^{r} \langle \xi_j, \xi_{j_1} \rangle_{\mu_1} (\xi_{j\mu_1})_{l_1} - \sum_{i=1}^{R} \langle \alpha_i, \xi_{j_1} \rangle_{\mu_1} (\alpha_{i\mu_1})_{l_1} \right].$$

Weiter gilt

$$\begin{aligned} \frac{\partial^2 f_1(\hat{\xi})}{\partial \hat{\xi}_{\mu_1 j_1 l_1} \partial \hat{\xi}_{\mu_2 j_2 l_2}} &= \frac{1}{\|\alpha\|^2} \frac{\partial}{\partial \hat{\xi}_{\mu_2 j_2 l_2}} \left[\sum_{j=1}^{r} \langle \xi_j, \xi_{j_1} \rangle_{\mu_1} (\xi_{j\mu_1})_{l_1} \right. \\ &- \left. \sum_{i=1}^{R} \langle \alpha_i, \xi_{j_1} \rangle_{\mu_1} (\alpha_{i\mu_1})_{l_1} \right] \end{aligned}$$

sowie

$$\begin{aligned}\frac{\partial}{\partial\hat{\xi}_{\mu_2 j_2 l_2}}\sum_{j=1}^{r}\langle\xi_j,\xi_{j_1}\rangle_{\mu_1}(\xi_{j\mu_1})_{l_1} &= \underbrace{\delta_{\mu_1\mu_2}\langle\xi_{j_1},\xi_{j_2}\rangle_{\mu_1}\delta_{l_1 l_2}}_{(A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}))_{l_1 l_2}=}\\ &+ \underbrace{\overline{\delta}_{\mu_1\mu_2}\langle\xi_{j_1},\xi_{j_2}\rangle_{\mu_1\mu_2}(\xi_{j_2\mu_1})_{l_1}(\xi_{j_1\mu_2})_{l_2}}_{(B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}))_{l_1 l_2}=}\\ &+ \underbrace{\overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}\sum_{j=1}^{r}\langle\xi_j,\xi_{j_1}\rangle_{\mu_1\mu_2}(\xi_{j\mu_1})_{l_1}(\xi_{j\mu_2})_{l_2}}_{(C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}))_{l_1 l_2}=},\end{aligned}$$

und

$$\frac{\partial}{\partial\hat{\xi}_{\mu_2 j_2 l_2}}\sum_{i=1}^{R}\langle\alpha_i,\xi_{j_1}\rangle_{\mu_1}(\alpha_{i\mu_1})_{l_1} = \underbrace{\overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}\sum_{i=1}^{R}\langle\alpha_i,\xi_{j_1}\rangle_{\mu_1\mu_2}(\alpha_{i\mu_1})_{l_1}(\alpha_{i\mu_2})_{l_2}}_{(D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}))_{l_1 l_2}=}.$$

Insgesamt folgt somit die Behauptung. ■

Lemma 3.4.4. *Seien $\mu_1,\mu_2\in\mathbb{N}_{\leq d}$, $j_1,j_2\in\mathbb{N}_{\leq r}$ sowie $\xi:=\sum_{j=1}^{r}\xi_j=\sum_{j=1}^{r}\bigotimes_{\mu=1}^{d}\xi_{j\mu}\in\mathcal{T}_r$ und g_1 wie in Gleichung (3.11) definiert. Dann gilt für die Segmente von $g_1''(\hat{\xi})$*

$$g_1''{}_{\mu_1\mu_2 j_1 j_2}(\hat{\xi})=\frac{1}{\sqrt[d]{\|\alpha\|^4}}\left[G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi})+G_{2\mu_1\mu_2 j_1 j_2}(\hat{\xi})\right], \tag{3.34}$$

wobei

$$\begin{aligned}G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi}) &= \delta_{\mu_1\mu_2}\delta_{j_1 j_2}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2-\|\xi_{j_1\mu}\|^2\right)Id_{\mathbb{R}^{t_{\mu_1}}}\right.\\ &+ \left.2(d-1)\xi_{j_1\mu_1}\xi_{j_1\mu_1}^t\right],\\ G_{2\mu_1\mu_2 j_1 j_2}(\hat{\xi}) &= \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}(-2)\xi_{j_1\mu_1}\xi_{j_1\mu_2}^t\end{aligned}$$

gesetzt sind.

Beweis. Seien $\mu_1,\mu_2\in\mathbb{N}_{\leq d}$, $j_1,j_2\in\mathbb{N}_{\leq r}$ und $l_1\in\mathbb{N}_{\leq t_{\mu_1}}$, $l_2\in\mathbb{N}_{\leq t_{\mu_2}}$. Gemäß Gleichung (3.22) gilt

$$\frac{\partial g_1(\hat{\xi})}{\partial\hat{\xi}_{\mu_1 j_1 l_1}}=\frac{1}{\sqrt[d]{\|\alpha\|^4}}\underbrace{\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2-\|\xi_{j_1\mu}\|^2\right)\right](\xi_{j_1\mu_1})_{l_1}}_{\tilde{g_1}'_{j_1\mu_1 l_1}(\hat{\xi}):=}.$$

Ferner folgt mit $\Delta^{j_1 j_2}_{\mu_1\mu_2} := \delta_{\mu_1\mu_2}\delta_{j_1 j_2}$

$$\begin{aligned}\frac{\partial \tilde{g_1}'_{j_1\mu_1 l_1}(\hat{\xi})}{\partial \hat{\xi}_{\mu_2 j_2 l_2}} &= \Delta^{j_1 j_2}_{\mu_1\mu_2}\left[\sum_{\substack{\mu\neq\mu_1\\ \mu=1}}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2\right)\delta_{l_1 l_2} 2(d-1)(\xi_{j_1\mu_1})_{l_1}(\xi_{j_1\mu_1})_{l_2}\right]\\ &+ \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}(-2)(\xi_{j_1\mu_1})_{l_1}(\xi_{j_1\mu_2})_{l_2},\end{aligned}$$

woraus die Behauptung folgt. ■

Lemma 3.4.5. *Seien $\xi := \sum_{j=1}^{r}\xi_j = \sum_{j=1}^{r}\bigotimes_{\mu=1}^{d}\xi_{j\mu} \in \mathcal{T}_r$, $\mu_1,\mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ und g_2 wie in Gleichung (3.13) definiert. Dann berechnen sich die Segmente von $g_2''(\hat{\xi})$ wie folgt:*

$$g''_{2\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \frac{1}{\|\alpha\|^2}\delta_{j_1 j_2}\begin{cases}\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1}\mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, & \mu_1 = \mu_2;\\ 2\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1\mu_2}\xi_{j_1\mu_1}\xi^t_{j_1\mu_2}, & \textit{sonst.}\end{cases} \tag{3.35}$$

Beweis. Für $\mu_1,\mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ und $l_1 \in \mathbb{N}_{\leq t_{\mu_1}}$, $l_2 \in \mathbb{N}_{\leq t_{\mu_2}}$, ist gemäß Gleichung (3.23)

$$\frac{\partial g_2(\hat{\xi})}{\partial \hat{\xi}_{\mu_1 j_1 l_1}} = \frac{1}{\|\alpha\|^2}\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1}(\xi_{j_1\mu_1})_{l_1}.$$

Weiter folgt

$$\begin{aligned}\frac{\partial}{\partial \hat{\xi}_{\mu_2 j_2 l_2}}\left[\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1}(\xi_{j_1\mu_1})_{l_1}\right] &= \delta_{\mu_1\mu_2}\delta_{j_1 j_2}\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1}\delta_{l_1 l_2}\\ &+ \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}\langle \xi_{j_1}, \xi_{j_1}\rangle_{\mu_1\mu_2}(\xi_{j_1\mu_1})_{l_1}(\xi_{j_1\mu_2})_{l_2},\end{aligned}$$

womit die Behauptung gezeigt ist. ■

Korollar 3.4.6. *Seien $\mu_1,\mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ sowie $\xi := \sum_{j=1}^{r}\xi_j = \sum_{j=1}^{r}\bigotimes_{\mu=1}^{d}\xi_{j\mu} \in \mathcal{T}_r$ und f wie in Gleichung (3.15) definiert. Dann gilt für die Segmente von $f''(\hat{\xi})$ folgendes:*

$$f''_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = f''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + \lambda_1 g''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + \lambda_2 g''_{2\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}), \tag{3.36}$$

wobei $f''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi})$, $g''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi})$ bzw. $g''_{2\,\mu_1\mu_2 j_1 j_2}(\hat{\xi})$ wie in den Gleichungen (3.29), (3.34) bzw. (3.35) definiert sind.

4 Methoden zur numerischen Behandlung nichtlinearer Optimierungsaufgaben

Bevor die numerische Lösung der Approximationsaufgabe diskutiert wird, sei an dieser Stelle ein Querschnitt durch die Theorie nichtlinearer Optimierungsaufgaben eingefügt. Dieses Kapitel gibt einen kurz gefassten Überblick über bekannte Minimierungsverfahren und deren Konvergenzeigenschaften und dient damit dem besseren Verständnis. Daneben werden im letzten Abschnitt leicht abgewandelte Methoden vorgestellt, welche auf das im Mittelpunkt der Arbeit stehende Optimierungsproblem abgestimmt sind.
Seien im Weiteren $n \in \mathbb{N}$ und $f : \mathbb{R}^n \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion.

4.1 Newton-Verfahren

In diesem Abschnitt wird das Newton[1]-Verfahren zur Minimierung einer zweimal stetig differenzierbaren Funktion vorgestellt und dessen Konvergenztheorie untersucht. Hierbei besteht die zentrale Idee des Newton-Verfahrens darin, das unrestringierte Minimierungsproblem

$$\text{Finde ein } x^* \in \mathbb{R}^n \text{ mit: } f(x^*) \leq f(x) \text{ für alle } x \in \mathbb{R}^n. \tag{4.1}$$

zu lösen, indem man sukzessiv die quadratische Näherung

$$q_k(x) := f(x^k) + \left\langle f'(x^k), x - x^k \right\rangle + \frac{1}{2} \left\langle x - x^k, f''(x^k)(x - x^k) \right\rangle \tag{4.2}$$

zu minimieren versucht, wobei $x^k \in \mathbb{R}^n$ den aktuellen Iterationspunkt bezeichnet. Ist die Hesse[2]-Matrix f'' positiv definit, so ist x^{k+1} genau dann Lösung von

$$\min_{x \in \mathbb{R}^n} q_k(x), \tag{4.3}$$

[1] Sir Isaac Newton, englischer Physiker, Mathematiker, Astronom, Alchemist, Philosoph und Verwaltungsbeamter, nach Gregorianischem Kalender: $*$ 4. Januar 1643, Woolsthorpe-by-Colsterworth in Lincolnshire, † 31. März 1727 in Kensington (nach dem damals in England noch geltenden Julianischen Kalender: $*$ 25. Dezember 1642, † 20. März 1727).

[2] Ludwig Otto Hesse, deutscher Mathematiker, $*$ 22. April 1811, Königsberg (Ostpreußen), † 4. August 1874 in München.

wenn x^{k+1} der Bedingung

$$q_k'(x) = 0 \tag{4.4}$$

für einen stationären Punkt[3] von q_k genügt. Wegen

$$g_k'(x) = f'(x^k) + f''(x^k)(x - x^k) \tag{4.5}$$

ergibt sich hieraus

$$x^{k+1} = x^k - \Big(f''(x^k)\Big)^{-1} f'(x^k). \tag{4.6}$$

Natürlich wird man die explizite Berechnung der inversen Hesse-Matrix vermeiden, vielmehr bestimmt man zunächst eine Lösung $d^k \in \mathbb{R}^n$ des linearen Gleichungssystems

$$f''(x^k)d = f'(x^k) \tag{4.7}$$

und setzt anschließend

$$x^{k+1} := x^k - d^k. \tag{4.8}$$

Das hierbei zu lösende lineare Gleichungssystem (4.7) wird häufig Newton-Gleichung und der Vektor d^k Newton-Richtung im Punkt x^k genannt. Insgesamt erhält man folgenden Algorithmus.

Algorithmus 4.1.1 Lokales Newton-Verfahren

Wähle $x^1 \in \mathbb{R}^n$, $\varepsilon \in \mathbb{R}_{\geq 0}$ und setze $k := 1$.
while $\|f'(x^k)\| > \varepsilon$ **do**
  Berechne $d^k \in \mathbb{R}^n$ mit: $f''(x^k)d^k = f'(x^k)$.
  Setze $x^{k+1} := x^k - d^k$ und $k \leftarrow k + 1$.
end while

Notation 4.1.1 (Newton-Folge)**.** *Eine mittels Algorithmus 4.1.1 erzeugte Folge heißt* Newton-Folge *von f zum Startwert $x^1 \in \mathbb{R}^n$.*

Satz 4.1.2. *Sei $x^* \in \mathbb{R}^n$ ein stationärer Punkt von f und $f''(x^*)$ regulär. Dann existiert eine Umgebung U von x^*, so dass für jeden Startwert $x^1 \in U$ die Newton-Folge in U wohldefiniert ist und gegen x^* konvergiert. Daneben gelten:*

(i) Die Konvergenzrate ist mindestens superlinear.

(ii) Ist f'' lokal Lipschitz[4]-stetig, so ist die Konvergenzrate mindestens quadratisch.

Beweis. [9, Satz 9.2, Seiten 84 f.] oder [24, Satz 3.1.1, Seiten 68 f., Satz 3.1.2, Seite 69]. ■

Bemerkung 4.1.3. *Das lokale Newton-Verfahren wird im Allgemeinen nicht nur gegen lokale Minima konvergieren, sondern auch gegen lokale Maxima von f. Dieser eigentlich unerwünschte Effekt tritt beim gedämpften Newton-Verfahren nicht mehr auf.*

[3] $x^* \in \mathbb{R}^n$ ist ein stationärer Punkt von $f :\Leftrightarrow f'(x^*) = 0$.

[4] Rudolf Otto Sigismund Lipschitz, deutscher Mathematiker, $*$ 14. Mai 1832, Königsberg (Ostpreußen), † 7. Oktober 1903 in Bonn.

4.2 Newton-ähnliches Verfahren

Im vorangegangenen Abschnitt wurde gezeigt, dass das Newton-Verfahren für zufriedenstellend dicht an einem stationären Punkt x^* gelegene Startwerte x^1 unbeschränkt durchführbar ist und mindestens superlinear bzw. quadratisch gegen x^* konvergiert. Dazu ist pro Iterationsschritt die Berechnung der Hesse-Matrix $f''(x^k)$ bzw. der Newton-Richtung notwendig, was aus verschiedenen Gründen häufig nicht erwünscht ist. Allerdings kann man das Newton-Verfahren als einen Prototyp schnell konvergenter Verfahren ansehen. Denn die folgenden Sätze besagen, dass eine gegen x^* konvergente Folge $(x^k)_{k\in\mathbb{N}}$ genau dann mindestens superlinear bzw. quadratisch konvergiert, wenn x^{k+1} $(k \in \mathbb{N})$ die Gleichung

$$f'(x^k) + f''(x^k)(x - x^k) = 0$$

bis auf $o(\|x^{k+1} - x^k\|)$ bzw. $O(\|x^{k+1} - x^k\|^2)$ erfüllt. Dabei heißt eine Folge $(u^k)_{k\in\mathbb{N}}$ in einem normierten Raum X bezüglich einer Nullfolge $(\alpha_k)_{k\in\mathbb{N}}$ positiver reeller Zahlen

$$o(\alpha_k) \quad \text{bzw.} \quad O(\alpha_k),$$

falls $\left(\frac{\|u^k\|}{\alpha_k}\right)_{k\in\mathbb{N}}$ eine Nullfolge bzw. beschränkte Folge ist.

Satz 4.2.1 (Charakterisierung der superlinearen Konvergenz bei Optimierungsaufgaben)**.** *Seien $(x^k)_{k\in\mathbb{N}}$ eine konvergente nichttriviale[5] Folge in $\mathbb{R}^n$ mit Grenzwert $x^* \in \mathbb{R}^n$ und $f''(x^*)$ regulär. Dann sind äquivalent:*

(i) $x^k \xrightarrow[k\to\infty]{} x^*$ *superlinear und* $f'(x^*) = 0$.

(ii) $\|f'(x^k) + f''(x^k)(x^{k+1} - x^k)\| = o(\|x^{k+1} - x^k\|)$.

(iii) $\|f'(x^k) + f''(x^*)(x^{k+1} - x^k)\| = o(\|x^{k+1} - x^k\|)$.

Beweis. [9, Satz 7.8, Seiten 60 ff.]. ■

Dieser Satz rechtfertigt folgende Definition.

Definition 4.2.2 (Newton-ähnlich)**.** *Eine nichttriviale Folge $(x^k)_{k\in\mathbb{N}}$ in $\mathbb{R}^n$ heißt* Newton-ähnlich *bezüglich f, wenn*

$$\|f'(x^k) + f''(x^k)(x^{k+1} - x^k)\| = o(\|x^{k+1} - x^k\|) \tag{4.9}$$

gilt.

Bemerkung 4.2.3.

- *Mit Hilfe von Definition 4.2.2 lässt sich die Behauptung des Satzes 4.2.1 folgendermaßen aussprechen: „Die gegen x^* konvergente Folge $(x^k)_{k\in\mathbb{N}}$ ist genau dann superlinear konvergent und es gilt $f'(x^*) = 0$, wenn $(x^k)_{k\in\mathbb{N}}$ Newton-ähnlich bezüglich f ist.“*

[5] Eine Folge $(x^k)_{k\in\mathbb{N}}$ heißt nichttrivial $:\Leftrightarrow \quad \forall k \in \mathbb{N} : x^{k+1} \neq x^k$

- *Ist die Iterationsfolge $(x^k)_{k\in\mathbb{N}}$ von der Gestalt $x^{k+1} = x^k - d^k$, wobei $d^k \in \mathbb{R}^n$, für alle $k \in \mathbb{N}$, dann ist Gleichung (4.9) äquivalent zu*

$$\|f'(x^k) - f''(x^k)d^k\| = o(\|d^k\|). \tag{4.10}$$

- *Das Newton-Verfahren ist offensichtlich Newton-ähnlich. Im Übrigen genügt es, in jedem Iterationschritt die Newton-Gleichung gemäß Gleichung (4.10) mit einer erforderlichen Genauigkeit nur näherungsweise zu lösen.*

Korollar 4.2.4. *Seien $(x^k)_{k\in\mathbb{N}}$ eine konvergente nichttriviale Folge in $\mathbb{R}^n$ mit Grenzwert $x^* \in \mathbb{R}^n$, $f''(x^*)$ regulär und $(H_k)_{k\in\mathbb{N}} \subset \mathbb{R}^{n\times n}$ eine Folge invertierbarer Matrizen. Ferner gelte für alle $k \in \mathbb{N}$*

$$x^{k+1} = x^k - d^k, \tag{4.11}$$

wobei $d^k \in \mathbb{R}^n$ Lösung von

$$H_k d^k = f'(x^k) \tag{4.12}$$

ist.
Dann sind äquivalent:

(i) $x^k \xrightarrow[k\to\infty]{} x^*$ *superlinear und* $f'(x^*) = 0$.

(ii) $\| \left(f''(x^k) - H_k\right) d^k\| = o(\|d^k\|)$.

(iii) $\| \left(f''(x^*) - H_k\right) d^k\| = o(\|d^k\|)$.

Beweis. Setzt man den Ausdruck für den Gradienten $f'(x^k)$ aus Gleichung (4.12) in die entsprechenden Aussagen des Satzes 4.2.1 ein, so ergibt sich gerade die Behauptung. ■

Bemerkung 4.2.5. *Die Bedingung (iii) aus Korollar 4.2.4 ist erfüllt, wenn die Folge der Matrizen $(H_k)_{k\in\mathbb{N}}$ gegen die Hesse-Matrix $f''(x^*)$ konvergiert. Dies ist insbesondere beim Newton-Verfahren der Fall.*
Allerdings erlaubt die Bedingung (iii) wesentlich allgemeinere Folgen von Matrizen $(H_k)_{k\in\mathbb{N}}$. Lediglich die Auswertung von H_k und $f''(x^*)$ bzw. $f''(x^k)$ auf die Richtung d^k muss in etwa gleich sein. Die Folge $(H_k)_{k\in\mathbb{N}}$ muss also keineswegs gegen die Hesse-Matrix in x^* konvergieren.

Abschließend wird der Charakterisierungssatz der quadratischen Konvergenz angegeben.

Satz 4.2.6 (Charakterisierung der quadratischen Konvergenz bei Optimierungsaufgaben)**.** *Seien $(x^k)_{k\in\mathbb{N}}$ eine konvergente nichttriviale Folge in $\mathbb{R}^n$ mit Grenzwert $x^* \in \mathbb{R}^n$, $f''(x^*)$ regulär und f'' lokal Lipschitz-stetig. Dann sind äquivalent:*

(i) $x^k \xrightarrow[k\to\infty]{} x^*$ *quadratisch und* $f'(x^*) = 0$.

(ii) $\|f'(x^k) + f''(x^k)(x^{k+1} - x^k)\| = O(\|x^{k+1} - x^k\|^2)$.

(iii) $\|f'(x^k) + f''(x^*)(x^{k+1} - x^k)\| = O(\|x^{k+1} - x^k\|^2)$.

Beweis. [9, Satz 7.10, Seiten 63 f.]. ■

Bemerkung 4.2.7. *Es gelten die Voraussetzungen von Satz 4.2.6. Dann sind äquivalent:*

(i) *Die nichttriviale Folge* $(x^k)_{k\in\mathbb{N}}$ *aus Korollar 4.2.4 konvergiert quadratisch gegen* x^* *und es gilt* $f'(x^*) = 0$.

(ii) $\| \left(f''(x^k) - H_k\right) d^k\| = O(\|d^k\|^2)$.

(iii) $\| \left(f''(x^*) - H_k\right) d^k\| = O(\|d^k\|^2)$.

Beweis. Die Behauptung folgt aus Satz 4.2.6 (in Analogie zu Korollar 4.2.4). ■

4.3 Allgemeine Theorie der Abstiegsverfahren

Im vorigen Abschnitt hat sich das Newton-Verfahren als ein schnelles, aber nur lokal konvergentes Verfahren erwiesen. Bei schlechten Startwerten und stark nichtlinearen Funktionen, wenn also die quadratische Näherung q_k von Gleichung (4.2) keine gute Approximation von f ist, dann divergieren auch Newton-ähnliche Verfahren. Dies äußert sich häufig in einem kontinuierlichen Anwachsen von $\|x^k\|$ und $f(x^k)$ im Laufe der Iteration und kann bis zum Abbruch des Programms führen, falls keine zusätzlichen Maßnahmen getroffen werden.
Für die in diesem Abschnitt vorgestellte Klasse von Verfahren kann man die globale Konvergenz nachweisen. Wird das Newton-Verfahren entsprechend modifiziert, so gehört diese umgewandelte Variante des Newton-Verfahrens in die eben angesprochene Klasse und es konvergiert somit global gegen ein gesuchtes Minimum. Überdies geht die Modifikation in der Nähe der Lösung in das eigentliche Newton-Verfahren über.
Als Rahmen soll folgende Iterationsfolge dienen. Ausgehend von einem Startwert $x^1 \in \mathbb{R}^n$ und $k \in \mathbb{N}$ konstruiert man rekursiv

$$x^{k+1} = x^k - \alpha_k d^k, \tag{4.13}$$

wobei $d^k \in \mathbb{R}^n$ und $\alpha_k \in \mathbb{R}_{>0}$ sind. Die Folge $(d^k)_{k\in\mathbb{N}}$ wird Folge der Suchrichtungen und $(\alpha_k)_{k\in\mathbb{N}}$ Folge der Schrittweiten genannt. Im Hinblick auf die angestrebte globale Konvergenz scheint es natürlich, das Verfahren so zu modifizieren, dass

$$f(x^{k+1}) \leq f(x^k) \tag{4.14}$$

gilt. Ist diese Bedingung für alle Iterierten erfüllt, so wird das Verfahren als Abstiegsverfahren bezeichnet. Alle renommierten und auf globale Konvergenz ausgerichteten Minimierungsverfahren sind von diesem Typ. Gleichung (4.14) führt zu der folgenden Definition einer Abstiegsrichtung.

Definition 4.3.1 (Abstiegsrichtung)**.** *Ein $d \in \mathbb{R}^n$ heißt* Abstiegsrichtung *von f im Punkt $x \in \mathbb{R}^n$, falls für die Richtungsableitung von d im Punkte x*

$$f'(x, d) > 0 \tag{4.15}$$

gilt.

Bemerkung 4.3.2.

- *Sei $d \in \mathbb{R}^n$ eine Abstiegsrichtung von f in $x \in \mathbb{R}^n$. Dann existiert ein $r \in \mathbb{R}_{>0}$ derart, dass für alle $\alpha \in (0, r]$*

  $$f(x - \alpha d) < f(x)$$

  gilt. Dies folgt direkt aus

  $$0 > -f'(x, d) = \lim_{\alpha \to 0} \frac{f(x - \alpha d) - f(x)}{\alpha}.$$

- *Seien $(H_k)_{k \in \mathbb{N}}$ eine Folge positiv definiter Matrizen, $(x^k)_{k \in \mathbb{N}}$ und $(d^k)_{k \in \mathbb{N}}$ wie in Gleichung (4.13), wobei*

  $$\forall k \in \mathbb{N} : H_k d^k = f'(x^k).$$

  Dann ist d^k eine Abstiegsrichtung von f in x^k für alle $k \in \mathbb{N}$. Denn

  $$f'(x^k, d^k) \quad = \quad \left\langle f'(x^k), d^k \right\rangle = \left\langle f'(x^k), H_k^{-1} f'(x^k) \right\rangle > 0.$$

- *Offenbar ist $d^k := f'(x^k)$ eine Abstiegsrichtung von f im Punkte x^k. Die Richtung $-d^k$ wird Richtung des größten Gefälles genannt. Denn ist $v \in \mathbb{R}^n$ eine Abstiegsrichtung von f in $x \in \mathbb{R}^n$ mit $\|v\| = \|f'(x)\|$, dann gilt:*

  $$0 < f'(x, v) = \left\langle f'(x), v \right\rangle \leq \|f'(x)\| \|v\| = \left\langle f'(x), f'(x) \right\rangle = f'(x, f'(x)).$$

Wie schon erwähnt, ist es für eine nichtkonvexe Funktion f schwierig, ein globales Minimum zu ermitteln. Für solche Funktionen sollte daher die Minimierungsaufgabe zur Aufgabe: „Finde ein lokales Minimum x^*“ abgeschwächt werden. Ist $f''(x^*)$ verfügbar und positiv definit, so ist x^* ein lokales Minimum von f. Um die positive Definitheit zu überprüfen, ist die Kenntnis der Eigenwerte nötig, daneben ist auch die Existenz der Cholesky[6]-Zerlegung garantiert. Beide Methoden sind im Allgemeinen mit hohen Kosten von $O(n^3)$ verbunden. Wenn $f''(x^*)$ nicht positiv definit oder aus Kostengründen die positive Definitheit nicht zugänglich ist, kann diese Entscheidung im Allgemeinen nicht getroffen werden. In diesem Fall sollte man das Ziel haben, eine Abstiegsfolge zu erzeugen, welche gerade noch die Eigenschaft

$$\lim_{k \to \infty} x^k = x^* \quad \text{mit} \quad f'(x^*) = 0$$

erfüllt. Selbst diese Forderung muss in vielen Fällen noch weiter abgeschwächt werden.
Zum Nachweis der globalen Konvergenz sind Lemma 4.3.3 und die darauf folgende Definition 4.3.4 von zentraler Bedeutung.

[6] André-Louis Cholesky, französischer Mathematiker, ∗ 15. Oktober, Montguyon, † 31. August 1918 in Nordfrankreich.

Lemma 4.3.3. *Seien $\sigma, \beta \in (0,1)$ und $d \in \mathbb{R}^n$ eine Abstiegsrichtung von f in $x \in \mathbb{R}^n$, d.h. $\langle f'(x), d\rangle > 0$. Dann existiert ein $l \in \mathbb{N}$ mit:*

$$f(x) - f(x - \beta^l d) \geq \sigma\beta^l \left\langle f'(x), d\right\rangle. \tag{4.16}$$

Beweis. Angenommen, für alle $l \in \mathbb{N}$ gelte

$$f(x) - f(x - \beta^l d) < \sigma\beta^l \left\langle f'(x), d\right\rangle.$$

Dann ist auch

$$-\frac{f(x - \beta^l d) - f(x)}{\beta^l} < \sigma \left\langle f'(x), d\right\rangle.$$

Für $l \to \infty$ folgt dann:

$$\left\langle f'(x), d\right\rangle \leq \sigma \left\langle f'(x), d\right\rangle.$$

Wegen $1 - \sigma > 0$ folgt ferner:

$$\left\langle f'(x), d\right\rangle \leq 0.$$

Dies ist ein Widerspruch zur Voraussetzung $\langle f'(x), d\rangle > 0$. ■

Definition 4.3.4 (Gradientenähnliche Folge)**.** *Sei $(x^k)_{k\in\mathbb{N}}$ eine Folge in $\mathbb{R}^n$. Eine Folge $(d^k)_{k\in\mathbb{N}} \subseteq \mathbb{R}^n$ heißt* gradientenähnlich *bezüglich f und $(x^k)_{k\in\mathbb{N}}$, wenn für jede gegen einen nichtstationären Punkt von f konvergente Teilfolge $(x^j)_{j\in K}$, $K \subseteq \mathbb{N}$, Konstanten $c_1, c_2 \in \mathbb{R}_{>0}$ existieren, so dass*

$$\forall j \in K \quad : \quad \|d^j\| \leq c_1, \tag{4.17}$$

$$\exists j_0 \in K \, \forall j \in K_{\geq j_0} \quad : \quad c_2 \leq \left\langle f'(x^j), d^j\right\rangle \tag{4.18}$$

gelten.

Offensichtlich ist $(f'(x^k))_{k\in\mathbb{N}}$ gradientenähnlich bezüglich f und $(x^k)_{k\in\mathbb{N}}$, womit die Namensgebung gerechtfertigt ist.

Lemma 4.3.5. *Seien $p_1, p_2 \in \mathbb{R}_{\geq 0}$, $c_1, c_2 \in \mathbb{R}_{>0}$, $(x^k)_{k\in\mathbb{N}}$ und $(d^k)_{k\in\mathbb{N}}$ Folgen in $\mathbb{R}^n$, welche für alle $k \in \mathbb{N}$*

$$\|d^k\| \quad \leq \quad c_1\|f'(x^k)\|^{p_1}, \tag{4.19}$$

$$c_2\|f'(x^k)\|^{p_2} \quad \leq \quad \left\langle f'(x^k), d^k\right\rangle. \tag{4.20}$$

erfüllen. Dann ist $(d^k)_{k\in\mathbb{N}}$ gradientenähnlich bezüglich f und $(x^k)_{k\in\mathbb{N}}$.

Beweis. [9, Korollar 8.10, Seite 77.]. ■

Korollar 4.3.6. *Seien $\mu_1, \mu_2 \in \mathbb{R}_{>0}$, $(x^k)_{k\in\mathbb{N}}$, $(d^k)_{k\in\mathbb{N}}$ Folgen in $\mathbb{R}^n$ und $(H_k)_{k\in\mathbb{N}}$ eine Folge von symmetrischen und positiv definiten Matrizen im $\mathbb{R}^{n\times n}$, welche für alle $k \in \mathbb{N}$*

$$\forall u \in \mathbb{R}^n : \mu_1\|u\|^2 \leq \langle u, H_k u\rangle \leq \mu_2\|u\|^2 \tag{4.21}$$

und

$$H_k d^k = f'(x^k) \tag{4.22}$$

erfüllen. Dann ist $(d^k)_{k\in\mathbb{N}}$ gradientenähnlich bezüglich f und $(x^k)_{k\in\mathbb{N}}$.

Beweis. [9, Korollar 8.11, Seiten 77 f.] ■

Bemerkung 4.3.7. *Im Beweis von Korollar 4.3.6 wird in [9, Korollar 8.11, Seiten 77 f.] gezeigt, dass mit $p_1 = 1$ und $p_2 = 2$ die Voraussetzungen von Lemma 4.3.5 erfüllt sind, siehe [9, Seite 78, Zeilen 6 und 9]. Dieser Umstand wird für die Realisierung von Algorithmus 4.4.1 von Bedeutung sein, denn hierbei erhält man folgenden Spezialfall:*

$$\gamma_k := \frac{\left\langle f'(x^k), d^k \right\rangle}{\|f'(x^k)\| \|d^k\|} \geq \delta \in \mathbb{R}_{>0} \tag{4.23}$$

für alle $k \in \mathbb{N}$. Die Bedingung (4.23) wird Winkelbedingung genannt, da γ_k gleich dem Cosinus des von d^k und $f'(x^k)$ eingeschlossenen Winkels ist.

Mit Hilfe von Korollar 4.3.6 kann man gradientenähnliche Folgen erzeugen, so entsteht z.B. für $H_k := \mathbf{Id}$ das Gradientenverfahren. Daneben wird mit Hilfe von Lemma 4.3.5 die Gradientenähnlichkeit überprüft.

