Dimension algebraischer Varietäten und numerische Methoden

Dissertation

zur Erlangung des akademischen Grades
„Doktor der Naturwissenschaften“
(Dr. rer. nat.)

am Fachbereich
Mathematik und Informatik, Physik, Geographie
der Justus-Liebig-Universität Gießen

vorgelegt von
Dipl.-Math. Johannes Czekansky

Betreuer:
Prof. Dr. Martin Buhmann
Lehrstuhl für Numerische Mathematik
Justus-Liebig-Universität Gießen

Zweitgutachter:
Prof. Dr. Tomas Sauer
Lehrstuhl für Mathematik mit
Schwerpunkt Digitale Bildverarbeitung
Universität Passau

An dieser Stelle möchte ich einigen Personen für Ihre Hilfe in vielen Dingen danken. Herr Prof. Dr. Tomas Sauer hat mich mit dem Thema approximative H-Basen vertraut gemacht und diese Arbeit weitgehend motiviert. Zahlreiche Besuche in Passau ermöglichten aufschlussreiche Diskussionen und haben viele gute Hinweise geliefert. Dafür gilt ihm mein besonderer Dank. Herrn Prof. Dr. Martin Buhmann danke ich für die Betreuung dieser Arbeit und die Aufnahme in seine Arbeitsgruppe. Die angenehme Atmosphäre und die gute Zusammenarbeit mit den Kollegen haben mir stets optimale Arbeitsbedingungen geboten. Dabei möchte ich meinen Kommilitonen und Kollegen Frank Lamping hervorheben, dem ich für zahllose wertvolle Diskussionen danke.

Weiterhin danke ich meiner Familie für die großartige Unterstützung in den letzten Jahren. Jeder von Euch hat seinen Anteil an dieser Arbeit – sei es unmittelbar durch Korrekturlesen oder einfach nur als der notwendige Gegenpol zu meiner Arbeit, der dafür sorgte, dass ich die Motivation nie verloren habe und stets den nötigen Abstand hatte.

Zusammenfassung

Als *algebraische Varietäten* bezeichnet man Mengen aller gemeinsamen Nullstellen einer endlichen Menge von Polynomen. Die Polynome, die an allen Punkten einer algebraischen Varietät *exakt* verschwinden, bilden ein Ideal, das sogenannte *Verschwindungsideal*. Verschwindungsideale zeichnen sich damit durch eine Forderung aus, die numerisch nicht haltbar ist. Daher wurde u. a. von Sauer eine numerische Entsprechung, die *approximativen Ideale* eingeführt und ein Verfahren zur Konstruktion *approximativer H-Basen* angegeben. Diese Arbeit entwickelt den Ansatz der approximativen H-Basis weiter, wobei Startwertabhängigkeiten sowie die Auswirkung einer schrittweisen Interpolation untersucht werden. Dies führt zu neuen Verfahren zur Konstruktion approximativer H-Basen, insbesondere bzgl. der verwendeten Norm. Weiterhin werden neue Abschätzungen für Rechenoperationen auf approximativen Idealen angegeben, die aufgrund ihrer approximativen Definition die bekannten Abschlusseigenschaften eines Ideals nicht mehr erfüllen. Es werden Methoden zur Bestimmung dünn besetzter H-Basen konstruiert und das *approximative Ideal-Membership-Problem* besprochen. Als Anwendungsmöglichkeit wird die Gelenkerkennung in kinematischen Ketten vorgestellt. Zudem werden numerische Vergleiche mit anderen Methoden, u. a. approximativen Randbasen, durchgeführt.

Abstract

Algebraic varieties are sets of common zeros of a finite set of polynomials. The polynomials vanishing *exactly* at all points of an algebraic variety form an ideal, denoted as *vanishing ideal*. By definition vanishing ideals are based on conditions that do not hold in numerical situations. Therefore Sauer and others introduced *approximate ideals* and also presented a method for constructing *approximate H-bases*. This thesis advances the approach of approximate H-bases and gives an analysis of dependency of initial values and stepwise interpolation. This yields new methods for constructing approximate H-bases, especially in terms of the used norm. Moreover new estimates for operations in approximate ideals are given because the closure conditions of ideals do not hold in the approximate case. We also give methods for calculating

sparse H-bases and discuss the *approximate ideal-membership-problem*. As an application we present a method for joint-detection in kinematic chains. Furthermore we numerically compare approximate H-bases with other methods like approximate border bases.

Inhaltsverzeichnis

1. Einleitung und Motivation **1**

1.1. Gliederung der Arbeit 3

1.2. Wichtige Resultate und Thesen 5

2. Multivariate Polynome **7**

2.1. Grundlagen und Notationen 8

2.2. Speicherung von Koeffizientenvektoren 14

2.3. Auswertung von multivariaten Polynomen 20

2.4. Rechnen mit multivariaten Polynomen 26

2.4.1. Addition/Subtraktion von Polynomen 26

2.4.2. Multiplikation von Polynomen mit Termen 27

2.4.3. Multiplikation von Polynomen 29

2.4.4. Berechnung von Basen homogener Teilräume 37

2.4.5. Polynomdivision 43

3. Varietäten und Ideale **49**

3.1. Ideale und Idealbasen 50

3.2. Darstellung von Idealen 53

3.3. Varietäten und der Nullstellensatz 62

3.4. Endlichkeit vs. Geometrie 67

4. Approximative Ideale **75**
4.1. Approximative H-Basen . . . 77
4.1.1. Numerische Bestimmung einer approximativen H-Basis . . . 78
4.1.2. Wahl der Toleranzschwelle . . . 88
4.1.3. Wahl des Startpunkts . . . 90
4.1.4. Interpolation vs. Approximation . . . 97
4.1.5. Alternative Zerlegung . . . 104
4.2. Ringoperationen auf approximativen Idealen . . . 107
4.2.1. Addition und Linearkombination . . . 107
4.2.2. Multiplikation mit Monomen . . . 109
4.2.3. Multiplikation mit Polynomen . . . 112
4.3. Das approximative Ideal-Membership-Problem . . . 125
4.4. Dünn besetzte H-Basen . . . 128

5. Analyse von Daten aus kinematischen Systemen **137**
5.1. Planare kinematische Ketten . . . 138
5.1.1. Charakterisierende Gelenkbedingungen . . . 139
5.1.2. Methoden zur Gelenkerkennung . . . 142
5.2. Kinematische Ketten im Raum . . . 152
5.2.1. Drehgelenke mit fester Drehebene . . . 154
5.2.2. Weitere Gelenktypen . . . 160

6. Weitere Anwendungen und numerische Ergebnisse **165**
6.1. Fehlerbehaftete Messdaten aus einfachen geometrischen Strukturen . 166
6.2. H-Basen vs. Randbasen . . . 170
6.3. Implizite Funktionen . . . 173
6.4. Datensätze mit Ausreißern . . . 176

7. Zusammenfassung und Ausblick **179**

A. Funktionsübersicht **183**
A.1. Datenstrukturen . . . 183
A.2. Polynomoperationen . . . 185
A.3. H-Basen . . . 188
A.4. Approximative H-Basen . . . 189

A.5. Wichtige Hilfsfunktionen 190
A.6. Planare kinematische Ketten 195

Symbolverzeichnis **197**

Liste der Algorithmen **202**

Abbildungsverzeichnis **203**

Literaturverzeichnis **205**

Stichwortverzeichnis **213**

1 Einleitung und Motivation

Inhalt

1.1. Gliederung der Arbeit 3

1.2. Wichtige Resultate und Thesen 5

Die Untersuchung von *algebraischen Varietäten* ist ein Problem aus der klassischen algebraischen Geometrie, vgl. [Grö49]. Dabei definiert der Begriff der *Varietät* ein geometrisches Objekt, das sich durch polynomielle Gleichungen beschreiben lässt. Ein einfaches Beispiel ist ein Kreis um den Ursprung in der Ebene, der durch die quadratische Gleichung $x^2 + y^2 - r^2 = 0$ gegeben ist.

In den Anfängen der algebraischen Geometrie wurden vor allem abstrakte Aussagen über die Struktur von Varietäten getroffen. Durch den engen Zusammenhang von Varietäten und Polynomidealen ermöglichte insbesondere die Entwicklung der *Gröbnerbasen* durch Buchberger im Jahr 1965 (siehe [Buc65]) die Konstruktion effizienter Verfahren für den Umgang mit Varietäten. Dieser algorithmische Zugang führte zu Begriffen wie *Computational Algebraic Geometrie* (D. Cox, J. Litte, D. O'Shea, vgl. [CLO07]) oder *Computational Commutative Algebra* (M. Kreuzer, L. Robbiano, vgl. [KR00]).

Gröbnerbasen sind mittlerweile in vielen Computeralgebrasystemen wie Singular [DGPS14] oder CoCoA [CT14] verfügbar und werden durch die *Symbolic Math*

Toolbox [TM14b] auch von MATLAB [TM14a] unterstützt. Grundlage all dieser Implementierungen ist jedoch eine Umgebung, die symbolisches Rechnen ermöglicht. Durch die Einschränkung auf numerische Rechnung und endliche Rechengenauigkeit können Gröbnerbasen jedoch instabil werden. Dies haben H.-M. Möller und T. Sauer in [MS00c] gezeigt. Eine numerisch stabile Alternative bieten *H-Basen*, teilweise auch *Macaulay-Basen* genannt (vgl. [HKPP09]), die bereits 1916 von F. S. Macaulay in [Mac16] beschrieben wurden. Eine Implementierung von H-Basen sucht man in den gängigen Computeralgebrasystemen allerdings vergeblich. Selbst die nach Macaulay benannte Software *Macaulay2* [GSE14] unterstützt nur Gröbnerbasen.

Alle Verfahren, die auf Gröbnerbasen oder H-Basen basieren, setzen exakte Daten voraus, wohingegen die numerische Rechnung in der Regel auf näherungsweisen bzw. fehlerbehafteten Daten – bedingt durch Rundungsfehler, Messungenauigkeiten oder Ähnliches – basiert. Um dies zu berücksichtigen, wurde das Konzept der *approximativen Ideale* von T. Sauer in [Sau07] bzw. der *Almost Vanishing Ideals* von D. Heldt, M. Kreuzer, S. Pokutta und H. Poulisse in [HKPP09] eingeführt. Da bisher keine Umsetzung von H-Basen bzw. approximativen H-Basen in numerischer Software existiert, besteht ein wesentlicher Teil dieser Arbeit in der Implementierung effizienter Verfahren für die frei verfügbare Software GNU OCTAVE [EBH08], deren Syntax mit MATLAB kompatibel ist.

Aus einer approximativen H-Basis des Ideals einer algebraischen Varietät können dann Rückschlüsse auf deren geometrische Eigenschaften gezogen werden. Mit anderen Worten: Man untersucht, ob eine endliche Varietät in einer Untermannigfaltigkeit des d-dimensionalen Raumes liegt. Die Forderung der Endlichkeit der Varietät beruht dabei auf der Endlichkeit des Speichers in Computersystemen. Für gemessene oder abgetastete Werte ist dies ohnehin eine natürliche Bedingung. Allerdings erfordern endliche Varietäten eine Unterscheidung zwischen trivialen Informationen in der H-Basis, die lediglich die Endlichkeit der Punktmenge beschreiben, und den Informationen, die auf geometrische Eigenschaften schließen lassen.

Alle Berechnungen werden dabei auf der algebraischen Seite der Varietät-Ideal Dualität durchgeführt. Dies erfordert einerseits die Übertragung wichtiger Methoden wie die Lösung des *Ideal Membership Problems* auf approximative Ideale. Andererseits muss das Verhalten von approximativen Idealen in numerischer Rechnung unter-

sucht werden, da approximative Ideale definitionsgemäß Toleranzen aufweisen, die sich durch jede Rechenoperation verändern können.

Letztlich sind *reduzierte H-Basen* im Gegensatz zu *reduzierten Gröbnerbasen* nicht eindeutig. Die dadurch entstehenden Freiheitsgrade kann man nutzen, um möglichst dünn besetzte Basen, d. h. Basen, deren Elemente möglichst wenige von Null verschiedene Koeffizienten haben, zu bestimmen. Dies ermöglicht dann unter anderem die Analyse von Bewegungsprofilen planarer kinematischer Ketten *ohne* das Ideal-Membership-Problem explizit zu lösen.

1.1. Gliederung der Arbeit

In dieser Arbeit wird die Untersuchung algebraischer Varietäten in einer numerischen Umgebung beschrieben. Dazu stellt Kapitel 2 zunächst einen konzeptionellen Rahmen bereit, der grundlegende Methoden zur Speicherung von multivariaten Polynomen und deren Auswertung mit Hilfe des Horner-Schemas beschreibt. Ebenso wird die Effizienz sowie die numerische Stabilität dieses Verfahrens aufgezeigt. Es werden Algorithmen für elementare Rechenoperationen wie Addition und Multiplikation von multivariaten Polynomen angegeben und im Falle der Multiplikation zwei unterschiedliche Ansätze verglichen. In diesem Zusammenhang wird mit der Faltungsmatrix ein wichtiges Hilfsmittel eingeführt, das für spätere Resultate von zentraler Bedeutung ist. Dem folgt ein Verfahren zur Konstruktion homogener linearer Teilräume in Abhängigkeit einer Menge von Polynomen, das die Grundlage der numerischen Bestimmung von H-Basen bildet und mit dessen Hilfe auch ein Algorithmus zur Polynomdivision formuliert werden kann.

Kapitel 3 beschreibt die Darstellung von Polynomidealen durch H-Basen und ermöglicht so die Einbettung in den in Kapitel 2 geschaffenen Rahmen. Der dabei hergestellte Zusammenhang zwischen algebraischen Varietäten und Polynomidealen basiert im Wesentlichen auf den Ausführungen von D. Cox, J. Little und D. O'Shea in [CLO07]. Anschließend wird ein Verfahren zur Generierung von H-Basen nulldimensionaler Polynomideale vorgestellt und an Beispielen erläutert, das auf die Arbeit von H.-M. Möller und T. Sauer zurückgeht. Mit Hilfe dieser Grundlagen kann man zeigen, dass die Anzahl der Basispolynome eines bestimmten Grades eine Invariante

aller H-Basen eines Ideals ist. Dieses Resultat basiert auf den Untersuchungen von Möller und Sauer in [MS00a]. Zur Differenzierung zwischen dem Endlichkeits- und dem Geometrieanteil einer H-Basis wird abschließend ein neues Resultat in Form eines notwendigen Kriteriums für die Existenz von Polynomen im Geometrieanteil bewiesen.

Der Übergang zu approximativen Idealen findet in Kapitel 4 statt, wobei zunächst der Algorithmus zur Bestimmung einer approximativen H-Basis von T. Sauer aus [Sau07] präsentiert und analysiert wird. Dabei werden Probleme bei der Wahl eines geeigneten Startwerts aufgezeigt und verschiedene neue Modifikationen und Erweiterungen des Verfahrens angegeben. Weiterhin findet eine Fehleranalyse der Rechenoperationen auf approximativen Idealen statt, die speziell im Fall der Polynommultiplikation ein neues Resultat beinhaltet. Dem folgt eine Untersuchung des approximativen *Ideal Membership Problems*, die sich auf die Arbeit von D. Heldt, M. Kreuzer, S. Pokutta und H. Poulisse in [HKPP09] stützt. Außerdem wird ein neues Verfahren zur Bestimmung dünn besetzter approximativer H-Basen vorgestellt und zu diesem Zweck die *approximative* 0-*Norm* eingeführt, die als Maß für die Besetztheit eines Koeffizientenvektors unter numerischen Bedingungen verwendet werden kann.

Kapitel 5 zeigt eine praktische Anwendung der in dieser Arbeit gewonnenen Resultate: die Analyse von kinematischen Ketten und deren Bewegungsprofilen, d. h. endlichen Mengen möglicher Gelenkpositionen. Es werden neue Methoden präsentiert, die anhand dieser Bewegungsprofile eine Rekonstruktion des kinematischen Systems, im Sinne von strukturellen Eigenschaften wie Gelenktypen und Verkettungen, ermöglichen. Diese Methoden basieren einerseits auf der expliziten Lösung des *Ideal Membership Problems* sowie alternativ auf den Verfahren zur Bestimmung dünn besetzter approximativer H-Basen aus Kapitel 4.

In Kapitel 6 werden numerische Ergebnisse der vorgestellten Verfahren und Implementierungen in verschiedenen Anwendungen gezeigt. Dabei dienen einfache polynomielle Beziehungen dem Vergleich mit bereits vorhandenen Implementierungen ähnlicher Verfahren in ApCoCoA [AT13]. Als Referenz dazu werden die Ergebnisse von J. Limbeck aus [Lim13] herangezogen, die auf einer Erweiterung des Verfahrens von D. Heldt, M. Kreuzer, S. Pokutta und H. Poulisse aus [HKPP09] basieren. Anschließend werden die Unterschiede zwischen approximativen H-Basen und approxi-

mativen Randbasen bezüglich der Anwendbarkeit auf bestimmte Probleme erörtert, wobei die in [Lim13] genannten Punkte unter Berücksichtigung der hier vorgestellten Verfahren aufgegriffen werden. Eine weitere präsentierte Anwendung ist die Berechnung impliziter Darstellungen von parametrischen Kurven. Dies führt zu einem Vergleich mit den Resultaten von C. Fassino und M.-L. Torrente aus [FT13].

Kapitel 7 liefert abschließend eine Zusammenfassung der gewonnenen Resultate und zeigt weitere Ansätze und Möglichkeiten auf.

Im Anhang werden die Implementierungen aller hier untersuchten Verfahren besprochen. Die Quellcodes der Funktionen für die frei verfügbare Software GNU Octave können als SVN-Repository über folgende Adresse bezogen werden:

`https://subversion.forwiss.uni-passau.de/publications/Dissertation-Czekansky/`

Alle Funktionen können alternativ auch mit Matlab verwendet werden.

1.2. Wichtige Resultate und Thesen

1. Die bekannten Verfahren zur Konstruktion approximativer Ideale von Sauer (siehe [Sau07]) bzw. Heldt, Kreuzer, Pokutta und Poulisse (siehe [HKPP09]) benötigen effiziente Verfahren zur Auswertung von multivariaten Polynomen. Dazu eignen sich insbesondere Methoden, die auf dem Horner-Schema basieren. Eine Untersuchung solcher Algorithmen im Sinne von Rechenaufwand und numerischer Stabilität wurde vom Autor in [CS14] durchgeführt. Die Ergebnisse sind in Abschnitt 2.3 zusammengefasst.

2. In Satz 3.32 wird ein neues Kriterium entwickelt und bewiesen, das die Entscheidung, ob eine endliche Punktmenge in einer niederdimensionalen Untermannigfaltigkeit liegt, ermöglicht. Zudem werden Tabellen angegeben, die für bestimmte Fälle eine Lösung dieses Problems durch einfaches Nachschlagen erlauben.

3. Der Algorithmus zur Konstruktion einer approximativen H-Basis von Sauer ist auf die Verwendung der Maximumsnorm beschränkt und hängt von einem Startwert ab. Mit Algorithmus 4.23 und Algorithmus 4.26 werden zwei neue Modifikationen des Verfahrens für die Maximumsnorm sowie für die euklidische Norm angegeben, die keine Startwertabhängigkeit aufweisen. Im Falle der euklidischen Norm ist die Unabhängigkeit von einem Startwert eine notwendige Eigenschaft.

4. Da bisher nur die Addition in approximativen Idealen untersucht wurde (siehe [Sau07]), präsentiert Satz 4.41 eine neue Fehlerabschätzung für die Multiplikation in approximativen Idealen. Die dabei konstruierte Schranke basiert auf dem kleinsten Singulärwert einer Faltungsmatrix, für den in Satz 4.52 ebenfalls eine neue Abschätzung bereitgestellt wird. Diese Aussage verschärft ein bekanntes Resultat von Batselier aus [Bat13].

5. In Algorithmus 4.65 wird ein neues Verfahren konstruiert, das zu einer gegebenen H-Basis eine bzgl. der approximativen 0-Norm minimale H-Basis – im Sinne von dünn besetzten Koeffizientenvektoren – desselben Ideals berechnet. Dazu kann ein bekanntes Invarianzkriterium von Möller und Sauer genutzt werden, dessen Herleitung in Satz 3.19 zu finden ist.

6. Kapitel 5 präsentiert zwei neue Verfahren zur Gelenkanalyse in kinematischen Ketten und vergleicht diese. Dabei ist Satz 5.4 ein zentrales Resultat, das die Minimalität charakterisierender Gelenkbedingungen bzgl. der 0-Norm belegt.

2 Multivariate Polynome

Inhalt

2.1. Grundlagen und Notationen 8
2.2. Speicherung von Koeffizientenvektoren 14
2.3. Auswertung von multivariaten Polynomen 20
2.4. Rechnen mit multivariaten Polynomen 26

Im klassischen Verständnis eines Polynoms liegt diesem immer eine Ringstruktur zugrunde, aus der die Koeffizienten entnommen sind. So entsprechen etwa die aus der Schulmathematik bekannten *ganzrationalen Funktionen* genau den univariaten Polynomen über den reellen Zahlen $\mathbb{R}$. Wollen wir mit diesen Polynomen in einem Computersystem rechnen, so sorgt die Endlichkeit des Speicherplatzes dafür, dass viele Polynome nicht darstellbar sind. Bereits das Polynom $f(x) = \frac{1}{3}x$ lässt sich wegen der unendlichen Dezimalbruchentwicklung von $\frac{1}{3}$ nicht in einem Standarddatentyp, wie `int`, `single`, `float` etc., speichern. Zusätzliche Bibliotheken (vgl. *GNU Multiple Precision Arithmetic Library*, [Fre14]) ermöglichen zwar die Verwendung von rationalen Zahlen, aber auch diese sind durch den endlichen Speicher beschränkt. Demnach wird es – unabhängig von der Größe des verfügbaren Speichers – immer Polynome über $\mathbb{Q}$ geben, die nicht von einem Computersystem verarbeitet werden können.

Im Gegensatz zu vielen *Computeralgebrasystemen*, die Polynome mit rationalen Koeffizienten verarbeiten können, beschränkt sich diese Arbeit auf die in der Numerik üblichen *Gleitkommazahlen*, also eine Darstellung mit endlicher Genauigkeit (vgl. dazu [SK11, 1.2], [FH07, 1.1], [SW05, 1.2]). Dadurch sind natürlich auch nur näherungsweise Ergebnisse zu erwarten, was sich in vielen Rechenoperationen, insbesondere auch bei der Auswertung eines Polynoms, bemerkbar macht. Es ist daher zu beachten, dass die im folgenden Abschnitt definierten Polynome über $\mathbb{R}$ tatsächlich nur Koeffizienten in einer *Gleitkommadarstellung* mit endlicher Genauigkeit haben.

Multivariate Polynome, also Polynome in mehreren Variablen $x_1, \ldots, x_d$ für $d > 1$, unterscheiden sich auf den ersten Blick nicht wesentlich von den univariaten Polynomen. Die ersten Schwierigkeiten ergeben sich aber bereits in der Anordnung der beteiligen Summanden.

Da multivariate Polynome und deren numerische Behandlung eine wesentliche Grundlage dieser Arbeit bilden, werden in diesem Kapitel zunächst einige elementare Konzepte und Notationen eingeführt. Anschließend werden Methoden zur Speicherung und effizienten Auswertung multivariater Polynome vorgestellt und Algorithmen für die Grundrechenoperationen – Addition, Subtraktion, Multiplikation und Division – von multivariaten Polynomen angegeben.

2.1. Grundlagen und Notationen

Im Folgenden werden einige grundlegende Begriffe definiert, die für den Umgang mit multivariaten Polynomen benötigt werden. Wir beginnen mit einem wichtigen Hilfsmittel, das die Notation deutlich vereinfacht: den Multiindizes.

Definition 2.1. *Die Elemente* $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ *bezeichnet man als* Multiindizes. *Dabei gilt* $\alpha! = \alpha_1! \cdots \alpha_d!$ *und* $|\alpha| = \alpha_1 + \cdots + \alpha_d$, *sowie* $x^\alpha = x_1^{\alpha_1} \cdots x_d^{\alpha_d}$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

Tatsächlich sind die Multiindizes aber mehr als nur eine abkürzende Schreibweise. Ihnen liegt eine algebraische Struktur zugrunde, die es ermöglicht, Rechenoperationen auf Monomen in Operationen auf den Multiindizes zu überführen. Die Multiindizes

$\alpha \in \mathbb{N}_0^d$ bilden mit der komponentenweisen Addition einen kommutativen *Monoid*. Mit anderen Worten: Es existiert eine binäre Verknüpfung $+ : \mathbb{N}_0^d \times \mathbb{N}_0^d \to \mathbb{N}_0^d$ mit den Eigenschaften $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$ und $\alpha + \beta = \beta + \alpha$ für $\alpha, \beta, \gamma \in \mathbb{N}_0^d$ und ein Element $0 = [0, \ldots, 0] \in \mathbb{N}_0^d$, sodass für alle $\alpha \in \mathbb{N}_0^d$ stets $\alpha + 0 = \alpha$ gilt.

Fassen wir die Menge aller Multiindizes $\mathbb{N}_0^d$ als Monoid auf, so ist die Betragsfunktion $|\cdot| : \mathbb{N}_0^d \to \mathbb{N}_0$ aus Definition 2.1 ein *Monoidhomomorphismus*, d. h. es gilt $|\alpha + \beta| = |\alpha| + |\beta|$ für $\alpha, \beta \in \mathbb{N}_0^d$ und $|0| = 0$.

Mit Hilfe der Multiindizes können alle *Monome* über $\mathbb{R}$ in den Variablen $x_1, \ldots, x_d$ durch $f_\alpha x^\alpha$ mit $\alpha \in \mathbb{N}_0^d$ und $f_\alpha \in \mathbb{R}$ dargestellt werden. Bei verschiedenen Autoren wie beispielsweise Cox, Litte und O'Shea in [CLO07] findet man für diesen Ausdruck auch die Bezeichnung *Term*. An dieser Stelle ist die Terminologie nicht eindeutig. In [KR00] haben Kreuzer und Robbiano eine Liste der unterschiedlichen Verwendung dieser Bezeichnungen zusammengestellt und den Variantenreichtum der verschiedenen Autoren aufgezeigt. Wir werden in dieser Arbeit [KR00] folgen und verwenden die Bezeichnungen *Term* für x^α und *Monom* für $f_\alpha x^\alpha$.

Analog zum univariaten Fall sind auch die multivariaten Polynome über den reellen Zahlen nichts weiter als eine endliche Summe von Termen, die durch Koeffizienten $f_\alpha \in \mathbb{R}$ gewichtet werden. Formal entspricht dies der folgenden Definition.

Definition 2.2. *Mit $\Pi_d = \mathbb{R}[x_1, \ldots, x_d]$ bezeichnet man den Ring aller* Polynome

$$f(x) = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha, \quad f_\alpha \in \mathbb{R}, \quad \#\{\alpha \in \mathbb{N}_0^d : f_\alpha \neq 0\} < \infty,$$

in den Variablen $x_1, \ldots, x_d$.

Identifizieren wir ein Polynom $f = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha \in \Pi_d$, $f_\alpha \in \mathbb{R}$, mit seinem *Koeffizientenvektor* $\left[f_\alpha : \alpha \in \mathbb{N}_0^d\right]$, so lässt sich der Ring Π_d auch als Vektorraum über $\mathbb{R}$ auffassen. Die Terme x^α, $\alpha \in \mathbb{N}_0^d$, bilden dabei eine Basis von Π_d und die Koeffizientenvektoren sind die Darstellungen bzgl. dieser Basis. Natürlich muss zuvor eine Anordnung der Koeffizienten im Koeffizientenvektor in Abhängigkeit von α festgelegt werden. Dieses Problem wird in Abschnitt 2.2 im Detail untersucht. Im Folgenden wird auch in der Notation nicht mehr zwischen Polynomen und ihren Koeffizien-

tenvektoren unterschieden. Dabei gelte die Konvention, dass Koeffizientenvektoren stets als Zeilenvektoren dargestellt werden.

Die Verwendung von Multiindizes ermöglicht es zudem, den Gradbegriff von den univariaten auf die multivariaten Polynome zu übertragen. Im univariaten Fall legt der Term mit der höchsten Potenz den Grad des Polynoms fest – hier ist es entsprechend der Term mit dem betragsmäßig größten Multiindex.

Definition 2.3. *Sei $f \in \Pi_d$ ein Polynom wie oben. Dann bezeichnet*

$$\deg(f) = \max\{|\alpha| : f_\alpha \neq 0\} \in \mathbb{N}_0$$

den Totalgrad *von f.*

Hier fällt der erste deutliche Unterschied zu den univariaten Polynomen auf: Es kann mehrere Terme geben, deren zugehörige Multiindizes den gleichen Betrag haben. Sind in einem Polynom nur Terme mit betragsgleichen Multiindizes vorhanden, so nennt man dieses *homogen*.

Definition 2.4. *Der Vektorraum aller* homogenen Polynome *in $d \in \mathbb{N}$ Variablen vom Grad $k \in \mathbb{N}_0$ ist definiert als*

$$\Pi^0_{k,d} := \left\{ f(x) = \sum_{|\alpha|=k} f_\alpha x^\alpha : f_\alpha \in \mathbb{R} \right\}.$$

Damit ergibt sich die Menge aller homogenen Polynome in d Variablen als $\Pi^0_d := \bigcup_{j\in\mathbb{N}_0} \Pi^0_{j,d}$ und wir können den Vektorraum aller Polynome mit Totalgrad höchstens k darstellen als

$$\Pi_{k,d} := \bigoplus_{j=0}^{k} \Pi^0_{j,d} = \left\{ \sum_{|\alpha|\leq k} f_\alpha x^\alpha : f_\alpha \in \mathbb{R} \right\}. \tag{2.1}$$

Die Notation $\oplus$ in (2.1) beschreibt eine innere *direkte Summe* (vgl. [Fis05, 1.6]). Dies verdeutlicht, dass für zwei homogene Räume $\Pi^0_{j,d}$ und $\Pi^0_{k,d}$, $j \neq k$, stets

$$\Pi^0_{j,d} \cap \Pi^0_{k,d} = \{0\}$$

gilt. Umgekehrt kann man die direkte Summe in (2.1) auch als *direkte Zerlegung* des Raums $\Pi_{k,d}$ auffassen, sodass wir zu jedem Polynom $f \in \Pi_{k,d}$ eine *eindeutige* Darstellung durch homogene Polynome $f^{(j)} \in \Pi^0_{j,d}$, $j = 0, \dots, \deg(f)$, erhalten. Das homogene Polynom $f^{(\deg(f))}$, also das Polynom größten Grades in der direkten Zerlegung, hat eine besondere Bedeutung. Wir bezeichnen dieses *eindeutige* Polynom als *Leitform* von f. Die Leitform übernimmt im multivariaten Fall die Rolle des von den univariaten Polynomen bekannten *Leitkoeffizienten*. Formal führt dies zu folgender Definition:

Definition 2.5. *Sei $f \in \Pi_d$ ein Polynom. Dann definiert die Abbildung $\Lambda : \Pi_d \to \Pi^0_d$ die* Leitform *von f als das homogene Polynom*

$$\Lambda(f)(x) := \sum_{|\alpha|=\deg(f)} f_\alpha x^\alpha, \quad \textit{für} \quad f(x) = \sum_{|\alpha|\leq\deg(f)} f_\alpha x^\alpha.$$

Betrachtet man den Polynomring Π_d als Vektorraum über $\mathbb{R}$, so lässt sich ein *Skalarprodukt* zweier Polynome definieren. Für diese Arbeit ist das im Folgenden eingeführte *monomiale Skalarprodukt* von zentraler Bedeutung. Auf ein weiteres Skalarprodukt wird in (2.5) hingewiesen.

Definition 2.6. *Seien zwei Polynome $f, g \in \Pi_d$ gegeben durch $f(x) = \sum_{\alpha\in\mathbb{N}^d_0} f_\alpha x^\alpha$, $g(x) = \sum_{\alpha\in\mathbb{N}^d_0} g_\alpha x^\alpha$. Dann bezeichnen wir die Abbildung $(\cdot,\cdot) : \Pi_d \times \Pi_d \to \mathbb{R}$ mit*

$$(f, g) := \sum_{\alpha\in\mathbb{N}^d_0} f_\alpha g_\alpha \tag{2.2}$$

als monomiales Skalarprodukt *auf Π_d.*

Fassen wir f und g wie oben beschrieben als Koeffizientenvektoren der Länge n auf, so entspricht das monomiale Skalarprodukt genau dem *Standardskalarprodukt* auf $\mathbb{R}^n$. Bezüglich dieses Skalarprodukts können wir nun homogene Teilräume von Π_d in Abhängigkeit einer Menge von Polynomen $F \subset \Pi_d$ und ihre orthogonalen Komplemente definieren.

Definition 2.7. *Sei $(\cdot,\cdot) : \Pi_d \times \Pi_d \to \mathbb{R}$ das Skalarprodukt aus Definition 2.6,*

$F \subset \Pi_d$ *und* $k \in \mathbb{N}_0$. *Wir definieren durch*

$$\mathcal{V}^0_{k,d}(F) := \left\{ \sum_{f \in F \cap \Pi_{k,d}} g_f \Lambda(f) : g_f \in \Pi^0_{k-\deg(f)} \right\} \subseteq \Pi^0_{k,d} \tag{2.3}$$

den von F *bzw.* $\Lambda(F)$ *erzeugten* homogenen Teilraum *von* $\Pi^0_{k,d}$ *und das* orthogonale Komplement *als*

$$\mathcal{W}^0_{k,d}(F) := \Pi^0_{k,d} \ominus \mathcal{V}^0_{k,d}(F) = \left\{ g \in \Pi^0_{k,d} : \left(g, \mathcal{V}^0_{k,d}(F) \right) = 0 \right\}. \tag{2.4}$$

Mit Hilfe dieser Darstellung lassen sich auch die nichthomogenen Räume $\Pi_{k,d}$ *und* Π_d *bzgl.* $F \subset \Pi_d$ *orthogonal zerlegen. Die Bezeichnungen werden dabei analog gewählt.*

$$\Pi_{k,d} = \bigoplus_{j=0}^{k} \mathcal{V}^0_{j,d}(F) \oplus \bigoplus_{j=0}^{k} \mathcal{W}^0_{j,d} =: \mathcal{V}_{k,d}(F) \oplus \mathcal{W}_{k,d}(F),$$
$$\Pi_d = \bigoplus_{j \in \mathbb{N}_0} \mathcal{V}^0_{j,d}(F) \oplus \bigoplus_{j \in \mathbb{N}_0} \mathcal{W}^0_{j,d} =: \mathcal{V}_d(F) \oplus \mathcal{W}_d(F).$$

Die Definition der Räume $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$ durch (2.3) bzw. (2.4) bietet jedoch keine effiziente Darstellung. Da es sich um lineare Teilräume eines Vektorraums handelt, sucht man natürlich nach einer geeigneten *Basis*. Der folgende Satz zeigt, wie sich ein Erzeugendensystem von $\mathcal{V}^0_{k,d}(F)$ bestimmen lässt, das für endliche Mengen $F \subset \Pi_d$ ebenfalls endlich ist:

Satz 2.8. *Sei* $F \subset \Pi_d$. *Es gilt*

$$\mathcal{V}^0_{k,d}(F) = \operatorname{span}\left\{ x^\alpha \cdot \Lambda(f)(x) : f \in F, \deg(f) \le k, \alpha \in \mathbb{N}_0^d, |\alpha| = k - \deg(f) \right\}.$$

Beweis. Betrachten wir die Definition des homogenen Teilraums $\mathcal{V}^0_{k,d}(F)$ in Definition 2.7, so lassen sich die dort verwendeten Polynome $g_f \in \Pi^0_{k-\deg(f),d}$ durch

$$g_f(x) = \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| = k - \deg(f)}} (g_f)_\alpha x^\alpha, \quad (g_f)_\alpha \in \mathbb{R},$$

darstellen. Setzen wir dies in (2.3) ein, so erhalten wir

$$\begin{aligned}\mathcal{V}_{k,d}^0(F) &= \left\{ \sum_{f\in F\cap\Pi_{k,d}} \left(\sum_{\substack{\alpha\in\mathbb{N}_0^d \\ |\alpha|=k-\deg(f)}} (g_f)_\alpha x^\alpha \right) \Lambda(f) \right\} \\ &= \left\{ \sum_{f\in F\cap\Pi_{k,d}} \sum_{\substack{\alpha\in\mathbb{N}_0^d \\ |\alpha|=k-\deg(f)}} (g_f)_\alpha \left(x^\alpha \Lambda(f)\right) \right\} \\ &= \operatorname{span}\left\{ x^\alpha \cdot \Lambda(f)(x) : f\in F, \deg(f)\le k, \alpha\in\mathbb{N}_0^d, |\alpha| = k-\deg(f) \right\},\end{aligned}$$

da die Polynome in $\mathcal{V}_{k,d}^0(F)$ linear von $x^\alpha \cdot \Lambda(f)(x)$ abhängen. □

Das folgende Beispiel zeigt die Berechnung einiger Erzeugendensysteme nach Satz 2.8:

Beispiel 2.9. *Sei $F = \{x_1 + x_2, x_3^2\} \subset \Pi_3$, dann erhalten wir für $k = 0, 1, 2, 3$ folgende Erzeugendensysteme:*

$$\begin{aligned}\mathcal{V}_{0,3}^0(F) &= \operatorname{span}\{\emptyset\} = \{0\},\\ \mathcal{V}_{1,3}^0(F) &= \operatorname{span}\{x_1 + x_2\},\\ \mathcal{V}_{2,3}^0(F) &= \operatorname{span}\{x_1^2 + x_1x_2, x_1x_2 + x_2^2, x_1x_3 + x_2x_3, x_3^2\},\\ \mathcal{V}_{3,3}^0(F) &= \operatorname{span}\{x_1^3 + x_1^2x_2, x_1^2x_2 + x_1x_2^2, x_1x_2^2 + x_2^3, x_1^2x_3 + x_1x_2x_3, x_1x_2x_3 + x_2^2x_3,\\ &\qquad x_1x_3^2 + x_2x_3^2, x_1x_3^2, x_2x_3^2, x_3^3\}.\end{aligned}$$

Für $k > \max_{f\in F}(\deg(f))$ kann dabei ein bereits bekanntes Erzeugendensystem von $\mathcal{V}_{k,d}^0(F)$ als Zwischenergebnis genutzt werden: Man muss lediglich jedes einzelne Element mit jedem Term x_j, $j = 1, \ldots, d$, multiplizieren und erhält ein Erzeugendensystem von $\mathcal{V}_{k+1,d}^0(F)$. In [PS07] haben Peña und Sauer eine ähnliche Methode zur Bestimmung einer Basis von $\mathcal{V}_{k+1,d}^0(F)$ aus einer bereits bekannten Basis von $\mathcal{V}_{k,d}^0(F)$ vorgestellt. Dieser Methode liegt jedoch ein von (2.2) verschiedenes *Skalarprodukt* $(\cdot,\cdot)_D$ zugrunde, das durch

$$(f,g)_D := (f(D)g)(0) = \sum_{\alpha\in\mathbb{N}_0^d} \alpha! f_\alpha g_\alpha, \tag{2.5}$$

definiert ist. Dabei beschreibt $f(D) = f(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_d})$ einen linearen, von f induzierten, partiellen Differentialoperator mit konstanten Koeffizienten. Das folgende Beispiel verdeutlicht die Anwendung dieses Operators:

Beispiel 2.10. *Seien $f, g \in \Pi_{2,2}$ gegeben durch $f(x_1, x_2) = x_1x_2 + 2x_2^2 - x_1$ und $g(x_1, x_2) = 2x_1^2 - 3x_2^2 + x_1 + 4x_2$, dann gilt*

$$f(D) = f\left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}\right) = \frac{\partial^2}{\partial x_1 \partial x_2} + 2\frac{\partial^2}{\partial x_2^2} - \frac{\partial}{\partial x_1},$$

sowie

$$\begin{aligned}(f(D)g)(x_1, x_2) &= \frac{\partial^2}{\partial x_1 \partial x_2} g(x_1, x_2) + 2\frac{\partial^2}{\partial x_2^2} g(x_1, x_2) - \frac{\partial}{\partial x_1} g(x_1, x_2) \\ &= 0 + 2 \cdot (-6) - 1 = -13\end{aligned}$$

und damit $(f, g)_D = (f(D)g)(0) = -13$. Für das monomiale Skalarprodukt erhalten wir hingegen $(f, g) = -7$.

Das Skalarprodukt $(f, g)_D$ wurde auch von Ron und de Boor im Zusammenhang mit multivariater Interpolation verwendet, vgl. [BR90]. Wir werden in dieser Arbeit jedoch stets auf das monomiale Skalarprodukt aus Definition 2.6 zurückgreifen, da die Beziehung zum Standardskalarprodukt der Koeffizientenvektoren die Anwendung vieler bekannter Resultate aus der Linearen Algebra ermöglicht. Dies wird insbesondere bei der Berechnung von Basen für das orthogonale Komplement $\mathcal{W}_{k,d}^0(F)$ in Abschnitt 2.4.4 ausgenutzt.

2.2. Speicherung von Koeffizientenvektoren

In der Theorie bieten Multiindizes eine elegante Möglichkeit zur Darstellung multivariater Polynome. Sollen die Polynome jedoch in Form von *Koeffizientenvektoren* gespeichert werden, so versagt dieses Konzept, da den Multiindizes eine kanonische lineare Ordnung fehlt. Im univariaten Fall ist eine solche Ordnung trivialerweise durch die Relation $\leq$ gegeben. Überträgt man diese Relation komponentenweise auf die Multiindizes, so ergibt sich die folgende Definition:

Definition 2.11. *Seien $\alpha, \beta \in \mathbb{N}_0^d$, dann ist $\alpha \leq \beta$ genau dann, wenn $\alpha_j \leq \beta_j$ für alle $j = 1, \ldots, d$ gilt.*

Diese Relation ist jedoch ungeeignet, da es sich nur um eine *Halbordnung* handelt – d. h. es gibt Multiindizes, die bezüglich der Relation nicht vergleichbar sind.

Beispiel 2.12. *Seien $\alpha = (1, 0, 0)$ und $\beta = (0, 1, 0)$ Multiindizes, dann gilt weder $\alpha \leq \beta$ noch $\beta \leq \alpha$, da $\beta_1 \leq \alpha_1$ und $\alpha_2 \leq \beta_2$.*

Um die Multiindizes linear anzuordnen, ist stattdessen eine *totale Ordnung* notwendig. In [CLO07] werden die Anforderungen an eine solche *Termordnung* wie folgt festgelegt:

Definition 2.13. *Eine Ordnung $\prec$ auf $\mathbb{N}_0^d$ heißt* Termordnung*, wenn sie*

1. *eine* totale Ordnung *ist, d. h.*

$$\gamma, \gamma' \in \mathbb{N}_0^d,\ \gamma \neq \gamma' \quad \Longrightarrow \quad \gamma \prec \gamma' \text{ oder } \gamma' \prec \gamma.$$

2. *kompatibel mit der Halbgruppenoperation des Monoids $\mathbb{N}_0^d$ ist, also*

$$\gamma, \gamma' \in \mathbb{N}_0^d,\ \gamma \prec \gamma' \quad \Longrightarrow \quad (\gamma + \eta) \prec (\gamma' + \eta), \quad \eta \in \mathbb{N}_0^d.$$

3. *eine* Wohlordnung *ist, d. h. jede strikt absteigende Folge $\gamma^{(1)} \succ \gamma^{(2)} \succ \cdots$, $\gamma^{(j)} \in \mathbb{N}_0^d$, endlich ist.*

Auch hier findet man gelegentlich die Bezeichnung *Monomordnung*. Da die Koeffizienten eines Monoms für die Ordnung jedoch irrelevant sind, werden wir stets von *Termordnungen* sprechen. Bekannte Termordnungen für Multiindizes $\alpha, \beta \in \mathbb{N}_0^d$ sind etwa die *lexikographische* (lex), die *graduiert-lexikographische* (glex) oder die *umgekehrt-lexikographische* (rlex) Ordnung:

$$\begin{aligned}
\alpha \prec_{lex} \beta \quad &:\Longleftrightarrow \quad \alpha_j = \beta_j,\ \alpha_k < \beta_k,\ j = 1, \ldots, k-1,\ k \leq d, \\
\alpha \prec_{glex} \beta \quad &:\Longleftrightarrow \quad |\alpha| < |\beta| \quad \text{oder} \quad |\alpha| = |\beta|,\ \alpha \prec_{lex} \beta, \\
\alpha \prec_{rlex} \beta \quad &:\Longleftrightarrow \quad \alpha_j = \beta_j,\ \alpha_k < \beta_k,\ j = k+1, \ldots, d,\ 1 \leq k.
\end{aligned}$$

Die Wahl der Termordnung hängt entscheidend von der Anwendung ab. In dieser Arbeit werden wir stets die graduiert-lexikographische Ordnung verwenden. Dies hat vor allem folgende Gründe:

1. Die graduiert-lexikographische Ordnung ermöglicht eine einfache Addition von Polynomen verschiedenen Grades, vgl. Abschnitt 2.4.1. Diese Operation ist essenziell für viele der im Folgenden untersuchten Algorithmen.

2. Ein multivariates Polynom, dessen Koeffizienten in glex-Ordnung vorliegen, lässt sich schneller und bzgl. des Rückwärtsfehlers stabiler auswerten als beispielsweise eine Darstellung in rlex-Ordnung. Dies wurde vom Autor in [CS14] gezeigt. Details zu diesem Auswertungsverfahren sowie Rechenaufwand und Stabilität werden in Abschnitt 2.3 besprochen.

Durch die Festlegung einer Termordnung können nun alle Multiindizes $\alpha \in \mathbb{N}_0^d$, $|\alpha| \leq k$, in geordneter Weise angegeben werden. Als Darstellungsform verwenden wir eine Matrix, die zeilenweise alle relevanten Multiindizes in aufsteigender Reihenfolge bzgl. der graduiert-lexikographischen Termordnung enthält. Dieses Konzept findet sich ebenso bei de Boor, vgl. [Boo00].

Definition 2.14. *Die* Termmatrix $\boldsymbol{T}_{k,d}$ *besteht aus allen Multiindizes* $\alpha \in \mathbb{N}_0^d$ *mit* $|\alpha| \leq k$ *in graduiert-lexikographischer Ordnung. Mit anderen Worten: Für je zwei Zeilen* j *und* $j+1$ *einer Termmatrix gilt* $(\boldsymbol{T}_{k,d})_j \prec_{glex} (\boldsymbol{T}_{k,d})_{j+1}$.

Beispiel 2.15.

$$\boldsymbol{T}_{0,3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{T}_{1,3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{T}_{2,3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}.$$

Durch die graduiert-lexikographische Ordnung zerfällt die Matrix $\boldsymbol{T}_{k,d}$ in $k+1$ Blöcke, in denen die Multiindizes jeweils betragsgleich und lexikographisch geordnet sind. Außerdem erfüllen die Termmatrizen die Rekursionsformel

$$\boldsymbol{T}_{k,d} = \left[{\boldsymbol{T}^0_{0,d}}^T, \ldots, {\boldsymbol{T}^0_{k,d}}^T \right]^T = \left[{\boldsymbol{T}_{k-1,d}}^T, {\boldsymbol{T}^0_{k,d}}^T \right]^T, \tag{2.6}$$

wobei $\boldsymbol{T}^0_{j,d}$ die Matrix aller Multiindizes $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = j$, $j \leq k$, bezeichnet. Da die Terme mit den Multiindizes aus $\boldsymbol{T}^0_{k,d}$ eine Basis des homogenen Raums $\Pi^0_{k,d}$ bilden, nennen wir $\boldsymbol{T}^0_{k,d}$ eine *homogene Termmatrix*.

Beispiel 2.16.

$$\boldsymbol{T}^0_{0,3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{T}^0_{1,3} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{T}^0_{2,3} = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}.$$

Per definitionem bestehen die Matrizen $\boldsymbol{T}_{k,d}$ bzw. $\boldsymbol{T}^0_{k,d}$ aus d Spalten. Um die Anzahl der Zeilen anzugeben, können wir das *Urnenmodell* aus der Kombinatorik verwenden: Man erhält einen Multiindex $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k$, indem man mit $\alpha = (0, \ldots, 0)$ startet, k-mal eine Komponente j aus $\{1, \ldots, d\}$ mit Zurücklegen zieht und deren Wert um 1 erhöht. Offensichtlich kommt es dabei nicht auf die Reihenfolge an und das *Urnenmodell* liefert $\binom{d+k-1}{k}$ verschiedene Möglichkeiten solcher Ziehungen, vgl. [Aig06, 1.2].

Lemma 2.17. *Die homogene Termmatrix $\boldsymbol{T}^0_{k,d}$ besteht aus $\binom{d+k-1}{k}$ Zeilen. Die Zeilenanzahl der Termmatrix $\boldsymbol{T}_{k,d}$ entspricht daher $\binom{d+k}{k}$.*

Die Aussage für $\boldsymbol{T}_{k,d}$ folgt dabei aus der Blockdarstellung in (2.6) und der bekannten Identität für die Summe verschobener Binomialkoeffizienten:

$$\sum_{j=0}^{k} \binom{d+j-1}{j} = \binom{d+k}{k}.$$

Im Folgenden bezeichnen wir die Anzahl der Zeilen einer Termmatrix auch als $\#\boldsymbol{T}_{k,d}$, bzw. im homogenen Fall als $\#\boldsymbol{T}^0_{k,d}$. Da viele dieser Werte in den hier vorgestellten Verfahren benötigt werden, ist es sinnvoll, die folgende Matrix bereitzustellen:

Definition 2.18. *Die Matrix*

$$\boldsymbol{P}_{k,d} := \left[\binom{i+j-1}{j} : \begin{array}{c} i = 1,\dots,d \\ j = 0,\dots,k \end{array} \right] = \left[\binom{i+j}{j} : \begin{array}{c} i = 0,\dots,d-1 \\ j = 0,\dots,k \end{array} \right] \in \mathbb{N}^{d\times(k+1)}$$

bezeichnet man als Pascal-Matrix.

In [BP92] sind wichtige Eigenschaften dieser und ähnlicher Matrizen zu finden. Für diese Arbeit sind jedoch hauptsächlich die Einträge der Pascal-Matrix relevant:

$$\boldsymbol{P}_{k,d} = \begin{bmatrix} 1 & 1 & 1 & \cdots & \binom{j}{j} & \cdots & 1 \\ 1 & 2 & 3 & \cdots & \binom{j+1}{j} & \cdots & k+1 \\ 1 & 3 & 6 & & & & \\ \vdots & \vdots & & \ddots & & & \\ \binom{i}{0} & \binom{i}{1} & & & \binom{i+j}{j} & & \vdots \\ \vdots & \vdots & & & & \ddots & \\ 1 & d & & & \cdots & & \binom{d+k-1}{k} \end{bmatrix}.$$

Damit reduziert sich die Bestimmung der Werte $\#\boldsymbol{T}_{k,d}$ bzw. $\#\boldsymbol{T}^0_{k,d}$ auf ein „Nachschlagen“ der passenden Werte in der Pascal-Matrix. In vielen Systemen wie MATLAB bzw. OCTAVE oder MATHEMATICA ist diese Matrix schon vorimplementiert und kann direkt abgefragt werden.

Identifizieren wir nun die Multiindizes einer Termmatrix $\boldsymbol{T}_{k,d}$ mit den Indizes ihrer Zeilen, so erhalten wir eine Bijektion zwischen $\mathbb{N}_0^d$ und $\mathbb{N}$. Damit entspricht jedes Polynom $f \in \Pi_{k,d} \setminus \Pi_{k-1,d}$ genau einem Koeffizientenvektor $f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$. Die Eindeutigkeit dieser Zuordnung für ein festes $d \in \mathbb{N}$ rechtfertigt es, in der Notation nicht zwischen Polynom und Koeffizientenvektor zu unterscheiden. Dass die Eindeutigkeit verloren geht, falls die Anzahl der Variablen nicht angegeben ist, zeigt das folgende Beispiel:

Beispiel 2.19. *Seien die Polynome*

$$f(x_1, x_2, x_3) = -3x_1x_2 + x_3^2 + 0.5x_2 - 42, \quad g(x_1, x_2) = -3x_1^2x_2 + x_1x_2 + 0.5x_1 - 42$$

gegeben. Obwohl $f \neq g$ ist, haben beide den gleichen Koeffizientenvektor

$$[-42, 0, 0.5, 0, 1, 0, 0, 0, -3, 0] \in \mathbb{R}^{10}.$$

Natürlich lässt sich die Umrechnung von Multiindizes in Zeilenindizes der Termmatrix auch durchführen, ohne die Termmatrix selbst zu bestimmen. Da eine Termmatrix in homogene Termmatrizen zerfällt und die Länge aller homogenen Termmatrizen bekannt ist, reicht es aus, die Position eines Multiindex $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k$ in der homogenen Termmatrix $\boldsymbol{T}_{k,d}^0$ zu bestimmen. Dazu können wir die folgende Rekursionseigenschaft aus [Boo00] verwenden, die auch zur Konstruktion von homogenen Termmatrizen verwendet werden kann (vgl. Abschnitt A.5):

Lemma 2.20. *Sei $\boldsymbol{T}_{k,d}^0$ die Matrix der Multiindizes $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k$. Dann gilt die rekursive Beziehung*

$$\boldsymbol{T}_{k,d}^0 = \begin{bmatrix} 0 & \boldsymbol{T}_{k,d-1}^0 \\ 1 & \boldsymbol{T}_{k-1,d-1}^0 \\ \vdots & \vdots \\ j & \boldsymbol{T}_{k-j,d-1}^0 \\ \vdots & \vdots \\ k & \boldsymbol{T}_{0,d-1}^0 \end{bmatrix}, \quad k \in \mathbb{N}_0,\ d > 1,$$

und $\boldsymbol{T}_{k,1}^0 = k$, $k \in \mathbb{N}_0$.

Beweis. Im univariaten Fall, also $d = 1$, gibt es zum Grad $k \in \mathbb{N}_0$ nur den Term x^k und damit gilt $\boldsymbol{T}_{k,1}^0 = k$. Sei nun $d > 1$. Da die Zeilen der Matrix $\boldsymbol{T}_{k,d}^0$ in lexikographischer Ordnung sind, gilt für die Einträge der erste Spalte $(\boldsymbol{T}_{k,d}^0)_{i,1} \leq (\boldsymbol{T}_{k,d}^0)_{i+1,1}$. Beginnt nun eine Zeile mit $j \in \{0, \ldots, k\}$, so entspricht sie einem Multiindex $\alpha \in \mathbb{N}_0^d$ mit $\alpha_1 = j$. Mit $|\alpha| = k$ folgt $|(\alpha_2, \ldots, \alpha_d)| = k - j$, also muss die Zeile durch einen Multiindex $\widetilde{\alpha} \in \mathbb{N}_0^{d-1}$ mit $|\widetilde{\alpha}| = k - j$ ergänzt werden. Dies sind jedoch genau die Zeilen der Matrix $\boldsymbol{T}_{k-j,d-1}^0$. □

Der folgende Satz stammt von de Boor (vgl. [Boo00]) und beschreibt die Indexkonvertierung von Multiindizes in Zeilenindizes der Termmatrix. Die dabei verwendete Notation $\alpha_{(r:d)}$, $r \in \{1, \ldots, d\}$, für einen Multiindex $\alpha \in \mathbb{N}_0^d$ ist durch die MATLAB Syntax motiviert und beschreibt die letzten $d - r + 1$ Komponenten von α, also $(\alpha_r, \ldots, \alpha_d)$.

Satz 2.21. *Sei $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k$, dann hat α als Zeile der Matrix $\boldsymbol{T}_{k',d}$, $k' \geq k$, den Zeilenindex*

$$\sum_{r=1}^{d-1} \sum_{0 \leq i < \alpha_r} \binom{(d-r) + (|\alpha_{(r:d)}| - i) - 1}{|\alpha_{(r:d)}| - i}.$$

2.3. Auswertung von multivariaten Polynomen

In diesem Abschnitt wird eine effiziente und numerisch stabile Methode zur *Auswertung* multivariater Polynome vorgestellt, die eine Verallgemeinerung des univariaten *Horner-Schemas* darstellt und von de Boor in [Boo00] beschrieben wurde. In [CS14] wurde dieses Verfahren vom Autor bzgl. Rechenaufwand und numerischer Stabilität im Sinne des Rückwärtsfehlers untersucht und mit anderen Methoden zur Auswertung multivariater Polynome verglichen. Wir stellen hier die Konstruktion des Verfahrens von de Boor vor und fassen die Ergebnisse aus [CS14] zusammen.