Algorithmus 4.3.1 Gradientenähnliches Verfahren

1: Wähle $x^1 \in \mathbb{R}^n$, $\varepsilon \in \mathbb{R}_{>0}$, $\sigma, \beta \in (0,1)$ und setzte $k := 1$.
2: **while** $\|f'(x^k)\| > \varepsilon$ **do**
3: Berechne $d^k \in \mathbb{R}^n$ mit $\left\langle f'(x^k), d^k \right\rangle > 0$, so dass d^k gradientenähnlich bezüglich f und x^k ist.
4: Bestimme $\alpha_k := \max\{\beta^l : l \in \mathbb{N}_{\geq 0}\}$ mit

$$\alpha_k := \max_{l \in \mathbb{N}_{\geq 0}} \left\{ \beta^l : f(x^k) - f(x^k - \beta^l d^k) \geq \sigma \beta^l \left\langle f'(x^k), d^k \right\rangle \right\}. \tag{4.24}$$

5: Setze $x^{k+1} := x^k - \alpha_k d^k$ und $k \leftarrow k + 1$.
6: **end while**

Bemerkung 4.3.8. *Sei $\sigma \in (0,1)$. Die durch*

$$\alpha_k := \max_{l \in \mathbb{N}_{\geq 0}} \left\{ \beta^l : f(x^k) - f(x^k - \beta^l d^k) \geq \sigma \beta^l \left\langle f'(x^k), d^k \right\rangle \right\}$$

für alle $k \in \mathbb{N}$ bestimmte Schrittweite wird mittels Armijo-Regel berechnete Schrittweite genannt, siehe [1]. Die Armijo-Regel ist wegen Lemma 4.3.3 wohldefiniert.

Satz 4.3.9. *Seien $(x^k)_{k \in \mathbb{N}}$, $(d^k)_{k \in \mathbb{N}}$ und $(\alpha_k)_{k \in \mathbb{N}}$ mittels Algorithmus 4.3.1 erzeugt. Dann ist jeder Häufungspunkt $x^* \in \mathbb{R}^n$ von $(x^k)_{k \in \mathbb{N}}$ ein stationärer Punkt von f.*

Beweis. Ähnlich zu [9, Satz 8.9, Seiten 76 f.]. Angenommen, es existiere ein Häufungspunkt $x^* \in \mathbb{R}^n$ von $(x^k)_{k \in \mathbb{N}}$ mit $f'(x^*) \neq 0$. Ferner sei $(x^j)_{j \in K}$ eine gegen x^* konvergente Teilfolge von $(x^k)_{k \in \mathbb{N}}$. Da $(f(x^k))_{k \in \mathbb{N}}$ monoton fällt und die Teilfolge $(f(x^j))_{j \in K}$, $K \subset \mathbb{N}$, gegen $f(x^*)$ konvergiert, ist bereits die gesamte Folge $(f(x^k))_{k \in \mathbb{N}}$ gegen $f(x^*)$ konvergent. Daher ist

$$f(x^k) - f(x^{k+1}) \xrightarrow[k \to \infty]{} 0.$$

Mit Gleichung (4.24) folgt somit

$$\underbrace{\alpha_k \left\langle f'(x^k), d^k \right\rangle}_{>0} \xrightarrow[k\to\infty]{} 0. \tag{4.25}$$

Da $(d^k)_{k\in\mathbb{N}}$ nach Voraussetzung gradientenähnlich bezüglich f und $(x^k)_{k\in\mathbb{N}}$ ist, existieren Konstanten $c_1, c_2 \in \mathbb{R}_{>0}$ mit

$$\forall j \in K \quad : \quad \|d^j\| \leq c_1, \tag{4.26}$$

$$\exists j_0 \in K : \forall j \in K_{\geq j_0} \quad : \quad c_2 \leq \left\langle f'(x^j), d^j \right\rangle. \tag{4.27}$$

Aus (4.25) und (4.27) folgt unmittelbar $\alpha_j \xrightarrow[j\to\infty]{} 0$. Weiter folgt dann aus (4.24)

$$f(x^j) - f(x^j - \beta^{(l_j-1)} d^j) < \sigma \beta^{(l_j-1)} \left\langle f'(x^j), d^j \right\rangle$$

für alle hinreichend großen $j \in K$ (wobei $\alpha_j = \beta^{l_j}$). Hieraus erhält man

$$\frac{f(x^j) - f(x^j - \beta^{(l_j-1)} d^j)}{\beta^{(l_j-1)}} < \sigma \left\langle f'(x^j), d^j \right\rangle.$$

Wegen (4.26) konvergiert $(d^j)_{j\in K}$ ggf. nach Übergang zu einer weiteren Teilfolge gegen ein $d^* \in \mathbb{R}^n$. Da $(\beta^{(l_j-1)})_{j\in K}$ eine Nullfolge ist, folgt für $j \to \infty$

$$(1-\sigma) \left\langle f'(x^*), d^* \right\rangle \leq 0.$$

Widerspruch, denn $1 - \sigma > 0$ und $\langle f'(x^*), d^* \rangle \geq c_2 > 0$. ■

Algorithmus 4.3.2 Globalisiertes Newton-ähnliches Verfahren

1: Wähle $x^1 \in \mathbb{R}^n$, $\varepsilon \in \mathbb{R}_{>0}$, $\beta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$ $c \in \mathbb{R}_{>0}$, $p \in \mathbb{R}_{>2}$ und setze $k := 1$.
2: **while** $\|f'(x^k)\| > \varepsilon$ **do**
3: Wähle ein $\eta_k \in \mathbb{R}_{\geq 0}$ und bestimme $d^k \in \mathbb{R}^n$ mit

$$\|f'(x^k) - f''(x^k) d^k\| \leq \eta_k \|f'(x^k)\|.$$

Ist dies nicht möglich oder ist die Bedingung

$$c\|f'(x^k)\|^p \leq \left\langle f'(x^k), d^k \right\rangle$$

nicht erfüllt, so setze $d^k := f'(x^k)$.
4: Bestimme

$$\alpha_k := \max_{l\in\mathbb{N}_{\geq 0}} \left\{ \beta^l : f(x^k) - f(x^k - \beta^l d^k) \geq \sigma \beta^l \left\langle f'(x^k), d^k \right\rangle \right\}.$$

5: Setze $x^{k+1} := x^k - \alpha_k d^k$ und $k \leftarrow k+1$.
6: **end while**

Wie eingangs erwähnt, wird nun ein schnelles und global konvergentes Verfahren vorgestellt, wobei für die schnelle Konvergenz Lemma 4.3.10 von Bedeutung ist. Für nichtkonvexe Funktionen f (bzw. konvexe Funktionen, für die f'' nicht in allen Punkten positiv definit ist), wird durch $d^k = \left(f''(x^k)\right)^{-1} f'(x^k)$ nicht unbedingt eine Abstiegsrichtung erklärt. Die einfachste Wahl einer Ersatzabstiegsrichtung ist die Gradientenrichtung. Die Konvergenz für das daraus resultierende Verfahren kann dann erzwungen werden. Ist die Iterationsfolge gegen eine reguläre lokale Minimallösung x^* konvergent, so wird man es lokal um x^* mit einer stark konvexen Funktion zu tun haben und das in Algorithmus 4.3.2 vorliegende Verfahren geht in das eigentliche Newton-ähnliche Verfahren über.

Lemma 4.3.10. *Seien $x^* \in \mathbb{R}^n$ ein stationärer Punkt von f, $f''(x^*)$ positiv definit, $(x^k)_{k\in\mathbb{N}}$ eine gegen x^* konvergente Folge und $(d^k)_{k\in\mathbb{N}}$ eine Folge Newton-ähnlicher Richtungen. Dann existiert ein $k_0 \in \mathbb{N}$, so dass für alle $k \in \mathbb{N}_{\geq k_0}$ und jedes $\sigma \in (0, \frac{1}{2})$*

$$f(x^k) - f(x^k - d^k) \leq \sigma \left\langle f'(x^k), d^k \right\rangle \tag{4.28}$$

gilt. D.h., unter den gegebenen Voraussetzungen wird für $x^{k+1} = x^k - \alpha_k d^k$ die volle Schrittweite ($\alpha_k = 1$) nach endlich vielen Schritten akzeptiert, wobei die Folge $(x^k)_{k\in\mathbb{N}}$ wie in Gleichung (4.13) definiert ist.

Beweis. [9, Satz 10.6, Seiten 114 ff.]. ■

Satz 4.3.11. *Sei $(x^k)_{k\in\mathbb{N}}$ eine durch Algorithmus 4.3.2 erzeugte Folge mit $\eta_k \xrightarrow[k\to\infty]{} 0$. Ist x^* ein Häufungspunkt von $(x^k)_{k\in\mathbb{N}}$ mit $f''(x^*)$ positiv definit, so gelten die folgenden Aussagen:*

- *Die gesamte Folge $(x^k)_{k\in\mathbb{N}}$ konvergiert gegen x^* und x^* ist ein striktes lokales Minimum von f.*
- *Nach endlich vielen Schritten wird die Newton-ähnliche Richtung und die volle Schrittweite, $\alpha_k = 1$, akzeptiert.*
- *Die Folge $(x^k)_{k\in\mathbb{N}}$ konvergiert superlinear gegen x^*.*
- *Ist $f''(x^*)$ lokal Lipschitz-stetig und $\eta_k = O(\|f'(x^k)\|)$, so konvergiert die Folge $(x^k)_{k\in\mathbb{N}}$ quadratisch gegen x^*.*

Beweis. [9, Satz 10.8, Seiten 117 f.]. ■

4.4 Verfahren mit parameterabhängiger Abstiegsrichtung zur Minimierung nichtkonvexer Funktionen

Alle vorgestellten Verfahren fügen sich in das allgemeine Schema eines Abstiegsverfahrens mit Schrittweitenstrategie ein, d.h. ausgehend von einem gegebenen Iterationspunkt $x^k \in \mathbb{R}^n$ ($k \in \mathbb{N}$) bestimmt man zunächst eine Abstiegsrichtung

$d^k \in \mathbb{R}^n$ von f in x^k und anschließend wird auf dem Strahl $\{x^k - \alpha d^k : \alpha \in \mathbb{R}_{\geq 0}\}$, der durch x^k und d^k festgelegt ist, ein Nachfolger x^{k+1} mit gegenüber $f(x^k)$ genügend verkleinertem Funktionswert bestimmt. Die Suchrichtung d^k ergab sich dabei als Lösung der unrestringierten quadratischen Optimierungsaufgabe

$$\min_{d \in \mathbb{R}^n} q_k(d), \tag{4.29}$$

wobei

$$q_k(d) := f(x^k) + \left\langle f'(x^k), d \right\rangle + \frac{1}{2} \langle H_k d, d \rangle \tag{4.30}$$

eine quadratische Approximation von f an der Stelle x^k mit einer symmetrischen und möglichst positiv definiten Matrix H_k darstellt. Hierbei ist $H_k = f''(x^k)$ beim Newton-Verfahren, $H_k \approx f''(x^k)$ im Sinne von Definition 4.2.2 bei Newton-ähnlichen Verfahren und $H_k = \mathbf{Id}$ beim Gradientenverfahren. Nun wird die quadratische Approximation q_k die nichtlineare Funktion im Allgemeinen nur lokal gut annähern. Dennoch wird als Suchrichtung d^k genommen und im Übrigen noch eine Liniensuche nachgeschaltet. Eine Erweiterung dieses Prinzips besteht darin, den Nachfolger x^{k+1} auf einer nicht notwendig geradlinig verlaufenden Raumkurve zu suchen. Hierbei wird die Berechnung der Abstiegsrichtung direkt an die Schrittweite gekoppelt; eine anschließende Schrittweitenberechnung ist dann entbehrlich. Verfahren dieser Art sind besonders für den nichtkonvexen Fall geeignet, siehe [8, Seiten 257 ff.] oder [37, Seiten 247 ff.]. Aber die Kopplung der Schrittweite an das Berechnen der Abstiegsrichtung besitzt einen evidenten Nachteil, denn nach der Berechnung des Nachfolgers x^{k+1} muss dieser bewertet werden. Ist der Nachfolger nicht zulässig, so wird er verworfen und ein neues lineares Gleichungssystem gelöst, was hohe numerische Berechnungskosten verursacht, siehe dazu „Inexaktes Trust-Region-Newton-Verfahren“, [8, Algorithmus 14.37, Seiten 304 f.], in den Zeilen (S.3) und (S.4) sowie „Regularisiertes Verfahren vom Newton-Typ“, [37, Verfahren 8.2.1, Seiten 248 f.], im 4. und 6. Schritt. Dieser Nachteil kann in mancher Hinsicht behoben werden.

Lemma 4.4.1. *Seien $a \in \mathbb{R}$, $0 \neq b \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{n \times n}$ positiv definit und symmetrisch und*

$$d \mapsto q(d) := a + \langle b, d - x \rangle + \frac{1}{2} \langle B(d - x), d - x \rangle \tag{4.31}$$

für alle $d \in \mathbb{R}^n$. Dann ist für jedes $r \in \left(0, \|B^{-1}b\|_A\right)$ die eindeutige Minimallösung d^* von q auf der abgeschlossenen Kugel $\overline{K}_A(x, r) := \{d \in \mathbb{R}^n : \|d - x\|_A \leq r\}$ von der Gestalt

$$d^* = x - (\lambda A + B)^{-1} b, \tag{4.32}$$

wobei $\lambda \in \mathbb{R}_{>0}$ als Funktion von r eindeutig durch die nichtlineare eindimensionale Gleichung

$$\varphi(\lambda) := \|(\lambda A + B)^{-1} f'(x^k)\|_A = r \tag{4.33}$$

bestimmt ist.

Beweis. Seien $\lambda \in \mathbb{R}_{>0}$, $r \in \left(0, \|B^{-1}b\|_A\right)$ und $d \in \mathbb{R}^n$. Die Nebenbedingung $\|d-x\|_A^2 \leq r^2$ liefert die folgende Lagrange[7]-Funktion:

$$L_\lambda(d) := q(d) + \frac{1}{2}\lambda\left(\|d-x\|_A^2 - r^2\right).$$

L_λ ist konvex, denn $\lambda > 0$ und B ist positiv definit. Damit ist d^* genau dann eine Minimallösung von L_λ auf $\mathbb{R}^n$, wenn

$$0 = L'_\lambda(d^*) = q'(d^*) + \lambda A(d^*-x) = b + B(d^*-x) + \lambda A(d^*-x)$$

bzw.

$$d^* = x - (\lambda A + B)^{-1} b$$

gilt.
Da $\varphi(\lambda) \xrightarrow[\lambda\to 0]{} \|B^{-1}b\|_A$ und mit der Substitution $\lambda = \frac{1-\alpha}{\alpha}$, wobei $\alpha \in (0,1)$, $\varphi(\lambda) \xrightarrow[\lambda\to\infty]{} 0$ folgt, besitzt die Gleichung (4.33) eine Lösung in $\mathbb{R}_{>0}$. Zu zeigen bleibt die Eindeutigkeit. Es gilt:

$$\begin{aligned}
\varphi^2(\lambda) &= \|(\lambda A + B)^{-1}b\|_A^2 \\
&= \|A^{-\frac{1}{2}}(\lambda \mathbf{Id} + A^{-\frac{1}{2}}BA^{-\frac{1}{2}})^{-1}A^{-\frac{1}{2}}b\|_A^2 \\
&= \|A^{-\frac{1}{2}}(\lambda \mathbf{Id} + \tilde{B})^{-1}\tilde{b}\|_A^2 \\
&= \|(\lambda \mathbf{Id} + \tilde{B})^{-1}\tilde{b}\|^2,
\end{aligned}$$

wobei $\tilde{B} := A^{-\frac{1}{2}}BA^{-\frac{1}{2}}$ und $\tilde{b} := A^{-\frac{1}{2}}b$ gesetzt wurde. Offenbar ist $\tilde{B}$ symmetrisch und wegen [15, Bemerkung 2.10.7, Seite 53] hat $\tilde{B}$ nur positive Eigenwerte. Damit ist $\tilde{B}$ insgesamt positiv definit. Demnach existieren $U \in \mathbb{R}^{n\times n}$ (orthogonal, d.h. $U^tU = UU^t = \mathbf{Id}$) und $D \in \mathbb{R}^{n\times n}$ (diagonal), so dass $\tilde{B} = UDU^t$ und D nur positive Diagonaleinträge $d_i \in \mathbb{R}_{>0}$ für alle $i \in \mathbb{N}_{\leq n}$ hat. Ferner gilt mit $\hat{b} := U^t\tilde{b}$:

$$\begin{aligned}
\varphi^2(\lambda) &= \|U\left(\lambda \mathbf{Id} + D\right)^{-1}U^t\tilde{b}\|^2 \\
&= \|\left(\lambda \mathbf{Id} + D\right)^{-1}\hat{b}\|^2 \\
&= \sum_{i=1}^{n}(\lambda + d_i)^{-2}\hat{b}_i^2.
\end{aligned}$$

Daraus folgt

$$\begin{aligned}
(\varphi^2)'(\lambda) &= -2\sum_{i=1}^{n}\underbrace{(\lambda + d_i)^{-3}}_{>0}\hat{b}_i^2 \\
&< 0,
\end{aligned}$$

denn $\hat{b} \neq 0$. Damit ist φ eine streng monoton fallende Funktion, also injektiv. ■

[7] Joseph Louis Lagrange, italienischer Mathematiker und Astronom, ∗ 25. Januar 1736 in Turin, † 10. April 1813 in Paris.

Die nun folgende Herangehensweise stellt eine leichte Abwandlung des „Verfahrens N3“, [24, Seite 140], und des „Modifizierten regularisierten Verfahrens vom Newton-Typ“, [37, Verfahren 8.2.6, Seite 251], dar. Wenn auch die wesentlichen Ideen identisch sind, führt dieses modifizierte Verfahren zu besseren Konvergenzergebnissen und ist speziell auf das im Zentrum dieser Arbeit stehende Minimierungsproblem zugeschnitten.
Seien im Folgenden $c_1, c_2 \in \mathbb{R}_{>0}$ und $(A_k)_{k\in\mathbb{N}}$ eine Folge von symmetrischen und positiv definiten Matrizen im $\mathbb{R}^{n\times n}$, welche für alle $k \in \mathbb{N}$ folgende Bedingungen erfüllen:

- Für alle $u \in \mathbb{R}^n$ gilt:

$$c_1\|u\|^2 \leq \|u\|_{A_k}^2 \leq c_2\|u\|^2, \tag{4.34}$$

  wobei

$$\|u\|_{A_k}^2 := \langle A_k u, u\rangle \tag{4.35}$$

  ist.

- Lineare Gleichungen mit A_k sind leicht aufzulösen.

In der Praxis werden geeignete positiv definite Anteile von $f''(x^k)$ zur Generierung der A_k benutzt.

Bemerkung 4.4.2. *Bei den alternativen Verfahren von [8, Seiten 257 ff.], [24, Seiten 138 ff.] oder [37, Seiten 247 ff.] ist $A_k = \mathbf{Id}$ für alle $k \in \mathbb{N}$ gewählt.*

Ausgangspunkt der nachfolgenden Betrachtung ist eine symmetrische Approximation H_k von $f''(x^k)$, $k \in \mathbb{N}$, welche zunächst als positiv definit vorausgesetzt wird. Mit deren Hilfe bildet man das quadratische Approximationspolynom q_k aus Gleichung (4.30) an der Stelle x^k. Bei vorgegebenem $r \in \mathbb{R}_{>0}$ bestimmt man nun $x^{k+1}(r)$ als Lösung der Aufgabe

$$q_k(x^{k+1}(r)) = \min\{q_k(x) : \|x - x^k\|_{A_k} \leq r\}, \tag{4.36}$$

d.h. $x^{k+1}(r)$ ist Minimum von q_k auf einer abgeschlossenen Kugel um x^k mit dem Radius r. Angesichts der Einschränkung auf einen gewissen Vertrauensbereich werden Methoden dieses Typs auch Trust-Region-Verfahren genannt, siehe z.B. [8, Seiten 255 ff.].
Zu jedem r existiert genau eine Lösung $x^{k+1}(r)$ von (4.36), denn für $r \geq r_k := \|H_k^{-1}f'(x^k)\|_{A_k}$ ist $x^{k+1}(r) = x^k - H_k^{-1}f'(x^k)$. Diese Tatsache folgt aus Gleichung (4.6) und $\|x^{k+1}(r) - x^k\|_{A_k} = \|H_k^{-1}f'(x^k)\|_{A_k} = r_k \leq r$. Für $r \in (0, r_k)$ ist $x^{k+1}(r)$ von der Gestalt

$$x^{k+1}(r) = x^k - (\lambda A_k + H_k)^{-1}f'(x^k), \tag{4.37}$$

wobei $\lambda \in \mathbb{R}_{>0}$ als Funktion von r eindeutig durch die nichtlineare eindimensionale Gleichung

$$\varphi(\lambda) := \|(\lambda A_k + H_k)^{-1}f'(x^k)\|_{A_k} = r$$

bestimmt ist, denn es gilt das Lemma 4.4.1. Mit der Substitution $\alpha = (1-\lambda)/\lambda$ lässt sich der λ-Bereich $(0,\infty)$ in das α-Intervall $(0,1)$ überführen. Damit geht Gleichung (4.37) in

$$x^{k+1}(r) = x^k - \alpha d^k(\alpha) \tag{4.38}$$

mit

$$d^k(\alpha) = ((1-\alpha)A_k + \alpha H_k)^{-1} f'(x^k) \tag{4.39}$$

über. Die hier vorkommende Matrix

$$C_k(\alpha) := (1-\alpha)A_k + \alpha H_k \tag{4.40}$$

ist im Fall der positiven Definitheit von H_k für alle $\alpha \in [0,1]$ positiv definit, so dass $d^k(\alpha)$ gebildet werden kann und eine Abstiegsrichtung ist. Für $\alpha = 1$ ergibt sich die approximierte Newton-Richtung $d^k(1) = H_k^{-1} f'(x^k)$, dagegen erhält man für $\alpha = 0$ die Richtung $d^k(0) = A_k^{-1} f'(x^k)$. In diesem Sinne ist die Richtung $d^k(\alpha)$ eine Kombination von diesen beiden Basisrichtungen, welche durch den nichtlinear auftretenden Parameter α gesteuert wird.
Falls H_k nicht positiv definit ist, so wird $C_k(\alpha)$ für bestimmte Werte von $\alpha \in [0,1]$ singulär, womit dann $d^k(\alpha)$ nicht definiert ist. Daneben braucht $d^k(\alpha)$ auch für reguläres $C_k(\alpha)$ keine gradientenähnliche Abstiegsrichtung mehr sein. Für genügend kleines α kann dies allerdings nicht eintreten, denn wegen der regularisierenden Wirkung von $(1-\alpha)A_k$ ist dann $C_k(\alpha)$ sicher positiv definit. Dieser Sachverhalt motiviert die folgende Strategie. Es wird zunächst versucht, mit $\alpha_k = 1$ bzw. genügend großen α_k-Werten ein positiv definites $C_k(\alpha_k)$ und zugehöriges $x^{k+1}(\alpha_k)$ mit hinreichend gutem Abstieg zu berechnen. Ist dies nicht möglich, so wird α_k fortlaufend verkleinert, wobei $d^k(\alpha_k)$ der Richtung $d^k(0)$ immer näher kommt. Eine mögliche Realisierung dieses Prinzips stellt der Algorithmus „Regularisiertes Verfahren vom Newton-Typ“, [37, Verfahren 8.2.1, Seiten 248 f.] dar, wobei $A_k = \mathbf{Id}$ für alle $k \in \mathbb{N}$ ist. Wie eingangs erwähnt, hat diese Vorgehensweise den Nachteil, dass beim Zurückweisen der Abstiegsrichtung ein neues Gleichungssystem gelöst werden muss. Dieses Manko kann teilweise abgebaut werden, wenn zunächst ein α_k mit positiv definitem $C_k(\alpha_k)$ bestimmt und ein Nachfolger in der Form $x^{k+1} := x^k - \bar{\alpha}_k d^k(\alpha_k)$ mit $\bar{\alpha}_k \neq \alpha_k$ gesucht wird. Dies führt zu dem nachstehenden Algorithmus 4.4.1, Seite 61.

Satz 4.4.3. *Sei $(x^k)_{k\in\mathbb{N}}$ eine durch Algorithmus 4.4.1 erzeugte Folge, wobei $\|H_k - f''(x^k)\| \xrightarrow[k\to\infty]{} 0$ gilt. Dann ist jeder Häufungspunkt $x^* \in \mathbb{R}^n$ von $(x^k)_{k\in\mathbb{N}}$ ein stationärer Punkt von f. Ist ferner $x^* \in \mathbb{R}^n$ ein Häufungspunkt von $(x^k)_{k\in\mathbb{N}}$ mit $f''(x^*)$ positiv definit, so konvergiert die gesamte Folge $(x^k)_{k\in\mathbb{N}}$ gegen x^* und x^* ist ein striktes lokales Minimum von f. Daneben wird nach endlich vielen Schritten die Newton-ähnliche Richtung angenommen und für die Schrittweite gilt $\alpha_k = \bar{\alpha}_k = 1$. Die Folge $(x^k)_{k\in\mathbb{N}}$ konvergiert dann superlinear gegen x^*. Ist ferner $f''(x^*)$ lokal Lipschitz-stetig, so konvergiert die Folge $(x^k)_{k\in\mathbb{N}}$ quadratisch gegen x^*.*

Beweis. Aus $C_k(\alpha_k) \xrightarrow[\alpha_k\to 0]{} A_k$ folgt mit Korollar 4.3.6 und Bemerkung 4.3.7 die Realisierbarkeit von Algorithmus 4.4.1. Außerdem folgt dann mit Satz 4.3.9,

dass jeder Häufungspunkt $x^* \in \mathbb{R}^n$ von $(x^k)_{k\in\mathbb{N}}$ ein stationärer Punkt von f ist, denn $(d^k)_{k\in\mathbb{N}}$ ist gradientenähnlich bezüglich f und $(x^k)_{k\in\mathbb{N}}$. Der Rest der Behauptung folgt analog zu [9, Satz 10.8, Seiten 117 f.]. ■

Algorithmus 4.4.1 Globalisiertes Verfahren mit parameterabhängiger Suchrichtung

1: Wähle $x^1 \in \mathbb{R}^n$, $\gamma, \beta \in (0,1)$, $\varepsilon \in \mathbb{R}_{>0}$, $\sigma \in (0, \frac{1}{2})$ $\delta \in \mathbb{R}_{>0}$, $p \in \mathbb{R}_{\geq 2}$, setze $k := 1$ und $\alpha_0 := 1$.
2: **while** $\|f'(x^k)\| > \varepsilon$ **do**
3: $\alpha_k := \min\left\{\frac{\alpha_{k-1}}{\gamma}, 1\right\}$.
4: Bestimme $d^k \in \mathbb{R}^n$ mit

$$C_k(\alpha_k)d^k = f'(x^k), \tag{4.41}$$

wobei

$$C_k(\alpha_k) := (1-\alpha_k)A_k + \alpha_k H_k \tag{4.42}$$

gesetzt ist.
Ist dies nicht möglich oder ist die Bedingung

$$\frac{\left\langle f'(x^k), d^k\right\rangle}{\|f'(x^k)\|\|d^k\|} \geq \min\{\delta, \|f'(x^k)\|^2\}$$

nicht erfüllt, so setze $\alpha_k := \gamma\alpha_k$ und gehe zu 4.
5: Bestimme

$$\bar{\alpha}_k := \max_{l\in\mathbb{N}_{\geq 0}}\left\{\beta^l : f(x^k) - f(x^k - \beta^l d^k) \geq \sigma\beta^l\left\langle f'(x^k), d^k\right\rangle\right\}.$$

6: Setze $x^{k+1} := x^k - \bar{\alpha}_k d^k$ und $k \leftarrow k+1$.
7: **end while**

5 Lösung der Approximationsaufgabe

Im gesamten Kapitel seien $d \in \mathbb{N}_{\geq 3}$, $\mathcal{S} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}$ und $t_\mu \in \mathbb{N}$ für alle $\mu \in \mathbb{N}_{\leq d}$. Darüber hinaus wird die Vereinbarung getroffen, dass die Bezeichnungen und Definitionen von Kapitel 3 gelten, insbesondere sind die Notation der Zielfunktion f und deren Anteile (f_1, g_1 und g_2) sowie des Gradienten und der Hesse-Matrix als auch deren Teilglieder mit eingeschlossen, siehe Seiten 37 ff.
Im Anschluss wird die Lösung der Approximationsaufgaben 3.1.1 und 3.1.2 beschrieben, wobei zuerst die Aufgabe 3.1.1 diskutiert wird. Aus diesem Grund seien $R \in \mathbb{N}$ und zunächst $r \in \mathbb{N}_{<R}$ fest gewählt. Algorithmus 4.4.1, Seite 61, wird auf die Zielfunktion f aus Gleichung (3.15), Seite 40, und eine gegebene Elementartensor-Summe, bezeichnet mit

$$\alpha := \sum_{i=1}^{R} \underbrace{\bigotimes_{\mu=1}^{d} \alpha_{i\mu}}_{\alpha_i :=} \in \mathcal{S}_R, \tag{5.1}$$

angewendet. Die Folgenglieder der hierdurch erzeugten Folge

$$(\xi^k)_{k\in\mathbb{N}} \subset \mathcal{T}_r \tag{5.2}$$

von Elementartensor-Summen mit beschränkten Summanden werden im Folgenden mit

$$\xi^k := \sum_{j=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} \xi_{j\mu}^k}_{\xi_j^k :=} \tag{5.3}$$

bezeichnet, wobei $k \in \mathbb{N}$ immer der Iterationsindex des Algorithmus 4.4.1 ist. Ohne Beschränkung der Allgemeinheit kann man annehmen, dass

$$\|\alpha\| = 1 \tag{5.4}$$

gilt, andernfalls normiert man α und multipliziert im Anschluss des Verfahrens den Grenzwert von $(\xi^k)_{k\in\mathbb{N}}$ mit der Norm von α. Mit dieser Rechtfertigung werden die Konstanten der Zielfunktion, z.B. $\frac{1}{\|\alpha\|^2}$, bei der nachstehenden Untersuchung nicht weiter berücksichtigt. Ferner kann man annehmen, dass die

Nebenbedingung (3.7), Seite 38, für jedes ξ^k erfüllt ist, denn ein nachträgliches Reskalieren ändert die Funktionswerte von f_1 und g_2 nicht, d.h. es gilt

$$\forall j \in \mathbb{N}_{\leq r} \forall \mu \in \mathbb{N}_{\leq d} \forall \nu \in \mathbb{N}_{\leq d} \setminus \{\mu\} : \|{\xi^k}_{j\mu}\| = \|{\xi^k}_{j\nu}\|, \tag{5.5}$$

für alle $k \in \mathbb{N}$.

5.1 Festlegung der Blockstruktur

Entsprechend Abschnitt 3.3 bzw. Abschnitt 3.4 hat der Gradient bzw. die Hesse-Matrix der Zielfunktion f eine Blockstruktur, welche durch die Komponenten von $f'(\hat{\xi}^k)$ bzw. den Segmenten von $f''(\hat{\xi}^k)$ vorgegeben ist. Daneben kann man die Blockstufung über $\mu_1 \in \mathbb{N}_{\leq d}$ bzw. $j_1 \in \mathbb{N}_{\leq r}$ umkehren, siehe zur Anschauung Abbildung 5.1.
Daraus resultieren zwei unterschiedliche Blockstufungen, welche natürlich aus

$$\hat{\xi}_1 = \begin{pmatrix} \left.\begin{pmatrix} \xi_{11} \\ \vdots \\ \xi_{1d} \end{pmatrix}\right\} & j_1 = 1 \\ \vdots & \\ \left.\begin{pmatrix} \xi_{r1} \\ \vdots \\ \xi_{rd} \end{pmatrix}\right\} & j_1 = r \end{pmatrix} \qquad \hat{\xi}_2 = \begin{pmatrix} \left.\begin{pmatrix} \xi_{11} \\ \vdots \\ \xi_{r1} \end{pmatrix}\right\} & \mu_1 = 1 \\ \vdots & \\ \left.\begin{pmatrix} \xi_{1d} \\ \vdots \\ \xi_{rd} \end{pmatrix}\right\} & \mu_1 = d \end{pmatrix}.$$

(a) $\hat{\xi}_1 \in \times_{j=1}^{r} \left(\times_{\mu=1}^{d} \mathbb{R}^{t_\mu} \right)$. (b) $\hat{\xi}_2 \in \times_{\mu=1}^{d} \left(\times_{j=1}^{r} \mathbb{R}^{t_\mu} \right)$.

Abbildung 5.1: Darstellung der zwei unterschiedlichen Blockstrukturen, zum einen bezüglich dir lexikographischen Anordnung von (j_1, μ_1), siehe (a), und zum anderen bezüglich (μ_1, j_1), siehe (b).

theoretischer Sicht äquivalent sind. Bei der praktischen Umsetzung werden BLAS- und LAPACK-Routinen verwendet, daher weist die nun folgende Anordnung den Vorteil einer bezüglich Speicherverwendung konformen Darstellung auf, so dass beim Aufruf der oben genannten Routinen kein zusätzlicher Speicher verwendet wird. Darüber hinaus entfallen unnötige Kopieroperationen.