Eine zentrale Voraussetzung für die Variante des Horner-Schemas von de Boor besteht darin, dass die Koeffizienten gemäß der im letzten Abschnitt beschriebenen *graduiert-lexikographischen* Ordnung vorliegen. Bevor wir jedoch den Algorithmus angeben, soll die Idee zunächst an einem Beispiel verdeutlicht werden:

Beispiel 2.22. *Sei ein Polynom $f \in \Pi_3$ mit $\deg(f) = 3$ gegeben durch*

$$\begin{aligned} f(x_1, x_2, x_3) = {} & x_1^3 + 4x_1^2x_2 - 3x_1^2x_3 + 4x_1x_2^2 - 4x_1x_2x_3 - 2x_2^3 + 3x_2^2x_3 + 3x_2x_3^2 - 4x_3^3 \\ & - 2x_1^2 - 4x_1x_2 - 2x_1x_3 - 3x_2^2 - 2x_2x_3 - 4x_3^2 + 2x_2 + x_3 - 1. \end{aligned}$$

Wir faktorisieren $f(x_1, x_2, x_3)$ nun stufenweise. Dabei müssen die folgenden Regeln beachtet werden:

1. *Es werden nur die Monome vom höchsten Grad faktorisiert.*

2. *Es wird jeweils nur der lexikographisch größte mögliche Term, also x_1 vor x_2 etc., vom Grad 1 abgespaltet.*

Damit erhalten wir in zwei Schritten

$$\begin{aligned}
&f(x_1, x_2, x_3) \\
&\quad = (-2 + x_1)x_1^2 + (-4 + 4x_1)x_1x_2 + (-2 - 3x_1)x_1x_3 + (-3 + 4x_1 - 2x_2)x_2^2 \\
&\qquad\quad + (-2 - 4x_1 + 3x_2)x_2x_3 + (-4 + 3x_2 - 4x_3)x_3^2 + 2x_2 + x_3 - 1 \\
&\quad = \Big((-2 + x_1)x_1\Big)x_1 + \Big(2 + (-4 + 4x_1)x_1 + (-3 + 4x_1 - 2x_2)x_2\Big)x_2 \\
&\qquad\quad + \Big(1 + (-2 - 3x_1)x_1 + (-2 - 4x_1 + 3x_2)x_2 + (-4 + 3x_2 - 4x_3)x_3\Big)x_3 - 1.
\end{aligned}$$

Um das Polynom an einer Stelle $\xi = (1, 2, 3) \in \mathbb{R}^3$ auszuwerten, setzen wir nun die Werte der Auswertungsstelle sukzessive von innen nach außen ein. Damit reduziert sich der Grad des Polynoms in jedem Schritt, bis nur noch ein konstantes Polynom übrig bleibt. Der Koeffizient dieses Polynoms entspricht dann dem Wert des Ausgangspolynoms am Auswertungspunkt.

$$\begin{aligned}
f^{(1)}(x_1, x_2, x_3) &= -1x_1^2 + 0x_1x_2 - 5x_1x_3 - 3x_2^2 + 0x_2x_3 - 10x_3^2 + 2x_2 + x_3 - 1, \\
&= \Big((-1x_1)x_1 + \Big(2 + 0x_1 - 3x_2\Big)x_2 + \Big(1 - 5x_1 + 0x_2 - 10x_3\Big)x_3 - 1, \\
f^{(2)}(x_1, x_2, x_3) &= (-1)x_1 + (-4)x_2(-34)x_3 - 1 \\
f^{(3)}(x_1, x_2, x_3) &= -112.
\end{aligned}$$

Eine strukturierte Darstellung dieser Berechnung erhält man durch das in Abbildung 2.1 angegebene Schema, in dem die Multiindizes der Terme und ihre Koeffizienten eingetragen sind. Leere Felder sind dabei gleichbedeutend mit dem Wert Null.

Fasst man die in Beispiel 2.22 beschriebenen Faktorisierungsregeln und die anschließende schrittweise Auswertung zusammen, so erhält man das in Algorithmus 2.23 beschriebene Verfahren, vgl. [Boo00] bzw. [CS14]. In dieser Formulierung ist das Verfahren jedoch nicht implementierbar, da die Koeffizienten des Polynoms über Multiindizes identifiziert werden. Daher kombiniert man Algorithmus 2.23 mit der graduiert-lexikographischen Ordnung, die in Abschnitt 2.2 beschrieben wurde, was

$\alpha \in \boldsymbol{T}_{3,3}$			$f_\alpha^{(0)}$	$f_\alpha^{(1)}$	$f_\alpha^{(2)}$	$f_\alpha^{(3)}$
0	0	0	-1	-1	-1	-112
0	0	1	1	1	-34	
0	1	0	2	2	-4	
1	0	0	0	0	-1	
0	0	2	-4	-10		
0	1	1	-2	0		
0	2	0	-3	-3		
1	0	1	-2	-5		
1	1	0	-4	0		
2	0	0	-2	-1		
0	0	3	-4			
0	1	2	3			
0	2	1	3			
0	3	0	-2			
1	0	2	0			
1	1	1	-4			
1	2	0	4			
2	0	1	-3			
2	1	0	4			
3	0	0	1			

Abbildung 2.1.: Auswertungsschema für multivariate Polynome. Das Vorgehen erfolgt blockweise von unten nach oben, wobei die farblich markierten Koeffizienten mit den Komponenten der Auswertungsstellen gewichtet (grün: $x_1 = 1$, blau: $x_2 = 2$, rot: $x_3 = 3$) und die Ergebnisse zu dem darüber liegenden Block addiert werden. Dabei muss der Koeffizientenvektor ggf. mit Nullen nach unten hin aufgefüllt werden.

Algorithmus 2.23 : Multivariate Polynomauswertung mittels Horner-Schema

Input : Polynom $f(x) = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha$, Auswertungsstelle $\xi = (\xi_1, \ldots, \xi_d) \in \mathbb{R}^d$
Output : $f_{(0,\ldots,0)} = f(\xi)$

for $j = \deg(f), \ldots, 1$ **do**
  for $i = 1, \ldots, d$ **do**
    foreach $\alpha \in \boldsymbol{T}^0_{j,d}$, $\alpha_1 = \cdots = \alpha_{i-1} = 0 \neq \alpha_i$ **do**
      $f_{\alpha - \varepsilon_i} \leftarrow f_{\alpha-\varepsilon_i} + \xi_i f_\alpha$, mit $\varepsilon_i = (\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0) \in \mathbb{N}_0^d$
      $f_\alpha \leftarrow 0$
    end
  end
end

Algorithmus 2.24 : *glex*-Version von Algorithmus 2.23

Input : $f_1, \ldots, f_M$, $M = \#\boldsymbol{T}_{k,d}$, $\xi = (\xi_1, \ldots, \xi_d) \in \mathbb{R}^d$
Output : $f_1 = f(\xi)$

for $j = \deg(f), \ldots, 1$ **do**
  for $i = 1, \ldots, d$ **do**
    for $k = 1, \ldots, \#\boldsymbol{T}^0_{j-1,d-i}$ **do**
      $f_{\#\boldsymbol{T}^0_{j-2,d+1}+k} \leftarrow f_{\#\boldsymbol{T}^0_{j-2,d+1}+k} + \xi_{d-i+1} f_{\#\boldsymbol{T}^0_{j-1,d+1}+\#\boldsymbol{T}^0_{j,i-1}+k}$
      $f_{\#\boldsymbol{T}^0_{j-1,d+1}+\#\boldsymbol{T}^0_{j,i-1}+k} \leftarrow 0$
    end
  end
end

letztlich zu Algorithmus 2.24 führt. Diese Vorgehensweise wird ebenfalls in [Boo00] vorgeschlagen. Zur Verifikation des Verfahrens sei auf [Boo00] und [CS14] verwiesen. Ebenso findet man dort die folgende Aussage über den Rechenaufwand von Algorithmus 2.24:

Satz 2.25. *Die Auswertung eines Polynoms $f \in \Pi_d$ mit $\deg(f) = k$ durch Algorithmus 2.24 benötigt $\sum_{j=1}^{k} \binom{d+j-1}{j}$ Multiplikationen und ebenso viele Additionen.*

In einer naiven Berechnung werden zur Auswertung eines Monoms vom Grad j bereits $j-1$ Multiplikationen für den Term und eine zusätzliche Multiplikation für den

	Additionen	Multiplikationen
Direkt	$\sum_{j=1}^{k} \binom{d+j-1}{j}$	$\sum_{j=1}^{k} j \cdot \binom{d+j-1}{j}$
de Boor	$\sum_{j=1}^{k} \binom{d+j-1}{j}$	$\sum_{j=1}^{k} \binom{d+j-1}{j}$
Peña/Sauer	$\sum_{j=1}^{k} \binom{d+j-1}{j} + \binom{d+k}{d-1}$	$\sum_{j=1}^{k} \binom{d+j-1}{j}$

Tabelle 2.1.: Gegenüberstellung verschiedener Verfahren zur Polynomauswertung bzgl. der benötigten Rechenoperationen.

Koeffizienten benötigt. Da es $\binom{d+j-1}{j}$ Terme vom Grad j gibt, sind $\sum_{j=1}^{k} j \cdot \binom{d+j-1}{j}$ Multiplikationen zur Auswertung eines Polynoms $f \in \Pi_d$ mit $\deg(f) = k$ erforderlich – also deutlich mehr als in Algorithmus 2.24. Die Anzahl der Additionen bleibt hingegen gleich, da $\sum_{j=0}^{k} \binom{d+j-1}{j}$ Monome aufsummiert werden müssen, was genau $\sum_{j=0}^{k} \binom{d+j-1}{j} - 1 = \sum_{j=1}^{k} \binom{d+j-1}{j}$ Additionen entspricht. Natürlich lässt sich das Horner-Schema auch mit anderen Termordnungen konstruieren. Ein Verfahren für die umgekehrt-lexikographische Termordnung wurde von Peña und Sauer in [PS00] vorgestellt und vom Autor in [CS14] hinsichtlich der benötigten Rechenoperationen untersucht. Ein Vergleich ist in Tabelle 2.1 dargestellt. Insgesamt lässt sich feststellen, dass die Variante von de Boor den geringsten Rechenaufwand erfordert. Dieses Resultat bestärkt die Wahl der graduiert-lexikographischen Termordnung in Abschnitt 2.2.

In [CS14] wurde zudem die numerische Stabilität des Verfahrens von de Boor untersucht. Die dort erzielten Resultate setzen einige Grundbegriffe der Fehleranalyse voraus, die im Folgenden zusammengefasst werden. Die Notation orientiert sich dabei weitgehend an [Hig02].

Definition 2.26 (Standardmodell der Gleitkommaarithmetik)**.** *Sei $a \in \mathbb{R}$, dann bezeichnet* $\mathrm{fl}(a)$ *den berechneten Wert in* Gleitkommaarithmetik*. Für die Berechnung gilt das Modell*

$$\mathrm{fl}(a \cdot b) =: a \odot b = (a \cdot b)(1 + \delta), \quad |\delta| \leq u, \quad \cdot = +, -, \times, /,$$

wobei u den Einheitsrundungsfehler *bezeichnet.*

In [Hig02, Lemma 3.1, S. 69] hat Higham gezeigt, dass für dieses Modell folgende

Schranken gelten:

Lemma 2.27. *Seien $|\delta_i| \leq u$ und $\rho_i = \pm 1$, $i = 1, \ldots, n$ und $nu < 1$, dann gilt*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1 + \theta_n, \quad \textit{mit} \quad |\theta_n| \leq \frac{nu}{1-nu} =: \gamma_n.$$

Damit können wir ein Resultat aus [CS14] angeben, das den *Rückwärtsfehler* bei der Polynomauswertung nach Algorithmus 2.24 abschätzt.

Satz 2.28. *Sei $f(x) = \sum_{\alpha \in \mathbb{N}_0^d} c_\alpha x^\alpha$, $\deg(f) = k > 0$, ein multivariates Polynom und $(2k + d - 1)u < 1$. Ist nun $\mathrm{fl}(f(x)) = \sum_{\alpha \in \mathbb{N}_0^d} \hat{c}_\alpha x^\alpha$ der von Algorithmus 2.24 berechnete Wert, dann gilt*

$$\frac{|c_\alpha - \hat{c}_\alpha|}{|c_\alpha|} \leq \begin{cases} \gamma_{2|\alpha|+d}, & |\alpha| < k, \\ \gamma_{2|\alpha|+d-\mu(\alpha)}, & |\alpha| = k, \end{cases} \tag{2.7}$$

mit $\mu(\alpha) := \min\{j : \alpha_j \neq 0\}$.

In [PS00] haben Peña und Sauer für die Variante des Horner-Schemas in umgekehrt-lexikographischer Termordnung gezeigt, dass der Rückwärtsfehler unter den Voraussetzungen von Satz 2.28 durch

$$\frac{|\hat{c}_\alpha - c_\alpha|}{|c_\alpha|} \leq \begin{cases} \gamma_{2|\alpha|+d}, & |\alpha| < k, \\ \gamma_{2|\alpha|+d-1}, & |\alpha| = k, \end{cases} \tag{2.8}$$

abgeschätzt werden kann. Da in (2.7) stets $\mu(\alpha) \geq 1$ gilt, erfüllt das Verfahren von de Boor ebenfalls die Schranke aus (2.8). Für alle Koeffizienten c_α mit $|\alpha| = k$, $\alpha_1 = 0$, erhalten wir wegen $\mu(\alpha) > 1$ sogar eine schärfere Abschätzung für den Rückwärtsfehler.

2.4. Rechnen mit multivariaten Polynomen

In diesem Abschnitt werden die Ringoperationen des Polynomrings Π_d auf den Vektorraum der Koeffizientenvektoren übertragen. Dies ermöglicht es uns, Probleme aus dem Bereich der multivariaten Polynome mit Methoden der numerischen Linearen Algebra zu lösen. Dabei lässt sich ausnutzen, dass Π_d nicht nur ein Ring, sondern auch ein *Vektorraum* über $\mathbb{R}$ ist. Die Darstellung von Polynomen als Koeffizientenvektoren ist damit insbesondere ein *Vektorraumisomorphismus* zwischen $\Pi_{k,d}$ und $\mathbb{R}^{\#\boldsymbol{T}_{k,d}}$. Somit entsprechen die Vektorraumoperationen auf Π_d – also die Addition von Polynomen und die Multiplikation von Polynomen mit Skalaren – genau den Vektorraumoperationen des Standardvektorraums.

Die Polynommultiplikation ist hingegen keine Vektorraumoperation. Allerdings lässt sie sich durch die Konstruktion spezieller Matrizen auf eine Matrix-Vektor Multiplikation zurückführen. Dieser Ansatz wurde auch von Batselier bei der Entwicklung des *Polynomial Numerical Linear Algebra Frameworks* (PLNA) in [Bat13] verfolgt. Während die dort präsentierten Verfahren in Verbindung mit Gröbnerbasen verwendet werden, sind unsere Algorithmen bereits auf H-Basen ausgerichtet. Dies zeigt sich insbesondere bei der Polynomdivision. Dabei sind Methoden zur Bestimmung von Basen der homogenen Räume $\mathcal{V}^0_{k,d}(F)$ bzw. $\mathcal{W}^0_{k,d}(F)$, $F \subset \Pi_d$, die ebenfalls in diesem Abschnitt entwickelt werden, von besonderer Bedeutung.

2.4.1. Addition/Subtraktion von Polynomen

Auf dem Ring Π_d ist die Summe zweier Polynome $f, g \in \Pi_d$ mit $f(x) = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha$, bzw. $g(x) = \sum_{\alpha \in \mathbb{N}_0^d} g_\alpha x^\alpha$ definiert als das multivariate Polynom

$$(f+g)(x) = \sum_{\alpha \in \mathbb{N}_0^d} (f_\alpha + g_\alpha) x^\alpha.$$

Gilt $\deg(f) = \deg(g) = k$, so ergibt sich der Koeffizientenvektor von $(f+g)$ als Summe der Koeffizientenvektoren $f, g \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$, da diese in Länge und Ordnung der Koeffizienten übereinstimmen.

Sei nun $k_1 = \deg(f) < \deg(g) = k_2$, dann muss der Koeffizientenvektor von f um die Koeffizienten $f_\alpha = 0$ für $\deg(f) < |\alpha| \leq \deg(g)$ ergänzt werden. Algebraisch entspricht dies der Einbettung des Raumes $\Pi_{k_1,d}$ in den Raum $\Pi_{k_2,d}$. Da die Koeffizientenvektoren in graduiert-lexikographischer Ordnung vorliegen, ist dazu lediglich der Vektor $f \in \mathbb{R}^{\#\boldsymbol{T}_{k_1,d}}$ am Ende mit Nullen aufzufüllen:

$$(f + g) = [f, 0] + g \in \mathbb{R}^{\#\boldsymbol{T}_{k_2,d}}.$$

Man beachte, dass bei der Addition der Fall $\deg(f + g) < \max(\deg(f), \deg(g)) =: k$ auftreten kann. Daher sollte noch überprüft werden, ob der Koeffizientenvektor der Summe die Bedingung $(f + g)_j = 0$ für $\#\boldsymbol{T}_{i,d} < j \leq \#\boldsymbol{T}_{k,d}$, $i < k$, erfüllt. Ist dies der Fall, so werden die entsprechenden Nullen am Ende des Koeffizientenvektors abgeschnitten, sodass die Beziehung zwischen dem Grad des Polynoms und der Länge des Koeffizientenvektors wieder hergestellt wird.

2.4.2. Multiplikation von Polynomen mit Termen

Die einfachste Form der Multiplikation auf dem Ring Π_d ist sicherlich die Multiplikation mit konstanten Polynomen $g(x) = g_0 x^0$. Für die Multiplikation eines Polynoms $f \in \Pi_d$, $\deg(f) = k$, mit einem konstanten Polynom g ist lediglich das Produkt des Koeffizientenvektors $f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$ mit dem Skalar $g_0 \in \mathbb{R}$ zu berechnen. Diese Multiplikation ist bereits durch die entsprechende Vektorraumoperation definiert. Offensichtlich werden dabei nur die Werte des Koeffizientenvektors verändert, die Länge bleibt hingegen gleich.

Eine ähnlich einfache Operation, die alle Koeffizienten im Koeffizientenvektor verschiebt, aber deren Werte nicht verändert, ist die Multiplikation mit einem Term $g(x) = x^\beta$, $\beta \in \mathbb{N}_0^d$. Es gilt dabei

$$(f \cdot g)(x) = f(x) \cdot g(x) = x^\beta \cdot \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha \left(x^{\alpha+\beta}\right).$$

Daraus lassen sich nun die Koeffizienten des Polynoms $(f \cdot g)(x)$ unter Verwendung

Algorithmus 2.30 : Multiplikation eines multivariaten Polynoms mit einem Term

Input : $f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$, $g(x) = x^\beta$, $\beta \in \mathbb{N}_0^d$
Output : $(f \cdot g) \in \mathbb{R}^{\#\boldsymbol{T}_{k+|\beta|,d}}$
$(f \cdot g) \leftarrow (0, \ldots, 0) \in \mathbb{R}^{\#\boldsymbol{T}_{k+|\beta|,d}}$.
$I \leftarrow \{i : \beta \leq (\boldsymbol{T}_{k+|\beta|,d})_i\}$.
for $j = 1, \ldots, \#\boldsymbol{T}_{k,d}$ **do**
  $(f \cdot g)_{I(j)} \leftarrow f_j$.
end

der Halbordnung aus Definition 2.11 ablesen:

$$(f \cdot g)_\alpha = \begin{cases} f_{\alpha-\beta} & \text{falls } \beta \leq \alpha \\ 0 & \text{sonst} \end{cases}, \qquad \alpha \in \mathbb{N}_0^d.$$

Dieses Konzept können wir auch auf die Termmatrizen übertragen. Dazu fassen wir zunächst die Matrizen $\boldsymbol{T}_{k,d}^0$ als geordnete Mengen von Zeilenvektoren auf und erhalten folgendes Resultat:

Lemma 2.29. *Sei $\beta \in \mathbb{N}_0^d$, dann gilt*

$$\beta + \boldsymbol{T}_{k,d}^0 := \left\{\alpha + \beta : \alpha \in \mathbb{N}_0^d, |\alpha| = k\right\} = \left\{\alpha \in \mathbb{N}_0^d : |\alpha| = |\beta| + k, \beta \leq \alpha\right\} \subseteq \boldsymbol{T}_{k+|\beta|,d}^0.$$

Beweis. Die letzte Inklusion ist klar, da die Matrix $\boldsymbol{T}_{k+|\beta|,d}^0$ alle Multiindizes vom Betrag $k + |\beta|$ enthält. Sei $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k$. Da die Betragsfunktion ein *Monoidhomomorphismus* ist (vgl. Abschnitt 2.1), gilt $|\alpha + \beta| = |\alpha| + |\beta| = k + |\beta|$. Weiterhin ist $\beta \leq (\alpha + \beta)$, da $\beta_j \leq (\alpha_j + \beta_j)$ für $j = 1, \ldots, d$. Sei umgekehrt $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = k + |\beta|$ und $\beta \leq \alpha$, dann gibt es ein $\overline{\alpha} \in \mathbb{N}_0^d$ mit $\beta + \overline{\alpha} = \alpha$. Wie oben gilt $k + |\beta| = |\alpha| = |\overline{\alpha} + \beta| = |\overline{\alpha}| + |\beta|$, also ist $|\overline{\alpha}| = k$. □

Da die Zeilen in $\boldsymbol{T}_{k,d}^0$ bezüglich einer Termordnung sortiert sind, gilt

$$\alpha \prec_{glex} \gamma \quad \Longrightarrow \quad (\alpha + \beta) \prec_{glex} (\gamma + \beta), \quad \alpha, \beta, \gamma \in \mathbb{N}_0^d.$$

Damit bleibt die Ordnung beim Übergang von $\boldsymbol{T}_{k,d}^0$ zu $\beta + \boldsymbol{T}_{k,d}^0$ erhalten. Aufgrund der Konstruktion der Termmatrizen in (2.6) überträgt sich Lemma 2.29 blockweise

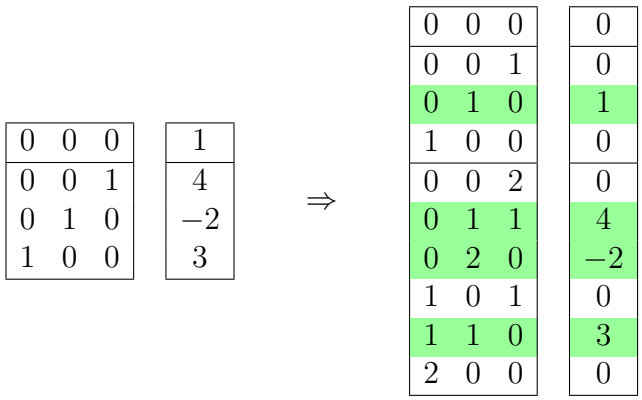


Abbildung 2.2.: Schematische Darstellung der Multiplikation des Polynoms $f(x_1, x_2, x_3) = 3x_1 - 2x_2 + 4x_3 + 1$ mit dem Term $g(x_1, x_2, x_3) = x_2$. Links ist der Koeffizientenvektor von f samt zugehörigen Multiindizes der Termmatrix $\boldsymbol{T}_{1,3}$ dargestellt. Der Multiindex $\beta = (0, 1, 0)$ des Terms g erfordert auf der rechten Seite die Termmatrix $\boldsymbol{T}_{1+|\beta|,3} = \boldsymbol{T}_{2,3}$, in der alle Zeilen, die größer oder gleich β sind, grün eingefärbt wurden.

auf die Matrizen $\boldsymbol{T}_{k,d}$ und induziert eine Bijektion zwischen den Zeilen von $\boldsymbol{T}_{k,d}$ und den Zeilen $(\boldsymbol{T}_{k+|\beta|,d})_i \geq \beta$. Diese Bijektion beschreibt dann die Zuordnung der Koeffizienten von $f(x)$ zu den Koeffizienten von $(f \cdot g)(x)$. Eine Zusammenfassung dieser Konstruktion ist in Algorithmus 2.30 angegeben, dessen Anwendung das folgende Beispiel verdeutlicht:

Beispiel 2.31. *Sei $f(x_1, x_2, x_3) = 3x_1 - 2x_2 + 4x_3 + 1$ und $g(x_1, x_2, x_3) = x_2$, dann liefert Algorithmus 2.30 das in Abbildung 2.2 dargestellte Ergebnis. Zuerst bestimmt der Algorithmus die Indexmenge $I = \{3, 6, 7, 9\}$ aller Zeilen von $\boldsymbol{T}_{2,3}$, die größer oder gleich dem Multiindex $\beta = (0, 1, 0)$ sind. Diesen werden die Koeffizienten von f in der von f vorgegebenen Ordnung zugeordnet. Dadurch erhält man $(f \cdot g)(x_1, x_2, x_3) = 3x_1x_2 - 2x_2^2 + 4x_2x_3 + x_2$.*

2.4.3. Multiplikation von Polynomen

Nun werden wir die gerade beschriebene Multiplikation von Polynomen mit Termen auf die Multiplikation zweier Polynome erweitern. Seien dazu $f, g \in \Pi_d$ zwei multi-

variate Polynome mit $f(x) = \sum_{\alpha \in \mathbb{N}_0^d} f_\alpha x^\alpha$ und $g(x) = \sum_{\alpha \in \mathbb{N}_0^d} g_\alpha x^\alpha$. Für das Produkt $(f \cdot g)(x)$ gilt dann

$$(f \cdot g)(x) = f(x)g(x) = \sum_{\alpha \in \mathbb{N}_0^d} g_\alpha x^\alpha \cdot f(x). \tag{2.9}$$

Dabei müssen lediglich Multiplikationen von $f(x)$ mit Termen x^α und Skalaren g_α durchgeführt werden. Dies lässt sich einfach durch Algorithmus 2.30 realisieren. Die anschließende Addition der einzelnen Produkte ist in Abschnitt 2.4.1 beschrieben. Alternativ kann die *gewichtete Summe* von *verschobenen Koeffizientenvektoren* in (2.9) auch als Matrix-Vektor Multiplikation interpretiert werden, wie das folgende Beispiel zeigt:

Beispiel 2.32. *Seien $f, g \in \Pi_2$ gegeben durch*

$$f(x_1, x_2) = \frac{4}{5}x_1 + \frac{3}{5}x_2, \quad g(x_1, x_2) = 3x_1 + 2x_2 + 1.$$

Für das Produkt von $f(x_1, x_2)$ und $g(x_1, x_2)$ folgt dann

$$\begin{aligned}(f \cdot g)(x_1, x_2) &= 3x_1 \cdot f(x_1, x_2) + 2x_2 \cdot f(x_1, x_2) + 1 \cdot f(x_1, x_2) \\ &= 3 \cdot \left(\frac{4}{5}x_1^2 + \frac{3}{5}x_1x_2\right) + 2 \cdot \left(\frac{4}{5}x_1x_2 + \frac{3}{5}x_2^2\right) + 1 \cdot \left(\frac{4}{5}x_1 + \frac{3}{5}x_2\right).\end{aligned} \tag{2.10}$$

Stellen wir die Koeffizientenvektoren der Polynome aus (2.10) als Spalten einer Matrix dar, so erhalten wir

$$(f \cdot g)^T = \frac{1}{5}\begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 4 & 3 \\ 0 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 0 \\ 3 \\ 4 \\ 6 \\ 17 \\ 12 \end{bmatrix}.$$

Dies entspricht genau dem Koeffizientenvektor des Produkts von f und g:

$$(f \cdot g)(x_1, x_2) = \frac{12}{5}x_1^2 + \frac{17}{5}x_1x_2 + \frac{6}{5}x_2^2 + \frac{4}{5}x_1 + \frac{3}{5}x_2.$$

Die dafür notwendige Matrix besteht aus *verschobenen Koeffizientenvektoren* von f in der Form $(x^\alpha \cdot f(x))^T$. Formal führt dies zu folgender Definition:

Definition 2.33. *Sei $f \in \Pi_d$ ein Polynom, dann ordnet die Abbildung*

$$\boldsymbol{C}_{k,d} : \mathbb{R}^{\#\boldsymbol{T}_{\deg(f),d}} \to \mathbb{R}^{\#\boldsymbol{T}_{k,d} \times \#\boldsymbol{T}_{k-\deg(f),d}}, \quad k \geq \deg(f)$$

dem Koeffizientenvektor von f die Faltungsmatrix

$$\boldsymbol{C}_{k,d}(f) := \left[(x^\alpha f)^T : \alpha \in \mathbb{N}_0^d,\ |\alpha| \leq k - \deg(f)\right]$$

vom Grad $k \in \mathbb{N}_0$ zu. Dabei seien die Spalten von $\boldsymbol{C}_{k,d}(f)$ graduiert-lexikographisch bzgl. α angeordnet. Für $k < \deg(f)$ ist die Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ nicht definiert.

Die Einschränkung der Definition auf $k \geq \deg(f)$ ist notwendig, da andernfalls Spalten gebildet werden, die nicht alle Koeffizienten von f enthalten. Dies ist für die Multiplikation eines Polynoms mit einem Term nicht möglich. Für $k > \deg(f)$ können die Faltungsmatrizen auch rekursiv definiert werden, denn es gilt

$$\boldsymbol{C}_{k,d}(f) = \begin{bmatrix} \boldsymbol{C}_{k-1,d}(f) & * \\ 0 & * \end{bmatrix}. \tag{2.11}$$

Die durch $*$ beschriebenen Spalten in (2.11) entsprechen dabei $(x^\alpha f)^T$, $|\alpha| = k - \deg(f)$. Mit Hilfe der Faltungsmatrix lässt sich der Koeffizientenvektor des Produkts zweier Polynome $f, g \in \Pi_d$ nun schreiben als

$$(f \cdot g) = \left(\boldsymbol{C}_{\deg(f)+\deg(g),d}(f) \cdot g^T\right)^T,$$

was einer gewöhnlichen Matrix-Vektor Multiplikation entspricht.

Die Motivation der Bezeichnung *Faltungsmatrix*, die unter anderem von Kaltofen in [KMYZ08] verwendet wird, liegt in der Tatsache begründet, dass die Multiplikation zweier multivariater Polynome auch als *diskrete Faltung* aufgefasst werden kann. In [Bat13] verwendet Batselier stattdessen die Bezeichnung *Multiplikationsmatrix*. Das ist jedoch irreführend, da dieser Begriff im Kontext der polynomiellen Algebra eine andere Bedeutung hat. Für Details zu *Multiplikationsmatrizen*, die auch *Mul-*

tiplikationstabellen genannt werden, sei auf die Ausarbeitung von Stetter in [Ste04] verwiesen. Wir definieren zunächst die diskrete Faltung zweier reellwertiger Folgen im Sinne der digitalen Signalverarbeitung, vgl. [Mal08, Abschnitt 3.3.4].

Definition 2.34. *Seien $f, g : D \to \mathbb{R}$ zwei Folgen und $D \subseteq \mathbb{N}_0^d$, dann ist die* diskrete Faltung *von f und g gegeben durch*

$$(f * g)(n) := \sum_{k \in D} f(k) g(n-k).$$

Fassen wir die Koeffizientenvektoren zweier Polynome $f, g \in \Pi_d$ als Folgen entsprechend Definition 2.34 auf, so erhalten wir die folgende Darstellung der Polynommultiplikation:

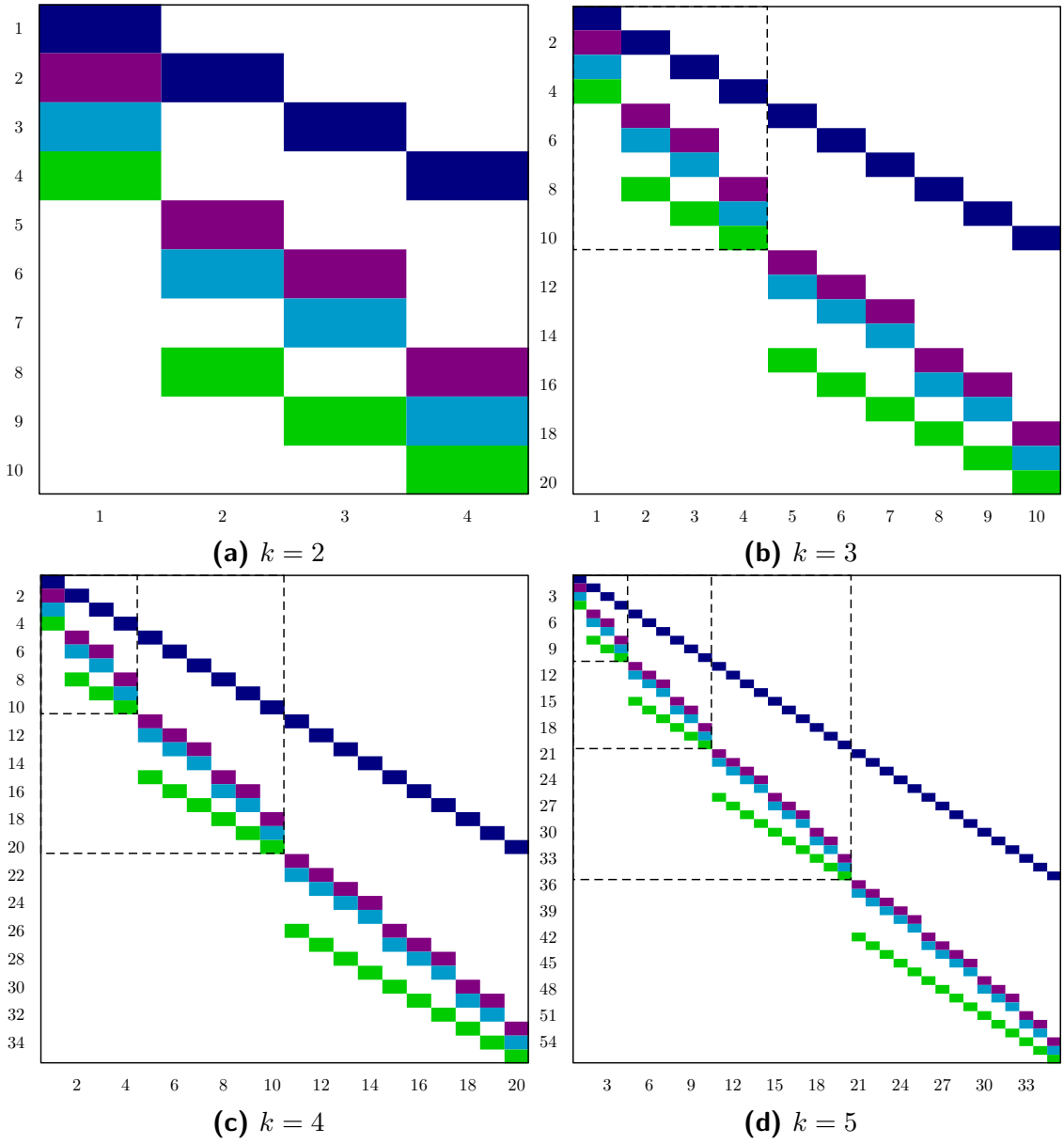
$$(f \cdot g)(x) = \sum_{\alpha \in \mathbb{N}_0^d} g_\alpha x^\alpha \cdot f(x) = \sum_{\alpha \in \mathbb{N}_0^d} \sum_{\beta \in \mathbb{N}_0^d} g_\alpha f_\beta \cdot x^{\alpha+\beta} = \sum_{\alpha \in \mathbb{N}_0^d} \underbrace{\left(\sum_{\beta \in \mathbb{N}_0^d} \tilde{g}_{\alpha-\beta} f_\beta \right)}_{=f*g} \cdot x^\alpha \tag{2.12}$$

mit

$$\tilde{g}_{\alpha-\beta} := \begin{cases} g_{\alpha-\beta} & \text{falls } \beta \leq \alpha, \\ 0 & \text{sonst.} \end{cases}$$

Im univariaten Fall lässt sich daraus eine schnelle Methode zur Polynommultiplikation entwickeln, vgl. hierzu [GG03, Algorithm 8.16]: Man erweitert die Faltungsmatrix zu einer zyklischen Matrix. Damit wird die Faltung der Koeffizientenvektoren zu einer zyklischen Faltung, die extrem schnell durch die *Fast Fourier Transform* (FFT) von Cooley und Tukey (siehe [CT65]) berechnet werden kann. Leider ist es für $d > 1$ im Allgemeinen nicht möglich, die diskrete Faltung (2.12) zyklisch zu erweitern, da multivariate Faltungsmatrizen eine andere Struktur haben – insbesondere sind sie nicht mehr bandiert, vgl. Abbildung 2.3.

Um die oben genannte Methode der *schnellen Polynommultiplikation* dennoch zu verwenden, kann man den sogenannten *Kronecker-Trick* anwenden, den Moenck in [Moe76] beschreibt. Das Prinzip dieses Verfahrens besteht darin, ein multivariates Polynom durch eine geeignete Substitution als univariates Polynom aufzufassen. Auf



(a) $k = 2$ **(b)** $k = 3$

(c) $k = 4$ **(d)** $k = 5$

Abbildung 2.3.: Faltungsmatrizen $\boldsymbol{C}_{k,3}(f)$ eines linearen Polynoms $f \in \Pi_{1,3}$ für $k = 2, \ldots, 5$. Die Bandiertheit der univariaten Faltungsmatrizen geht hier durch ein „Abkippen“ der Bänder verloren. Zudem fällt die in (2.11) beschriebene Rekursionseigenschaft der Faltungsmatrizen auf: Die Matrix $\boldsymbol{C}_{(k-1),3}(f)$ ist als Block links oben in der Matrix $\boldsymbol{C}_{k,3}(f)$ enthalten.

dieses univariate Polynom lassen sich dann die bekannten Verfahren zur Polynommultiplikation anwenden und durch anschließende Resubstitution erhält man das gewünschte Ergebnis. Das folgende Beispiel verdeutlicht dieses Konzept:

Beispiel 2.35. *Seien $f, g \in \Pi_2$ gegeben durch*

$$f(x_1, x_2) = -x_1x_2 + 2x_1 + x_2 - 1, \quad g(x_1, x_2) = 3x_1x_2 - x_1 + 2x_2 + 2.$$

Durch die Substitution $x_2 := x_1^3$ erhält man

$$\widetilde{f}(x_1) = f(x_1, x_1^3) = -x_1^4 + x_1^3 + 2x_1 - 1, \quad \widetilde{g}(x_1) = g(x_1, x_1^3) = 3x_1^4 + 2x_1^3 - x_1 + 2.$$

Anschließend berechnet man $\widetilde{f} \cdot \widetilde{g}$ mit bekannten Methoden (z. B. schnelle Faltung)

$$(\widetilde{f} \cdot \widetilde{g})(x_1) = -3x_1^8 + x_1^7 + 2x_1^6 + 7x_1^5 - 2x_1^4 - 2x_1^2 + 5x_1 - 2$$

und führt eine Rücksubstitution mittels Polynomdivision mit Rest durch:

$$\begin{aligned}(f \cdot g)(x_1, x_2) &= (-3x_1^2 + x_1 + 2)x_2^2 + (7x_1^2 - 2x_1 + 0)x_2 + (-2x_1^2 + 5x_1 - 2) \\ &= -3x_1^2x_2^2 + 7x_1^2x_2 + x_1x_2^2 - 2x_1^2 - 2x_1x_2 + 2x_2^2 + 5x_1 - 2.\end{aligned}$$

Nun stellt sich natürlich die Frage, für welche Substitutionen diese Vorgehensweise möglich ist. Das folgende Resultat von Moenck aus [Moe76] liefert uns die entsprechende Antwort:

Satz 2.36 (Moenck, 1976). *Die Abbildung $\Phi : \Pi_d \to \Pi_1$,*

$$\Phi : x_i \mapsto x_1^{n_i}, \quad 1 \leq i \leq d, \tag{2.13}$$

mit $n_d > \cdots > n_1 = 1$ ist ein Ringhomomorphismus. Sei weiterhin $\Psi : \Pi_1 \to \Pi_d$ ein Homomorphismus mit

$$\Psi : x_1^k \mapsto \begin{cases} 1 & \text{falls } k = 0 \\ \Psi(x_1^r)x_i^q & \text{sonst} \end{cases}, \quad 1 \leq i \leq d, \tag{2.14}$$

wobei $n_{i+1} > k > n_i$, $k = q \cdot n_i + r$, $0 \leq r < n_i$ und $n_d > \cdots > n_1 = 1$. Für alle

Polynome $f \in \Pi_d$ *gilt* $\Psi(\Phi(f)) = f$ *genau dann, wenn*

$$\sum_{j=1}^{i} (\max_{f_\alpha \neq 0} \alpha_j) n_j < n_{i+1}, \quad 1 \leq i < d. \tag{2.15}$$

Die Abbildung $\Phi : \Pi_d \to \Pi_1$ ist unter der Voraussetzung (2.15) ein Isomorphismus, d. h. wir erhalten eine *eindeutige* und somit auch *umkehrbare* Zuordnung von multivariaten Polynomen zu univariaten Polynomen. Die Umkehrung entspricht dabei der in (2.14) angegebenen Abbildung Ψ. Um diese Invertierbarkeit in Beispiel 2.35 herzustellen, muss die Bedingung (2.15) für das Produkt $(f \cdot g)$ sichergestellt werden. Dabei ist zu bemerken, dass $\max_{(f \cdot g)_\alpha \neq 0} \alpha_j = 2$ sein muss, da $\max_{f_\alpha \neq 0} \alpha_j = 1 = \max_{g_\alpha \neq 0} \alpha_j$ gilt. Somit ergibt sich $n_2 = 3$, was der in Beispiel 2.35 gewählten Substitution entspricht.

An dieser Stelle fällt ebenfalls auf, dass die Polynome in Beispiel 2.35 die günstigste Situation des Kronecker-Tricks für quadratische Polynome in zwei Variablen darstellen, denn es gilt $1 \leq \max_{f_\alpha \neq 0} \alpha_j \leq 2$ für $f \in \Pi_2$, $\deg(f) = 2$, $j = 1, 2$. Der Fall $\alpha_j = 0$ für alle $f_\alpha \neq 0$ kann dabei explizit ausgeschlossen werden, da sich ein solches Polynom auch ohne Substitution als univariates Polynom schreiben lässt.

Um die Polynommultiplikation mittels *Kronecker-Trick* mit der zuvor beschriebenen Methode unter Verwendung der *Faltungsmatrix* zu vergleichen, gehen wir im Folgenden stets von dem *worst-case* Szenario $\max_{f_\alpha \neq 0} \alpha_j = \deg(f)$ aus. Diese Annahme ist in numerischer Rechnung durchaus realistisch, da numerisch bestimmte Koeffizientenvektoren im Allgemeinen sehr wenige Einträge mit dem Wert 0 enthalten. Die kleinste mögliche Substitution $\Phi(f)$ für ein Polynom $f \in \Pi_d$ mit $\deg(f) = k$, die (2.15) erfüllt, ist in diesem Fall durch $n_j = (k+1)^{j-1}$, $j = 1, \ldots, d$, gegeben.

Für eine Implementierung der Polynommultiplikation mittels *Kronecker-Trick* benötigen wir nun noch eine Darstellung der Substitution $\Phi : \Pi_d \to \Pi_1$ für Koeffizientenvektoren in graduiert-lexikographischer Ordnung, vgl. Abschnitt 2.2. Dabei hilft uns die *Termmatrix* $\boldsymbol{T}_{k,d}$ aus Definition 2.14: Der Vektor $\boldsymbol{T}_{k,d} \cdot [n_1, \ldots, n_d]^T$ liefert die Indizes des univariaten Polynoms $\Phi(f)$, denen die graduiert-lexikographisch angeordneten Koeffizienten von f zugewiesen werden. Das folgende Beispiel zeigt eine solche Umrechnung für das *worst-case* Szenario $n_j = (k+1)^{j-1}$:

Algorithmus 2.38 : Polynommultiplikation mittels univariater Faltung

Input : $f \in \mathbb{R}^{\#\boldsymbol{T}_{k_1,d}}$, $g \in \mathbb{R}^{\#\boldsymbol{T}_{k_2,d}}$
Output : $(f \cdot g) \in \mathbb{R}^{\#\boldsymbol{T}_{k_1+k_2,d}}$

$k \leftarrow k_1 + k_2$
$I \leftarrow \boldsymbol{T}_{k,d} \cdot [(k+1)^0, \ldots, (k+1)^{d-1}]^T$
$\tilde{f} \leftarrow (0, \ldots, 0) \in \mathbb{R}^{\max(I)}$
$\tilde{g} \leftarrow (0, \ldots, 0) \in \mathbb{R}^{\max(I)}$
for $j = 1, \ldots, \#\boldsymbol{T}_{k_1,d}$ **do**
  $\tilde{f}_{I_j} \leftarrow f_j$
end
for $j = 1, \ldots, \#\boldsymbol{T}_{k_2,d}$ **do**
  $\tilde{g}_{I_j} \leftarrow g_j$
end
$\widetilde{(fg)} \leftarrow \tilde{f} * \tilde{g}$
for $j = 1, \ldots, \#\boldsymbol{T}_{k,d}$ **do**
  $(f \cdot g)_j \leftarrow \widetilde{(fg)}_{I_j}$
end

Beispiel 2.37. *Ist $f \in \Pi_{2,3}$, so erhalten wir die Zuordnung der Multiindizes des Polynoms f in graduiert-lexikographischer Ordnung zu den Indizes des univariaten Polynoms $\Phi(f)$ durch*

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 9 \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \\ 3 \\ 1 \\ 18 \\ 12 \\ 6 \\ 10 \\ 4 \\ 2 \end{bmatrix}.$$

Damit lässt sich nun die Multiplikation von multivariaten Polynomen mittels univariater zyklischer Faltung formulieren, siehe Algorithmus 2.38. Der Operator $*$ steht dabei für die univariate Faltung, die beispielsweise mittels FFT durchgeführt werden kann.

Diese Variante der Multiplikation bringt jedoch nicht in allen Fällen Vorteile. Durch die Substitution im Kronecker-Trick verändert sich die Länge des Koeffizientenvektors eines Polynoms in d Variablen vom Grad k im schlechtesten Fall von $\binom{d+k}{k}$ auf $k(k+1)^{d-1}+1$. Dabei gilt stets $\binom{d+k}{k} \leq k(k+1)^{d-1}+1$ und Gleichheit tritt insbesondere für die Fälle $d = 1$ bzw. $k = 0$, also die trivialen Situationen univariater bzw. konstanter Polynome, ein.

Abbildung 2.4 vergleicht diese Werte und zeigt eine deutliche Verlängerung der Koeffizientenvektoren durch den Kronecker-Trick – insbesondere für viele Variablen. So wird ein Koeffizientenvektor für $d = 7$ bereits bei kubischen Polynomen um einen Faktor der Größenordnung 100 verlängert. Da sich die Anzahl der von Null verschiedenen Koeffizienten durch den Kronecker-Trick nicht ändert, sind die verlängerten Koeffizientenvektoren zwar sehr dünn besetzt, aber diese Eigenschaft geht durch die Fouriertransformation verloren. Dies bremst den Vorteil der schnellen Faltung per FFT derart aus, dass sich das Verfahren nur für eine geringe Anzahl an Variablen lohnt. Eine präzise Analyse dieses Verhaltens erfordert eine genauere Untersuchung der verwendeten Implementierungen von Matrix-Vektor Multiplikation und FFT, die wir an dieser Stelle jedoch nicht durchführen werden.

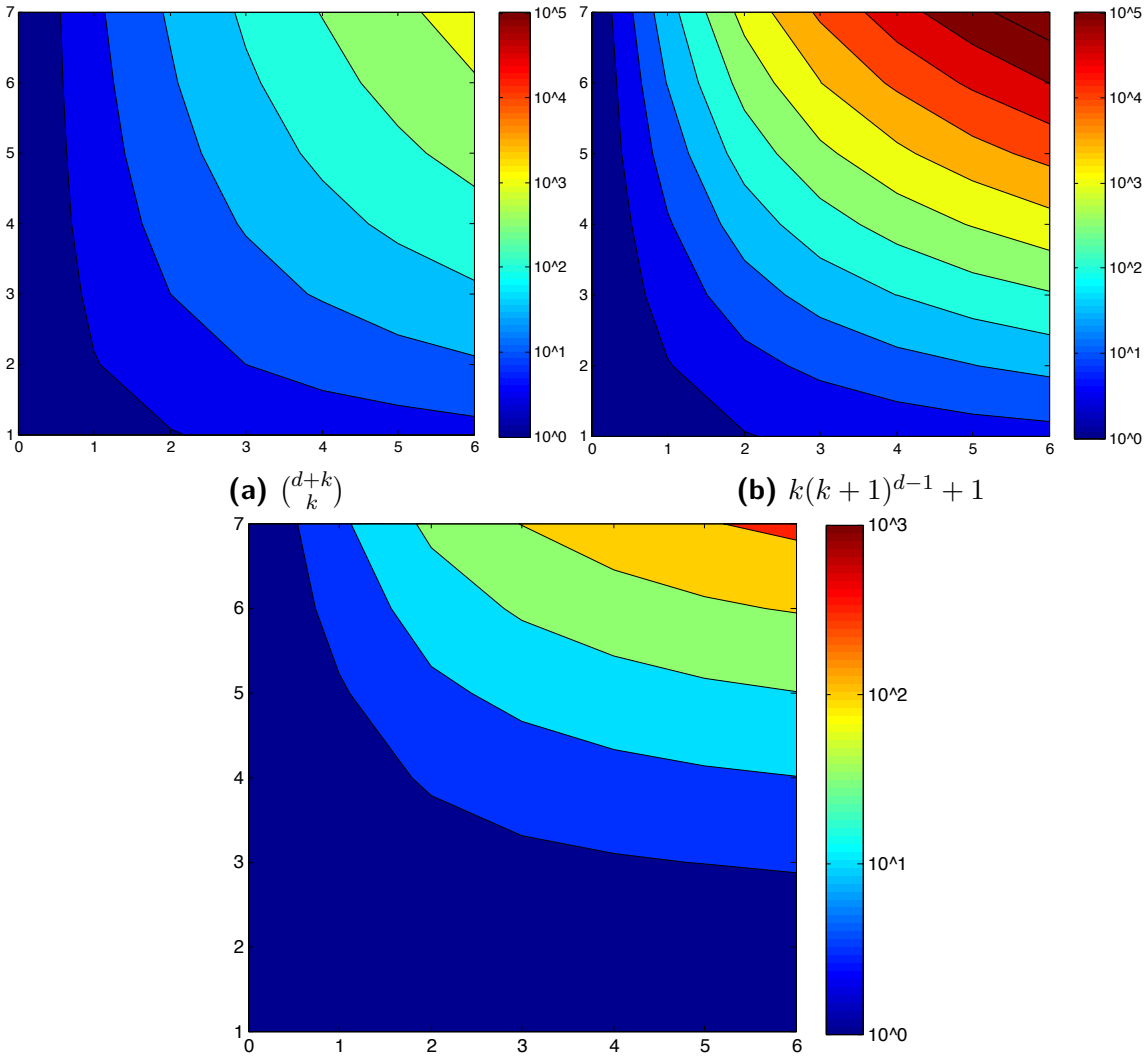
2.4.4. Berechnung von Basen homogener Teilräume

Im Folgenden sei stets $F \subset \Pi_d$ eine endliche Menge von Polynomen. Da im Allgemeinen $\#\mathcal{V}^0_{k,d}(F) = \#\mathcal{W}^0_{k,d}(F) = \infty$ gilt, ist es nicht möglich, die Mengen durch Aufzählung aller Elemente zu beschreiben. Allerdings haben sie als Teilräume des *endlichdimensionalen* Vektorraums $\Pi^0_{k,d}$ eine endliche Basis und es gilt

$$\#\boldsymbol{T}^0_{k,d} = \dim(\Pi^0_{k,d}) = \dim(\mathcal{V}^0_{k,d}(F)) + \dim(\mathcal{W}^0_{k,d}(F)).$$

In Satz 2.8 wurde gezeigt, wie man ein endliches Erzeugendensystem von $\mathcal{V}^0_{k,d}(F)$ bestimmen kann. Es gilt

$$\mathcal{V}^0_{k,d}(F) = \operatorname{span}\left\{x^\alpha \cdot \Lambda(f)(x) : f \in F, \deg(f) \leq k, \alpha \in \mathbb{N}_0^d, |\alpha| = k - \deg(f)\right\}. \tag{2.16}$$



(a) $\binom{d+k}{k}$ **(b)** $k(k+1)^{d-1}+1$

(c) $\Big(k(k+1)^{d-1}+1\Big)/\Big(\binom{d+k}{k}\Big)$

Abbildung 2.4.: Vergleich der Länge der Koeffizientenvektoren in logarithmischer Skala vor und nach der Substitution durch den Kronecker-Trick. Die horizontale Achse beschreibt dabei den Polynomgrad $k \in \mathbb{N}_0$ und die vertikale Achse die Anzahl an Variablen $d \in \mathbb{N}$.

Die dabei konstruierten Polynome $x^\alpha \cdot \Lambda(f)(x)$ erinnern stark an die im letzten Abschnitt definierte Faltungsmatrix. Die folgende Definition der *homogenen Faltungsmatrix* unterscheidet sich daher von Definition 2.33 nur in dem Punkt, dass $|\alpha| = k - \deg(f)$ mit Gleichheit gefordert wird.

Definition 2.39. *Sei $f \in \Pi_d$ ein Polynom, dann ist die* homogene Faltungsmatrix *zu f vom Grad $k \in \mathbb{N}_0$, $k \geq \deg(f)$, definiert durch*

$$\boldsymbol{C}^0_{k,d}(f) := \left[(x^\alpha f)^T : \alpha \in \mathbb{N}_0^d,\ |\alpha| = k - \deg(f)\right].$$

Dabei seien die Spalten von $\boldsymbol{C}^0_{k,d}(f)$ graduiert-lexikographisch bzgl. α angeordnet.

Insbesondere enthält eine *Faltungsmatrix* auch alle Spalten der entsprechenden *homogenen Faltungsmatrix*. Mit Hilfe dieser Definition kann (2.16) nun etwas kompakter dargestellt werden:

$$\mathcal{V}^0_{k,d}(F) = \operatorname{span}\left\{\boldsymbol{C}^0_{k,d}(\Lambda(f)) : f \in F, \deg(f) \leq k\right\} =: \operatorname{span}\left\{\boldsymbol{C}^0_{k,d}(\Lambda(F))\right\}. \quad (2.17)$$

In (2.17) wird dabei implizit die abkürzende Schreibweise

$$\boldsymbol{C}^0_{k,d}(F) := [\boldsymbol{C}^0_{k,d}(f) : f \in F, \deg(f) \leq k]$$

für die Matrix eingeführt, die durch spaltenweise Aneinanderreihung der homogenen Faltungsmatrizen $\boldsymbol{C}^0_{k,d}(f)$, $f \in F$ entsteht. Die Forderung $\deg(f) \leq k$ ergibt sich dabei auf natürliche Weise, da die Matrix $\boldsymbol{C}^0_{k,d}(f)$ für Polynome f mit $\deg(f) > k$ nicht definiert ist. Da die Reihenfolge der Polynome $f \in F$ nicht festgelegt wird, ist die Matrix $\boldsymbol{C}^0_{k,d}(F)$ *nicht eindeutig*. Im Folgenden ist jedoch hauptsächlich der Spaltenraum der Matrix relevant und dieser wird durch eine Vertauschung von Spalten innerhalb der Matrix nicht beeinflusst.

Auch die homogene Faltungsmatrix generiert in (2.17) nicht notwendigerweise eine Basis von $\mathcal{V}^0_{k,d}(F)$, sondern nur ein möglicherweise überrepräsentiertes Erzeugendensystem. Dies wird im folgenden Beispiel deutlich:

Beispiel 2.40. *Sei $F = \{x_1 + 42, -2x_1 + 3x_2 - 5\} \subset \Pi_3$, dann gilt $\Lambda(F) = \{x_1, -2x_1 + 3x_2\}$. Nach Satz 2.8 bzw. (2.16) erhalten wir ein Erzeugendensystem*

von $\mathcal{V}^0_{2,3}(F)$ *durch*

$$\begin{aligned}\mathcal{V}^0_{2,3}(F) &= \operatorname{span}\{x_1^2, x_1x_2, x_1x_3, (-2x_1+3x_2)x_1, (-2x_1+3x_2)x_2, (-2x_1+3x_2)x_3\}\\ &= \operatorname{span}\{x_1^2, x_1x_2, x_1x_3, -2x_1^2+3x_1x_2, -2x_1x_2+3x_2^2, -2x_1x_3+3x_2x_3\}.\end{aligned}$$

Dabei gilt $(-2x_1^2+3x_1x_2) \in \operatorname{span}\{x_1^2, x_1x_2\}$ *und das Erzeugendensystem ist überrepräsentiert. Ein* minimales Erzeugendensystem *– also eine* Basis *– ist durch*

$$\mathcal{V}^0_{2,3}(F) = \operatorname{span}\{x_1^2, x_1x_2, x_1x_3, -2x_1x_2+3x_2^2, -2x_1x_3+3x_2x_3\}$$

gegeben. Betrachten wir die homogene Faltungsmatrix $\boldsymbol{C}^0_{2,3}(\Lambda(F))$*, so erhalten wir ein entsprechendes Resultat, denn die Matrix weist einen Rangdefekt von 1 auf:*

$$\boldsymbol{C}^0_{2,3}(\Lambda(F)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 3\\ 0 & 0 & 0 & 0 & 3 & 0\\ 0 & 0 & 1 & 0 & 0 & -2\\ 0 & 1 & 0 & 3 & -2 & 0\\ 1 & 0 & 0 & -2 & 0 & 0 \end{bmatrix}.$$

In Beispiel 2.40 fällt auf, dass homogene Faltungsmatrizen, die nur von Leitformen abhängen, auf triviale Weise viele Nullzeilen enthalten. Da die Matrix $\boldsymbol{C}^0_{k,d}(\Lambda(F))$ erst ab der Zeile $\#\boldsymbol{T}_{k-1,d}+1$ von Null verschiedene Werte enthalten kann, reicht es aus, nur diese Teilmatrix zu betrachten. Wir definieren dazu die Notation

$$\mathbb{R}^{\#\boldsymbol{T}_{k,d}\times m} \ni \boldsymbol{C}^0_{k,d}(F) =: \begin{bmatrix} \overbrace{0}^{\sim} \\ \boldsymbol{C}^0_{k,d}(\Lambda(F)) \end{bmatrix}, \quad \boldsymbol{C}^0_{k,d}(\widetilde{\Lambda(F)}) \in \mathbb{R}^{\#\boldsymbol{T}^0_{k,d}\times m}.$$

Diese Teilmatrix ist dabei nicht nur effizienter im Bezug auf die Speicherung. Sie ermöglicht es uns, während der Bestimmung einer Basis von $\mathcal{V}^0_{k,d}(F)$ direkt eine Basis von $\mathcal{W}^0_{k,d}(F)$ mitzuberechnen. Zunächst sei jedoch bemerkt, dass auch die Teilmatrix

$\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$ Nullzeilen enthalten kann.