Notation 5.1.1 (Blockstruktur). *Seien $\mathfrak{R}_{d,r,\underline{t}} = \times_{\mu=1}^{d} \left(\times_{j=1}^{r} \mathbb{R}^{t_\mu} \right)$ wie in Notation 3.2.1, Seite 38, $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$. Ein $x := (x_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$ wird bezüglich (μ_1, j_1) lexikographisch angeordnet,*

d.h.

$$x = \begin{pmatrix} \left.\begin{pmatrix} x_{11} \\ \vdots \\ x_{r1} \end{pmatrix}\right\} & \mu_1 = 1 \\ \vdots & \\ \left.\begin{pmatrix} x_{1\mu} \\ \vdots \\ x_{r\mu} \end{pmatrix}\right\} & \mu_1 = \mu \\ \vdots & \\ \left.\begin{pmatrix} x_{1d} \\ \vdots \\ x_{rd} \end{pmatrix}\right\} & \mu_1 = d \end{pmatrix}. \tag{5.6}$$

Natürlich kann man wegen der Isometrie von $\mathfrak{R}_{d,r,\underline{t}}$ und $\mathbb{R}^{r|\underline{t}|}$, $|\underline{t}| := \sum_{\mu=1}^{d} t_\mu$, x auch eindeutig mit einem Element aus $\mathbb{R}^{r|\underline{t}|}$ identifizieren; man setzt dann

$$x = \sum_{\mu=1}^{d} e_\mu^{(d)} \otimes \left(\sum_{j=1}^{r} e_j^{(r)} \otimes x_{j\mu} \right), \tag{5.7}$$

oder schreibt kürzer

$$x = \sum_{\mu=1}^{d} \sum_{j=1}^{r} e_\mu \otimes e_j \otimes x_{j\mu},$$

wobei $\left\{ e_\mu^{(d)} \in \mathbb{R}^d : \mu \in \mathbb{N}_{\leq d} \right\}$ bzw. $\left\{ e_j^{(r)} \in \mathbb{R}^r : j \in \mathbb{N}_{\leq r} \right\}$ die kanonischen Basen von $\mathbb{R}^d$ bzw. $\mathbb{R}^r$ sind.
Daneben sei $x_\mu \in \mathbb{R}^{t_\mu \times r}$ für alle $\mu \in \mathbb{N}_{\leq d}$ wie folgt definiert:

$$x_\mu := (x_{1\mu}, x_{2\mu}, \ldots, x_{r\mu}). \tag{5.8}$$

Für $h \in \{f, f_1, g_1, g_2\}$ ist $h''(\hat{\xi})$ eine lineare Abbildung von $\mathfrak{R}_{d,r,\underline{t}}$ nach $\mathfrak{R}_{d,r,\underline{t}}$. Es wird vereinbart, dass die Segmente von $h''(\hat{\xi})$ derart angeordnet werden, dass die durch Gleichung (5.7) definierte Ordnung erhalten bleibt, d.h., sind $H_{\mu_1\mu_2 j_1 j_2} := h''(\hat{\xi})_{\mu_1\mu_2 j_1 j_2}$ die Segmente von $h''(\hat{\xi})$, dann ist die Hesse-Matrix von h an der Stelle $\hat{\xi}$ wie folgt angeordnet:

$$\begin{aligned} h''(\hat{\xi}) &= \begin{pmatrix} \begin{pmatrix} H_{1111} & \ldots & H_{111r} \\ \vdots & \ddots & \vdots \\ H_{111r} & \ldots & H_{11rr} \end{pmatrix} & \cdots & \begin{pmatrix} H_{1d11} & \ldots & H_{1d1r} \\ \vdots & \ddots & \vdots \\ H_{1d1r} & \ldots & H_{1drr} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ \begin{pmatrix} H_{d111} & \ldots & H_{d11r} \\ \vdots & \ddots & \vdots \\ H_{d11r} & \ldots & H_{d1rr} \end{pmatrix} & \cdots & \begin{pmatrix} H_{dd11} & \ldots & H_{dd1r} \\ \vdots & \ddots & \vdots \\ H_{dd1r} & \ldots & H_{ddrr} \end{pmatrix} \end{pmatrix} \\ &= \sum_{\mu_1,\mu_2=1}^{d} \mathbb{E}_{\mu_1\mu_2}^{(d)} \otimes \left(\sum_{j_1,j_2=1}^{r} \mathbb{E}_{j_1 j_2}^{(r)} \otimes H_{\mu_1\mu_2 j_1 j_2} \right), \end{aligned}$$

hierbei sind $\{\mathbb{E}^{(d)}_{\mu_1\mu_2} \in \mathbb{R}^{d\times d} : \mu_1, \mu_2 \in \mathbb{N}_{\le d}\}$ *bzw.* $\{\mathbb{E}^{(r)}_{j_1 j_2} \in \mathbb{R}^{r\times r} : j_1, j_2 \in \mathbb{N}_{\le r}\}$ *die kanonische Basen von* $\mathbb{R}^{d\times d}$ *bzw.* $\mathbb{R}^{r\times r}$*. Ist dies aus dem Zusammenhang klar, dann verwendet man kürzer*

$$h''(\hat{\xi}) = \sum_{\mu_1,\mu_2=1}^{d} \sum_{j_1,j_2=1}^{r} \mathbb{E}_{\mu_1\mu_2} \otimes \mathbb{E}_{j_1 j_2} \otimes H_{\mu_1\mu_2 j_1 j_2}.$$

5.2 Analyse ausgewählter Teile der Hesse-Matrix

Bei der numerischen Lösung der Minimierungsaufgabe 3.1.1 hat die Matrix $A(\hat{\xi}^k)$ aus Gleichung (3.30), Seite 44, eine besondere Bedeutung, daher werden im Anschluss charakterisierende Eigenschaften der Matrix $A(\hat{\xi}^k)$ vorgestellt.

Bemerkung 5.2.1. *Seien* $\xi = \sum_{j=1}^{r} \xi_j = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu} \in \mathcal{S}_{\le r}$*,* $j_1, j_2 \in \mathbb{N}_{\le r}$ *und* $\mu_1, \mu_2 \in \mathbb{N}_{\le d}$*. Gemäß Lemma 3.4.3, Seite 44, sind die Segmente der Matrix* $A(\hat{\xi})$ *durch*

$$A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \delta_{\mu_1\mu_2} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}$$

gegeben. Ordnet man diese Segmente gemäß Notation 5.1.1 an, dann erhält man folgende Blockmatrix:

$$A(\hat{\xi}) = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes G_\mu \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}, \tag{5.9}$$

wobei die Einträge von $\mathbb{E}_\mu \in \mathbb{R}^{d\times d}$ *und* $G_\mu \in \mathbb{R}^{r\times r}$ *wie folgt definiert sind:*

$$\begin{aligned} (\mathbb{E}_\mu)_{\mu_1\mu_2} &:= \delta_{\mu_1\mu}\delta_{\mu_2\mu}, \\ (G_\mu)_{j_1 j_2} &:= \langle \xi_{j_1}, \xi_{j_2} \rangle_\mu . \end{aligned}$$

Lemma 5.2.2. *Seien* $c \in \mathbb{R}_{>0}$ *und* $(\xi^k)_{k\in\mathbb{N}} \subset \mathcal{S}^c_r$ *die Folge von Elementartensor-Summen mit beschränkten Summanden aus Gleichung (5.2). Ferner sei zur Abkürzung* $\hat{A}_k := A(\hat{\xi}^k)$ *für alle* $k \in \mathbb{N}$ *gesetzt. Dann gilt:*

(i) Für alle $k \in \mathbb{N}$ *ist* $\hat{A}_k$ *symmetrisch und positiv definit, mithin regulär.*

(ii) Für alle $k \in \mathbb{N}$ *gilt*

$$\hat{A}_k^{-1} = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left[G^{(k)}_\mu\right]^{-1} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}, \tag{5.10}$$

wobei für $j_1, j_2 \in \mathbb{N}_{\le r}$

$$(G^{(k)}_\mu)_{j_1 j_2} := \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_\mu$$

gesetzt ist.

(iii) Es existiert ein $M \in \mathbb{R}_{>0}$, so dass für alle $u \in \mathfrak{R}_{d,r,\underline{t}}$ und alle $k \in \mathbb{N}$ die Ungleichung

$$\left\langle u, \hat{A}_k u \right\rangle \leq M\|u\|^2 \tag{5.11}$$

erfüllt ist.

Beweis.
(i) Sei $k \in \mathbb{N}$. Dann ist $\hat{A}_k$ symmetrisch, denn

$$\begin{aligned}
\hat{A}_k^t &= \sum_{\mu=1}^{d} \mathbb{E}_\mu^t \otimes {G_\mu^{(k)}}^t \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}^t \\
&= \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} \\
&= \hat{A}_k.
\end{aligned}$$

Für alle $\mu \in \mathbb{N}_{\leq d}$ ist $G_\mu^{(k)}$ eine Gramsche[1] Matrix und wegen Lemma 1.4.15, Seite 22, positiv definit, denn $\xi^k \in \mathcal{S}_r^c$. Damit ist auch $\hat{A}_k$ positiv definit, denn $\hat{A}_k$ ist eine Blockdiagonalmatrix, deren Diagonalblöcke $G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}_\mu^t}$ positiv definit sind.

(ii) Seien $k \in \mathbb{N}$ und

$$\widetilde{A}_k := \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left[G_\mu^{(k)}\right]^{-1} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}$$

gesetzt. Es gilt

$$\begin{aligned}
\hat{A}_k \cdot \widetilde{A}_k &= \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left(G_\mu^{(k)} \cdot \left[G_\mu^{(k)}\right]^{-1}\right) \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} \\
&= \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \mathbf{Id}_{\mathbb{R}^r} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} \\
&= \mathbf{Id}_{\mathbb{R}^{r|\underline{t}|}}.
\end{aligned}$$

Ebenso gilt

$$\widetilde{A}_k \cdot \hat{A}_k = \mathbf{Id}_{\mathbb{R}^{r|\underline{t}|}}.$$

(iii) Setze $M := rc^{2\frac{d-1}{d}} \in \mathbb{R}_{>0}$. Für alle $k \in \mathbb{N}$ und $u \in \mathfrak{R}_{d,r,\underline{t}}$ gilt folgende Kette von Abschätzungen:

$$\begin{aligned}
\left\langle u, \hat{A}_k u \right\rangle &\leq \|\hat{A}_k\| \|u\|^2 \\
&\leq \max_{\mu \in \mathbb{N}_{\leq d}} \|G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}\| \|u\|^2 \\
&\leq \max_{\mu \in \mathbb{N}_{\leq d}} \|G_\mu^{(k)}\| \|u\|^2 \\
&\leq \max_{\mu \in \mathbb{N}_{\leq d}} |\lambda_{\max}(G_\mu^{(k)})| \|u\|^2.
\end{aligned}$$

[1] Jørgen Pedersen Gram, dänischer Mathematiker, $*$ 27. Juni 1850, † 29. April 1916.

Wegen Gleichung (5.3) und $\xi^k \in \mathcal{S}_r^c$ ist

$$\|\xi_j^k\|_\mu \leq c^{\frac{d-1}{d}}$$

für alle $j \in \mathbb{N}_{\leq r}$. Der Satz von Gerschgorin[2] impliziert

$$|\lambda_{\max}(G_\mu^{(k)})| \in \bigcup_{j=1}^{r} \{\lambda \in \mathbb{R} : \left| \|\xi_j^k\|_\mu^2 - \lambda \right| \leq r_j\},$$

wobei

$$\begin{aligned} r_j &:= \sum_{j' \in \mathbb{N}_{\leq r} \setminus \{j\}} \left| \left\langle \xi_j^k, \xi_{j'}^k \right\rangle_\mu \right| \\ &\leq \sum_{j' \in \mathbb{N}_{\leq r} \setminus \{j\}} \|\xi_j^k\|_\mu \|\xi_{j'}^k\|_\mu \\ &\leq (r-1) c^{2\frac{d-1}{d}} \end{aligned}$$

ist, woraus dann

$$|\lambda_{\max}(G_\mu^{(k)})| \leq r c^{2\frac{d-1}{d}} = M$$

folgt. Insgesamt erhält man

$$\left\langle u, \hat{A}_k u \right\rangle \leq M \|u\|^2.$$

■

5.3 Analyse und Komplexität eines Minimierungsschrittes

Im Folgenden wird ein Minimierungsschritt beschrieben und dessen Komplexität untersucht. Begonnen wird mit dem Aufstellen des Gleichungssystems, im Anschluss daran steht die Berechnung der Abstiegsrichtung und der Schrittweite im Mittelpunkt der Analyse.

5.3.1 Datenvorbereitung

Notwendig für die Berechnung der Abstiegsrichtung ist die Bereitstellung des Gleichungssystems (4.41), Seite 61, dazu muss der Gradient von f an der Stelle $\hat{\xi}^k$ berechnet werden, außerdem werden für die Systemmatrix $C_k(\alpha_k)$ Segmente der Hesse-Matrix $f''(\hat{\xi}^k)$ benötigt, siehe dazu Abschnitt 5.3.2.1.

[2] Semjon Aranowitsch Gerschgorin, russischer Mathematiker, ∗ 24. August 1901, Pruzhany, † 30. Mai 1933, Weißrussland.

5.3.1.1 Präprozess zur Reduktion der Komplexität

Zur Berechnung des Gradienten und der Hesse-Matrix von f im aktuellen Iterationspunkt $\hat{\xi}^k$ müssen laut Abschnitt 3.3 bzw. Abschnitt 3.4 die in Notation 3.3.2 bzw. Notation 3.4.2 definierten Skalarprodukte mit Auslassung ausgewertet werden. Diese Terme waren für alle $j_1, j_2 \in \mathbb{N}_{\leq r}$, $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und alle $i \in \mathbb{N}_{\leq R}$ wie folgt definiert:

$$\begin{aligned} \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_{\mu_1} &:= \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1\}} \left\langle \xi_{j_1\mu}^k, \xi_{j_2\mu}^k \right\rangle, \\ \left\langle \xi_{j_1}^k, \alpha_i \right\rangle_{\mu_1} &:= \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1\}} \left\langle \xi_{j_1\mu}^k, \alpha_{i\mu} \right\rangle \end{aligned}$$

bzw.

$$\begin{aligned} \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_{\mu_1\mu_2} &:= \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1,\mu_2\}} \left\langle \xi_{j_1\mu}^k, \xi_{j_2\mu}^k \right\rangle, \\ \left\langle \xi_{j_1}^k, \alpha_i \right\rangle_{\mu_1\mu_2} &:= \prod_{\mu \in \mathbb{N}_{\leq d} \setminus \{\mu_1,\mu_2\}} \left\langle \xi_{j_1\mu}^k, \alpha_{i\mu} \right\rangle. \end{aligned}$$

Offenbar werden hierbei die Skalarprodukte

$$\left\langle \xi_{j_1\mu}^k, \xi_{j_2\mu}^k \right\rangle \quad \text{bzw.} \quad \left\langle \xi_{j_1\mu}^k, \alpha_{i\mu} \right\rangle \tag{5.12}$$

mehrfach benötigt, daher werden diese zuvor berechnet und abgespeichert. In Folge dessen sind redundante Berechnungen überflüssig und der zusätzliche Speicherbedarf,

$$d \cdot \binom{r+1}{2} \quad \text{bzw.} \quad d \cdot R \cdot r, \tag{5.13}$$

ist von moderater Größe, wobei im ersten Fall die Symmetrie des Skalarproduktes ausgenutzt wurde. Der Rechenaufwand für ein Skalarprodukt beträgt $2 \cdot t_\mu - 1$ Operationen, so dass man insgesamt

$$\binom{r+1}{2} \cdot \sum_{\mu=1}^{d} (2t_\mu - 1) \quad \text{bzw.} \quad R \cdot r \cdot \sum_{\mu=1}^{d} (2t_\mu - 1) \tag{5.14}$$

Rechenoperationen benötigt. Damit ξ^k die Bedingung (5.5), d.h.

$$\forall j \in \mathbb{N}_{\leq r} \forall \mu \in \mathbb{N}_{\leq d} \forall \nu \in \mathbb{N}_{\leq d} \setminus \{\mu\} : \|{\xi^k}_{j\mu}\| = \|{\xi^k}_{j\nu}\| = \sqrt[d]{\|\xi_j^k\|},$$

erfüllt, sind die Repräsentantenvektoren von ξ^k zuvor zu skalieren. Hierfür werden

$$r \cdot \sum_{\mu=1}^{d} t_\mu$$

Operationen verbucht. Zusätzlich werden für die Berechnung der Skalierungsfaktoren

$$r \cdot \left(2 \sum_{\mu=1}^{d} t_\mu + d - 1 \right)$$

Rechenoperationen in Anspruch genommen, so dass man insgesamt eine Komplexität von

$$r \cdot \left(3 \sum_{\mu=1}^{d} t_\mu + d - 1 \right)$$

erhält.

5.3.1.2 Berechnung des Gradienten und der System-Matrix

In der anschließenden Komplexitätsanalyse kann man unter anderem wegen Abschnitt 5.3.1.1 die Vereinbarung treffen, dass die Skalarprodukte aus Gleichung (5.12) schon vorab berechnet sind und daher nicht weiter in die Analyse einfließen. Wie bereits erwähnt, gelten auch hier die in Abschnitt 3.3 bzw. 3.4 verwendeten Bezeichnungen.
Für die Berechnung der Abstiegsrichtung muss die Hesse-Matrix der Zielfunktion f bereitgestellt werden. Gemäß Abschnitt 3.4 sind die Segmente von f'' an der Stelle ξ^k für alle $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und alle $j_1, j_2 \in \mathbb{N}_{\leq r}$ wie folgt definiert:

$$f''_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) = f''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) + \lambda_1 g''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) + \lambda_2 g''_{2\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k),$$

wobei z. B.

$$\begin{aligned} f''_{1\,\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) &= \frac{1}{\|\alpha\|^2} \Big[A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) + B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) + C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) \\ &- D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) \Big], \end{aligned}$$

$$\begin{aligned} A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) &:= \delta_{\mu_1\mu_2} \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, \\ B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) &:= \overline{\delta}_{\mu_1\mu_2} \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2} \xi^k_{j_2\mu_1} (\xi^k_{j_1\mu_2})^t, \\ C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) &:= \overline{\delta}_{\mu_1\mu_2} \delta_{j_1 j_2} \sum_{j=1}^{r} \left\langle \xi^k_{j}, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} \xi^k_{j\mu_1} (\xi^k_{j\mu_2})^t, \\ D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}^k) &:= \overline{\delta}_{\mu_1\mu_2} \delta_{j_1 j_2} \sum_{i=1}^{R} \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} \alpha_{i\mu_1} \alpha^t_{i\mu_2} \end{aligned}$$

gesetzt sind. Im Folgenden ist es nicht notwendig, die Hesse-Matrix von f bzw. Teile der Hesse-Matrix aufzustellen, denn zur Berechnung der Abstiegsrichtung wird ein iteratives Verfahren verwendet, welches nur die Matrix-Vektor-Multiplikation benötigt. Daher genügt es, die Skalarprodukte mit Auslassung,

$$\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1}, \quad \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2}, \quad \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2}, \quad i \in \mathbb{N}_{\leq R},$$

zu berechnen und abzuspeichern. Hierfür ist lediglich ein zusätzlicher Speicherbedarf von

$$\binom{d+1}{2} \cdot \left[\binom{r+1}{2} + R \cdot r \right]$$

notwendig, da die Ausdrücke der Form $\xi^k_{j_2\mu_1}(\xi^k_{j_1\mu_2})^t$ und $\alpha_{i\mu_1}\alpha^t_{i\mu_2}$ durch die Eingabe bzw. Ausgabe der Tensorsumme α bzw. ξ^k vorgegeben sind und nicht eigens bereitgestellt werden müssen. Im Übrigen trifft diese Betrachtung auch in analoger Weise für die Segmente von g_1'' und g_2'' zu, wobei einige Teile von g_1'' in Übereinstimmung mit Gleichung (5.5) gleich null sind und somit nicht weiter berücksichtigt werden.

Lemma 5.3.1. *Seien $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ und $i \in \mathbb{N}_{\leq R}$. Dann werden für die Berechnung von $\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1}$, $\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2}$ und $\left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2}$ insgesamt*

$$\binom{d}{2} \cdot (d-3) \cdot \left[\binom{r+1}{2} + R \cdot r \right] + d \cdot \binom{r+1}{2} \tag{5.15}$$

Rechenoperationen benötigt, wobei gemäß Abschnitt 5.3.1.1 vorausgesetzt ist, dass die Skalarprodukte der einzelnen Repräsentantenvektoren zuvor berechnet worden sind.

Beweis. Seien $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$, $j_1, j_2 \in \mathbb{N}_{\leq r}$ und $i \in \mathbb{N}_{\leq R}$. Für die Berechnung von

$$\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2} = \prod_{\nu \in \mathbb{N}_{\leq d} \setminus \{\mu_1, \mu_2\}} \left\langle \xi^k_{j_1\nu}, \xi^k_{j_2\nu} \right\rangle$$

bzw.

$$\left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} = \prod_{\nu \in \mathbb{N}_{\leq d} \setminus \{\mu_1, \mu_2\}} \left\langle \alpha_{i\nu}, \xi^k_{j_1\nu} \right\rangle$$

werden jeweils $d-3$ Multiplikationen benötigt, so dass unter Ausnutzung der Symmetrie insgesamt

$$\binom{d}{2} \cdot (d-3) \cdot \binom{r+1}{2}$$

bzw.

$$\binom{d}{2} \cdot (d-3) \cdot R \cdot r$$

Operationen anfallen. Dies ergibt in der Summe

$$\binom{d}{2} \cdot (d-3) \cdot \left[\binom{r+1}{2} + R \cdot r \right]$$

Rechenoperationen. Ferner benutzt man für die Skalarprodukte mit einfacher Auslassung folgende Gleichung:

$$\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1} = \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2} \left\langle \xi^k_{j_1\mu_2}, \xi^k_{j_2\mu_2} \right\rangle.$$

Hierfür werden dann noch einmal

$$d \cdot \binom{r+1}{2}$$

Rechenoperationen benötigt. Summa summarum ergibt dies die Behauptung. ■

Lemma 5.3.2. *Die Komplexität bei der Berechnung des Gradienten von f an der Stelle $\hat{\xi}^k$ beträgt*

$$r \cdot \sum_{\mu_1=1}^{d} ((2(R+r)-1) \cdot t_{\mu_1} + 2) + d \cdot R \cdot r. \tag{5.16}$$

Beweis. Seien $j_1 \in \mathbb{N}_{\leq r}$ und $\mu_1 \in \mathbb{N}_{\leq d}$. Laut Abschnitt 3.3 sind die Komponenten des Gradienten von f an der Stelle $\hat{\xi}^k$ wie folgt bestimmt:

$$f'_{j_1\mu_1}(\hat{\xi}^k) = f'_{1\,j_1\mu_1}(\hat{\xi}^k) + \lambda_1 g'_{1\,j_1\mu_1}(\hat{\xi}^k) + \lambda_2 g'_{2\,j_1\mu_1}(\hat{\xi}^k),$$

hierbei waren $f'_{1\,j_1\mu_1}(\hat{\xi}^k)$, $g'_{1\,j_1\mu_1}(\hat{\xi}^k)$ und $g'_{2\,j_1\mu_1}(\hat{\xi}^k)$ von folgender Gestalt:

$$\begin{aligned}
f'_{1\,j_1\mu_1}(\hat{\xi}^k) &= \frac{1}{\|\alpha\|^2}\left[-\sum_{i=1}^{R}\left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1} \alpha_{i\mu_1} + \sum_{j=1}^{r}\left\langle \xi^k_j, \xi^k_{j_1}\right\rangle_{\mu_1} \xi^k_{j\mu_1}\right], \\
g'_{1\,j_1\mu_1}(\hat{\xi}^k) &= \frac{1}{\sqrt[d]{\|\alpha\|^4}}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi^k_{j_1\mu_1}\|^2 - \|\xi^k_{j_1\mu}\|^2\right)\right]\xi^k_{j_1\mu_1}, \\
g'_{2\,j_1\mu_1}(\hat{\xi}^k) &= \frac{1}{\|\alpha\|^2}\left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1} \xi^k_{j_1\mu_1}.
\end{aligned}$$

Ferner sollte man die Ausdrücke $f'_{1\,j_1\mu_1}(\hat{\xi}^k) + \lambda_2 g'_{2\,j_1\mu_1}(\hat{\xi}^k) =: \tilde{f}_{j_1\mu_1}(\hat{\xi}^k)$ zusammenfassen, d.h.

$$\begin{aligned}
\tilde{f}_{j_1\mu_1}(\hat{\xi}^k) &= \underbrace{\frac{1}{\|\alpha\|^2}}_{=1}\left[-\sum_{i=1}^{R}\left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1} \alpha_{i\mu_1}\right. \\
&+ \left.(1+\lambda_2)\left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1} \xi^k_{j_1\mu_1} + \sum_{j=1,j\neq j_1}^{r}\left\langle \xi^k_j, \xi^k_{j_1}\right\rangle_{\mu_1} \xi^k_{j\mu_1}\right].
\end{aligned}$$

Als Erstes sind die Skalarprodukte mit Auslassung zu berechnen, d.h.

$$\left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1} \quad \text{bzw.} \quad \left\langle \xi^k_j, \xi^k_{j_1}\right\rangle_{\mu_1}.$$

Dabei fallen insgesamt

$$d \cdot R \cdot r$$

Rechenoperationen an, denn es gilt

$$\left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1} = \left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1\mu_2}\left\langle \alpha_{i\mu_2}, \xi^k_{j_1\mu_2}\right\rangle.$$

Im Übrigen wurden $\left\langle \alpha_i, \xi^k_{j_1}\right\rangle_{\mu_1\mu_2}$ sowie $\left\langle \xi^k_j, \xi^k_{j_1}\right\rangle_{\mu_1}$ bereits in Lemma 5.3.1 ermittelt.
Bei der Berechnung einer Komponente $\tilde{f}_{j_1\mu_1}(\hat{\xi}^k)$ werden ferner

$$(2R - 1 + 2r - 1) \cdot t_{\mu_1} + t_{\mu_1} + 2 = (2(R+r)-1) \cdot t_{\mu_1} + 2$$

Rechenoperationen benötigt. Wegen (5.5) und der im Abschnitt 5.3.1.1 beschriebenen Skalierung gilt

$$g'_{1\,j_1\mu_1}(\hat{\xi}^k) = 0,$$

daher sind diese Anteile nicht zu berechnen. Mittels Summation über alle $j_1 \in \mathbb{N}_{\leq r}$ und $\mu_1 \in \mathbb{N}_{\leq d}$ folgt nun die Behauptung

$$r \cdot \sum_{\mu_1=1}^{d} \big((2(R+r)-1) \cdot t_{\mu_1} + 2\big) + d \cdot R \cdot r.$$

■

Bemerkung 5.3.3. *Ein wesentlicher Anteil zur Bestimmung der Hesse-Matrix und des Gradienten von f ist die Berechnung der Skalarprodukte mit Auslassung*

$$\left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1}, \quad \left\langle \xi^k_{j_1}, \xi^k_{j_2} \right\rangle_{\mu_1\mu_2}, \quad \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1}, \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2},$$

wobei $j_1, j_2 \in \mathbb{N}_{\leq r}$, $i \in \mathbb{N}_{\leq R}$ und $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ sind. Getreu Lemmata 5.3.1 und 5.3.2 beträgt hier die Komplexität

$$\binom{d}{2} \cdot (d-3) \cdot \left[\binom{r+1}{2} + R \cdot r \right] + d \cdot \left[\binom{r+1}{2} + r \cdot R \right],$$

somit erhält man eine kubische Komplexität in d. Setzt man voraus, dass die Skalarprodukte

$$\left\langle \xi^k_{j_1\mu_1}, \xi^k_{j_2\mu_1} \right\rangle, \quad \left\langle \xi^k_{j_1\mu_1}, \alpha^k_{i\mu_1} \right\rangle$$

ungleich null sind, dann kann man diese Komplexität um eine Potenz reduzieren. Dies wird im Folgenden am Beispiel von $\left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2}$ erläutert. Vorab berechnet man

$$\left\langle \alpha_i, \xi^k_{j_1} \right\rangle = \prod_{\mu=1}^{d} \left\langle \alpha_{i\mu}, \xi^k_{j_1\mu} \right\rangle,$$

hierfür werden

$$(d-1) \cdot (r+R)$$

Operationen benötigt. Für die Berechnung von $\left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2}$ sind dann noch

$$2\binom{d}{2} \cdot (r+R)$$

Rechenoperationen erforderlich ($\mu_1 \neq \mu_2$), denn es gilt

$$\left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} = \frac{\left\langle \alpha_i, \xi^k_{j_1} \right\rangle}{\left\langle \alpha_{i\mu_1}, \xi^k_{j_1\mu_1} \right\rangle \left\langle \alpha_{i\mu_2}, \xi^k_{j_1\mu_2} \right\rangle}. \tag{5.17}$$

Insgesamt werden somit nur

$$2\binom{d}{2} \cdot (r+R) + (d-1) \cdot (r+R)$$

Operationen benötigt, so dass die Komplexität nur quadratisch mit der Dimension wächst. Bei praktischen Berechnungen können die Skalarprodukte

$$\left\langle \xi^k_{j_1\mu_1}, \alpha^k_{i\mu_1} \right\rangle$$

fast null sein, dies konnte z.B. bei den Daten aus [5] und [6] beobachtet werden. Bei diesen Berechnungen führte die in Gleichung (5.17) vorgestellte Methode in einigen Fällen zu einem instabilen Verfahren. Der hier entwickelte Algorithmus wurde hauptsächlich für Probleme konzeptioniert, bei denen die Dimension d kleiner als etwa eintausend ist. Daher wird in dieser Arbeit die stabile Berechnung getreu Lemma 5.3.1 sowie Lemma 5.3.2 bei der softwaremäßigen Umsetzung favorisiert.

5.3.2 Berechnung der Abstiegsrichtung

Wie bereits erwähnt, wird das Minimierungsproblem 3.1.1 mit Hilfe von Algorithmus 4.4.1 iterativ gelöst. In erster Linie ist hierbei die Berechnung der Abstiegsrichtung d^k von Bedeutung, welche gemäß Gleichung (4.41), Seite 61, durch die Wahl von

$$C_k(\alpha_k) := (1-\alpha_k)A_k + \alpha_k H_k, \quad \alpha_k \in (0,1],$$

festgelegt ist. Aus diesem Grund wird im Anschluss die Wahl der Matrizen A_k und H_k beschrieben.

5.3.2.1 Definition der Abstiegsrichtung

Der blockdiagonale Anteil der Hesse-Matrix von f_1, $A(\hat{\xi^k})$ aus Gleichung (5.9), ist bei der numerischen Umsetzung von zentraler Bedeutung. Im Folgenden kann ohne Beschränkung der Allgemeinheit vorausgesetzt werden, dass ein sehr kleines $m \in \mathbb{R}_{>0}$ existiert, so dass

$$\forall u \in \mathfrak{R}_{d,r,\underline{t}} : m\|u\|^2 \leq \left\langle u, A(\hat{\xi^k})u \right\rangle$$

für alle Iterationsschritte erfüllt ist, denn anderenfalls könnte man wegen Lemma 1.4.15, Seite 22, einen Tensor mit kleinerem Tensorrang konstruieren, welcher eine vergleichbare Approximationsgenauigkeit aufweist. Darüber hinaus gilt wegen Lemma 5.2.2 sogar

$$\forall u \in \mathfrak{R}_{d,r,\underline{t}} : m\|u\|^2 \leq \left\langle u, A(\hat{\xi^k})u \right\rangle \leq M\|u\|^2.$$

Ferner können lineare Gleichungen mit $A(\hat{\xi^k})$ gemäß Gleichung (5.10) mit einem Aufwand von $\mathcal{O}(d \cdot r^3)$ aufgelöst werden. Insgesamt erfüllt die Folge der Matrizen $(A(\hat{\xi^k}))_{k\in\mathbb{N}}$ die Voraussetzungen an $(A_k)_{k\in\mathbb{N}}$, siehe Gleichung (4.34), Seite 59. Aus diesem Grund wird

$$A_k := A(\hat{\xi^k}) \tag{5.18}$$

für $k \in \mathbb{N}$ gesetzt.
Die erste Wahl von H_k ist

$$\begin{aligned} H_k &:= f''(\hat{\xi}^k) && (5.19)\\ &= A_k + B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}, && (5.20)\end{aligned}$$

wobei die Matrizen B_k, C_k, D_k durch Lemma 3.4.3 und $G_{1k} := g_1''(\hat{\xi}^k)$ bzw. $G_{2k} := g_2''(\hat{\xi}^k)$ wie in Lemma 3.4.4 bzw. Lemma 3.4.5 definiert sind. Diese Festlegung hat gemäß Satz 4.4.3 den Vorteil eines quadratisch konvergenten Verfahrens.
Bei vielfältigen numerischen Tests mit praxisrelevanten Daten, siehe Abschnitt 6.3, stellte sich folgende abgeschwächte Variante als sehr vorteilhaft heraus:

$$H_k = \begin{cases} A_k + B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}, & \overline{\alpha_k} \neq 1 \\ A_k + B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}, & \overline{\alpha_k} = 1, \end{cases} \tag{5.21}$$

wobei $\overline{\alpha_k} \in (0, 1]$, wie im Algorithmus 4.4.1 beschrieben, mittels Armijo-Regel berechnet wird. Bei dieser Festlegung konnte in vielen Beispielen eine verringerte Anzahl von Iterationsschritten beobachtet werden, daneben ist die Abstiegsrichtung für $\overline{\alpha_k} \neq 1$ kostengünstiger zu berechnen; diese Tatsache wird im Abschnitt 5.3.2.3 eingehender beschrieben. Ferner konnte man beobachten, dass nach wenigen Iterationen die volle Schrittweite, $\overline{\alpha_k} = 1$, angenommen wurde, wenn auch die Voraussetzungen von Lemma 4.3.10, Seite 56, nicht erfüllt sind. In diesen Fällen erhielt man ein quadratisch konvergentes Verfahren, denn die Methode ging in das eigentliche Newton-Verfahren über. Unabhängig davon ist die Konvergenz des Verfahrens gegen einen stationären Punkt von f gesichert, denn die Abstiegsrichtungen sind gradientenähnlich und die Schrittweiten werden mittels Armijo-Regel berechnet.
Für die Matrix $C_k(\alpha_k)$ bedeutet diese Wahl im ersten Fall, d.h. $H_k = f''(\hat{\xi}^k)$,

$$C_k(\alpha_k) = A_k + \alpha_k \left(B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right) \tag{5.22}$$

bzw. bei der zweiten Definition erhält man

$$C_k(\alpha_k) = \begin{cases} A_k + \alpha_k \left(B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right), & \overline{\alpha_k} \neq 1 \\ A_k + \alpha_k \left(B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right), & \overline{\alpha_k} = 1. \end{cases} \tag{5.23}$$

Mit der vereinfachten Wahl

$$C_k(\alpha_k) = A_k + \alpha_k \left(B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right) \tag{5.24}$$

konnte man bei allen Rechenbeispielen auch gute Ergebnisse erzielen, siehe Abschnitt 6.3. Ferner ist diese Wahl kostengünstiger für Tensoren mit sehr großem Eingabetensorrang R, dies wird im Abschnitt 5.3.2.3 deutlicher.

Notation 5.3.4. *Im Folgenden wird zur abkürzenden Schreibweise folgende Notation für die beiden unterschiedlichen Matrizen aus Gleichung (5.23) verwendet*

$$C_k^{(1)}(\alpha_k) := A_k + \alpha_k \left(B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right) \tag{5.25}$$

und

$$C_k^{(2)}(\alpha_k) := A_k + \alpha_k \left(B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right). \tag{5.26}$$

5.3.2.2 Lösen des Gleichungssystems und Wahl des Vorkonditionierers

Maßgebend für die Komplexität ist neben dem Aufstellen des Gradienten und der Hesse-Matrix von f vor allem die Berechnung der Abstiegsrichtung. Hierbei wird die Lösung des linearen Gleichungssystems

$$C_k(\alpha_k)d^k = f'(\hat{\xi}^k) \tag{5.27}$$

mit Hilfe des konjugierten Gradienten-Verfahrens (cg-Verfahren) berechnet, wobei $\alpha_k \in [0,1]$ entsprechend Algorithmus 4.4.1 modifiziert wird. Für die Konvergenzrate des cg-Verfahrens gilt folgender Satz.

Satz 5.3.5. *Sei C positiv definit mit $\lambda := \lambda_{\min}(C)$, $\Lambda := \lambda_{\max}(C)$ und der Konditionszahl $\kappa := \Lambda/\lambda$. Dann erfüllt der Fehler $e^m := Cd^m - g$ der cg-Iterierten d^m folgende Abschätzung:*

$$\|e^m\|_C \leq \frac{2(1-1/\kappa)^m}{(1+1/\sqrt{\kappa})^{2m} + (1-1/\sqrt{\kappa})^{2m}}\|e^0\|_C = c^m \frac{2}{1+c^{2m}}\|e^0\|_C, \tag{5.28}$$

wobei $c := \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{\sqrt{\Lambda}-\sqrt{\lambda}}{\sqrt{\Lambda}+\sqrt{\lambda}}$ gesetzt ist.

Beweis. [15, Satz 9.4.12, Seite 257.]. ■

Ein Vorteil des cg-Verfahrens besteht darin, dass die asymptotische Konvergenzrate vom gesamten Spektrum von C beeinflusst wird und nicht ausschließlich von den extremen Eigenwerten abhängt. So wird in den ersten cg-Schritten der cg-Fehler e^m zuerst in $V := \operatorname{span}\{v_{\min}, v_{\max}\}$ aufgelöst, wobei $v_{\min}$ bzw. $v_{\max}$ die zugehörigen Eigenvektoren von $\lambda_{\min}$ bzw. $\lambda_{\max}$ sind. Ferner werden sich die cg-Fehler in den anschließenden Schritten auf $V^\perp$ zubewegen. C beschränkt auf $V^\perp$ hat jedoch das Spektrum $\sigma(C) \setminus \{\lambda_{\min}, \lambda_{\max}\}$ und die Konditionszahl Λ_2/λ_2, wobei λ_2 der zweitkleinste und Λ_2 der zweitgrößte Eigenwert von C ist. Damit erhält man insgesamt ein verbessertes Fehlerverhalten. Eine genaue Analyse hierzu findet man in [41].
Um die Konvergenzgeschwindigkeit des cg-Verfahrens zu verbessern, werden nun geeignete Vorkonditionierer vorgestellt. Bei Daten aus praktischen Anwendungen können die Normen der Summanden von ξ^k in ihrer Größe sehr stark variieren; dies führt zu einer verminderten Konvergenzgeschwindigkeit. Diesem Umstand kann man aber mit Hilfe einer Maßstabsänderung entgegenwirken, siehe [24, Abschnitt 7.3 Beschleunigung durch Maßstabsänderung, Seiten 116 f.]. Hierfür wird Gleichung (6.7) mittels folgender Matrix V_k^{-1}

$$V_k^{-1} := \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \operatorname{diag}\left(\|\xi_j^k\|_\mu^{-1}\right)_{j\in\mathbb{N}_{\leq r}} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} \in \mathbb{R}^{r|\underline{t}|\times r|\underline{t}|} \tag{5.29}$$

beidseitig vorkonditioniert. Damit geht Gleichung (6.7) in Gleichung

$$V_k^{-1}C_k(\alpha_k)V_k^{-1}d^k = V_k^{-1}f'(\hat{\xi}^k)$$

über.

Notation 5.3.6. *Seien $H_k \in \{A_k, B_k, C_k, D_k, G_{1k}, G_{2k}\}$ und V_k^{-1} wie in Gleichung (5.29) definiert. Dann werden abkürzend folgende Bezeichnungen verwendet:*

$$\tilde{f}'(x^k) := V_k^{-1} f'(x^k) \in \mathbb{R}^{r|\underline{t}|} \tag{5.30}$$

und

$$\tilde{H}_k := V_k^{-1} H_k V_k^{-1} \in \mathbb{R}^{r|\underline{t}| \times r|\underline{t}|}. \tag{5.31}$$

Wie schon erwähnt, hat die Matrix A_k eine besondere Bedeutung. Nach Lemma 5.2.2 ist A_k leicht zu invertieren, mit einem Aufwand von $d \cdot \mathcal{O}(r^3)$. Diese Tatsache trifft auch analog auf $\tilde{A}_k$ zu, denn es gilt offenbar

$$\tilde{A}_k^{-1} = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes V_{k\mu} G_\mu^{-1} V_{k\mu} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}},$$

wobei $V_{k\mu} := \operatorname{diag}\left(\|\xi_j^k\|_\mu\right)_{j \in \mathbb{N}_{\le r}} \in \mathbb{R}^{r \times r}$ gesetzt ist. Verwendet man nun den blockdiagonalen Anteil von $f''(\hat{\xi}_k)$, $\tilde{A}_k^{-1}$, als weiteren Vorkonditionierer, dann erhält man für $\tilde{A}_k^{-1}\tilde{C}_k(\alpha_k)$

$$\tilde{A}_k^{-1}\tilde{C}_k(\alpha_k) = \begin{cases} \mathbf{Id} + \alpha_k \tilde{A}_k^{-1}\left(\tilde{B}_k + \lambda_1 \tilde{G}_{1k} + \lambda_2 \tilde{G}_{2k}\right), & \overline{\alpha_k} \neq 1 \\ \mathbf{Id} + \alpha_k \tilde{A}_k^{-1}\left(\tilde{B}_k + \tilde{C}_k - \tilde{D}_k + \lambda_1 \tilde{G}_{1k} + \lambda_2 \tilde{G}_{2k}\right), & \overline{\alpha_k} = 1. \end{cases}$$

Getreu Algorithmus 4.4.1 wird zunächst versucht, mit $\alpha_k = 1$ bzw. genügend großen α_k-Werten ein positiv definites $C_k(\alpha_k)$ zu bestimmen. Ist dies nicht möglich oder Gleichung (6.7) schwer aufzulösen, so wird α_k fortlaufend verkleinert. Auf diese Weise wird $C_k(\alpha_k)$ positiv definit und die Kondition von $\tilde{A}_k^{-1}\tilde{C}_k(\alpha_k)$ speziell in den problematischen Fällen schrittweise verbessert.
Im Anschluss wird das cg-Verfahren zur Lösung des linearen Gleichungssystems $Cd = g$ mit der Systemmatrix C, dem Vorkonditionierer W^{-1} und der rechten Seite g angegeben. Da das cg-Verfahren nur für positiv definite und symmetrische Matrizen wohldefiniert ist, müssen bei der Umsetzung spezielle Vorsichtsmaßnahmen getroffen werden.
In der Praxis kann die Matrix $\tilde{A}_k^{-1}\tilde{C}_k(\alpha_k)$ für $\alpha_k \approx 1$ extrem schlecht konditioniert sein. Diese Tatsache trat bei den Testberechnungen insbesondere dann auf, wenn man versucht hatte, eine Niedrig-Tensorrang-Approximationen zu berechnen, deren Güte deutlich unterhalb der Genauigkeit lag, welche durch das zugrunde liegende Modell vorgegeben war, siehe z. B. Abschnitt 6.1.2, Tabelle 6.2, Seite 99, für r=3 und r=4.
Mit Hilfe der Wahl von $m_{\max}$ kann man die maximale Anzahl der cg-Iterationen beschränken. Ist die maximale Anzahl von Iterationen erreicht und die Norm des cg-Fehlers noch nicht klein genug, dann wird die Kondition der Matrix $\tilde{A}_k^{-1}\tilde{C}_k(\alpha_k)$ als zu schlecht angesehen und α_k verkleinert. Damit verbessert man sukzessive die Kondition der Systemmatrix. Bei allen numerischen Testrechnungen im Kapitel 6 wurde $m_{\max} = 80$ und $\gamma = 0.9$, siehe Algorithmus 4.4.1, fest gewählt.

Algorithmus 5.3.1 Methode der konjugierten Gradienten

1: Wähle $\varepsilon \in \mathbb{R}_{>0}$, $m_{\max} \in \mathbb{N}$ und eine gute Näherungslösung d^0. Setze $m := 0$, $r^0 := g - Cd^0$, $p^0 := W^{-1}r^0$ und $\varrho_0 := \langle p^0, r^0 \rangle$.
2: **while** $\|r^m\| > \varepsilon$ und $m \leq m_{\max}$ **do**
3: Setze $a^m := Cp^m$.
4: **if** $\langle a^m, p^m \rangle > 0$ **then**
5: Setze:

$$\begin{aligned} \lambda &:= \frac{\varrho_m}{\langle a^m, p^m \rangle}, \\ d^{m+1} &:= d^m + \lambda p^m, \\ r^{m+1} &:= r^m - \lambda a^m, \\ q^{m+1} &:= W^{-1} r^{m+1}, \\ \varrho_{m+1} &:= \left\langle p^{m+1}, r^{m+1} \right\rangle. \end{aligned}$$

6: Setze $p^{m+1} := q^{m+1} + \frac{\varrho_{m+1}}{\varrho_m} p^m$ und $m \leftarrow m + 1$.
7: **else**
8: Stop, denn C ist nicht positiv definit. Setze $m := m_{\max} + 1$.
9: **end if**
10: **end while**

5.3.2.3 Komplexität der Matrix-Vektor-Multiplikation

Im folgenden Abschnitt wird die Komplexität der Matrix-Vektor-Auswertung mit der Matrix $C_k(\alpha_k)$ angegeben, wobei entsprechend Absatz 5.3.2.1 auf die beiden unterschiedlichen Festlegungen eingegangen wird. Hierzu werden einzelne Teilsegmente der Hesse-Matrix von f an der Stelle $\hat{\xi}^k$ untersucht, denn die Matrix $C_k(\alpha_k)$ ist unabhängig von der Festsetzung aus diesen Anteilen zusammengesetzt.

Lemma 5.3.7. *Sei $A_k := A(\hat{\xi}^k)$ die Matrix aus Lemma 3.4.3, d.h. die Segmente von A_k sind für alle $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$ gegeben durch*

$$A_{k\mu_1\mu_2 j_1 j_2} = \delta_{\mu_1\mu_2} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}.$$

Dann werden für eine Matrix-Vektor-Multiplikation mit A_k

$$r \cdot (2r - 1) \cdot \sum_{\mu=1}^{d} t_\mu \tag{5.32}$$

Operationen benötigt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$ und $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$. Für die Komponenten von $A_k v$ gilt

$$\begin{aligned} (A_k v)_{j_1\mu_1} &= \sum_{\mu_2=1}^{d} \sum_{j_2=1}^{r} \delta_{\mu_1\mu_2} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1} v_{j_2\mu_2} \\ &= \sum_{j_2=1}^{r} \langle \xi_{j_1}, \xi_{j_2} \rangle_{\mu_1} v_{j_2\mu_1}. \end{aligned}$$

Dementsprechend fallen hierbei $(2r-1)\cdot t_{\mu_1}$ Rechenoperationen an, so dass für alle Komponenten

$$r\cdot(2r-1)\cdot\sum_{\mu=1}^{d} t_\mu$$

arithmetische Operationen benötigt werden. ■

Bemerkung 5.3.8. *Gemäß Lemma 5.2.2 ist* $A_k = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}$ *invertierbar und es gilt*

$$A_k^{-1} = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left[G_\mu^{(k)}\right]^{-1} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}.$$

Da A_k^{-1} *die gleiche Struktur wie* A_k *hat, folgt analog zu Lemma 5.3.7, dass eine Matrix-Vektor-Multiplikation mit* A_k^{-1} *eine Komplexität von*

$$r\cdot(2r-1)\cdot\sum_{\mu=1}^{d} t_\mu \tag{5.33}$$

hat.

Lemma 5.3.9. *Sei* $B_k := B(\hat{\xi}^k)$ *die Matrix aus Lemma 3.4.3, d.h. die Segmente von* B_k *sind für alle* $\mu_1,\mu_2 \in \mathbb{N}_{\le d}$ *und* $j_1, j_2 \in \mathbb{N}_{\le r}$ *mittels*

$$B_{k\mu_1\mu_2 j_1 j_2} = \overline{\delta}_{\mu_1\mu_2} \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_{\mu_1\mu_2} \xi_{j_2\mu_1}^k (\xi_{j_1\mu_2}^k)^t$$

gegeben. Dann werden für eine Matrix-Vektor-Multiplikation mit B_k

$$r\cdot(4(d-1+r)-1)\cdot\sum_{\mu=1}^{d} t_\mu \tag{5.34}$$

Operationen benötigt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\le d}$, $j_1 \in \mathbb{N}_{\le r}$ und $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\le d}, j \in \mathbb{N}_{\le r}) \in \mathfrak{R}_{d,r,\underline{t}}$. Für die Komponenten von $B_k v$ gilt

$$\begin{aligned}(B_k v)_{j_1\mu_1} &= \sum_{\mu_2=1}^{d}\sum_{j_2=1}^{r} \overline{\delta}_{\mu_1\mu_2} \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_{\mu_1\mu_2} \left\langle \xi_{j_1\mu_2}^k, v_{j_2\mu_2} \right\rangle \xi_{j_2\mu_1}^k \\ &= \sum_{\mu_2=1,\mu_2\neq\mu_1}^{d}\sum_{j_2=1}^{r} \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_{\mu_1\mu_2} \left\langle \xi_{j_1\mu_2}^k, v_{j_2\mu_2} \right\rangle \xi_{j_2\mu_1}^k.\end{aligned}$$

Hierfür fallen

$$2(d-1+r)\cdot t_{\mu_1} + (2(d-1+r)-1)\cdot t_{\mu_1} \;=\; (4(d-1+r)-1)\cdot t_{\mu_1}$$

Rechenoperationen an, so dass für alle Komponenten

$$r\cdot(4(d-1+r)-1)\cdot\sum_{\mu=1}^{d} t_\mu$$

arithmetische Operationen benötigt werden. ■

Lemma 5.3.10. *Sei $C_k := C(\hat{\xi}^k)$ die Matrix aus Lemma 3.4.3, d.h. die Segmente von C_k sind für alle $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$ durch*

$$C_{k\mu_1\mu_2 j_1 j_2} = \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2} \sum_{j=1}^{r} \left\langle \xi_j^k, \xi_{j_1}^k \right\rangle_{\mu_1\mu_2} \xi_{j\mu_1}^k (\xi_{j\mu_2}^k)^t$$

gegeben. Dann werden für eine Matrix-Vektor-Multiplikation mit C_k

$$r \cdot (4(d-1+r)-1) \cdot \sum_{\mu=1}^{d} t_\mu \tag{5.35}$$

Operationen benötigt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$ und $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$. Es gilt für die Komponenten von $C_k v$

$$\begin{aligned}
(C_k v)_{j_1\mu_1} &= \sum_{\mu_2=1}^{d} \sum_{j_2=1}^{r} \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2} \sum_{j=1}^{r} \left\langle \xi_j^k, \xi_{j_1}^k \right\rangle_{\mu_1\mu_2} \xi_{j\mu_1}^k (\xi_{j\mu_2}^k)^t v_{j_2\mu_2} \\
&= \sum_{\mu_2=1, \mu_2\neq\mu_1}^{d} \sum_{j=1}^{r} \left\langle \xi_j^k, \xi_{j_1}^k \right\rangle_{\mu_1\mu_2} \left\langle \xi_{j\mu_2}^k, v_{j_1\mu_2} \right\rangle \xi_{j\mu_1}^k.
\end{aligned}$$

Hierfür fallen auch

$$2(d-1+r)\cdot t_{\mu_1} + (2(d-1+r)-1)\cdot t_{\mu_1} = (4(d-1+r)-1)\cdot t_{\mu_1}$$

Rechenoperationen an, so dass für alle Komponenten

$$r \cdot (4(d-1+r)-1) \cdot \sum_{\mu=1}^{d} t_\mu$$

arithmetische Operationen benötigt werden. ■

Lemma 5.3.11. *Sei $D_k := D(\hat{\xi}^k)$ die Matrix aus Lemma 3.4.3, d.h. die Segmente von D_k sind für alle $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$ durch*

$$D_{k\mu_1\mu_2 j_1 j_2} = \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2} \sum_{i=1}^{R} \left\langle \alpha_i, \xi_{j_1}^k \right\rangle_{\mu_1\mu_2} \alpha_{i\mu_1}\alpha_{i\mu_2}^t$$

vorgegeben. Dann werden für eine Matrix-Vektor-Multiplikation mit D_k

$$r \cdot (4(d-1+R)-1) \cdot \sum_{\mu=1}^{d} t_\mu \tag{5.36}$$

Operationen benötigt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$ und $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$. Es folgt für die Komponenten von $D_k v$

$$\begin{aligned}(D_k v)_{j_1\mu_1} &= \sum_{\mu_2=1}^{d}\sum_{j_2=1}^{r} \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2} \sum_{i=1}^{R} \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} \alpha_{i\mu_1}\alpha^t_{i\mu_2} v_{j_2\mu_2} \\ &= \sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} \sum_{i=1}^{R} \left\langle \alpha_i, \xi^k_{j_1} \right\rangle_{\mu_1\mu_2} \langle \alpha_{i\mu_2}, v_{j_1\mu_2}\rangle \, \alpha_{i\mu_1}.\end{aligned}$$

Hierfür werden $(4(d-1+R)-1)\cdot t_{\mu_1}$ Rechenoperationen benötigt, so dass für alle Komponenten

$$r \cdot (4(d-1+R)-1) \cdot \sum_{\mu=1}^{d} t_\mu$$

arithmetische Operationen anfallen. ■

Getreu Lemma 3.4.4 sind die Segmente der Hesse-Matrix von g_1 an der Stelle ξ^k durch die Summe der folgenden zwei Matrizen festgelegt:

$$\begin{aligned}G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi}) &:= \delta_{\mu_1\mu_2}\delta_{j_1 j_2}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d} \left(\|\xi^k_{j_1\mu_1}\|^2 - \|\xi^k_{j_1\mu}\|^2\right) Id_{\mathbb{R}^{t_{\mu_1}}}\right. \\ &+ \left. 2(d-1)\xi^k_{j_1\mu_1}(\xi^k_{j_1\mu_1})^t\right], \\ G_{2\mu_1\mu_2 j_1 j_2}(\hat{\xi}) &:= \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}(-2)\xi^k_{j_1\mu_1}(\xi^k_{j_1\mu_2})^t,\end{aligned}$$

wobei $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$ sind. Gemäß Gleichung (5.5) erfüllen die Repräsentantenvektoren von ξ^k die Bedingung

$$\forall j \in \mathbb{N}_{\leq r} \forall \mu \in \mathbb{N}_{\leq d} \forall \nu \in \mathbb{N}_{\leq d} \setminus \{\mu\} : \|\xi^k{}_{j\mu}\| = \|\xi^k{}_{j\nu}\|.$$

Daher gilt für alle μ_1 und j_1

$$\sum_{\mu=1,\mu\neq\mu_1}^{d} \left(\|\xi^k_{j_1\mu_1}\|^2 - \|\xi^k_{j_1\mu}\|^2\right) = 0,$$

und die Segmente von G_1 vereinfachen sich zu

$$G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \delta_{j_1 j_2}\delta_{\mu_1\mu_2} 2(d-1)\xi^k_{j_1\mu_1}(\xi^k_{j_1\mu_1})^t. \tag{5.37}$$

Lemma 5.3.12. *Für eine Matrix-Vektor-Multiplikation von $G_1 + G_2$ an der Stelle ξ^k werden*

$$r \cdot \left[(2d+1)\sum_{\mu=1}^{d} t_\mu + d\right] \tag{5.38}$$

Operationen benötigt, wobei vorausgesetzt wird, dass ξ^k die Bedingung von Gleichung (5.5) erfüllt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$, $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$ und $G_k := G_1(\hat{\xi}^k) + G_2(\hat{\xi}^k)$ zur Abkürzung gesetzt. Für die Komponenten von $G_k v$ gilt

$$(G_k v)_{j_1\mu_1} = 2\left[(d-1)\left\langle \xi_{j_1\mu_1}, v_{j_1\mu_1}\right\rangle - \sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} \left\langle \xi_{j_1\mu_2}, v_{j_1\mu_2}\right\rangle\right] \xi_{j_1\mu_1}.$$

Hierfür werden $2\sum_{\mu=1}^{d} t_\mu + 1 + t_{\mu_1}$ Rechenoperationen benötigt, so dass für alle Komponenten

$$r \cdot \left[(2d+1)\sum_{\mu=1}^{d} t_\mu + d\right]$$

arithmetische Operationen anfallen. ■

Lemma 5.3.13. *Ist G_k die Hesse-Matrix von g_2 an der Stelle ξ^k, dann sind gemäß Lemma 3.4.5 die Segmente von G_k für alle $\mu_1, \mu_2 \in \mathbb{N}_{\leq d}$ und $j_1, j_2 \in \mathbb{N}_{\leq r}$ wie folgt gegeben:*

$$G_{k\mu_1\mu_2 j_1 j_2} := \underbrace{\frac{1}{\|\alpha\|^2}}_{=1} \delta_{j_1 j_2} \begin{cases} \left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, & \mu_1 = \mu_2 \\ 2\left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1\mu_2} \xi^k_{j_1\mu_1}(\xi^k_{j_1\mu_2})^t, & \textit{sonst.} \end{cases}$$

Für eine Matrix-Vektor-Multiplikation mit G_k werden

$$r \cdot \sum_{\mu_1=1}^{d} \left[3t_{\mu_1} + \sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} 2t_{\mu_2} + d - 1\right] \tag{5.39}$$

Operationen benötigt.

Beweis. Seien $\mu_1 \in \mathbb{N}_{\leq d}$, $j_1 \in \mathbb{N}_{\leq r}$ und $v := (v_{j\mu} \in \mathbb{R}^{t_\mu} : \mu \in \mathbb{N}_{\leq d}, j \in \mathbb{N}_{\leq r}) \in \mathfrak{R}_{d,r,\underline{t}}$. Für die Komponenten von $G_k v$ gilt

$$\begin{aligned}(G_k v)_{j_1\mu_1} &= \left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1} v_{j_1\mu_1} \\ &+ \left[2\sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} \left\langle \xi^k_{j_1}, \xi^k_{j_1}\right\rangle_{\mu_1\mu_2} \left\langle \xi^k_{j_1\mu_2}, v_{j_1\mu_2}\right\rangle\right] \xi^k_{j_1\mu_1}.\end{aligned}$$

Für eine Komponente werden

$$\sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} 2t_{\mu_2} + d - 1 + 3t_{\mu_1}$$

Rechenoperationen benötigt, so dass für alle Komponenten

$$r \cdot \sum_{\mu_1=1}^{d} \left[3t_{\mu_1} + \sum_{\mu_2=1,\mu_2\neq\mu_1}^{d} 2t_{\mu_2} + d - 1\right]$$

arithmetische Operationen anfallen. ■

Summiert man nun alle Rechenoperationen auf, dann erhält man für die Komplexität einer Matrix-Vektor-Multiplikation mit der Matrix

$$C_k(\alpha_k) = \begin{cases} A_k + \alpha_k \left(B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right), & \overline{\alpha_k} \neq 1 \\ A_k + \alpha_k \left(B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right), & \overline{\alpha_k} = 1 \end{cases}$$

den folgenden Gesamtaufwand.

Korollar 5.3.14. *Die Komplexität der Matrix-Vektor-Multiplikation mit* $A_k + \alpha_k \left(B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right)$ *bzw.* $A_k + \alpha_k \left(B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}\right)$ *beträgt*

$$\mathcal{O}\left(r \cdot (d+r) \cdot \sum_{\mu=1}^{d} t_\mu\right) \tag{5.40}$$

bzw.

$$\mathcal{O}\left(r \cdot (d+r+R) \cdot \sum_{\mu=1}^{d} t_\mu\right). \tag{5.41}$$

5.3.3 Bestimmung der Schrittweite

Für die globale Konvergenz eines Minimierungsverfahrens ist die Berechnung der Schrittweite mittels Armijo-Regel von Bedeutung. Hierbei wird die Schrittweite wie folgt ermittelt: Finde $\bar{\alpha}_k := \max\{\beta^l : l \in \mathbb{N}_{\geq 0}\} \in (0,1]$ mit

$$f(\hat{\xi}^k) - f(\hat{\xi}^k - \bar{\alpha}_k d^k) \geq \sigma \bar{\alpha}_k \left\langle f'(\hat{\xi}^k), d^k \right\rangle,$$

dabei sind $\beta \in (0,1)$ und $\sigma \in (0, \frac{1}{2})$. Maßgebend für die Komplexität ist die mehrfache Berechnung des Ausdrucks $f(\hat{\xi}^k - \beta^l d^k)$, wobei $l \in \mathbb{N}_{\leq 0}$ variiert und β, $\hat{\xi}^k$ und d^k vorgegeben sind. Setzt man $\hat{\xi}^k - \beta^l d^k$ in f ein, dann erhält man

$$\begin{aligned} f(\hat{\xi}^k - \beta^l d^k) \;=\; & \underbrace{\frac{1}{\|\alpha\|^2}}_{=1} \Bigg[- \sum_{j=1}^{r} \sum_{i=1}^{R} \prod_{\mu=1}^{d} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \alpha_{i\mu} \right\rangle \\ & + \sum_{j=1}^{r} \left(\sum_{j'=1}^{j-1} \prod_{\mu=1}^{d} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \xi_{j'\mu}^k - \beta^l d_{j'\mu}^k \right\rangle \right. \\ & \left. + \underbrace{\frac{1+\lambda_2}{2}}_{C:=} \prod_{\mu=1}^{d} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \xi_{j\mu}^k - \beta^l d_{j\mu}^k \right\rangle \right) \Bigg] \\ & + \frac{\lambda_1}{8 \underbrace{\sqrt[d]{\|\alpha\|^4}}_{=1}} \sum_{j=1}^{r} \sum_{\mu,\nu=1,\nu\neq\mu}^{d} \left(\|\xi_{j\mu}^k - \beta^l d_{j\mu}^k\|^2 - \|\xi_{j\nu}^k - \beta^l d_{j\nu}^k\|^2 \right). \end{aligned}$$

Sind $\mu, \nu \in \mathbb{N}_{\leq d}$, $j, j' \in \mathbb{N}_{\leq r}$, $i \in \mathbb{N}_{\leq R}$ und setzt man ferner die Bedingung von Gleichung (5.5) voraus, dann folgt

$$\begin{aligned} \|\xi_{j\mu}^k - \beta^l d_{j\mu}^k\|^2 - \|\xi_{j\nu}^k - \beta^l d_{j\nu}^k\|^2 &= -2\beta^l \left(\left\langle \xi_{j\mu}^k, d_{j\mu}^k \right\rangle - \left\langle \xi_{j\nu}^k, d_{j\nu}^k \right\rangle \right) \\ &+ \beta^{2l} \left(\|d_{\mu j}^k\|^2 - \|d_{\nu j}^k\|^2 \right), \end{aligned}$$

$$\left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \alpha_{i\mu} \right\rangle = \left\langle \xi_{j\mu}^k, \alpha_{i\mu} \right\rangle - \beta^l \left\langle d_{j\mu}^k, \alpha_{i\mu} \right\rangle$$

und

$$\begin{aligned} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \xi_{j'\mu}^k - \beta^l d_{j'\mu}^k \right\rangle &= \left\langle \xi_{j\mu}^k, \xi_{j'\mu}^k \right\rangle - \beta^l \left[\left\langle \xi_{j\mu}^k, d_{j'\mu}^k \right\rangle + \left\langle \xi_{j'\mu}^k, d_{j\mu}^k \right\rangle \right] \\ &+ \beta^{2l} \left\langle d_{j\mu}^k, d_{j'\mu}^k \right\rangle. \end{aligned}$$

Ähnlich wie im Abschnitt 5.3.1.1 lohnt es sich auch hier, die von l unabhängigen Skalarprodukte

$$\left\langle d_{j\mu}^k, \alpha_{i\mu} \right\rangle, \ \left\langle \xi_{j\mu}^k, d_{j'\mu}^k \right\rangle \text{ und } \left\langle d_{j\mu}^k, d_{j'\mu}^k \right\rangle$$

vorab zu berechnen und zu speichern. Der hierfür zusätzlich benötigte Speicherbedarf von

$$d \cdot \left[r \cdot R + 2 \cdot \binom{r+1}{2} \right] \tag{5.42}$$

Einträgen ist von moderater Größe. Der Rechenaufwand für ein Skalarprodukt beträgt $2t_\mu - 1$ Rechenoperationen, so dass insgesamt

$$\left[r \cdot R + 2 \cdot \binom{r+1}{2} \right] \cdot \sum_{\mu=1}^{d} (2t_\mu - 1) \tag{5.43}$$

Operationen anfallen. Die Skalarprodukte $\left\langle \xi_{j\mu}^k, \alpha_{i\mu} \right\rangle$ und $\left\langle \xi_{j\mu}^k, \xi_{j'\mu}^k \right\rangle$ müssen nicht erneut berechnet werden, denn im Abschnitt 5.3.1.1 wurden sie bereits bestimmt. Bei der anschließenden Komplexitätsanalyse kann die Berechnung der oben erwähnten Skalarprodukte vorausgesetzt werden.