Beispiel 2.41. *Sei $\boldsymbol{C}^0_{2,3}(\Lambda(F))$ die Matrix aus Beispiel 2.40, dann gilt*

$$\widetilde{\boldsymbol{C}^0_{2,3}(\Lambda(F))} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 3 & -2 & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 \end{bmatrix}.$$

An dieser Stelle benötigen wir zwei wichtige Begriffe aus der Linearen Algebra:

Definition 2.42. *Sei $A \in \mathbb{R}^{m\times n}$ eine Matrix, dann definiert diese eine lineare Abbildung $A : \mathbb{R}^n \to \mathbb{R}^m$, $A : x \mapsto Ax$. Mit*

$$\mathcal{R}(A) := \{Ax : x \in \mathbb{R}^n\}$$

bezeichnen wir den Bildraum (engl.: „range“) *dieser Abbildung und mit*

$$\mathcal{N}(A) := \{x \in \mathbb{R}^n : Ax = 0\}$$

ihren Kern *bzw.* Nullraum. *Der Kern ist trivial – d. h. $\mathcal{N}(A) = \{0\}$ – falls A vollen Spaltenrang hat.*

Damit können wir nun einen Zusammenhang zwischen der abgeschnittenen homogenen Faltungsmatrix $\boldsymbol{C}^0_{k,d}(\Lambda(F))$ und den Räumen $\mathcal{V}^0_{k,d}(F)$ bzw. $\mathcal{W}^0_{k,d}(F)$ herstellen.

Satz 2.43. *Es gilt*

$$v \in \mathcal{R}\left(\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}\right) \quad \Leftrightarrow \quad \begin{bmatrix} 0 \\ v \end{bmatrix} \in \mathcal{V}^0_{k,d}(F)$$

sowie

$$w \in \mathcal{N}\left(\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}^T\right) \quad \Leftrightarrow \quad \begin{bmatrix} 0 \\ w \end{bmatrix} \in \mathcal{W}^0_{k,d}(F).$$

Es reicht also aus, eine Basis des Bildraums von $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$ bzw. des Nullraums von $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}^T$ zu bestimmen. Dieses Problem ist jedoch sehr gut untersucht und numerisch stabil lösbar. Ein bekanntes Verfahren aus der numerischen Linearen Algebra verwendet dazu die *Singulärwertzerlegung* (SVD, engl.: „singular value decomposition“), die durch folgenden Satz gegeben ist, vgl. [HB02, Satz 12.1]:

Satz 2.44. *Sei $A \in \mathbb{R}^{m\times n}$ eine Matrix mit Rang $p \leq \min\{m,n\}$, dann gibt es orthogonale Matrizen $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ und eine Matrix*

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & \sigma_p & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right] \in \mathbb{R}^{m\times n}, \quad \sigma_j > 0,$$

sodass $A = U\Sigma V^T$ gilt.

Die Einträge σ_j werden als *Singulärwerte* bezeichnet, die Spalten von U bzw. V als *Singulärvektoren*. Der folgende Satz zeigt, dass sich die Basisvektoren von $\mathcal{R}(A)$ und $\mathcal{N}(A^T)$ aus der SVD einer Matrix A ablesen lassen, vgl. [HB02, Satz 12.4]:

Satz 2.45. *Sei $A \in \mathbb{R}^{m\times n}$ mit Rang p und $A = U\Sigma V^T$ wie in Satz 2.44, dann bilden die Singulärvektoren $u_1, \ldots, u_p$ eine Basis von $\mathcal{R}(A)$ und die Singulärvektoren $u_{p+1}, \ldots, u_m$ eine Basis von $\mathcal{N}(A^T)$.*

Kombinieren wir nun Satz 2.43 und Satz 2.45, so können wir Basen von $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$ mit Methoden der numerischen Linearen Algebra bestimmen. Algorithmus 2.46 fasst das beschriebene Vorgehen noch einmal zusammen und liefert mit $V, W \subset \Pi^0_{k,d}$ die gesuchten Basen. In einer numerischen Implementierung sollte man Algorithmus 2.46 noch leicht modifizieren. An Stelle von $\sigma_p > 0$ in Zeile 9 fordert man üblicherweise nur $\sigma_p > \varepsilon > 0$ für eine gegebene Toleranz ε. Eine mögliche Vorgabe für diese Toleranz, die von GNU Octave für die Bestimmung des Nullraumes von $A \in \mathbb{R}^{m\times n}$ verwendet wird, lautet $\varepsilon = \max\{n,m\} \cdot \max_j \sigma_j \cdot \widetilde{\varepsilon}$, wobei $\widetilde{\varepsilon}$ die Maschinengenauigkeit beschreibt. Die Berechnung der Singulärwertzerlegung in Zeile 8 von Algorithmus 2.46 kann man mit Hilfe von LAPACK (siehe [LAP13]) oder Ähnlichem durchführen.

Algorithmus 2.46 : Bestimmung einer Basis von $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$

Input : $F \subset \Pi_d$, $k \in \mathbb{N}_0$
Output : $V \leftarrow \{u_1, \ldots, u_p\}$, $W \leftarrow \{u_{p+1}, \ldots, u_{\#\boldsymbol{T}^0_{k,d}}\}$

if $\min_{f \in F} \deg(f) \geq k$ **then**
  $p \leftarrow 0$
  for $j = 1, \ldots, \#\boldsymbol{T}^0_{k,d}$ **do**
    $u_j \leftarrow (\underbrace{0, \ldots, 0}_{j-1}, 1, 0, \ldots, 0)^T \in \mathbb{R}^{\#\boldsymbol{T}^0_{k,d}}$
  end
else
  Bestimme $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$.
  Zerlege $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))} = U\Sigma V^T$ mittels SVD
  $p \leftarrow \max\{p : \sigma_p > 0\}$
end
for $j = 1, \ldots, \#\boldsymbol{T}^0_{k,d}$ **do**
  $u_j \leftarrow [\underbrace{0, \ldots, 0}_{\#\boldsymbol{T}_{k-1,d}}, u_j^T]^T$
end

2.4.5. Polynomdivision

Als letzte der vier Grundrechenoperationen fehlt noch die Division. Im Allgemeinen existiert zu zwei Polynomen $f, g \in \Pi_d$ jedoch *kein* Polynom $h \in \Pi_d$ mit $f \cdot h = g$, da Π_d kein Körper ist. Für $d > 1$ ist Π_d auch kein euklidischer Ring, sodass die aus dem univariaten Fall bekannte *Polynomdivision mit Rest* (vgl. [Fis05, 1.3.7]) der Form

$$g = h \cdot f + r, \quad h, r \in \Pi_1, \ \deg(r) < \deg(f), \quad f, g \in \Pi_1, \tag{2.18}$$

nicht anwendbar ist. Mit Hilfe der linearen Räume $\mathcal{V}_d(F)$ und $\mathcal{W}_d(F)$, $F \subset \Pi_d$, aus Definition 2.7 können wir jedoch eine ähnliche Zerlegung eines Polynoms $g \in \Pi_d$ bzgl. einer Menge F definieren, vgl. [Sau01].

Definition 2.47. *Eine endliche Menge $F \subset \Pi_d$ teilt $g \in \Pi_d$ mit Rest $r \in \Pi_d$, wenn*

$$g = \sum_{f \in F} g_f f + r, \quad r \in \mathcal{W}_{\deg(g),d}(F), \quad \deg(g_f f) \leq \deg(g). \tag{2.19}$$

Die Division mit Rest in (2.19) liefert also eine Zerlegung des Polynoms g in einen Anteil aus $\mathcal{V}_{\deg(g),d}(F)$ und einen dazu orthogonalen Rest. Die Forderungen an den Totalgrad $\deg(g_f f) \leq \deg(g)$ und $\deg(r) \leq \deg(g)$ sorgen dabei für eine gewisse *Redundanzfreiheit*, da in $g_f f$ keine Terme höheren Grades entstehen können, die durch r wieder ausgelöscht werden. Dennoch ist die Forderung schwächer als die Gradforderung in (2.18), die für die Eindeutigkeit der Zerlegung sorgt, denn diese Eindeutigkeit gilt unter den Voraussetzungen von Definition 2.47 nicht.

Im Folgenden wird nun eine Methode zur Bestimmung einer orthogonalen Zerlegung von g bzgl. F im Sinne von Definition 2.47 sowie der Faktoren $g_f \in \Pi_d$, $f \in F$, hergeleitet. Da im letzten Abschnitt bereits ein Algorithmus zur Berechnung von Basen der homogenen Räume $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$ bereitgestellt wurde, sei zunächst vereinfachend angenommen, dass g homogen vom Grad $k \in \mathbb{N}_0$ ist, alle $f \in F$ ebenfalls homogen sind und $\deg(f) \leq k$, $f \in F$, gilt. Das folgende Beispiel zeigt eine Zerlegung nach (2.19) unter diesen Voraussetzungen und ist an ein Beispiel aus [Sau10] angelehnt:

Beispiel 2.48. *Sei $F = \{x_1^2 + x_2^2\}$ und $g = x_1^3 + x_2^3$ gegeben. Da die Polynome $F \subset \Pi^0_2$ und $g \in \Pi^0_{3,2}$ alle homogen sind, reicht es aus, die homogenen Räume*

$$\mathcal{V}^0_{3,2}(F) = \operatorname{span}\left\{x_1^3 + x_1x_2^2, x_1^2x_2 + x_2^3\right\}, \quad \mathcal{W}^0_{3,2}(F) = \operatorname{span}\left\{x_1^3 - x_1x_2^2, x_1^2x_2 - x_2^3\right\}$$

zu betrachten. Zerlegen wir nun $g \in \Pi^0_{3,2} = \mathcal{V}^0_{3,2}(F) \oplus \mathcal{W}^0_{3,2}(F)$ bezüglich dieser Räume, so erhalten wir

$$\begin{aligned} g(x_1, x_2) &= \underbrace{\left(\frac{1}{2}x_1^3 + \frac{1}{2}x_1^2x_2 + \frac{1}{2}x_1x_2^2 + \frac{1}{2}x_2^3\right)}_{\in \mathcal{V}^0_{3,2}(F)} + \underbrace{\left(\frac{1}{2}x_1^3 - \frac{1}{2}x_1^2x_2 - \frac{1}{2}x_1x_2^2 + \frac{1}{2}x_2^3\right)}_{\in \mathcal{W}^0_{3,2}(F)} \\ &= \underbrace{\left(\frac{1}{2}x_1 + \frac{1}{2}x_2\right)}_{=:g_f}\left(x_1^2 + x_2^2\right) + \underbrace{\left(\frac{1}{2}x_1^3 - \frac{1}{2}x_1^2x_2 - \frac{1}{2}x_1x_2^2 + \frac{1}{2}x_2^3\right)}_{=:r}. \end{aligned}$$

Seien nun durch $\operatorname{span}\{v_1, \ldots, v_p\} = \mathcal{V}^0_{k,d}(F)$ und $\operatorname{span}\{w_1, \ldots, w_q\} = \mathcal{W}^0_{k,d}(F)$ zwei Basen dieser Räume gegeben. Die orthogonale Zerlegung von

$$g \in \Pi^0_{k,d} = \mathcal{V}^0_{k,d}(F) \oplus \mathcal{W}^0_{k,d}(F) = \operatorname{span}\{v_1, \ldots, v_p, w_1, \ldots, w_q\},$$

lässt sich mit Hilfe der Basisvektoren v_j und w_j darstellen durch

$$g^T = \underbrace{\sum_{j=1}^{p} c_j v_j}_{\in \mathcal{V}^0_{k,d}(F)} + \underbrace{\sum_{j=1}^{q} c_{p+j} w_j}_{\in \mathcal{W}^0_{k,d}(F)} =: \sum_{j=1}^{p} c_j v_j + r^T. \tag{2.20}$$

Die Koeffizienten $c_j \in \mathbb{R}$ in der Darstellung (2.20) erhält man als Lösung des linearen Gleichungssystems

$$[v_1, \ldots, v_p, w_1, \ldots, w_q] \cdot c = g^T.$$

Dies liefert zusammen mit der Homogenität von F und g die gesuchte Zerlegung in $\mathcal{V}^0_{k,d}(F) \subseteq \mathcal{V}_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F) \subseteq \mathcal{W}_{k,d}(F)$. Damit kennen wir bereits den Divisionsrest $r \in \Pi_d$, vgl. Beispiel 2.48. Für den Anteil aus $\mathcal{V}^0_{k,d}(F)$ ist jedoch nur eine Darstellung bzgl. der Basisvektoren v_j, $j = 1, \ldots, p$, und nicht, wie gewünscht, bzgl. $f \in F$ gegeben. Um die Vektoren v_j wieder auf die Polynome $f \in F$ zurückzuführen, müssen wir die Berechnung von Algorithmus 2.46, insbesondere die Singulärwertzerlegung, rückwärts rechnen.

Betrachten wir die Singulärwertzerlegung $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))} = U \cdot \Sigma \cdot V^T$, so gilt für die einzelnen Spalten der Zusammenhang

$$(U)_j = \frac{1}{\sigma_j} \widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))} (V)_j, \quad j = 1, \ldots, p. \tag{2.21}$$

Erinnern wir uns nun, dass in Algorithmus 2.46 eine Basis von $\mathcal{V}^0_{k,d}(F)$ aus den ersten p Spalten der Matrix U konstruiert wurde, so können wir (2.21) in (2.20) einsetzen und erhalten

$$g^T = \sum_{j=1}^{p} c_j (U)_j + r^T = \boldsymbol{C}^0_{k,d}(\Lambda(F)) \underbrace{\left(\sum_{j=1}^{p} \frac{c_j}{\sigma_j} (V)_j \right)}_{=: \widetilde{c}} + r^T = \boldsymbol{C}^0_{k,d}(\Lambda(F)) \widetilde{c} + r^T. \tag{2.22}$$

Der Übergang von $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$ in (2.21) zu $\boldsymbol{C}^0_{k,d}(\Lambda(F))$ in (2.22) entspricht dabei dem Auffüllen der Nullen in Zeile 12 von Algorithmus 2.46. Im nächsten Schritt teilen wir die homogene Faltungsmatrix $\boldsymbol{C}^0_{k,d}(\Lambda(F))$ wieder in die Blöcke $\boldsymbol{C}^0_{k,d}(\Lambda(f))$, $f \in F$

auf, was zu

$$g^T = \sum_{f \in F} \boldsymbol{C}^0_{k,d}(\Lambda(f))\widetilde{c}_f + r^T, \quad \widetilde{c}_f \in \mathbb{R}^{\#T^0_{k-\deg(f),d}}$$

führt. In Abschnitt 2.4.3 wurde gezeigt, dass die Multiplikation einer Faltungsmatrix mit einem Koeffizientenvektor genau der Multiplikation der zugrunde liegenden Polynome entspricht. Dies gilt natürlich auch für homogene Faltungsmatrizen. Das Produkt $\boldsymbol{C}^0_{k,d}(\Lambda(f))\widetilde{c}_f$ entspricht also der Multiplikation von $\Lambda(f)$ mit einem Polynom g_f, das definiert ist durch

$$g_f^T := \begin{bmatrix} 0 \\ \widetilde{c}_f \end{bmatrix} \in \mathbb{R}^{\#T_{k-\deg(f),d}}.$$

Somit erhalten wir die gesuchte Darstellung

$$g = \Lambda(g) = \sum_{f \in F} g_f \Lambda(f) + r = \sum_{f \in F} g_f f + r.$$

Um eine solche Polynomdivision auch für nichthomogene Polynome g berechnen zu können, müssen wir das Verfahren erweitern. Dies führt zu Algorithmus 2.49, der im Wesentlichen dem *Reduktionsalgorithmus* aus [Sau01] entspricht. Das Prinzip der *Reduktion* besteht darin, den Grad von g schrittweise durch *homogene Polynomdivision* der aktuellen Leitform mit dem oben beschriebenen Verfahren zu verkleinern. Einen ähnlichen Ansatz verfolgen auch die Divisionsalgorithmen aus [CLO07, Kapitel 2, §3, Theorem 3] und [GG03], die im Gegensatz zu dem hier vorgestellten Verfahren auf einer Termordnung basieren.

Da für die Reste der homogenen Polynomdivision in Zeile 9 von Algorithmus 2.49 stets $r^{(j)} \in \Pi^0_{j,d}$ gilt, erhält man automatisch auch eine Zerlegung $r = \sum_{j=0}^{\deg(g)} r^{(j)}$ des Gesamtrests bzgl. $\mathcal{W}_{\deg(g),d}(F) = \bigoplus_{j=0}^{\deg(g)} \mathcal{W}^0_{j,d}(F)$ im Sinne der in Abschnitt 2.1 beschriebenen *direkten Summe*. Diese Zerlegung wird in Abschnitt 4.3 noch einmal aufgegriffen.

Ein bekanntes Problem bei dieser Art der Polynomdivision besteht nun darin, dass die erzeugte Darstellung der Form (2.19) nicht eindeutig ist. Das folgende Beispiel zeigt, dass eine solche Situation bereits im univariaten Fall auftreten kann:

Algorithmus 2.49 : Verallgemeinerte Polynomdivision

Input : $F = \{f_1, \ldots, f_n\} \subset \Pi_d$, $g \in \Pi_{k,d}$
Output : $g_f \in \Pi_d$, $r \in \mathcal{W}_{k,d}(F)$ mit $g = \sum_{f \in F} g_f f + r$

$r \leftarrow 0$
$g_f \leftarrow 0$, $f \in F$
$g^{(\deg(g))} \leftarrow g$
for $j = \deg(g), \ldots, 0$ **do**
  if $\deg(g(j)) < j$ **then**
    $g^{(j-1)} \leftarrow g^{(j)}$
    $r^{(j)} \leftarrow 0$
  else
    Berechne homogene Polynomdivision $\Lambda(g^{(j)}) = \sum_{f \in F} h_f \Lambda(f) + r^{(j)}$
    $r \leftarrow r + r^{(j)}$
    $g^{(j-1)} \leftarrow g^{(j)} - \sum_{f \in F} h_f f - r^{(j)}$
    $g_f \leftarrow g_f + h_f$
  end
  if $g^{(j)} = 0$ **then break**;
end

Beispiel 2.50. *Sei $d = 1$, $F = \{f_1, f_2\}$ mit $f_1(x) = x$, $f_2(x) = x^2 + 1$ und $g(x) = x^2 + x + 1$ gegeben. Wenden wir Algorithmus 2.49 auf diese Polynome an, so erhalten wir*

1. $g^{(2)}(x) = x^2 + x + 1$, $\Lambda(g^{(2)})(x) = x^2 = x \cdot \Lambda(f_1)(x) + 0 \cdot \Lambda(f_2)(x)$, $r^{(2)}(x) = 0$,
2. $g^{(1)}(x) = x + 1$, $\Lambda(g^{(1)})(x) = x = 1 \cdot \Lambda(f_1)(x) + 0 \cdot \Lambda(f_2)(x)$, $r^{(1)}(x) = 0$,
3. $g^{(0)}(x) = 1$, $\Lambda(g^{(0)})(x) = 0 \cdot \Lambda(f_1)(x) + 0 \cdot \Lambda(f_2)(x) + 1$, $r^{(0)}(x) = 1$.

$$\Longrightarrow\ g(x) = (x + 1) \cdot f_1(x) + 1.$$

Dies ist jedoch nicht die einzige Lösung des gegebenen Problems, denn die Zuordnung $x^2 = x \cdot \Lambda(f_1)(x)$ in Schritt 1 ist nicht eindeutig. An dieser Stelle wäre auch die Wahl $x^2 = \Lambda(f_2)(x)$ möglich gewesen. Vertauschen wir die Priorität von f_1 und f_2, so erhalten wir ein anderes Resultat, das ebenfalls der gewünschten Form entspricht:

1. $g^{(2)}(x) = x^2 + x + 1$, $\Lambda(g^{(2)})(x) = x^2 = 0 \cdot \Lambda(f_1)(x) + 1 \cdot \Lambda(f_2)(x)$, $r^{(2)}(x) = 0$,
2. $g^{(1)}(x) = x$, $\Lambda(g^{(1)})(x) = x = 1 \cdot \Lambda(f_1)(x) + 0 \cdot \Lambda(f_2)(x)$, $r^{(1)}(x) = 0$,

3. $g^{(0)}(x) = 0 \implies$ *Abbruch!*

$$\implies g(x) = 1 \cdot f_1(x) + 1 \cdot f_2(x).$$

Das in Beispiel 2.50 gezeigte Problem der Mehrdeutigkeit tritt immer dann auf, wenn es zwei Polynome $f_1 \neq f_2 \in F$ mit $x^\alpha \cdot \Lambda(f_1) = x^\beta \cdot \Lambda(f_2)$, $\alpha, \beta \in \mathbb{N}_0^d$ gibt. Um solche Fälle auszuschließen, gilt es weitere Forderungen an die Menge $F \subset \Pi_d$ zu stellen. Die im nächsten Kapitel definierten *H-Basen* erfüllen diese Forderungen und ermöglichen insbesondere eine Polynomdivision im Sinne von Algorithmus 2.49 mit *eindeutigem Rest*.

3 Varietäten und Ideale

Inhalt

3.1. Ideale und Idealbasen **50**

3.2. Darstellung von Idealen **53**

3.3. Varietäten und der Nullstellensatz **62**

3.4. Endlichkeit vs. Geometrie **67**

Nachdem nun die Grundlagen der multivariaten Polynome besprochen wurden, beschreibt dieses Kapitel eine wichtige Charakteristik: die *Nullstellen*. Im univariaten Fall folgt aus dem *Fundamentalsatz der Algebra* (vgl. [Fis05, 1.3]), dass ein Polynom $f \in \Pi_1$ vom Grad k höchstens k reelle Nullstellen hat. Für multivariate Polynome gilt dies hingegen nicht: Bereits das lineare Polynom $f(x_1, x_2) = x_1 - x_2$ hat unendlich viele Nullstellen in $\mathbb{R}^2$. Somit ist es möglich, geometrische Objekte durch die *gemeinsamen Nullstellen* einer Menge multivariater Polynome zu beschreiben. Solche Punktmengen werden auch als *algebraische Varietäten* bezeichnet.

Da diese Varietäten in einem engen Zusammenhang mit den *Idealen* des Polynomrings Π_d stehen, werden zunächst einige Grundlagen der Idealtheorie angegeben. In [CLO07] wird diese Verbindung auch als das *Algebra-Geometry Dictionary* bezeichnet, denn in vielen Fällen kann man zwischen Algebra und Geometrie „übersetzen“. Als zentrales Hilfsmittel zur Darstellung von Idealen werden H-Basen eingeführt und ein konstruktives Verfahren zur Berechnung solcher Idealbasen beschrieben, das auf

die Arbeit von Möller und Sauer zurückgeht, vgl. [MS00b]. Dabei nehmen die homogenen Teilräume $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$ aus Abschnitt 2.4.4 eine zentrale Rolle ein. Ein weiteres wichtiges Resultat von Möller und Sauer aus [MS00a] zeigt eine invariante Eigenschaft aller H-Basen eines Ideals und ermöglicht es insbesondere, aus einer bekannten H-Basis weitere H-Basen desselben Ideals zu konstruieren.

Im nächsten Abschnitt wird ein Zusammenhang zwischen dem geometrischen Standpunkt, der Operationen wie Schnitt, Vereinigung etc. von algebraischen Varietäten beschreibt, und den entsprechenden algebraischen Verknüpfungen auf dem Ring Π_d hergestellt. Dieser Teil folgt im Wesentlichen den Ausführungen von Cox, Little und O'Shea in [CLO07]. Mit Hilfe dieser Grundlagen wird gezeigt, dass zur Untersuchung endlicher, reeller Varietäten die Betrachtung reeller Polynome genügt.

Abschließend findet eine Untersuchung von Diskretisierungen algebraischer Varietäten statt. Darauf aufbauend wird ein Kriterium für die Entscheidung entwickelt, ob eine Varietät in einer niederdimensionalen Untermannigfaltigkeit liegt. Dies ermöglicht dann die Erkennung von nichttrivialen geometrischen Eigenschaften einer Varietät.

3.1. Ideale und Idealbasen

In diesem Abschnitt werden zunächst einige zentrale Begriffe aus dem Bereich der (Computer-)Algebra definiert, die auch für multivariate Polynome von Bedeutung sind. Die hier vorgestellten Resultate finden sich in vielen einführenden Werken wie [CLO07] oder [KR00]. Die Definition und die Eigenschaften eines Ideals können dabei auch für allgemeine Ringe formuliert werden – für die Untersuchungen dieser Arbeit ist jedoch die Betrachtung des Polynomrings Π_d hinreichend.

Definition 3.1. *Eine Menge* $\emptyset \neq \mathcal{F} \subseteq \Pi_d$ *heißt* Ideal *von* Π_d*, wenn* $\mathcal{F} + \mathcal{F} = \mathcal{F}$ *und* $\Pi_d \cdot \mathcal{F} = \mathcal{F}$ *gilt.*

Ein Ideal $\mathcal{F}$ ist damit abgeschlossen bzgl. der Addition zweier Polynome des Ideals sowie der Multiplikation mit Polynomen des Rings Π_d. Diese Eigenschaft wird in folgendem Resultat zur Konstruktion von Idealen verwendet:

Satz 3.2. *Seien $F \subseteq \Pi_d$ und $\{f_1, \ldots, f_n\} \subset \Pi_d$, dann sind die Mengen*

$$\langle F \rangle := \bigcap_{\substack{\mathcal{G} \text{ Ideal von } \Pi_d \\ F \subseteq \mathcal{G}}} \mathcal{G} \tag{3.1}$$

und

$$\langle f_1, \ldots, f_n \rangle := \left\{ \sum_{j=1}^{n} g_j f_j : g_j \in \Pi_d \right\} \subseteq \Pi_d \tag{3.2}$$

Ideale in Π_d. Für $F = \{f_1, \ldots, f_n\}$ stimmen die Ideale überein. Dabei bezeichnen wir F bzw. $\{f_1, \ldots, f_n\}$ als Basis *des jeweiligen Ideals.*

Jedes Ideal $\mathcal{F} \subseteq \Pi_d$ hat eine triviale Basis im Sinne von Satz 3.2: das Ideal $\mathcal{F}$ selbst. Interessant sind daher vor allem Basen, die aus weniger Elementen bestehen. Besonders einfache Basen lassen sich zu *monischen Idealen* angeben: Ein monisches Ideal wird von einer Menge von Termen erzeugt und es reichen stets endlich viele davon aus. Dieses Resultat ist auch als *Dicksons Lemma* bekannt, vgl. [CLO07, Kapitel 2, §4, Theorem 5].

Satz 3.3 (Dicksons Lemma)**.** *Jedes* monische Ideal $I_A := \langle x^\alpha : \alpha \in A \rangle$, $\emptyset \neq A \subset \mathbb{N}_0^d$, *ist endlich erzeugt. D. h. zu jeder Menge $\emptyset \neq A \subset \mathbb{N}_0^d$ gibt es eine endliche Teilmenge $\widetilde{A} \subseteq A$, $\#\widetilde{A} < \infty$, sodass $I_A = I_{\widetilde{A}}$ gilt.*

Monische Ideale lassen sich zudem sehr leicht auf einem d-dimensionalen ganzzahligen Gitter visualisieren. In Abbildung 3.1 ist ein Beispiel einer solchen Darstellung gegeben. Zudem zeigt die Abbildung das Komplement des monischen Ideals im Sinne aller Terme, die nicht im Ideal liegen. Diese Menge bildet ein sogenanntes *Ordnungsideal*, bzw. eine *untere Menge* (aus dem Englischen „*lower set*“, siehe [Bla00]).

Definition 3.4. *Eine Menge von Termen $\mathcal{O} \subset \{x^\alpha : \alpha \in \mathbb{N}_0^d\}$ heißt* Ordnungsideal *oder* untere Menge*, falls zu jedem $x^\alpha \in \mathcal{O}$ auch alle Teiler in $\mathcal{O}$ liegen, d. h. $x^\beta \in \mathcal{O}$ für alle $\beta \leq \alpha$ erfüllt ist.*

Ordnungsideale stellen die Grundlage von *Randbasen* dar, die in Kapitel 6 in Form von approximativen Randbasen von Bedeutung sind. Für weitere Details zu diesem

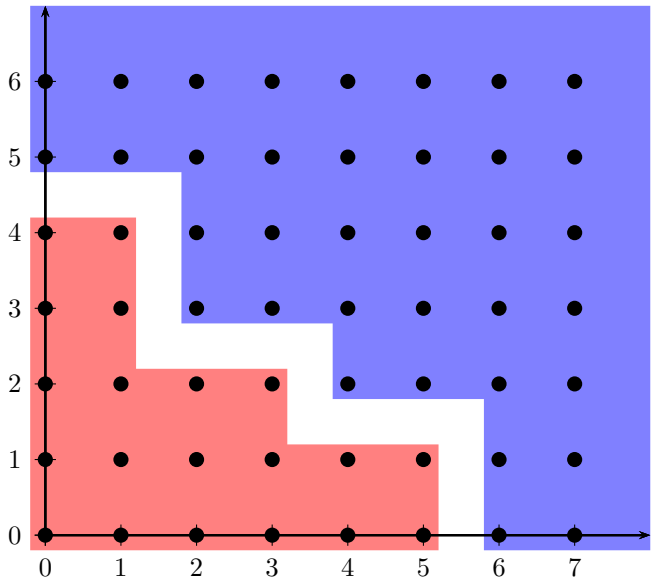


Abbildung 3.1.: Graphische Darstellung des monischen Ideals $\langle x_1^6, x_1^4 x_2^2, x_1^2 x_2^3, x_2^5 \rangle$. Die Punkte des ganzzahligen Gitters stellen dabei die Terme $x_1^{\alpha_1} x_2^{\alpha_2}$ dar. Das Ideal (blauer Bereich) ist die Vereinigung der von den Basistermen aufgespannten Kegeln – der rot markierte Bereich ist ein Ordnungsideal. Die Abbildung ist angelehnt an die Darstellungen in [CLO07].

Thema sei auf [KR05] verwiesen. Für *nichtmonische* Ideale ist eine Darstellung wie in Abbildung 3.1 natürlich nicht möglich. Dennoch sind auch diese Ideale endlich erzeugt. Dies besagt der *Basissatz von Hilbert*.

Satz 3.5 (Basissatz von Hilbert)**.** *Jedes Ideal von Π_d hat eine endliche Basis.*

Einen Beweis dieses Satzes findet man beispielsweise in [CLO07, Kapitel 2, §5, Theorem 4]. Damit kann nun jedes Ideal von Π_d durch endliche Information beschrieben werden, was eine Darstellung von Idealen im endlichen Speicher eines Computersystems ermöglicht. Im Folgenden gehen wir daher stets von endlichen Basen bzw. endlichen Erzeugnissen im Sinne von (3.1) aus. Zudem können Verknüpfungen von Idealen, wie die im Folgenden definierten Summen und Produkte, auf endlich viele Basispolynome zurückgeführt werden. Der folgende Satz stellt einige solcher Verknüpfungen von Idealen zusammen, die selbst auch wieder Ideale sind. Die einzelnen Aussagen sind dabei aus [CLO07, Kapitel 4, §3] entnommen.

Satz 3.6. *Seien $\mathcal{F} = \langle F \rangle$, $\mathcal{G} = \langle G \rangle$ Ideale von Π_d, $F, G \subset \Pi_d$ endlich, dann sind auch*

1. $\mathcal{F} + \mathcal{G} := \{f + g : f \in \mathcal{F}, g \in \mathcal{G}\} = \langle F \cup G \rangle$,
2. $\mathcal{F} \cdot \mathcal{G} := \{f_1 g_1 + \cdots + f_r g_r : f_j \in \mathcal{F}, g_j \in \mathcal{G}, r \in \mathbb{N}_0\} = \langle fg : f \in F, g \in G \rangle$,
3. $\mathcal{F} \cap \mathcal{G} := \{f : f \in \mathcal{F} \wedge f \in \mathcal{G}\}$

Ideale von Π_d.

Die Definition der Summe zweier Ideale ermöglicht auch die Darstellung eines Ideals als Summe der von den einzelnen Basispolynomen aufgespannten Hauptideale: Sei $F = \{f_1, \ldots, f_r\} \subset \Pi_d$, dann gilt $\langle F \rangle = \langle f_1 \rangle + \cdots + \langle f_r \rangle$. Abschließend wird noch eine spezielle Klasse von Idealen definiert, die insbesondere für den Zusammenhang zwischen Idealen und Varietäten von Bedeutung ist.

Definition 3.7. *Ein Ideal $\mathcal{F}$ heißt* radikales Ideal*, wenn gilt*

$$f^m \in \mathcal{F},\ m \geq 1 \quad \Longrightarrow \quad f \in \mathcal{F}. \tag{3.3}$$

Offensichtlich wird die starke Forderung (3.3) von vielen Idealen nicht erfüllt. So ist z. B. $\langle x^2 \rangle$ *kein* radikales Ideal, da $x \notin \langle x^2 \rangle$ gilt. Allerdings kann jedes Ideal um diese Eigenschaft erweitert werden. Dies führt zur Definition des *Radikals*:

Definition 3.8. *Sei $\mathcal{F}$ ein Ideal von Π_d, dann ist das* Radikal *von $\mathcal{F}$ definiert durch*

$$\sqrt{\mathcal{F}} := \{f : f^m \in \mathcal{F}, m \geq 1\}.$$

Dabei ist die konstruierte Menge $\sqrt{\mathcal{F}}$ ein radikales Ideal. Zum Beweis dieser Eigenschaft sei auf [CLO07, Kapitel 4, §2, Lemma 5] verwiesen.

3.2. Darstellung von Idealen

Eine wichtige Frage im Zusammenhang mit Idealen ist die Entscheidung, ob für ein Polynom $f \in \Pi_d$ und ein Ideal $\mathcal{F} \subset \Pi_d$ die Beziehung $f \in \mathcal{F}$ gilt. Man bezeichnet

dies auch als das *Ideal-Membership-Problem*. Ein bekanntes Verfahren zur Lösung dieses Problems verwendet *Gröbnerbasen*. Diese wurden 1965 von B. Buchberger als Hilfsmittel zur Basiskonstruktion in Restklassenringen (siehe [Buc65]) entwickelt und nach W. Gröbner, dem Doktorvater von Buchberger, benannt. Weitere Details dazu finden sich unter anderem in den Büchern von Becker und Weispfenning [BW93], Cox, Little und O'Shea [CLO07] bzw. [CLO98] oder Kreuzer und Robbiano [KR00].

Mittlerweile haben sich Gröbnerbasen zu einem wichtigen Werkzeug in vielen Computeralgebrasystemen entwickelt, da sie unter anderem die Lösung polynomieller Gleichungssysteme ermöglichen. Allerdings sind Gröbnerbasen – aufgrund ihrer Abhängigkeit von einer Termordnung – numerisch instabil, vgl. [MS00b]. Verwendet man stattdessen die bereits 1916 von Macaulay in [Mac16] eingeführten *H-Basen*, so erhält man in numerischer Rechnung deutlich bessere Ergebnisse. Auch dies wurde von Möller und Sauer in [MS00b] gezeigt.

H-Basen unterscheiden sich von Gröbnerbasen im Wesentlichen darin, dass sie keine Termordnung benötigen. Stattdessen werden Polynome nach ihrem *Totalgrad* klassifiziert, was auch die Bezeichnung *H-Basis* als *homogene* Basis motiviert.

Definition 3.9. *Eine endliche Menge $F \subset \Pi_d$ heißt* H-Basis *eines Ideals $\mathcal{F} \subset \Pi_d$, falls $\langle F \rangle = \mathcal{F}$ und sich jedes Polynom des Ideals schreiben lässt als*

$$\mathcal{F} \ni g = \sum_{f \in F} g_f f, \quad \deg(g) \geq \deg(g_f) + \deg(f),\ g_f \in \Pi_d. \tag{3.4}$$

Die Forderung an den Totalgrad in (3.4) sorgt dabei für eine *redundanzfreie* Darstellung der Polynome in $\mathcal{F}$ durch die Basispolynome in dem Sinne, dass keine Gradreduktion durch Auslöschung der Leitform auftreten kann. Genau diese Forderung zeichnet eine H-Basis aus, weshalb wir im Folgenden auch von der *H-Basis Eigenschaft* sprechen. Ersetzen wir den Totalgrad in (3.4) durch eine beliebige, auf einer Termordnung basierende Graduierung nach $\mathbb{N}_0^d$, so erhalten wir die Definition einer Gröbnerbasis. Anstelle von weiterführenden Untersuchungen sei für nähere Informationen zum Thema Gröbnerbasen erneut auf die oben zitierten Werke verwiesen.

Es gibt viele äquivalente Beschreibungen von H-Basen durch wichtige charakterisierende Eigenschaften. Der folgende Satz stellt einige dieser Eigenschaften zusammen, die von Möller und Sauer in [MS00a] und [MS00b] bzw. von Sauer in [Sau01] und [Sau07] gezeigt wurden und teilweise auf Macaulay zurückgehen, vgl. [Mac16]:

Satz 3.10. *Sei $F = \{f_1, \dots, f_s\} \subset \Pi_d$ eine Menge von Polynomen und $\mathcal{F} = \langle F \rangle \subseteq \Pi_d$ ein Ideal, dann sind folgende Aussagen äquivalent:*

1. *F ist eine H-Basis des Ideals $\mathcal{F}$.*
2. *$\langle \Lambda(f) : f \in \mathcal{F} \rangle =: \langle \Lambda(\mathcal{F}) \rangle = \langle \Lambda(F) \rangle := \langle \Lambda(f_1), \dots, \Lambda(f_s) \rangle$.*
3. *$\langle \Lambda(F) \rangle$ ist das kleinste Ideal, das alle $\Lambda(f)$, $0 \neq f \in \mathcal{F}$ enthält.*
4. *$\Lambda(\mathcal{F}) \cap \Pi_{k,d}^0 = \mathcal{V}_{k,d}^0(F)$ für alle $k \in \mathbb{N}_0$.*
5. *Die Division eines Polynoms $g \in \Pi_d$ durch F mittels Algorithmus 2.49 liefert den Rest 0 genau dann, wenn $g \in \mathcal{F}$.*

An dieser Stelle ist es wichtig, zwischen $\Lambda(\mathcal{F})$ und $\langle \Lambda(\mathcal{F}) \rangle$ zu unterscheiden. Es gilt $\Lambda(\mathcal{F}) \subseteq \Pi_d^0$, d. h. diese Menge enthält tatsächlich nur homogene Polynome. Dies gilt jedoch nicht mehr für $\langle \Lambda(\mathcal{F}) \rangle$, da bereits die Summe zweier homogener Polynome verschiedenen Grades nicht mehr homogen ist. Insbesondere ist $\Lambda(\mathcal{F})$ im Allgemeinen *kein* Ideal.

Die letzte Aussage in Satz 3.10 ermöglicht die Lösung des *Ideal-Membership-Problems*: Ist eine H-Basis F eines Ideals gegeben, so lässt sich algorithmisch entscheiden, ob ein Polynom in diesem Ideal liegt oder nicht. Im negativen Fall erhält man sogar noch mehr Informationen: Algorithmus 2.49 liefert einen Rest $0 \neq r \in \mathcal{W}_d(F)$, der genau den Teil beschreibt, der dem Polynom g fehlt, um im Ideal zu liegen. Ist F eine H-Basis und $g = \sum_{f \in F} g_f f + r$, so gilt stets $(g - r) \in \langle F \rangle$.

Eine weitere wichtige Eigenschaft ist die Eindeutigkeit dieses Rests für beliebige H-Basen eines Ideals. Zum Beweis dieser Eindeutigkeit sei auf die Arbeit von Möller und Sauer in [MS00a] bzw. von Sauer in [Sau07] verwiesen. Für Gröbnerbasen und einen entsprechenden Divisionsalgorithmus gilt ein analoges Resultat, vgl. [CLO07, Kapitel 2, §6, Proposition 1].

Definition 3.11. *Sei $F \subset \Pi_d$ eine H-Basis des Ideals $\mathcal{F}$. Dann hat jedes Polynom $g \in \Pi_d$, $k = \deg(g)$, eine* H-Darstellung

$$g = \sum_{f \in F \cap \Pi_{k,d}} g_f f + \nu_{\mathcal{F}}(g), \quad g_f \in \Pi_{k-\deg(f),d},\ \nu_{\mathcal{F}}(g) \in \mathcal{W}_{k,d}(F).$$

Das Polynom $\nu_{\mathcal{F}}(g)$ wird auch als Normalform *von f bzgl. des Ideals $\mathcal{F}$ bezeichnet.*

Die Normalform eines Polynoms hängt aufgrund der oben beschriebenen Eindeutigkeit nur vom Ideal und nicht von der gewählten H-Basis ab. Dank dieser Unabhängigkeit können wir die Menge aller Normalformen $\nu_{\mathcal{F}}(g)$, $g \in \Pi_d$, eines Ideals $\mathcal{F}$ angeben.

Definition 3.12. *Sei $\mathcal{F} \subset \Pi_d$ ein Ideal, dann wird der Π_d-Vektorraum $\Pi_d/\mathcal{F}$ als* Normalformenraum *des Ideals $\mathcal{F}$ bezeichnet. Das Ideal heißt* nulldimensional*, falls der zugehörige Normalformenraum endlichdimensional ist.*

Der folgende Satz liefert weitere äquivalente Beschreibungen eines nulldimensionalen Ideals. Die Aussage ist in ähnlicher Formulierung in [CLO07, Kapitel 5, §3, Theorem 6] nachzulesen.

Satz 3.13. *Sei $\mathcal{F} \subset \Pi_d$ ein Ideal, dann sind folgende Aussagen äquivalent:*

1. *$\Pi_d/\mathcal{F}$ ist endlichdimensional.*
2. *$\operatorname{span}\{f \in \Pi_d : f \notin \langle \Lambda(\mathcal{F}) \rangle\}$ ist endlichdimensional.*
3. *Für jedes i, $1 \leq i \leq d$, gibt es ein $m_i \geq 0$ mit $x_i^{m_i} \in \langle \Lambda(\mathcal{F}) \rangle$.*

Zum Beweis sei ebenfalls auf [CLO07] verwiesen. Da die Menge $\langle \Lambda(\mathcal{F}) \rangle$ in der Regel nicht bekannt ist, sind diese Kriterien in der Praxis schwer nachprüfbar. Ist jedoch eine H-Basis F des Ideals $\mathcal{F}$ gegeben, so gilt nach Satz 3.10 die Beziehung $\langle \Lambda(\mathcal{F}) \rangle = \langle \Lambda(F) \rangle = \mathcal{V}_d(F)$. In diesem Fall ist das Ideal genau dann nulldimensional, wenn der Raum $\mathcal{W}_d(F)$ endlichdimensional ist.

Beispiel 3.14. *Sei $F = \{2x_1 - x_2 + 1, x_1^3 - x_1\}$ eine H-Basis des Ideals $\mathcal{F} \subset \Pi_2$. Wir bestimmen die Mengen $\mathcal{V}_{k,2}^0(F)$ und $\mathcal{W}_{k,2}^0(F)$:*

- $\mathcal{V}^0_{0,2}(F) = \operatorname{span}\{\emptyset\} = \{0\}$, $\mathcal{W}^0_{0,2}(F) = \operatorname{span}\{1\}$
- $\mathcal{V}^0_{1,2}(F) = \operatorname{span}\{2x_1 - x_2\}$, $\mathcal{W}^0_{1,2}(F) = \operatorname{span}\{x_1 + 2x_2\}$
- $\mathcal{V}^0_{2,2}(F) = \operatorname{span}\{2x_1^2 - x_1x_2, 2x_1x_2 - x_2^2\}$, $\mathcal{W}^0_{2,2}(F) = \operatorname{span}\{x_1^2 + 2x_1x_2 + 4x_2^2\}$
- $\mathcal{V}^0_{3,2}(F) = \operatorname{span}\{2x_1^3 - x_1^2x_2, 2x_1^2x_2 - x_1x_2^2, 2x_1x_2^2 - x_2^3, x_1^3\}$
  $= \operatorname{span}\{x_1^{\alpha_1}x_2^{\alpha_2} : \alpha \in \mathbb{N}_0^2, |\alpha| = 3\} = \Pi^0_{3,2}$, $\mathcal{W}^0_{3,2}(F) = \operatorname{span}\{\emptyset\} = \{0\}$

Damit ist $\mathcal{W}^0_{k,2}(F) = \{0\}$ *für* $k \geq 3$ *und es gilt* $\dim(\mathcal{W}_2(F)) = \operatorname{span}\{f \in \Pi_d : f \notin \langle\Lambda(\mathcal{F})\rangle\} = 3 < \infty$ *und Satz 3.13 liefert die Nulldimensionalität des Ideals* $\mathcal{F}$*.*

Man kann also die Nulldimensionalität eines Ideals mit Hilfe von H-Basen überprüfen. Umgekehrt lässt sich für nulldimensionale Ideale auch ein konstruktives Verfahren zur Bestimmung einer H-Basis angeben. Ein solcher Algorithmus wurde von Möller und Sauer in [MS00b] vorgestellt. Dabei wird im Wesentlichen die folgende Eigenschaft einer H-Basis ausgenutzt:

Satz 3.15. *Sei* $\mathcal{F} = \langle F\rangle \subseteq \Pi_d$ *ein nulldimensionales Ideal. Die Menge* F *ist genau dann eine H-Basis von* $\mathcal{F}$*, wenn ein* $k \in \mathbb{N}_0$ *existiert, sodass* $\mathcal{W}^0_{k,d}(F) = \{0\}$*.*

Beweis. Nach Satz 3.10 ist F genau dann eine H-Basis, wenn $\mathcal{V}^0_{k,d}(F) = \Lambda(\mathcal{F}) \cap \Pi^0_{k,d}$ für alle $k \in \mathbb{N}_0$ gilt. Da $\mathcal{F}$ nulldimensional ist, gibt es nach Satz 3.13 ein $\widetilde{k} \in \mathbb{N}_0$, sodass

$$\Pi^0_{\widetilde{k},d} = \Lambda(\mathcal{F}) \cap \Pi^0_{\widetilde{k},d} = \mathcal{V}^0_{\widetilde{k},d}(F).$$

Dies ist jedoch äquivalent zu $\mathcal{W}^0_{\widetilde{k},d}(F) = \operatorname{span}\{\emptyset\} = \{0\}$. □

Das Verfahren von Möller und Sauer ist in Algorithmus 3.16 beschrieben. Da die Abbruchbedingung $\mathcal{W}^0_{k,d}(F) = \{0\}$ nur für nulldimensionale Ideale erfüllt ist, kann die Terminierung des Algorithmus auch als Kriterium für die Nulldimensionalität eines Ideals interpretiert werden.

Algorithmus 3.16 fügt einer vorgegebenen Basis lediglich Polynome hinzu. Dabei entstehen möglicherweise auch Redundanzen, die die Forderung an den Grad in (3.4), d. h. die H-Basis Eigenschaft, respektieren. Diese können durch Reduktion eines jeden Polynoms $f \in F$ modulo $F \setminus \{f\}$ mit Hilfe des Divisionsalgorithmus

Algorithmus 3.16 : Bestimmung einer H-Basis

Input : Basis $F \subset \Pi_d$ eines nulldimensionalen Ideals $\mathcal{F}$
Output : H-Basis F des Ideals $\mathcal{F}$

```
1  k ← 0
2  while W_{k,d}(F) ≠ {0} do
3  |  V ← {Σ_{f∈F} g_f Λ(f) : g_f ∈ Π^0_{k−deg(f),d}}
4  |  Suche lineare Abhängigkeiten in V:
5  |  if Σ_{f∈F} g_f Λ(f) = 0 then
6  |  |  if Σ_{f∈F} g_f f ⇝_F f̃ ≠ 0 then
7  |  |  |  F ← F ∪ {f̃}
8  |  |  |  k ← −1
9  |  |  end
10 |  end
11 |  k ← k + 1
12 end
```

bestimmt werden: Gibt es Basispolynome, die sich zu 0 reduzieren lassen, in Zeichen $f \rightsquigarrow_{F\setminus\{f\}} 0$, so sind diese redundant und können aus der Basis gestrichen werden. Lässt sich auf diese Art kein Polynom mehr entfernen, so sprechen wir von einer *reduzierten H-Basis*. Das folgende Beispiel zeigt die Anwendung von Algorithmus 3.16 mit anschließender Reduktion der H-Basis:

Beispiel 3.17. *Es sei ein Ideal $\mathcal{F} \subset \Pi_3$ gegeben durch $\mathcal{F} = \langle x_1^2 - x_2, x_2x_3 + 4x_1 - 5x_3, x_1^3 - x_3\rangle$. Wir wenden Algorithmus 3.16 an, um eine H-Basis des Ideals zu bestimmen.*

1. $F^{(0)} := \{x_1^2 - x_2, x_2x_3 + 4x_1 - 5x_3, x_1^3 - x_3\}$

   $\mathcal{V}^0_{0,3}(F^{(0)}) = \mathcal{V}^0_{1,3}(F^{(0)}) = \operatorname{span}\{\emptyset\} = \{0\}$
   $\mathcal{V}^0_{2,3}(F^{(0)}) = \operatorname{span}\{x_1^2, x_2x_3\}$
   $\mathcal{V}^0_{3,3}(F^{(0)}) = \operatorname{span}\{x_1^3, x_1^2x_2, x_1^2x_3, x_1x_2x_3, x_2^2x_3, x_2x_3^2, x_1^3\}$

   Hier tritt mit x_1^3 erstmals eine lineare Abhängigkeit auf, die durch $x_1\Lambda(x_1^2 - x_2) - \Lambda(x_1^3 - x_3) = 0$ beschrieben werden kann. Demnach müssen wir

$$x_1(x_1^2 - x_2) - (x_1^3 - x_3) = -x_1x_2 + x_3,$$

berechnen und erhalten ein von Null verschiedenes Polynom, das sich nicht weiter reduzieren lässt. Wir fügen dieses Polynom der Menge $F^{(0)}$ hinzu.

2. $F^{(1)} := F^{(0)} \cup \{x_1x_2 - x_3\} = \{x_1^2 - x_2, x_1x_2 - x_3, x_2x_3 + 4x_1 - 5x_3, x_1^3 - x_3\}$

$$\begin{aligned}
\mathcal{V}^0_{0,3}(F^{(1)}) &= \mathcal{V}^0_{1,3}(F^{(1)}) = \operatorname{span}\{\emptyset\} = \{0\}\\
\mathcal{V}^0_{2,3}(F^{(1)}) &= \operatorname{span}\{x_1^2, x_1x_2, x_2x_3\}\\
\mathcal{V}^0_{3,3}(F^{(1)}) &= \operatorname{span}\{x_1^3, x_1^2x_2, x_1^2x_3, x_1^2x_2, x_1x_2^2, x_1x_2x_3, x_1x_2x_3, x_2^2x_3, x_2x_3^2, x_1^3\}
\end{aligned}$$

Nun sind zwei weitere linear abhängige Paare enthalten: Es gilt $x_2\Lambda(x_1^2 - x_2) - x_1\Lambda(x_1x_2 - x_3) = 0$, sowie $x_3\Lambda(x_1x_2 - x_3) - x\Lambda(x_2x_3 + 4x_1 - 5x_3) = 0$. Damit erhalten wir

$$x_2(x_1^2 - x_2) - x_1(x_1x_2 - x_3) = x_1x_3 - x_2^2$$

und

$$x_3(x_1x_2 - x_3) - x_1(x_2x_3 + 4x_1 - 5x_3) = -4x_1^2 + 5x_1x_3 - x_3^2 = 5x_1x_3 - x_3^2 - 4x_2.$$

Im letzten Schritt erfolgte dabei eine Reduktion des Monoms $-4x_1^2$ durch das Basispolynom $x_1^2 - x_2$. Da sich beide Polynome nicht weiter reduzieren lassen, fügen wir sie der Menge $F^{(1)}$ hinzu.

3. $$\begin{aligned}
F^{(2)} &:= F^{(1)} \cup \{x_1x_3 - x_2^2, 5x_1x_3 - x_3^2 - 4x_2\}\\
&= \{x_1^2 - x_2, x_1x_2 - x_3, x_1x_3 - x_2^2, 5x_1x_3 - x_3^2 - 4x_2, x_2x_3 + 4x_1 - 5x_3, x_1^3 - x_3\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{V}^0_{0,3}(F^{(2)}) &= \mathcal{V}^0_{1,3}(F^{(1)}) = \operatorname{span}\{\emptyset\} = \{0\}\\
\mathcal{V}^0_{2,3}(F^{(2)}) &= \operatorname{span}\{x_1^2, x_1x_2, x_1x_3 - x_2^2, 5x_1x_3 - x_3^2, x_2x_3\}\\
\mathcal{V}^0_{3,3}(F^{(2)}) &= \operatorname{span}\{x_1^3, x_1^2x_2, x_1^2x_3, x_1^2x_2, x_1x_2^2, x_1x_2x_3, x_1^2x_3 - x_1x_2^2, x_1x_2x_3 - x_2^3,\\
&\qquad x_1x_3^2 - x_2^2x_3, 5x_1^2x_3 - x_1x_3^2, 5x_1x_2x_3 - x_2x_3^2,\\
&\qquad 5x_1x_3^2 - x_3^3, x_1x_2x_3, x_2^2x_3, x_2x_3^2, x_1^3\}\\
&= \operatorname{span}\{x_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3} : \alpha \in \mathbb{N}_0^3, |\alpha| = 3\}.
\end{aligned}$$

An dieser Stelle bricht der Algorithmus ab und liefert $F^{(2)}$ als H-Basis des Ideals $\mathcal{F}$. Da $\mathcal{W}^0_{3,3}(F^{(2)}) = \{0\}$ gilt, wissen wir nach Satz 3.13 ebenfalls, dass $\mathcal{F}$ ein nulldimensionales Ideal ist.

Nun prüfen wir noch, ob eine Reduktion der Polynome $F^{(2)}$ möglich ist. Durch die Darstellung $x_1^3 - x_3 = x_1(x_1^2 - x_2) + (x_1x_2 - x_3)$ ist das Polynom $x_1^3 - x_3$ in der H-Basis überflüssig und kann entfernt werden. Wir erhalten somit die reduzierte H-Basis

$$\mathcal{F} = \left\langle x_1^2 - x_2, x_1x_2 - x_3, x_1x_3 - x_2^2, 5x_1x_3 - x_3^2 - 4x_2, x_2x_3 - 5x_3 + 4x_1 \right\rangle.$$

Wenn wir im Folgenden von einer H-Basis sprechen, so gehen wir stets von einer *reduzierten H-Basis* aus, d. h. einer H-Basis, aus der wir kein Basispolynom entfernen können, ohne das erzeugte Ideal zu verändern oder die H-Basis Eigenschaft zu verletzen. Eine weitere äquivalente Beschreibung ist durch den folgenden Satz gegeben:

Satz 3.18. *Sei $F \subset \Pi_d$ eine H-Basis. F ist genau dann* reduziert*, wenn $\Lambda(F) \cap \Pi_{k,d}^0 \subseteq \mathcal{W}_{k,d}^0(F \cap \Pi_{k-1,d})$ für alle $k \in \mathbb{N}$ gilt und die Mengen $\Lambda(F) \cap \Pi_{k,d}^0$, $k \in \mathbb{N}_0$, jeweils linear unabhängig sind.*

Beweis. Sei F eine reduzierte H-Basis. Angenommen, es gibt eine linear abhängige Menge $\Lambda(F) \cap \Pi_{k,d}^0$, $k \in \mathbb{N}_0$, dann kann aus dem Aufspann dieser Menge ein Polynom $\widetilde{f}$ mit $\deg(\widetilde{f}) < k$ konstruiert werden. Gilt dabei $\widetilde{f} \rightsquigarrow_F \hat{f} \neq 0$, d. h. lässt sich $\widetilde{f}$ nicht bzgl. F zum Nullpolynom reduzieren, so erhalten wir einen Widerspruch zur H-Basis Eigenschaft von F. Andererseits widerspricht $\widetilde{f} \rightsquigarrow_F 0$ der Annahme, dass F reduziert ist. Die Argumentation gilt analog, falls $\Lambda(F) \cap \Pi_{k,d}^0 \subseteq \mathcal{W}_{k,d}^0(F \cap \Pi_{k-1,d})$ nicht erfüllt ist.