Lemma 5.3.15. *Sei $l \in \mathbb{N}_{\leq 0}$. Die Anzahl der Rechenoperationen, um $f(\hat{\xi}^k - \beta^l d^k)$ zu berechnen, ist von der Ordnung*

$$\mathcal{O}\left(d \cdot r \cdot (r + R + d)\right). \tag{5.44}$$

Beweis. Für die Berechnung von

$$\sum_{j=1}^{r} \sum_{i=1}^{R} \prod_{\mu=1}^{d} \left(\left\langle \xi_{j\mu}^k, \alpha_{i\mu} \right\rangle - \beta^l \left\langle d_{j\mu}^k, \alpha_{i\mu} \right\rangle \right)$$

benötigt man

$$4d \cdot R \cdot r - 1$$

arithmetische Operationen und für

$$\sum_{j=1}^{r} \left(\sum_{j'=1}^{j-1} \prod_{\mu=1}^{d} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \xi_{j'\mu}^k - \beta^l d_{j'\mu}^k \right\rangle + C \prod_{\mu=1}^{d} \left\langle \xi_{j\mu}^k - \beta^l d_{j\mu}^k, \xi_{j\mu}^k - \beta^l d_{j\mu}^k \right\rangle \right)$$

fallen

$$\binom{r}{2} \cdot (8d-1) + r \cdot (8d+1) - 1$$

Operationen an. Bei der Berechnung des letzten Ausdrucks

$$\lambda_1 \sum_{j=1}^{r} \sum_{\mu=1}^{d} \sum_{\nu=1, \nu \neq \mu}^{d} \left[-2\beta^l \left(\left\langle \xi_{j\mu}^k, d_{j\mu}^k \right\rangle - \left\langle \xi_{j\nu}^k, d_{j\nu}^k \right\rangle \right) + \beta^{2l} \left(\|d_{\mu j}^k\|^2 - \|d_{\nu j}^k\|^2 \right) \right]$$

werden dann noch Rechenoperationen in der Größenordnung von

$$\mathcal{O}(d^2 \cdot r)$$

benötigt, so dass insgesamt die Behauptung folgt. ■

Korollar 5.3.16. *Die Gesamtkomplexität zur Berechnung der Schrittweite α_k mit Hilfe der Armijo-Regel ist von der Ordnung*

$$\mathcal{O}\left(r \cdot \left[(r+R) \cdot \sum_{\mu=1}^{d} t_\mu + d \cdot (r+R+d) \right] \right). \tag{5.45}$$

5.3.4 Zusammenfassung

An dieser Stelle sei die in den vorangegangenen Abschnitten untersuchte Komplexität eines Minimierungsschrittes zusammengefasst.
Zum Anfang wurden die Skalarprodukte der Repräsentantenvektoren berechnet, hierfür benötigt man

$$\mathcal{O}\left(r \cdot (r+R) \cdot \sum_{\mu=1}^{d} t_\mu \right)$$

Operationen. Im Anschluss daran mussten der Gradient von f und die Skalarprodukte mit Auslassung, für die Hesse-Matrix, berechnet werden. Die Komplexitätsordnung für den Gradienten betrug

$$\mathcal{O}\left(r \cdot (r+R) \cdot \sum_{\mu=1}^{d} t_\mu + d \cdot r \cdot R \right),$$

dies entspricht der Aussage von Lemma 5.3.2. Für die Hesse-Matrix waren

$$\mathcal{O}\left(d^3 \cdot r \cdot (r+R)\right)$$

Rechenoperationen nötig, siehe Lemma 5.3.1. Danach wurde die Komplexität einer Vektorauswertung mit der Systemmatrix und das Berechnen des Vorkonditionieres analysiert. Für den Vorkonditionier werden

$$\mathcal{O}\left(d \cdot r^3\right)$$

und je nach Wahl der Systemmatrix

$$\mathcal{O}\left(r \cdot (d+r) \cdot \sum_{\mu=1}^{d} t_\mu\right)$$

bzw.

$$\mathcal{O}\left(r \cdot (d+r+R) \cdot \sum_{\mu=1}^{d} t_\mu\right)$$

Rechenoperationen benötigt, siehe Korollar 5.3.14. Abschließend sind gemäß Korollar 5.3.16

$$\mathcal{O}\left(r \cdot \left[(r+R) \cdot \sum_{\mu=1}^{d} t_\mu + d \cdot (r+R+d)\right]\right)$$

Operationen zur Berechnung der Schrittweite nötig. Zusammenfassend erhält man folgendes Korollar 5.3.17.

Korollar 5.3.17. *Die Gesamtkomplexität eines Minimierungsschrittes ist von der Ordnung*

$$\mathcal{O}\left(r \cdot (r+R) \cdot d^3 + d \cdot r^3 + r \cdot (r+R+d) \cdot \sum_{\mu=1}^{d} t_\mu\right). \tag{5.46}$$

Bemerkung 5.3.18. *Die Anzahl der nötigen Minimierungsschritte hängt sehr stark von der Wahl des Startwertes ab und kann bei einer ungünstigen Wahl zu einem deutlich erhöhten Aufwand führen. Die zufällige Wahl des Startwertes, wie in [33] beschrieben, ist in der Praxis nicht zu empfehlen.*
Im folgenden Abschnitt 5.4 wird eine andere Wahl vorgeschlagen, welche bei den numerischen Tests zu einer geringen Anzahl von Iterationsschritten führte, siehe dazu Kapitel 6. In Abhängigkeit vom Startwert kann aber im Allgemeinen die Anzahl der nötigen Minimierungsschritte nicht abgeschätzt werden.

5.4 Lösung der erweiterten Approximationsaufgabe

Getreu Definition 3.1.2, Seite 38, ist die erweiterte Approximationsaufgabe, unter Vernachlässigung der Nebenbedingung, wie folgt gestellt: Sei $\varepsilon \in \mathbb{R}_{>0}$ fest vorgegeben. Zu bestimmen ist ein $\xi_\varepsilon \in \mathcal{S}_{r_\varepsilon}$ derart, dass

$$\begin{aligned} \|\alpha - \xi_\varepsilon\| &\leq \varepsilon, \\ \|\alpha - \xi_\varepsilon\| &= \min_{\xi \in \mathcal{S}^{\leq}_{r_\varepsilon} \cap U(\xi_\varepsilon)} \|\alpha - \xi\| \end{aligned}$$

erfüllt sind, wobei $r_\varepsilon \in \mathbb{N}_{\leq R}$ kleinstmöglich sein soll.
Um diese Aufgabe zu lösen, wird zuerst eine beste Approximation $\xi_1 \in \mathcal{S}_1$ von α berechnet; dazu benutzt man das beschriebene Verfahren. Als Startwert konstruiert man hierfür einen Elementartensor, welcher α auf einem „Kreuz“ interpoliert, siehe dazu Lemma 5.4.2. Ist der Abstand klein genug, d.h. $\|\alpha - \xi_1\| \leq \varepsilon$, so wird das Verfahren abgebrochen. In der Regel wird dies kaum der Fall sein, daher muss der Tensorrank sukzessive erhöht werden. Ausgehend von einer bereits berechneten besten Approximation $\xi_r \in \mathcal{S}_r$ von α ist das Residuum

$$\varrho_r := \alpha - \xi_r \in \mathcal{S}_{\leq R+r} \tag{5.47}$$

definiert. Man berechnet nun eine beste Approximation ζ_1 von ϱ_r in $\mathcal{S}_1$ und setzt dann anschließend den Startwert $\xi^{(0)}_{r+1} \in \mathcal{S}_{r+1}$ für die beste Approximation $\xi_{r+1} \in \mathcal{S}_{r+1}$ von α wie folgt:

$$\xi^{(0)}_{r+1} := \xi_r + \zeta_1. \tag{5.48}$$

Eine exakte Beschreibung dieser Herangehensweise ist im Algorithmus 5.4.1 dargestellt. Zuvor wird jedoch die Berechnung einer Kreuzinterpolation im Lemma 5.4.2 verdeutlicht.

Definition 5.4.1 (Streifen und Kreuz)**.** *Seien $\underline{i} := (i_1, \ldots, i_d) \in \times_{\mu=1}^{d} \mathbb{N}_{\leq t_\mu}$ ein Multiindex und $\mu \in \mathbb{N}_{\leq d}$. Unter dem* Streifen *von $\underline{i}$ in Richtung μ versteht man folgende Menge:*

$$\underline{i}^{\mu} := \left(\bigtimes_{\nu=1}^{\mu-1} \{i_\nu\} \right) \times \mathbb{N}_{\leq t_\mu} \times \left(\bigtimes_{\nu=\mu+1}^{d} \{i_\nu\} \right). \tag{5.49}$$

Daneben ist das Kreuz *von $\underline{i}$ wie folgt definiert:*

$$\kappa^{\underline{i}} := \bigcup_{\mu=1}^{d} \underline{i}^{\mu}. \tag{5.50}$$

Lemma 5.4.2. *Seien $k \in \mathbb{N}$, $\beta := \sum_{j=1}^{k} \bigotimes_{\mu=1}^{d} \beta_{j\mu} \in \mathcal{S}_k$ und $\underline{i} \in \times_{\mu=1}^{d} \mathbb{N}_{\leq t_\mu}$, etwa $\underline{i} := (i_1, \ldots, i_d)$, mit*

$$\beta_{\underline{i}} \in \mathbb{R} \setminus \{0\}. \tag{5.51}$$

Ferner sei

$$\xi^{\underline{i}} := \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \bigotimes_{\mu=1}^{d} \xi^{\underline{i}}_{\mu} \in \mathcal{S}_1 \tag{5.52}$$

gesetzt, wobei für alle $\mu \in \mathbb{N}_{\leq d}$ *die Vektoren* $\xi_\mu^{\underline{i}} \in \mathbb{R}^{t_\mu}$ *wie folgt definiert sind:*

$$\xi_\mu^{\underline{i}} := \sum_{j=1}^{k} \left(\prod_{\nu=1,\nu\neq\mu}^{d} (\beta_{j\nu})_{i_\nu} \right) \beta_{j\mu}. \tag{5.53}$$

Dann gilt für alle $\underline{m} \in \kappa^{\underline{i}}$

$$\xi^{\underline{i}}(\underline{m}) = \beta(\underline{m}). \tag{5.54}$$

Beweis. Seien $\underline{m} \in \kappa^{\underline{i}}$ und o. B. d. A. $\underline{m} := (i_1, \ldots, i_{d-1}, l_d)$, $l_d \in \mathbb{N}_{\leq t_d}$. Es gilt

$$\begin{aligned}
\xi_{\underline{m}}^{\underline{i}} &= \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \prod_{\mu=1}^{d-1} \left[\sum_{j=1}^{k} \prod_{\nu=1,\nu\neq\mu}^{d} (\beta_{j\nu})_{i_\nu} (\beta_{j\mu})_{i_\mu} \right] \left[\sum_{j=1}^{k} \prod_{\nu=1}^{d-1} (\beta_{j\nu})_{i_\nu} (\beta_{jd})_{l_d} \right] \\
&= \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \left[\beta_{\underline{i}}\right]^{d-1} \left[\sum_{j=1}^{k} \prod_{\nu=1}^{d-1} (\beta_{j\nu})_{i_\nu} (\beta_{jd})_{l_d} \right] \\
&= \sum_{j=1}^{k} \prod_{\nu=1}^{d-1} (\beta_{j\nu})_{i_\nu} (\beta_{jd})_{l_d} \\
&= \beta_{\underline{m}}.
\end{aligned}$$

■

Bemerkung 5.4.3. *Bei der in Gleichung (5.52) definierten Kreuzinterpolation von* β *kann*

$$\|\beta - \xi^{\underline{i}}\| \geq \|\beta\| \tag{5.55}$$

auftreten. In diesem Fall würde der Nulltensor β *besser approximieren als* $\xi^{\underline{i}}$*. Dieser Umstand ist aber durch eine Skalierung von* $\xi^{\underline{i}}$ *mit* $\left\langle \beta, \xi^{\underline{i}} \right\rangle / \|\xi^{\underline{i}}\|^2$ *zu beheben, denn es gilt*

$$\begin{aligned}
\left\| \beta - \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle}{\|\xi^{\underline{i}}\|^2} \xi^{\underline{i}} \right\|^2 &= \|\beta\|^2 - 2\frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^2} + \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^4} \|\xi^{\underline{i}}\|^2 \\
&= \|\beta\|^2 - \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^2} \\
&\leq \|\beta\|^2.
\end{aligned}$$

Bemerkung 5.4.4. *Getreu Korollar 2.3.6, Seite 35, ist* $\mathcal{S}_1$ *kompakt, daher wird bei der Berechnung einer besten Approximation in* $\mathcal{S}_1$ *die zweite Nebenbedingung ausgeschaltet, d.h. in der Definition der Zielfunktion* f*, siehe Gleichung (3.13), Seite 40, ist dann* $\lambda_2 = 0$*.*
Bei der Berechnung einer besten Approximation existieren unterschiedliche lokale Minima, deshalb sollten ungleiche Startwerte nacheinander ausprobiert werden. Dies wird aber zu hohen Kosten führen. Aus diesem Grund beschränkt man sich nur bei der Berechnung von ζ_1 *auf mehrere Startwerte und wählt dann den Elementartensor aus, welcher* ϱ_r *am besten approximiert.*

Algorithmus 5.4.1 Berechnung einer optimalen ε-Approximation

1: Wähle $\varepsilon \in \mathbb{R}_{>0}$ und setze $r := 1$.
2: Bestimme eine Kreuzinterpolation $\xi^{\underline{i}}$ von α in $\mathcal{S}_1$.
Berechne eine lokal beste Approximation ξ_r von α in $\mathcal{S}_1$ mittels Alg. 4.4.1 angewandt auf f, $\lambda_2 = 0$, und mit Hilfe des Startwertes $\xi^{\underline{i}}$.
Setze $\varrho_r := \alpha - \xi_r$.
3: **while** $\|\varrho_r\| > \varepsilon$ und $r < R$ **do**
4: Bestimme eine Kreuzinterpolation $\zeta^{\underline{i}}$ von ϱ_r in $\mathcal{S}_1$.
Berechne eine lokal beste Approximation ζ_1 von ϱ_r in $\mathcal{S}_1$ mittels Alg. 4.4.1 angewandt auf f, $\lambda_2 = 0$, und mit Hilfe des Startwertes $\zeta^{\underline{i}}$.
5: Setze $\xi_{r+1}^{(0)} := \xi_r + \zeta_1$.
Berechne eine lokal beste Approximation ξ_{r+1} von α in $\mathcal{S}_{r+1}^c$ mittels Alg. 4.4.1 angewandt auf f und mit Hilfe des Startwertes $\xi_{r+1}^{(0)}$.
6: Setze $\varrho_{r+1} := \alpha - \xi_{r+1}$ und $r \leftarrow r + 1$.
7: **end while**
8: **if** $r = R$ **then**
9: Setze $\xi_r := \alpha$.
10: **end if**

Lemma 5.4.5. *Es seien die Bezeichnungen von Algorithmus 5.4.1 und 5.4.2 vorausgesetzt. Die Startwerte $\xi_{r+1}^{(0)} \in \mathcal{S}_{r+1}$ und die lokalen besten Approximationen $\xi_r \in \mathcal{S}_r$ von α erfüllen dann für alle $r \leq R$ folgende Ungleichung:*

$$\|\alpha - \xi_{r+1}^{(0)}\| \leq \|\alpha - \xi_r\|. \tag{5.56}$$

Beweis. Sei $r \leq R$. Es gilt

$$\|\alpha - \xi_{r+1}^{(0)}\| = \|\alpha - \xi_r - \zeta_1\| = \|\varrho_r - \zeta_1\|.$$

Hiermit folgt gemäß Lemma 2.1.10, Seite 27, und Bemerkung 1.4.4, Seite 16,

$$\|\varrho_r - \zeta_1\| \leq \|\varrho_r\| = \|\alpha - \xi_r\|.$$

■

Bei späteren Anwendungen ist häufig eine gute Approximation $\tilde{\xi}$ von α bekannt, wobei $\operatorname{rang}_{\mathcal{S}}(\tilde{\xi}) \geq 1$ sein kann. In diesen Fällen wird der Algorithmus 5.4.1 abgeändert, siehe Algorithmus 5.4.2.

Bemerkung 5.4.6. *Setzt man voraus, dass die Anzahl der Iterationsschritte von Algorithmus 4.4.1 konstant ist, dann hat das Verfahren aus Algorithmus 5.4.1 bzw. 5.4.2, gemäß Korollar 5.3.17, eine Komplexität der Ordnung*

$$\mathcal{O}\left(\sum_{r=1}^{r_\varepsilon}\left[r\cdot(r+R)\cdot d^3 + d\cdot r^3 + r\cdot(r+R+d)\cdot\sum_{\mu=1}^{d} t_\mu\right]\right) \tag{5.57}$$

bzw.

$$\mathcal{O}\left(\sum_{r=\operatorname{rang}_{\mathcal{S}}(\tilde{\xi})}^{r_\varepsilon}\left[r\cdot(r+R)\cdot d^3 + d\cdot r^3 + r\cdot(r+R+d)\cdot\sum_{\mu=1}^{d} t_\mu\right]\right). \tag{5.58}$$

Algorithmus 5.4.2 Berechnung einer optimalen ε-Approximation bei bekannter guter Näherung $\tilde{\xi}$

1: Wähle $\varepsilon \in \mathbb{R}_{>0}$ und setze $r := \mathrm{rang}_{\mathcal{S}}(\tilde{\xi})$.
2: Berechne eine lokal beste Approximation ξ_r von α in $\mathcal{S}_r$ mittels Alg. 4.4.1 angewandt auf f und mit Hilfe des Startwertes $\tilde{\xi}$.
Setze $\varrho_r := \alpha - \xi_r$.
3: **while** $\|\varrho_r\| > \varepsilon$ und $r < R$ **do**
4: Bestimme eine Kreuzinterpolation $\zeta^{\underline{i}}$ von ϱ_r in $\mathcal{S}_1$.
Berechne eine lokal beste Approximation ζ_1 von ϱ_r in $\mathcal{S}_1$ mittels Alg. 4.4.1 angewandt auf f, $\lambda_2 = 0$, und mit Hilfe des Startwertes $\zeta^{\underline{i}}$.
5: Setze $\xi_{r+1}^{(0)} := \xi_r + \zeta_1$.
Berechne eine lokal beste Approximation ξ_{r+1} von α in $\mathcal{S}_{r+1}^{c}$ mittels Alg. 4.4.1 angewandt auf f und mit Hilfe des Startwertes $\xi_{r+1}^{(0)}$.
6: Setze $\varrho_{r+1} := \alpha - \xi_{r+1}$ und $r \leftarrow r + 1$.
7: **end while**
8: **if** $r = R$ **then**
9: Setze $\xi_r := \alpha$.
10: **end if**

Notation 5.4.7. *Seien $\alpha \in \mathcal{T}_R$ und $\varepsilon \in \mathbb{R}_{>0}$. Eine mittels Algorithmus 5.4.1 oder 5.4.2 berechnete ε-Approximation von α wird künftig mit*

$$\mathfrak{App}_{\varepsilon}(\alpha) \tag{5.59}$$

bezeichnet.

6 Anwendungen

In diesem Kapitel soll anhand ausgewählter Anwendungen die vorgestellte Methode untersucht und das Potenzial von Elementartensor-Summen verdeutlicht werden.
Im ersten Beispiel wird eine partielle Differenzialgleichung mit variierender hoher Dimension untersucht. Elementartensor-Summen dieser Art wurden bereits in [11] sehr effizient mit Hilfe von hierarchischen Matrizen in sehr hohen Dimensionen generiert, daneben wurden zum Beispiel in [30] und [11] die Existenz der Inversen von weiter gefassten Operatoren bewiesen. Die hier vorgestellte Methode unterscheidet sich in der grundsätzlichen Vorgehensweise kaum. Der Unterschied liegt in einer verbesserten Approximation der Funktion

$$\begin{aligned} \widetilde{\varphi} : [1, R] &\to \mathbb{R} \\ t &\mapsto \frac{1}{t} \end{aligned}$$

mittels Exponentialsummen, wie in [3] beschrieben. Darüber hinaus steht hier nicht die effiziente Erzeugung im Vordergrund, sondern die anschließende Approximation der angenäherten Lösung in Abhängigkeit vom Modellfehler.
Im zweitem Abschnitt werden iterative Verfahren mit Elementartensor-Summen diskutiert. Insbesondere werden die Berechnung der Maximumnorm und der punktweisen Inversen in sehr hohen Dimensionen vorgestellt. Die hierbei verwendeten iterativen Verfahren führen zu einem stetigen Anstieg des Tensorrangs, so dass eine Tensorrang-Approximation der Folgenglieder notwendig wird. Die Konvergenz des dadurch erzeugten inexakten Iterationverfahrens wird untersucht, hierbei werden die Resultate von [18] wiedergegeben.
Abschließend werden die in Abschnitt 5.3.2.1 definierten unterschiedlichen Abstiegsrichtungen verglichen.
Alle Rechnungen wurden auf einem Intel Core 2 Duo Prozessor T7300 mit 2,0 GHz durchgeführt.

6.1 Das Modellproblem

In der folgenden Analyse werden Summen von Elementartensoren aus dem Umfeld partieller Differenzialgleichungen in hohen Dimensionen approximiert. Für die Generierung dieser Beispiele soll folgendes Modellproblem dienen.

Definition 6.1.1 (Modellproblem)**.** *Seien $d \in \mathbb{N}_{\geq 3}$ und $\Omega := (0,1)^d \subset \mathbb{R}^d$. Unter dem* Modellproblem *wird folgende Aufgabe verstanden: Auf dem Gebiet Ω soll die Poisson[1]-Gleichung*

$$-\Delta u = h \qquad \left(\Delta := \sum_{\mu=1}^{d} \partial_{x_\mu}^2 \right), \tag{6.1}$$

mit Dirichlet[2]-Bedingung

$$u|_{\partial\Omega} = 0 \tag{6.2}$$

für folgendes h gelöst werden,

$$\begin{aligned} h : \overline{\Omega} &\to \mathbb{R} \\ \underline{x} := (x_1, \ldots, x_d) &\mapsto \sum_{\mu=1}^{d} \prod_{\nu=1, \nu\neq\mu}^{d} \varphi(x_\nu) \left(-2 + (4 - 12x_\mu) \prod_{\nu=1, \nu\neq\mu}^{d} 2x_\nu \right), \end{aligned} \tag{6.3}$$

wobei

$$\begin{aligned} \varphi : [0,1] &\to \mathbb{R} \\ t &\mapsto \varphi(t) := (1-t)t \end{aligned} \tag{6.4}$$

gesetzt ist.

Bemerkung 6.1.2. *Die Funktion h aus Gleichung (6.3) wurde so gewählt, dass die folgende Funktion u mit Tensorrang 2 die Lösung des Modellproblems ist*

$$\begin{aligned} u : \overline{\Omega} &\to \mathbb{R} \\ \underline{x} := (x_1, \ldots, x_d) &\mapsto u(\underline{x}) := \prod_{\nu=1}^{d} \varphi(x_\nu) \left(1 + \prod_{\nu=1}^{d} 2x_\nu \right). \end{aligned} \tag{6.5}$$

Um zusätzliche Nebeneffekte zu vermeiden, wurde ferner darauf geachtet, dass man u schlecht durch Elementartensoren approximieren kann.

Die hier untersuchte hochdimensionale partielle Differenzialgleichung ist von relativ einfacher Art. Dennoch ist dieses Problem sehr gut für eine erste Analyse des Approximationsverfahrens geeignet, denn:

- Nach der Diskretisierung haben sowohl h als auch Δ das notwendige Datenformat. Ferner kann man Δ^{-1} mit Hilfe von Elementartensor-Summen sehr gut approximieren, siehe dazu Lemma 6.1.6.
- Durch Vorgabe der Lösung u ist der Modellfehler explizit gegeben und lässt sich mit Hilfe der Approximationsgüte steuern. Damit kann der Algorithmus speziell bei variierenden (bekannten) Modellfehlern untersucht werden.

[1] Siméon-Denis Poisson, französischer Physiker und Mathematiker, $*$ 21. Juni 1781 in Pithiviers, † 25. April 1840 in Paris.

[2] Peter Gustav Lejeune Dirichlet, deutscher Mathematiker, $*$ 13. Februar 1805 in Düren, † 5. Mai 1859 in Göttingen.

6.1.1 Diskretisierung und Approximation der Inversen

Die Diskretisierung erfolgt auf einem uniformen Gitter $\Gamma_n \subset \Omega$ mit $n \in \mathbb{N}$ Freiheitsgraden in jeder Raumrichtung, d.h.

$$\Gamma_n := \left\{ \underline{x} \in \Omega \mid \exists \underline{i} \in (\mathbb{N}_{\leq n})^d : \underline{x} = \eta \cdot \underline{i} \right\}, \tag{6.6}$$

wobei $\eta := \frac{1}{n+1} \in \mathbb{R}_{>0}$ gesetzt ist. Daneben vereinbart man, dass die Einträge der Gitterpunkte $\underline{x} \in \Gamma_n$ lexikographisch angeordnet sind. Ferner wird das Modellproblem auf diesem Gitter mit Hilfe der Differenzenmethode diskretisiert. Das hierdurch erzeugte lineare Gleichungssystem wird mit

$$\mathbf{L}\underline{u} = \underline{h} \tag{6.7}$$

bezeichnet.

Lemma 6.1.3. *Sei* $\mathcal{S} := \otimes^d \mathbb{R}^n$.

(i) Es seien die Einträge von $\underline{1}, \underline{\chi}, \underline{\varphi}, \underline{\psi} \in \mathbb{R}^n$ *für alle* $l \in \mathbb{N}_{\leq n}$ *wie folgt definiert:*

$$\underline{1}_l = 1, \tag{6.8}$$
$$\underline{\chi}_l = 4 - 12t_l, \tag{6.9}$$
$$\underline{\varphi}_l = \varphi(t_l), \tag{6.10}$$
$$\underline{\psi}_l = 2t_l\varphi(t_l), \tag{6.11}$$

wobei $t_l := l\eta \in (0,1)$ *gesetzt ist. Ferner seien*

$$b_1 := \bigotimes_{\nu=1}^{d} \underline{\varphi} \in \mathcal{S}_1 \quad \text{und} \quad b_2 := \bigotimes_{\nu=1}^{d} \underline{\psi} \in \mathcal{S}_1.$$

Es gilt dann

$$\underline{h} = \sum_{\mu=1}^{d} b_1^\mu(\underline{1}) + \sum_{\mu=1}^{d} b_2^\mu(\underline{\chi}) \in \mathcal{S}_{\leq 2\cdot d}, \tag{6.12}$$

wobei $b_1^\mu(\underline{1})$ *und* $b_2^\mu(\underline{\chi})$ *wie in Notation 1.4.10, Seite 20, definiert sind.*

(ii) Seien

$$\mathbf{T} := \frac{1}{\eta^2} \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n}$$

und

$$\mathbf{Id}_{\mathcal{S}} := \bigotimes_{\mu=1}^{d} \mathbf{Id}_{\mathbb{R}^n},$$

dann gilt

$$\mathbf{L} = \sum_{\mu=1}^{d} \mathbf{Id}_{\mathcal{S}}^{\mu}(\mathbf{T}). \tag{6.13}$$

Beweis.
(i): Folgt unmittelbar aus Gleichung (6.3).

(ii): Seien $\underline{i} := (i_1, \ldots, i_d) \in (\mathbb{N}_{\leq n})^d$ ein Multiindex und $u \in \mathcal{S}$. Es gilt für die Einträge von $\mathbf{L}u$

$$\begin{aligned}
(\mathbf{L}u)_{\underline{i}} &= \sum_{\mu=1}^{d} \sum_{j_\mu=1}^{n} \mathbf{T}_{(i_\mu, j_\mu)} u_{(i_1,\ldots,j_\mu,\ldots,i_d)} \\
&= \sum_{j_1,\ldots,j_d=1}^{n} \sum_{\mu=1}^{d} \delta_{i_1,j_1} \cdots \delta_{i_{\mu-1},j_{\mu-1}} \mathbf{T}_{(i_\mu,j_\mu)} \delta_{i_{\mu+1},j_{\mu+1}} \cdots \delta_{i_d,j_d} u_{(j_1,\ldots,j_d)} \\
&= \sum_{j_1=1}^{n} \cdots \sum_{j_d=1}^{n} \left(\sum_{\mu=1}^{d} \mathbf{Id}_{\mathcal{S}}^{\mu}(\mathbf{T}) \right)_{((i_1,\ldots,i_d),(j_1,\ldots,j_d))} u_{(j_1,\ldots,j_d)}.
\end{aligned}$$

■

Gemäß Lemma 6.1.3 sind $\underline{h}$ und $\mathbf{L}$ Summen von Elementartensoren. Von größerer Bedeutung ist jedoch die Frage nach der Existenz einer tensoriellen Darstellung von $\mathbf{L}^{-1}$. Mit Hilfe der folgenden Approximationsaussage kann diese interessante Frage beantwortet werden.

Lemma 6.1.4. *Seien $R \in \mathbb{R}_{>0}$ und*

$$\begin{aligned}
\widetilde{\varphi} : [1, R] &\to \mathbb{R} \\
t &\mapsto \frac{1}{t}.
\end{aligned}$$

Ferner seien $k \in \mathbb{N}$, $\underline{\alpha} := (\alpha_1, \ldots, \alpha_k)^t$, $\underline{\omega} := (\omega_1, \ldots, \omega_k)^t \in \mathbb{R}^k$,

$$\begin{aligned}
s_k : [1, R] &\to \mathbb{R} \\
t &\mapsto s_k(t) := \sum_{j=1}^{k} \omega_j \exp(-\alpha_j t),
\end{aligned}$$

und

$$E_k(\underline{\alpha}, \underline{\omega}) := \|\widetilde{\varphi} - s_k\|_{\infty,[1,R]} := \max\{|f(t) - s_k(t)| : t \in [1, R]\}.$$

Dann existieren $\underline{\alpha}^*, \underline{\omega}^* \in \mathbb{R}^k$ und $c_1, c_2 \in \mathbb{R}_{>0}$ mit

$$E_k(\underline{\alpha}^*, \underline{\omega}^*) = \min_{\underline{\alpha},\underline{\omega} \in \mathbb{R}^k} E_k(\underline{\alpha}, \underline{\omega}) \leq c_1 \exp(-c_2 k). \tag{6.14}$$

Beweis. [3, Seiten 687 ff.] ■

Lemma 6.1.5. *Seien $\alpha \in \mathbb{R}$ und $\mathbf{L}$ wie in Gleichung (6.13). Dann gilt*

$$\exp(\alpha \mathbf{L}) = \exp\left(\alpha \sum_{\mu=1}^{d} \mathbf{Id}_{\mathcal{S}}^{\mu}(\mathbf{T})\right) = \bigotimes_{\mu=1}^{d} \exp\left(\alpha \mathbf{T}\right),$$

insbesondere ist der Tensorrang von $\exp(\alpha \mathbf{L})$ gleich 1.

Beweis. Sei $\alpha \in \mathbb{R}$. Da die Summanden von $\mathbf{L}$ kommutieren, gilt

$$\begin{aligned}
\exp(\alpha \mathbf{L}) &= \exp\left(\alpha \sum_{\mu=1}^{d} \mathbf{Id}_{\mathcal{S}}^{\mu}(\mathbf{T})\right) \\
&= \prod_{\mu=1}^{d} \exp\left(\alpha \mathbf{Id}_{\mathcal{S}}^{\mu}(\mathbf{T})\right) \\
&= \prod_{\mu=1}^{d} \mathbf{Id}_{\mathcal{S}}^{\mu}(\alpha \exp(\mathbf{T})) \\
&= \bigotimes_{\mu=1}^{d} \exp(\alpha \mathbf{T}).
\end{aligned}$$

■

Lemma 6.1.6. *Seien $d \in \mathbb{N}$, $n \in \mathbb{N}$ fest gewählt, $\mathbf{L}$ wie in Gleichung (6.13) und $\mathcal{T} := \bigotimes^{d} \mathbb{R}^{n \times n}$. Getreu Lemma 6.1.4 seien $k \in \mathbb{N}$, $\underline{\alpha}^*, \underline{\omega}^* \in \mathbb{R}^k$, $c_1, c_2 \in \mathbb{R}_{>0}$ und $R := \lambda_{\max}(\mathbf{L})/\lambda_{\min}(\mathbf{L})$ derart, dass*

$$E_k(\underline{\alpha}^*, \underline{\omega}^*) = \min_{\underline{\alpha}, \underline{\omega} \in \mathbb{R}^k} E_k(\underline{\alpha}, \underline{\omega}) \leq c_1 \exp(-c_2 k)$$

sowie

$$\begin{aligned}
\mathrm{inv}_k : [1, R] &\to \mathbb{R} \\
t &\mapsto \mathrm{inv}_k(t) := \sum_{j=1}^{k} \frac{\omega_j^*}{\lambda_{\min}(\mathbf{L})} \exp(-\frac{\alpha_j^*}{\lambda_{\min}(\mathbf{L})} t).
\end{aligned}$$

Dann gelten:

(i)

$$\mathrm{inv}_k(\mathbf{L}) = \sum_{j=1}^{k} \frac{\omega_j^*}{\lambda_{\min}(\mathbf{L})} \bigotimes_{\mu=1}^{d} \exp\left(-\frac{\alpha_j^*}{\lambda_{\min}(\mathbf{L})} \mathbf{T}\right) \in \mathcal{T}_{\leq k}, \tag{6.15}$$

(ii)

$$\|\mathbf{L}^{-1} - \mathrm{inv}_k(\mathbf{L})\| \leq \frac{c_1}{\lambda_{\min}(\mathbf{T})} \frac{\exp(-c_2 k)}{d}. \tag{6.16}$$

Beweis. Sei $\mathbf{L}$ wie in Gleichung (6.13) und $k \in \mathbb{N}$.
(i): Gemäß Lemma 6.1.5 gilt

$$\exp(\alpha\mathbf{L}) = \bigotimes_{\mu=1}^{d} \exp(\alpha\mathbf{T}) \quad (\alpha \in \mathbb{R}),$$

daher folgt

$$\begin{aligned} \mathrm{inv}_k(\mathbf{L}) &= \sum_{j=1}^{k} \frac{\omega_j^*}{\lambda_{\min}(\mathbf{L})} \exp\left(-\frac{\alpha_j^*}{\lambda_{\min}(\mathbf{L})}\mathbf{L}\right) \\ &= \sum_{j=1}^{k} \frac{\omega_j^*}{\lambda_{\min}(\mathbf{L})} \bigotimes_{\mu=1}^{d} \exp\left(-\frac{\alpha_j^*}{\lambda_{\min}(\mathbf{L})}\mathbf{T}\right). \end{aligned}$$

(ii): Abkürzend seien $\lambda_1 := \lambda_{\min}(\mathbf{L})$ und $\lambda_2 := \lambda_{\max}(\mathbf{L})$ gesetzt. Es gilt

$$\begin{aligned} \|\mathbf{L}^{-1} - \mathrm{inv}_k(\mathbf{L})\| &= \max_{t\in[\lambda_1,\lambda_2]} \left|\frac{1}{t} - \mathrm{inv}_k(t)\right| \\ &= \max_{t\in[\lambda_1,\lambda_2]} \left|\frac{1}{t} - \frac{1}{\lambda_1}\sum_{j=1}^{k} \omega_j^* \exp\left(-\frac{\alpha_j^*}{\lambda_1}t\right)\right| \\ &= \max_{t\in[\lambda_1,\lambda_2]} \frac{1}{\lambda_1}\left|\frac{1}{\frac{t}{\lambda_1}} - \sum_{j=1}^{k} \omega_j^* \exp\left(-\alpha_j^*\frac{t}{\lambda_1}\right)\right| \\ &= \frac{1}{\lambda_1}\max_{t'\in[1,\frac{\lambda_2}{\lambda_1}]}\left|\frac{1}{t'} - \sum_{j=1}^{k} \omega_j^* \exp\left(-\alpha_j^* t'\right)\right| \\ &= \frac{1}{\lambda_1} E_k(\underline{\alpha}^*, \underline{\omega}^*) \\ &\le \frac{c_1}{\lambda_1}\exp(-c_2 k) = \frac{c_1}{\lambda_{\min}(\mathbf{L})}\exp(-c_2 k) \\ &= \frac{c_1}{\lambda_{\min}(\mathbf{T})}\frac{\exp(-c_2 k)}{d}, \end{aligned}$$

denn es sind

$$\begin{aligned} \lambda_{\min}(\mathbf{L}) &= d\lambda_{\min}(\mathbf{T}), \\ \lambda_{\max}(\mathbf{L}) &= d\lambda_{\max}(\mathbf{T}). \end{aligned}$$

Ferner folgt damit

$$R = \lambda_{\max}(\mathbf{L})/\lambda_{\min}(\mathbf{L}) = \lambda_{\max}(\mathbf{T})/\lambda_{\min}(\mathbf{T}),$$

so dass R, c_1, und c_2 unabhängig von d sind. ■

Bemerkung 6.1.7. *Für die Berechnung von* $\mathrm{inv}_k(\mathbf{L})$ *ist die Kenntnis der Vektoren* $\underline{\alpha}$ *und* $\underline{\omega}$ *notwendig. Diese wurden bereits für unterschiedliche* k *berechnet und sind für den freien Gebrauch verfügbar, siehe [16]. Darüber hinaus ist*

auch $\exp(\alpha T)$ *zu berechnen, hierfür wird* $\mathbf{T}$ *mittels unitärer Transformation* $\mathbf{U} \in \mathbb{R}^{n\times n}$ *diagonalisiert, d.h.*

$$\mathbf{T} = \mathbf{U}\mathbf{D}\mathbf{U}^t$$

und $\mathbf{D} := \operatorname{diag}(\lambda_l)_{l\in\mathbb{N}_{\leq n}} \in \mathbb{R}^n$ *ist eine Diagonalmatrix.* $\mathbf{T}$ *hat eine besonders einfache tridiagonale Struktur, daher können die benötigten Eigenwerte und Eigenvektoren von* $\mathbf{T}$ *analytisch angegeben werden, siehe [47]. Ferner gilt für* $\operatorname{inv}_k(\mathbf{L})$

$$\operatorname{inv}_k(\mathbf{L}) \quad = \quad \left(\otimes^d \mathbf{U}\right) \left(\sum_{j=1}^{k} \frac{\omega_j^*}{\lambda_{\min}(\mathbf{L})} \bigotimes_{\mu=1}^{d} \operatorname{diag}\left(\exp\left(\frac{-\alpha_j^* \lambda_l}{\lambda_{\min}(\mathbf{L})} \right) \right)_{l=1}^{n} \right) \left(\otimes^d \mathbf{U}\right)^t,$$

wobei $\otimes^d \mathbf{U} := \bigotimes_{\mu=1}^{d} \mathbf{U}$ *gesetzt ist.*

6.1.2 Numerische Ergebnisse

Das vorliegende Modellproblem wurde für $n := 1\times 10^3$ und mit verschiedenen Dimensionen generiert. Die analytische Lösung u aus Gleichung (6.5) wurde an den Gitterpunkten ausgewertet, die hierdurch erzeugte Summe von Elementartensoren wird mit $\underline{u} \in \mathcal{S}_2$ bezeichnet, wobei im Folgenden $\mathcal{S} := \otimes^d \mathbb{R}^n$ und $\mathcal{T} := \otimes^d \mathbb{R}^{n\times n}$ gesetzt sind. Wendet man $\operatorname{inv}_k(\mathbf{L}) \in \mathcal{T}_{\leq k}$ auf $\underline{h}$ an, dann erhält man eine Näherungslösung $\tilde{\underline{u}}_k$, d.h.

$$\tilde{\underline{u}}_k := \operatorname{inv}_k(\mathbf{L})\underline{h} \in \mathcal{S}_{\leq 2dk}. \tag{6.17}$$

Der Modellfehler ist wie folgt definiert:

$$\mathcal{E}_k := \frac{\|\tilde{\underline{u}}_k - \underline{u}\|}{\|\tilde{\underline{u}}_k\|}. \tag{6.18}$$

Die numerischen Beispiele wurden für zwei verschiedene Werte von k erzeugt. Im ersten Fall wurde $k = 42$ gesetzt. Für alle $d \in \{10, 15, 20, 50, 75, 100\}$ galt dann für den Modellfehler

$$\mathcal{E}_{42} \leq 9.3\times 10^{-7} \tag{6.19}$$

und

$$\tilde{\underline{u}}_{42} \in \mathcal{S}_{\leq 84d}. \tag{6.20}$$

Um das Verhalten des Algorithmus auch bei größerem Modellfehler zu analysieren, wurde ein zweiter Fall, $k = 15$, betrachtet. Hier galt dann für alle $d \in \{10, 15, 20\}$

$$\mathcal{E}_{15} \leq 1.125\times 10^{-4} \tag{6.21}$$

und

$$\tilde{\underline{u}}_{15} \in \mathcal{S}_{\leq 30d}. \tag{6.22}$$

Im Anschluss wurde $\underline{\tilde{u}}_{42}$ bzw. $\underline{\tilde{u}}_{15}$ mit Hilfe von Elementartensor-Summen approximiert. Diese Approximation ist sinnvoll, denn es gilt $\underline{u} \in \mathcal{S}_2$. Hierfür wurde der Algorithmus zur Lösung der erweiterten Approximationsaufgabe verwendet und $\varepsilon_{42} := 1.0 \times 10^{-7} \|\underline{\tilde{u}}_{42}\|$ für $k = 42$ bzw. $\varepsilon_{15} := 1.2 \times 10^{-5} \|\underline{\tilde{u}}_{15}\|$ für $k = 15$ gewählt. Gemäß Korollar 5.3.17, Seite 86, beträgt hier die Komplexität ($R = 2kd$)

$$\mathcal{O}\left(d^4 n\right).$$

Beim bevorstehenden Test sollen folgende Fragen untersucht werden:

- Wie verhält sich der Algorithmus, wenn man $\underline{\tilde{u}}_k$ bis auf den Modellfehler approximiert, kann insbesondere $\underline{\tilde{u}}_k$ mit dem richtigen Tensorrang aufgelöst werden?
- Was passiert mit dem Verfahren, wenn man versucht, $\underline{\tilde{u}}_k$ unterhalb der bekannten Modellgenauigkeit $\mathcal{E}_k$ zu approximieren?
- Wieviele Iterationsschritte werden in beiden Fällen benötigt?

Folgende Bezeichnungen werden im Folgenden benötigt. Die Approximationen von $\underline{\tilde{u}}_k$, welche gemäß Algorithmus 5.4.1, Seite 89, während der Berechnung anfallen, werden mit $\underline{u}_r \in \mathcal{S}_r$ bezeichnet, wobei $r \in \mathbb{N}$ der Tensorrang von $\underline{u}_r$ ist. Zudem beschreibt $\underline{u}_r^{(0)} \in \mathcal{S}_r$ den dafür benötigten Startwert.
In den Tabellen sind spaltenweise der Tensorrang r, der relative Approximationsfehler des Startwertes $\underline{u}_r^{(0)}$, der relative Approximationsfehler von $\underline{u}_r$, die Norm des Gradienten der Zielfunktion f im Grenzwert, die Anzahl der Minimierungsschritte und die verwendete Rechenzeit in Sekunden aufgelistet.
Unabhängig von der Dimension und der Modellgenauigkeit wurde bei allen Rechenbeispielen für $r = 2$ die Elementartensor-Summe $\underline{\tilde{u}}_k$ bis auf den gegebenen Modellfehler approximiert. Die Anzahl der Iterationsschritte ist in diesen Fällen sehr gering, so wurden hier maximal 12 Iterationen benötigt. Anhand der Daten aus Tabelle 6.1 wird der detaillierte Verlauf des Minimierungsverfahrens in den Tabellen 6.10 und 6.11, Seite 101, exemplarisch dokumentiert, wobei l der Laufindex und α_l die Schrittweite sind. Ferner ist in Abbildung 6.1, Seite 102, der Verlauf des Gradienten der Zielfunktion f abgebildet. Hier wird insgesamt, unter den Voraussetzungen von Satz 4.4.3, Seite 60, die quadratische Konvergenz des Verfahrens deutlich. Bei diesen Berechnungen wurde die in Algorithmus 4.4.1, Seite 61, beschriebene Verschiebung nicht vollzogen, d.h. in Gleichung (4.42), wurde $\alpha_k = 1$ akzeptiert. Anders verhält es sich in den Fällen, in denen versucht wurde, über die Modellgenauigkeit hinaus zu approximieren. Hier wurden deutlich mehr Iterationsschritte benötigt, maximal 171, und die erwähnte Verschiebung angewandt. Ein charakteristischer Verlauf ist in Tabelle 6.12, Seite 101, abgebildet, wobei als Beispiel die Daten aus Tabelle 6.2 mit $r = 4$ dienen. Typisch ist dabei, dass der Gradient von f im Startwert und in allen Iterationspunkten sehr klein ist. Zügige superlineare oder gar quadratische Konvergenz tritt in diesen Fällen nicht ein. Dies hat aber keine Bedeutung, denn eine Approximation ist nur bis zur vorgegebenen Modellgenauigkeit von Signifikanz.

r	$\frac{\|\tilde{u}_k - \underline{u}_r^{(0)}\|}{\|\tilde{u}_k\|}$	$\frac{\|\tilde{u}_k - \underline{u}_r\|}{\|\tilde{u}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	3.098×10^{-1}	1.861×10^{-1}	1.642×10^{-9}	4	0.07
2	8.722×10^{-2}	5.162×10^{-8}	1.362×10^{-9}	9	0.15

Tabelle 6.1: Niedrigtensorrang-Approximation des Modellproblems mit d=10, k=42, R=840, n=1000 und $\mathcal{E}_{42}$=2.508×10^{-7}.

r	$\frac{\|\tilde{u}_k - \underline{u}_r^{(0)}\|}{\|\tilde{u}_k\|}$	$\frac{\|\tilde{u}_k - \underline{u}_r\|}{\|\tilde{u}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	2.199×10^{-1}	1.861×10^{-1}	7.447×10^{-9}	10	0.04
2	1.346×10^{-1}	1.587×10^{-5}	1.917×10^{-9}	7	0.06
3	1.344×10^{-5}	1.241×10^{-5}	9.401×10^{-9}	35	0.68
4	1.101×10^{-5}	9.789×10^{-6}	9.799×10^{-9}	146	3.29

Tabelle 6.2: Niedrigtensorrang-Approximation des Modellproblems mit d=10, k=15, R=300, n=1000 und $\mathcal{E}_{15}$=5.981×10^{-5}.

r	$\frac{\|\tilde{u}_k - \underline{u}_r^{(0)}\|}{\|\tilde{u}_k\|}$	$\frac{\|\tilde{u}_k - \underline{u}_r\|}{\|\tilde{u}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	3.938×10^{-1}	2.159×10^{-1}	2.869×10^{-9}	12	0.49
2	1.307×10^{-1}	6.322×10^{-8}	1.099×10^{-9}	5	0.44

Tabelle 6.3: Niedrigtensorrang-Approximation des Modellproblems mit d=15, k=42, R=1260, n=1000 und $\mathcal{E}_{42}$=5.228×10^{-7}.

r	$\frac{\|\tilde{u}_k - \underline{u}_r^{(0)}\|}{\|\tilde{u}_k\|}$	$\frac{\|\tilde{u}_k - \underline{u}_r\|}{\|\tilde{u}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	2.842×10^{-1}	2.159×10^{-1}	4.285×10^{-9}	11	0.16
2	1.365×10^{-1}	3.696×10^{-5}	1.951×10^{-9}	8	0.24
3	2.936×10^{-5}	2.058×10^{-5}	9.507×10^{-9}	171	9.38
4	1.419×10^{-5}	1.175×10^{-5}	9.911×10^{-9}	97	6.51

Tabelle 6.4: Niedrigtensorrang-Approximation des Modellproblems mit d=15, k=15, R=450, n=1000 und $\mathcal{E}_{15}$=9.174×10^{-5}.

r	$\frac{\|\tilde{u}_k - \underline{u}_r^{(0)}\|}{\|\tilde{u}_k\|}$	$\frac{\|\tilde{u}_k - \underline{u}_r\|}{\|\tilde{u}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	5.163×10^{-1}	1.990×10^{-1}	5.546×10^{-9}	10	1.07
2	9.536×10^{-1}	1.490×10^{-8}	8.088×10^{-9}	4	0.97

Tabelle 6.5: Niedrigtensorrang-Approximation des Modellproblems mit d=20, k=42, R=1680, n=1000 und $\mathcal{E}_{42}$=8.789×10^{-7}.

r	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r^{(0)}\|}{\|\tilde{\underline{u}}_k\|}$	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r\|}{\|\tilde{\underline{u}}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	2.311×10^{-1}	1.990×10^{-1}	3.847×10^{-9}	9	0.34
2	2.107×10^{-1}	4.609×10^{-5}	3.548×10^{-9}	2	0.22
3	3.131×10^{-5}	2.384×10^{-5}	9.906×10^{-9}	84	11.9
4	1.597×10^{-5}	1.183×10^{-5}	9.836×10^{-9}	102	17.3

Tabelle 6.6: Niedrigtensorrang-Approximation des Modellproblems mit d=20, k=15, R=600, n=1000 und $\mathcal{E}_{15}$=1.125×10^{-4}.

r	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r^{(0)}\|}{\|\tilde{\underline{u}}_k\|}$	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r\|}{\|\tilde{\underline{u}}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	9.144×10^{-1}	3.540×10^{-2}	1.051×10^{-8}	5	16.63
2	3.530×10^{-3}	1.125×10^{-8}	2.187×10^{-8}	2	17.46

Tabelle 6.7: Niedrigtensorrang-Approximation des Modellproblems mit d=50, k=42, R=4200, n=1000 und $\mathcal{E}_{42}$=9.261×10^{-7}.

r	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r^{(0)}\|}{\|\tilde{\underline{u}}_k\|}$	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r\|}{\|\tilde{\underline{u}}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	9.290×10^{-1}	6.729×10^{-3}	5.142×10^{-11}	4	77.66
2	1.546×10^{-4}	5.206×10^{-8}	1.720×10^{-8}	1	62.22

Tabelle 6.8: Niedrigtensorrang-Approximation des Modellproblems mit d=75, k=42, R=6300, n=1000 und $\mathcal{E}_{42}$=4.047×10^{-7}.

r	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r^{(0)}\|}{\|\tilde{\underline{u}}_k\|}$	$\frac{\|\tilde{\underline{u}}_k - \underline{u}_r\|}{\|\tilde{\underline{u}}_k\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	9.644×10^{-1}	1.271×10^{-3}	2.888×10^{-8}	3	183.50
2	6.308×10^{-6}	1.577×10^{-8}	7.925×10^{-11}	1	199.00

Tabelle 6.9: Niedrigtensorrang-Approximation des Modellproblems mit d=100, k=42, R=8400, n=1000 und $\mathcal{E}_{42}$=2.013×10^{-7}.

l	$\left\|f'(\underline{u}_r^{(l)})\right\|$	α_l	$\frac{\left\|\tilde{\underline{u}}_{42} - \underline{u}_r^{(l)}\right\|}{\|\tilde{\underline{u}}_{42}\|}$
0	2.0538×10^{-1}	-	3.0979×10^{-1}
1	5.1688×10^{-2}	1.00	1.9475×10^{-1}
2	1.1656×10^{-2}	1.00	1.8611×10^{-1}
3	6.0924×10^{-5}	1.00	1.8607×10^{-1}
4	1.6418×10^{-9}	1.00	1.8607×10^{-1}

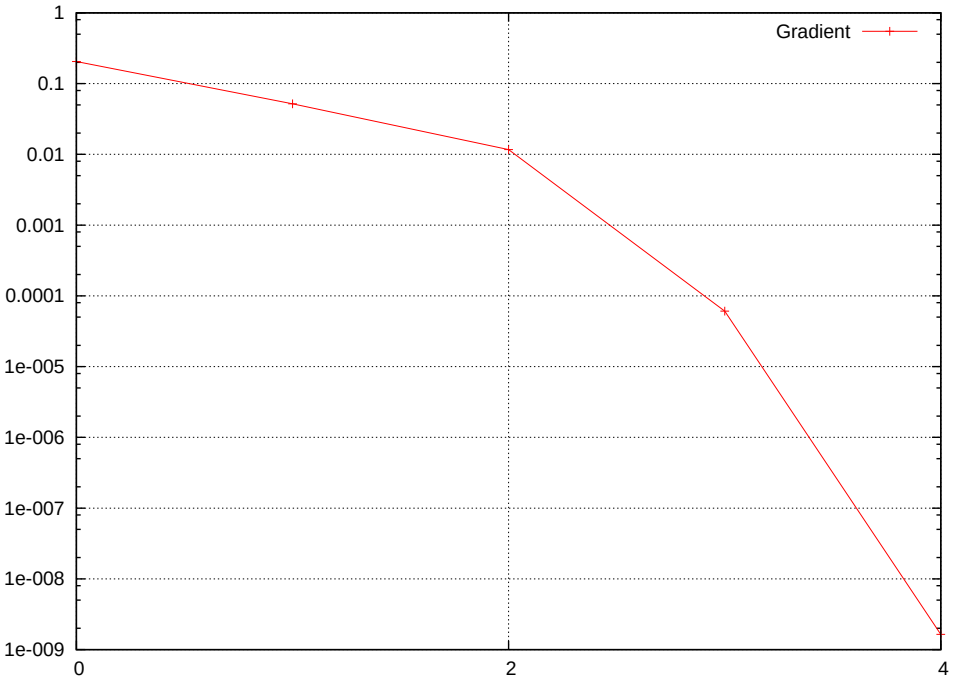
Tabelle 6.10: Verlauf des Verfahrens bei d=10, k=42, R=840, r=1 und n=1000.

l	$\left\|f'(\underline{u}_r^{(l)})\right\|$	α_l	$\frac{\left\|\underline{\tilde{u}}_{42}-\underline{u}_r^{(l)}\right\|}{\|\underline{\tilde{u}}_{42}\|}$
0	3.0571×10^{-2}	-	8.7216×10^{-2}
1	7.7305×10^{-2}	0.25	7.5917×10^{-2}
2	1.0317×10^{-1}	0.25	6.9140×10^{-2}
3	1.0359×10^{-1}	0.25	5.9843×10^{-2}
4	1.0125×10^{-1}	0.50	4.6120×10^{-2}
5	2.2740×10^{-2}	1.00	9.3248×10^{-3}
6	5.8127×10^{-4}	1.00	8.6426×10^{-4}
7	5.6276×10^{-5}	1.00	2.5567×10^{-5}
8	1.4504×10^{-8}	1.00	6.9094×10^{-8}
9	1.3624×10^{-9}	1.00	5.1619×10^{-8}

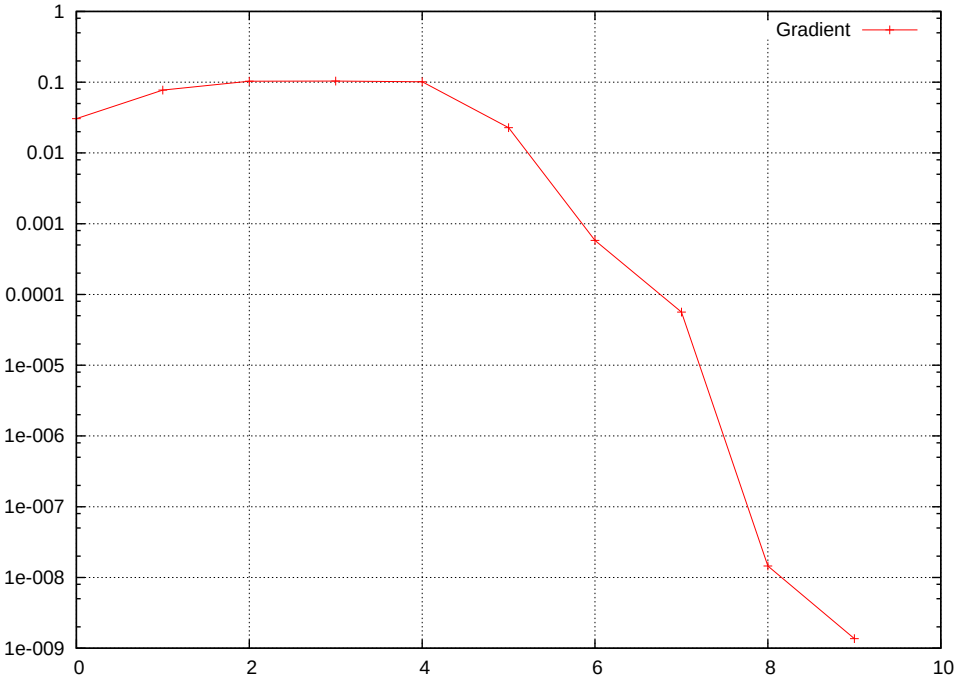
Tabelle 6.11: Verlauf des Verfahrens bei d=10, k=42, R=840, r=2 und n=1000.

l	$\left\|f'(\underline{u}_r^{(l)})\right\|$	α_l	$\frac{\left\|\underline{\tilde{u}}_{42}-\underline{u}_r^{(l)}\right\|}{\|\underline{\tilde{u}}_{42}\|}$
0	2.8259×10^{-8}	-	1.1014×10^{-5}
30	2.0969×10^{-8}	0.50	1.0490×10^{-5}
60	1.7256×10^{-8}	0.25	1.0217×10^{-5}
90	2.0321×10^{-8}	0.50	1.0029×10^{-5}
120	1.0128×10^{-8}	0.25	9.8876×10^{-6}
146	9.7994×10^{-9}	0.25	9.7889×10^{-6}

Tabelle 6.12: Verlauf des Verfahrens bei d=10, k=15, R=300, r=4 und n=1000.



(a) r=1.



(b) r=2.

Abbildung 6.1: Übersicht über den Verlauf des Gradienten von f bei d=10, k=42, R=840 und n=1000, siehe Tabellen 6.10 und 6.11.

6.2 Iterative Verfahren mit Summen von Elementartensoren

Im Folgenden seien $d, n \in \mathbb{N}_{>1}$, $\underline{I} := \times_{\mu=1}^{d} \mathbb{N}_{\leq n}$ und $\mathcal{T} := \bigotimes^{d} \mathbb{R}^n$. Im Mittelpunkt der Betrachtung steht eine rekursiv gebildete Folge $(x_k)_{k\in\mathbb{N}_{\geq 0}}$ von Elementartensor-Summen. Demzufolge existiert für alle $k \in \mathbb{N}$ eine Abbildung $\Phi_k : \mathcal{T} \to \mathcal{T}$, so dass für gegebenes $x_0 \in \mathcal{T}$ folgende Rekursion gilt

$$x_k := \Phi_k(x_{k-1}). \tag{6.23}$$

Darüber hinaus sei für einen hinreichend guten Startwert x_0 die Folge $(x_k)_{k\in\mathbb{N}_{\geq 0}}$ gegen $b \in \mathcal{T}$ konvergent. Nachstehendes Lemma 6.2.1 präzisiert diese Aussage.

Lemma 6.2.1. *Seien $k \in \mathbb{N}$, $b \in \mathcal{T}$ und Φ_k wie oben beschrieben. Ferner existieren $c_\Phi, \varepsilon_\Phi \in \mathbb{R}_{>0}$ und $\alpha \in \mathbb{R}_{>1}$ derart, dass für alle $x \in \mathcal{T}$ mit $\|x - b\| \leq \varepsilon_\Phi$ die folgende Ungleichung*

$$\|\Phi_k(x) - b\| \leq c_\Phi \|x - b\|^\alpha \tag{6.24}$$

erfüllt. Daneben seien

$$c := \sqrt[\alpha-1]{c_\Phi}, \tag{6.25}$$

$$\varepsilon := \min\{\varepsilon_\Phi, 1/c\} \tag{6.26}$$

gesetzt. Dann gilt für alle $x_0 \in \mathcal{T}$ mit $\|x_0 - b\| \leq \varepsilon$

$$\lim_{k\to\infty} x_k = b \tag{6.27}$$

und für alle $k \in \mathbb{N}_{\geq 0}$

$$\|x_k - b\| \leq c^{-1} \left(c \|x_0 - b\|\right)^{\alpha^k}. \tag{6.28}$$

Beweis. [18, Lemma 2.1, Seite 4]. ■

Definition 6.2.2 (Wertebereich)**.** *Sei $u \in \mathcal{T}$, etwa $u := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu}$, $r \in \mathbb{N}$. Man bezeichnet den* Wertebereich von u *mit $\mathfrak{W}(u)$. Dieser ist wie folgt definiert:*

$$\mathfrak{W}(u) := \left\{ u_{\underline{i}} := \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i_\mu} \in \mathbb{R} : \underline{i} := (i_1, \ldots, i_d) \in \underline{I} \right\}. \tag{6.29}$$

Notation 6.2.3. *Seien $u \in \mathcal{T}$ und $f : D(f) \to \mathbb{R}$, wobei $D(f) \subseteq \mathbb{R}$ der Definitionsbereich von f ist. Ferner sei $\mathfrak{W}(u) \subseteq D(f)$, dann bezeichnet man mit $f(u) \in \mathcal{T}$ den Tensor, welcher durch die punktweise Anwendung von u auf f entsteht. D.h., es gilt für alle Multiindizes $\underline{i} \in \underline{I}$*

$$(f(u))_{\underline{i}} := f(u_{\underline{i}}). \tag{6.30}$$

Bei einigen später beschriebenen Anwendungen wird $b = f(u)$ sein, wobei aus Gründen der praktischen Umsetzung vorausgesetzt ist, dass $f(u)$ gut mit Elementartensor-Summen kleineren Tensorrangs zu approximieren ist.
Neben der üblichen Addition und skalaren Multiplikation ist der Vektorraum $\mathcal{T}$ mit der punktweisen Multiplikation versehen.

Definition 6.2.4 (Punktweise Multiplikation, Potenz)**.** *Die* punktweise Multiplikation *ist folgendermaßen definiert:*

$$\begin{aligned} \cdot : \mathcal{T} \times \mathcal{T} &\to \mathcal{T} \\ (u, v) &\mapsto u \cdot v := uv := (u_{\underline{i}} v_{\underline{i}})_{\underline{i} \in \underline{I}}. \end{aligned} \tag{6.31}$$

Seien ferner $k \in \mathbb{N}$, $u \in \mathcal{T}$, $\underline{1} := (1, \ldots, 1)^t \in \mathbb{R}^n$ *und* $\mathbb{1} := \bigotimes_{\mu=1}^{d} \underline{1}$. *Durch*

$$u^0 := \mathbb{1}, \tag{6.32}$$

$$u^k := u^{k-1}\, u \tag{6.33}$$

sind u^0 *und die* k-te Potenz von u *definiert.*

Der Tensorrang der Summe und des Produktes zweier Elementartensor-Summen steigt im Allgemeinen an.

Lemma 6.2.5. *Seien* $r_1, r_2 \in \mathbb{N}$, $u \in \mathcal{T}_{\leq r_1}$ *und* $v \in \mathcal{T}_{\leq r_2}$. *Dann gelten:*

(i) $u + v \in \mathcal{T}_{\leq r_1 + r_2}$,

(ii) $uv \in \mathcal{T}_{\leq r_1 r_2}$.

Beweis. (i): Klar, wie in Bemerkung 1.4.4, Seite 16.

(ii): Für $u := \sum_{j_1=1}^{r_1} \bigotimes_{\mu=1}^{d} u_{j_1\mu}$, $v := \sum_{j_2=1}^{r_2} \bigotimes_{\mu=1}^{d} v_{j_2\mu}$ und $\underline{i} := (i_1, \ldots, i_d) \in \underline{I}$ gilt

$$\begin{aligned} u_{\underline{i}} v_{\underline{i}} &= \left[\sum_{j_1=1}^{r_1} \prod_{\mu=1}^{d} (u_{j_1\mu})_{i_\mu} \right] \left[\sum_{j_2=1}^{r_2} \prod_{\mu=1}^{d} (v_{j_2\mu})_{i_\mu} \right] \\ &= \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \prod_{\mu=1}^{d} (u_{j_1\mu})_{i_\mu} \prod_{\mu=1}^{d} (v_{j_2\mu})_{i_\mu} \\ &= \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \prod_{\mu=1}^{d} \left[(u_{j_1\mu})_{i_\mu} (v_{j_2\mu})_{i_\mu} \right]. \end{aligned}$$

Hieraus folgt $uv \in \mathcal{T}_{\leq r_1 r_2}$. ∎

Bemerkung 6.2.6. *Die Komplexität der Addition beträgt*

$$d(r_1 + r_2)n \tag{6.34}$$

Kopieroperationen.
Für die punktweise Multiplikation werden

$$dr_1 r_2 n \tag{6.35}$$

Operationen benötigt.

6.2.1 Inexakte Iterationsverfahren

Die Abbildungen Φ_k aus Gleichung (6.23) werden mit Hilfe der Addition, skalaren und punktweisen Multiplikation von Elementartensor-Summen gebildet. Getreu Lemma 6.2.5 erhöht sich damit im Allgemeinen der Tensorrang von x_k derart ungünstig, dass eine praktische Durchführung des Iterationsverfahrens nicht möglich ist. Daher versucht man, x_k durch Elementartensor-Summen kleineren Tensorrangs zu approximieren und setzt dann die Rekursion mit einer hinreichend guten Näherung fort. Hierdurch entsteht folgendes gestörtes Iterationsverfahren, wobei die gerade erwähnte Approximation, gemäß Notation 5.4.7, Seite 90, mit $\mathfrak{App}_{\varepsilon_k}(\cdot)$ bezeichnet wird:

$$\begin{aligned} y_0 &:= \mathfrak{App}_{\varepsilon_0}(x_0), \\ z_k &:= \Phi_k(y_{k-1}), \\ y_k &:= \mathfrak{App}_{\varepsilon_k}(z_k). \end{aligned} \tag{6.36}$$

Wählt man bei der Approximation ε_k genügend klein, dann folgt gemäß Lemma 6.2.7 die Konvergenz von $(y_k)_{k\in\mathbb{N}_{\geq 0}}$ gegen b. Zusätzlich wird die Konvergenzordnung von $(x_k)_{k\in\mathbb{N}_{\geq 0}}$ auf $(y_k)_{k\in\mathbb{N}_{\geq 0}}$ übertragen.

Lemma 6.2.7. *Es gelten die Bezeichnungen und Voraussetzungen von Lemma 6.2.1. Daneben seien $(y_k)_{k\in\mathbb{N}_{\geq 0}}$ und $(z_k)_{k\in\mathbb{N}_{\geq 0}}$ wie oben beschrieben. Ferner seien $c \in \mathbb{R}_{>0}$ und für alle $k \in \mathbb{N}$ $\varepsilon_k := c\|z_k - b\|$, d.h.*

$$\left\|z_k - \mathfrak{App}_{\varepsilon_k}(z_k)\right\| \leq c\|z_k - b\|. \tag{6.37}$$

Dann existiert ein $\delta \in \mathbb{R}_{>0}$, so dass $(y_k)_{k\in\mathbb{N}_{\geq 0}}$ gegen b konvergiert. Überdies gilt für alle Startwerte $y_0 \in \mathcal{T}$ mit $\|y_0 - b\| \leq \delta$ folgende Ungleichung:

$$\|y_k - b\| \leq (c+1)c_\Phi\|y_{k-1} - b\|^\alpha. \tag{6.38}$$

Beweis. [18, Lemma 2.2, Seite 4]. ■

Bemerkung 6.2.8. *Da der Tensorrang von b im Allgemeinen sehr groß sein kann, muss man aus Gründen der praktischen Umsetzbarkeit annehmen, dass b gut mit Summen von Elementartensoren moderaten Tensorrangs approximiert wird. In vielen interessanten Fällen wurde diese Annahme bereits bewiesen, siehe z.B. [17], [19], [20], [21], [30], [42] und [43]. Ist die Approximation von b nur bis zu einem $\widetilde{\varepsilon} \in \mathbb{R}_{>0}$ möglich, dann gibt [18, Theorem 2.4, Seite 5] Auskunft über das Konvergenzverhalten von $(y_k)_{k\in\mathbb{N}_{\geq 0}}$.*
Konvergiert $(x_k)_{k\in\mathbb{N}_{\geq 0}}$ linear gegen b, dann überträgt sich die lineare Konvergenz auch auf $(y_k)_{k\in\mathbb{N}_{\geq 0}}$, falls $(c+1)c_\Phi < 1$ erfüllt ist, siehe dazu [18, Bemerkung 2.6, Seite 6].