Sei umgekehrt F eine H-Basis, $\Lambda(F) \cap \Pi_{k,d}^0 \subseteq \mathcal{W}_{k,d}^0(F \cap \Pi_{k-1,d})$ für alle $k \in \mathbb{N}$ und die Mengen $F \cap \Pi_{k,d}^0$, $k \in \mathbb{N}_0$, jeweils linear unabhängig. Dann folgt $\Lambda(f) \in \mathcal{W}_{\deg(f),d}^0(F \setminus \{f\})$ für alle $f \in F$, was bedeutet, dass sich F nicht weiter reduzieren lässt. □

Es ist bekannt, dass jedes Polynomideal eine bzgl. einer festen Termordnung eindeutige reduzierte Gröbnerbasis besitzt, vgl. [CLO07, Kapitel 2, §7, Proposition 6]. Diese Eindeutigkeit ist für H-Basen nicht gegeben. Dennoch erhält man durch die Berechnung einer reduzierten H-Basis eine invariante Eigenschaft des Ideals: die Anzahl der Basispolynome eines bestimmten Grades. Damit ist sichergestellt, dass ein

nulldimensionales Ideal keine einfachere Darstellung im Sinne eines kleineren Totalgrades durch eine andere H-Basis hat. Der folgende Satz zeigt diese Eigenschaft:

Satz 3.19. *Seien $\mathcal{F} \subset \Pi_d$ ein Ideal und $F_1 \subseteq \mathcal{F}$, $F_2 \subseteq \mathcal{F}$ zwei reduzierte H-Basen von $\mathcal{F}$. Dann gilt*

$$\#\left(F_1 \cap (\Pi_{k,d} \setminus \Pi_{k-1,d})\right) = \#\left(F_2 \cap (\Pi_{k,d} \setminus \Pi_{k-1,d})\right), \quad k \in \mathbb{N}_0. \tag{3.5}$$

Mit anderen Worten: Die Anzahl der Basispolynome vom Grad k hängt nur vom Ideal und nicht von der gewählten H-Basis ab.

Beweis. Angenommen, die Behauptung ist falsch. Dann gibt es einen kleinsten Wert $\tilde{k} \in \mathbb{N}_0$, für den (3.5) nicht erfüllt ist. Wir können also annehmen, dass

$$\#\left(F_1 \cap (\Pi_{\tilde{k},d} \setminus \Pi_{\tilde{k}-1,d})\right) < \#\left(F_2 \cap (\Pi_{\tilde{k},d} \setminus \Pi_{\tilde{k}-1,d})\right).$$

Da F_1 und F_2 reduzierte H-Basen sind, folgt daraus $\dim(\mathcal{V}^0_{\tilde{k},d}(F_1)) < \dim(\mathcal{V}^0_{\tilde{k},d}(F_2))$. Andererseits gilt nach Satz 3.10 für alle H-Basen

$$\mathcal{V}^0_{\tilde{k},d}(F_1) = \Lambda(\mathcal{F}) \cap \Pi^0_{\tilde{k},d} = \mathcal{V}^0_{\tilde{k},d}(F_2),$$

was zu einem Widerspruch führt. Damit muss (3.5) für alle $k \in \mathbb{N}_0$ richtig sein. □

Dieses Resultat wurde auch von Möller und Sauer in [MS00a] gezeigt. Ebenso findet man dort die Aussage, dass die Anzahl aller Basispolynome zweier reduzierter H-Basen desselben Ideals übereinstimmen, was eine direkte Konsequenz aus Satz 3.19 ist, und die Koeffizientenmatrizen dieser Basispolynome durch Multiplikation mit einer orthogonalen Matrix ineinander überführt werden können, vgl. [MS00a, Theorem 6.5]. Mit diesem Wissen lassen sich beliebige reduzierte H-Basen eines nulldimensionalen Ideals aus einer speziellen reduzierten H-Basis konstruieren. Dazu müssen folgende Bedingungen beachtet werden:

1. Es dürfen nur Polynome gleichen Grades gegeneinander ausgetauscht werden.
2. Die Räume $\mathcal{V}^0_{k,d}(F)$ müssen erhalten bleiben.

In Abschnitt 4.4 werden wir genauer auf diese beiden Punkte eingehen und dementsprechende Konstruktionen explizit angeben. Diese können wir dann ausnutzen, um *dünn besetzte H-Basen*, d. h. Basispolynome mit möglichst wenigen von Null verschiedenen Koeffizienten, zu bestimmen.

3.3. Varietäten und der Nullstellensatz

In diesem Abschnitt wird der Zusammenhang zwischen *Polynomidealen* und *Nullstellen* dargestellt, wobei sich die Ausführungen im Wesentlichen an [CLO07] orientieren. Wir beginnen mit der Definition des zentralen Begriffs dieses Abschnitts, der die Menge aller gemeinsamen Nullstellen der Polynome $F \subset \Pi_d$ beschreibt.

Definition 3.20. *Zu einer Menge von Polynomen $F \subset \Pi_d$ definieren wir die* Varietät *von F durch*

$$\mathfrak{V}(F) := \left\{ \xi \in \mathbb{C}^d : f(\xi) = 0, f \in F \right\} \subseteq \mathbb{C}^d.$$

Kennt man die Varietät einer Menge von Polynomen $F \subset \Pi_d$, so kennt man auch die Varietät des von F aufgespannten Ideals: Für jedes Polynom $g \in \langle F \rangle$ gilt an einer Stelle $\xi \in \mathfrak{V}(F)$ die Auswertung

$$\langle F \rangle \ni g(\xi) = \sum_{f \in F} g_f(\xi) \cdot f(\xi) = \sum_{f \in F} g_f(\xi) \cdot 0 = 0,$$

was bereits $\mathfrak{V}(F) \subseteq \mathfrak{V}(\langle F \rangle)$ impliziert. Die umgekehrte Inklusion ist trivial, da $F \subseteq \langle F \rangle$ gilt. Also stimmen die Varietäten von F und $\langle F \rangle$ überein, vgl. [CLO07, Kapitel 2, §5, Proposition 9]. Damit können wir uns zur Bestimmung der Varietät eines Ideals auf die Varietät einer *endlichen* Basis beschränken. Dabei ist stets der algebraisch abgeschlossene Körper $\mathbb{C}$ zu betrachten, um sicherzustellen, dass alle Nullstellen erfasst werden. Umgekehrt lässt sich zu jeder Punktmenge $\Xi \subseteq \mathbb{C}^d$ eine Menge von Polynomen finden, die an diesen Punkten verschwinden. Da die Punkte in $\mathbb{C}^d$ liegen, muss dabei auch der Polynomring $\overline{\Pi}_d := \mathbb{C}[x_1, \ldots, x_d]$ zugrunde gelegt werden.

Satz 3.21. *Zu jeder Menge $\Xi \subseteq \mathbb{C}^d$ bilden die Polynome*

$$\mathfrak{I}(\Xi) := \left\{ f \in \overline{\Pi}_d : f(\xi) = 0, \xi \in \Xi \right\}$$

ein Ideal *von $\overline{\Pi}_d$. Wir sprechen dabei auch von dem* Verschwindungsideal *zu Ξ.*

Beweis. Da das Nullpolynom $0 \in \overline{\Pi}_d$ an allen Punkten verschwindet, gilt $0 \in \mathfrak{I}(\Xi)$. Seien $f, g \in \mathfrak{I}(\Xi)$, $h \in \overline{\Pi}_d$ und $\xi \in \Xi$. Es folgt $(f+g)(\xi) = f(\xi) + g(\xi) = 0 + 0 = 0$ und $(h \cdot f)(\xi) = h(\xi) \cdot f(\xi) = h(\xi) \cdot 0 = 0$, also $(f+g) \in \mathfrak{I}(\Xi)$ und $(h \cdot f) \in \mathfrak{I}(\Xi)$. Damit ist $\mathfrak{I}(\Xi)$ ein Ideal von $\overline{\Pi}_d$. □

Die Abbildungen $\mathfrak{I}$ und $\mathfrak{V}$ erlauben also einen Übergang von Varietäten zu Polynomidealen und umgekehrt. Es gilt sogar folgender Zusammenhang, vgl. [CLO07, Kapitel 4, §2, Theorem 7]:

Satz 3.22. *Die Abbildungen $\mathfrak{I}$ und $\mathfrak{V}$ sind inklusionsumkehrend und für eine Varietät $\Xi \subseteq \mathbb{C}^d$ gilt $\mathfrak{V}(\mathfrak{I}(\Xi)) = \Xi$.*

Eine analoge Beziehung zwischen einem Ideal $\mathcal{F}$ und $\mathfrak{I}(\mathfrak{V}(\mathcal{F}))$ gilt jedoch *nicht*, denn im Fall $d = 1$ stimmen zwar die Varietäten $\mathfrak{V}(\langle x \rangle) = \{0\} = \mathfrak{V}(\langle x^2 \rangle)$ überein, aber man erhält $\mathfrak{I}(\mathfrak{V}(\langle x^2 \rangle)) = \langle x \rangle \neq \langle x^2 \rangle$. Durch den Übergang zur Varietät gehen Informationen über die Vielfachheit der Nullstellen verloren. Berücksichtigt man dies auch auf der Idealseite, so ergibt sich das folgende Resultat aus [CLO07, Kapitel 4, §2, Lemma 1]:

Lemma 3.23. *Sei $\Xi \subseteq \mathbb{C}^d$ und $f^m \in \mathfrak{I}(\Xi)$ für ein $m \geq 1$, dann gilt auch $f \in \mathfrak{I}(\Xi)$.*

Beweis. Sei $\xi \in \Xi$. Ist nun $f^m \in \mathfrak{I}(\Xi)$, dann erhalten wir $(f(\xi))^m = 0$. Dies impliziert aber $f(\xi) = 0$ und damit gilt $f \in \mathfrak{I}(\Xi)$. □

Diese Eigenschaft des Ideals $\mathfrak{I}(\mathfrak{V}(F))$ wurde bereits im letzten Abschnitt definiert: Für $\Xi \subseteq \mathbb{C}^d$ ist $\mathfrak{I}(\Xi)$ ein radikales Ideal. Insbesondere gilt der *Nullstellensatz von Hilbert*, der in [CLO07, Kapitel 4, §4, Theorem 6] nachzulesen ist.

Satz 3.24 (Nullstellensatz von Hilbert). *Sei $F \subset \overline{\Pi}_d$, dann gilt $\mathfrak{I}(\mathfrak{V}(F)) = \sqrt{\langle F \rangle}$.*

Damit erhalten wir für radikale Ideale, also $\langle F \rangle = \sqrt{\langle F \rangle}$, die gesuchte Umkehrung von Satz 3.22, was eine eindeutige Zuordnung zwischen radikalen Polynomidealen und Varietäten ermöglicht. Ebenso lassen sich die in Satz 3.6 definierten Idealverknüpfungen in den Varietätenkontext übertragen. Diese Dualität wird von Cox, Little und O'Shea in [CLO07] auch als das *Algebra-Geometry Dictionary* bezeichnet. Eine Zusammenfassung der Resultate über den Zusammenhang zwischen Verknüpfungen von Idealen und Verknüpfungen von Varietäten aus [CLO07, Kapitel 4, §3] wird in folgendem Satz gegeben:

Satz 3.25. *Seien $\mathcal{F}, \mathcal{G} \subseteq \Pi_d$ Ideale, dann gilt für die in Satz 3.6 definierten Ideale:*

1. $\mathfrak{V}(\mathcal{F} + \mathcal{G}) = \mathfrak{V}(\mathcal{F}) \cap \mathfrak{V}(\mathcal{G})$,
2. $\mathfrak{V}(\mathcal{F} \cdot \mathcal{G}) = \mathfrak{V}(\mathcal{F}) \cup \mathfrak{V}(\mathcal{G})$,
3. $\mathfrak{V}(\mathcal{F} \cap \mathcal{G}) = \mathfrak{V}(\mathcal{F}) \cup \mathfrak{V}(\mathcal{G})$.

Die erste Aussage dieses Satzes bedeutet insbesondere, dass sich die Varietät eines Ideals durch den Schnitt der Varietäten der einzelnen Basispolynome darstellen lässt, also $\mathfrak{V}(\mathcal{F}) = \bigcap_{j=1}^{n} \mathfrak{V}(f_j)$ für $\mathcal{F} = \langle f_1, \ldots, f_n \rangle$. Das folgende Beispiel nutzt diese Eigenschaft zur graphischen Bestimmung des reellen Anteils einer Varietät aus:

Beispiel 3.26. *Wir betrachten das Ideal aus Beispiel 3.14, verwenden hier jedoch aus Darstellungsgründen eine andere Basis: $\mathcal{F} = \langle 2x_1 - x_2 + 1, x_1^3 - x_1 \rangle = \langle 2x_1 - x_2 + 1, 2x_1^3 - x_2 + 1 \rangle$. Es gilt*

$$\mathfrak{V}(2x_1 - x_2 + 1) = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{C}^2 : x_2 = 2x_1 + 1 \right\},$$

sowie

$$\mathfrak{V}(2x_1^3 - x_2 + 1) = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{C}^2 : x_2 = 2x_1^3 + 1 \right\}.$$

Damit erhält man die gesuchte Varietät als Schnitt einer Geraden und einer Hyperbel (siehe Abbildung 3.2), denn es gilt

$$\mathfrak{V}(\mathcal{F}) = \mathfrak{V}(2x_1 - x_2 + 1) \cap \mathfrak{V}(2x_1^3 - x_2 + 1) = \left\{ \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix} \right\} \subset \mathbb{R}^2.$$

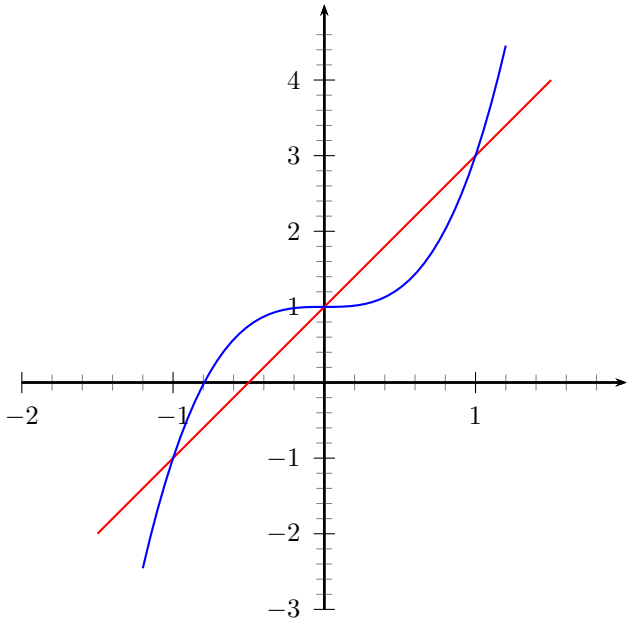


Abbildung 3.2.: Varietät des Ideals $\langle 2x_1 - x_2 + 1, 2x_1^3 - x_2 + 1\rangle$ als Schnittpunkte von einer Geraden und einer Hyperbel.

An dieser Stelle ist es kein Zufall, dass $\mathfrak{V}(\mathcal{F})$ nur aus endlich vielen Punkten besteht. Vielmehr ist dies eine Konsequenz der Nulldimensionalität von $\mathcal{F}$, vgl. Beispiel 3.14. Somit kann eine weitere äquivalente Bedingung für nulldimensionale Ideale im Sinne von Satz 3.13 formuliert werden.

Satz 3.27. *Die Varietät einer Menge $F \subset \overline{\Pi}_d$ ist genau dann endlich, wenn $\langle F\rangle$ ein nulldimensionales Ideal ist. Insbesondere ist die Mächtigkeit von $\mathfrak{V}(F)$ höchstens* $\dim(\overline{\Pi}_d/\langle F\rangle)$. *Ist $\langle F\rangle$ ein radikales Ideal, dann gilt sogar Gleichheit.*

Zum Beweis dieser Aussage sei auf [CLO07, Kapitel 5, §3] verwiesen. Beschränken wir uns in Satz 3.27 auf Polynome mit reellen Koeffizienten und den reellwertigen Anteil der Varietät, so gilt die Aussage im Allgemeinen nur als Implikation: Ist $\langle F\rangle \in \Pi_d$ ein nulldimensionales Ideal, so gilt $\#(\mathfrak{V}(F) \cap \mathbb{R}^d) < \infty$. Die Umkehrung wird im Allgemeinen jedoch falsch, wie das folgende Beispiel zeigt:

Beispiel 3.28. *Wir betrachten das Polynom $f = x_1^2 + x_2^2 \in \Pi_2$. Um zu entscheiden, ob $\langle f\rangle$ nulldimensional ist, verwenden wir Satz 3.13 und überprüfen die Dimension*

von $\mathcal{W}_2(f)$*. Wegen*

$$\dim(\mathcal{V}_{k,2}^0(f)) = \#\left\{x_1^{\alpha_1}x_2^{\alpha_2}(x_1^2+x_2^2) : |\alpha| = k-2\right\} = \binom{2+(k-2)-1}{k-2} = k-1$$

und $\dim(\Pi_{k,2}^0) = \binom{2+k-1}{k} = k+1$ *gilt* $\dim(\mathcal{W}_{k,d}^0) = 2$ *für alle* $k \geq 2$*. Damit ist*

$$\dim(\mathcal{W}_2(f)) = \sum_{k\in\mathbb{N}_0} \dim(\mathcal{W}_{k,2}^0(f)) = \infty$$

und das Ideal $\langle f\rangle$ *ist* nicht *nulldimensional. Betrachten wir nun*

$$\mathfrak{V}(f)\cap\mathbb{R}^2 = \left\{\begin{pmatrix}x_1\\x_2\end{pmatrix}\in\mathbb{R}^2 : x_2 = \sqrt{-x_1^2}\right\} = \left\{\begin{pmatrix}0\\0\end{pmatrix}\right\},$$

so deutet diese endliche Menge fälschlicherweise auf ein nulldimensionales Ideal hin. Erst die Betrachtung der Varietät in $\mathbb{C}^2$ *liefert durch*

$$\mathfrak{V}(f) = \left\{\begin{pmatrix}x_1\\x_2\end{pmatrix}\in\mathbb{C}^2 : x_2 = \sqrt{-x_1^2}\right\} = \left\{\begin{pmatrix}x_1\\ix_1\end{pmatrix}\in\mathbb{C}^2 : x_1\in\mathbb{C}\right\},$$

eine unendliche Menge und widerlegt somit die Nulldimensionalität.

Um wieder eine Äquivalenz in Satz 3.27 zu erhalten, können wir $F \subset \Pi_d$ so einschränken, dass $\mathfrak{V}(F) \subseteq \mathbb{R}^d$ gilt. Allerdings stellt nicht jede Menge $\Xi \subseteq \mathbb{R}^d$ eine Varietät dar. So ist beispielsweise

$$\Xi = \left\{t\cdot\begin{pmatrix}1\\1\end{pmatrix} : t\in\mathbb{R}\setminus\{0\}\right\}\subset\mathbb{R}^2,$$

keine Varietät, da jedes Polynom $f \in \overline{\Pi}_d$ mit $f(\Xi) = 0$ auch $f(0,0) = 0$ erfüllt. Die Menge $\Xi \cup \{(0,0)\}$ ist hingegen eine Varietät. Diese Erweiterung von Ξ bezeichnet man als den *Zariski-Abschluss*, den wir zunächst wieder für komplexwertige Mengen definieren.

Definition 3.29. *Zu einer Menge* $\Xi \subseteq \mathbb{C}^d$ *ist der* Zariski-Abschluss $\overline{\Xi}$ *die kleinste Varietät, die* Ξ *enthält.*

Der Zariski-Abschluss lässt sich mit Hilfe der Abbildungen $\mathfrak{V}$ und $\mathfrak{I}$ bestimmen. Für eine Menge $\Xi \subseteq \mathbb{C}^d$ gilt $\overline{\Xi} = \mathfrak{V}(\mathfrak{I}(\Xi))$, vgl. [CLO07, Kapitel 4, §3]. Außerdem sind alle endlichen Teilmengen von $\mathbb{C}^d$ bereits Zariski-abgeschlossen, wie der folgende Satz zeigt:

Satz 3.30. *Sei $\Xi \subset \mathbb{C}^d$ mit $\#\Xi < \infty$, so gilt $\overline{\Xi} = \Xi$.*

Beweis. Nach Definition des Zariski-Abschlusses gilt $\Xi \subseteq \overline{\Xi}$. Sei $\#\Xi = n$ und $\Xi = \{\xi^{(1)}, \ldots, \xi^{(n)}\}$. Für diese Punkte gilt $\mathfrak{I}(\xi^{(j)}) = \langle x_1 - \xi_1^{(j)}, \ldots, x_d - \xi_d^{(j)} \rangle$. Nun ist

$$\mathfrak{I}(\Xi) = \mathfrak{I}\left(\bigcup_{j=1}^{n} \{\xi^{(j)}\}\right) = \sqrt{\prod_{j=1}^{n} \mathfrak{I}(\xi^{(j)})} \supseteq \prod_{j=1}^{n} \mathfrak{I}(\xi^{(j)}).$$

Daher liegen alle Polynome $\prod_{j=1}^{n}(x_{i_j} - \xi_{i_j}^{(j)})$ mit $i_j \in \{1, \ldots, d\}$, also die Produkte von jeweils einem Linearfaktor aus $\mathfrak{I}(\xi^{(j)})$, $j = 1, \ldots, d$, im Ideal $\mathfrak{I}(\Xi)$. Diese Polynome haben jedoch nur $\xi \in \Xi$ als gemeinsame Nullstellen. Damit ist dann $\overline{\Xi} = \mathfrak{V}(\mathfrak{I}) \subseteq \Xi$. □

Insbesondere ist für *endliche reelle Punktmengen* $\Xi \subset \mathbb{R}^d$ die Untersuchung von *reellen Varietäten* hinreichend. Außerdem hat das nulldimensionale Ideal $\mathfrak{I}(\Xi)$ in diesem Fall eine Basis $F \subset \Pi_d$ mit *reellen Koeffizienten*. Da wir in der numerischen Rechnung ohnehin nur endlich viele Punkte betrachten können und uns auf reellwertige Daten beschränken wollen, genügt die Untersuchung von Polynomen in Π_d.

3.4. Endlichkeit vs. Geometrie

Im letzten Abschnitt wurde deutlich, dass jede endliche Punktmenge $\Xi \subset \mathbb{R}^d$ als Varietät aufgefasst werden kann. Unter Umständen ist die Endlichkeit der Punktmenge aber keine Eigenschaft der zu untersuchenden Daten, sondern nur durch Messintervalle, Abtastung oder den endlichen diskreten Speicher im Computersystem begründet.

Fassen wir eine endliche Punktmenge $\Xi \subset \mathbb{R}^d$ als Diskretisierung einer unendlichen Varietät $\widetilde{\Xi} \subseteq \mathbb{R}^d$ auf, so setzt sich das Ideal $\mathfrak{I}(\Xi)$ aus zwei Kategorien von Polyno-

men zusammen. Einerseits ist aufgrund der Inklusionsumkehrung der Abbildung $\mathfrak{I}$ jedes Polynom aus $\mathfrak{I}(\widetilde{\Xi})$ auch in $\mathfrak{I}(\Xi)$ enthalten. Diese Polynome enthalten also Informationen über die Geometrie der kontinuierlichen Varietät. Andererseits kommen weitere Polynome hinzu, die die Abtastungsstellen der Diskretisierung beschreiben. Daher ist es im Allgemeinen nicht möglich, aus einer Basis von $\mathfrak{I}(\Xi)$ auf die Varietät $\widetilde{\Xi}$ zu schließen, was der Berechnung einer Differenz von Idealen entsprechen würde. Der folgende Satz belegt dies:

Satz 3.31. *Seien $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3 \subset \Pi_d$ Ideale, dann folgt aus $(\mathcal{F}_1 + \mathcal{F}_2) = (\mathcal{F}_1 + \mathcal{F}_3)$ im Allgemeinen* nicht *$\mathcal{F}_2 = \mathcal{F}_3$.*

Beweis. Wir zeigen die Behauptung durch Konstruktion eines Gegenbeispiels. Sei $\mathcal{F}_1 = \langle x_1 - x_2 \rangle$, $\mathcal{F}_2 = \langle x_1^2 + x_2^2 - 2 \rangle$, $\mathcal{F}_3 = \langle x_1^2 + x_1 x_2 + x_2^2 - 3 \rangle$, dann gilt

$$\begin{aligned}
\mathcal{F}_1 + \mathcal{F}_2 &= \langle x_1 - x_2, x_1^2 + x_2^2 - 2 \rangle \\
&= \left\langle x_1 - x_2, \frac{-(x_1 - x_2)}{2}(x_1 - x_2) + \frac{3}{2}(x_1^2 + x_2^2 - 2) \right\rangle \\
&= \langle x_1 - x_2, x_1^2 + x_1 x_2 + x_2^2 - 3 \rangle = \mathcal{F}_1 + \mathcal{F}_3.
\end{aligned}$$

Da $\mathcal{F}_2 =: \langle f_2 \rangle$ und $\mathcal{F}_3 =: \langle f_3 \rangle$ Hauptideale sind und $f_2(x_1, x_2) = x_1^2 + x_2^2 - 2$ und $f_3(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 - 3$ teilerfremd sind, ist $\mathcal{F}_2 \neq \mathcal{F}_3$. □

Wir können nun das Ideal $\mathcal{F}_1 + \mathcal{F}_2 = \langle x_1 - x_2, x_1^2 + x_2^2 - 2 \rangle = \langle x_1^2 - 1, x_1^2 + x_2^2 - 2 \rangle$ als Diskretisierung des Ideals $\mathcal{F}_2$ an den Stellen $(\pm 1, \pm 1) \in \mathbb{R}^2$ auffassen. Ebenso entspricht das Ideal $\mathcal{F}_1 + \mathcal{F}_3 = \langle x_1 - x_2, x_1^2 + x_1 x_2 + x_2^2 - 3 \rangle = \langle x_1^2 - 1, x_1^2 + x_1 x_2 + x_2^2 - 3 \rangle$ einer Diskretisierung des Ideals $\mathcal{F}_3$ an denselben Stellen. Wegen $\mathcal{F}_1 + \mathcal{F}_2 = \mathcal{F}_1 + \mathcal{F}_3$ sind diese Diskretisierungen nicht unterscheidbar und daher kann auch nicht auf die zugrunde liegende kontinuierliche Varietät $\mathfrak{V}(\mathcal{F}_2)$ bzw. $\mathfrak{V}(\mathcal{F}_3)$ geschlossen werden.

Dieses idealtheoretische Gegenbeispiel lässt sich auch anhand der Varietäten verdeutlichen: Die reellen Anteile der Varietäten von $\mathcal{F}_1$, $\mathcal{F}_2$ und $\mathcal{F}_3$ sind eine Ursprungsgerade mit Steigung 1, ein Kreis um den Ursprung mit Radius $\sqrt{2}$ und eine Ellipse um den Ursprung mit den Hauptscheiteln $(\pm\sqrt{3}, \mp\sqrt{3})$ und den Nebenscheiteln $(\pm 1, \pm 1)$. Abbildung 3.3 zeigt diese reellen Varietäten graphisch. Offensichtlich

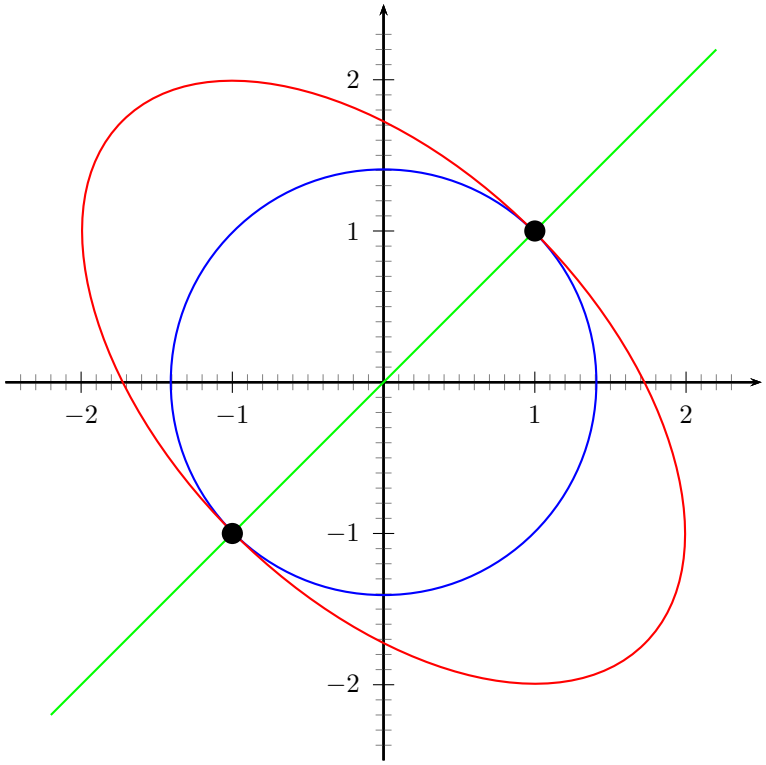


Abbildung 3.3.: Reelle Anteile der Varietäten zu den Idealen des Gegenbeispiels aus Satz 3.31. Die Varietät von $\mathcal{F}_1$ entspricht einer Geraden (grün), die Varietäten von $\mathcal{F}_2$ und $\mathcal{F}_3$ sind ein Kreis (blau) bzw. eine Ellipse (rot). Alle drei Varietäten schneiden sich paarweise in den Punkten $(-1,-1)$ und $(1,1)$.

schneiden sich Gerade und Kreis bzw. Gerade und Ellipse jeweils genau in den Punkten $(\pm 1, \pm 1)$. Tatsächlich entsprechen diese Punkte auch den komplexen Varietäten von $\mathcal{F}_1 + \mathcal{F}_2$ und $\mathcal{F}_1 + \mathcal{F}_3$, denn es gilt

$$\begin{aligned}
\mathfrak{V}(\mathcal{F}_1) &= \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{C}^2 : x_1 - x_2 = 0 \right\} = \left\{ \begin{pmatrix} a+ib \\ a+ib \end{pmatrix} : a, b \in \mathbb{R} \right\}, \\
\mathfrak{V}(\mathcal{F}_2) &= \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{C}^2 : x_1^2 + x_2^2 - 2 = 0 \right\} \\
&= \left\{ \begin{pmatrix} a+ib \\ c+id \end{pmatrix} : \begin{matrix} a^2 + c^2 - b^2 - d^2 = 2 \\ ab + cd = 0 \end{matrix}, \ a, b, c, d \in \mathbb{R} \right\},
\end{aligned}$$

$$\begin{aligned}
\mathfrak{V}(\mathcal{F}_3) &= \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{C}^2 : x_1^2 + x_1 x_2 + x_2^2 - 3 = 0 \right\} \\
&= \left\{ \begin{pmatrix} a+ib \\ c+id \end{pmatrix} : \begin{matrix} a^2 + c^2 + ac - b^2 - d^2 - bd = 2 \\ 2ab + 2cd + ad + bc = 0 \end{matrix},\ a,b,c,d \in \mathbb{R} \right\}, \\
\mathfrak{V}(\mathcal{F}_1 + \mathcal{F}_2) &= \left\{ \begin{pmatrix} a+ib \\ a+ib \end{pmatrix} : \begin{matrix} 2a^2 - 2b^2 = 2 \\ 2ab = 0 \end{matrix},\ a,b \in \mathbb{R} \right\} = \left\{ \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}, \\
\mathfrak{V}(\mathcal{F}_1 + \mathcal{F}_3) &= \left\{ \begin{pmatrix} a+ib \\ a+ib \end{pmatrix} : \begin{matrix} 3a^2 - 3b^2 = 3 \\ 6ab = 0 \end{matrix},\ a,b \in \mathbb{R} \right\} = \left\{ \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}.
\end{aligned}$$

Zusammen ergibt sich also

$$\mathfrak{V}(\mathcal{F}_1 + \mathcal{F}_2) = \mathfrak{V}(\mathcal{F}_1) \cap \mathfrak{V}(\mathcal{F}_2) = \{(-1,-1),(1,1)\} = \mathfrak{V}(\mathcal{F}_1) \cap \mathfrak{V}(\mathcal{F}_3) = \mathfrak{V}(\mathcal{F}_1 + \mathcal{F}_3).$$

Es ist damit nicht entscheidbar, ob der Kreis oder die Ellipse durch die Gerade diskretisiert wurde. Ebenso wenig lässt sich darauf schließen, welches der beteiligten Ideale bzw. welche der zugehörigen Varietäten die Diskretisierung beschreibt, was in diesem Fall der geringen Anzahl an Abtastungen geschuldet ist. Für hinreichend viele Punkte können wir jedoch ein Kriterium für die Entscheidung angeben, ob eine endliche Varietät Teil einer einfacheren – im Sinne eines kleineren Polynomgrads – kontinuierlichen Varietät ist. Dazu untersuchen wir zunächst einmal Punktmengen ohne einen nichttrivialen geometrischen Zusammenhang.

Satz 3.32. *Sei $\Xi \subset \mathbb{R}^d$, $\#\Xi = N > d > 1$, eine Punktmenge, die nicht in einer niederdimensionalen Untermannigfaltigkeit liegt, und sei $k \in \mathbb{N}$ so gewählt, dass $\#\boldsymbol{T}_{k-1,d} \leq N < \#\boldsymbol{T}_{k,d}$. Dann besteht jede reduzierte H-Basis des Ideals $\mathfrak{I}(\Xi)$ aus $\#\boldsymbol{T}_{k,d} - N$ Polynomen vom Grad k sowie*

$$\max\left\{0, \#\boldsymbol{T}^0_{k+1,d} - d\,(\#\boldsymbol{T}_{k,d} - N)\right\}$$

Polynomen vom Grad $k+1$.

Beweis. Seien $\Xi = \{\xi_1, \ldots, \xi_N\} \subset \mathbb{R}^d$ und $f \in \Pi_d$, $\deg(f) = k$, dann hat f einen Koeffizientenvektor $[f_1, \ldots, f_{\#\boldsymbol{T}_{k,d}}] \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$. Nach Voraussetzung gilt dabei $d < N < \#\boldsymbol{T}_{k,d}$ bzw. $d < \binom{d+k}{k} - 1$, was $k \geq 2$ impliziert. Die Forderung $f(\Xi) = 0$

entspricht somit dem linearen Gleichungssystem

$$\underbrace{\begin{bmatrix} 1 & \xi_{1,d} & \xi_{1,d-1} & \cdots & \xi_{1,1}^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{N,d} & \xi_{N,d-1} & \cdots & \xi_{N,1}^k \end{bmatrix}}_{=:A} \cdot \begin{bmatrix} f_1 \\ \vdots \\ f_{\#\boldsymbol{T}_{k,d}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.6}$$

Die Matrix $A = [\xi^\alpha : \xi \in \Xi, \alpha \in \mathbb{N}_0, |\alpha| \leq k]$ stellt dabei die Auswertungsmatrix aller Terme vom Grad kleiner oder gleich k an den Punkten Ξ in graduiert-lexikographischer Ordnung dar. Da die Punkte Ξ nicht in einer niederdimensionalen Untermannigfaltigkeit liegen, können wir $\operatorname{rank}(A) = N$ voraussetzen. Das homogene lineare Gleichungssystem (3.6) hat damit genau dann eine nichttriviale Lösung, wenn $\#\boldsymbol{T}_{k,d} > N$ gilt. In diesem Fall ist $\dim(\mathcal{N}(A)) = \#\boldsymbol{T}_{k,d} - N$. Wegen $\#\boldsymbol{T}_{k-1,d} = \binom{d+(k-1)}{k-1} \leq N$ gilt zudem

$$\dim(\mathcal{N}(A)) \leq \binom{d+k}{k} - \binom{d+(k-1)}{(k-1)} = \binom{d+k-1}{k} = \dim(\Pi_{k,d}^0).$$

Wir können also eine Menge $F_k \subset \Pi_{k,d} \setminus \Pi_{k-1,d}$ mit $\operatorname{span}(F_k) = \mathcal{N}(A)$ und $\Lambda(F_k)$ linear unabhängig wählen, was eine Voraussetzung für reduzierte H-Basen ist, vgl. Satz 3.18. Somit erhalten wir $\#F_k = \#\boldsymbol{T}_{k,d} - N$ Basispolynome vom Grad k.

Aus demselben Argument kann es in der H-Basis höchstens $\binom{d+(k+1)-1}{k+1} = \#\boldsymbol{T}_{k+1,d}^0$ Polynome vom Grad $k+1$ geben, da andernfalls linear abhängige Leitformen auftreten. Wir bezeichnen diese Polynome im Folgenden als $F_{k+1} \subset \Pi_{k+1,d} \setminus \Pi_{k,d}$. Zusätzlich muss nach Satz 3.18 die Beziehung $\Lambda(F_{k+1}) \subseteq \mathcal{W}_{k+1,d}^0(F_k)$ gelten, denn die gesuchte H-Basis ist nach Voraussetzung reduziert. Wir erhalten somit $\#F_{k+1} = \dim(\mathcal{W}_{k+1,d}^0(F_k)) = \#\boldsymbol{T}_{k+1,d}^0 - \dim(\mathcal{V}_{k+1,d}^0(F_k))$ Polynome vom Grad $k+1$. Analog zur Konstruktion eines Erzeugendensystems von $\mathcal{V}_{k+1}^0(F_k)$ nach Satz 2.8 gilt

$$\dim(\mathcal{V}_{k+1,d}^0(F_k)) \leq d \cdot \dim(\mathcal{V}_{k,d}^0(F_k)) = d \cdot (\#\boldsymbol{T}_{k,d} - N). \tag{3.7}$$

Wir betrachten daher die verschobenen Leitformen von F_k als Zeilen der Matrix

$$V(F_k) := \left[x^\alpha \Lambda(f) : f \in F_k,\ \alpha \in \mathbb{N}_0^d,\ |\alpha| = 1\right] \in \mathbb{R}^{(d\cdot(\#\boldsymbol{T}_{k,d}-N))\times\#\boldsymbol{T}_{k+1,d}^0}$$

mit dem Zeilenraum $\operatorname{span}(V) = \mathcal{V}^0_{k+1,d}(F_k)$. Zur Bestimmung des Rangs von $V(F_k)$ verwenden wir folgenden Ansatz: Man wähle eine Teilmenge $\widetilde{\Xi} \subset \Xi$ mit $\#\widetilde{\Xi} = \#\boldsymbol{T}_{k-1,d}$ und konstruiere dazu Polynome $\widetilde{F}_k \subset \Pi_{k,d} \setminus \Pi_{k-1,d}$ mit $\widetilde{F}_k(\widetilde{\Xi}) = 0$ und $\Lambda(\widetilde{F}_k)$ linear unabhängig wie oben. Dann gilt $\operatorname{span}(\Lambda(\widetilde{F}_k)) = \Pi^0_{k,d}$ und $\operatorname{rank}(V(\widetilde{F}_k)) = \#\boldsymbol{T}^0_{k+1,d}$. Nun werden der Menge $\widetilde{\Xi}$ schrittweise die entfernten Punkte wieder hinzugefügt, sodass eine aufsteigende Kette

$$\widetilde{\Xi} =: \widetilde{\Xi}^{(0)} \subset \widetilde{\Xi}^{(1)} \subset \widetilde{\Xi}^{(2)} \subset \cdots \subset \Xi$$

entsteht. Die zugehörigen Polynome $\widetilde{F}_k^{(j)}$ seien analog zu $\widetilde{F}_k$ gewählt und es gilt $\widetilde{F}_k^{(j+1)} \in \operatorname{span}(\widetilde{F}_k^{(j)})$ mit $\#\widetilde{F}_k^{(j+1)} = \#\widetilde{F}_k^{(j)} - 1$, $j = 1, \ldots, \#(\Xi \setminus \widetilde{\Xi}) - 1$. Wir unterscheiden nun die folgenden Fälle:

1. Für $\#\boldsymbol{T}_{k,d} - N = \#F_k \geq \frac{\#\boldsymbol{T}^0_{k+1,d}}{d}$ gilt $\operatorname{rank}(V(F_k)) = \operatorname{rank}(V(\widetilde{F}_k)) = \#\boldsymbol{T}^0_{k+1,d}$, da für $j < \#(\Xi \setminus \widetilde{\Xi})$ aus jeder Menge $\widetilde{F}_k^{(j)}$ ein Polynom entfernt werden kann, ohne den Rang von $V(\widetilde{F}_k^{(j)})$ zu verändern. Dabei ist die Voraussetzung $k \geq 2$ notwendig, denn für $d = 3$ gilt beispielsweise $\operatorname{rank}(V(\{x_1-1, x_2-1, x_3-1\})) = 6 = \#\boldsymbol{T}^0_{2,2}$, aber jede zweielementige Teilmenge führt zu Rang 5.

2. Ist hingegen $\#\boldsymbol{T}_{k,d} - N = \#F_k < \frac{\#\boldsymbol{T}^0_{k+1,d}}{d}$, so besteht die Matrix $V(F_k)$ aus weniger Zeilen als Spalten. Da in diesem Fall durch das Entfernen von Polynomen aus $\widetilde{F}_k$ keine linearen Abhängigkeiten entstehen können, hat die Matrix $V(F_k)$ vollen Zeilenrang bzw. $\operatorname{rank}(V(F_k)) = (d \cdot (\#\boldsymbol{T}_{k,d} - N))$.

Zusammen ergibt sich

$$\#F_{k+1} = \dim(\mathcal{W}^0_{k+1,d}(F_k)) = \max\left\{0, \#\boldsymbol{T}^0_{k+1,d} - d\,(\#\boldsymbol{T}_{k,d} - N)\right\}. \tag{3.8}$$

Für $\#F_{k+1} = 0$ folgt nun $\dim(\mathcal{V}^0_{k+1,d}(F_k)) = \#\boldsymbol{T}^0_{k+1,d}$ und damit $\mathcal{W}^0_{k+1,d}(F_k) = \{0\}$. Andernfalls ist $\mathcal{V}^0_{k+1,d}(F_k \cup F_{k+1}) = \Pi^0_{k+1,d}$, was ebenfalls zu $\mathcal{W}^0_{k+1,d}(F_k \cup F_{k+1}) = \{0\}$ führt. In beiden Fällen ist die konstruierte Menge $F = F_k \cup F_{k+1}$ nach Satz 3.15 eine H-Basis und nach Satz 3.18 reduziert. □

Die Werte für $d = 2, 3, 4$ und $N = 1, \ldots, 30$ sind in Tabelle 3.1 angegeben. Die Einträge mit $N \leq d$ sind dabei nach den Voraussetzungen von Satz 3.32 nicht zulässig und hier nur zur Vollständigkeit der Tabelle angegeben. Insbesondere wird

	k						
N	1	2	3	4	5	6	7
1	2						
2	1	1					
3		3					
4		2					
5		1	2				
6			4				
7			3				
8			2	1			
9			1	3			
10				5			
11				4			
12				3			
13				2	2		
14				1	4		
15					6		
16					5		
17					4		
18					3	1	
19					2	3	
20					1	5	
21						7	
22						6	
23						5	
24						4	
25						3	2
26						2	4
27						1	6
28							8
29							7
30							6

(a) $d = 2$

	k				
N	1	2	3	4	5
1	3				
2	2	1			
3	1	3			
4		6			
5		5			
6		4			
7		3	1		
8		2	4		
9		1	7		
10			10		
11			9		
12			8		
13			7		
14			6		
15			5		
16			4	3	
17			3	6	
18			2	9	
19			1	12	
20				15	
21				14	
22				13	
23				12	
24				11	
25				10	
26				9	
27				8	
28				7	
29				6	3
30				5	6

(b) $d = 3$

	k			
N	1	2	3	4
1	4			
2	3	1		
3	2	3		
4	1	6		
5		10		
6		9		
7		8		
8		7		
9		6		
10		5		
11		4	4	
12		3	8	
13		2	12	
14		1	16	
15			20	
16			19	
17			18	
18			17	
19			16	
20			15	
21			14	
22			13	
23			12	
24			11	
25			10	
26			9	
27			8	3
28			7	7
29			6	11
30			5	15

(c) $d = 4$

Tabelle 3.1.: Anzahl der Elemente einer H-Basis des Ideals $\mathfrak{I}(\Xi)$ zu einer Punktmenge $\Xi \subset \mathbb{R}^d$ mit $\#\Xi = N$, die nicht in einer niederdimensionalen Untermannigfaltigkeit liegt.

die Anzahl der Basispolynome vom Grad $k+1$ nach Gleichung (3.8) falsch: Für $N = 2$, $d = 3$ erhalten wir $2 = \#\boldsymbol{T}_{1,3} - 2$ Basispolynome vom Grad 1, aber auch ein Basispolynom vom Grad 2, im Widerspruch zu Gleichung (3.8), die $\max\{0, \#\boldsymbol{T}^0_{2,3} - 3(\#\boldsymbol{T}_{1,3} - 2) = 0$ liefert.

Für eine Punktmenge $\Xi \subset \mathbb{R}^d$ der Mächtigkeit N und $\#\boldsymbol{T}_{k-1,d} \leq N < \#\boldsymbol{T}_{k,d}$ beschreiben die Basispolynome, deren Grad kleiner als k ist, nichttriviale geometrische Eigenschaften des Ideals $\mathfrak{I}(\Xi)$ bzw. dessen Varietät. Dieses Resultat trägt entscheidend zur Interpretierbarkeit der numerischen Ergebnisse in Kapitel 6 bei. Die Einschränkung $N > d$ ist dabei notwendig, da eine Punktmenge Ξ mit $\#\Xi = N \leq d$ trivialerweise in einer niederdimensionalen Untermannigfaltigkeit des $\mathbb{R}^d$ liegt. In diesem Fall ist stets $k = 1$ und jede H-Basis des Ideals $\mathfrak{I}(\Xi)$ besteht aus $d + 1 - N$ linearen Polynomen und $\binom{N}{N-2}$ Polynomen vom Grad 2. Man vergleiche dazu das Gegenbeispiel aus Satz 3.31 bzw. Abbildung 3.3.

4 Approximative Ideale

Inhalt

4.1. Approximative H-Basen 77
4.2. Ringoperationen auf approximativen Idealen 107
4.3. Das approximative Ideal-Membership-Problem 125
4.4. Dünn besetzte H-Basen 128

In numerischer Rechnung ist nicht davon auszugehen, dass für eine exakte Nullstelle $\xi \in \mathbb{R}^d$ eines Polynoms $f \in \Pi_d$ auch die Polynomauswertung $f(\xi) = 0$ liefert. Sowohl die Koeffizientenvektoren als auch die Auswertungsstellen liegen nur in endlicher Genauigkeit vor, sodass trotz stabiler Verfahren (vgl. Abschnitt 2.3) das Auftreten von Rundungsfehlern nicht verhindert werden kann. Eine ähnliche Situation ist aus einem klassischen Problem der numerischen Mathematik bekannt: der iterativen Nullstellenbestimmung, vgl. [FH07], [SK11]. Da auch für dieses Problem keine exakte Lösung $\xi \in \mathbb{R}$ mit $f(\xi) = 0$ zu erwarten ist, verwenden die Standardmethoden, wie z. B. das Newton-Verfahren, eine Toleranz $\varepsilon > 0$ und akzeptieren einen Punkt $\widetilde{\xi} \in \mathbb{R}$ als Lösung, wenn $|f(\widetilde{\xi})| < \varepsilon$ gilt.

Analog dazu muss auch das notwendige Kriterium des Ideals $\mathfrak{I}(\xi)$ zu $|f(\xi)| \leq \varepsilon$ für eine Toleranz $\varepsilon > 0$ abgeschwächt werden. Ist – wie im letzten Kapitel – eine endliche Punktmenge $\Xi \subset \mathbb{R}^d$ vorgegeben, so lässt sich der Fehler, im Sinne des Abstands zur Null, durch eine p-Norm beschreiben. Diese ist wie folgt definiert:

Definition 4.1. *Zu einem Vektor* $\xi \in \mathbb{R}^d$ *und* $1 \leq p \leq \infty$ *ist die* p-Norm *gegeben als*

$$\|\xi\|_p := \sqrt[p]{\sum_{j=1}^{d} |\xi_j|^p},\ 1 \leq p < \infty, \quad \|\xi\|_\infty := \max_{j=1,\dots,d} |\xi_j|.$$

Dies ermöglicht die Formulierung einer numerischen Entsprechung des exakten Ideals $\mathfrak{I}(\Xi)$ aus Satz 3.21 für endliche Mengen $\Xi \subset \mathbb{R}^d$. Die Endlichkeit von Ξ sei ebenfalls als Generalvoraussetzung für das gesamte Kapitel angenommen. Die folgende Definition des *p-approximativen Ideals* wird von Sauer in [Sau07] angegeben:

Definition 4.2. *Sei* $\varepsilon > 0$ *und* $\Xi \subset \mathbb{R}^d$*, dann ist*

$$\mathfrak{I}_{p,\varepsilon}(\Xi) := \left\{ f \in \Pi_d : \frac{\|f(\Xi)\|_p}{\|f\|_2} \leq \varepsilon \right\}, \quad 1 \leq p \leq \infty, \tag{4.1}$$

das p-approximative Ideal mit Toleranz ε*.*

Obwohl die in Definition 4.2 verwendeten Normen jeweils p-Normen sind, unterscheiden sie sich in ihrer Funktion: Die Norm $\|f(\Xi)\|_p$ beschreibt den Abstand des Auswertungsvektors $f(\Xi)$ zur Null im oben angegebenen Sinne, was die Bezeichnung dieser Norm als *Auswertungsnorm* motiviert. Im Folgenden kommen dafür die p-Normen mit $p = 2$ und $p = \infty$ zur Anwendung. Die Norm $\|f\|_2 = \sqrt{(f,f)}$ normalisiert den Koeffizientenvektor $f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$, weshalb diese auch als *Koeffizientennorm* bezeichnet wird. Die Normalisierung der Polynome in (4.1) ist notwendig, da andernfalls alle Polynome mit hinreichend kleinen Koeffizienten die Forderung $\|f(\Xi)\|_p \leq \varepsilon$ erfüllen.

In Satz 3.21 wurde gezeigt, dass die Menge $\mathfrak{I}(\Xi)$ für $\Xi \subset \mathbb{R}^d$ ein Ideal von Π_d ist. Dies trifft auf p-approximative Ideale im Allgemeinen nicht zu. Das folgende Beispiel verdeutlicht diese Eigenschaft:

Beispiel 4.3. *Sei* $\Xi := \left\{ (t,t)^T \in \mathbb{R}^2 : t \in \{-2,-1,0,1,2\} \right\}$ *eine Menge von Punkten und* $0 < \varepsilon < 1$*, dann ist* $\mathfrak{I}_{\infty,\varepsilon}(\Xi) \ni f(x_1,x_2) = x_1 - x_2 + \varepsilon$*, da*

$$\frac{\|f(\Xi)\|_\infty}{\|f\|_2} = \frac{\varepsilon}{\sqrt{2+\varepsilon^2}} \leq \varepsilon.$$

Allerdings ist $(x_1 \cdot f)(x_1, x_2) \notin \mathfrak{I}_{\infty,\varepsilon}(\Xi)$*, da*

$$\frac{\|(x_1 \cdot f)(\Xi)\|_\infty}{\|x_1 \cdot f\|_2} = \frac{|(x_1 \cdot f)(2,2)|}{\sqrt{2+\varepsilon^2}} = \frac{2\varepsilon}{\sqrt{2+\varepsilon^2}} = \varepsilon \underbrace{\sqrt{\frac{4}{2+\varepsilon^2}}}_{>1,\ da\ \varepsilon<1.} > \varepsilon.$$

Die Einschränkung $0 < \varepsilon < 1$, die in Beispiel 4.3 gemacht wurde, ist für die Definition eines ∞-approximativen Ideals zwar nicht notwendig, aber dennoch sinnvoll. Die Wahl einer Toleranz $\varepsilon \geq 1$ hat zur Folge, dass die Forderung $\|f(\Xi)\|_\infty \leq \varepsilon \|f\|_2$ für alle konstanten Polynome $f \in \Pi_{0,d}$ erfüllt ist. Das konstante Polynom $f = 1$ stellt jedoch das Einselement des Rings Π_d dar und somit muss $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ das triviale Ideal Π_d sein. Dies wurde auch in [Sau07] bemerkt.

In [HKPP09] haben Heldt, Kreuzer, Pokutta und Poulisse mit dem *Almost Vanishing Ideal* (AVI) eine ähnliche Definition angegeben, die sich aber in einem Punkt ganz wesentlich unterscheidet: Während das oben definierte p-approximative Ideal *kein* Ideal ist, wird für AVIs die Bedingung $\|f(\Xi)\|_2 \leq \varepsilon \|f\|_2$ nur für die Basis eines Ideals gefordert. Damit sind AVIs tatsächlich Ideale. Im nächsten Abschnitt wird durch die Einführung von *approximativen H-Basen* das Analogon zu AVIs angegeben.

4.1. Approximative H-Basen

Obwohl das p-approximative Ideal $\mathfrak{I}_{p,\varepsilon}(\Xi)$ im Allgemeinen kein Ideal ist, kann aus Polynomen dieser Menge eine H-Basis konstruiert werden. Eine solche Basis wird dann als *approximative H-Basis* bezeichnet.

Definition 4.4. *Sei* $\varepsilon > 0$ *und* $\Xi \subset \mathbb{R}^d$*. Eine Menge von Polynomen* $F \subset \mathfrak{I}_{p,\varepsilon}(\Xi)$ *heißt* approximative H-Basis*, falls*

$$\langle F \rangle \ni g = \sum_{f \in F} g_f \cdot f, \quad \deg(g) \geq \deg(g_f) + \deg(f).$$

Approximative H-Basen wurden bereits von Sauer (vgl. [Sau07]) bzw. Heldt, Kreuzer, Pokutta und Poulisse (siehe [HKPP09], dort unter der Bezeichnung *Macaulay-Basis*) untersucht. Dieser Abschnitt folgt in weiten Teilen der Arbeit von Sauer und der

dort vorgestellten Konstruktion für $p = \infty$. Darüber hinaus werden einige neue Modifikationen des Verfahrens angegeben, die Abhängigkeit von Startwerten untersucht und die Vorgehensweise verallgemeinert, sodass auch der Fall $p = 2$ behandelt werden kann. Dies liefert eine Alternative zu der von Heldt, Kreuzer, Pokutta und Poulisse präsentierten Methode, die speziell für $p = 2$ konstruiert ist.

4.1.1. Numerische Bestimmung einer approximativen H-Basis

Ein zentrales Hilfsmittel zur Bestimmung einer approximativen H-Basis nach dem Verfahren von Sauer ist eine Erweiterung der bekannten QR-Zerlegung von Matrizen, vgl. [GVL96]. Dabei wird zusätzlich zu einer orthogonalen Matrix Q und einer Rechtsdreiecksmatrix R eine Permutationsmatrix P durch Spaltenpivotisierung bestimmt. Aus diesem Grund ist die Zerlegung auch als *QRP-Zerlegung* bekannt. Diese wurde von Businger und Golub in [BG65] zur Lösung überbestimmter linearer Gleichungssysteme entwickelt. Zur Bestimmung einer approximativen H-Basis werden zwei wichtige Eigenschaften der QRP-Zerlegung benötigt, die in folgendem Lemma zusammengefasst sind:

Lemma 4.5. *Sei $A \in \mathbb{R}^{m\times n}$, $m \leq n$, und $A \cdot P^T = Q \cdot R$ eine QRP-Zerlegung von A. Dann sind die Diagonalelemente $(R)_{kk}$, $k = 1, \ldots, m$, bezüglich ihrer Absolutbeträge in absteigender Reihenfolge geordnet und dominieren die jeweilige Zeile, d. h.*

$$|(R)_{11}| \geq \cdots \geq |(R)_{mm}|, \quad \textit{und} \quad |(R)_{jj}| \geq |(R)_{jk}|, \quad k = j+1, \ldots, n.$$

Durch diesen Zusammenhang ist die QRP-Zerlegung auch ein wichtiges Hilfsmittel zur numerischen Bestimmung des Rangs einer Matrix, vgl. [GVL96]. Ein Beweis von Lemma 4.5 ist in [Sau07] angegeben, ebenso wie eine Methode zur numerischen Bestimmung der QRP-Zerlegung mittels *Householder Transformationen*. Im Folgenden wird diese Methode unter Verwendung einer Toleranzschwelle $\varepsilon > 0$ zu der in Algorithmus 4.6 dargestellten *abgebrochenen QRP-Zerlegung* modifiziert. Dieser Ansatz findet sich auch in [SW05, 4.2]. Der folgende Satz verifiziert Algorithmus 4.6 und zeigt eine wichtige Eigenschaft der konstruierten Matrix B_k:

Algorithmus 4.6 : Abgebrochene QR-Zerlegung mit Spaltenpivotisierung

Input : $F \in \mathbb{R}^{m\times n}$, $m \leq n$, $\varepsilon > 0$
Output : $Q_k \in \mathbb{R}^{m\times m}$, $P_k \in \mathbb{R}^{n\times n}$, $R_k \in \mathbb{R}^{k\times k}$, $A_k \in \mathbb{R}^{k\times n-k}$, $B_k \in \mathbb{R}^{m-k\times n-k}$

$Q_0 \leftarrow I_{m\times m}$
$P_0 \leftarrow I_{n\times n}$
$R_0 \leftarrow [] \in \mathbb{R}^{0\times 0}$
$A_0 \leftarrow [] \in \mathbb{R}^{0\times 0}$
$B_0 \leftarrow F$
$k \leftarrow 0$
while $\max_{j=1,\dots,n-k} \|(B_k)_j\|_2 \geq \varepsilon$ **do**
  $i \leftarrow \arg\max_{j=1,\dots,n-k} \|(B_k)_j\|_2$
  $\widetilde{P}_k \leftarrow [e_i, e_2, \dots, e_{i-1}, e_1, e_{i+1}, \dots, e_{n-k}], \quad e_j = (\underbrace{0, \dots, 0}_{j-1}, 1, 0, \dots, 0)^T$
  $y \leftarrow (B_k\widetilde{P}_k)_1 + \operatorname{sign}((B_k\widetilde{P}_k)_{1,1}) \cdot \|(B_k\widetilde{P}_k)_1\| \cdot e_1$
  $y \leftarrow y/\|y\|_2$
  $H_k \leftarrow I_{m-k\times m-k} - 2yy^T$
  $Q_{k+1} \leftarrow Q_k \cdot \begin{bmatrix} I & 0 \\ 0 & H_k \end{bmatrix}$
  $P_{k+1} \leftarrow \begin{bmatrix} I & 0 \\ 0 & \widetilde{P}_k \end{bmatrix} \cdot P_k$
  $\begin{bmatrix} R_{k+1} & A_{k+1} \\ 0 & B_{k+1} \end{bmatrix} \leftarrow Q_{k+1}^T F P_{k+1}^T$, mit $\begin{cases} R_{k+1} \in \mathbb{R}^{k+1\times k+1}, \\ A_{k+1} \in \mathbb{R}^{k+1\times n-(k+1)}, \\ B_{k+1} \in \mathbb{R}^{m-(k+1)\times n-(k+1)} \end{cases}$
  $k \leftarrow k + 1$
end

Satz 4.7. *Zu einer Matrix $F \in \mathbb{R}^{m\times n}$, $m \leq n$, und einer Toleranz $\varepsilon > 0$ liefert Algorithmus 4.6 eine Zerlegung*

$$Q_k^T F P_k^T = \begin{bmatrix} R_k & A_k \\ 0 & B_k \end{bmatrix},$$

wobei $Q_k \in \mathbb{R}^{m\times m}$ eine orthogonale Matrix, $P_k \in \mathbb{R}^{n\times n}$ eine Permutationsmatrix und $R_k \in \mathbb{R}^{k\times k}$ eine rechte obere Dreiecksmatrix beschreibt. Für die Diagonalelemente von R_k gilt dabei $|r_{11}| \geq \cdots \geq |r_{kk}| > \varepsilon$ und alle Einträge der Matrix B_k erfüllen $|b_{rs}| \leq \varepsilon$, $r = 1, \dots, m-k$, $s = 1, \dots, n-k$.

Beweis. Nach Konstruktion der Matrizen Q_k und P_k gilt

$$\begin{aligned}\begin{bmatrix} R_k & A_k \\ 0 & B_k \end{bmatrix} = Q_k^T F P_k^T &= \left(Q_{k-1} \cdot \begin{bmatrix} I & 0 \\ 0 & H_{k-1} \end{bmatrix} \right)^T \cdot F \cdot \left(\begin{bmatrix} I & 0 \\ 0 & \widetilde{P}_{k-1} \end{bmatrix} \cdot P_{k-1} \right)^T \\ &= \begin{bmatrix} I & 0 \\ 0 & H_{k-1}^T \end{bmatrix} \cdot (Q_{k-1}^T \cdot F \cdot P_{k-1}^T) \begin{bmatrix} I & 0 \\ 0 & \widetilde{P}_{k-1}^T \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & H_{k-1}^T \end{bmatrix} \cdot \begin{bmatrix} R_{k-1} & A_{k-1} \\ 0 & B_{k-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \widetilde{P}_{k-1}^T \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & H_{k-1}^T \end{bmatrix} \cdot \begin{bmatrix} R_{k-1} & A_{k-1}\widetilde{P}_{k-1}^T \\ 0 & B_{k-1}\widetilde{P}_{k-1}^T \end{bmatrix} = \begin{bmatrix} R_{k-1} & A_{k-1}\widetilde{P}_{k-1}^T \\ 0 & H_{k-1}^T B_{k-1} \widetilde{P}_{k-1}^T \end{bmatrix}.\end{aligned}$$

Da sowohl die *Householdermatrizen* H_{k-1} als auch die Transpositionsmatrizen $\widetilde{P}_{k-1}$ orthogonal und symmetrisch sind und nach Konstruktion

$$H_{k-1}(B_{k-1}\widetilde{P}_{k-1})_1 = [\alpha, 0, \ldots, 0]^T$$

für $|\alpha| = \|(B_{k-1}\widetilde{P}_{k-1})_1\|_2$ gilt, erhalten wir

$$\begin{bmatrix} R_k & A_k \\ 0 & B_k \end{bmatrix} = \left[\begin{array}{cc|c} R_{k-1} & (A_{k-1}\widetilde{P})_{i,1} & (A_{k-1}\widetilde{P}_{k-1})_{i,j} \\ 0 & \alpha & (H_{k-1}B_{k-1}\widetilde{P}_{k-1})_{1,j} \\ \hline 0 & 0 & (H_{k-1}B_{k-1}\widetilde{P}_{k-1})_{l,j} \end{array}\right], \quad \begin{array}{l} i = 1, \ldots, k-1, \\ j = 2, \ldots, n-(k-1), \\ l = 2, \ldots, m-(k-1). \end{array} \tag{4.2}$$

Also ist für alle $j = 1, \ldots, k$ die Matrix R_j eine rechte obere Dreiecksmatrix, Q_j orthogonal und P_j eine Permutationsmatrix.

Angenommen, es gibt ein Element b_{rs} der Matrix B_k mit $|b_{rs}| \geq \varepsilon$, dann gilt auch für die zugehörige Spalte der Matrix $\|(B_k)_s\|_2 \geq \varepsilon$. Dies steht jedoch im Widerspruch zu der Voraussetzung, dass der Algorithmus nach Schritt k terminiert und dementsprechend die Bedingung $\max_{j=1,\ldots,n-k} \|(B_k)_j\|_2 \geq \varepsilon$ *nicht* erfüllt ist.

Die Eigenschaft $|r_{11}| \geq \cdots \geq |r_{kk}| > \varepsilon$ folgt aus dem entsprechenden Resultat für die QRP-Zerlegung in Lemma 4.5 bzw. dem Beweis dazu aus [Sau07]. □

Natürlich lässt sich ein analoges Resultat auch für die gewöhnliche QRP-Zerlegung

herleiten – man vergleiche dazu [Sau07]. Eine Implementierung dieses Verfahrens findet man in der Bibliothek LAPACK [LAP13], auf die bereits im Zusammenhang mit der Singulärwertzerlegung verwiesen wurde. Die abgebrochene Zerlegung beschleunigt die Berechnung aber nicht unerheblich, da $(m - k)$ Iterationen entfallen.

An dieser Stelle benötigen wir nun wieder einige Verknüpfungen der Theorie multivariater Polynome mit der Linearen Algebra. Die Auswertung einer endlichen Menge von Polynomen $F \subset \Pi_d$ an einer endlichen Menge von Punkten $\Xi \subset \mathbb{R}^d$ lässt sich als Matrix schreiben, wobei die Polynome den Zeilen der Matrix und die Punkte entsprechend den Spalten zugeordnet werden. Das Resultat wird auch als *Vandermonde-Matrix* oder *Auswertungsmatrix* bezeichnet.

Definition 4.8 (Vandermonde-Matrix)**.** *Seien $F \subset \Pi_d$ endlich viele Polynome und $\Xi \subset \mathbb{R}^d$ endlich viele Punkte, dann ist*

$$F(\Xi) := \begin{bmatrix} f(\xi) : & f \in F \\ & \xi \in \Xi \end{bmatrix} \in \mathbb{R}^{\#F \times \#\Xi}$$

die zugehörige Vandermonde-Matrix *oder auch* Auswertungsmatrix*.*

Eine weitere Matrix erhalten wir aus einer endlichen Menge von Polynomen $F \subset \Pi_d$, wenn wir die Koeffizientenvektoren zeilenweise untereinander anordnen. Dabei sind Polynome mit kleinerem Grad als $\max\{\deg(f) : f \in F\}$ gegebenenfalls mit Nullen aufzufüllen. Natürlich ist diese Darstellung nicht eindeutig, da man die Zeilen der Matrix beliebig vertauschen kann. Dennoch wird eine Menge $F \subset \Pi_d$ im Folgenden stets mit einer solchen Matrix identifiziert. Analog dazu können endliche Punktmengen $\Xi \subset \mathbb{R}^d$ als Matrix $\Xi := [\xi \in \Xi] \in \mathbb{R}^{\#\Xi \times d}$ aufgefasst werden. Die Mehrdeutigkeit der Darstellung bzgl. der Vertauschung von Zeilen gilt hier ebenfalls.

Die Verknüpfung von multivariaten Polynomen und Matrizen ermöglicht nun die Formulierung der folgenden Aussage aus [Sau07]. Dabei bezeichnen wir eine Menge von Polynomen $F \subseteq \Pi_d$ als *orthonormal*, falls diese paarweise orthogonal bzgl. des monischen Skalarprodukts und bzgl. der Koeffizientennorm normiert sind.

Lemma 4.9. *Seien $F_n \subset \Pi_{n,d} \setminus \Pi_{n-1,d}$ orthonormale Polynome und $Q \in \mathbb{R}^{\#F \times \#F}$ eine orthogonale Matrix, dann sind auch die Zeilen von $F' = Q^T F$ orthonormal.*

Untersucht man die Multiplikation einer Vandermonde-Matrix $F(\Xi)$ von links, so operiert diese auf dem Zeilenraum von $F(\Xi)$. Da die Zeilen der Vandermonde-Matrix den Polynomen $F \subset \Pi_d$ zugeordnet sind, genügt es, die Multiplikation mit der Koeffizientenmatrix zu betrachten. Dies folgt aus dem nächsten Lemma, vgl. [Sau07]:

Lemma 4.10. *Seien $F \subset \Pi_d$, $\Xi \subset \mathbb{R}^d$, beide endlich, und eine Matrix $A \in \mathbb{R}^{m \times \#F}$, $m \in \mathbb{N}$, gegeben, dann gilt $A \cdot F(\Xi) = (A \cdot F)(\Xi)$.*

Beweis. Sei $\Xi = \{\xi_j \in \mathbb{R}^d : j = 1, \ldots, \#\Xi\}$, dann gilt

$$\begin{aligned}((A \cdot F)(\Xi))_{i,j} &= \sum_{\alpha \in \mathbb{N}_0^d} \left(\sum_{f \in F} a_{i,f} f_\alpha \right) (\xi_j)^\alpha = \sum_{\alpha \in \mathbb{N}_0^d} \sum_{f \in F} a_{i,f} f_\alpha (\xi_j)^\alpha \\ &= \sum_{f \in F} a_{i,f} \left(\sum_{\alpha \in \mathbb{N}_0^d} f_\alpha (\xi_j)^\alpha \right) = \sum_{f \in F} a_{i,f} f(\xi_j) \\ &= (A \cdot (F(\Xi)))_{i,j}\end{aligned}$$

für alle $i = 1, \ldots, m$, $j = 1, \ldots, \#\Xi$. □

Der folgende Satz von Sauer aus [Sau07] verbindet nun die Resultate der Lemmata 4.9 und 4.10 mit Satz 4.7 und einer Anwendung auf Vandermonde-Matrizen:

Satz 4.11. *Sei $F_n \subset \Pi_{n,d} \setminus \Pi_{n-1,d}$ eine endliche Menge orthonormaler Polynome, $\Xi_n \subset \mathbb{R}^d$ eine endliche Punktmenge und $\varepsilon > 0$. Dann gibt es zwei endliche Mengen $F_n^+, F_n^0 \subset \operatorname{span}\{F_n\}$ orthonormaler Polynome, sodass*

$$F_n^+(\Xi_n^+) = R_n \in \mathbb{R}^{\#F_n^+ \times \#F_n^+}, \quad \min_{j=1,\ldots,\#F_n^+} |(R_n)_{jj}| > \varepsilon, \tag{4.3}$$

und

$$F_n^0 \subset \mathfrak{I}_{\infty,\varepsilon}(\Xi_n), \tag{4.4}$$

wobei $\Xi_n^+ \subseteq \Xi_n$ und $\#\Xi_n^+ = \#F_n^+$ gilt.

Beweis. Wir zerlegen die Vandermonde-Matrix $F_n(\Xi_n)$ nach Algorithmus 4.6. Nach

Satz 4.7 erhalten wir damit zur Fehlertoleranz $\varepsilon > 0$ die Situation

$$Q^T F_n(\Xi) P^T = \left[\begin{array}{c|c} R_n & A_n \\ \hline 0 & B_n \end{array}\right] =: \left[\begin{array}{c|c} F_n^+(\Xi_n^+) & F_n^+(\Xi_n^0) \\ \hline F_n^0(\Xi_n^+) & F_n^0(\Xi_n^0) \end{array}\right],$$

wobei

$$F_n^+ = (Q^T F_n)_{(i,k)}, \quad F_n^0 = (Q^T F_n)_{(j,k)}, \qquad \begin{array}{l} i = 1, \ldots, \#R, \\ j = \#R + 1, \ldots, \#F_n, \\ k = 1, \ldots, \#F_n, \end{array}$$

und

$$\Xi_n^+ = (P\Xi_n)_{(i,k)}, \quad \Xi_n^0 = (P\Xi_n)_{(j,k)}, \qquad \begin{array}{l} i = 1, \ldots, \#R, \\ j = \#R + 1, \ldots, \#\Xi_n, \\ k = 1, \ldots, d. \end{array}$$

Die Behauptung (4.3) folgt dann sofort nach Satz 4.7. Weiterhin gilt für alle $f \in F_n^0$ die Abschätzung

$$|f(\xi)| \leq \begin{cases} 0 & \text{falls } \xi \in \Xi_n^+ \\ \varepsilon & \text{falls } \xi \in \Xi_n^0 \end{cases}$$

und da $\Xi_n = \Xi_n^+ \cup \Xi_n^0$ folgt $|f(\xi)| \leq \varepsilon$ für alle $\xi \in \Xi_n$, $f \in F_n^0$. Nach Lemma 4.9 sind die Polynome F_n^0 orthonormal, da die Polynome F_n nach Voraussetzung orthonormal sind. Damit gilt

$$\frac{\|f(\Xi_n)\|_\infty}{\|f\|_2} = \|f(\Xi_n)\|_\infty = \max_{\xi \in \Xi_n} |f(\xi)| \leq \varepsilon, \quad f \in F_n^0.$$

Dies ist gleichbedeutend mit $F_n^0 \subset \mathfrak{I}_{\infty,\varepsilon}(\Xi_n)$ und zeigt somit (4.4). □

Satz 4.11 liefert den zentralen Baustein einer Methode zur Bestimmung einer approximativen H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$, die von Sauer in [Sau07] vorgestellt wurde. In Algorithmus 4.12 geben wir eine modifizierte Variante des Verfahrens an, die insbesondere Probleme mit den Abbruchbedingungen behebt. In der ursprünglichen Formulierung von Sauer bricht der Algorithmus ab, falls zur Laufzeit in Schritt n einer der folgenden Fälle auftritt:

1. Die Menge der Punkte $\Xi_n \subset \mathbb{R}^d$ ist leer oder

Algorithmus 4.12 : Bestimmung einer approximativen H-Basis

Input : Endliche Punktmenge $\Xi \subset \mathbb{R}^d$, Toleranz $0 < \varepsilon < 1$, Startwert $\xi^{(0)} \in \Xi$

Output : ∞-approximative H-Basis $F^0 = F_0^0 \cup \cdots \cup F_{n-1}^0$

$F_0^0 \leftarrow \emptyset$
$F_0^+ \leftarrow 1$
$\Xi_0 \leftarrow \Xi \setminus \xi^{(0)}$
$\Xi_0^+ \leftarrow \xi^{(0)}$
$R_0 \leftarrow 1$
$n \leftarrow 1$
while 1 **do**
  Bestimme eine Basis G_n von $\mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0)$.
  if $G_n = \emptyset$ **then break**;
  foreach $g \in G_n$ **do**
    $g_0 \leftarrow g$
    for $k = 0, \ldots, n-1$ **do**
      $g_{k+1} \leftarrow g_k - g_k(\Xi_k^+)^T R_k^{-1} F_k^+, \quad k = 0, \ldots, n-1.$
    end
  end
  Bestimme eine Orthonormalbasis F_n für den Raum $\operatorname{span}\{g_n : g \in G\}$.
  $\Xi_n \leftarrow \Xi \setminus (\Xi_0^+ \cup \cdots \cup \Xi_{n-1}^+) = \Xi_{n-1} \setminus \Xi_{n-1}^+$
  if $\Xi_n \neq \emptyset$ **then**
    Zerlege $F_n(\Xi_n) = Q \cdot R \cdot P^T$
    $k \leftarrow \max\{j : |R_{j,j}| > \varepsilon\}$
    $F_n^+ \leftarrow [(Q^T F_n)_j : j \leq k]$
    $F_n^0 \leftarrow [(Q^T F_n)_j : j > k]$
    $R_n \leftarrow \left[R_{i,j} : \begin{smallmatrix} i \leq k \\ j \leq k \end{smallmatrix}\right]$
    $\Xi_n^+ \leftarrow [(\Xi_n^T P)_j^T : j \leq k]$
  else
    $F_n^+ \leftarrow \emptyset$
    $F_n^0 \leftarrow F_n$
  end
  $n \leftarrow n + 1$
end

2. die Menge der Polynome $F_n^+ \subset \Pi_{n,d}$ ist leer.

Es wird jedoch nicht überprüft, ob die zu Beginn einer jeden Iteration konstruierte Menge G_n überhaupt Polynome enthält, man vergleiche die Zeilen 9-11 von Algorithmus 4.12. Ist die Menge G_n leer, so folgt $\mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0) = \operatorname{span}\{\emptyset\} = \{0\}$. Dies bedeutet aber, dass die Menge $F^0 := F_0^0 \cup \cdots \cup F_{n-1}^0$ nach Satz 3.15 bereits eine H-Basis ist. Der Algorithmus kann also an dieser Stelle abbrechen. Das folgende Beispiel zeigt eine Situation, in der $G_n = \emptyset$ und $F_{n-1}^+ \neq \emptyset \neq \Xi_{n-1}$ eintritt:

Beispiel 4.13. *Seien die Punkte $\Xi = \{(0,0),(1,0),(0,1),(1,1)\} \subset \mathbb{R}^2$ gegeben, der Startwert $\xi^{(0)} = (0,0)$ und $\varepsilon > 0$ nahe der Rechengenauigkeit gewählt. Dann liefert Algorithmus 4.12 bis auf Normierung die folgenden Mengen:*

1. $F_0^0 = \emptyset$, $F_0^+ = \{1\}$, $\Xi_0 = \Xi$,
2. $F_1^0 = \emptyset$, $F_1^+ = \{x_1 + x_2, x_1 - x_2\}$, $\Xi_1 = \{(1,0),(0,1),(1,1)\}$,
3. $F_2^0 = \{x_2^2 - x_2, x_1^2 - x_1\}$, $F_2^+ = \{x_1^2 - 2x_1x_2 + x_1\}$, $\Xi_2 = \{(1,0)\}$.

Hier kann der Algorithmus abbrechen, da

$$\mathcal{V}_{3,2}^0(F_0^0 \cup F_1^0 \cup F_2^0) = \operatorname{span}\left\{x_1x_2^2, x_2^3, x_1^3, x_1^2x_2\right\} = \Pi_{3,2}^0$$

gilt, was gleichbedeutend mit $\mathcal{W}_{3,2}^0(F_0^0 \cup F_1^0 \cup F_2^0) = \{0\}$ bzw. $G_3 = \emptyset$ ist.

Es ist also sinnvoll, dem Algorithmus das Abbruchkriterium $G_n = \emptyset$ hinzuzufügen, da in diesem Fall keine Polynome in F_n zu konstruieren sind. In der Theorie führt dies trivialerweise zu $F_n(\Xi_n) = \emptyset$ sowie $F_n^0 = F_n^+ = \emptyset$ und somit zu einem Abbruch. In der Praxis wird aber ein unnötiger Iterationsschritt ausgeführt und es werden Ausnahmebehandlungen trivialer Fälle, wie die Auswertung einer leeren Polynommenge etc. benötigt. Weiterhin wird in Abschnitt 4.1.4 eine Variante von Algorithmus 4.12 vorgestellt, in der das Abbruchkriterium $G_n = \emptyset$ tatsächlich notwendig ist.

Umgekehrt impliziert jedoch $F_n^+ = \emptyset$ bereits $G_{n+1} = \emptyset$, da in diesem Fall

$$\operatorname{span}\left(\Lambda(F_n^0)\right) = \operatorname{span}\left(\Lambda(F_n)\right) = \mathcal{W}_{n,d}^0\left(F_0^0 \cup \cdots \cup F_{n-1}^0\right), \tag{4.5}$$

also $\mathcal{W}^0_{n,d}(F^0_0 \cup \cdots \cup F^0_n) = \{0\}$ und damit $\operatorname{span}(G_{n+1}) = \mathcal{W}^0_{n+1,d}(F^0_0 \cup \cdots \cup F^0_n) = \{0\}$ bzw. $G_{n+1} = \emptyset$ gilt. Dies bedeutet, dass die Bedingung $F^+_n = \emptyset$ durch $G_n = \emptyset$ ersetzt werden kann. Konsequenterweise müsste man beide Abbruchbedingungen beibehalten, da sonst auch für $F^+_n = \emptyset$ eine unnötige Überprüfung von G_{n+1} durchgeführt wird. Da die Ausnahmebehandlung von $F^+_n = \emptyset$ aber ohnehin in der Initialisierung des Verfahrens benötigt wird, sei zugunsten der Übersichtlichkeit des Pseudocodes darauf verzichtet.

Untersucht man das zweite Abbruchkriterium aus [Sau07], $\Xi_n = \emptyset$, so fällt auf, dass in diesem Fall die Mengen F^0_n bzw. F^+_n nicht mehr konstruiert werden. Die Basis F^0 enthält also keine Polynome vom Grad n. Das folgende Beispiel zeigt, dass dieses Verhalten falsche Ergebnisse liefert:

Beispiel 4.14. *Wir betrachten die Punkte* $\Xi = \{(1,1), (-1,1), (1,-1)\} \subset \mathbb{R}^2$ *und wenden Algorithmus 4.12 mit dem Startwert* $\xi^{(0)} = (1,1)$ *und* $\varepsilon > 0$ *nahe der Rechengenauigkeit an. Dann erhalten wir bis auf Normierung:*

1. $F^0_0 = \emptyset$, $F^+_0 = \{1\}$, $\Xi_0 = \Xi$,
2. $F^0_1 = \emptyset$, $F^+_1 = \{x_1 - 2x_2 + 1, x_1 - 1\}$, $\Xi_1 = \{(-1,1), (1,-1)\}$,
3. $F^0_2 = \{x_1^2 - 2x_1x_2 - x_2^2 + 2x_1 + 2x_2 - 4, x_1^2 + x_1x_2 + x_2^2 - x_1 - x_2 - 1, x_1^2 - x_2^2\}$, $F^+_2 = \emptyset$, $\Xi_2 = \emptyset$.

Bricht der Algorithmus ab, ohne F^0_2 *zu bestimmen, so wird die leere Menge als Basis für* $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ *ausgegeben, was offensichtlich falsch ist.*

Da im Fall $\Xi_n = \emptyset$ keine Zerlegung der Vandermonde-Matrix $F_n(\Xi_n)$ möglich ist, müssen die Mengen F^0_n und F^+_n manuell festgelegt werden. Nach Konstruktion verschwinden die Polynome F_n an den Punkten $\Xi^+_0 \cup \cdots \cup \Xi^+_{n-1}$. Wenn nun $\Xi_n = \Xi \setminus (\Xi^+_0 \cup \cdots \cup \Xi^+_{n-1}) = \emptyset$ ist, muss bereits $\Xi^+_0 \cup \cdots \cup \Xi^+_{n-1} = \Xi$ sein. Damit gehören aber alle Polynome aus F_n zum Ideal $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$. Wir können also $F^0_n = F_n$ und $F^+_n = \emptyset$ wählen. Wie bereits oben gezeigt wurde impliziert $F^+_n = \emptyset$ direkt $G_{n+1} = \emptyset$, sodass es auch in diesem Fall ausreicht, die Mächtigkeit der Menge G_{n+1} zu überprüfen. Insgesamt können damit beide Abbruchkriterien aus [Sau07] durch die Bedingung $G_n = \emptyset$ ersetzt werden.

Der folgende Satz verifiziert Algorithmus 4.12. Der Beweis dazu wurde aus [Sau07] entnommen und an die oben beschriebenen Modifikationen des Verfahrens angepasst.

Satz 4.15. *Algorithmus 4.12 terminiert und liefert für eine endliche Menge $\Xi \subset \mathbb{R}^d$ und eine Toleranz $\varepsilon > 0$ eine approximative H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$.*

Beweis. In Schritt n muss entweder $G_n = \emptyset$ oder $G_n \neq \emptyset$ gelten. Da $G_n = \emptyset$ ein Abbruchkriterium ist, terminiert der Algorithmus in diesem Fall. Sei also $G_n \neq \emptyset$. Weiterhin kann $\Xi_n = \emptyset$ oder $\Xi_n \neq \emptyset$ sein. Im Fall $\Xi_n = \emptyset$ wird $F_n^+ = \emptyset$ gesetzt und damit gilt $G_{n+1} = \emptyset$, was bedeutet, dass der Algorithmus terminiert. Ist hingegen $\Xi_n \neq \emptyset$, so werden F_n^0 und F_n^+ berechnet. Auch hier unterscheiden wir $F_n^+ = \emptyset$ und $F_n^+ \neq \emptyset$. Für $F_n^+ = \emptyset$ terminiert der Algorithmus, da im nächsten Schritt $G_{n+1} = \emptyset$ gilt.

Wir betrachten nun den Fall

$$G_n \neq \emptyset,\ \Xi_n \neq \emptyset \text{ und } F_n^+ \neq \emptyset. \tag{4.6}$$

Es folgt $\#\Xi_n^+ = \#F_n^+ > 0$ und damit $\#\Xi_{n+1} = \#\Xi_n - \#\Xi_n^+ < \#\Xi_n$. Wir erhalten also eine absteigende Kette

$$\Xi_n \supset \Xi_{n+1} \supset \cdots \quad \text{mit} \quad \#\Xi_n > \#\Xi_{n+1} > \cdots . \tag{4.7}$$

Da Ξ endlich ist, bricht die Kette (4.7) nach endlich vielen Schritten ab, sodass die Konstellation (4.6) nur endlich oft auftreten kann. Damit terminiert der Algorithmus auch in diesem Fall.

Nach Konstruktion verschwinden die Polynome F_n an allen Punkten $\Xi_0^+ \cup \cdots \cup \Xi_{n-1}^+$. Damit gilt

$$F_n^0 \subseteq \operatorname{span}\{F_k : 0 \leq k \leq n-1\} \subseteq \mathfrak{I}(\Xi \setminus \Xi_n). \tag{4.8}$$

Ist nun $\Xi_n \neq \emptyset$, so folgt durch die QRP-Zerlegung $F_n^0 \subseteq \mathfrak{I}_{\infty,\varepsilon}(\Xi_n)$ und zusammen mit (4.8) gilt $F_n^0 \subseteq \mathfrak{I}_{\infty,\varepsilon}(\Xi)$. Sei nun $F^0 = \bigcup_{k=0}^n F_k^0$, dann erhalten wir die Zerlegungen

$$\Pi_{k,d}^0 = \mathcal{V}_{k,d}^0(F^0) \oplus \mathcal{W}_{k,d}^0(F^0), \quad 0 \leq k \leq n.$$

Da bei Terminierung des Algorithmus $\operatorname{span}(G_{n+1}) = \mathcal{W}^0_{n+1,d}(F^0) = \{0\} = \mathcal{W}^0_{k,d}(F^0)$ für $k > n$ gilt, ist F_0 nach Satz 3.15 eine H-Basis. □

4.1.2. Wahl der Toleranzschwelle

Der Beweis zu Satz 4.15 zeigt nicht nur, dass die konstruierte Menge F_0 eine H-Basis ist. Vielmehr muss das von dieser Basis erzeugte Ideal nulldimensional sein, da der lineare Raum $\mathcal{W}_d(F^0)$ nach Konstruktion endlichdimensional ist – man vergleiche dazu Satz 3.13 bzw. die darauf folgende Bemerkung. Dies führt zu folgendem Resultat:

Satz 4.16. *Sei $\Xi \subset \mathbb{R}^d$ endlich und $F \subset \Pi_d$ eine approximative H-Basis von $\mathfrak{I}_{\varepsilon,\infty}(\Xi)$. Dann ist $\langle F\rangle$ ein nulldimensionales Ideal und es gilt $\#\mathfrak{V}(\langle F\rangle) \leq \#\Xi$.*

Der Fall $\#\mathfrak{V}(\langle F\rangle) < \#\Xi$ tritt dabei immer dann auf, wenn zwei oder mehr Punkte der Menge Ξ bzgl. der Toleranz ε nicht unterscheidbar sind. Diese können dann in der Varietät von $\langle F\rangle$ durch einen Punkt dargestellt werden. Das folgende Beispiel zeigt eine Situation, in der zwei Punkte approximativ zu einem Punkt verschmelzen. Man vergleiche dazu auch die entsprechenden Beobachtungen von Heldt, Kreuzer et. al., die in [HKPP09] zu den dort vorgestellten Algorithmen gemacht werden.

Beispiel 4.17. *Es sei durch $\Xi = \{(1,1),(0,2),(1.1,1.1)\} \subset \mathbb{R}^2$ eine Menge von Punkten in der Ebene gegeben. Wir unterscheiden nun die folgenden Fälle:*

1. *Für $\varepsilon < 1/\sqrt{150}$ ist*

$$F = \{x_1(x_1 - x_2),\ x_2^2 + 0.45x_1 - 2.55x_2 + 1.1,\ x_1^2 - 1.55x_1 - 0.55x_2 + 1.1\}$$

   eine approximative H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ und es gilt $\mathfrak{V}(\langle F\rangle) = \mathfrak{V}(F) = \Xi$.

2. *Für $\varepsilon \geq 1/\sqrt{150}$ erhalten wir eine approximative H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ durch*

$$F = \{x_1 + x_2 - 2,\ x_1^2 - x_1x_2 + x_2^2 + 1.5x_1 - 1.5x_2 - 1\}.$$

   Offensichtlich hat sich die Mächtigkeit der H-Basis um Eins verringert. Für

die Varietät gilt

$$\mathfrak{V}(\langle F\rangle) = \mathfrak{V}(F) = \mathfrak{V}(\{x_1(x_1-1),\ x_1+x_2-2\}) = \{(1,1),(0,2)\}\,.$$

Der Punkt $(1.1, 1.1) \in \Xi$ *ist dabei verloren gegangen.*

Beide Situationen sind in Abbildung 4.1 graphisch dargestellt. Der Schwellenwert $\varepsilon = 1/\sqrt{150}$ *ergibt sich aus folgender Überlegung: Wir betrachten das Ideal*

$$\mathfrak{I}(\{(1,1),(0,2)\}) = \langle x_1(x_1-1), x_1+x_2-2\rangle$$

aller Polynome, die an den Punkten $(1,1)$ *und* $(0,2)$ *verschwinden. Damit der Punkt* $(1.1, 1.1)$ *von den Basispolynomen* $f_1(x_1,x_2) = x_1(x_1-1)$ *und* $f_2(x_1,x_2) = x_1+x_2-2$ *zumindest approximativ erreicht werden kann, muss* $|f_j(1.1,1.1)| \leq \varepsilon\|f_j\|_2$ *für* $j = 1,2$ *gelten. So erhält man durch*

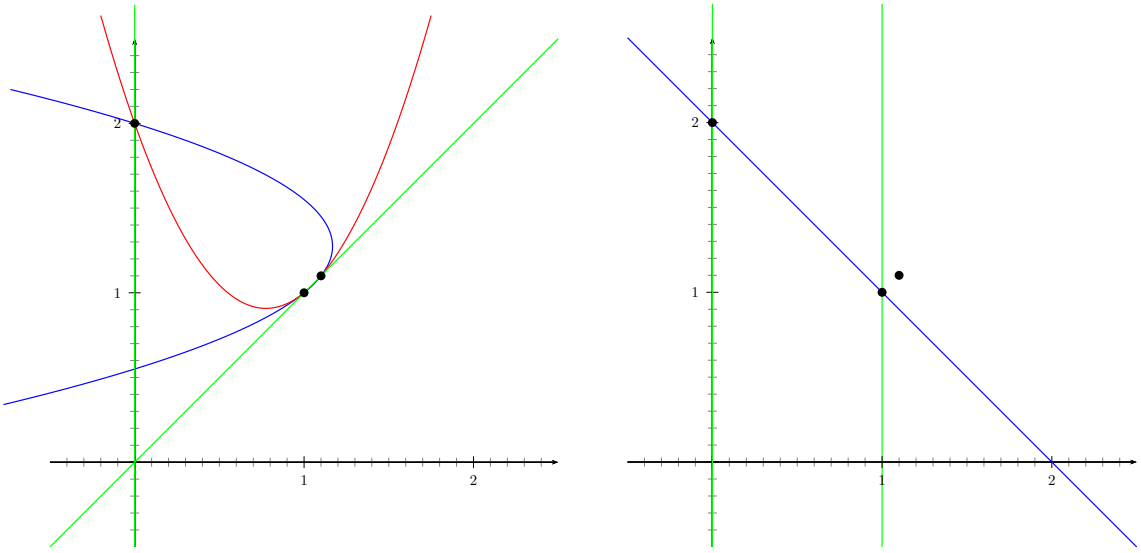
$$\frac{|f_1(1.1,1.1)|}{\|f_1\|_2} = \frac{0.11}{\sqrt{2}} < \frac{|f_2(1.1,1.1)|}{\|f_2\|_2} = \frac{0.2}{\sqrt{6}} = \frac{1}{\sqrt{150}} \leq \varepsilon$$

die kritische Toleranzschwelle.

Das gezeigte Beispiel wirft nun zwei weitere Fragen auf:

1. Warum wird das approximative Ideal so gebildet, dass $(1.1, 1.1)$ nur approximativ erreicht wird, während die beiden anderen Punkte exakt getroffen werden?
2. Wie werden die Punkte ausgewählt, an denen die Polynome exakt verschwinden sollen?

Diese beiden Fragestellungen werden in den nächsten Abschnitten ausführlich untersucht und es werden Methoden entwickelt, die ein anderes Verhalten aufweisen. Doch zunächst sei noch eine weitere Interpretation von Beispiel 4.17 bemerkt: Unter der Annahme, dass die Punkte aus einer fehlerbehafteten Abtastung einer Gerade stammen, liefert die Toleranz $\varepsilon > 1/\sqrt{150}$ zumindest die richtige geometrische Struktur im Sinne eines linearen Basispolynoms. Für $\varepsilon > 1/\sqrt{150}$ wird hingegen eine Basis aus drei quadratischen Polynomen bestimmt, was nach Satz 3.32 bedeutet, dass die Punkte *nicht* in einer niederdimensionalen Untermannigfaltigkeit liegen.



(a) $\varepsilon < 1/\sqrt{150}$ **(b)** $\varepsilon \geq 1/\sqrt{150}$

Abbildung 4.1.: Varietäten der Basispolynome aus Beispiel 4.17. Im linken Bild sieht man, dass die drei Polynome an allen Punkten verschwinden. Im rechten Bild ist die Toleranz hinreichend groß, sodass die Punkte $(1.1, 1.1)$ und $(1, 1)$ zusammenfallen. Die beiden Basispolynome verschwinden dabei an $(1.1, 1.1)$ nur approximativ.

Vermutet man also einen geometrischen Zusammenhang fehlerbehafteter Werte, so lässt sich folgende Vorgehensweise anwenden: Man beginne mit einer Toleranz $\varepsilon > 0$ nahe der Rechengenauigkeit und vergrößere ε solange, bis Basispolynome auftreten, die nach Satz 3.32 nicht dem Endlichkeitsanteil der H-Basis zuzuordnen sind. Eine offene Frage in obigem Beispiel bleibt jedoch, ob die Gerade durch den Punkt $(1, 1)$ oder durch $(1.1, 1.1)$ eine bessere Approximation an die abgetastete Gerade darstellt.

4.1.3. Wahl des Startpunkts

Eine noch offene Frage in Algorithmus 4.12 ist die Wahl des *Startpunkts* $\xi^{(0)} \in \Xi$. Dieser Punkt wird formal zur Konstruktion eines konstanten Polynoms benötigt, das anschließend entweder der Menge F^+ oder der Menge F^0 zugeordnet wird – je nachdem ob das Polynom an allen Punkten $\xi \in \Xi$ approximativ verschwindet. Da

alle konstanten Polynome $0 \neq f \in \Pi_{0,d}$ in ihrem Koeffizientenvektor nur *einen* von Null verschiedenen Wert haben, gilt für die Bedingung des approximativen Ideals

$$\frac{\|f(\Xi)\|_\infty}{\|f\|_2} = 1, \quad 0 \neq f \in \Pi_{0,d}.$$

Mit anderen Worten: Die Zugehörigkeit eines konstanten Polynoms zu F^+ oder F^0 hängt nur von der Toleranz ε und nicht vom Wert des Polynoms ab. Für $\varepsilon \geq 1$ liegt jedes konstante Polynom $f \neq 0$ in F^0, für $\varepsilon < 1$ entsprechend in F^+. Beachtet man, dass in Algorithmus 4.12 nur Toleranzen mit $0 < \varepsilon < 1$ zugelassen werden, muss das zu $\xi^{(0)} \in \Xi$ konstruierte Polynom stets in der Menge F^+ liegen – man vergleiche die Initialisierung von F_0^+ in Algorithmus 4.12. In diesem Sinne ist die Wahl des Startpunkts beliebig.

Der Punkt $\xi^{(0)} \in \Xi$ hat in Algorithmus 4.12 allerdings noch eine weitere Funktion: Dieser Punkt bildet das erste Element der Menge Ξ^+ und damit müssen alle Polynome, die im weiteren Verlauf der Menge F^0 hinzugefügt werden sollen, die *exakte* Bedingung

$$f(\xi^{(0)}) = 0, \quad f \in F^0, \tag{4.9}$$

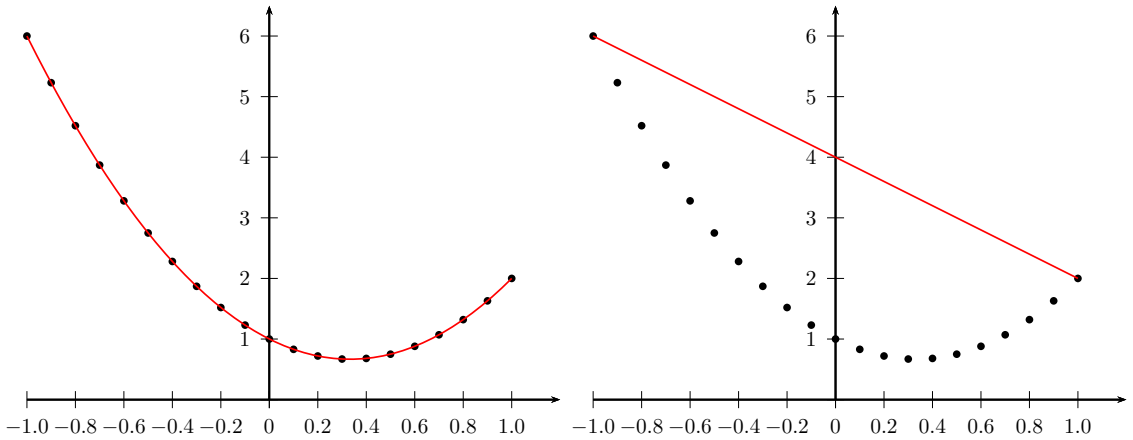
erfüllen. Das folgende Beispiel verdeutlicht den Einfluss der Forderung (4.9):

Beispiel 4.18. *Sei* $f(x_1, x_2) = 3x_1^2 - 2x_1 - x_2 + 1 \in \Pi_2$ *und eine Punktmenge*

$$\Xi = \{(x_1, f(x_1, 0)) : x_1 = -1, -0.9, \ldots, 1\} \subset \mathbb{R}^2.$$

Wir wählen $\xi^{(0)} = (-1, 6) \in \Xi$ *als Startwert und wenden Algorithmus 4.12 auf die Menge* Ξ *an. Dieser liefert für Toleranzen* $0 < \varepsilon \leq \frac{3}{\sqrt{21}}$ *das Polynom* $f(x_1, x_2)$ *als Polynom kleinsten Grades in der approximativen H-Basis. Das Ergebnis ist graphisch in Abbildung 4.2a dargestellt. Für* $\frac{3}{\sqrt{21}} < \varepsilon < 1$ *erhält man jedoch das lineare Polynom* $g_1(x_1, x_2) = -2x_1 - x_2 + 4$ *als Basispolynom kleinsten Grades. Zum Vergleich wird dieses Ergebnis in Abbildung 4.2b gezeigt. Der Schwellenwert* $\varepsilon = \frac{3}{\sqrt{21}}$ *ergibt sich dabei durch das Polynom* $g_1(x_1, x_2)$*, denn es gilt*

$$\frac{\|g_1(\Xi)\|_\infty}{\|g_1\|_2} = \frac{\max_{\xi \in \Xi} |g_1(\xi)|}{\sqrt{21}} = \frac{|g_1(0,1)|}{\sqrt{21}} = \frac{3}{\sqrt{21}}.$$



(a) $0 < \varepsilon \leq \frac{3}{\sqrt{21}}$. **(b)** $\frac{3}{\sqrt{21}} < \varepsilon < 1$.

Abbildung 4.2.: Punkte der Menge Ξ aus Beispiel 4.18 und Polynom kleinsten Grades der approximativen H-Basis für die angegebene Toleranz. Im ersten Fall liegen alle Punkte in der Varietät des Basispolynoms. Die Geometrie der Punkte wird also korrekt wiedergegeben. Im zweiten Fall erlaubt die größere Toleranz die approximative Darstellung durch eine lineare Funktion. Der größte Abstand zwischen Punktmenge und Geraden liegt dabei zwischen den Punkten $(0, 1)$ und $(0, 4)$.

Offensichtlich hängt $g_1(x_1, x_2)$ vom Startwert $\xi^{(0)}$ und einem weiteren Punkt $\xi^{(1)} \in \Xi$ ab, der durch den Algorithmus bestimmt wird. Damit beeinflusst der Startwert auch die Toleranzgrenze ε.

Wählen wir nun $\xi^{(0)} = (0, 1)$ als Startwert, so ergibt sich nach obigem Schema $f(x_1, x_2)$ für $\varepsilon < \frac{6}{\sqrt{27}}$ bzw. $g_2(x_1, x_2) = -5x_1 - x_2 + 1$ für $\varepsilon \geq \frac{6}{\sqrt{27}}$ als Basispolynome kleinsten Grades in der approximativen H-Basis. Die Toleranzschwelle berechnet sich dabei analog zum ersten Fall. Nun ist für die Konstruktion von g_2 jedoch $\varepsilon \geq \frac{6}{\sqrt{27}} > 1$ notwendig, was in approximativen Idealen per definitionem ausgeschlossen ist. Demnach kann für den Startwert $\xi^{(0)} = (0, 1)$ – unabhängig von der gewählten Toleranz – kein lineares Basispolynom in der approximativen H-Basis enthalten sein.

Das Beispiel hat gezeigt, dass die Toleranzschwelle zur Erkennung der Geometrie durch die Wahl des Startpunktes beeinflusst wird. Aus diesem Grund ist es nicht möglich, die Qualität einer Punktmenge im Bezug auf Störungen, Rauschen etc. nur

mit Hilfe von ε zu bewerten.

Die Wahl des optimalen Startwerts ist ebenfalls schwierig. Einerseits sollte der Punkt so exakt wie möglich sein, andererseits muss aber auch die Geometrie der Punktmenge, beispielsweise die Konvexität der Punkte in Beispiel 4.18, berücksichtigt werden. Ist ein exakter Punkt durch Rahmenbedingungen, Startpunkt der Messung, Aufhängungspunkt oder Ähnliches bekannt, so kann man durch die Wahl dieses Punktes zumindest die Interpolation an dieser Stelle garantieren. Verfügt man jedoch über keinerlei Informationen bzgl. der gegebenen Punktmenge, so bleiben nur heuristische Ansätze. Mögliche Kriterien dafür sind beispielsweise

1. Randpunkte der Menge Ξ im Bezug auf eine beliebige Norm oder
2. zentrale Punkte durch Minimierung des Abstandes zu einem Mittelwert der Menge Ξ.

Die einfachste Möglichkeit besteht sicherlich darin, den ersten Punkt der Menge Ξ in der Matrixdarstellung zu verwenden. Dabei sollte allerdings beachtet werden, dass damit die Wahl des Startpunktes nur auf die Anordnung der Zeilen in der Matrix Ξ verschoben wird, vgl. Abschnitt 4.1.1.

Ist der gewählte Startwert ein *Datenausreißer*, d. h. ein Punkt, der stark fehlerbehaftet ist, so erhält man möglicherweise eine sehr schlechte Approximation an die übrigen Punkte und entsprechend große Toleranzschwellen. Wir werden dies in Abschnitt 6.4 an einem einfachen Beispiel untersuchen und Vergleiche mit Verfahren durchführen, die *nicht* von einem Startpunkt abhängen.

Betrachten wir nun Beispiel 4.18 erneut, so wird deutlich, dass auch für exakte Startwerte nicht notwendigerweise die bestmögliche Approximation gewählt wird. Fixieren wir den Startpunkt $\xi^{(0)} = (-1, 6)$ und betrachten die Gerade g, die diesen Startwert mit einem anderen Punkt der Menge $\xi^{(1)} \in \Xi \backslash \{\xi^{(0)}\}$ verbindet, so erhalten wir für die Approximationsgüte im Sinne des Kriteriums

$$\frac{\|g(\Xi)\|_\infty}{\|g\|_2}$$

die in Tabelle 4.1 dargestellten Werte. In Beispiel 4.18 wurde $\xi^{(1)} = (1, 2)$ gewählt,

$\xi^{(1)}$	$\|g(\Xi)\|_\infty$	$\|g\|_2$	$\frac{\|g(\Xi)\|_\infty}{\|g\|_2}$
(-0.9, 5.23)	11.40	7.9486	1.4342
(-0.8, 4.52)	10.80	7.5974	1.4215
(-0.7, 3.87)	10.20	7.2540	1.4061
(-0.6, 3.28)	9.60	6.9195	1.3874
(-0.5, 2.75)	9.00	6.5955	1.3646
(-0.4, 2.28)	8.40	6.2833	1.3369
(-0.3, 1.87)	7.80	5.9850	1.3033
(-0.2, 1.52)	7.20	5.7026	1.2626
(-0.1, 1.23)	6.60	5.4387	1.2135
(0.0, 1.00)	6.00	5.1962	1.1547
(0.1, 0.83)	5.40	4.9780	1.0848
(0.2, 0.72)	4.80	4.7875	1.0026
(0.3, 0.67)	4.20	4.6282	0.9075
(0.4, 0.68)	3.60	4.5033	0.7994
(0.5, 0.75)	3.00	4.4159	0.6794
(0.6, 0.88)	2.40	4.3681	0.5494
(0.7, 1.07)	2.16	4.3612	0.4953
(0.8, 1.32)	2.43	4.3955	0.5528
(0.9, 1.63)	2.70	4.4699	0.6040
(1.0, 2.00)	3.00	4.5826	0.6547

Tabelle 4.1.: Toleranzschwellen für ε bei Interpolation am Startpunkt $\xi^{(0)} = (-1, 6)$ und dem jeweils angegebenen Punkt $\xi^{(1)} \in \Xi \setminus \{\xi^{(0)}\}$. Das Polynom g ist dabei im Sinne der *Zweipunkteform* einer Geraden konstruiert durch $g(x_1, x_2) = \frac{\xi_2^{(1)}-6}{\xi_1^{(1)}+1}(x_1 + 1) - x_2 + 6$.

obwohl der Punkt $\xi^{(1)} = (0.7, 1.07)$ zu einer niedrigeren Toleranzschwelle und somit einer besseren Approximation geführt hätte. Das Vorgehen in Algorithmus 4.12 minimiert daher *nicht notwendigerweise* die Bedingung

$$\min_{f \in \Pi_{1,2}} \frac{\|f(\Xi)\|_\infty}{\|f\|_2}, \quad f\left(\xi^{(0)}\right) = f\left(\xi^{(1)}\right) = 0, \tag{4.10}$$

sondern erfüllt lediglich beide Nebenbedingungen.

Abschließend folgt noch eine Untersuchung von Algorithmus 4.12 bzgl. der Wahl des Punktes $\xi^{(1)}$ in Abhängigkeit vom Startwert $\xi^{(0)}$. Dazu betrachten wir exemplarisch

den Fall $d = 2$. Sei eine endliche Menge $\Xi \subset \mathbb{R}^2$ gegeben und der Startwert als

$$\xi^{(0)} \leftarrow \theta = (\theta_1, \theta_2) \in \Xi$$

festgelegt. Da ein approximatives Ideal für $\varepsilon \geq 1$ trivial wird – man vergleiche dazu die Bemerkung nach Beispiel 4.3 – sei außerdem noch $\varepsilon < 1$ vorausgesetzt.

Zunächst berechnet der Algorithmus eine Basis G_θ des linearen Raums $\mathcal{W}^0_{1,2}(\emptyset)$, da aufgrund der oben beschriebenen Wahl von ε kein konstantes Polynom im Ideal liegt. Dabei werden die Basispolynome so gewählt, dass $G_\theta(\theta) = 0$ gilt. Dies ist beispielsweise für $G_\theta = \{x_1 - \theta_1, x_2 - \theta_2\}$ erfüllt und die Koeffizientenmatrix dazu lautet

$$G_\theta = \begin{bmatrix} -\theta_2 & 1 & 0 \\ -\theta_1 & 0 & 1 \end{bmatrix}. \tag{4.11}$$

Im nächsten Schritt wird daraus eine Orthonormalbasis F_θ konstruiert. Die Orthogonalisierung können wir hier rein formal nach dem Verfahren von Gram und Schmidt (vgl. [FH07, 4.7], [SK11, 11.4]) durchführen. In der Implementierung sollte stattdessen ein numerisch stabileres Verfahren, wie z. B. *Householder Transformationen*, verwendet werden. Wir erhalten somit

$$F_\theta = \begin{bmatrix} \sqrt{\theta_2^2 + 1} & 0 \\ 0 & \sqrt{\theta_1^2 + \theta_1^2\theta_2^2 + (\theta_2^2 + 1)^2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\theta_2 & 1 & 0 \\ -\theta_1 & -\theta_1\theta_2 & (\theta_2^2 + 1) \end{bmatrix}.$$

Zerlegen wir nun die Vandermonde-Matrix $F_\theta(\Xi \setminus \theta)$ im Sinne von Satz 4.11, so erhalten wir eine orthogonale Matrix Q und eine Permutationsmatrix P mit

$$Q^T \cdot F_\theta(\Xi \setminus \theta) \cdot P = \begin{bmatrix} \alpha & * \\ 0 & * \end{bmatrix}. \tag{4.12}$$

Aufgrund der in Algorithmus 4.6 gewählten Pivotstrategie gilt dabei

$$|\alpha| = \max_{j=1,\ldots,\#\Xi-1} \left\| \left(F_\theta(\Xi \setminus \theta)\right)_j \right\|_2 = \max_{\xi \in \Xi \setminus \theta} \|F_\theta(\xi)\|_2 . \tag{4.13}$$

Sei nun $f_2 \in \Pi_{1,2}$ das Polynom, das man durch $[f_1^T, f_2^T]^T = Q^T \cdot F_\theta$ erhält, dann muss

f_2 wegen (4.11) und (4.12) an den Stellen $\xi^{(0)} = \theta$ und $\xi^{(1)} = \arg\max_{\xi \in \Xi \setminus \theta} \|F_\theta(\xi)\|_2$ verschwinden. Dabei können wir die zu maximierende Norm in (4.13) durch

$$\begin{aligned}\|F_\theta(\xi)\|_2^2 &= \left\{ \frac{(\xi_2 - \theta_2)^2}{\theta_2^2 + 1} + \frac{((\theta_2^2+1)\xi_1 - (\theta_1\theta_2)\xi_2 - \theta_1)^2}{\theta_1^2 + \theta_1^2\theta_2^2 + (\theta_2^2+1)^2} \right\} \\ &= \frac{1}{(\theta_1^2 + \theta_2^2 + 1)} \left(\xi^T \underbrace{\begin{bmatrix} \theta_2^2+1 & -\theta_1\theta_2 \\ -\theta_1\theta_2 & \theta_1^2+1 \end{bmatrix}}_{=:A} \xi - 2\xi^T \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{=:b} + \underbrace{(\theta_1^2 + \theta_2^2)}_{=:c} \right)\end{aligned}$$

als quadratische Funktion $\mathbb{R}^2 \to \mathbb{R}$ in ξ auffassen.

Die Punkte $(\xi_1, \xi_2, \|F_\theta(\xi)\|_2^2) \in \mathbb{R}^3$ beschreiben dann eine Hyperfläche der Ordnung 2, für die wir eine *Hauptachsentransformation* (siehe [Fis05, 5.7], [SK11, 5.1.3]) durchführen können. Zunächst bemerken wir jedoch, dass $(\theta_1^2+\theta_2^2+1) > 0$ als ein positiver Faktor nur den Wert des Maximums, nicht aber dessen Lage verändert. Das Gleiche gilt für die Addition von c, sodass zur Bestimmung von $\arg\max_{\xi \in \Xi \setminus \theta} \|F_\theta(\xi)\|_2$ die Betrachtung von $\xi^T A\xi - 2\xi^T b$ ausreicht.

Die Matrix A hat die Eigenwerte $\lambda_1 = 1$, $\lambda_2 = \theta_1^2 + \theta_2^2 + 1$ und Eigenvektoren $v_1 = [\theta_1, \theta_2]^T$, $v_2 = [-\theta_2, \theta_1]^T$, d. h. die Drehmatrix

$$V := \frac{1}{\sqrt{\theta_1^2 + \theta_2^2}} \begin{bmatrix} \theta_1 & -\theta_2 \\ \theta_2 & \theta_1 \end{bmatrix}$$

diagonalisiert die Matrix A. Wir erhalten

$$\begin{aligned}\left(\xi^T V \begin{bmatrix} 1 & 0 \\ 0 & \theta_1^2 + \theta_2^2 + 1 \end{bmatrix} V^T \xi - 2\xi^T V V^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right) &= \widehat{\xi}^T \begin{bmatrix} 1 & 0 \\ 0 & (\theta_1^2 + \theta_2^2 + 1) \end{bmatrix} \widehat{\xi} \\ &= \left\| \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{\theta_1^2 + \theta_2^2 + 1} \end{bmatrix} \widehat{\xi} \right\|_2^2\end{aligned}$$

mit

$$\widehat{\xi} = \frac{1}{\sqrt{\theta_1^2 + \theta_2^2}} \begin{bmatrix} \theta_1 & \theta_2 \\ -\theta_2 & \theta_1 \end{bmatrix} \cdot \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} - \begin{bmatrix} \sqrt{\theta_1^2 + \theta_2^2} \\ 0 \end{bmatrix}.$$

Zusammenfassend lässt sich feststellen, dass Algorithmus 4.12 ein lineares Polynom bestimmt, das in den Punkten $\xi^{(0)} = \theta$ und

$$\xi^{(1)} = \arg\max_{\xi \in \Xi \setminus \theta} \left\| \frac{1}{\sqrt{\theta_1^2 + \theta_2^2}} \begin{bmatrix} \theta_1 & \theta_2 \\ -\theta_2 & \theta_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{\theta_1^2 + \theta_2^2 + 1} \end{bmatrix} \cdot \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} - \begin{bmatrix} \sqrt{\theta_1^2 + \theta_2^2} \\ 0 \end{bmatrix} \right\|_2$$

verschwindet und dadurch eindeutig festgelegt ist. Mit anderen Worten: Der Punkt $\xi^{(1)}$ entsteht nach Skalierung, Drehung und Verschiebung in Abhängigkeit vom Startwert $\xi^{(0)}$ als maximaler Punkt der Menge Ξ bzgl. der euklidischen Norm.

Damit erklärt sich die Wahl des Punktes $\xi^{(1)} = (1, 2)$ in Beispiel 4.18. Zudem sieht man, dass die in Algorithmus 4.12 verwendete QRP-Zerlegung das Minimierungsproblem (4.10) nicht lösen kann. Dies gilt unabhängig von der in (4.13) gewählten Pivotstrategie, denn die Pivotisierung kann nur auf Grundlage der Vandermonde-Matrix $F_\theta(\Xi \setminus \theta)$ durchgeführt werden. Die Minimierung in (4.10) erfordert aber Kenntnis über die Norm der zweiten Zeile von $Q^T \cdot F_\theta(\Xi \setminus \theta)$, d. h. über die Vandermonde-Matrix *nach* der orthogonalen Transformation. Da die Multiplikation mit orthogonalen Matrizen von links zwar die Norm der Spalten, nicht aber die Norm der Zeilen erhält, können wir über $\|Q^T \cdot F_\theta(\Xi \setminus \theta)_2\|_\infty$ nichts aussagen.