6.2.2 Berechnung der Maximumnorm und des zugehörigen Index

Seien $r \in \mathbb{N}$ und $u = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \in \mathcal{T}_{\leq r}$. In diesem Abschnitt wird eine Methode zur Berechnung der Maximumnorm von u,

$$\|u\|_\infty := \max_{\underline{i}:=(i_1,\ldots,i_d)\in\underline{I}} |u_{\underline{i}}| = \max_{\underline{i}:=(i_1,\ldots,i_d)\in\underline{I}} \left| \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i_\mu} \right|, \tag{6.39}$$

und des zugehörigen Index vorgestellt. Da die Mächtigkeit von $\underline{I}$ exponentiell mit d anwächst, $\#\underline{I} = n^d$, sind alle bekannten Methoden schon für kleine Werte von n und d nicht mehr effizient einsetzbar. Die Berechnung der Maximumnorm ist in vielen Bereichen des wissenschaftlichen Rechnens von Bedeutung, so zum Beispiel bei Differentialgleichungen mit stochastischen Koeffizienten. Um der häufig gestellten Frage nach einem effizienten Algorithmus nachzugehen, wird die spezielle Struktur von u ausgenutzt. Es zeigt sich, dass die Berechnung von $\|u\|_\infty$ äquivalent zu einem bekannten Problem ist, welches man effizient lösen kann. Sei dazu $\underline{i}^* := (i_1^*, \ldots, i_d^*) \in \underline{I}$ der Index mit

$$\|u\|_\infty = |u_{\underline{i}^*}| = \left| \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i_\mu^*} \right|$$

und

$$e^{(\underline{i}^*)} := \bigotimes_{\mu=1}^{d} e_{i_\mu^*}$$

definiert, wobei $e_{i_\mu^*} \in \mathbb{R}^n$ der i_μ^*-te kanonische Einheitsvektor im $\mathbb{R}^n$ ist ($\mu \in \mathbb{N}_{\leq d}$). Dann gilt für das punktweise Produkt von $ue^{(\underline{i}^*)}$

$$\begin{aligned} ue^{(\underline{i}^*)} &= \left[\sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \right] \left[\bigotimes_{\mu=1}^{d} e_{i_\mu^*} \right] \\ &= \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} e_{i_\mu^*} \\ &= \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \left[(u_{j\mu})_{i_\mu^*} \, e_{i_\mu^*} \right] \\ &= \underbrace{\left[\sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i_\mu^*} \right]}_{u_{\underline{i}^*} =} \bigotimes_{\mu=1}^{d} e_{(i_\mu^*)}, \end{aligned}$$

woraus dann

$$ue^{(\underline{i}^*)} = u_{\underline{i}^*} e^{(\underline{i}^*)} \tag{6.40}$$

folgt. Gleichung (6.40) ist von der Struktur einer Eigenwertgleichung. Definiert man folgende Diagonalmatrix

$$D(u) := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \operatorname{diag}\left((u_{j\mu})_{l_\mu}\right)_{l_\mu \in \mathbb{N}_{\le n}} \tag{6.41}$$

mit Tensorrang r, dann gilt für alle $v \in \mathcal{T}$

$$D(u)v \quad = \quad uv.$$

Korollar 6.2.9. *Es seien u, $\underline{i}^*$ und $D(u)$ wie oben definiert. Dann sind die Einträge von u die Eigenwerte von $D(u)$ und alle Eigenvektoren $e^{(\underline{i})}$ von folgender Gestalt:*

$$e^{(\underline{i})} = \bigotimes_{\mu=1}^{d} e_{i_\mu}, \tag{6.42}$$

wobei $\underline{i} := (i_1, \ldots, i_d) \in \underline{I}$ der Index von $u_{\underline{i}}$ ist. Damit ist insbesondere $\|u\|_\infty$ der betragsmäßig größte Eigenwert von $D(u)$ zum Eigenvektor $e^{(\underline{i}^*)}$.

Zur Berechnung des betragsmäßig größten Eigenwerts und des zugehörigen Eigenvektors gibt es eine Vielzahl etablierter Verfahren, siehe [10, Kapitel 8, Seiten 391 ff.]. Wegen der linearen Komplexität in d, r und n sind solche iterative Verfahren besonders gut geeignet, bei denen die Matrix-Vektor-Multiplikation mit $D(u)$ benötigt wird.
Zum numerischen Test soll folgende Elementartensor-Summe dienen:

$$u = \sum_{j=1}^{2} \bigotimes_{\mu=1}^{d} u_{j\mu}, \tag{6.43}$$

wobei für alle $l \in \mathbb{N}_{\le n}$ die Einträge von $u_{j\mu}$ wie folgt definiert sind:

$$(u_{j\mu})_l = [8((l-1)\eta)(1-(l-1)\eta)]^{j-1} \quad (\eta := 1/(n-1)). \tag{6.44}$$

Das in den Gleichungen (6.43) und (6.44) definierte Beispiel entsteht, wenn man die Funktion

$$\begin{aligned} \tilde{u} : [0,1]^d &\to \mathbb{R}_{>0} \\ (x_1, \ldots, x_d) &\mapsto \tilde{u}(x_1, \ldots, x_d) = \sum_{j=1}^{2} \prod_{\mu=1}^{d} [8x_\mu(1-x_\mu)]^{j-1} \end{aligned}$$

auf dem uniformen Gitter

$$\Gamma_n := \left\{ \eta \cdot \underline{i} \in [0,1]^d : \underline{i} \in (\mathbb{N}_{\le n})^d \right\}$$

auswertet. Die Funktion $\tilde{u}$ nimmt offenbar in $x^* := \left(\frac{1}{2}, \ldots, \frac{1}{2}\right)^t \in [0,1]^d$ ihr Maximum an, wobei $\tilde{u}(x^*) = 1+2^d$ ist. Wählt man $m \in \mathbb{N}$ und setzt $n := 2m-1$, so gilt

$$u_{\underline{i}^*} = \|u\|_\infty = \tilde{u}(x^*),$$

wobei $\underline{i}^* = (m, \ldots, m) \in \underline{I}$ ist.

Bemerkung 6.2.10. *Die Funktion $\tilde{u}$ wurde derart gewählt, dass sie Lösung folgender partieller Differentialgleichung ist:*

$$\begin{aligned} -\Delta\tilde{u} &= h, \quad (\,in\ \Omega\,) \\ \tilde{u}|_{\partial\Omega} &= 1. \end{aligned}$$

Hierbei sind $\Omega := (0,1)^d$ und h wie folgt definiert:

$$\begin{aligned} h : \Omega &\to \mathbb{R} \\ x := (x_1, \ldots, x_d) &\mapsto h(x) := 16 \sum_{\nu=1}^{d} \prod_{\mu=1, \mu\neq\nu}^{d} 8x_\mu(1-x_\mu). \end{aligned}$$

Bei diesem Beispiel bietet sich zur numerischen Lösung des Eigenwertproblems die einfache Vektoriteration an, welche z. B. in [10, Abschnitt 8.2.1, Seiten 406 ff.] beschrieben ist. Da der Tensorrang der Iterierten z_k monoton wächst, wird

Algorithmus 6.2.1 Näherungsweise Berechnung der Maximumnorm von $u \in \mathcal{T}_r$ mittels Vektoriteration

1: Wähle $y_0 := \bigotimes_{\mu=1}^{d} \frac{1}{\sqrt{n}}\underline{1}$, wobei $\underline{1} := (1, \ldots, 1)^t \in \mathbb{R}^n$ ist, $k_{\max} \in \mathbb{N}$ und setze $\varepsilon := 1\times10^{-7}$.

2: **for** $k = 1, 2, \ldots, k_{\max}$ **do**

3:

$$\begin{aligned} q_k &= uy_{k-1}, \quad \lambda_k = \langle y_{k-1}, q_k \rangle, \quad z_k = q_k/\|q_k\|, \\ y_k &= \mathfrak{App}_\varepsilon(z_k). \end{aligned}$$

4: **end for**

Algorithmus 6.2.1 entsprechend Gleichung (6.36) modifiziert. Die in Algorithmus 6.2.1 beschriebene Methode ist ein Näherungsverfahren zur Berechnung von $\|u\|_\infty$ und zugehörigem Index. Gemäß [12, Satz 4.31, Seiten 62 f.] werden

$$\mathcal{O}\left(\frac{d\log n - \log\varepsilon}{\varepsilon}\right). \tag{6.45}$$

Iterationsschritte benötigt, um die Maximumnorm von u bis auf einen relativen Fehler von $\varepsilon \in \mathbb{R}_{>0}$ zu berechnen. Um die Konvergenz zu garantieren, wählt man

$$y_0 := \frac{1}{n^d} \sum_{l_1=1}^{n} \cdots \sum_{l_d=1}^{n} \bigotimes_{\mu=1}^{d} e_{l_\mu} = \frac{1}{n^d} \bigotimes_{\mu=1}^{d} \left(\sum_{l_\mu=1}^{n} e_{l_\mu} \right) = \frac{1}{n^d} \bigotimes_{\mu=1}^{d} \underline{1}. \tag{6.46}$$

Wie schon erwähnt ist die vorgestellte Methode ein approximatives Verfahren, somit können $\|u\|_\infty$ und $e^{(\underline{i}^*)}$ nur näherungsweise bestimmt werden. Da aber die Eigenvektoren eine besonders prägnante Struktur haben, lohnt sich eine Post-Analyse von $y_{k_{\max}}$. Ist $\lambda_{k_{\max}}$ eine hinreichend gute Approximation von $\|u\|_\infty$, dann wird $\operatorname{rang}_{\mathcal{T}}(y_{k_{\max}}) = 1$ erfüllt sein. Ferner sind dann fast alle Einträge der

Repräsentantenvektoren von $y_{k_{\max}}$ sehr klein (praktisch null) und ein Eintrag hebt sich heraus. Setzt man diesen auf eins und alle anderen Einträge auf null, dann kann man den so erzeugten Eigenwert auf u testen und anschließend das Ergebnis mit $\lambda_{k_{\max}}$ vergleichen.
Der numerische Test wurde für $d \in \{25, 50, 75, 100, 125, 150\}$ und $m = 50$ durchgeführt. Die Anzahl der Gitterpunkte pro Raumrichtung beträgt dann 99, denn $n = 2m - 1$. In allen Beispielrechnungen konnte der gesuchte Multiindex $\underline{i}^* = (50, \ldots, 50) \in \mathbb{N}^d_{\leq n}$ mit der vorgestellten Methode ermittelt werden. In der nachstehenden Tabelle 6.13 sind die Ergebnisse der Rechnung zusammengefasst. Getreu Korollar 5.3.17, Seite 86, beträgt hier die Komplexität

$$\mathcal{O}\left(d^3 n\right).$$

Die in den vorangegangenen Kapiteln beschriebene Methode zur Approximati-

d	$\|u\|_\infty$ und $\underline{i}^*$ bestimmt	Rechenzeit in Sekunden
25	+	0,16
50	+	0,42
75	+	1,16
100	+	2,58
125	+	4,97
150	+	8,66

Tabelle 6.13: Beispiel aus Gleichung (6.43) zur Bestimmung der Maximumnorm und zugehörigen Index für m=50 und n=99.

on von Elementartensor-Summen arbeitet bei diesen Beispielen besonders gut; so wurden z. B. selten mehr als 10 Iterationsschritte benötigt. Überdies wurden in der Regel nur beste Tensorrang-1-Approximationen von z_k benötigt; dies liegt freilich an der besonders einfachen Struktur des Eigenwertproblems.

Bemerkung 6.2.11. *Im Allgemeinen wird die hier verwendete Vektoriteration keine geeignete Methode sein, um das Eigenwertproblem zu lösen. Eine mögliche Verbesserung ist die inverse Vektoriteration, welche auf ein im Spektrum verschobenes u angewandt wird. Hierfür ist die Berechnung der punktweisen Inversen notwendig, siehe dazu Abschnitt 6.2.3. Viele weitere anerkannte Verfahren benötigen einen Orthogonalisierungsschritt, dieser ist aber anscheinend für Elementartensor-Summen nicht praktikabel.*

6.2.3 Berechnung der punktweisen Inversen

Seien $r \in \mathbb{N}$ und $u \in \mathcal{T}_r$. Ferner seien $\mathbb{1} := \bigotimes_{\mu=1}^d \underline{1}$ und $\underline{1} := (1, \ldots, 1)^t \in \mathbb{R}^n$ gesetzt.
Die Berechnung von u^{-1} ist bei vielen interessanten Anwendungen von besonderer Bedeutung, z. B. bei der verbesserten Berechnung der Maximumnorm, wie bereits in Bemerkung 6.2.11 erwähnt. Daneben ist die Ermittlung der punktweisen Inversen auch bei der iterativen Berechnung von $\sqrt{u}$ erforderlich, siehe

[18]. Aber auch einige Methoden zur Berechnung von sign(u) benötigen die punktweise Inverse, siehe [23].

Bemerkung 6.2.12. *Die Berechnung von* sign(u) *hat eine interessante Anwendung. Ohne Beschränkung der Allgemeinheit sei hier kurz angenommen, dass* $\mathfrak{W}(u) = [0,1]$ *erfüllt sei. Zu gegebenem* $\mu \in [0,1]$ *ist ein* $u_{\leq\mu} \in \mathcal{T}$ *gesucht, dessen Einträge für alle* $\underline{i} \in \underline{I}$ *folgende Bedingung erfüllen:*

$$(u_{\leq\mu})_{\underline{i}} := \begin{cases} u_{\underline{i}}, & u_{\underline{i}} \leq \mu \\ 0, & u_{\underline{i}} > \mu \end{cases} \tag{6.47}$$

Bei stochastischen Anwendungen in hohen Dimensionen ist die Berechnung von $(u_{\leq\mu})_{\underline{i}}$ *eine wichtige Teilaufgabe. Berechnet man jetzt* sign$(u - \mu\mathbb{1})$*, dann folgt unmittelbar*

$$u_{\leq\mu} = \frac{1}{2}(\mathbb{1} - \operatorname{sign}(u - \mu\mathbb{1}))u, \tag{6.48}$$

wobei die Frage nach der Größe von $\operatorname{rang}_{\mathcal{T}}(\operatorname{sign}(u - \mu\mathbb{1}))$ *im Allgemeinen noch nicht beantwortet ist.*

Im Folgenden sei $\mathfrak{W}(u) \subset \mathbb{R} \setminus \{0\}$. Die Abbildung $\Phi_k : \mathcal{T} \to \mathcal{T}$ aus Gleichung (6.36) ist bei der Berechnung der Inversen wie folgt definiert:

$$x \mapsto \Phi_k(x) := x(2\mathbb{1} - ux). \tag{6.49}$$

Motiviert ist diese Rekursion durch die Anwendung des Newton-Verfahrens auf die Funktion $f(x) := u - x^{-1}$, siehe [18, Abschnitt 4.1, Seite 10]. Definiert man den Fehler

$$e_k := \mathbb{1} - ux_k, \tag{6.50}$$

so folgt

$$\begin{aligned} e_k &= \mathbb{1} - ux_k = \mathbb{1} - ux_{k-1}\left(\mathbb{1} + e_{k-1}\right) \\ &= e_{k-1} - ux_{k-1}e_{k-1} = \left(\mathbb{1} - ux_{k-1}\right)e_{k-1} \\ &= e_{k-1}^2 \\ &= e_0^{2^k} \end{aligned}$$

und $(x_k)_{k\in\mathbb{N}}$ konvergiert quadratisch, falls $\|e_0\| < 1$ erfüllt ist. Ferner folgt mit der Definition von e_k

$$\begin{aligned} u^{-1} - x_k &= u^{-1}e_k = \left(u^{-1} - x_{k-1}\right)u\left(u^{-1} - x_{k-1}\right) \\ &= u\left(u^{-1} - x_{k-1}\right)^2. \end{aligned}$$

Setzt man nun $\alpha := 2$ und $c_\Phi := \|u\|$, dann sind die Voraussetzungen von Lemma 6.2.7 erfüllt.
Zum numerischen Test wurden folgende zwei Funktionen

$$\varphi_1 : [0,1]^d \to \mathbb{R} \tag{6.51}$$

$$\underline{x} := (x_1, \ldots, x_d) \mapsto \varphi_1(x) := 1 + \frac{9}{d}\sum_{\mu=1}^{d} x_\mu \tag{6.52}$$

und

$$\varphi_2 : [0,1]^d \rightarrow \mathbb{R} \tag{6.53}$$

$$\underline{x} := (x_1, \ldots, x_d) \mapsto \varphi_2(x) := 1 + \sum_{l=1}^{2} \prod_{\mu=1}^{d} (x_\mu)^{\frac{l}{d}} \tag{6.54}$$

auf dem uniformen Gitter

$$\Gamma_n := \left\{ \eta(i_1 - 1, i_2 - 1, \ldots, i_d - 1)^t \in [0,1]^d : (i_1, \ldots, i_d) \in (\mathbb{N}_{\leq n})^d \right\}$$

ausgewertet, wobei $\eta := 1/(n-1)$ gesetzt ist. Die hierdurch erzeugten Summen von Elementartensoren werden mit u_1 und u_2 bezeichnet. Für die Wertebereiche von u_1 und u_2 gilt:

$$\mathfrak{W}(u_1) = [1, 10]$$

und

$$\mathfrak{W}(u_2) = [1, 2].$$

In Bemerkung 6.2.8 wurde gefordert, dass man u_1^{-1} und u_2^{-1} gut durch Summen von Elementartensoren approximieren kann. Für u_1 kann man diese Forderung beweisen, denn gemäß Lemma 6.1.4, Seite 94, existiert eine beste Approximation der Inversen-Abbildung mittels Exponentialsummen bezüglich der Maximumnorm. Sei etwa

$$s_{\hat{k}}(t) := \sum_{j=1}^{\hat{k}} \omega_j \exp(-\alpha_j t)$$

diese beste Approximation mit $\hat{k} \in \mathbb{N}$ und $t \in [1, 10]$, dann gilt für $s_{\hat{k}}(\varphi_1(\underline{x}))$

$$\begin{aligned} s_{\hat{k}}(\varphi_1(\underline{x})) &= \sum_{j=1}^{\hat{k}} \omega_j \exp(-\alpha_j \varphi_1(\underline{x})) \\ &= \sum_{j=1}^{\hat{k}} \omega_j \exp\left(-\alpha_j - \frac{9\alpha_j}{d} \sum_{\mu=1}^{d} x_\mu \right) \\ &= \sum_{j=1}^{\hat{k}} \omega_j \exp(-\alpha_j) \prod_{\mu=1}^{d} \exp\left(-\frac{9\alpha_j}{d} x_\mu \right), \end{aligned}$$

wobei $\underline{x} := (x_1, \ldots, x_d)^t \in [0,1]^d$ ist. Ferner gilt getreu Lemma 6.1.4 für den Approximationsfehler unabhängig von der Dimension d

$$\left| \frac{1}{\varphi_1(x)} - s_{\hat{k}}(\varphi_1(\underline{x})) \right| \leq c_1 \exp(-c_2 \hat{k}).$$

Wertet man jetzt $s_{\hat{k}} \circ \varphi_1$ in den Gitterpunkten von Γ_n aus, dann erhält man zumindest eine gute Approximation von u_1^{-1} in $\mathcal{T}_{\leq \hat{k}}$.
Für u_2 kann diese Herangehensweise nicht angewendet werden, denn $\exp(-\alpha_j \cdot) \circ \varphi_2$ ist keine separable Funktion. Überdies ist kein Beweis bekannt, welcher die Voraussetzung von Bemerkung 6.2.8 garantiert. Dennoch versucht man im Folgenden, u_2^{-1} mit Hilfe von Elementartensor-Summen zu approximieren.

Notation 6.2.13. *Seien u_1 und u_2 wie oben beschrieben. Zur Ermittlung des Approximationsfehlers werden folgende zwei Funktionen verwendet:*

$$\begin{aligned} \varrho_1 : \mathcal{T} &\to \mathbb{R} \qquad (6.55)\\ y &\mapsto \varrho_1(y) := \frac{\|\mathbb{1} - u_1 y\|}{\|\mathbb{1}\|},\\ \varrho_2 : \mathcal{T} &\to \mathbb{R} \qquad (6.56)\\ y &\mapsto \varrho_2(y) := \frac{\|\mathbb{1} - u_2 y\|}{\|\mathbb{1}\|}. \end{aligned}$$

Notwendig für die Konvergenz des Verfahren ist die Bestimmung eines Startwertes y_0 bzw. y_0' mit $\varrho_1(y_0) < 1$ bzw. $\varrho_2(y_0') < 1$. Im vorliegenden Fall genügte es, die beste Tensorrang-1-Approximation von u_1 bzw. u_2 zu berechnen und diese dann zu invertieren, wobei die Null kein Element des Wertebereichs dieser Approximationen war. Die Inverse eines Elementartensors ist leicht zu berechnen, denn sei $v := \bigotimes_{\mu=1}^{d} v_\mu \in \mathcal{T}_1$ mit $0 \notin \mathfrak{W}(v)$, dann ist $v^{-1} = \bigotimes_{\mu=1}^{d} v_\mu^{-1}$. Es ist davon auszugehen, dass y_{k-1} eine gute Approximation von y_k ist, daher wurde Algorithmus 5.4.2, Seite 90, mit bekannter guter Näherung y_{k-1} zum Approximieren von y_k verwendet.
Nachstehend sind die Berechnungen für $d \in \{20, 50, 100, 150\}$ und n:=100 tabellarisch protokolliert. Auch hier benötigte der vorgestellte Algorithmus 5.4.2 nur wenige Minimierungsschritte, um eine lokale beste Approximation von z_k zu ermitteln. Die Endergebnisse dieser Rechnungen wurden in den Tabellen 6.14 und 6.19 zusammengefasst. In den Tabellen 6.14 und 6.19 bezeichnet $y_{(r)}$ das Endergebnis der Iteration zur näherungsweisen Berechnung von u_1^{-1} bzw. u_2^{-1}. Der Tensorrang von $y_{(r)}$ ist mit $r \in \mathbb{N}$ gekennzeichnet. In den übrigen Tabellen gelten folgende Bezeichnungen:

k	-	Iterationsindex des Verfahrens zur Berechnung der Inversen,
z_k	-	Folgenglied aus Gleichung (6.36), Seite 105,
$\operatorname{rang}_{\mathcal{T}}(z_k)$	-	Tensorrang von z_k,
y_{r_k}	-	beste Approximation von z_k mit Tensorrang r_k,
$y_{r_k}^{(0)}$	-	Startwert der Bestapproximation von z_k mit Tensorrang r_k,
f	-	Zielfunktion, siehe Gleichung (3.15), Seite 40,
Iter.	-	Anzahl der Iterationsschritte von Algorithmus 5.4.2,
Zeit [Sek.]	-	benötigte Rechenzeit in Sekunden.

d	r	$\varrho_1(y_{(r)})$
20	4	3.690×10^{-6}
50	3	2.660×10^{-6}
100	3	2.137×10^{-6}
150	2	3.141×10^{-6}

Tabelle 6.14: Zusammenfassung der Berechnung von u_1^{-1} mit n=100.

k	$\text{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\|z_k - y_{r_k}\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	22	1	3.25×10^{-5}	3.25×10^{-5}	1.23×10^{-7}	1	0.02
1	22	2	2.70×10^{-5}	2.02×10^{-5}	4.35×10^{-8}	27	0.234
2	86	2	2.02×10^{-5}	2.02×10^{-5}	4.71×10^{-7}	1	0.01
2	86	3	1.09×10^{-5}	9.47×10^{-6}	4.42×10^{-9}	12	0.30
3	192	3	9.47×10^{-6}	9.47×10^{-6}	4.82×10^{-7}	1	0.08
3	192	4	6.02×10^{-6}	3.67×10^{-6}	3.83×10^{-10}	15	0.83

Tabelle 6.15: Berechnung von u_1^{-1} mit d=20, n=100 und $\varrho_1(y_{(4)})$=3.690×10^{-6}.

k	$\text{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\|z_k - y_{r_k}\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	52	1	1.32×10^{-5}	1.32×10^{-5}	3.21×10^{-7}	1	0.02
1	52	2	1.11×10^{-5}	9.16×10^{-6}	4.69×10^{-8}	11	0.95
2	206	2	9.16×10^{-6}	8.63×10^{-6}	4.89×10^{-10}	14	3.03
2	206	3	4.89×10^{-6}	4.51×10^{-6}	4.42×10^{-7}	5	2.52
3	462	3	4.51×10^{-6}	2.64×10^{-6}	2.74×10^{-10}	9	6.75

Tabelle 6.16: Berechnung von u_1^{-1} mit d=50, n=100 und $\varrho_1(y_{(3)})$=2.660×10^{-6}.

k	$\text{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\|z_k - y_{r_k}\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	102	1	6.63×10^{-6}	6.63×10^{-6}	1.15×10^{-7}	1	0.33
1	102	2	5.61×10^{-6}	4.43×10^{-6}	3.54×10^{-7}	25	21.19
2	406	2	4.43×10^{-6}	4.17×10^{-6}	4.50×10^{-7}	21	56.70
2	406	3	2.14×10^{-6}	2.13×10^{-6}	3.44×10^{-7}	2	12.14

Tabelle 6.17: Berechnung von u_1^{-1} mit d=100, n=100 und $\varrho_1(y_{(3)})$=2.137×10^{-6}.

k	$\text{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\|z_k - y_{r_k}\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	152	1	4.44×10^{-6}	4.44×10^{-6}	4.26×10^{-7}	1	1.55
1	152	2	3.77×10^{-6}	3.15×10^{-6}	3.79×10^{-7}	11	40.17

Tabelle 6.18: Berechnung von u_1^{-1} mit d=150, n=100 und $\varrho_1(y_{(3)})$=3.141×10^{-6}.

d	r	$\varrho_2(y_{(r)})$
20	3	1.614×10^{-6}
50	6	1.636×10^{-6}
100	7	1.541×10^{-6}
150	3	1.384×10^{-6}

Tabelle 6.19: Zusammenfassung der Berechnung von u_2^{-1} mit n:=100.

k	$\mathrm{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\left\|z_k - y_{r_k}\right\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	4	1	1.67×10^{-3}	1.67×10^{-3}	1.48×10^{-9}	1	0.00
1	4	2	1.05×10^{-3}	4.00×10^{-6}	2.60×10^{-7}	6	1.97
2	14	2	3.83×10^{-6}	3.52×10^{-6}	1.84×10^{-10}	2	0.77
3	14	2	3.51×10^{-6}	3.51×10^{-6}	6.49×10^{-10}	1	0.11
3	14	3	3.01×10^{-6}	2.81×10^{-6}	1.71×10^{-7}	2	1.70
4	30	3	1.70×10^{-6}	1.60×10^{-6}	2.77×10^{-7}	3	2.31

Tabelle 6.20: Berechnung von u_2^{-1} mit d=20, n=100 und $\varrho_2(y_{(3)})$=1.614×10^{-6}.

k	$\mathrm{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\left\|z_k - y_{r_k}\right\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	4	1	1.31×10^{-3}	1.31×10^{-3}	1.58×10^{-7}	1	0.00
1	4	2	9.35×10^{-4}	9.04×10^{-6}	2.35×10^{-7}	7	0.52
2	14	2	9.66×10^{-6}	8.54×10^{-6}	5.48×10^{-9}	2	0.17
2	14	3	7.35×10^{-6}	6.85×10^{-6}	2.11×10^{-7}	2	0.42
3	30	3	6.85×10^{-6}	4.74×10^{-6}	9.92×10^{-8}	10	3.31
4	30	3	4.73×10^{-6}	4.55×10^{-6}	1.47×10^{-9}	9	2.44
4	30	4	3.87×10^{-6}	3.52×10^{-6}	2.80×10^{-7}	4	2.39
5	52	4	3.52×10^{-6}	3.47×10^{-6}	1.07×10^{-9}	4	3.43
5	52	5	3.45×10^{-6}	3.45×10^{-6}	2.35×10^{-7}	1	4.40
5	52	6	2.04×10^{-6}	1.64×10^{-6}	2.12×10^{-9}	9	7.78

Tabelle 6.21: Berechnung von u_2^{-1} mit d=50, n=100 und $\varrho_2(y_{(6)})$=1.636×10^{-6}.

k	$\mathrm{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\left\|z_k - y_{r_k}\right\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	4	1	1.67×10^{-3}	1.67×10^{-3}	1.54×10^{-7}	1	0.01
1	4	2	1.06×10^{-3}	3.98×10^{-6}	1.99×10^{-7}	6	1.72
2	14	2	5.30×10^{-6}	3.53×10^{-6}	1.52×10^{-7}	2	0.73
3	14	2	3.52×10^{-6}	3.51×10^{-6}	2.96×10^{-7}	2	0.92
3	14	3	3.51×10^{-6}	3.50×10^{-6}	2.54×10^{-7}	1	2.14
3	14	4	2.99×10^{-6}	2.90×10^{-6}	2.84×10^{-7}	4	4.41
4	52	4	2.90×10^{-6}	2.90×10^{-6}	2.61×10^{-7}	1	11.84
4	52	5	2.84×10^{-6}	2.78×10^{-6}	1.56×10^{-7}	4	24.33
4	52	6	2.77×10^{-6}	2.72×10^{-6}	2.83×10^{-7}	5	31.06
4	52	7	1.72×10^{-6}	1.54×10^{-6}	2.21×10^{-7}	2	36.81

Tabelle 6.22: Berechnung von u_2^{-1} mit d=100, n=100 und $\varrho_2(y_{(7)})$=1.541×10^{-6}.

k	$\mathrm{rang}_{\mathcal{T}}(z_k)$	r_k	$\frac{\left\|z_k - y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\|z_k - y_{r_k}\|}{\|z_k\|}$	$\|f'(y_{r_k})\|$	Iter.	Zeit [Sek.]
1	4	1	1.66×10^{-3}	1.66×10^{-3}	1.02×10^{-7}	1	0.08
1	4	2	9.33×10^{-4}	2.34×10^{-6}	2.93×10^{-7}	5	4.06
2	14	2	3.72×10^{-6}	1.88×10^{-6}	1.86×10^{-7}	2	1.84
3	14	2	1.89×10^{-6}	1.89×10^{-6}	2.96×10^{-7}	2	1.11
3	14	3	1.57×10^{-6}	1.41×10^{-6}	2.33×10^{-7}	2	4.52

Tabelle 6.23: Berechnung von u_2^{-1} mit d=150, n=100 und $\varrho_2(y_{(3)})$=1.384×10^{-6}.

6.3 Vergleich der Abstiegsrichtungen

Für das Folgende gelten die in Kapitel 3 und 5 benutzten Notationen. Daneben sei $k \in \mathbb{N}$ der Iterationsindex des Minimierungsverfahrens zur Approximation von Elementartensor-Summen.
Die erste Wahl der Abstiegsrichtung war gemäß Gleichung (5.19), Seite 75, die Newton-Richtung, d.h.

$$H_k^{(1)} \quad := \quad f''(\hat{\xi}^k) = A_k + B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}.$$

Ferner wurde eine zweite Richtung empfohlen, die gemäß Gleichung (5.21) wie folgt definiert ist:

$$H_k^{(2)} := \begin{cases} A_k + B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}, & \overline{\alpha_k} \neq 1 \\ A_k + B_k + C_k - D_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}, & \overline{\alpha_k} = 1. \end{cases}$$

In Korollar 5.3.14 wurde gezeigt, dass diese Abstiegsrichtung schneller zu berechnen ist. Außerdem soll folgende vereinfachte Wahl von $H_k^{(2)}$ separat numerisch getestet werden:

$$H_k^{(3)} \quad := \quad A_k + B_k + \lambda_1 G_{1k} + \lambda_2 G_{2k}.$$

Zum numerischen Test wurden das Modellproblem aus Abschnitt 6.1 und zufällig ausgewählte Beispiele aus dem Abschnitt 6.2.3 verwendet. In allen Rechnungen unterschieden sich die vorgeschlagenen Richtungen kaum. Die Richtung mit $H_k^{(2)}$ erwies sich allerdings stets als die bessere Wahl, zumal sie auch in den ersten Schritten günstiger zu berechnen ist. Selbst die vereinfachte Abstiegsrichtung mit der Matrix $H_k^{(3)}$ lieferte sehr gute Ergebnisse.
In den folgenden Diagrammen ist auf der Abszisse der Laufindex des Minimierungsverfahrens zur Approximation von Elementartensor-Summen aufgetragen. Auf der Ordinate sind die Werte des Gradienten der Zielfunktion f, siehe Gleichung (3.15), Seite 40, abgebildet. Daneben sind mit $H\hat{}1$, $H\hat{}2$ bzw. $H\hat{}3$ die Verläufe des Verfahrens bei der unterschiedlichen Wahl von $H_k^{(1)}$, $H_k^{(2)}$ bzw. $H_k^{(3)}$ gekennzeichnet.

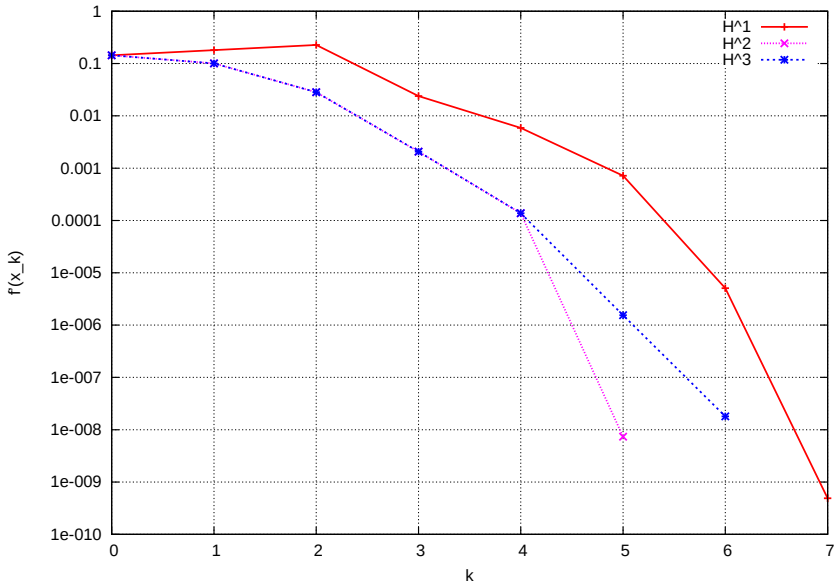


Abbildung 6.2: Vergleich der einzelnen Abstiegsrichtungen beim Modellproblem aus Abschnitt 6.1, mit d=25, r=2, R=2100 und n=1000.