4.1.4. Interpolation vs. Approximation

Ein wichtiges Merkmal von Algorithmus 4.12 besteht darin, dass für die Polynome F_n nach Konstruktion stets $F_n(\Xi_j^+) = 0$, $j = 0, \ldots, n-1$, gilt, vgl. Zeile 13 in Algorithmus 4.12. Es findet also eine schrittweise Interpolation der Punktmenge Ξ im Sinne einer Newton-Darstellung (siehe [GS00], für den univariaten Fall vergleiche man [SK11, 3.1.3], [SW05, 8.6]) statt. Dies reduziert natürlich die Anzahl der Freiheitsgrade, die für die Wahl von F_n zur Verfügung stehen. Dadurch besteht die Möglichkeit, dass potentielle Kandidaten für F_n nicht konstruiert werden können und somit Basispolynome niedrigen Grades übersprungen werden. Dieses Verhalten kann man an der Punktmenge aus Beispiel 4.18 bzw. den Abbildungen 4.2a und 4.2b nachvollziehen.

Beispiel 4.19. *Seien analog zu Beispiel 4.18 das Polynom* $f(x_1, x_2) = 3x_1^2 - 2x_1 -$

$x_2 + 1$ *und eine diskrete Abtastung durch*

$$\Xi := \left\{ \Big(x_1, f(x_1, 0) \Big) : x_1 = -1, -0.9, \ldots, 0.9, 1 \right\} \subset \mathbb{R}^2$$

gegeben. In Beispiel 4.18 wurde gezeigt, dass für den Startwert $\xi^{(0)} = (-1, 6) \in \Xi$ *und eine Toleranz* $\varepsilon > \frac{3}{\sqrt{21}}$ *ein lineares Polynom in der durch Algorithmus 4.12 berechneten H-Basis des approximativen Ideals* $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ *liegt. Allerdings lässt sich das Polynom* f *nicht für alle diese* ε *durch die H-Basis darstellen, obwohl* $f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ *gilt.*

Für $\frac{3}{\sqrt{21}} < \varepsilon < 0.777$ *erhalten wir das Ideal*

$$\langle -2x_1 - x_2 + 4, g(x_1, x_2) \rangle , \quad \textit{mit } \deg(g) = 4,$$

das f nicht *enthält, da* $\Lambda(f) \notin \mathcal{V}^0_{2,2}(-2x_1 - x_2 + 4, g(x_1, x_2)) = \mathcal{V}^0_{2,2}(-2x_1 - x_2 + 4)$. *Wählen wir hingegen* $\varepsilon > 0.778$, *so berechnet Algorithmus 4.12 eine H-Basis des Ideals*

$$\left\langle -2x_1 - x_2 + 4, x_1^2 - 2x_1x_2 + 4x_2^2 + \frac{600}{7}x_1 + \frac{48}{7}x_2 - \frac{787}{7} \right\rangle \ni f.$$

Damit die H-Basis ein quadratisches Basispolynom $\widetilde{f} \in \Pi_2$ *enthalten kann, muss für die Leitform* $\Lambda(\widetilde{f}) \in \mathcal{W}^0_{2,2}(-2x_1 - x_2 + 4)$ *gelten, vgl. Satz 3.18. Da für diesen homogenen Raum* $\dim(\mathcal{W}^0_{2,2}(-2x_1 - x_2 + 4)) = 1$ *gilt, sind damit bereits zwei der sechs Freiheitsgrade des Polynoms* $\widetilde{f}$ *besetzt. Das lineare Polynom* $-2x_1 - x_2 + 4$ *verschwindet an zwei Stellen von* Ξ *(siehe Abbildung 4.2b), also muss auch* $\widetilde{f}(x_1, x_2)$ *an diesen Stellen interpolieren. Dafür werden drei weitere Freiheitsgrade benötigt. Letztlich bleiben für* $\widetilde{f}(x_1, x_2)$ *nur noch das Polynom*

$$\widetilde{f}(x_1, x_2) = x_1^2 - 2x_1x_2 + 4x_2^2 + \frac{600}{7}x_1 + \frac{48}{7}x_2 - \frac{787}{7}$$

oder skalare Vielfache davon zur Auswahl. Für alle Polynome der Form $\lambda\widetilde{f}$, $\lambda \neq 0$, *gilt aber*

$$\frac{\|\lambda\widetilde{f}(\Xi)\|_\infty}{\|\lambda\widetilde{f}\|_2} = \frac{\max_{\xi\in\Xi} |\widetilde{f}(\xi)|}{\|\widetilde{f}\|_2} = \frac{1927107}{2500\sqrt{982702}} \approx 0.7775976.$$

Damit wird für $\frac{3}{\sqrt{21}} < \varepsilon < 0.777$ *kein quadratisches Basispolynom konstruiert und das Polynom* f *kann* nicht *durch die berechnete H-Basis dargestellt werden.*

Diese *Lücke im Toleranzbereich* lässt sich schließen, indem man auf die Interpolationseigenschaft $F_n(\Xi_j^+) = 0$, $j = 0, \ldots, n-1$, verzichtet und somit zusätzliche Freiheitsgrade in F_n erhält. Umgekehrt bedeuten zusätzliche Freiheitsgrade aber eine größere Anzahl an Polynomen F_n und somit mehr Zeilen in der Vandermonde-Matrix $F_n(\Xi_n)$. Durch die fehlende Interpolation ist auch $\Xi_n = \Xi$ für alle Schritte n anzunehmen, da die a priori-Information $F_n(\Xi_j^+) = 0$ fehlt. Damit erhöht sich die Spaltenanzahl von $F_n(\Xi_n)$ ebenfalls. Insgesamt führt das zu einem deutlich höheren Rechenaufwand in der QRP-Zerlegung der Vandermonde-Matrix.

Für eine entsprechende Modifikation von Algorithmus 4.12 wird eine Methode zur Umformung von Matrizen in *Zeilenstufenform* (siehe beispielsweise [Fis05, 0.4]) benötigt. Einen ähnlichen Ansatz verwenden auch Heldt, Kreuzer, Pokutta und Poulisse in [HKPP09, Lemma 3.2]. Dort basiert das Verfahren allerdings auf einer QR-Zerlegung, die mit Hilfe der Gram-Schmidt Orthogonalisierung durchgeführt wird. Da diese Methode ohne eine geeignete Modifikation numerisch instabil ist (siehe dazu [Bjö94]), generieren wir die Zeilenstufenform mit Hilfe einer QR-Zerlegung durch *Householder Transformationen*, die auch schon in der QRP-Zerlegung verwendet wurden, vgl. Algorithmus 4.6. Im Vergleich zur klassischen QR-Zerlegung müssen dazu nur wenige Zeilen im Pseudocode des Verfahrens ergänzt werden. Das Ergebnis ist in Algorithmus 4.20 dargestellt. Ein weiterer Unterschied zu dem in [HKPP09] gezeigten Verfahren besteht darin, dass wir keine *reduzierte* Zeilenstufenform benötigen. Dadurch entfällt die Elimination der Einträge oberhalb der Pivotelemente.

Auf einen Beweis der Terminierung und die Verifikation von Algorithmus 4.20 sei an dieser Stelle verzichtet. Beides folgt analog zu den entsprechenden Resultaten bzgl. der QR-Zerlegung. Unter Verwendung dieses Verfahrens lässt sich die Bestimmung einer ∞-approximativen H-Basis ohne schrittweise Interpolation formulieren. Wir definieren dazu noch die folgende Notation:

Definition 4.21. *Sei* $A \in \mathbb{R}^{n \times m} = (a_1, \ldots, a_m)$ *eine Matrix mit den Spalten* a_j*, dann ist* $\overleftarrow{A} := (a_m, \ldots, a_1)$ *die Matrix, die durch Umkehrung der Spaltenreihenfolge in* A *entsteht.*

Algorithmus 4.20 : Zeilenstufenform via QR-Zerlegung

Input : $A \in \mathbb{R}^{m\times n}$, $m \leq n$, $\varepsilon > 0$
Output : Orthogonale Matrix $Q \in \mathbb{R}^{m\times m}$, Zeilenstufenmatrix $R \in \mathbb{R}^{m\times n}$

$Q_1 \leftarrow I_{m\times m}$
$A_1 \leftarrow A$
for $k = 1, \ldots, m$ **do**
  $j \leftarrow k$
  while $\|(A)_{k:m,j}\|_2 < \varepsilon$ & $j < n$ **do**
    $j \leftarrow j + 1$
  end
  $y \leftarrow (A)_{k:m,j}$
  $(y)_1 \leftarrow (y)_1 + \operatorname{sign}((y)_1)\|y\|$
  $y \leftarrow y/\|y\|_2$
  $H_k \leftarrow I_{m-k\times m-k} - 2yy^T$
  $Q_{k+1} \leftarrow Q_k \cdot \begin{bmatrix} I & 0 \\ 0 & H_k \end{bmatrix}$
  $A_{k+1} \leftarrow \begin{bmatrix} I & 0 \\ 0 & H_k \end{bmatrix} \cdot A_k$
end
$R \leftarrow A_m$, $Q \leftarrow Q_m$

Beispiel 4.22.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \implies \overleftarrow{A} = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$$

Damit lässt sich das Verfahren von Sauer aus Algorithmus 4.12 zu der in Algorithmus 4.23 beschriebenen Variante ohne Startwertabhängigkeit modifizieren.

Ein für die Fehleranalyse wesentlicher Unterschied der beiden Verfahren besteht darin, dass die mittels QRP-Zerlegung bestimmten Polynome in Algorithmus 4.23 noch weiterverarbeitet werden. Wir können also nicht davon ausgehen, dass die konstruierte Eigenschaft $\|f(\Xi)\|_\infty \leq \varepsilon\|f\|_2$ noch gilt. Durch die Umformung in *Zeilenstufenform* verändert sich auch der Wert der Polynome an den Auswertungsstellen Ξ, sodass man lediglich eine approximative H-Basis von $\mathfrak{I}_{\infty,\varepsilon'}(\Xi)$ erwarten kann. Dieses Verhalten wurde auch von Heldt, Kreuzer, Pokutta und Poulisse im Zusammenhang mit ähnlichen Algorithmen untersucht, vgl. [HKPP09]. Der folgende Satz liefert ne-

Algorithmus 4.23 : Bestimmung einer approximativen H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ *ohne* Interpolation

Input : Endliche Punktmenge $\Xi \subset \mathbb{R}^d$, Toleranz $0 < \varepsilon < 1$
Output : ∞-approximative H-Basis $F^0 := \bigcup_{k=1}^{n-1} F_k^0$, $m := \max_{k=1,\dots,n-1} \#F_k^-$

1 $F_0^0 \leftarrow \emptyset$
2 $F_0^+ \leftarrow 1$
3 $n \leftarrow 1$
4 **while** 1 **do**
5 Bestimme eine Basis G_n von $\mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0)$.
6 **if** $G_n = \emptyset$ **then break**;
7 $F_n \leftarrow \begin{pmatrix} G_n \\ I_{\#\boldsymbol{T}_{n-1,d} \times \#\boldsymbol{T}_{n-1,d}} \quad 0_{\#\boldsymbol{T}_{n-1,d} \times \#\boldsymbol{T}_{n-1,d}^0} \end{pmatrix}$
8 Bestimme QRP Zerlegung $F_n(\Xi) = QRP^T$.
9 $j \leftarrow \max\{j : |R_{j,j}| > \varepsilon\}$
10 $F_n^+ \leftarrow [(Q^T \cdot F_n)_k : k \le j]$
11 $F_n^- \leftarrow [(Q^T \cdot F_n)_k : k > j]$
12 Zerlege $\overleftarrow{F_n^-} = \widetilde{Q} \cdot L$ mit L in Zeilenstufenform
13 $L \leftarrow \overleftarrow{L}$
14 $j \leftarrow \max\{j : |L_{j,\binom{d+n-1}{n-1}+1}| > 0\}$
15 $F_n^0 \leftarrow [L_k : k \le j]$
16 $n \leftarrow n + 1$
17 **end**

ben der Verifikation des Algorithmus auch eine Abschätzung für ε', wobei wir in diesem Punkt einem Argument aus [HKPP09] folgen:

Satz 4.24. *Algorithmus 4.23 terminiert und liefert eine approximative H-Basis von* $\mathfrak{I}_{\infty,\sqrt{m}\varepsilon}(\Xi)$.

Beweis. Das Verfahren terminiert genau dann, wenn $G_n = \emptyset$ für ein $n \in \mathbb{N}$ gilt. Ist in Schritt n noch kein Abbruch erfolgt, so muss $\dim(G_k) \ge 1$ für alle $k \le n$ gelten. Da wir stets die Endlichkeit der Menge $\Xi \subset \mathbb{R}^d$ annehmen, sei hier $\#\Xi = N < \infty$. Durch die Konstruktion der Polynome F_n in Zeile 7 von Algorithmus 4.23 gilt $\dim(F_n) = \#\boldsymbol{T}_{n-1,d} + \dim(G_n)$ und nach der QRP-Zerlegung erhalten wir

$$\dim(F_n^-) \ge \#\boldsymbol{T}_{n-1,d} + \dim(G_n) - N, \tag{4.14}$$

da die Vandermonde-Matrix $F_n(\Xi)$ genau N Spalten hat und somit auch die Matrix R in der QRP-Zerlegung höchstens N von der Nullzeile verschiedene Zeilen haben kann. Die Konstruktion der Zeilenstufenform verändert den Rang von F_n^- nicht, sodass $\dim(L) = \dim(F_n^-)$ gilt. Zudem zerfällt die Matrix L zeilenweise in $n+1$ Blöcke, die jeweils Polynome vom Grad $k \leq n$ enthalten – man vergleiche dazu Abbildung 4.3. Die Blöcke $k = 0, \ldots, n-1$ können dabei höchstens $\#\boldsymbol{T}_{k,d}^0 - 1$ Zeilen enthalten, denn für $\#\boldsymbol{T}_{k,d}^0$ Polynome vom Grad k hätte bereits $G_k = \emptyset$ sein müssen. Die Zeilen aus L, die Polynomen vom Grad n entsprechen, sind genau die Zeilen von F_n^0. Zusammen mit (4.14) erhalten wir die Abschätzung

$$\begin{aligned} \dim(F_n^0) &\geq \dim(F_n^-) - \left(\sum_{k=0}^{n-1} (\#\boldsymbol{T}_{k,d}^0 - 1)\right) && (4.15) \\ &\geq (\#\boldsymbol{T}_{n-1,d} + \dim(G_n) - N) - (\#\boldsymbol{T}_{n-1,d} - n) = \dim(G_n) - N + n. && (4.16) \end{aligned}$$

Für $n \geq N$ ist damit $\dim(F_n^0) \geq \dim(G_n)$, was direkt zu $G_{n+1} = \emptyset$ führt. Somit terminiert Algorithmus 4.23 nach höchstens N Schritten.

Analog zu Algorithmus 4.12 gilt in Schritt n des Verfahrens $F_n^- \subseteq \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ durch die Konstruktion der QRP-Zerlegung. Für $f \in F_n$ ist hier jedoch nur $\deg(f) \leq n$, sodass wir

$$\begin{aligned} \Lambda(F_n^-) \cap \Pi_{n,d}^0 &\subseteq \operatorname{span}\{\Lambda(F_n)\} \cap \Pi_{n,d}^0 = \operatorname{span}\left\{\Lambda(F_n) \cap \Pi_{n,d}^0\right\} \\ &\subseteq \mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0) \end{aligned} \tag{4.17}$$

erhalten. Die zweite Zerlegung liefert eine linke obere Dreiecksmatrix L in Zeilenstufenform, vgl. auch Abbildung 4.3, durch $L = \widetilde{Q}^T \cdot F_n^-$ mit $\operatorname{span} L = \operatorname{span} F_n^-$. Nun werden für F_n^0 nur die Zeilen der Matrix L ausgewählt, die Polynomen vom Grad n entsprechen. Damit gilt dann wieder die gewünschte Beziehung $\Lambda(F_n^0) \subseteq \mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0)$ und die Polynome $f \in F_n^0$ sind orthogonal bzgl. des monomialen Skalarprodukts und normiert.

Seien nun f_j^-, $j = 1, \ldots, m_n$, die Polynome aus der Menge F_n^-, dann gibt es für $f \in F_0$ eine Spalte q der Matrix Q, sodass $f = \sum_{j=1}^{m_n} q_j f_j^-$ gilt. Damit folgt die

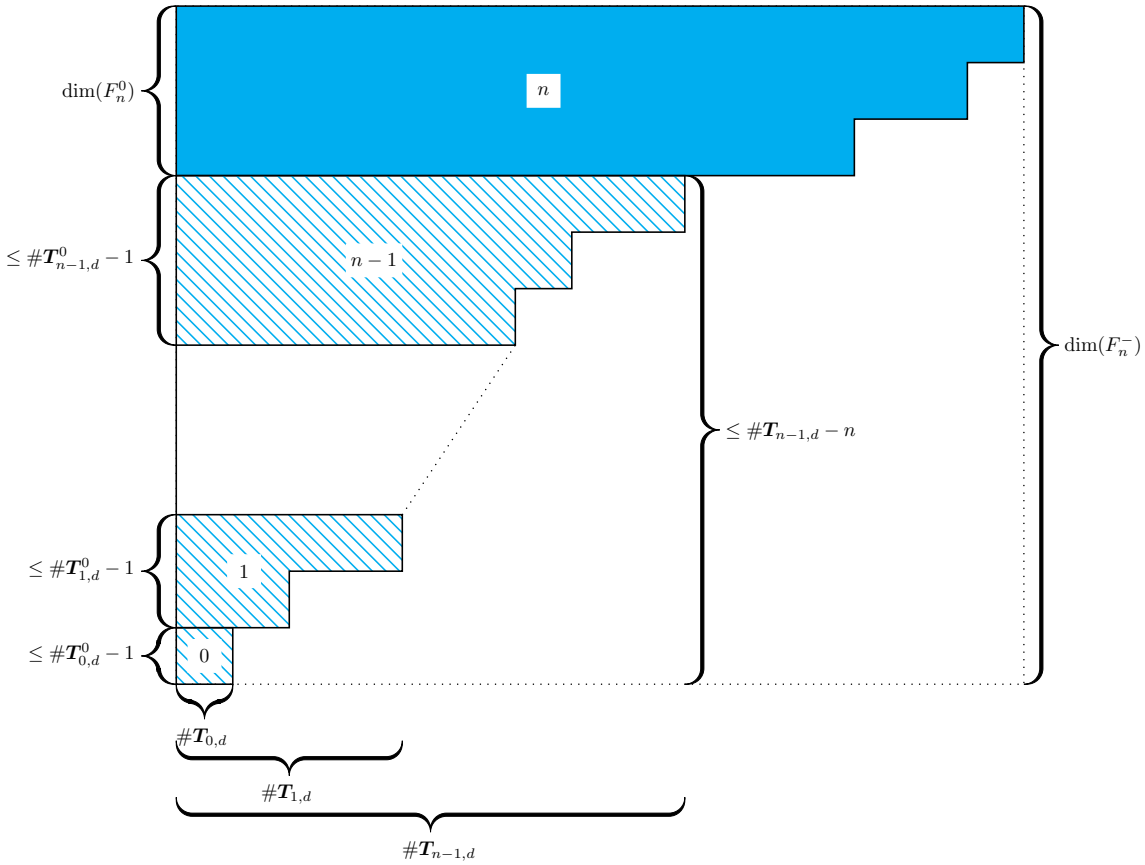


Abbildung 4.3.: Aufbau einer Matrix L aus Algorithmus 4.23 mit den Blöcken $k = 0, \ldots, n$. Die schraffierten Zeilen entsprechen Polynomen vom Grad kleiner n, die für F_n^0 nicht berücksichtigt werden. Die Ungleichung (4.15) lässt sich direkt aus dieser Skizze ablesen.

Abschätzung

$$\begin{aligned}\|f(\Xi)\|_\infty = \left\|\sum_{j=1}^{m_n} q_j f_j^-(\Xi)\right\|_\infty &\leq \sum_{j=1}^{m_n} |q_j| \left\|f_j^-(\Xi)\right\|_\infty \leq \varepsilon \sum_{j=1}^{m_n} |q_j| = \varepsilon \|q\|_1 \\ &\leq \varepsilon\sqrt{m_n}\|q\|_2 = \sqrt{m_n}\varepsilon.\end{aligned}$$

Wir erhalten also in Schritt n eine Menge $F_n^0 \subseteq \mathfrak{I}_{\infty,\sqrt{m_n}\varepsilon}(\Xi)$. Zudem gilt per definitionem die Beziehung $\mathfrak{I}_{p,\varepsilon_2}(\Xi) \subseteq \mathfrak{I}_{p,\varepsilon_1}(\Xi)$ für zwei p-approximative Ideale $\mathfrak{I}_{p,\varepsilon_1}(\Xi)$ und $\mathfrak{I}_{p,\varepsilon_2}(\Xi)$ mit $\varepsilon_1 > \varepsilon_2 > 0$. Es reicht also aus, die Menge F_n^- mit den meisten Elementen zu betrachten. Damit können wir insgesamt $m = \max_{k\in\mathbb{N}}\{\#F_k^-\}$ und

$F_0 = \bigcup F_k^0 \subset \mathfrak{I}_{\infty,\sqrt{m}\varepsilon}$ annehmen. □

Somit liefern beide Varianten die gesuchte approximative H-Basis. Letztlich hängt es von der Anwendung ab, ob man die schnellere Berechnung durch Algorithmus 4.12 verwendet und dafür in Kauf nimmt, dass möglicherweise Polynome übergangen werden. Ist jedoch die Einhaltung der vorgegebenen Toleranz ε von Bedeutung, so ist Algorithmus 4.12 vorzuziehen, da der Faktor $\sqrt{m}$ in Algorithmus 4.23 erst *nach* der Berechnung der approximativen H-Basis bekannt ist. Anstelle der a priori-Schranke ε erhalten wir so nur die a posteriori-Schranke $\sqrt{m}\varepsilon$.

4.1.5. Alternative Zerlegung

In diesem Abschnitt wird eine weitere Modifikation von Algorithmus 4.12 vorgestellt, die es ermöglicht, eine H-Basis des 2-approximativen Ideals $\mathfrak{I}_{2,\varepsilon}(\Xi)$ zu bestimmen. In Satz 4.11 wurde gezeigt, dass die QRP-Zerlegung der Vandermonde-Matrix $F_n(\Xi)$ darüber entscheidet, ob ein Polynom $f \in F_n$ in $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ liegt oder nicht. Demnach wird hier ein entsprechendes Entscheidungskriterium für die Bedingung

$$\frac{\|f(\Xi)\|_2}{\|f\|_2} \leq \varepsilon \tag{4.18}$$

benötigt. Dazu verwenden wir eine Matrixzerlegung, die bereits in Kapitel 2 (vgl. Satz 2.44) benutzt wurde: die *Singulärwertzerlegung*. Für die Vandermonde-Matrix $F_n(\Xi)$ einer endlichen Menge von orthonormalen Polynomen $F_n \subset \Pi_{n,d} \setminus \Pi_{n-1,d}$ und einer Punktmenge $\Xi \subset \mathbb{R}^d$ erhalten wir durch die Singulärwertzerlegung eine Darstellung

$$F_n(\Xi) = U \cdot \begin{bmatrix} \sigma_1 & & 0 & 0 & \cdots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ 0 & & \sigma_p & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot V^T, \quad p = \operatorname{rank}(F_n(\Xi)), \tag{4.19}$$

wobei U und V orthogonale Matrizen sind. Nach Lemma 4.10 gilt $U^T \cdot F_n(\Xi) = (U^T \cdot F_n)(\Xi)$ und damit lässt sich die euklidische Norm der j-ten Zeile von $(U^T \cdot F_n)(\Xi)$ durch

$$\|(U^T \cdot F_n)_j(\Xi)\|_2 = \|\sigma_j V_j^T\|_2 = \sigma_j, \quad j = 1, \ldots, p, \tag{4.20}$$

angeben. Weiterhin wurde in Lemma 4.9 gezeigt, dass die Multiplikation der Koeffizientenmatrix F_n von links mit einer orthogonalen Matrix U die paarweise orthogonale Beziehung der Polynome $f \in F_n$ und deren Normierung erhält. Sei nun $j \in \{1, \ldots, p\}$ so gewählt, dass $\sigma_j \leq \varepsilon < \sigma_{j+1}$ gilt. Da die Singulärwerte $\sigma_1 > \cdots > \sigma_p$ in absteigender Reihenfolge sortiert sind, erfüllen alle Polynome $(U^T \cdot F)_{\widetilde{j}}$ mit $\widetilde{j} \geq j$ die Forderung (4.18).

Da die Matrix $(U^T \cdot F_n)(\Xi)$ im Allgemeinen voll besetzt ist, erhalten wir keine schrittweise Interpolation der Punkte Ξ. Die Polynome müssen an keinem der Punkte aus Ξ verschwinden. Daher ist ein Ansatz mit vorgegebenen Interpolationsstellen – analog zu Algorithmus 4.12 – nicht zielführend. Stattdessen können wir wie in Algorithmus 4.23 vorgehen und erhalten das in Algorithmus 4.26 angegebene Verfahren.

Satz 4.25. *Algorithmus 4.26 terminiert und liefert eine H-Basis des approximativen Ideals* $\mathfrak{I}_{2,\sqrt{m}\varepsilon}(\Xi)$.

Beweis. Da sich die Algorithmen 4.23 und 4.26 nur in der verwendeten Matrixzerlegung unterscheiden, folgt die Terminierung und die H-Basis Eigenschaft der Menge F_n^0 analog zu Satz 4.24. Die Menge F_n^- ist entsprechend den Konstruktionen in (4.19) und (4.20) gewählt, sodass $F_n^- \subset \mathfrak{I}_{2,\varepsilon}(\Xi)$ gilt. Durch den Übergang von F_n^- zu F_n^0 ändert sich dabei die Toleranz ε nach der aus Satz 4.24 bekannten Abschätzung. □

Algorithmus 4.26 : Bestimmung einer approximativen H-Basis von $\mathfrak{I}_{2,\varepsilon}(\Xi)$

Input : Punktmenge $\Xi \subset \mathbb{R}^d$, Toleranz $0 < \varepsilon < 1$
Output : 2-approximative H-Basis $F^0 := \bigcup_{k=1}^{n-1} F_k^0$, $m := \max_{k=1,\dots,n-1} \#F_n^-$

$F_0^0 \leftarrow \emptyset$
$F_0^+ \leftarrow 1$
$n \leftarrow 1$
while 1 **do**
  Bestimme eine Basis G_n von $\mathcal{W}_{n,d}^0(F_0^0 \cup \cdots \cup F_{n-1}^0)$.
  if $G_n = \emptyset$ **then break**;
  $F_n \leftarrow \begin{pmatrix} G_n \\ I_{\#\boldsymbol{T}_{n-1,d} \times \#\boldsymbol{T}_{n-1,d}} \quad 0_{\#\boldsymbol{T}_{n-1,d} \times \#\boldsymbol{T}_{n-1,d}^0} \end{pmatrix}$
  Bestimme Singulärwertzerlegung $F_n(\Xi) = U \cdot \Sigma \cdot V^T$
  $j \leftarrow \max\{j : \Sigma_{j,j} > \varepsilon\}$
  $F_n^+ \leftarrow [(U^T \cdot F)_k : k \leq j]$
  $F_n^- \leftarrow [(U^T \cdot F)_k : k > j]$
  Zerlege $\overleftarrow{F_n^-} = Q \cdot R$ mit R in Zeilenstufenform
  $L \leftarrow \overleftarrow{R}$
  $j \leftarrow \max\{j : |L_{j,\#\boldsymbol{T}_{n-1,d}+1}| > 0\}$
  $F_n^0 \leftarrow [L_k : k \leq j]$
  $n \leftarrow n + 1$
end

4.2. Ringoperationen auf approximativen Idealen

Im folgenden Abschnitt soll untersucht werden, wie sich die Rechenoperationen des Rings Π_d auf die Polynome eines approximativen Ideals auswirken. Da approximative Ideale für $\varepsilon > 0$ keine Ideale sind, kann die Abgeschlossenheit bzgl. der Addition zweier Polynome des approximativen Ideals und der Multiplikation mit einem Polynom des Rings nicht vorausgesetzt werden. In Beispiel 4.3 wurde deutlich, dass bereits die Multiplikation mit einem Term genügt, um die definierende Eigenschaft des approximativen Ideals zu verletzen. Daher werden in diesem Abschnitt Kriterien entwickelt, unter denen sich die Vergrößerung der Toleranz ε unter Ringoperationen beschränken lässt.

4.2.1. Addition und Linearkombination

Ein Kriterium zur Fehlerabschätzung für die Addition zweier Polynome aus einem approximativen Ideal wurde bereits von Sauer in [Sau07, Lemma 1] bemerkt. Unter der Annahme, dass die beiden Polynome orthogonal bzgl. des monomialen Skalarprodukts sind, gilt folgende Aussage:

Satz 4.27. *Seien $f, g \in \mathfrak{I}_{p,\varepsilon}(\Xi)$, $\varepsilon > 0$, $1 \leq p \leq \infty$, $\Xi \subset \mathbb{R}^d$ und $(f, g) = 0$, so folgt $(f + g) \in \mathfrak{I}_{p,\sqrt{2}\varepsilon}(\Xi)$.*

Zudem sind approximative Ideale *skalierungsinvariant*, was in der Normierung der Koeffizientennorm in Definition 4.2 begründet ist. Wir halten diese Eigenschaft in folgendem Lemma fest:

Lemma 4.28. *Sei $\Xi \subset \mathbb{R}^d$, $f \in \mathfrak{I}_{p,\varepsilon}(\Xi)$ und $\lambda \in \mathbb{R}$, dann ist $(\lambda f) \in \mathfrak{I}_{p,\varepsilon}(\Xi)$.*

Beweis. Aus der Linearität der p-Normen folgt

$$\frac{\|(\lambda f)(\Xi)\|_p}{\|\lambda f\|_2} = \frac{|\lambda| \cdot \|f(\Xi)\|_p}{|\lambda| \cdot \|f\|_2} = \frac{\|f(\Xi)\|_p}{\|f\|_2} \leq \varepsilon$$

und damit die Behauptung. □

Da das monomiale Skalarprodukt dem Standardskalarprodukt der Koeffizientenvektoren entspricht, überträgt sich auch dessen *Bilinearität*, vgl. [Fis05, 5.4]. Damit ergibt sich auf natürliche Weise eine Fehlerabschätzung für beliebige Linearkombinationen paarweise orthogonaler Polynome aus einem approximativen Ideal.

Satz 4.29. *Seien $\Xi \subset \mathbb{R}^d$ und $f_1, \dots, f_n \in \mathfrak{I}_{p,\varepsilon}(\Xi)$, $(f_i, f_j) = 0$ für $i \neq j$ gegeben. Ist weiter $f \in \operatorname{span}\{f_1, \dots, f_n\}$, so gilt*

$$f \in \mathfrak{I}_{p,\sqrt{\widetilde{n}}\varepsilon}(\Xi), \quad \widetilde{n} := 2^{\lceil \log_2(n) \rceil}.$$

Beweis. Zunächst ergänzen wir die Polynome $f_{n+1} = \cdots = f_{\widetilde{n}} = 0$. Da $f \in \operatorname{span}\{f_1, \dots, f_{\widetilde{n}}\}$ erhalten wir nach Satz 4.27 eine Zerlegung

$$f = \sum_{j=1}^{\widetilde{n}} \lambda_j f_j = \sum_{j=1}^{\widetilde{n}/2} \underbrace{(\lambda_{2j} f_{2j} + \lambda_{2j-1} f_{2j-1})}_{\in \mathfrak{I}_{p,\sqrt{2}\varepsilon}(\Xi)}. \tag{4.21}$$

Nach Konstruktion ist $\widetilde{n}$ eine Potenz von 2, daher können wir den Koeffizientenvektor analog zu (4.21) weiter teilen, bis jeder Teil nur noch aus einem Summanden besteht. Dies liefert

$$f \in \mathfrak{I}_{p,\sqrt{2}^{\log_2(\widetilde{n})}\varepsilon}(\Xi) = \mathfrak{I}_{p,\sqrt{\widetilde{n}}\varepsilon}(\Xi).$$

mit $\widetilde{n} = 2^{\lceil \log_2(n) \rceil}$. □

Durch eine Erweiterung der Beweisidee zu Satz 4.27 von Sauer, vgl. [Sau07, Lemma 1], können wir auf die Auffüllung der Polynome $f_{n+1} = \cdots = f_{\widetilde{n}}$ verzichten und erhalten so eine schärfere Aussage.

Satz 4.30. *Seien $f_1, \dots, f_n \in \mathfrak{I}_{p,\varepsilon}(\Xi)$ gegeben mit $(f_i, f_j) = 0$ für $i \neq j$ und $f \in \operatorname{span}\{f_1, \dots, f_n\}$, so gilt $f \in \mathfrak{I}_{p,\sqrt{n}\varepsilon}(\Xi)$.*

Beweis. Für $f \in \operatorname{span}\{f_1, \dots, f_n\}$ gibt es eine Darstellung der Form

$$f = \sum_{j=1}^{n} \lambda_j f_j, \quad \lambda_j \in \mathbb{R}.$$

Unter Verwendung von Lemma 4.28 und durch Abschätzungen mit der Dreiecksungleichung und der Hölder'schen Ungleichung erhält man dann

$$\begin{aligned}\|f(\Xi)\|_p &= \left\| \left(\sum_{j=1}^{n} \lambda_j f_j \right)(\Xi) \right\|_p \leq \sum_{j=1}^{n} \|\lambda_j f_j(\Xi)\|_p \leq \varepsilon \sum_{j=1}^{n} \|\lambda_j f_j\|_2 \\ &\leq \varepsilon\sqrt{n} \left(\sum_{j=1}^{n} \|\lambda_j f_j\|_2^2 \right)^{1/2} = \varepsilon\sqrt{n} \left(\sum_{j=1}^{n} \|\lambda_j f_j\|_2^2 + \sum_{j=1}^{n} \sum_{k=j+1}^{n} \underbrace{(\lambda_j f_j, \lambda_k f_k)}_{=0} \right)^{1/2} \\ &= \varepsilon\sqrt{n} \left(\left\| \sum_{j=1}^{n} \lambda_j f_j \right\|_2^2 \right)^{1/2} = \varepsilon\sqrt{n} \left\| \sum_{j=1}^{n} \lambda_j f_j \right\|_2 = \varepsilon\sqrt{n} \|f\|_2,\end{aligned}$$

was genau der Bedingung für $f \in \mathfrak{I}_{p,\sqrt{n}\varepsilon}(\Xi)$ entspricht. □

4.2.2. Multiplikation mit Monomen

Im Folgenden wird für die Multiplikation eine ähnliche Aussage hergeleitet, wobei zunächst nur die Multiplikation eines Polynoms mit einem Monom betrachtet werden soll. Dazu bemerken wir zwei Eigenschaften der beteiligten Normen. Für die *Koeffizientennorm* gilt folgender Zusammenhang:

Lemma 4.31. *Seien $f, g \in \Pi_d$, dann ist $\|f \cdot g\|_2 \leq \|f\|_2 \cdot \|g\|_1$. Falls g ein Monom ist, gilt Gleichheit.*

Beweis. Zunächst bemerken wir, dass die Multiplikation eines Polynoms mit einem Term die Koeffizienten des Polynoms nur verschiebt – vgl. dazu die Konstruktion in Abschnitt 2.4.2 bzw. Algorithmus 2.30. Dabei bleibt insbesondere die euklidische Norm des Koeffizientenvektors erhalten, d. h. es gilt $\|x^\alpha \cdot f\|_2 = \|f\|_2$ für alle $\alpha \in \mathbb{N}_0^d$.

Mit diesem Zusammenhang und der Dreiecksungleichung erhalten wir

$$\begin{aligned}\|f \cdot g\|_2 &= \left\| \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| \leq \deg(g)}} g_\alpha x^\alpha \cdot f \right\|_2 \leq \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| \leq \deg(g)}} |g_{\alpha \in \mathbb{N}_0^d}| \cdot \|x^\alpha \cdot f\|_2 = \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| \leq \deg(g)}} |g_{\alpha \in \mathbb{N}_0^d}| \cdot \|f\|_2 \\ &= \|f\|_2 \cdot \|g\|_1.\end{aligned}$$

Ist g ein Monom, so hat der Koeffizientenvektor nur ein von Null verschiedenes Element. Die Summe besteht damit nur aus einem Summanden und die Dreiecksungleichung entfällt. Somit gilt in diesem Fall Gleichheit. □

Für die *Auswertungsnorm* benötigen wir folgende Hilfsaussage, die eine multiplikative Dreiecksungleichung im Sinne der komponentenweisen Multiplikation zweier Vektoren bereitstellt:

Lemma 4.32. *Seien $x, y \in \mathbb{R}^n$ und $1 \leq p \leq \infty$. Dann gilt*

$$\left\| \begin{pmatrix} x_1 \cdot y_1 \\ x_2 \cdot y_2 \\ \vdots \\ x_n \cdot y_n \end{pmatrix} \right\|_p \leq \left\| \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right\|_p \cdot \left\| \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \right\|_p . \tag{4.22}$$

Beweis. Sei zunächst $1 \leq p < \infty$. Mit der Cauchy-Schwarz'schen Ungleichung und der Monotonie der p-Normen erhält man:

$$\begin{aligned} \left\| \begin{pmatrix} x_1 \cdot y_1 \\ x_2 \cdot y_2 \\ \vdots \\ x_n \cdot y_n \end{pmatrix} \right\|_p &= \left(\sum_{j=1}^{n} |x_j|^p |y_j|^p \right)^{1/p} = \left((|x_1|^p, |x_2|^p, \ldots, |x_n|^p) \cdot \begin{pmatrix} |y_1|^p \\ |y_2|^p \\ \vdots \\ |y_n|^p \end{pmatrix} \right)^{1/p} \\ &\leq \left(\left\| \begin{pmatrix} |x_1|^p \\ |x_2|^p \\ \vdots \\ |x_n|^p \end{pmatrix} \right\|_2 \cdot \left\| \begin{pmatrix} |y_1|^p \\ |y_2|^p \\ \vdots \\ |y_n|^p \end{pmatrix} \right\|_2 \right)^{1/p} = \left(\sum_{j=1}^{n} |x_j|^{2p} \right)^{1/(2p)} \left(\sum_{j=1}^{n} |y_j|^{2p} \right)^{1/(2p)} \\ &= \left\| \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right\|_{2p} \cdot \left\| \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \right\|_{2p} \leq \left\| \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right\|_p \cdot \left\| \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \right\|_p . \end{aligned}$$

Für $p = \infty$ gilt

$$\max_{j=1,\ldots,n} |x_j \cdot y_j| = \max_{j=1,\ldots,n} |x_j| \cdot |y_j| \leq \max_{j=1,\ldots,n} |x_j| \cdot \max_{j=1,\ldots,n} |y_j|$$

und damit die Behauptung. □

Interpretiert man die Vektoren $x, y \in \mathbb{R}^n$ als $n \times 1$-Matrizen, so entspricht die komponentenweise Multiplikation auf der linken Seite von (4.22) dem *Hadamard-Produkt* $x \circ y$ (siehe [HJ91]) und die Abschätzung kann als *Submultiplikativität* aufgefasst werden. Für quadratische, positiv-definite Matrizen gilt bzgl. der Spektralnorm ein entsprechendes Resultat, das auf Schur zurückgeht, vgl. [HJ91]. Weitere Details zur Submultiplikativität von p-Normen bzgl. des Hadamard-Produktes werden in [JN00] beschrieben.

Da die Polynomauswertung ein Ringhomomorphismus ist, also $(f \cdot g)(\xi) = f(\xi) \cdot g(\xi)$ gilt, können wir mit Hilfe von Lemma 4.32 eine Abschätzung für die Multiplikation zweier Polynome in der Auswertungsnorm angeben.

Lemma 4.33. *Seien $f, g \in \Pi_d$, $\Xi = \{\xi_1, \ldots, \xi_n\} \subset \mathbb{R}^d$, dann ist $\|(f \cdot g)(\Xi)\|_p \leq \|f(\Xi)\|_p \cdot \|g(\Xi)\|_p$.*

Kombinieren wir nun die Resultate von Lemma 4.31 und Lemma 4.33, so erhalten wir eine Aussage über die Auswirkung der Multiplikation eines Polynoms mit einem Monom auf die Toleranz ε.

Satz 4.34. *Sei $\Xi \subset \mathbb{R}^d$, $f \in \mathfrak{I}_{p,\varepsilon}(\Xi)$, $1 \leq p \leq \infty$, und $g(x) = g_\alpha x^\alpha$ ein Monom, dann ist $(f \cdot g) \in \mathfrak{I}_{p,\varepsilon\|x^\alpha(\Xi)\|_p}(\Xi)$.*

Beweis.

$$\frac{\|(f \cdot g)(\Xi)\|_p}{\|f \cdot g\|_2} \leq \frac{\|f(\Xi)\|_p \cdot \|g(\Xi)\|_p}{\|g\|_1 \cdot \|f\|_2} = \frac{\|f(\Xi)\|_p \cdot |g_\alpha| \cdot \|x^\alpha(\Xi)\|_p}{|g_\alpha| \|f\|_2} \leq \varepsilon \cdot \|x^\alpha(\Xi)\|_p.$$

□

Die Veränderung der Toleranz hängt also von der Punktmenge $\Xi \subset \mathbb{R}^d$ und dem Exponenten des beteiligten Monoms ab. Dabei ist natürlich $\|x^\alpha(\Xi)\|_p \leq 1$ wünschenswert, da in diesem Fall das approximative Ideal nicht verlassen wird. Diese Voraussetzung können wir für die Maximumsnorm $p = \infty$ garantieren, indem wir $\Xi \subset [-1, 1]^d =: \mathbb{B}_\infty^d$ fordern. Dabei beschreibt $\mathbb{B}_\infty^d$ die d-dimensionale Einheitskugel

bzgl. der Maximumsnorm. Korollar 4.35 zeigt, dass die Abschätzung aus Satz 4.34 unter dieser Voraussetzung sogar unabhängig von dem Monom $g(x)$ wird.

Korollar 4.35. *Sei $\Xi \subset [-1,1]^d$, $f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ und $g(x) = g_\alpha x^\alpha$, dann ist $(f \cdot g) \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$.*

Eine solche Aussage lässt sich leider nicht für alle p-Normen formulieren. Das folgende Beispiel zeigt dies anhand der Betragssummennorm $p = 1$:

Beispiel 4.36. *Sei $\mathbb{B}_1^2 := \{\xi \in \mathbb{R}^2 : \|\xi\|_1 \leq 1\}$ die 2-dimensionale Einheitskugel bzgl. der Betragssummennorm, dann gilt*

$$\Xi := \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 \\ 0.9 \end{pmatrix} \right\} \subset \mathbb{B}_1^2.$$

Für $\alpha = (1,0) \in \mathbb{N}_0^2$ erhält man jedoch $\|x^\alpha(\Xi)\|_1 = \left\|(1,0.1)^T\right\|_1 = 1.1 > 1$.

In [HKPP09] wurde experimentell gezeigt, dass eine Einschränkung bzw. Umskalierung der Punkte auf den d-dimensionalen Einheitswürfel auch für die euklidische Norm bessere Ergebnisse liefert. Wir werden daher in diesem Kapitel stets $\Xi \subset [-1,1]^d$ annehmen.

4.2.3. Multiplikation mit Polynomen

Da sich die Multiplikation zweier Polynome auf die gewichtete Summe von Multiplikationen eines Polynoms mit Termen zurückführen lässt, kann man auf diesem Weg auch das Verhalten der Multiplikation eines Polynoms aus einem approximativen Ideal mit einem beliebigen Polynom untersuchen. In exakter Rechnung entspricht dies der Idealeigenschaft $\mathfrak{I} \cdot \Pi_d \subseteq \mathfrak{I}$. Zunächst benötigen wir noch einen Begriff, der den relevanten Anteil eines Polynoms im Sinne der von Null verschiedenen Koeffizienten beschreibt.

Definition 4.37. *Zu einem Polynom $f \in \Pi_d$ definieren wir den* Träger *als*

$$\operatorname{supp}(f) := \{\alpha \in \mathbb{N}_0^d : f_\alpha \neq 0\} \subset \mathbb{N}_0^d.$$

Die Anzahl der von Null verschiedenen Koeffizienten von f bezeichnen wir als 0-Norm*, in Zeichen:* $\|f\|_0 = \#\operatorname{supp}(f)$.

Die Definition der 0-Norm findet man insbesondere im Kontext des *Compressed Sensing*, vgl. [EK12]. Wir werden später noch detailliert auf diesen Zusammenhang eingehen. Auch wenn die Bezeichnung darauf hindeutet, handelt es sich bei der 0-Norm *nicht* um eine Norm. In Abschnitt 4.4 präzisieren wir diesen Punkt und leiten weitere wichtige Eigenschaften der 0-Norm her. Für die Betrachtungen an dieser Stelle genügt jedoch zunächst die Definition. Durch diese können wir Korollar 4.35 nun auf Polynome $g \in \Pi_d$ erweitern, die einer Forderung an $\operatorname{supp}(g)$ genügen.

Satz 4.38. *Sei* $\Xi \subset [-1,1]^d$*,* $f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$*,* $g(x) = \sum_{|\alpha|\leq n} g_\alpha x^\alpha$*. Ist nun*

$$\bigcap_{\alpha\in\operatorname{supp}(g)} \{\alpha + \alpha' : \alpha' \in \operatorname{supp}(f)\} = \emptyset, \tag{4.23}$$

so folgt $(f \cdot g) \in \mathfrak{I}_{\infty,\varepsilon\sqrt{\|g\|_0}}(\Xi)$.

Beweis. Für die Multiplikation gilt

$$(f \cdot g)(x) = \sum_{|\alpha|\leq n} g_\alpha x^\alpha f(x) = \sum_{\alpha\in\operatorname{supp}(g)} \underbrace{g_\alpha x^\alpha f(x)}_{\in\mathfrak{I}_{\infty,\varepsilon}(\Xi)}.$$

Wegen (4.23) gilt für die Summanden paarweise die Orthogonalitätsbeziehung

$$\Big(g_\alpha x^\alpha f(x), g_{\alpha'} x^{\alpha'} f(x)\Big) = 0, \quad \alpha \neq \alpha' \in \operatorname{supp}(g).$$

Mit $(f \cdot g) \in \operatorname{span}\{x^\alpha f(x) : \alpha \in \operatorname{supp}(g)\}$ sind alle Voraussetzungen von Satz 4.30 erfüllt und dieser liefert $(f \cdot g) \in \mathfrak{I}_{\infty,\varepsilon\sqrt{\|g\|_0}}(\Xi)$. □

Die Forderung (4.23) muss gestellt werden, um die Orthogonalität der Summanden zu garantieren, da ansonsten Satz 4.30 nicht anwendbar ist. Dies ist natürlich eine sehr starke Einschränkung an g und somit verliert die Aussage an praktischer Relevanz. Um eine universell gültige Abschätzung für die Veränderung von ε unter der

Polynommultiplikation zu entwickeln, betrachten wir den Ausdruck

$$\frac{\|(f \cdot g)(\Xi)\|_\infty}{\|f \cdot g\|_2}, \quad f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi),\ g \in \Pi_d. \tag{4.24}$$

Eine Abschätzung für den Zähler in (4.24) können wir aus Lemma 4.33 übernehmen. Gehen wir aufgrund der Bemerkungen im letzten Abschnitt zusätzlich von der Skalierung $\Xi \subset [-1,1]^d$ aus, so lässt sich die dort gezeigte Abschätzung zu folgender Aussage erweitern.

Korollar 4.39. *Seien $\Xi \subset [-1,1]^d$, $f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ mit $\|f\|_2 = 1$ und $g \in \Pi_d$ gegeben, dann ist $\|(f \cdot g)(\Xi)\|_\infty \leq \varepsilon \cdot \|g\|_1$.*

Beweis. Mit Hilfe der Submultiplikativität der Auswertungsnorm aus Lemma 4.33 und der Dreiecksungleichung gilt

$$\begin{aligned}\|(f \cdot g)(\Xi)\|_\infty \leq \|f(\Xi)\|_\infty \cdot \|g(\Xi)\|_\infty \leq \varepsilon \cdot \|g(\Xi)\|_\infty = \varepsilon \cdot \left\| \sum_{\alpha \in \mathbb{N}_0^d} g_\alpha x^\alpha(\Xi) \right\|_\infty \\ \leq \varepsilon \cdot \sum_{\alpha \in \mathbb{N}_0^d} |g_\alpha| \underbrace{\max_{\xi \in \Xi} \{|x^\alpha(\xi)|\}}_{\leq 1} \leq \varepsilon \cdot \sum_{\alpha \in \mathbb{N}_0^d} |g_\alpha| = \varepsilon \cdot \|g\|_1.\end{aligned}$$

Die Abschätzung $|x^\alpha(\xi)| \leq 1$, $\xi \in \Xi$, folgt dabei aus der Einschränkung der Punktmenge auf den Hyperwürfel $[-1,1]^d$. □

Um den Nenner von (4.24) abzuschätzen, benötigen wir eine weitere Darstellung der Polynommultiplikation, die in Abschnitt 2.4.3 vorgestellt wurde: die Multiplikation des Koeffizientenvektors mit einer *Faltungsmatrix*. Der folgende Satz liefert eine Voraussetzung, unter der sich die Koeffizientennorm in (4.24) durch die Koeffizientennorm von g beschränken lässt:

Satz 4.40. *Seien $0 \neq f, g \in \Pi_d$ und $k = \deg(f) + \deg(g)$. Sind alle Singulärwerte der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ größer oder gleich 1, dann gilt $\|f \cdot g\|_2 \geq \|g\|_2$.*

Beweis. Die Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ hat nach Konstruktion vollen Rang. Damit ist die Matrix $\boldsymbol{C}_{k,d}(f)^T \boldsymbol{C}_{k,d}(f)$ positiv definit. Zudem sind nach Voraussetzung alle

Eigenwerte von $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)$ größer oder gleich 1 und dementsprechend alle Eigenwerte der Matrix $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f) - I$ größer oder gleich 0, was bedeutet, dass diese Matrix positiv semi-definit ist. Es gilt also

$$\begin{aligned}\|f \cdot g\|_2^2 &= \left\|\left(\boldsymbol{C}_{k,d}(f) \cdot g^T\right)^T\right\|_2^2 = g\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)g^T \\ &= \|g\|_2^2 + \underbrace{g\underbrace{\left(\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f) - I\right)}_{\text{pos. semi-def.}}g^T}_{\geq 0} \geq \|g\|_2^2\end{aligned}$$

und damit die Behauptung. □

Die Forderung an die Singulärwerte der Faltungsmatrix in Satz 4.40 stellt zunächst wieder eine starke Einschränkung dar. Dies lässt sich aber leicht beheben, da man die Voraussetzung stets durch eine geeignete Skalierung herstellen kann. Die folgende Aussage kombiniert die Abschätzungen von Korollar 4.39 und Satz 4.40 mit der entsprechenden Skalierung:

Satz 4.41. *Seien $\Xi \subset [-1,1]^d$, $0 \neq f, g \in \Pi_d$, $f \in \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ mit $\|f\|_2 = 1$ und $k = \deg(f)+\deg(g)$. Ist $\sigma > 0$ der kleinste Singulärwert der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$, so gilt*

$$(f \cdot g) \in \mathfrak{I}_{\infty,\tilde{\varepsilon}}(\Xi), \quad \tilde{\varepsilon} := \varepsilon\frac{\sqrt{\|g\|_0}}{\sigma},$$

Beweis. Es gilt analog zum Beweis von Satz 4.40

$$\begin{aligned}\|f \cdot g\|_2^2 &= \left\|\left(\boldsymbol{C}_{k,d}(f) \cdot g^T\right)^T\right\|_2^2 \\ &= (\sigma g)(\sigma^{-1}\boldsymbol{C}_{k,d}(f)^T)(\sigma^{-1}\boldsymbol{C}_{k,d}(f))(\sigma g)^T \\ &= \|\sigma g\|_2^2 + \underbrace{(\sigma g)\underbrace{\left(\sigma^{-1}\boldsymbol{C}_{k,d}(f)^T\sigma^{-1}\boldsymbol{C}_{k,d}(f) - I\right)}_{\text{pos. semidef.}}(\sigma g)^T}_{\geq 0} \\ &\geq \|\sigma g\|_2^2 = \sigma^2\|g\|_2^2.\end{aligned}$$

Zusammen mit Korollar 4.39 und der Äquivalenz der p-Normen (siehe beispielsweise

[SK11, Kapitel 2.2.1]) folgt nun

$$\frac{\|(f \cdot g)(\Xi)\|_\infty}{\|f \cdot g\|_2} \leq \frac{\|g\|_1 \cdot \varepsilon}{\|f \cdot g\|_2} \leq \frac{\|g\|_1 \cdot \varepsilon}{\|g\|_2 \cdot \sigma} \leq \varepsilon \frac{\sqrt{\|g\|_0}}{\sigma}$$

was mit $\tilde{\varepsilon} := \varepsilon\sqrt{\|g\|_0}/\sigma$ zur Behauptung $(f \cdot g) \in \mathfrak{I}_{\infty,\tilde{\varepsilon}}(\Xi)$ führt. □

Die allgemeineren Voraussetzungen von Satz 4.41 schlagen sich also in dem Faktor $1/\sigma$ nieder, der für $\sigma < 1$ die Abschätzung verschlechtert. In diesen Fällen ist jedoch Satz 4.40 gar nicht anwendbar. Umgekehrt erhalten wir für $\sigma > 1$ eine Verschärfung der Aussage, auch im Vergleich zur Abschätzung in Satz 4.38. Das folgende Beispiel zeigt diesen Vergleich und eine Anwendung beider Resultate:

Beispiel 4.42. *Seien die Polynome $f, g \in \Pi_2$ aus Beispiel 2.32 gegeben durch*

$$f(x_1, x_2) = \frac{4}{5}x_1 + \frac{3}{5}x_2, \quad g(x_1, x_2) = 3x_1 + 2x_2 + 1.$$

Für die Koeffizientennorm von f und g gilt dann $\|f\|_2 = 1$ bzw. $\|g\|_2 = \sqrt{14}$. Es ist

$$\begin{aligned}
\|f \cdot g\|_2^2 &= g\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)g^T \\
&= \frac{1}{25}\begin{bmatrix}1 & 2 & 3\end{bmatrix}\begin{bmatrix}0 & 3 & 4 & 0 & 0 & 0\\ 0 & 0 & 0 & 3 & 4 & 0\\ 0 & 0 & 0 & 0 & 3 & 4\end{bmatrix} \cdot \begin{bmatrix}0 & 0 & 0\\ 3 & 0 & 0\\ 4 & 0 & 0\\ 0 & 3 & 0\\ 0 & 4 & 3\\ 0 & 0 & 4\end{bmatrix}\begin{bmatrix}1\\2\\3\end{bmatrix} \\
&= \frac{1}{25}\begin{bmatrix}1 & 2 & 3\end{bmatrix}\begin{bmatrix}25 & 0 & 0\\ 0 & 25 & 12\\ 0 & 12 & 25\end{bmatrix}\begin{bmatrix}1\\2\\3\end{bmatrix} = \frac{494}{25} > \frac{350}{25} = \|g\|_2^2.
\end{aligned}$$

Die Singulärwerte von $\boldsymbol{C}_{2,2}(f)$ sind jedoch $\sigma_1 = \sqrt{\frac{37}{25}}$, $\sigma_2 = 1$, $\sigma_3 = \sqrt{\frac{13}{25}}$, also ist Satz 4.40 nicht anwendbar, obwohl $\|f \cdot g\|_2 \geq \|g\|_2$ erfüllt ist. Stattdessen müssen wir Satz 4.41 verwenden, was zu einer schlechteren Abschätzung führt:

$$\|f \cdot g\|_2^2 = \frac{494}{25} > \frac{182}{25} = \sigma_3^2\|g\|_2^2.$$

Wählen wir anstatt g jedoch das Polynom $\tilde{g}(x_1, x_2) = -3x_1 + 2x_2 + 1$, $\|\tilde{g}\|_2 = \sqrt{14}$, so wird die Aussage von Satz 4.40 tatsächlich falsch. Es ist

$$\|f \cdot \tilde{g}\|_2^2 = \frac{1}{25}\begin{bmatrix}1 & 2 & -3\end{bmatrix}\begin{bmatrix}25 & 0 & 0\\ 0 & 25 & 12\\ 0 & 12 & 25\end{bmatrix}\begin{bmatrix}1\\ 2\\ -3\end{bmatrix} = \frac{206}{25} < \frac{350}{25} = \|g\|_2^2.$$

Die Abschätzung aus dem Beweis von Satz 4.41 gilt jedoch nach wie vor:

$$\|f \cdot \tilde{g}\|_2^2 = \frac{206}{25} > \frac{182}{25} = \sigma_3^2\|g\|_2^2.$$

Abschließend wird nun eine untere Schranke für den kleinsten Singulärwert einer Faltungsmatrix hergeleitet. Ein intuitiver Ansatz dazu besteht darin, die Eigenwerte der symmetrischen Matrizen $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)$, $k \geq \deg(f)$, durch die *Gerschgorin-Kreise* abzuschätzen, vgl. [Ger31]. Dazu bemerken wir zunächst die folgende Eigenschaft der Diagonalelemente dieser Matrix:

Lemma 4.43. *Sei $f \in \Pi_d$ und $k \geq \deg(f)$, dann gilt $(\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f))_{j,j} = \|f\|_2^2$.*

Beweis. Nach Definition 2.33 entsprechen die Spalten der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ den Koeffizientenvektoren von $x^\alpha f$, $\alpha \in \mathbb{N}_0$, $|\alpha| \leq k - \deg(f)$. Da die Multiplikation mit einem Term x^α nur die Position, aber nicht den Wert der Koeffizienten verändert, gilt $(x^\alpha f, x^\alpha f) = (f, f) = \|f\|_2^2$ für alle $\alpha \in \mathbb{N}_0^d$. □

Nun besagt die Abschätzung von Gerschgorin (siehe [Ger31]), dass alle Gerschgorin-Kreise der Matrizen $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)$, $k \geq \deg(f)$, ihr Zentrum im Punkt $\|f\|_2^2$ haben. Weiterhin sind die Matrizen symmetrisch und positiv definit, sodass die Gerschgorin-Kreise zu Intervallen auf der reellen Achse degenerieren. Sei $\rho \in \mathbb{R}$ der Radius des größten Gerschgorin-Kreises, dann liegen alle Eigenwerte im Intervall $[\|f\|_2^2 - \rho, \|f\|_2^2 + \rho]$. Wir erhalten damit folgende Abschätzungen:

$$\lambda_{\max} \leq \|f\|_2^2 + \rho \leq \|f\|_2^2 + \sum_{i \neq j}|f_i \cdot f_j| = \sum_i\sum_j |f_i||f_j| = \sum_i |f_i|\|f\|_1 = \|f\|_1^2, \tag{4.25}$$

$$\lambda_{\min} \geq \|f\|_2^2 - \rho \geq \|f\|_2^2 - \sum_{i \neq j} |f_i \cdot f_j| = 2\|f\|_2^2 - \sum_i |f_i| \|f\|_1 = 2\|f\|_2^2 - \|f\|_1^2. \tag{4.26}$$

Zunächst bemerken wir, dass beide Abschätzungen unabhängig von k sind, d. h. wir erhalten Schranken für *alle* Faltungsmatrizen $\boldsymbol{C}_{k,d}(f)$ zu einem Polynom $f \in \Pi_d$ und $k \geq \deg(f)$. Die Abschätzung des größten Eigenwertes in (4.25) bestätigt ein bekanntes Resultat von Batselier aus [Bat13], das zur Untersuchung der Konditionszahl von Faltungsmatrizen verwendet wurde. Die Abschätzung des kleinsten Eigenwerts durch (4.26) liefert weniger Erkenntnis, da sie sehr häufig trivial ist.