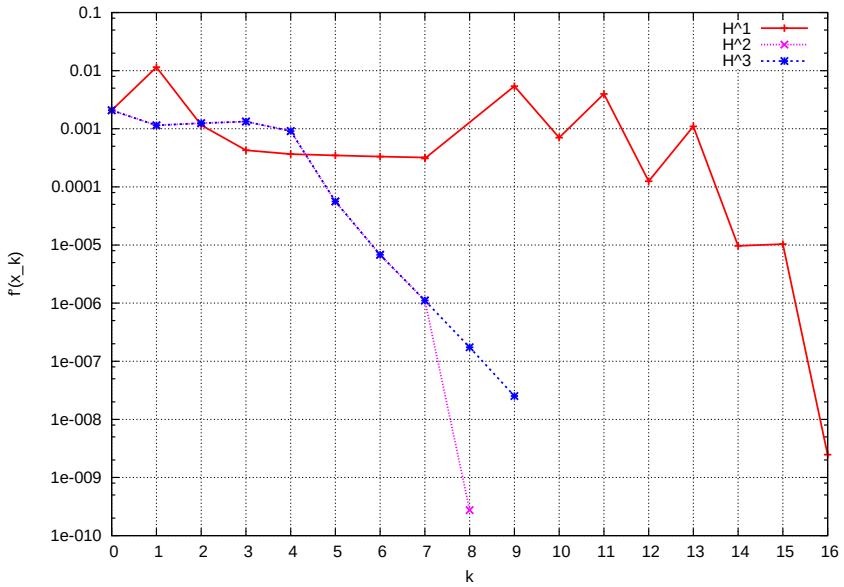


Abbildung 6.3: Vergleich der einzelnen Abstiegsrichtungen bei Beispiel u_1 aus Abschnitt 6.2.3, mit d=50, r=2, R=462 und n=100, siehe Tabelle 6.16, Seite 113.

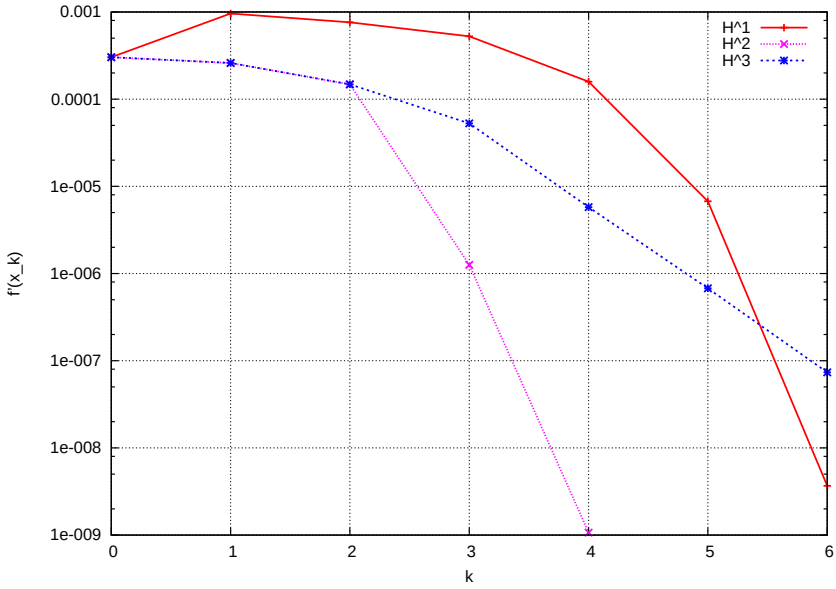


Abbildung 6.4: Vergleich der einzelnen Abstiegsrichtungen bei Beispiel u_2 aus Abschnitt 6.2.3, mit d=50, r=4, R=52 und n=100, siehe Tabelle 6.21, Seite 114.

6.4 Beispiele aus der Quantenchemie

Die in dieser Arbeit vorgestellte Methode zur Approximation mit Summen von Elementartensoren wurde in [6] und in der Dissertation von Sambasiva Rao Chinnamsetty [5] auf Probleme aus der Quantenchemie angewandt. In diesen Arbeiten wurden Niedrig-Tensorrang-Approximationen von Lösungen der Hartree-Fock-Gleichungen und des Hartree-Potentials untersucht. Die hierbei auftretenden Elementartensor-Summen entstehen bei der Diskretisierung von Funktionen folgender Art:

$$\begin{aligned} f : [-20,20]^3 &\to \mathbb{R} \\ (x_1,x_2,x_3) &\mapsto f(x_1,x_2,x_3) := \sum_{i=1}^{R} a_i \prod_{\mu=1}^{3} g_k^{(\mu)}(x_\mu - b_k^{(\mu)}), \end{aligned}$$

wobei $R \in \mathbb{N}$, $a_i \in \mathbb{R}$ und

$$\begin{aligned} g_k^{(\mu)} : [-20,20] &\to \mathbb{R} \\ x &\mapsto g_k^{(\mu)}(x) := x^{l_k} \exp(-\xi_k x^2) \quad (l_k \in \mathbb{N},\ \xi_k \in \mathbb{R}_{>0}) \end{aligned}$$

sind. Im Folgenden werden die numerischen Ergebnisse der Niedrigtensorrang-Approximation des Hartree-Potentials (u_X) von Methan (CH_4), Ethin (C_2H_2) und Ethan (C_2H_6) angegeben. Die hierbei verwendeten Ausgangsdaten wurden von Chinnamsetty erzeugt. Eine genaue Beschreibung des physikalischen Hintergrunds und der Diskretisierung kann man [5] und [6] entnehmen.

Folgende Bezeichnungen werden im Folgenden verwendet. Die Approximationen von u_X, welche gemäß Algorithmus 5.4.1, Seite 89, während der Berechnung anfallen, werden mit u_r bezeichnet, wobei $r \in \mathbb{N}$ der Tensorrang von u_r ist. Zudem beschreiben $u_r^{(0)}$ den dafür benötigten Startwert sowie $R \in \mathbb{N}$ den Tensorrang von u_X und $n \in \mathbb{N}$ die Anzahl der Gitterpunkte pro Raumrichtung.
Bei allen Berechnungen wurde ein lokales Minimum der Approximationsaufgabe ermittelt. Die hierfür benötigte Anzahl von Iterationsschritten ist in der Regel von moderater Größe. Einzig bei der Berechnung mit C_2H_2 wurden einmalig 113 Iterationen benötigt.

r	$\frac{\|u_{CH_4}-u_r^{(0)}\|}{\|u_{CH_4}\|}$	$\frac{\|u_{CH_4}-\underline{u}_r\|}{\|u_{CH_4}\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	2.55×10^{-1}	2.42×10^{-1}	7.49×10^{-7}	5	0.06
2	6.53×10^{-2}	6.45×10^{-2}	6.22×10^{-7}	3	0.06
3	2.88×10^{-2}	2.27×10^{-2}	2.91×10^{-7}	5	0.14
4	1.40×10^{-2}	1.24×10^{-2}	1.44×10^{-7}	6	0.23
5	1.40×10^{-2}	1.24×10^{-2}	1.44×10^{-7}	6	0.23
10	1.87×10^{-3}	1.81×10^{-3}	6.29×10^{-7}	25	2.42
20	2.42×10^{-4}	1.36×10^{-4}	6.84×10^{-7}	34	7.01
40	6.95×10^{-6}	6.94×10^{-6}	2.84×10^{-8}	2	1.55
62	1.02×10^{-6}	9.87×10^{-7}	1.35×10^{-8}	2	3.94

Tabelle 6.24: Niedrigtensorrang-Approximation des Hartree-Potentials von Methan (CH_4) mit d=3, R=2463 und n=5121.

r	$\frac{\|u_{C_2H_2}-u_r^{(0)}\|}{\|u_{C_2H_2}\|}$	$\frac{\|u_{C_2H_2}-\underline{u}_r\|}{\|u_{C_2H_2}\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	7.22×10^{-1}	7.21×10^{-1}	2.20×10^{-7}	3	0.04
2	2.67×10^{-1}	2.04×10^{-1}	6.22×10^{-11}	4	0.10
3	1.54×10^{-1}	1.53×10^{-2}	2.43×10^{-7}	2	0.07
4	1.22×10^{-2}	1.16×10^{-2}	9.93×10^{-8}	35	1.84
5	8.72×10^{-2}	7.80×10^{-2}	8.90×10^{-8}	14	0.98
10	1.53×10^{-2}	1.28×10^{-2}	8.19×10^{-8}	113	15.15
20	3.07×10^{-3}	2.98×10^{-3}	2.73×10^{-7}	33	8.68
40	2.84×10^{-4}	2.66×10^{-4}	1.79×10^{-7}	68	44.58
80	2.32×10^{-5}	2.31×10^{-5}	1.23×10^{-7}	6	10.91
160	1.35×10^{-6}	1.32×10^{-6}	4.62×10^{-9}	4	25.98
171	1.03×10^{-6}	9.85×10^{-7}	4.45×10^{-9}	5	31.67

Tabelle 6.25: Niedrigtensorrang-Approximation des Hartree-Potentials von Ethin (C_2H_2) mit d=3, R=2233 und n=5121.

r	$\frac{\|u_{C_2H_6}-u_r^{(0)}\|}{\|u_{C_2H_6}\|}$	$\frac{\|u_{C_2H_6}-\underline{u}_r\|}{\|u_{C_2H_6}\|}$	$\|f'(\underline{u}_r)\|$	Iterationen	Rechenzeit [Sek.]
1	2.51×10^{-1}	2.20×10^{-1}	2.39×10^{-7}	5	0.12
2	1.35×10^{-1}	1.34×10^{-1}	8.63×10^{-8}	7	0.32
3	8.22×10^{-2}	6.65×10^{-2}	2.23×10^{-7}	7	0.52
4	5.06×10^{-2}	5.06×10^{-2}	1.66×10^{-7}	1	0.14
5	4.61×10^{-2}	4.38×10^{-2}	1.99×10^{-7}	6	0.96
10	7.92×10^{-3}	7.81×10^{-3}	2.19×10^{-7}	19	4.34
20	1.62×10^{-3}	1.56×10^{-3}	1.01×10^{-7}	43	19.82
40	1.54×10^{-4}	1.45×10^{-4}	9.64×10^{-8}	57	54.01
80	1.79×10^{-5}	1.79×10^{-5}	2.28×10^{-7}	4	11.78
160	1.39×10^{-6}	1.37×10^{-6}	5.08×10^{-9}	4	35.99
175	1.02×10^{-6}	9.56×10^{-7}	6.21×10^{-9}	6	48.95

Tabelle 6.26: Niedrigtensorrang-Approximation des Hartree-Potentials von Ethan (C_2H_6) mit d=3, R=3744 und n=5121.

7 Entwurf und Realisierung

7.1 Programmentwurf

In den nun folgenden Abschnitten wird der Entwurf des Berechnungsmoduls kurz vorgestellt. Der gesamte Entwurf liegt dem objektorientierten Design zugrunde.

7.1.1 Problembeschreibung

Numerische Algorithmen zur Lösung der gestellten Approximationsaufgaben, siehe Abschnitt 3.1, sind zu implementieren. Daneben müssen geeignete Datenstrukturen zum Simulieren von Elementartensorsummen und deren algebraischen Operationen bereitgestellt werde.
Demnach kann die gesamte Umsetzung als eine Verkettung zweier getrennter Aufgaben angesehen werden:

- Generierung eines numerischen Algorithmus, welcher die gestellten Approximationsaufgaben getreu den vorangegangenen Kapiteln löst.
- Konstruktion problem-bezogener Datenstrukturen und Algorithmen zum Operieren mit Elementartensorsummen. Die dafür notwendigen Unterroutinen werden mittels BLAS- und LaPACK-Prozeduren bereitgestellt.

Geeignete Werkzeuge zum Visualisieren von Datensätzen werden vorausgesetzt.

7.1.2 Ein objektorientierter Entwurf

Im Rahmen des Entwurfs wird eine objektorientierte Klassenbibliothek erstellt. Die Vorteile der Objektorientierung sind vielfältig. Sowohl die erhöhte Lesbarkeit und vereinfachte Wartung des Codes als auch die Austauschbarkeit und Wiederverwendbarkeit können bei geeignetem Entwurf und Implementierung erreicht werden.
Um die Austauschbarkeit und Wiederverwendbarkeit zu gewährleisten, ist es wichtig, eine möglichst schwache Bindung zwischen den unterschiedlichen Elementen des Verfahrens herzustellen.

7.1.3 Klassenhierarchien

Wesentlich bei der Erstellung einer objektorientierten Klassenbibliothek ist der Entwurf dem Problem angemessener Vererbungshierarchien. Zunächst ist

es wichtig, Klassen zu identifizieren. Diese können aus der Problembeschreibung sowie aus der detaillierteren theoretischen Problembetrachtung gewonnen werden. Diese Klassen werden dann auf mögliche Spezialisierungen und Verallgemeinerungen untersucht. Es wird einerseits wegen der mangelnden Unterstützung in einigen Programmiersprachen und andererseits wegen der auftretenden Mehrdeutigkeiten bewusst auf Mehrfachvererbung verzichtet. In der folgenden Tabelle sind die Klassen aufgelistet, die nach diesem Prinzip identifiziert wurden. In der linken Spalte steht die Klassenbezeichnung, in der rechten Spalte wird die Klasse kurz beschrieben.

`DETS`	Summe von Elementartensoren aus $\bigotimes^d \mathbb{R}^n$.
`ETVector`	Basisklasse aller Repräsentantenvektoren
`DGETVector`	Repräsentantenvektor aus $\mathbb{R}^n$
`DGETSDDataBase`	Arbeitsspeicherverwaltung
`DGETSDDecomposer`	Schnittstelle zum Aufruf des Minimierungsverfahrens
`DGETSDFullMethod`	Basisklasse aller möglichen Minimierungsverfahren zum Approximieren von Elementartensor-Summen, welche alle Parameter simultan optimieren
`DGETSDNewton`	Das in Algorithmus 4.4.1, Seite 61, beschriebene Minimierungsverfahren

Tabelle 7.1: Objekte, die aus einer Analyse der Problembeschreibung und der theoretischen Behandlung des Problems gewonnen werden.

Die Klassen in Tabelle 7.1 sind bereits nach Zusammengehörigkeit gruppiert. Die Klassen einer Gruppe sind entweder Elemente der gleichen Klassenhierarchie oder aggregiert. Nun werden die identifizierten Klassen um Methoden und Attribute erweitert. Auf eine Diskussion der so erweiterten Klassen der einzelnen Gruppen aus Tabelle 7.1 wird im Folgenden verzichtet, denn im Wesentlichen ist deren Bedeutung und Verwendung klar. Ferner würde diese Analyse dem Grundgedanken dieser Arbeit nicht entsprechen. Die wesentlichen Beziehungen der Klassenhierarchien sind in Abbildung 7.1, Seite 125, zusammengefasst. Die in den Klassendiagrammen verwendete Notation ist der *Unified Modelling Language* (UML) entnommen, für eine komplette Übersicht kann man beispielsweise [32] empfehlen.

7.2 Implementierung

Das gesamte Programm ist in C++ geschrieben. In diesem Kapitel werden wichtige Implementierungsdetails und Besonderheiten kurz beschrieben. Auf einen Ausdruck von kompletten Teilen des Programms wird verzichtet.

7.2.1 Organisation der Module

Bis auf wenige Ausnahmen werden Schnittstelle und Implementierung der Klassen in zwei getrennten Modulen abgespeichert. Ein Modul mit Endung .hpp enthält die Schnittstelle und gegebenenfalls vorhandene `inline`-Methoden. Gehört ein Implementierungsteil zur Klasse, so enthält ein gleichnamiges Modul mit Endung .cpp die Implementierung der Klassenmethoden. Die Modulnamen werden den entsprechenden Klassenbezeichnungen entnommen.

Die Ausnahmen bilden die Klassen, die ihren Implementierungsteil erben, ohne eigene Methoden zur Verfügung zu stellen.

7.2.2 Besonderheiten im Quelltext

In Anlehnung an ein in Smalltalk übliches Konzept geben öffentliche Klassenmethoden, die normalerweise `void` zurückliefern würden, eine Referenz auf die Instanz der Klasse zurück, deren Methode aufgerufen wurde. Somit werden folgende Konstrukte möglich:

```
EineKlasse instanz;     //Instanz einer Klasse erzeugen
instanz
  .ersteOperation()
  .zweiteOperation()
  .dritteOperation()
    :
  ;
```

Anhand einer Reihe von Makros, die in der Datei macros.h definiert werden, können Klassenschnittstellen leichter und konsistenter erstellt werden, mit dem zusätzlichen Vorteil, dass der resultierende Code überschaubarer wird. Diese Makros umfassen die wesentlichen Elemente einer Klasse. So gibt es beispielsweise ein Makro für Klassenattribute, das folgendermaßen definiert wird:

```
#define ATTRIBUTE(type, getAttr, setAttr)             \
   private:                                           \
     type attr_##getAttr;                             \
   public:                                            \
     type  getAttr() const { return attr_##getAttr; } \
     This& setAttr( type __x ) { attr_##getAttr = __x; return *this; }
```

Nun kann ein Klassenattribut durch die Zeile

```
  ATTRIBUTE ( Typ, einAttribut, setEinAttribut )
```

in der Klassenschnittstelle angegeben werden. Von außen darf `attr_einAttribut` nur über den Aufruf `einAttribut()` gelesen oder `setEinAttribut(neuerWert)` gesetzt werden. Diese Makros setzen einen vorangestellten Aufruf des Makros `DECLARE` voraus, denn hier wird der `This`-Typ definiert, womit der oben beschriebene Rückgabemechanismus auch bei den Makros eingehalten wird. Es existieren einige solcher Makros, die in der Tabelle 7.2 angegeben werden.

DECLARE	Vorbereitung der Klasse auf die Benutzung der Makros
ATTRIBUTE	Attribut mit öffentlichen Methoden zum Lesen und Setzen
CRITICAL_ATTRIBUTE	Attribut, das von außen nicht gesetzt werden darf
CLASS_ATTRIBUTE	statisches Attribut
CRITICAL_CLASS_ATTRIBUTE	statisches Attribut, das von außen nicht gesetzt werden darf
FLAG	boolesches Attribut, mit Methoden zum Aktivieren, Deaktivieren, Lesen und Setzen
CRITICAL_FLAG	boolesches Attribut, das von außen nicht geändert werden kann
CONTAINS	Container-Beziehung beliebiger Kardinalität mit diversen Zugriffsmethoden

Tabelle 7.2: Die Makros der Datei macros.h.

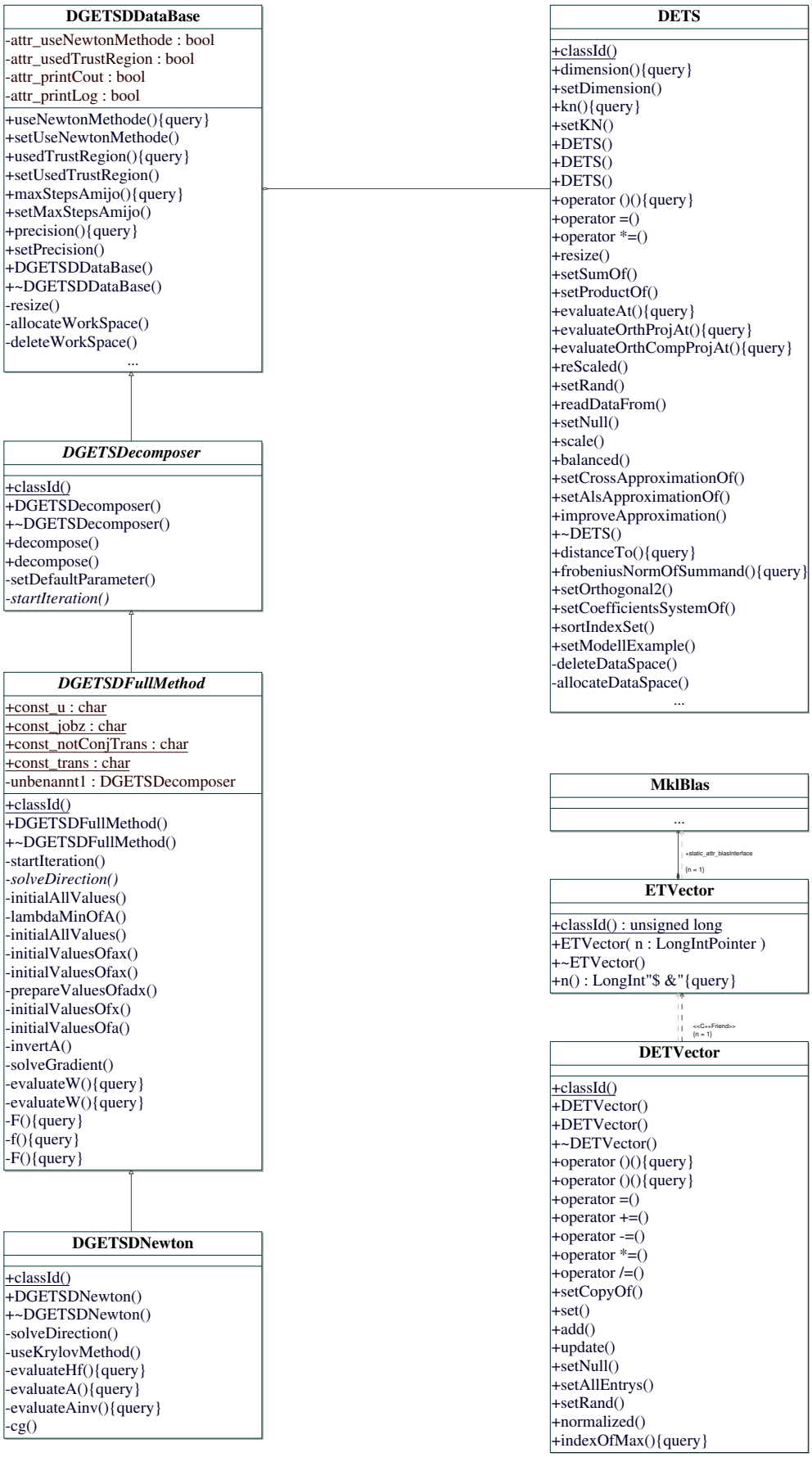


Abbildung 7.1: Vereinfachtes Klassendiagramm.

8 Zusammenfassung

Im ersten Kapitel wurden grundlegende Definitionen und Eigenschaften des Tensorproduktes erläutert und die Darstellung von Tensoren in Untervektorräumen untersucht.

Im anschließenden Kapitel stand die beste Approximation mit Summen von Elementartensoren im Zentrum der Analyse. Mit Hilfe von Satz 2.2.4 konnte die Approximation auf einen endlich-dimensionalen Teilraum eingeschränkt werden, wobei für die praktische Umsetzung vorausgesetzt war, dass der Koeffiziententensor und eine orthonormale Basis leicht zu berechnen sind. Mit Lemma 2.3.1 wurde gezeigt, dass dieser Unterraum im Allgemeinen nicht abgeschlossen ist, demzufolge kann die Existenz einer besten Approximation nicht garantiert werden. Dieser Mangel führte zur Definition von Elementartensoren mit beschränkten Summanden.

Nachdem die Existenz der besten Approximation gesichert war, konnte im dritten Kapitel die Approximationsaufgabe präzise formuliert werden. Daneben sind die Zielfunktion und die erste und zweite Ableitung der Zielfunktion dargestellt. Dies war für die numerische Umsetzung von Bedeutung.

Zur besseren Übersicht wurden bekannte numerische Verfahren zur Behandlung nichtlinearer Optimierungsaufgaben im vierten Kapitel zusammengetragen. Hierbei wurde besonderer Wert auf Methoden zur Lösung der Approximationsaufgabe gelegt.

Im fünften Kapitel stand die Lösung der Approximationsaufgabe im Mittelpunkt der Untersuchung. Im Besonderen wurde die praktische Umsetzung eines Minimierungsschrittes beschrieben und seine Komplexität angegeben.

Anhand spezifischer Anwendungen wurden die vorgestellte Methode im sechsten Kapitel revidiert und neue Berechnungsmethoden aufgezeigt. So konnte z. B. mit Hilfe des Approximationsverfahrens auf die ungewisse Frage nach einer effizienten Methode zur Berechnung der Maximumnorm einer Elementartensor-Summe und des zugehörigen Index eingegangen werden.

Die vorliegende Arbeit enthält eine neue iterative Methode zur lokal besten Approximation von Elementartensor-Summen in hohen Dimensionen mittels Tensoren niederen Tensorrangs. Das Konvergenzverhalten dieses Verfahrens ist in Satz 4.4.3 beschrieben. Die Anzahl der nötigen Minimierungsschritte konnte nicht abgeschätzt werden, wenn auch bei numerisch sinnvoll gestellten Aufgaben eine geringe Anzahl zu beobachten war. Insgesamt verbessert diese Studie den

effizienten Einsatz von Elementartensor-Summen und stellt damit ein neues, robustes Hilfsmittel zur Lösung hochdimensionaler Probleme bereit.

Literaturverzeichnis

[1] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives, Pac. J. Math. 16, 1-3, 1966.

[2] R. Bellman. Adaptive control processes: A guided tour. Princeton University Press, XVI, 255 p., 1961.

[3] D. Braess, W. Hackbusch. Approximation of $1/x$ by Exponential Sums in $[1, \infty)$, IMA Numer. Anal. 25, 685-697, 2005.

[4] H.J. Bungartz. Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, Institut für Informatik, Technische Universität München, 1992.

[5] S. R. Chinnamsetty. Wavelet Tensor Product Approximation in Electronic Structure Calculations, Dissertation, Universität Leipzig, 2007.

[6] S. R. Chinnamsetty, M. Espig, B. N. Khoromskij, W. Hackbusch, and H. J. Flad. Tensor product approximation with optimal rank in quantum chemistry, submitted to: Journal of Chemical Physics, 2007.

[7] I. P. Gavrilyuk, W. Hackbusch, B. N. Khoromskij. Hierarchical tensor-product approximation to the inverse and related operators for high-dimensional elliptic problems., Computing 74, No. 2, 131-157, 2005.

[8] C. Geiger, C. Kanzow. Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben, Springer-Verlag, Hamburg, 1999.

[9] C. Geiger, C. Kanzow. Theorie und Numerik restringierter Optimierungsaufgaben, Springer-Verlag, Hamburg und Würzburg, 2001.

[10] G. H. Golub, C. F. van Loan. Matrix Computations, third edition, Johns Hopkins Studies in the Mathematical Sciences, Third Edition, 1996.

[11] L. Grasedyck. Theorie und Anwendungen Hierarchischer Matrizen, Dissertation, Christian-Albrechts-Universität zu Kiel, 2001.

[12] L. Grasedyck. Existence and Computation of a Low Kronecker-Rank Approximant to the Solution of a Tensor System with Tensor Right-Hand Side, Computing 72 3-4, 247-265, 2004.

[13] W. H. Greub. Multilinear Algebra, Springer-Verlag, 1nd Edition, New York, 1967.

[14] M. Griebel. A parallelizable and vectorizable multi-level algorithm on sparse grids, Parallel algorithms for partial differential equations, Proc. 6th GAMM- Semin., Kiel 1990, Notes Numer. Fluid Mech. 31, 94-100, 1991.

[15] W. Hackbusch. Iterative Lösung großer schwachbesetzter Gleichungssysteme, B.G. Teubner Stuttgart, 2. Auflage, Kiel, 1992.

[16] W. Hackbusch. Entwicklungen nach Exponentialsummen, Technischer Report, Nr. 4/2005, Leipzig, 2005.

[17] W. Hackbusch, B. N. Khoromskij, E. E. Tyrtyshnikov. Hierarchical Kronecker tensor-product approximations, J. Numer. Math. 13, No. 2, 119-156 2005.

[18] W. Hackbusch, B. N. Khoromskij, E. E. Tyrtyshnikov. Approximate Iterations for Structured Matrices, Numer. Math., 2007.

[19] W. Hackbusch, B. N. Khoromskij. Low-Rank Kronecker-Product Approximation to Multi-Dimensional Nonlocal Operators. Part I. Separable Approximation of Multi-Variate Functions, Computing 76, 3/4 177-202, 2006.

[20] W. Hackbusch, B. N. Khoromskij. Low-Rank Kronecker Product Approximation to Multi-Dimensional Nonlocal Operators. Part II. HKT Representation of Certain Operators, Computing 76, 3/4 203-225, 2006.

[21] W. Hackbusch, B. N. Khoromskij. Tensor-Product Approximation to Multi-Dimensional Integral Operators and Green's Functions, SIAM J. Matrix Anal. Appl., 2007.

[22] J. Håstad. Tensor Rank is NP-complete, Lecture Notes In Computer Science; Vol. 372: 451-460, 1989.

[23] C. Kenney, A. J. Laub. Rational iterative methods for the matrix sign function, SIAM J. Matrix Anal. Appl. 12 273-291, 1991.

[24] P. Kosmol. Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben, B.G. Teubner Stuttgart, 2. Auflage, Kiel, 1993.

[25] J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, Linear Algebra Appl. 18, 95-138, 1977.

[26] J. B. Kruskal. Multidimensional scaling and other methods for discovering structure, Stat. Math. digital Comput. 296-339, 1977.

[27] J. B. Kruskal. Multilinear and geometrical models: Overview and relationships, Analyse de donnees et informatique, Fontainebleau 1979, 53-66, 1980.

[28] J. de Leeuw, Y. Takane, F. W. Young. Additive structure in qualitative data: an alternating least squares method with optimal scaling features, Psychometrika, 41, 471p., 1976.

[29] C. F. van Loan. The ubiquitous Kronecker product, J. Comput. Appl. Math. 123, No.1-2, 85-100, 2000.

[30] M. J. Mohlenkamp, G. Beylkin. Numerical operator calculus in higher dimensions, Proceedings of the National Academy of Sciences, 99(16):10246-10251, 2002.

[31] M. J. Mohlenkamp, G. Beylkin. Algorithms for Numerical Analysis in High Dimensions, SIAM Journal on Scientific Computing, 26(6):2133-2159, 2005.

[32] B. Oestereich. Objektorientierte Softwareentwicklung - Analyse und Design mit der Unified Modeling Language, R. Oldenbourg Verlag München Wien, 4. Auflage, 1998.

[33] I. V. Oseledets, D. V. Savost'yanov. Minimization methods for approximating tensors and their comparison, Computational Mathematics and Mathematical Physics, Volume 46 (10): 1641-1650, 2006.

[34] P. Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis, Chemometrics and Intelligent Laboratory Systems, 38 223-242, 1997.

[35] P. Paatero. The multilinear engine-a table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model. Journal of Computational and Graphical Statistics, 8(4) 1-35, 1999.

[36] C. Schwab, R.A. Todor. Sparse finite elements for stochastic elliptic problems - higher order moments, Computing 71, No. 1, 43-63, 2003.

[37] H. Schwetlick. Numerische Lösung nichtlinearer Gleichungen, VEB Deutscher Verlag der Wissenschaften, Dresden, 1978.

[38] V. de Silva, L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem, SCCM-06-06, Stanford, 2006.

[39] U. Storch, H. Wiebe. Lehrbuch der Mathematik, Band 4, Analysis auf Mannigfaltigkeiten, Funktionentheorie, Funktionalanalysis, Spektrum Akademischer Verlag, 2001.

[40] V. Strassen. Rank and Optimal Computation of Generic Tensors, Linear Algebra Appl. 52/53: 645-685, 1983.

[41] A. van der Sluis, H. A. van der Vorst. The rate of convergence of conjugate gradients, Numer. Math. 48, 543-560, 1986.

[42] E. E. Tyrtyshnikov. Tensor approximations of matrices generated by asymptotically smooth functions, Sb. Math. 194, No. 6, 941-954, 2003.

[43] E. E. Tyrtyshnikov. Kronecker-product approximations for some function-related matrices, Linear Algebra Appl. 379, 423-437 2004.

[44] D. Werner. Funktionalanalysis, Springer Verlag, 5. Auflage, 2004.

[45] T. Yokonuma. Tensor Spaces and Exterior Algebra. American Mathematical Society, 1991.

[46] F. Yates The analysis of replicated experiments when the field results are incomplete. Emp. J. Exp. Agric. 1, 129-142, 1933.

[47] W. C. Yueh Eigenvalues of several tridiagonal matrices. Applied Mathematics E-Notes 5, 66-74, 2005.

[48] C. Zenger. Sparse Grids, Parallel Algorithms for Partial Differential Equations. Parallel algorithms for partial differential equations, Proc. 6th GAMM- Semin., Kiel 1990, Notes Numer. Fluid Mech. 31, 241-251, 1991.

Hilfsmittel

Software-Bibliotheken

- **BLAS (Basic Linear Algebra Subprograms)**, http://www.netlib.org/blas/.
- **LAPACK (Linear Algebra PACKage)**, http://www.netlib.org/lapack/.