Beispiel 4.44. *Sei $f(x_1, x_2) = 4x_1 + 2x_2 + 1$, dann liefert (4.26) die Abschätzung $\lambda_{\min} \geq 2 \cdot 21 - 49 = -7$. Dies ist jedoch trivial, da die Matrix $\boldsymbol{C}_{k,2}(f)^T \boldsymbol{C}_{k,2}(f)$ für alle $k \geq 1$ positiv definit ist und somit per definitionem $\lambda_{\min} \geq 0$ gilt.*

Basierend auf Ergebnissen aus [Joh89] wurde in [Bat13] folgende Abschätzung des kleinsten Singulärwerts einer Faltungsmatrix gezeigt, die ebenfalls unabhängig von k für alle Faltungsmatrizen eines Polynoms gilt:

Satz 4.45. *Sei $f \in \Pi_d$ und $k \geq \deg(f)$, dann gilt für den kleinsten Singulärwert σ der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ die Abschätzung $\sigma \geq 2 \cdot |f_0| - \|f\|_1$, wobei f_0 den Koeffizienten des konstanten Terms von f beschreibt.*

Die Problematik der trivialen Fälle bleibt jedoch auch für Satz 4.45 bestehen: Wenn $|f_0| < \frac{\|f\|_1}{2}$ ist, erhält man eine schlechtere Abschätzung als die per definitionem geforderte Eigenschaft $\sigma \geq 0$. Damit ist das Resultat für Polynome ohne konstanten Anteil unbrauchbar. Wir können sogar ein notwendiges Kriterium für die Nichttrivialität von Satz 4.45 angeben:

Lemma 4.46. *Sei $f \in \Pi_d$ mit $2|f_0| - \|f\|_1 \geq 0$, dann ist $|f_0| = \|f\|_\infty$.*

Beweis. Ist $|f_0| < \|f\|_\infty = \max\{f_j\}$, dann gibt es ein k mit $|f_k| > |f_0|$ und es gilt $\|f\|_1 = \sum_j |f_j| \geq |f_0| + |f_k| > 2|f_0|$. □

An dieser Stelle ergibt sich die Frage, ob man auch für Polynome mit $|f_0| \neq \|f\|_\infty$ eine geeignete Abschätzung finden kann. Dazu bemerken wir zunächst, dass sich

$$\begin{array}{l} [1,2,3,4,5,6] \\ [1,3,2,6,5,4] \\ [4,2,5,1,3,6] \\ [4,5,2,6,3,1] \\ [6,3,5,1,2,4] \\ [6,5,3,4,2,1] \end{array} \quad \Longrightarrow \quad \left[\begin{array}{c|cc|ccc} 1 & x_2 & x_1 & x_2^2 & x_1x_2 & x_1^2 \\ 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ x_2^2 & x_2 & x_1x_2 & 1 & x_1 & x_1^2 \\ x_2^2 & x_1x_2 & x_2 & x_1^2 & x_1 & 1 \\ x_1^2 & x_1 & x_1x_2 & 1 & x_2 & x_2^2 \\ x_1^2 & x_1x_2 & x_1 & x_2^2 & x_2 & 1 \end{array}\right]$$

Tabelle 4.2.: Permutationen eines Koeffizientenvektors mit $d = 2$, $k = 2$, die den kleinsten Singulärwert der Faltungsmatrix nicht verändern.

zu jedem Polynom $f \in \Pi_{k,d}$ durch Permutation der Koeffizienten andere Polynome konstruieren lassen, sodass die Singulärwerte der zugehörigen Faltungsmatrizen übereinstimmen. Einige Beispiele solcher Permutationen sind in den Tabellen 4.2 - 4.4 dargestellt. Da alle diese Permutationen auch als Änderung der Termordnung interpretiert werden können, definieren wir im Folgenden Permutationsmatrizen die eben diese Änderung beschreiben:

Definition 4.47. *Sei $\prec$ eine beliebige Termordnung auf $\mathbb{N}_0^d$ im Sinne von Definition 2.13, dann bezeichnet $\boldsymbol{P}_{k,d}^{\prec}$ die Permutationsmatrix, die die Umordnung der Multiindizes $\alpha \in \mathbb{N}_0^d$, $|\alpha| \leq k$ von $\prec_{glex}$ nach $\prec$ beschreibt.*

Beispiel 4.48. *Für die umgekehrt-lexikographische Termordnung (vgl. Abschnitt 2.2) erhalten wir*

$$\boldsymbol{P}_{1,2}^{\prec_{rlex}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{2,2}^{\prec_{rlex}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Mit Hilfe dieser Permutationsmatrizen können Faltungsmatrizen $\boldsymbol{C}_{k,d}(f)$, $f \in \Pi_d$, auch bzgl. anderer Termordnungen dargestellt werden. Dabei ändert sich einerseits die Anordnung der Zeilen, da die Koeffizienten in f umgeordnet werden. Andererseits sind per definitionem auch die Spalten der Faltungsmatrix in graduiert-

$$\left[\begin{array}{c|cc|ccc|ccc|c}
1 & x_2 & x_1 & x_2^2 & x_1x_2 & x_1^2 & x_2^3 & x_1x_2^2 & x_1^2x_2 & x_1^3 \\
1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\
x_2^3 & x_2^2 & x_1x_2^2 & x_2 & x_1x_2 & x_1^2x_2 & 1 & x_1 & x_1^2 & x_1^3 \\
x_2^3 & x_1x_2^2 & x_2^2 & x_1^2x_2 & x_1x_2 & x_2 & x_1^3 & x_1^2 & x_1 & 1 \\
x_1^3 & x_1^2 & x_1^2x_2 & x_1 & x_1x_2 & x_1x_2^2 & 1 & x_2 & x_2^2 & x_2^3 \\
x_1^3 & x_1^2x_2 & x_1^2 & x_1x_2^2 & x_1x_2 & x_1 & x_2^3 & x_2^2 & x_2 & 1
\end{array}\right]$$

Tabelle 4.3.: Permutationen eines Koeffizientenvektors mit $d = 2$, $k = 3$, die den kleinsten Singulärwert der Faltungsmatrix nicht verändern.

$$\left[\begin{array}{c|ccc|ccccc|c}
1 & x_3 & x_2 & x_1 & x_3^2 & x_2x_3 & x_2^2 & x_1x_3 & x_1x_2 & x_1^2 \\
1 & x_3 & x_1 & x_2 & x_3^2 & x_1x_2 & x_1^2 & x_2x_3 & x_1x_2 & x_2^2 \\
1 & x_2 & x_3 & x_1 & x_2^2 & x_2x_3 & x_3^2 & x_1x_2 & x_1x_3 & x_1^2 \\
1 & x_2 & x_1 & x_3 & x_2^2 & x_1x_2 & x_1^2 & x_2x_3 & x_1x_3 & x_3^2 \\
1 & x_1 & x_3 & x_2 & x_1^2 & x_1x_3 & x_3^2 & x_1x_2 & x_2x_3 & x_2^2 \\
1 & x_1 & x_2 & x_3 & x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 \\
x_3^2 & x_3 & x_2x_3 & x_1x_3 & 1 & x_2 & x_2^2 & x_1 & x_1x_2 & x_1^2 \\
x_3^2 & x_3 & x_1x_3 & x_2x_3 & 1 & x_1 & x_1^2 & x_2 & x_1x_2 & x_2^2 \\
x_3^2 & x_2x_3 & x_3 & x_1x_3 & x_2^2 & x_2 & 1 & x_1x_2 & x_1 & x_1^2 \\
x_3^2 & x_2x_3 & x_1x_3 & x_3 & x_2^2 & x_1x_2 & x_1^2 & x_2 & x_1 & 1 \\
x_3^2 & x_1x_3 & x_3 & x_2x_3 & x_1^2 & x_1 & 1 & x_1x_2 & x_2 & x_2^2 \\
x_3^2 & x_1x_3 & x_2x_3 & x_3 & x_1^2 & x_1x_2 & x_2^2 & x_1 & x_2 & 1 \\
x_2^2 & x_2 & x_2x_3 & x_1x_2 & 1 & x_3 & x_3^2 & x_1 & x_1x_3 & x_1^2 \\
x_2^2 & x_2 & x_1x_2 & x_2x_3 & 1 & x_1 & x_1^2 & x_3 & x_1x_3 & x_3^2 \\
x_2^2 & x_2x_3 & x_2 & x_1x_2 & x_3^2 & x_3 & 1 & x_1x_3 & x_1 & x_1^2 \\
x_2^2 & x_2x_3 & x_1x_2 & x_2 & x_3^2 & x_1x_3 & x_1^2 & x_3 & x_1 & 1 \\
x_2^2 & x_1x_2 & x_2 & x_2x_3 & x_1^2 & x_1 & 1 & x_1x_2 & x_3 & x_3^2 \\
x_2^2 & x_1x_2 & x_2x_3 & x_2 & x_1^2 & x_1x_3 & x_3^2 & x_1 & x_3 & 1 \\
x_1^2 & x_1 & x_1x_3 & x_1x_2 & 1 & x_3 & x_3^2 & x_2 & x_2x_3 & x_2^2 \\
x_1^2 & x_1 & x_1x_2 & x_1x_3 & 1 & x_2 & x_2^2 & x_3 & x_2x_3 & x_3^2 \\
x_1^2 & x_1x_3 & x_1 & x_1x_2 & x_3^2 & x_3 & 1 & x_2x_3 & x_2 & x_2^2 \\
x_1^2 & x_1x_3 & x_1x_2 & x_1 & x_3^2 & x_2x_3 & x_2^2 & x_3 & x_2 & 1 \\
x_1^2 & x_1x_2 & x_1 & x_1x_3 & x_2^2 & x_2 & 1 & x_2x_3 & x_3 & x_3^2 \\
x_1^2 & x_1x_2 & x_1x_3 & x_2 & x_2^2 & x_2x_3 & x_3^2 & x_2 & x_3 & 1
\end{array}\right]$$

Tabelle 4.4.: Permutationen eines Koeffizientenvektors mit $d = 3$, $k = 2$, die den kleinsten Singulärwert der Faltungsmatrix nicht verändern.

lexikographischer Ordnung, sodass wir auch diese neu ordnen müssen. Wir erhalten

$$\boldsymbol{C}_{k,d}^{\prec}(f) := (\boldsymbol{P}_{k,d}^{\prec})^T \boldsymbol{C}_{k,d}(f) \boldsymbol{P}_{k-\deg(f),d}^{\prec}.$$

Lemma 4.49. *Ist $\prec$ eine Termordnung mit der gleichen (Graduierungs-) Blockstruktur wie die graduiert-lexikographische Ordnung, vgl. (2.6), so gilt*

$$\boldsymbol{C}_{k,d}^{\prec}(f) = (\boldsymbol{P}_{k,d}^{\prec})^T \boldsymbol{C}_{k,d}(f) \boldsymbol{P}_{k-\deg(f),d}^{\prec} = \boldsymbol{C}_{k,d}(f \cdot \boldsymbol{P}_{\deg(f),d}^{\prec}). \tag{4.27}$$

Beweis. Dies ist eine direkte Konsequenz aus der 2. Eigenschaft von Definition 2.13 (Termordnung), die besagt, dass die Reihenfolge der Elemente eines Koeffizientenvektors auch in den Spalten einer Faltungsmatrix erhalten bleibt. □

Im Beweis des nächsten Satzes werden Termordnungen konstruiert, die die Voraussetzung von Lemma 4.49 erfüllen. Zunächst verdeutlicht jedoch das folgende Beispiel die Notwendigkeit der geforderten Blockstruktur:

Beispiel 4.50. *Sei $f(x_1, x_2) = 3x_1 + 2x_2 + 1$ und $\prec_{grlex}$ die graduiert-umgekehrt-lexikographische Ordnung, die durch*

$$\alpha \prec_{grlex} \beta \quad :\Longleftrightarrow \quad |\alpha| < |\beta| \quad \textit{oder} \quad |\alpha| = |\beta|,\ \alpha \prec_{rlex} \beta,$$

definiert ist, dann gilt

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{(\boldsymbol{P}_{2,2}^{\prec_{grlex}})^T} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 3 & 2 \\ 0 & 0 & 3 \end{bmatrix}}_{\boldsymbol{C}_{2,2}(f)} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\boldsymbol{P}_{1,2}^{\prec_{grlex}}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 3 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix}}_{\boldsymbol{C}_{2,2}(f \cdot \boldsymbol{P}_{1,2}^{\prec_{grlex}})} .$$

Die Beziehung (4.27) wird jedoch falsch, wenn wir eine Termordnung verwenden, die die Blockstruktur der graduiert-lexikographischen Ordnung nicht respektiert. Für

die lexikographische Ordnung erhalten wir

$$\underbrace{\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\\0&0&1&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}}_{(\boldsymbol{P}_{2,2}^{\prec_{lex}})^T}\cdot\underbrace{\begin{bmatrix}1&0&0\\2&1&0\\3&0&1\\0&2&0\\0&3&2\\0&0&3\end{bmatrix}}_{\boldsymbol{C}_{2,2}(f)}\cdot\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}}_{\boldsymbol{P}_{1,2}^{\prec_{lex}}}=\begin{bmatrix}1&0&0\\2&1&0\\0&2&0\\3&0&1\\0&3&2\\0&0&3\end{bmatrix}\neq\underbrace{\begin{bmatrix}1&0&0\\2&1&0\\3&0&1\\0&2&0\\0&3&2\\0&0&3\end{bmatrix}}_{\boldsymbol{C}_{2,2}(f\cdot\boldsymbol{P}_{1,2}^{\prec_{lex}})}.$$

Satz 4.51. *Sei $f \in \Pi_d$ ein Polynom, dann gibt es $(d+1)!$ Permutationen $\widetilde{f}$ des Koeffizientenvektors von f, sodass für alle $k \geq \deg(f)$ die Singulärwerte von $\boldsymbol{C}_{k,d}(f)$ mit den Singulärwerten von $\boldsymbol{C}_{k,d}(\widetilde{f})$ übereinstimmen.*

Beweis. Die Faltungsmatrizen $\boldsymbol{C}_{k,d}(f)$, $k \geq \deg(f)$, haben vollen Rang und ihre Singulärwerte entsprechen den positiven Wurzeln der Eigenwerte von $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)$. Für $k = \deg(f)$ hat diese symmetrische und positiv definite Matrix nach Lemma 4.43 nur den Eintrag $\|f\|_2^2$, der somit auch dem einzigen Eigenwert entspricht. Wegen $\|\widetilde{f}\|_2^2 = \|f\|_2$ folgt die Behauptung für jede Permutation $\widetilde{f}$ von f.

Sei nun $k > \deg(f)$, dann hat die Matrix

$$\begin{aligned}\left(\boldsymbol{C}_{k,d}^{\prec}(f)\right)^T\boldsymbol{C}_{k,d}^{\prec}(f) &= \left((\boldsymbol{P}_{k,d}^{\prec})^T\boldsymbol{C}_{k,d}(f)\boldsymbol{P}_{k-\deg(f),d}^{\prec}\right)^T\left((\boldsymbol{P}_{k,d}^{\prec})^T\boldsymbol{C}_{k,d}(f)\boldsymbol{P}_{k-\deg(f),d}^{\prec}\right)\\ &= (\boldsymbol{P}_{k-\deg(f),d}^{\prec})^T\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)\boldsymbol{P}_{k-\deg(f),d}^{\prec}\end{aligned}$$

dieselben Eigenwerte wie $\boldsymbol{C}_{k,d}(f)^T\boldsymbol{C}_{k,d}(f)$. Damit stimmen auch die Singulärwerte von $\boldsymbol{C}_{k,d}^{\prec}(f)$ und $\boldsymbol{C}_{k,d}(f)$ überein. Ist dabei $\prec$ eine Termordnung mit der Blockstruktur der graduiert-lexikographischen Ordnung, so gilt (4.27) und wir erhalten

$$\boldsymbol{C}_{k,d}^{\prec}(f) = \boldsymbol{C}_{k,d}(\underbrace{f\cdot\boldsymbol{P}_{\deg(f),d}^{\prec}}_{=:\widetilde{f}}).$$

Es gilt nun, Termordnungen mit der Blockstruktur der graduiert-lexikographischen Ordnung anzugeben. Dafür kommen zwei Typen von Umordnungen in Frage:

1. Änderung der Ordnung *innerhalb* der Graduierung, was einer Vertauschung der Variablen $x_1, \ldots, x_d$ entspricht.
2. Änderung der Graduierung: Entweder Graduierung nach dem Totalgrad (glex) oder Graduierung nach einer einzelnen Variable, die wir wie folgt definieren:

$$\alpha \prec_j \beta \quad \Leftrightarrow \quad \begin{cases} \alpha_j > \beta_j \\ \alpha_j = \beta_j,\ \alpha \prec_{lex} \beta \end{cases}, \quad j = 1, \ldots, d. \tag{4.28}$$

Da es für ein Polynom in d Variablen $d!$ Möglichkeiten zur Vertauschung von Variablen gibt, sind damit bereits $d!$ der gesuchten Permutationen gefunden. Durch Änderung der Termordnung zu $\prec_j$ kann jedes $x_j^{\deg(f)}$, $j = 1, \ldots, d$, an der Position von f_0 stehen. Zusammen mit f_0 sind das $(d+1)$ Permutationen. Insgesamt erhalten wir also $d! \cdot (d+1) = (d+1)!$ Permutationen von f mit den gleichen Singulärwerten der zugehörigen Faltungsmatrix. □

Die durch (4.28) definierten Termordnungen ergeben sich ebenfalls aus der umgekehrt-lexikographischen Termordnung mit entsprechender Vertauschung der Variablen – man vergleiche die Tabellen 4.2 - 4.4 oder Beispiel 4.48. Die Aussage von Satz 4.51 erlaubt es uns nun, den Wert f_0 in Satz 4.45 durch den ersten Koeffizienten einer passenden Permutation zu ersetzen. Damit können wir das Resultat von Batselier erweitern und eine allgemeinere Abschätzung formulieren.

Satz 4.52. *Sei $f \in \Pi_d$ und $k \geq \deg(f)$, dann gilt für den kleinsten Singulärwert σ der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$ die Abschätzung*

$$\sigma \geq 2 \max \left\{ \{|f_\alpha| : \alpha = \deg(f) \cdot \varepsilon_j, j = 1, \ldots, d\} \cup \{|f_0|\} \right\} - \|f\|_1,$$

für $\varepsilon_j = (\underbrace{0, \ldots, 0}_{j-1}, 1, 0, \ldots, 0) \in \mathbb{N}_0^d$.

Natürlich wird auch diese Abschätzung für

$$\max \left\{ \{|f_\alpha| : \alpha = \deg(f) \cdot \varepsilon_j, j = 1, \ldots, d\} \cup \{|f_0|\} \right\} \neq \|f\|_\infty$$

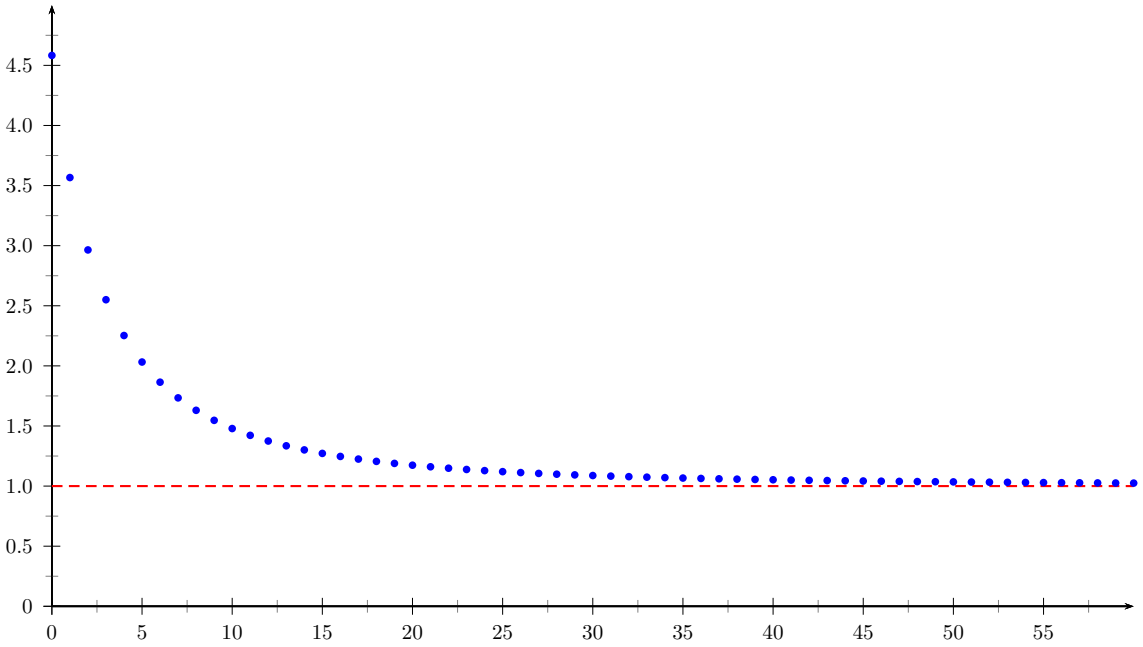


Abbildung 4.4.: Minimale Singulärwerte der Faltungsmatrix $\boldsymbol{C}_{1+k,2}(4x_1 + 2x_2 + 1)$ für $k = 0, \ldots, 60$ (in blau) und die nach Satz 4.52 bestimmte untere Schranke $2 \cdot 4 - 7 = 1$ (als rote Linie).

trivial, allerdings tritt dieser Fall seltener auf. Das folgende Beispiel zeigt die Verbesserung unserer Abschätzung im Vergleich zur Aussage von Batselier:

Beispiel 4.53. *Betrachten wir das Polynom $f(x_1, x_2) = 4x_1 + 2x_2 + 1$ aus Beispiel 4.44 erneut, so liefert Satz 4.45 die triviale Abschätzung $\sigma \geq -5$. Wenden wir hingegen Satz 4.52 an, so erhalten wir durch die Abschätzung $\sigma \geq 1$ ein deutlich besseres Ergebnis. Die exakten minimalen Singulärwerte der Faltungsmatrizen $\boldsymbol{C}_{1+k,2}(f)$ für $k = 0, \ldots, 60$ sind in Abbildung 4.4 dargestellt.*

Für lineare Polynome ist die Abschätzung aus Satz 4.52 besonders effektiv, da in diesem Fall alle Koeffizienten $f_\alpha \neq 0$ von der Form $\alpha = \deg(f) \cdot \varepsilon_j$ oder $\alpha = 0$ sind. Damit erhalten wir folgendes Resultat:

Korollar 4.54. *Sei $f \in \Pi_{1,d}$, $k \geq 1$ und σ der kleinste Singulärwert der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$, dann gilt $\sigma \geq 2\|f\|_\infty - \|f\|_1$.*

An dieser Stelle ist auch eine entsprechende Aussage für Polynome höheren Grades

wünschenswert. Da im Rahmen dieser Arbeit kein entsprechendes Gegenbeispiel gefunden werden konnte, formulieren wir die folgende Vermutung:

Vermutung 4.55. *Sei $f \in \Pi_d$, $k \geq \deg(f)$ und σ der kleinste Singulärwert der Faltungsmatrix $\boldsymbol{C}_{k,d}(f)$, dann gilt $\sigma \geq 2\|f\|_\infty - \|f\|_1$.*

4.3. Das approximative Ideal-Membership-Problem

Dieser Abschnitt untersucht das in Abschnitt 3.2 beschriebene *Ideal-Membership-Problem* nun im approximativen Kontext: Zu einem Ideal $\mathcal{F} = \langle f_1, \ldots, f_n \rangle \subset \Pi_d$ und einem Polynom $g \in \Pi_d$ soll entschieden werden, ob $g \in \mathcal{F}$ gilt. Ist die Varietät $\mathfrak{V}(\mathcal{F}) = \Xi$ des Ideals bekannt, so ist lediglich die Bedingung $g(\Xi) = 0$ bzw. im p-approximativen Fall $\|g(\Xi)\|_p \leq \varepsilon \|g\|_2$ zu überprüfen. Für nulldimensionale Ideale, also $\#\Xi < \infty$, lässt sich dies einfach nachrechnen.

In vielen praktischen Anwendungen interessiert man sich jedoch für die Lösung des *expliziten* Ideal-Membership-Problems, d. h. für eine konkrete Darstellung des Polynoms g der Form

$$\Pi_d \ni g = f_1 g_1 + \cdots + f_n g_n + \nu_{\mathcal{F}(g)}. \tag{4.29}$$

Das Standardverfahren zur Lösung dieses Problems besteht in der Berechnung einer Gröbnerbasis des Ideals und anschließender Anwendung des zugehörigen Divisionsalgorithmus. In näherungsweiser Rechnung ergeben sich dabei die folgenden in [HKPP09] genannten Komplikationen:

1. Das Polynom g kann *fast* im Ideal $\mathcal{F}$ liegen, sodass für die Normalform $\nu_{\mathcal{F}}(g) \approx 0$ gilt.
2. Die Berechnung der Gröbnerbasis ist numerisch nicht stabil.
3. Die Lösung des expliziten approximativen Ideal-Membership-Problems ist normalerweise nicht eindeutig.

Die numerische Instabilität der Gröbnerbasen wird hier bereits durch die Verwendung von H-Basen und einem entsprechenden Divisionsalgorithmus umgangen. Der

erste der oben genannten Punkte lässt sich durch das folgende Lemma beschreiben:

Lemma 4.56. *Sei $\Xi \subset \mathbb{R}^d$, $F \subset \Pi_d$ mit $\langle F \rangle = \mathfrak{I}(\Xi)$ eine H-Basis und $g \in \mathfrak{I}_{p,\varepsilon}(\Xi)$, dann ist*

$$\frac{\|(\nu_{\langle F \rangle}(g))(\Xi)\|_p}{\|g\|_2} \leq \varepsilon.$$

Beweis. Es gilt

$$\begin{aligned} \|g(\Xi)\|_p &= \left\| \sum_{f \in F} (g_f f)(\Xi) + (\nu_{\langle F \rangle}(g))(\Xi) \right\|_p = \left\| \sum_{f \in F} g_f(\Xi) \circ \underbrace{f(\Xi)}_{=0} + (\nu_{\langle F \rangle}(g))(\Xi) \right\|_p \\ &= \left\| (\nu_{\langle F \rangle}(g))(\Xi) \right\|_p \end{aligned}$$

und damit ist

$$\frac{\|(\nu_{\langle F \rangle}(g))(\Xi)\|_p}{\|g\|_2} = \frac{\|g(\Xi)\|_p}{\|g\|_2} \leq \varepsilon.$$

□

Damit ist nicht zu erwarten, dass der Divisionsalgorithmus 2.49 für ein Polynom $g \in \mathfrak{I}_{p,\varepsilon}(\Xi)$ das Nullpolynom als Rest bzgl. F liefert. Vielmehr sollte der Divisionsrest ebenso bzgl. der Toleranz ε betrachtet werden. Diese Fragestellung wurde bereits in [HKPP09] untersucht und führte zum Begriff der *approximativen Normalform*. Im Kontext dieser Arbeit lautet die Definition dazu wie folgt:

Definition 4.57. *Sei $F \subset \Pi_d$ eine H-Basis des Ideals $\mathcal{F}$ und $g \in \Pi_d$, $\deg(g) = k$. Sei weiterhin $\nu_{\mathcal{F}}(g) = r^{(0)} + \cdots + r^{(k)}$, $r^{(j)} \in \mathcal{W}^0_{j,d}(F) \subset \Pi^0_{j,d}$, und $\{w_1^{(j)}, \ldots, w_{\mu_j}^{(j)}\}$ eine Basis von $\mathcal{W}^0_{j,d}(F)$, sodass die Normalform von f dargestellt werden kann als*

$$\nu_{\mathcal{F}}(g) = \sum_{j=0}^{k} \sum_{i=1}^{\mu_j} c_{i,j} w_i^{(j)}, \quad c_{i,j} \in \mathbb{R}.$$

Dann ist die approximative Normalform *zur Toleranz $\varepsilon > 0$ definiert als*

$$\nu_{\mathcal{F},\varepsilon}(g) = \sum_{j=0}^{k} \sum_{i=1}^{\mu_j} \widetilde{c}_{i,j} w_i^{(j)}, \quad \widetilde{c}_{i,j} = \begin{cases} c_{i,j} & \text{falls } |c_{i,j}| > \varepsilon, \\ 0 & \text{sonst.} \end{cases}$$

Zur Berechnung der approximativen Normalform ist lediglich ein Schritt in Algorithmus 2.49 zu ergänzen. Da sowohl die Zerlegung $\nu_{\mathcal{F}}(g) = \sum_{j=0}^{k} r^{(j)}$ im Sinne der direkten Zerlegung $\mathcal{W}_{k,d} = \bigoplus_{j=0}^{k} \mathcal{W}_{j,d}^{0}$ als auch die Darstellung $r^{(j)} = \sum_{i=1}^{\mu_j} \widetilde{c}_{i,j} w_i^{(j)}$ bereits vorliegen, vgl. Abschnitt 2.4.5 und insbesondere (2.20), gilt es nur noch, die Koeffizienten $c_{i,j}$ zu überprüfen und ggf. zu streichen. In der Implementierung der Polynomdivision ist dies bereits realisiert, vgl. Anhang A.2.

Mit Hilfe der approximativen Normalform lässt sich die Aussage „Ein Polynom liegt fast im Ideal“ nun konkretisieren, vgl. [HKPP09]: Ein Polynom $g \in \Pi_d$ liegt ε-approximativ in $\mathcal{F}$, falls $\nu_{\mathcal{F},\varepsilon}(g) = 0$ gilt. Diese Beschreibung schließt auch die oben genannte Mehrdeutigkeit von Lösungen des expliziten approximativen Ideal-Membership-Problems mit ein: In der Darstellung $g = \sum_{f \in F} g_f \cdot f + \nu_{\mathcal{F}}(g)$ können alle $c_{i,j}$ mit Betrag kleiner oder gleich ε durch $\hat{c}_{i,j} \in \mathbb{R}$, $|\hat{c}_{i,j}| \leq \varepsilon$, ersetzt werden. Dies liefert ein neues Polynom $\hat{g}$ mit gleicher approximativer Normalform, d. h. $\nu_{\mathcal{F},\varepsilon}(g) = \nu_{\mathcal{F},\varepsilon}(\hat{g})$.

In [HKPP09] bezeichnet man die Änderungen eines Polynoms auf die oben beschriebene Weise als *almost syzygy*. Dies ist durch den Begriff *Syzygie* motiviert, der zu $F = \{f_1, \ldots, f_n\}$ einen Vektor $(s_1, \ldots, s_n)$, $s_j \in \Pi_d$, mit $\sum_{j=1}^{n} s_j f_j = 0$ beschreibt. Für weitere Details zu diesem Thema sei auf [CLO07, Kapitel 2, §9] verwiesen. Das folgende Beispiel verdeutlicht die Mehrdeutigkeit der approximativen Normalform:

Beispiel 4.58. *Sei* $\mathcal{F} = \langle x_1^2 + 1, x_2^3 \rangle \subset \Pi_2$, *dann liegen alle Polynome*

$$g(x_1, x_2) = x_1^2 + \varepsilon_1 x_1 x_2 + \varepsilon_2 x_2^2 + \varepsilon_3 x_1 + \varepsilon_4 x_2 + 1, \quad |\varepsilon_j| \leq \varepsilon,$$

ε-approximativ in $\mathcal{F}$. Das Polynom $s(x_1, x_2) = \varepsilon_1 x_1 x_2 + \varepsilon_2 x_2^2 + \varepsilon_3 x_1 + \varepsilon_4 x_2$ ist demnach eine almost syzygy *von $f(x_1, x_2) = x_1^2 + 1$ im Sinne von [HKPP09].*

Liegt eine approximative H-Basis $F \subset \mathfrak{I}_{p,\varepsilon}(\Xi)$ vor, so ist bei der Interpretation der Ergebnisse ein weiterer Punkt zu beachten. Während für exakte Ideale $\mathcal{F} = \mathfrak{I}(\Xi)$ die Beziehung $(f - \nu_{\mathcal{F}}(f)) \in \mathcal{F}$ gilt, können wir im approximativen Fall nur

$$(f - \nu_{\langle F \rangle,\varepsilon}(f)) = \left(\sum_{f \in F} g_f \cdot f \right) \in \mathfrak{I}_{p,\widetilde{\varepsilon}}(\Xi)$$

garantieren, wobei $\widetilde{\varepsilon}$ mit Hilfe der Resultate aus Abschnitt 4.2 bestimmt werden kann.

4.4. Dünn besetzte H-Basen

In Abschnitt 3.2 wurde deutlich, dass ein Ideal $\mathcal{F} \subseteq \Pi_d$ im Allgemeinen keine eindeutige H-Basis besitzt. Daher stellt sich natürlich die Frage nach Basen mit besonderen Eigenschaften. Eine solche Eigenschaft besteht darin, die Basispolynome möglichst *dünn besetzt* – d. h. mit möglichst wenig von Null verschiedenen Koeffizienten – zu wählen. Dies ist zum einen für die effiziente Speicherung von Bedeutung. Für dünn besetzte Polynome ist es meist günstiger, nur die von Null verschiedenen Koeffizienten samt ihrer Position im Koeffizientenvektor zu speichern. Zum anderen erleichtert die Wahl möglichst dünner Basispolynome auch deren Interpretierbarkeit. So ist zwar $\langle \sqrt{2}x_1 + \sqrt{2}x_2, \sqrt{2}x_1 - \sqrt{2}x_2 \rangle = \langle x_1, x_2 \rangle$ und das Ideal wird in beiden Fällen durch eine reduzierte und normierte H-Basis beschrieben, die zweite Darstellung ist aber deutlich übersichtlicher. Insbesondere hängen dünn besetzte Basispolynome im besten Fall von weniger Variablen ab. Dieser Effekt wird im nächsten Kapitel in einer Anwendung genutzt.

Zunächst muss jedoch ein Maß für die Besetztheit eines Koeffizientenvektors festgelegt werden. Dazu eignet sich die in Abschnitt 4.2.3 definierte 0-*Norm*, die alle von Null verschiedenen Einträge eines Vektors zählt. Eine triviale Abschätzung für die 0-Norm von $f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$ ist daher durch $0 \leq \|f\|_0 \leq \#\boldsymbol{T}_{k,d}$ gegeben. Das folgende Lemma trifft eine Aussage über das Verhalten der 0-Norm bzgl. der Addition:

Lemma 4.59. *Seien $f, g \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$, dann gilt*

$$\|f+g\|_0 \geq \|f\|_0 + \|g\|_0 - 2 \cdot \#\left(\operatorname{supp}(f) \cap \operatorname{supp}(g)\right), \tag{4.30}$$

$$\|f+g\|_0 \leq \|f\|_0 + \|g\|_0 - \#\left(\operatorname{supp}(f) \cap \operatorname{supp}(g)\right). \tag{4.31}$$

Beweis. Für die Summe $f+g$ gilt $\operatorname{supp}(f+g) \subseteq \operatorname{supp}(f) \cup \operatorname{supp}(g)$ und mit

$$\#\left(\operatorname{supp}(f) \cup \operatorname{supp}(g)\right) = \|f\|_0 + \|g\|_0 - \#\left(\operatorname{supp}(f) \cap \operatorname{supp}(g)\right)$$

folgt die Behauptung (4.31). Andererseits muss $\operatorname{supp}(f)\setminus\operatorname{supp}(g) \subseteq \operatorname{supp}(f+g)$ und $\operatorname{supp}(g)\setminus\operatorname{supp}(f) \subseteq \operatorname{supp}(f+g)$ gelten, da eine Auslöschung im Sinne von $f_\alpha = -g_\alpha$ nur für $\alpha \in \operatorname{supp}(f)\cap\operatorname{supp}(g)$ auftreten kann. Mit

$$\begin{aligned}&(\operatorname{supp}(f)\setminus\operatorname{supp}(g)) \cup (\operatorname{supp}(g)\setminus\operatorname{supp}(f))\\ &\quad= (\operatorname{supp}(f)\cup\operatorname{supp}(g)) \setminus (\operatorname{supp}(f)\cap\operatorname{supp}(g))\\ &\quad\subseteq \operatorname{supp}(f+g)\end{aligned}$$

und

$$\#\Big((\operatorname{supp}(f)\cup\operatorname{supp}(g))\setminus(\operatorname{supp}(f)\cap\operatorname{supp}(g))\Big) = \|f\|_0+\|g\|_0-2\#\left(\operatorname{supp}(f)\cap\operatorname{supp}(g)\right)$$

erhalten wir (4.30). □

Aus Lemma 4.59 folgt direkt die Dreiecksungleichung. Ebenso ist offensichtlich $\|f\|_0 = 0$ genau dann, wenn $f = 0$. Dennoch ist die 0-Norm – im Widerspruch zu ihrer Bezeichnung – *keine* Norm, denn die Normeigenschaft der absoluten Homogenität wird nicht erfüllt. Es gilt $\|\lambda\cdot f\|_0 \neq |\lambda|\cdot\|f\|_0$ für $\lambda \in \mathbb{R}\setminus\{-1,0,1\}$. Stattdessen erhält man folgende wichtige Eigenschaft:

$$\|\lambda f\|_0 = \|f\|_0, \quad f \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}},\ 0 \neq \lambda \in \mathbb{R}. \tag{4.32}$$

Die 0-Norm ist also *skalierungsinvariant*. Dies erlaubt es, die Abschätzung (4.30) auch für Linearkombinationen zweier Vektoren zu formulieren.

Korollar 4.60. *Seien $0 \neq a, b \in \mathbb{R}$ und $f, g \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$, dann ist*

$$\|af+bg\|_0 \geq \|f\|_0 + \|g\|_0 - 2\cdot\#\left(\operatorname{supp}(f)\cap\operatorname{supp}(g)\right).$$

Da in numerischer Rechnung häufig der Fall $0 \neq f_\alpha \approx 0$ auftritt, also Koeffizienten sehr nahe bei Null liegen und somit keine Relevanz haben, ist es sinnvoll, die Definition der 0-Norm im approximativen Sinne zu modifizieren. Die im Folgenden definierte *approximative 0-Norm* zählt nur Koeffizienten, die betragsmäßig über einer vorgegebenen Schranke liegen.

Definition 4.61. *Sei* $0 \neq f \in \Pi_d$ *und* $\varepsilon \geq 0$*, dann ist die* approximative 0-Norm *definiert durch*

$$\|f\|_{0,\varepsilon} = \#\left\{\alpha \in \mathbb{N}_0^d : \frac{|f_\alpha|}{\|f\|_2} > \varepsilon\right\}. \tag{4.33}$$

Für $\varepsilon = 0$ und $f \neq 0$ ist diese Definition mit der exakten 0-Norm kompatibel und es gilt $\|f\|_0 = \|f\|_{0,0}$. Durch die Normierung in (4.33) bleibt die Skalierungsinvarianz der 0-Norm auch für $\varepsilon > 0$ bestehen, denn es gilt

$$\|\lambda f\|_{0,\varepsilon} = \#\left\{\alpha \in \mathbb{N}_0^d : \frac{|\lambda| \cdot |f_\alpha|}{|\lambda| \cdot \|f\|_2} > \varepsilon\right\} = \|f\|_{0,\varepsilon}, \quad 0 \neq f \in \Pi_d, \lambda \neq 0.$$

Damit können wir nun folgender Problemstellung nachgehen:

> Zu einer endlichen Menge von Polynomen $F \subset \Pi_d$ und einer Toleranz $\varepsilon \geq 0$ bestimme man ein $f^{\min} \in \operatorname{span}(F)\backslash\{0\}$ mit $f^{\min} = \min_{f \in \operatorname{span}(F)\backslash\{0\}} \|f\|_{0,\varepsilon}$.

Da $\operatorname{span}(F)$ als linearer Vektorraum immer auch das Nullpolynom enthält, schließen wir diese triviale Lösung explizit aus. Wir identifizieren nun wie üblich die Menge $F = \{f_1, \ldots, f_m\} \subset \Pi_d$ mit der Matrix der Koeffizientenvektoren $F \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$. Weiter fordern wir $\operatorname{rank}(F) = m$, denn für $\operatorname{rank}(F) < m$ gibt es eine nichttriviale Darstellung des Nullpolynoms, das jedoch als Lösung ausgeschlossen wurde. Da die approximative 0-Norm skalierungsinvariant ist, können wir annehmen, dass es ein $\alpha \in \mathbb{N}_0^d$ mit $f_\alpha^{\min} = 1$ gibt. Wegen $f^{\min} \in \operatorname{span}(F) \setminus \{0\}$ und $\operatorname{rank}(F) = m$ lässt sich ein eindeutiger Vektor $c \in \mathbb{R}^m$ mit

$$F^T \cdot c = (f^{\min})^T \tag{4.34}$$

bestimmen. Ist nun $\|f^{\min}\|_{0,\varepsilon} = n$, so existiert eine Indexmenge $I \subseteq \{1, \ldots, \#\boldsymbol{T}_{k,d}\}$ mit $\#I = \#\boldsymbol{T}_{k,d} - n + 1$, sodass

$$\|(F_I^T \cdot c) - e_j\|_\infty = \|(f_I^{\min})^T - e_j\|_\infty < \varepsilon \|F^T \cdot c\|_2, \quad e_j = (\underbrace{0, \ldots, 0}_{j-1}, 1, 0, \ldots, 0)^T, \tag{4.35}$$

für ein $j \in \{1, \ldots, \#\boldsymbol{T}_{k,d} - n + 1\}$ gilt.

Algorithmus 4.62 : Approximative 0-Norm Minimierung

Input : $F \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$, $\operatorname{rank}(F) = m$, $\varepsilon \geq 0$
Output : $f^{\min} \in \mathbb{R}^{\#\boldsymbol{T}_{k,d}}$

```
1  for n = 1, ..., #T_{k,d} − 1 do
2      foreach I ⊆ {1, ..., #T_{k,d}}, #I = #T_{k,d} − n + 1 do
3          for j = 1, ..., #T_{k,d} − n + 1 do
4              Bestimme Least-Squares Lösung von F_I^T c = e_j ;
5              if ‖F_I^T c − e_j‖_∞ < ε‖F^T c‖_2 then
6                  f^min ← (F^T · c)^T
7                  return f^min
8              end
9          end
10     end
11 end
```

Die sukzessive Überprüfung der Bedingung (4.35) für alle möglichen Indexmengen mit $\#I = \#\boldsymbol{T}_{k,d} - n + 1$ und alle Einheitsvektoren e_j liefert dann entweder eine Lösung für $f^{\min}$ oder es muss $\|f^{\min}\|_{0,\varepsilon} > n$ gelten. Algorithmus 4.62 iteriert dieses Konzept für $n \geq 1$ aufsteigend und bestimmt somit ein bzgl. der approximativen 0-Norm minimales Element von $\operatorname{span}(F) \setminus \{0\}$.

Man beachte, dass sich das in (4.34) beschriebene Minimierungsproblem grundsätzlich von der aus dem *Compressed Sensing* bekannten Fragestellung der *Sparse Recovery* unterscheidet, vgl. [EK12]. Zur Rekonstruktion eines dünn besetzten Signals wird dort das Problem $\min_{0 \neq x \in \mathbb{R}^m} \|x\|_0$ unter der Nebenbedingung $Ax = b$, $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, gelöst. Man sucht also nach einem dünn besetzten *Lösungsvektor*. Das in (4.34) formulierte Problem besteht hingegen in der Suche nach einer dünn besetzten *rechten Seite* des linearen Gleichungssystems. Aus diesem Grund lassen sich die bekannten Lösungsansätze für das *Sparse Recovery* Problem nicht auf die hier gegebene Fragestellung übertragen.

Aufgrund der Auswahlmöglichkeiten für die Indexmenge I in Zeile 2 von Algorithmus 4.62 hat das Verfahren nichtpolynomielle Laufzeit. Die Anzahl aller möglichen Mengen I entspricht der Mächtigkeit der Potenzmenge von $\{1, \ldots, \#\boldsymbol{T}_{k,d}\}$, also $2^{\#\boldsymbol{T}_{k,d}}$. Es ist jedoch nicht notwendig alle diese Indexmengen zu betrachten, denn es gilt folgende Beschränkung:

Lemma 4.63. *Seien* $\varepsilon \geq 0$ *und* $F \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$ *gegeben mit* $\#\boldsymbol{T}_{k,d} \geq m$ *und* $\operatorname{rank}(F) = m$*, dann ist* $\min_{c \in \mathbb{R}^m} \|F^T c\|_{0,\varepsilon} \leq \#\boldsymbol{T}_{k,d} - m + 1$.

Beweis. Da F vollen Zeilenrang hat, muss es eine Indexmenge $I \subset \{1, \ldots, \#\boldsymbol{T}_{k,d}\}$ mit $\#I = m$ geben, sodass F_I regulär ist. Dies bedeutet aber, dass das Gleichungssystem $F_I^T c = e_j$ für jedes $j = 1, \ldots, m$ eindeutig lösbar ist. Damit ist $\|F^T c\|_0 \leq \#\boldsymbol{T}_{k,d} - m + 1$ und insbesondere $\|F^T c\|_{0,\varepsilon} \leq \#\boldsymbol{T}_{k,d} - m + 1$. □

Insgesamt lässt sich folgende *worst-case* Laufzeitabschätzung der 0-Norm Minimierung angeben:

Satz 4.64. *Sei* $F \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$*, dann sind zur Bestimmung eines bzgl. der* 0*-Norm minimalen Elements von* $\operatorname{span}(F) \setminus \{0\}$ *durch Algorithmus 4.62 höchstens*

$$\sum_{n=0}^{\#\boldsymbol{T}_{k,d}-m} \binom{\#\boldsymbol{T}_{k,d}}{n} (\#\boldsymbol{T}_{k,d} - n)$$

Gleichungssysteme zu lösen.

Beweis. Es gibt $\binom{\#\boldsymbol{T}_{k,d}}{\#\boldsymbol{T}_{k,d}-n+1}$ Teilmengen von $\{1, \ldots, \#\boldsymbol{T}_{k,d}\}$ mit genau $(\#\boldsymbol{T}_{k,d}-n+1)$ Elementen. Zu jeder dieser Teilmengen müssen $(\#\boldsymbol{T}_{k,d} - n + 1)$ Gleichungssysteme gelöst werden. Wegen Lemma 4.63 bricht das Verfahren spätestens bei $n = \#\boldsymbol{T}_{k,d} - m + 1$ ab. Zusammen ergibt sich

$$\begin{aligned}\sum_{n=1}^{\#\boldsymbol{T}_{k,d}-m+1} \binom{\#\boldsymbol{T}_{k,d}}{\#\boldsymbol{T}_{k,d}-n+1} (\#\boldsymbol{T}_{k,d} - n + 1) &= \sum_{n=1}^{\#\boldsymbol{T}_{k,d}-m+1} \binom{\#\boldsymbol{T}_{k,d}}{n-1} (\#\boldsymbol{T}_{k,d} - n + 1) \\ &= \sum_{n=0}^{\#\boldsymbol{T}_{k,d}-m} \binom{\#\boldsymbol{T}_{k,d}}{n} (\#\boldsymbol{T}_{k,d} - n)\end{aligned}$$

und damit die Behauptung. □

Sind wir nicht nur an einem möglichst dünn besetzten Polynom aus $\operatorname{span}(F) \setminus \{0\}$, sondern vielmehr an einer Darstellung des von F aufgespannten linearen Raums durch eine möglichst dünn besetzte Basis interessiert, so können wir Algorithmus 4.62 durch folgenden Ansatz erweitern, vgl. Algorithmus 4.65:

Algorithmus 4.65 : Vollständige approximative 0-Norm Minimierung

Input : $F \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$, $\operatorname{rank}(F) = m$, $\varepsilon \geq 0$
Output : $F^{\min} \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$

```
C ← ∅
for n = 1, ..., #T_{k,d} − 1 do
    foreach I ⊆ {1, ..., #T_{k,d}}, #I = #T_{k,d} − n + 1 do
        for j = 1, ..., #T_{k,d} − n + 1 do
            Bestimme Least-Squares Lösung von F_I^T c = e_j
            if ‖F_I^T c − e_j‖_∞ < ε‖F^T c‖_2 & rank([C, c]) > rank(C) then
                C ← [C, c]
            end
            if rank(C) = rank(F) then
                return F^min ← (F^T · C)^T
            end
        end
    end
end
```

Die Suche bricht nicht nach einem gefundenen Minimum ab, sondern wird fortgesetzt bis $m = \operatorname{rank}(F)$ linear unabhängige Koeffizientenvektoren gefunden wurden. Diese spannen dann den Zeilenraum von F auf und sind minimal bzgl. der approximativen 0-Norm.

Nun bleibt noch die Frage, wie sich die Verfahren zur Minimierung der approximativen 0-Norm auf eine H-Basis $F \subset \Pi_d$ anwenden lassen. Eine wesentliche Bedingung dafür wurde bereits in Abschnitt 3.2 formuliert: Es können nur Polynome gleichen Grades gegeneinander ausgetauscht werden und die linearen Räume $\mathcal{V}^0_{k,d}(F)$ müssen erhalten bleiben. Der folgende Satz zeigt, dass diese Forderungen erfüllt sind, wenn wir eine Teilmenge $F_k = F \cap (\Pi_{k,d} \setminus \Pi_{k-1,d}) \subseteq F$ durch eine bzgl. der approximativen 0-Norm minimale, linear unabhängige Menge aus $\Pi_{k,d} \setminus \Pi_{k-1,d}$ mit dem gleichen linearen Aufspann ersetzen:

Satz 4.66. *Sei $F \subset \Pi_d$ eine H-Basis und $F_k = F \cap (\Pi_{k,d} \setminus \Pi_{k-1,d}) = \{f_1, \ldots, f_m\}$ mit $(f_i, f_j) = \delta_{i,j}$. Identifizieren wir die Koeffizientenvektoren von F_k mit den Zeilen einer Matrix $F_k \in \mathbb{R}^{m \times \#\boldsymbol{T}_{k,d}}$, so gilt für alle $\widetilde{k} \in \mathbb{N}_0$*

$$\mathcal{V}^0_{\widetilde{k},d}(F_k) = \mathcal{V}^0_{\widetilde{k},d}((F_k^T \cdot C)^T)$$

für reguläre Matrizen $C \in \mathbb{R}^{m\times m}$.

Beweis. Nach Voraussetzung ist C regulär und die Polynome f_j sind paarweise orthogonal, also erhalten wir $\operatorname{rank}(F_k) = \operatorname{rank}(F_k \cdot C) = m$ und $\operatorname{span}(F_k^T) = \operatorname{span}(F_k^T \cdot C)$. Da die Polynome F_k Teil einer H-Basis sind, gilt weiterhin $\deg(f) = k$ für alle $f \in \operatorname{span}(F_k^T)\setminus\{0\}$. Insbesondere erhalten wir $\mathcal{V}^0_{\widetilde{k},d}(F_k^T) = \{0\} = \mathcal{V}^0_{\widetilde{k},d}(F_k^T \cdot C)$ für alle $\widetilde{k} < k$.

Sei nun $\widetilde{k} \geq k$. Bezeichnen wir die linear unabhängigen Zeilen von $(F_k^T \cdot C)^T$ mit $\widetilde{f}_1, \ldots, \widetilde{f}_m$, so gilt

$$\widetilde{f}_j = \sum_{i=1}^{m} C_{i,j} f_j, \quad 1 \leq j \leq m.$$

Wegen $\operatorname{rank}(\Lambda(F_k^T)) = \operatorname{rank}(\Lambda(F_k^T \cdot C)) = m$ und $\operatorname{span}(\Lambda(F_k^T)) = \operatorname{span}(\Lambda(F_k^T \cdot C))$ folgt

$$\begin{aligned}
\mathcal{V}^0_{\widetilde{k},d}((F_k^T \cdot C)^T) &= \operatorname{span}\left\{x^\alpha \Lambda(\widetilde{f}_j) : 1 \leq j \leq m,\ \alpha \in \mathbb{N}_0^d,\ |\alpha| = \widetilde{k} - k\right\} \\
&= \operatorname{span}\left\{x^\alpha \sum_{i=1}^{m} C_{i,j}\Lambda(f_i) : 1 \leq j \leq m,\ \alpha \in \mathbb{N}_0^d,\ |\alpha| = \widetilde{k} - k\right\} \\
&= \operatorname{span}\left\{\sum_{i=1}^{m} C_{i,j} x^\alpha \Lambda(f_i) : 1 \leq j \leq m,\ \alpha \in \mathbb{N}_0^d,\ |\alpha| = \widetilde{k} - k\right\} \\
&= \operatorname{span}\left\{x^\alpha \Lambda(f_j) : 1 \leq j \leq m,\ \alpha \in \mathbb{N}_0^d,\ |\alpha| = \widetilde{k} - k\right\} = \mathcal{V}^0_{\widetilde{k},d}(F_k).
\end{aligned}$$

□

Für eine H-Basis $F \subset \Pi_d$ des Ideals $\mathcal{F}$ mit

$$F = \bigcup_{j=0}^{n} F_j, \quad F_j \subset \Pi_{j,d} \setminus \Pi_{j-1,d},$$

können nun die Koeffizientenmatrizen F_j durch Algorithmus 4.65 bzgl. der approximativen 0-Norm minimiert werden. Seien $F_j^{\min} = (F_j^T \cdot C^{(j)})^T$ die Ergebnisse dieser Minimierung, dann gilt

$$\Lambda(\mathcal{F}) \cap \Pi^0_{k,d} = \mathcal{V}^0_{k,d}(F) = \bigoplus_{j\leq k} \mathcal{V}^0_{k,d}(F_j) = \bigoplus_{j\leq k} \mathcal{V}^0_{k,d}(F_j^{\min}), \quad k \in \mathbb{N}_0.$$

Nach Satz 3.10 ist damit auch $F^{\min} = \bigcup_{j \le n} F_j^{\min}$ eine H-Basis des Ideals $\mathcal{F}$.

Dieses Ergebnis lässt sich ebenfalls auf das bereits in Abschnitt 3.2 zitierte Resultat von Möller und Sauer aus [MS00a, Theorem 6.5] zurückführen, wobei sich die Aussagen in einem Punkt unterscheiden: Möller und Sauer fordern, dass die Matrizen $C^{(j)}$ orthogonal sind, wohingegen hier nur die Regularität notwendig ist. Daraus resultiert, dass nach der Methode von Möller und Sauer eine mögliche Orthonormalitätsbeziehung zwischen den Basispolynomen erhalten bleibt. Für unsere Konstruktion gilt dies im Allgemeinen nicht.

Abschließend betrachten wir die Auswirkung einer Minimierung der approximativen 0-Norm auf approximative H-Basen. Unter Verwendung der in Abschnitt 4.2 hergeleiteten Abschätzungen für die Veränderung der Toleranz ε unter Linearkombinationen ergibt sich folgende Aussage:

Satz 4.67. *Sei $F = \bigcup_{j \le n} F_j$, $F_j \subset \Pi_{j,d} \setminus \Pi_{j-1,d}$, eine approximative H-Basis von $\mathfrak{I}_{p,\varepsilon}(\Xi)$, $\Xi \subset \mathbb{R}^d$. Ist nun $F^{\min} = \bigcup_{j \le n} F_j^{\min}$ das Ergebnis der Minimierung der approximativen 0-Norm von F_j, $j = 0, \ldots, n$, durch Algorithmus 4.65, so ist $F^{\min}$ eine approximative H-Basis von $\mathfrak{I}_{p,\widetilde{\varepsilon}}(\Xi)$ mit $\widetilde{\varepsilon} = \varepsilon \cdot \max_{j \le n} \sqrt{\#F_j}$.*

5 Analyse von Daten aus kinematischen Systemen

Inhalt

5.1. Planare kinematische Ketten 138

5.2. Kinematische Ketten im Raum 152

Nachdem nun viele theoretische Resultate über approximative Ideale und approximative H-Basen vorgestellt wurden, zeigt dieses Kapitel eine praktische Anwendung aus dem Bereich der *Kinematik* und *Robotik*. Dazu folgt zunächst eine kurze Übersicht über das Konzept der *kinematischen Ketten*, die sich im Wesentlichen an [HKSS97] orientiert. In der Mechanik beschreibt eine *kinematische Kette* ein System aus n Gliedern, die durch Gelenke verknüpft sind. Dabei wird gefordert, dass jedes Glied mit *mindestens einem* Gelenk verbunden ist. Strukturell lassen sich zwei Typen von kinematischen Ketten unterscheiden:

1. **Offene kinematische Ketten:** Kann eine kinematische Kette als Baumstruktur mit einer Wurzel, die als *Aufhängungspunkt* bzw. *Basis* bezeichnet wird, dargestellt werden, so handelt es sich um eine *offene kinematische Kette*. Typische Beispiele für solche Ketten sind Roboterarme.

2. **Geschlossene kinematische Ketten:** Besteht eine Verbindung zwischen Start und Ende der kinematischen Kette, so nennt man diese *geschlossen*. Beispiele für geschlossene kinematische Ketten sind unter anderem durch Koppelgetriebe gegeben, siehe dazu [HKSS97, Kapitel 1.1.2].

Eine äquivalente Definition für offene bzw. geschlossene kinematische Ketten aus [Ste12] lautet: Eine kinematische Kette ist *geschlossen*, wenn zu jedem Glied *genau zwei* Gelenke gehören – andernfalls ist die Kette *offen*. Wir werden im Folgenden stets offene Systeme betrachten.

Wichtige Anwendungen von kinematischen Ketten finden sich z. B. in der Berechnung der Bewegung von Industrierobotern (*inverses kinematisches Problem*, vgl. [CLO07, Kapitel 5, §3]), der Analyse menschlicher Bewegungen in den Sport- bzw. Bewegungswissenschaften (vgl. [MS07]) oder der Computeranimation von künstlichen Figuren (*Avataren*, siehe [Tüm07]). Dabei differenziert man typischerweise zwischen zwei Disziplinen anhand der vorgegebenen Problemstellung, vgl. [Ste12]:

1. **Analyse:** In der Analyse werden Zusammenhänge von Gelenken und Gliedern in einem gegebenen Mechanismus sowie deren Attribute wie Länge, Orientierung etc. untersucht.
2. **Synthese:** Die Synthese beschreibt die Konstruktion eines Mechanismus, der eine vorgegebene Bewegung realisiert.

Wir werden uns hier mit der Analyse von kinematischen Ketten beschäftigen. Dazu betrachten wir zunächst den *planaren* Fall, d. h. kinematische Ketten in der Ebene, und erweitern das Konzept anschließend auf kinematische Ketten im Raum.

5.1. Planare kinematische Ketten

Zur Untersuchung von *planaren kinematischen Ketten* betrachten wir folgendes Modell: Eine kinematische Kette bestehe aus n *Gelenken*, deren Positionen durch die Punkte $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1}) \in \mathbb{R}^2$ beschrieben werden. Zusätzlich sei an der Position $(x_n, y_n) \in \mathbb{R}^2$ ein Arbeitswerkzeug gegeben, das auch als *Manipulator* oder *Effektor* bezeichnet wird. Die Position des *Aufhängungspunkts* $(x_0, y_0) \in \mathbb{R}^2$ sei konstant, da die kinematische Kette an dieser Stelle fest mit ihrer Umgebung verbunden ist. Im Folgenden wird dabei $(x_0, y_0) = (0, 0)$ vorausgesetzt, da sich diese Forderung stets durch eine geeignete Verschiebung des Koordinatensystems herstellen lässt. Weiterhin seien je zwei Punkte (x_{j-1}, y_{j-1}) und (x_j, y_j), $j = 1, \ldots, n$, durch

ein Verbindungsglied der Länge ℓ_j verknüpft, sodass die Gelenkpositionen bzgl. der Gleichung

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \end{pmatrix} + \ell_j \cdot \begin{pmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{pmatrix}, \quad \alpha_j \in [0, 2\pi],\ \ell_j > 0, \tag{5.1}$$

voneinander abhängen. Man beachte, dass dieses Modell nicht für allgemeine kinematische Ketten anwendbar ist. Vielmehr fordern wir, dass die Kette *linear* aufgebaut ist, d. h. die Baumstruktur enthält *keine Schleifen* und *genau ein Blatt*. Diese Einschränkung dient jedoch nur der Vereinfachung unserer Darstellungen. Alle folgenden Resultate lassen sich auf allgemeine kinematische Ketten übertragen, indem man (5.1) auch für die Verknüpfung zweier Gelenkpositionen (x_j, y_j) und (x_k, y_k), $k > j + 1$, konstruiert.

Eine weitere Forderung an unser Modell besteht darin, dass in einem bestimmten Gelenk entweder *nur der Drehwinkel* α_j oder *nur die Länge* ℓ_j verändert werden – die jeweils andere Größe bleibt konstant. Dies motiviert die Unterscheidung zweier Gelenktypen, vgl. [Ste12]:

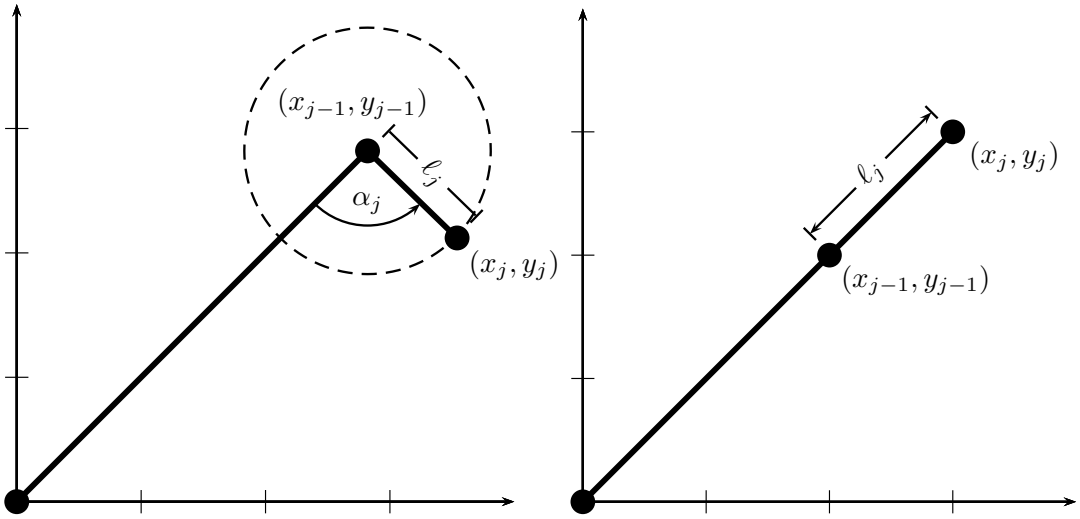
1. **Drehgelenke** (*revolute joints*), die eine Änderung des Drehwinkels erlauben,
2. **Schubgelenke** (*prismatic joints*), mit denen die Länge eines Glieds der Kette angepasst werden kann.

Eine schematische Darstellung dieser Gelenktypen ist in Abbildung 5.1 gegeben.

5.1.1. Charakterisierende Gelenkbedingungen

Im Folgenden werden diese beiden Gelenktypen genauer untersucht und charakterisierende Bedingungen in Abhängigkeit der Gelenkpositionen (x_j, y_j), $j = 1, \ldots, n$, festgelegt. Ausgehend von der allgemeinen Gelenkbeziehung (5.1), ergibt sich durch Umformung

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \end{pmatrix} + \ell_j \cdot \begin{pmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} (x_j - x_{j-1})^2 \\ (y_j - y_{j-1})^2 \end{pmatrix} = \ell_j^2 \cdot \begin{pmatrix} \cos^2(\alpha_j) \\ \sin^2(\alpha_j) \end{pmatrix}. \tag{5.2}$$



(a) Drehgelenk **(b)** Schubgelenk

Abbildung 5.1.: Schematische Darstellung der beiden Gelenktypen einer planaren kinematischen Kette.

Eine Addition der beiden Komponenten von (5.2) liefert $(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 = \ell_j^2 \left(\cos^2(\alpha_j) + \sin^2(\alpha_j)\right)$, bzw.

$$r_j(x_{j-1}, y_{j-1}, x_j, y_j) := (x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 - \ell_j^2 = 0, \quad j = 1, \ldots, n. \tag{5.3}$$

Damit erhalten wir eine *notwendige Bedingung* für die Existenz eines Drehgelenks an Position j einer planaren kinematischen Kette: Alle möglichen Paare von Gelenkpositionen $(x_{j-1}, y_{j-1}, x_j, y_j) \in \mathbb{R}^4$ müssen Nullstellen des quadratischen Polynoms (5.3) sein. Eine äquivalente Formulierung von (5.5) lautet

$$\left\| \begin{pmatrix} x_j - x_{j-1} \\ y_j - y_{j-1} \end{pmatrix} \right\|_2^2 = \ell_j^2,$$

was mathematisch genau der Bedingung „*Die (euklidische) Länge des Verbindungsglieds ändert sich nicht*“ entspricht.

Nun gilt es, eine entsprechende Bedingung für Schubgelenke zu entwickeln. Da sich der Drehwinkel in einem Schubgelenk nicht ändert, ist $\alpha_j = \alpha_{j-1}$. Dabei gelte die Konvention $j > 1$, da für $j = 1$ die Orientierung des Schubgelenks nicht bekannt

ist. Mit den allgemeinen Gelenkbedingungen

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \end{pmatrix} + \ell_j \cdot \begin{pmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} x_{j-1} \\ y_{j-1} \end{pmatrix} = \begin{pmatrix} x_{j-2} \\ y_{j-2} \end{pmatrix} + \ell_{j-1} \cdot \begin{pmatrix} \cos(\alpha_{j-1}) \\ \sin(\alpha_{j-1}) \end{pmatrix}$$

folgt dann

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \end{pmatrix} + \frac{\ell_j}{\ell_{j-1}} \cdot \begin{pmatrix} x_{j-1} - x_{j-2} \\ y_{j-1} - y_{j-2} \end{pmatrix}.$$

Durch Auflösen nach ℓ_j/ℓ_{j-1} und Gleichsetzen erhält man

$$\frac{x_j - x_{j-1}}{x_{j-1} - x_{j-2}} = \frac{\ell_j}{\ell_{j-1}} = \frac{y_j - y_{j-1}}{y_{j-1} - y_{j-2}}$$

bzw.

$$\begin{aligned} &p_j(x_{j-2}, y_{j-2}, x_{j-1}, y_{j-1}, x_j, y_j) \\ &\quad := (x_j - x_{j-1})\,(y_{j-1} - y_{j-2}) - (y_j - y_{j-1})\,(x_{j-1} - x_{j-2}) = 0, \quad j = 2, \ldots, n. \end{aligned} \tag{5.4}$$

Die *notwendige Bedingung* für die Existenz eines Schubgelenks an Position j einer planaren kinematischen Kette lautet demnach: Alle möglichen Tripel von Gelenkpositionen $(x_{j-2}, y_{j-2}, x_{j-1}, y_{j-1}, x_j, y_j) \in \mathbb{R}^6$ müssen Nullstellen des quadratischen Polynoms (5.4) sein. Auch diese Forderung lässt sich umformulieren zu

$$\Big((x_j - x_{j-1}), (y_j - y_{j-1})\Big) \cdot \begin{pmatrix} (y_{j-1} - y_{j-2}) \\ -(x_{j-1} - x_{j-2}) \end{pmatrix} = 0.$$

Da ebenfalls die Orthogonalitätsbeziehung

$$\Big((x_{j-1} - x_{j-2}), (y_{j-1} - y_{j-2})\Big) \cdot \begin{pmatrix} (y_{j-1} - y_{j-2}) \\ -(x_{j-1} - x_{j-2}) \end{pmatrix} = 0$$

gilt, stehen die Gelenkverbindungen von $(j-2)$ nach $(j-1)$ und von $(j-1)$ nach j senkrecht auf dem gleichen Vektor und müssen daher parallel sein. Dies entspricht genau der Bedingung „*Der Drehwinkel des Gelenks ändert sich nicht*“.

5.1.2. Methoden zur Gelenkerkennung

In diesem Abschnitt werden Verfahren vorgestellt, die aus einem *Bewegungsprofil* einer kinematischen Kette die Gelenkgeometrie rekonstruieren können. Der Begriff *Bewegungsprofil* steht dabei für eine endliche Menge von Positionen aller Gelenke, die als Vektor $(x_1, y_1, \ldots, x_n, y_n) \in \mathbb{R}^{2n}$ jeweils einen *Zustand* der kinematischen Kette beschreiben. In Abbildung 5.2 sind zwei Beispiele solcher Bewegungsprofile dargestellt. Man beachte, dass der Aufhängungspunkt (x_0, y_0) zwar formal auch eine Gelenkposition beschreibt, aber seine Koordinaten nie verändert werden. Daher ist es nicht notwendig, diesen Punkt in das Bewegungsprofil aufzunehmen. Im Folgenden sind Bewegungsprofile stets in Form einer Matrix $\Xi \subset \mathbb{R}^{N\times 2n}$ angegeben. Zudem werden die Bedingungen (5.3) und (5.4) für Dreh- bzw. Schubgelenke durch

$$r_j^{(n)}(x_1, y_1, \ldots, x_n, y_n) := r_j(x_{j-1}, y_{j-1}, x_j, y_j), \quad j = 1, \ldots, n \tag{5.5}$$

$$p_j^{(n)}(x_1, y_1, \ldots, x_n, y_n) := p_j(x_{j-2}, y_{j-2}, x_{j-1}, y_{j-1}, x_j, y_j), \quad j = 2, \ldots, n \tag{5.6}$$

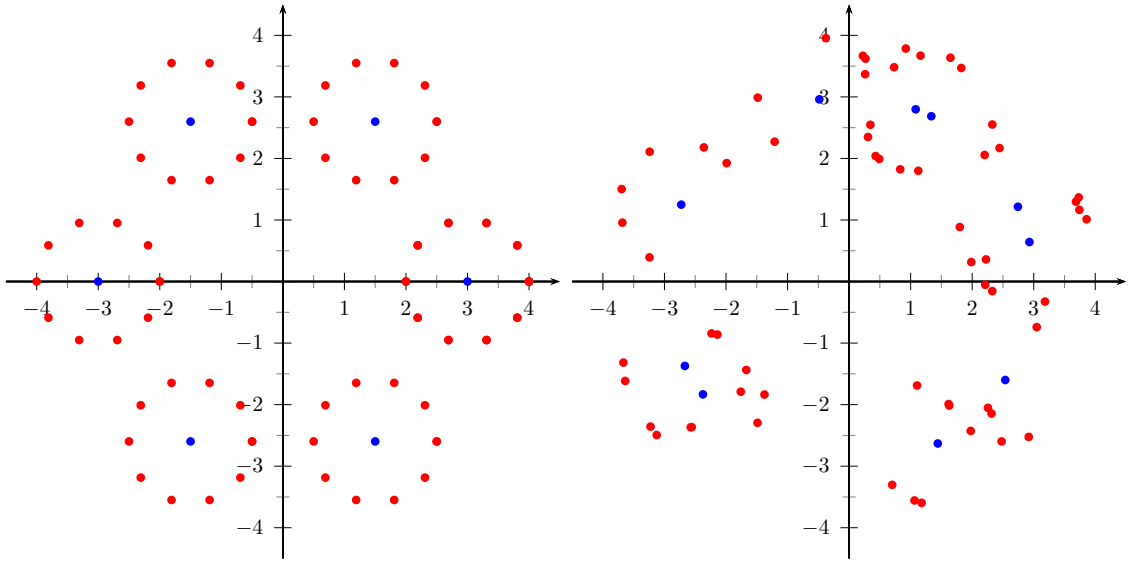
in den Polynomring $\mathbb{R}[x_1, y_1, \ldots, x_n, y_n] \simeq \Pi_{2n}$ in $2n$ Variablen eingebettet. Für den weiteren Verlauf dieses Kapitels sei stets dieser Polynomring sowie die dadurch gegebene Anordnung der Variablen vorausgesetzt. In graduiert-lexikographischer Termordnung gilt also z. B. für $n = 2$:

$$1 \prec y_2 \prec x_2 \prec y_1 \prec x_1 \prec y_2^2 \prec x_2 y_2 \prec x_2^2 \prec y_1 y_2 \prec x_2 y_1 \prec y_1^2 \prec \cdots$$

Damit können wir nun folgendes Kriterium festlegen:

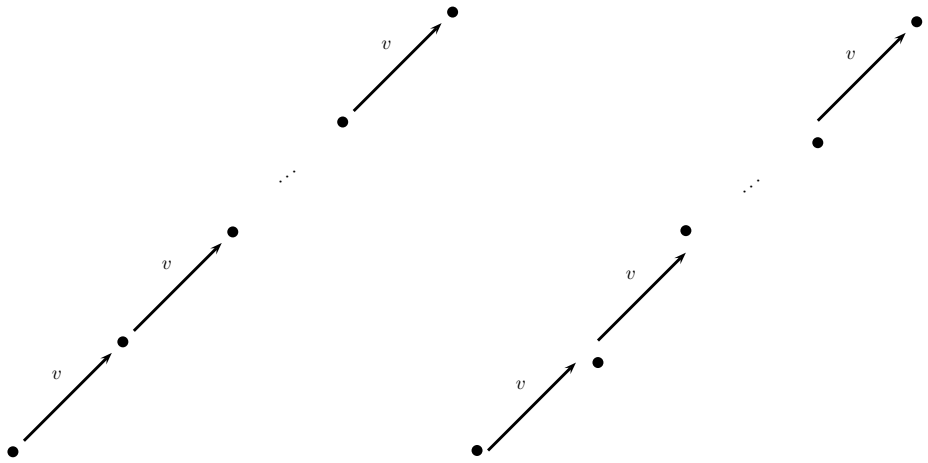
Satz 5.1. *Sei $\Xi \subset \mathbb{R}^{N\times 2n}$ ein Bewegungsprofil einer planaren kinematischen Kette. Die Kette hat genau dann ein Dreh- bzw. Schubgelenk an Position j, wenn $r_j^{(n)} \in \mathfrak{I}(\Xi)$ bzw. $p_j^{(n)} \in \mathfrak{I}(\Xi)$ gilt.*

In numerischer Rechnung ist das exakte Ideal $\mathfrak{I}(\Xi)$ in Satz 5.1 durch das approximative Ideal $\mathfrak{I}_{p,\varepsilon}(\Xi)$ zu ersetzen, da man für näherungsweise Daten nicht von $r_j^{(n)} \in \mathfrak{I}(\Xi)$ bzw. $p_j^{(n)} \in \mathfrak{I}(\Xi)$ ausgehen kann. Abbildung 5.3 verdeutlicht dies am Beispiel eines Schubgelenks. Damit ist es möglich, die Gelenkerkennung auf die Lösung entsprechender approximativer Ideal-Membership-Probleme zurückzuführen, sofern die Polynome $r_j^{(n)}$, $j = 1, \ldots, n$ und $p_j^{(n)}$, $j = 2, \ldots, n$ bereitgestellt werden.



(a) Gleichmäßig verteilte Drehwinkel **(b)** Zufällig verteilte Drehwinkel

Abbildung 5.2.: Bewegungsprofile einer planaren kinematischen Kette aus zwei Drehgelenken mit $\ell_1 = 3$ und $\ell_2 = 1$.



(a) Exakte Gelenkpositionen **(b)** Fehlerbehaftete Daten

Abbildung 5.3.: Bewegungsprofil eines Schubgelenks in Richtung v in exakter und fehlerbehafteter Darstellung. Im zweiten Fall enthält das Ideal $\mathfrak{I}(\Xi)$ für N Punkte nur Polynome vom Grad mindestens k, $\binom{k+1}{2} \leq N < \binom{k+2}{2}$.

Im Falle eines Schubgelenks lässt sich $p_j^{(n)}$ direkt angeben. Das charakterisierende Polynom eines Drehgelenks hängt jedoch noch von der *unbekannten* Länge ℓ_j ab. Daher verwenden wir folgenden Ansatz:

> Man konstruiere $\widetilde{r}_j^{(n)}$ aus $r_j^{(n)}$ durch Wahl der Länge $\ell_j = 0$ und wende den Divisionsalgorithmus 2.49 bzgl. einer approximativen H-Basis von $\mathfrak{I}_{p,\varepsilon}(\Xi)$ an. Nur, wenn dieser ein konstantes Polynom mit positivem Koeffizienten als Rest liefert, liegt an der Stelle j ein Drehgelenk vor. Die Länge des Drehgelenks entspricht dabei der positiven Wurzel des Rests.

Wir gehen nun davon aus, dass die approximative H-Basis $F \subset \mathfrak{I}_{\infty,\varepsilon}(\Xi)$ nur Polynome vom Grad mindestens 2 enthält, da es in unserem Modell keine linearen Gelenkbedingungen gibt. Ist dies nicht der Fall, so wurde entweder die Toleranz ε zu groß gewählt oder das Bewegungsprofil besteht aus zu wenigen bzw. schlecht gewählten Punkten. Ist $F_2 := F \cap \Pi_{2,2n}$ der quadratische Anteil von F, so liegt das erkannte charakterisierende Gelenkpolynom in $\mathfrak{I}_{\infty,\sqrt{\#F_2}\varepsilon}(\Xi)$.

Das oben beschriebene Verfahren löst also das Problem der Gelenkerkennung. Es bietet jedoch keine Möglichkeit, zu überprüfen, ob das Ideal bzw. die zugehörige Basis weitere noch nicht klassifizierte Polynome enthält. In Abschnitt 3.4 wurde gezeigt, dass durch die Berechnung einer Differenz von Idealen auch wichtige Informationen über die Varietät verloren gehen. Daher ist es *nicht sinnvoll*, bereits erkannte Gelenkpolynome schrittweise aus der Basis zu entfernen. Wir suchen vielmehr eine geeignete approximative H-Basis, deren quadratische Polynome bestenfalls den charakterisierenden Gelenkbedingungen $r_j^{(n)}$ bzw. $p_j^{(n)}$ entsprechen oder zumindest direkte Rückschlüsse darauf zulassen. In Satz 5.4 sowie den vorbereitenden Lemmata 5.2 und 5.3 wird eine Eigenschaft der Gelenkbedingungen gezeigt, die auch in einer H-Basis hergestellt werden kann.

Lemma 5.2. *Seien $r_j^{(n)}$, $j = 1, \ldots, n$, $n \geq 3$, die charakterisierenden Polynome von Drehgelenken einer planaren kinematischen Kette und $a = (a_3, \ldots, a_n) \in \mathbb{R}^{n-2}$ mit $\|a\|_0 > 0$, dann gilt*

$$\left\| \sum_{j=3}^{n} a_j r_j^{(n)} \right\|_0 \geq 7. \tag{5.7}$$

Beweis. Zunächst bemerken wir, dass die Koeffizientenvektoren der Polynome $r_j^{(n)}$ linear unabhängig sind und somit $\sum_{j=3}^{n} a_j r_j^{(n)} = 0$ nur für $a = (0, \dots, 0)$ auftreten kann, was nach Voraussetzung ausgeschlossen ist.

Wir zeigen die Behauptung durch Induktion über $m := \|a\|_0$. Da für Drehgelenke stets $\ell_j > 0$ vorausgesetzt wird, gilt $\|r_j^{(n)}\|_0 = 7$ und damit folgt die Behauptung für $m = 1$. Sei nun $j > k > 2$, dann ist

$$\#\left(\operatorname{supp}\left(r_j^{(n)}\right) \cap \operatorname{supp}\left(r_k^{(n)}\right)\right) = \begin{cases} 3 & \text{falls } j = k-1, \\ 1 & \text{falls } j > k-1. \end{cases}$$

Nach Korollar 4.60 folgt daraus für $a_j \neq 0 \neq a_k$ und $a_i = 0$, $j \neq i \neq k$, die Abschätzung $\left\|\sum_{j=3}^{n} a_j r_j^{(n)}\right\|_0 \geq 8$, was die Behauptung für $m = 2$ liefert.

Wir nehmen nun an, dass $m > 2$ und die Behauptung für alle $\tilde{a} \in \mathbb{R}^{n-2}$ mit $\|\tilde{a}\|_0 < m$ gilt. Sei weiter

$$0 \neq f = \underbrace{a_k r_k^{(n)}}_{=:f_k} + \underbrace{\sum_{j=3}^{k-1} a_j r_j^{(n)}}_{=:f_k^-} + \underbrace{\sum_{j=k+1}^{n} a_j r_j^{(n)}}_{=:f_k^+} \in \operatorname{span}\{r_3^{(n)}, \dots, r_n^{(n)}\}$$

mit $\|a\|_0 = m$ und k so gewählt, dass $f_k \neq 0$, $f_k^+ \neq 0$ und $f_k^- \neq 0$. Dies ist möglich, da a mindestens drei von Null verschiedene Einträge hat. Nun gilt nach Induktion $\|f_k^+\|_0 \geq 7$, $\|f_k^-\|_0 \geq 7$ und durch die Konstruktion von f_k^+ und f_k^- ist $\#(\operatorname{supp}(f_k^+) \cap \operatorname{supp}(f_k^-)) \leq 1$. Mit Korollar 4.60 erhalten wir $\|f_k^+ + f_k^-\|_0 \geq 12$. Da $\|f_k\|_0 = 7$ und nach Konstruktion $\#(\operatorname{supp}(f_k^+ + f_k^-) \cap \operatorname{supp}(f_k)) \leq 5$ gilt, können wir erneut Korollar 4.60 anwenden und erhalten $\|f\|_0 = \|f_k + (f_k^- + f_k^+)\|_0 \geq 9$. □

Dabei tritt die Gleichheit in Abschätzung (5.7) nur für $\|a\|_0 = 1$ ein, was bedeutet, dass die Polynome $r_j^{(n)}$, $j = 3, \dots, n$, in ihrem linearen Aufspann eine (bis auf Vielfache) eindeutige, bzgl. der 0-Norm minimale Basis bilden. Diese Eindeutigkeit geht allerdings verloren, wenn die Polynome $r_1^{(n)}$ und $r_2^{(n)}$ hinzukommen.

Lemma 5.3. *Seien $r_j^{(n)}$, $j = 1, \dots, n$, die charakterisierenden Polynome von Drehgelenken einer planaren kinematischen Kette und $\ell_3^2 \neq \ell_2^2 - \ell_1^2$, dann bilden die*

Polynome $r_j^{(n)}$, $j \neq 2$, *und* $\hat{r}_2^{(n)} := r_2^{(n)} - r_1^{(n)}$ *eine bzgl. der* 0*-Norm minimale Basis von* $\operatorname{span}\{r_1^{(n)}, \dots, r_n^{(n)}\}$.

Beweis. Lemma 5.2 zeigt die Behauptung für $j = 3, \dots, n$. Wir fügen nun schrittweise die Polynome $r_1^{(n)}$ und $r_2^{(n)}$ hinzu:

Für $r_1^{(n)}$ hängt die 0-Norm von der Position des Aufhängungspunktes ab, die wir wie oben beschrieben nicht als Variable sondern als Konstante auffassen. Gehen wir wie üblich von $(x_0, y_0) = (0, 0)$ aus, so gilt $\|r_1^{(n)}\|_0 = 3$ und $\#(\operatorname{supp}(r_1^{(n)}) \cap \operatorname{supp}(f)) = 1$ für $0 \neq f \in \operatorname{span}\{r_3^{(n)}, \dots, r_n^{(n)}\}$. Mit (5.7) folgt $\|f\|_0 \geq 7$ und Korollar 4.60 liefert $\|a_1 r_1^{(n)} + f\|_0 \geq 8$, $a_1 \neq 0$. Damit bilden die Polynome $r_1^{(n)}, r_3^{(n)}, \dots, r_n^{(n)}$ eine Basis von $\operatorname{span}\{r_1^{(n)}, r_3^{(n)}, \dots, r_n^{(n)}\}$, die bzgl. der 0-Norm minimal ist.

Ergänzen wir diese Basis um das Polynom $r_2^{(n)}$, so gibt es ein bzgl. der 0-Norm kleineres Polynom, denn es gilt $\|r_2^{(n)}\|_0 = 7$, aber $\| - r_1^{(n)} + r_2^{(n)}\|_0 = 5$, falls $\ell_1 \neq \ell_2$, bzw. sogar $\| - r_1^{(n)} + r_2^{(n)}\|_0 = 4$ für $\ell_1 = \ell_2$. Um die Minimalität der Basis bzgl. der 0-Norm zu erhalten, ist $r_2^{(n)}$ durch $\hat{r}_2^{(n)} := -r_1^{(n)} + r_2^{(n)}$ zu ersetzen. Wir betrachten nun $g = a_2 \hat{r}_2^{(n)} + f$ mit $f = a_1 r_1^{(n)} + \sum_{j=3}^{n} a_j r_j^{(n)}$ und $a_2 \neq 0$, $\|a\|_0 > 1$. Ist $\|a\|_0 > 2$, so gilt $\|f\|_0 \geq 8$ und $\#(\operatorname{supp}(\hat{r}_2^{(n)}) \cap \operatorname{supp}(f)) \leq 2$, falls $\ell_1 = \ell_2$, bzw. $\#(\operatorname{supp}(\hat{r}_2^{(n)}) \cap \operatorname{supp}(f)) \leq 3$, falls $\ell_1 \neq \ell_2$. Nach Korollar 4.60 folgt damit

$$\|g\|_0 \geq \begin{cases} 7 & \text{falls } \ell_1 \neq \ell_2, \\ 8 & \text{falls } \ell_1 = \ell_2. \end{cases}$$

Für $\|a\|_0 = 2$ sind erneut verschiedene Situationen in Abhängigkeit von a_1 und a_3 zu betrachten. Ist $a_1 = a_3 = 0$, so gilt $\#(\operatorname{supp}(\hat{r}_2^{(n)}) \cap \operatorname{supp}(f)) \leq 1$, $\|f\|_0 = 7$ und Korollar 4.60 liefert $\|g\|_0 \geq 9$. Im Fall $a_1 \neq 0$, $a_3 = 0$ erhalten wir mit $\|f\|_0 = 3$ und $\#(\operatorname{supp}(\hat{r}_2^{(n)}) \cap \operatorname{supp}(f)) \leq 1$ sogar nur $\|g\|_0 \geq 6$, was hier jedoch kein Problem darstellt, da die Polynome $r_1^{(n)}$ und $\hat{r}_2^{(n)}$ in $\operatorname{span}\{r_1^{(n)}, \hat{r}_2^{(n)}\}$ eine noch kleinere 0-Norm haben. Der letzte Fall, $a_1 = 0$, $a_3 \neq 0$, macht die Voraussetzung $\ell_3^2 \neq \ell_2^2 - \ell_1^2$ notwendig, da sonst für $\|f\|_0 = 7$ und $\#(\operatorname{supp}(\hat{r}_2^{(n)}) \cap \operatorname{supp}(f)) \leq 3$ die Situation $\|g\|_0 = 6$ auftreten kann und somit ein Polynom mit kleinerer 0-Norm in $\operatorname{span}\{\hat{r}_2^{(n)}, r_3^{(n)}\}$ liegen würde. Die Bedingung $\ell_3^2 \neq \ell_2^2 - \ell_1^2$ schließt dies explizit aus, sodass wir auch hier $\|g\|_0 \geq 7$ erhalten. □

Der Beweis von Lemma 5.3 hat gezeigt, dass $\operatorname{span}\{r_1^{(n)}, \dots, r_n^{(n)}\}$ verschiedene Basen mit minimaler 0-Norm besitzt. Die in Lemma 5.3 beschriebene Basis zeichnet sich jedoch dadurch aus, dass ihre Elemente den graduiert-lexikographisch größten Träger haben. Der folgende Satz schließt nun auch *Schubgelenke* mit ein:

Satz 5.4. *Seien die Voraussetzungen von Lemma 5.3 erfüllt und $J \subset \{2, \dots, n\}$ mit $j \in J \implies (j+1) \notin J$, dann bilden die Gelenkpolynome $r_k^{(n)}$, $k = 1, 3, \dots, n$, $\hat{r}_2^{(n)}$ und $p_j^{(n)}$, $j \in J$, eine bzgl. der 0-Norm minimale Basis von* $\operatorname{span}\{r_k^{(n)}, p_j^{(n)} : j \in J, 1 \le k \le n\}$.

Beweis. Für $f \neq 0$ in $\operatorname{span}\{r_k^{(n)}, p_j^{(n)} : j \in J, 1 \le k \le n\}$ gibt es eine Darstellung

$$f = \underbrace{\sum_{k=1}^{n} a_k r_k^{(n)}}_{=:f_r} + \underbrace{\sum_{j \in J} b_j p_j^{(n)}}_{=:f_p}, \quad a_k, b_j \in \mathbb{R}.$$

Nach Definition der Polynome $r_k^{(n)}$, und $p_j^{(n)}$ in (5.5) bzw. (5.6) gilt für deren Träger

$$\begin{aligned}
\alpha \in \operatorname{supp}\left(r_k^{(n)}\right) \implies (x_1, \dots, y_n)^\alpha \in {} & \{x_{i_1} x_{i_2} : 1 \le i_1, i_2 \le n\} \cup \{x_i : 1 \le i \le n\} \\
& \cup \{y_{i_1} y_{i_2} : 1 \le i_1, i_2 \le n\} \cup \{y_i : 1 \le i \le n\} \\
& \cup \{1\} \\
\alpha \in \operatorname{supp}\left(p_j^{(n)}\right) \implies (x_1, \dots, y_n)^\alpha \in {} & \{x_{i_1} y_{i_2} : 1 \le i_1, i_2 \le n\}
\end{aligned}$$

und damit $\operatorname{supp}(r_k) \cap \operatorname{supp}(p_j) = \emptyset$. Insbesondere sind auch die Träger von f_r und f_p disjunkt und es genügt, je eine bzgl. der 0-Norm minimale Basis der linearen Räume $\operatorname{span}\{r_1^{(n)}, \dots, r_n^{(n)}\}$ und $\operatorname{span}\{p_j^{(n)} : j \in J\}$ zu bestimmen. Für den ersten Teil liefert Lemma 5.3 die gesuchte Lösung durch $r_1^{(n)}, \hat{r}_2^{(n)}, r_3^{(n)}, \dots, r_n^{(n)}$. Für zwei Schubgelenke $p_j^{(n)}$, $p_k^{(n)}$ mit $j < k$ gilt

$$\begin{aligned}
p_j^{(n)} &= x_j y_{j-1} - x_j y_{j-2} + x_{j-1} y_{j-2} - x_{j-1} y_j + x_{j-2} y_j - x_{j-2} y_{j-1}, \\
p_k^{(n)} &= x_k y_{k-1} - x_k y_{k-2} + x_{k-1} y_{k-2} - x_{k-1} y_k + x_{k-2} y_k - x_{k-2} y_{k-1}.
\end{aligned}$$

Vergleicht man die beteiligten Terme, so erhält man $\operatorname{supp}(p_j^{(n)}) \cap \operatorname{supp}(p_k^{(n)}) = \emptyset$ für $j+1 < k$. Mit der Definition der Indexmenge J gilt dann $\bigcap_{j \in J} \operatorname{supp}(p_j^{(n)}) = \emptyset$, sodass nach Korollar 4.60 auch die Menge $\{p_j^{(n)} : j \in J\}$ minimal bzgl. der 0-Norm ist. □

Die Voraussetzung $J \subset \{2, \ldots, n\}$ mit $j \in J \implies (j+1) \notin J$ in Satz 5.4 ergibt sich auf natürliche Weise aus der Konstruktion planarer kinematischer Ketten: Zwei Schubgelenke, die direkt aufeinanderfolgen, sind *redundant* und lassen sich durch ein Schubgelenk mit größerer Länge ersetzen. Ebenso kann es per definitionem kein Schubgelenk p_1 geben, da zwei vorhergehende Gelenke bekannt sein müssen, um die Orientierung des Schubgelenks festzulegen.

Zur Gelenkerkennung aus einem Bewegungsprofil $\Xi \subset \mathbb{R}^{N \times 2n}$ mit hinreichend vielen Punkten, ist damit lediglich eine H-Basis des approximativen Ideals $\mathfrak{I}_{p,\varepsilon}(\Xi)$ zu bestimmen und diese bzgl. der approximativen 0-Norm zu minimieren. Die Anzahl der Punkte in Ξ muss dabei so groß sein, dass die Polynome des Endlichkeitsanteils mindestens vom Grad 3 sind. Gehen wir bei der Minimierung absteigend in graduiert-lexikographischer Ordnung vor, so lassen sich aus der so erzeugten Basis fast alle charakterisierenden Gelenkbedingungen der kinematischen Kette direkt ablesen. Dabei fällt auch die Redundanz zweier, per definitionem ausgeschlossener, aufeinanderfolgender Schubgelenke auf: Die approximative H-Basis vergrößert sich um ein Element, da zusätzlich zu den beiden Schubgelenken $p_j^{(n)}$ und $p_{j+1}^{(n)}$ noch ein weiteres Schubgelenk möglich ist: Die direkte Verbindung von (x_{j-1}, y_{j-1}) und (x_{j+1}, y_{j+1}). Die durch die Minimierung der approximativen 0-Norm entstandene Veränderung der Toleranz ε lässt sich nach Satz 4.67 durch $\widetilde{\varepsilon} = \sqrt{\#F_2} \cdot \varepsilon$ angeben und entspricht damit der Abschätzung, die wir für die Gelenkerkennung mittels Polynomdivision gemacht haben. Das folgende Beispiel zeigt das Resultat dieser Methode und die Rekonstruktion der fehlenden Gelenkbedingung $r_2^{(n)}$:

Beispiel 5.5. *Sei $\Xi \subset \mathbb{R}^6$ ein Bewegungsprofil einer planaren kinematischen Kette mit drei Drehgelenken und Abständen $\ell_1 = 3$, $\ell_2 = 4$, $\ell_3 = 6$. Durch Bestimmung einer 0-Norm minimalen approximativen H-Basis von $\mathfrak{I}_{\varepsilon,\infty}(\Xi)$ erhalten wir die folgenden quadratischen Polynome:*

$$\begin{aligned} f_1 &= x_1^2 + y_1^2 - 9, \\ f_2 &= x_1 x_2 + y_1 y_2 - 0.5 x_2^2 - 0.5 y_2^2 + 3.5, \\ f_3 &= (x_3 - x_2)^2 + (y_3 - y_2)^2 - 36. \end{aligned}$$

Das erste und das dritte Polynom entsprechen offensichtlich genau den definierenden Polynomen von Drehgelenken mit den Längen $\ell_1 = 3$ *und* $\ell_3 = 6$*. Um auch das zweite Polynom in die richtige Form zu bringen, berechnen wir*

$$f_1 - 2f_2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 - 16$$

und erhalten so $\ell_2 = 4$*.*

Zum Vergleich der beiden Methoden zur Gelenkerkennung wurden alle Varianten einer planaren kinematischen Kette mit $2 \leq n \leq 4$ Gelenken betrachtet und eine Minimierung der approximativen 0-Norm sowie eine Lösung des expliziten Ideal-Membership-Problems durchgeführt. Die Laufzeiten sind in Tabelle 5.1 dokumentiert. Zur Vollständigkeit wurden dabei auch kinematische Ketten mit mehreren aufeinanderfolgenden Schubgelenken einbezogen. Es ist jedoch zu beachten, dass die bzgl. der approximativen 0-Norm minimierte H-Basis in diesem Fall *nicht notwendigerweise* die charakterisierenden Gelenkpolynome enthält und somit zur Gelenkerkennung ungeeignet ist, man vergleiche die Voraussetzungen von Satz 5.4. Der Ansatz der Polynomdivision ist hingegen nach wie vor nutzbar. Weiterhin geben wir die Anzahl der beteiligten quadratischen Polynome im Sinne der Dimension von F_2 sowie die minimale und maximale approximative 0-Norm eines Polynoms in $\operatorname{span}(F_2)$ an. Insbesondere am Wert von $\dim(F_2)$ kann man die Redundanz aufeinanderfolgender Schubgelenke erkennen.

Abbildung 5.4 verdeutlicht die Rechenzeiten der beiden Methoden zur Gelenkerkennung noch einmal graphisch in logarithmischer Skalierung, wobei hier nur kinematische Ketten betrachtet wurden, die ausschließlich Drehgelenke enthalten. Für eine Kette mit $n = 7$ Drehgelenken ist die Minimierung der approximativen 0-Norm bereits um den Faktor 168 langsamer als die Lösung des expliziten Ideal-Membership-Problems durch Polynomdivision mit Rest.

Zusammenfassend lässt sich Folgendes festhalten: Liegt eine approximative H-Basis des Verschwindungsideals einer durch ein Bewegungsprofil beschriebenen planaren kinematischen Kette vor, so kann man mittels Minimierung der approximativen 0-Norm aller quadratischen Basispolynome auf die definierenden Polynome der Gelenke schließen. Damit sind Charakterisierungen der Gelenke bzgl. Art, Position und

n	Typ	$\dim(F_2)$	$\|f\|_{0,\varepsilon}$ $\min_{f\in F_2}$	$\max_{f\in F_2}$	0-Norm Min. t[sec]	Polynomdivision t[sec]
2	rr	2×15	3	7	0.058876	0.046151
2	rp	2×15	2	5	0.008268	0.046028
3	rrr	3×28	3	11	1.300929	0.164098
3	rrp	3×28	3	13	2.783481	0.163914
3	rpr	3×28	2	11	1.339117	0.164931
3	rpp	4×28	2	9	0.024872	0.177096
4	rrrr	4×45	3	15	11.968550	0.434601
4	rrrp	4×45	3	17	24.133943	0.442575
4	rrpr	4×45	3	19	41.956811	0.442551
4	rprr	4×45	2	15	12.166138	0.456452
4	rrpp	5×45	3	19	34.450221	0.471422
4	rprp	4×45	2	17	24.409154	0.456259
4	rrpp	5×45	2	15	10.106147	0.480430
4	rppp	7×45	2	17	18.613139	0.535170

Tabelle 5.1.: Laufzeitvergleich der Gelenkerkennung durch Minimierung der approximativen 0-Norm des quadratischen Anteils einer approximativen H-Basis bzw. Polynomdivision mit Rest. Zugrunde lagen Bewegungsprofile einer planaren kinematischen Kette mit n Gelenken, $N = 200$ Datenpunkten, Gelenkabständen $\ell_j = 2^{j-1}$ und gleichmäßig verteilten Drehwinkeln und Schublängen.

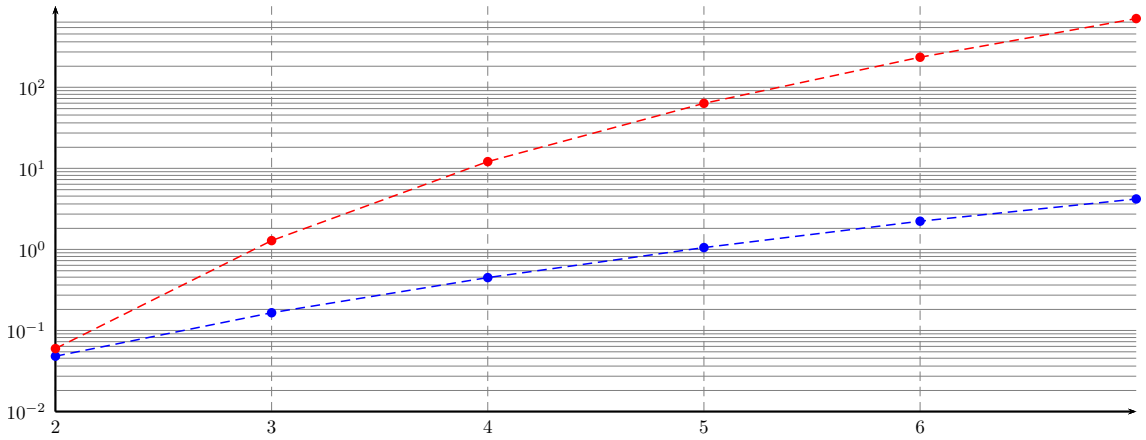
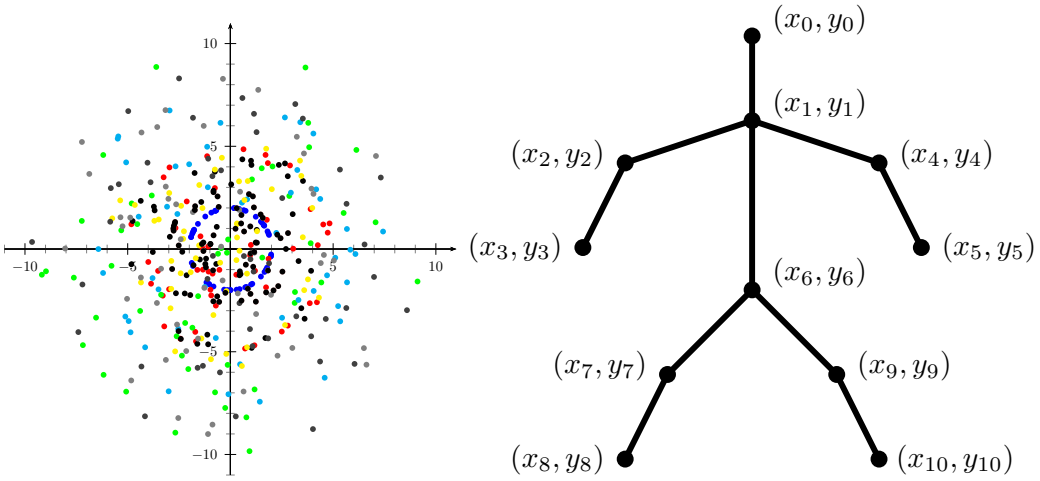


Abbildung 5.4.: Laufzeit der Gelenkerkennung mittels Minimierung der approximativen 0-Norm (rot) bzw. Polynomdivision (blau) in logarithmischer Skalierung. Die Abszissen beschreiben dabei die Anzahl der Gelenke, wobei ausschließlich Drehgelenke verwendet wurden.



(a) Bewegungsprofil **(b)** Zugehörige Geometrie

Abbildung 5.5.: Humanoides Modell als Anwendungsbeispiel der Gelenkerkennung: Aus einem Bewegungsprofil wird die Struktur der kinematischen Kette rekonstruiert.

Länge möglich. Zudem stellen die minimierten Basispolynome nach wie vor eine approximative H-Basis des Verschwindungsideals dar. Der benötigte Rechenaufwand steigt allerdings nichtpolynomiell mit der Anzahl der Gelenke, man vergleiche dazu Satz 4.64.

Verwendet man zur Charakterisierung der Gelenke die Polynomdivision mit Rest, so reduziert sich der Rechenaufwand erheblich. Dafür sind allerdings stärkere Voraussetzungen, wie eine Liste von Prototypen der Polynome $r_j^{(n)}$ und $p_j^{(n)}$, notwendig und man hat keine Information über die übrigen Polynome in der approximativen H-Basis des Verschwindungsideals. Betrachten wir allgemeine, also auch *nichtlineare* kinematische Ketten, so wächst die Anzahl der zu konstruierenden Prototypen nichtpolynomiell mit der Anzahl der Gelenke, was ebenfalls zu einem nichtpolynomiellen Rechenaufwand führt.

Abschließend zeigt Abbildung 5.5 die Anwendung der hier gewonnenen Resultate auf eine spezielle nichtlineare kinematische Kette: ein stark vereinfachtes humanoides Modell. Anhand des dargestellten Bewegungsprofils können sowohl die Zusammenhänge der einzelnen Glieder als auch deren Längen rekonstruiert werden. Im Rahmen dieser Arbeit wurden auch Datensätze *echter menschlicher Bewegungen*

aus einem *Motion Capture System* untersucht, die jedoch gezeigt haben, dass dieses Modell selbst für einfache Bewegungen zu stark idealisiert ist. Dabei stellt insbesondere die Bestimmung der Gelenkmittelpunkte aus gemessenen Marker-Daten ein wesentliches Problem dar. Für Details dazu und weitere Informationen zum Thema *Motion Capturing* sei auf [OBBH00] oder [Lau11] verwiesen.

5.2. Kinematische Ketten im Raum

In diesem Abschnitt werden lineare kinematische Ketten im Raum untersucht. Dabei soll der konzeptionelle Aufbau unseres Modells einer planaren kinematischen Kette aus Abschnitt 5.1 beibehalten werden, d. h. die Gelenke an den Positionen $(x_{j-1}, y_{j-1}, z_{j-1}) \in \mathbb{R}^3$ und $(x_j, y_j, z_j) \in \mathbb{R}^3$, $j = 1, \dots, n$, sind durch Glieder der Länge $\ell_j > 0$ miteinander verknüpft. Entsprechend lässt sich die Gelenkbeziehung (5.1) aus dem planaren Fall übertragen, indem wir von Polarkoordinaten zu *Kugelkoordinaten* übergehen:

$$\begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \\ z_{j-1} \end{pmatrix} + \ell_j \cdot \begin{pmatrix} \cos(\alpha_j)\cos(\beta_j) \\ \cos(\alpha_j)\sin(\beta_j) \\ \sin(\alpha_j) \end{pmatrix}, \quad \alpha_j, \beta_j \in [0, 2\pi],\ \ell_j > 0. \tag{5.8}$$

Fordern wir analog zum planaren Fall, dass sich in einem Gelenk entweder nur der Abstand ℓ_j oder die Drehwinkel α_j und β_j verändern, so erhalten wir Schub- bzw. Kugelgelenke im Raum. Die charakteristische Gelenkbedingung eines Kugelgelenks an der Stelle $j \in \{1, \dots, n\}$ ergibt sich entsprechend durch

$$r_j(x_{j-1}, y_{j-1}, z_{j-1}, x_j, y_j, z_j) := (x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 + (z_j - z_{j-1})^2 - \ell_j^2.$$

Wir unterscheiden dabei in der Bezeichnung nicht zwischen planaren Drehgelenken und Kugelgelenken im Raum. Analog zum planaren Fall erfolgt auch hier die Einbettung in den Polynomring $\mathbb{R}[x_1, y_1, z_1, \dots, x_n, y_n, z_n] \simeq \Pi_{3n}$ durch

$$r_j^{(n)}(x_1, y_1, z_1, \dots, x_n, y_n, z_n) := r_j(x_{j-1}, y_{j-1}, z_{j-1}, x_j, y_j, z_j), \quad j = 1, \dots, n.$$

Da sich auch für Schubgelenke im Raum die Orientierung nicht verändert, gilt $\alpha_j =$

$\alpha_{j-1},\ \beta_j = \beta_{j-1}$. Mit dieser Beziehung folgt aus der Gelenkbedingung (5.8) die Gleichung

$$\begin{pmatrix} x_j - x_{j-1} \\ y_j - y_{j-1} \\ z_j - z_{j-1} \end{pmatrix} = \ell_j \begin{pmatrix} \sin(\alpha_j)\cos(\beta_j) \\ \sin(\alpha_j)\sin(\beta_j) \\ \cos(\alpha_j) \end{pmatrix} = \frac{\ell_j}{\ell_{j-1}} \begin{pmatrix} x_{j-1} - x_{j-2} \\ y_{j-1} - y_{j-2} \\ z_{j-1} - z_{j-2} \end{pmatrix}.$$

Durch Auflösen nach $\frac{\ell_j}{\ell_{j-1}}$ ergibt sich

$$\frac{x_j - x_{j-1}}{x_{j-1} - x_j} = \frac{y_j - y_{j-1}}{y_{j-1} - y_j} = \frac{z_j - z_{j-1}}{z_{j-1} - z_j},$$

bzw.

$$(x_j - x_{j-1})(y_{j-1} - y_{j-2}) - (x_{j-1} - x_{j-2})(y_j - y_{j-1}) = 0, \tag{5.9a}$$
$$(y_j - y_{j-1})(z_{j-1} - z_{j-2}) - (y_{j-1} - y_{j-2})(z_j - z_{j-1}) = 0, \tag{5.9b}$$
$$(z_j - z_{j-1})(x_{j-1} - x_{j-2}) - (z_{j-1} - z_{j-2})(x_j - x_{j-1}) = 0. \tag{5.9c}$$

Damit an Position $j > 1$ in der kinematischen Kette ein Schubgelenk vorliegen kann, müssen alle Tripel aus Gelenkpositionen $(x_{j-2}, y_{j-2}, z_{j-2}, x_{j-1}, y_{j-1}, z_{j-1}, x_j, y_j, z_j) \in \mathbb{R}^9$ des Bewegungsprofils Nullstellen von (5.9a) - (5.9c) sein. Projiziert man diese Bedingungen auf die von den beteiligten Koordinatenachsen aufgespannte Ebene, so erhält man die aus dem planaren Fall bekannte Gelenkbedingung p_j.

Zur Gelenkerkennung von Kugelgelenken aus einer approximativen H-Basis können wir analog zum planaren Fall vorgehen. Wir konstruieren ein Polynom $\widetilde{r}_j^{(n)}$ aus $r_j^{(n)}$ indem wir $\ell_j = 0$ setzen und führen eine Polynomdivision mit Rest durch. Der Divisionsrest liefert dann die tatsächliche Länge ℓ_j. Für Schubgelenke verändert sich die Gelenkerkennung beim Übergang von der Ebene zum Raum: An Stelle *eines* charakterisierenden Polynoms müssen nun *drei Bedingungen* überprüft werden. Für das approximative Ideal-Membership-Problem bedeutet dies, dass dazu auch *drei Polynomdivisionen* durchgeführt werden müssen. Hier ist die Gelenkerkennung durch Minimierung der 0-Norm natürlich im Vorteil, da alle drei charakterisierenden Polynome direkt aus der minimierten approximativen H-Basis hervorgehen.

Einen ausführlichen Vergleich der benötigten Rechenzeit wie im planaren Fall wer-

den wir an dieser Stelle jedoch nicht durchführen, da im Raum auch Gelenktypen existieren, die nicht minimal bzgl. der 0-Norm sind. Damit ist das Verfahren der Gelenkerkennung durch Minimierung der approximativen 0-Norm aller quadratischen Basispolynome nicht für alle kinematischen Ketten im Raum anwendbar. Im folgenden Abschnitt werden wir einen solchen Gelenktyp untersuchen.

Die Abschätzungen für die Veränderung der Toleranz ε bei der Gelenkerkennung übertragen sich direkt aus dem planaren Fall, sofern sichergestellt ist, dass die kinematische Kette im Raum nur aus Kugel- und Schubgelenken besteht. Auch dieser Punkt gilt nicht mehr, wenn der im nächsten Abschnitt vorgestellte Gelenktyp beteiligt ist.

5.2.1. Drehgelenke mit fester Drehebene

Da die Gelenkbedingung (5.8) durch den zusätzlichen Drehwinkel β_j im Vergleich zu (5.1) einen Freiheitsgrad gewonnen hat, ist für kinematische Ketten im Raum ein neuer Gelenktyp möglich. Man kann durch eine geeignete Forderung an die Beziehung von α_j und β_j erreichen, dass alle Gelenkpositionen eines Kugelgelenks in einer Ebene liegen. Wir sprechen in diesem Fall von einem *Drehgelenk mit fester Drehebene* – häufig wird auch die Bezeichnung *Scharniergelenk* verwendet. Zu zwei linear unabhängigen Spannvektoren $s_1 \neq 0 \neq s_2 \in \mathbb{R}^3$ modifiziert sich die Gelenkbeziehung (5.8) zu

$$\begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} = \begin{pmatrix} x_{j-1} \\ y_{j-1} \\ z_{j-1} \end{pmatrix} + \ell_j \cdot \left(\cos(\alpha_j) \cdot \frac{s_1}{\|s_1\|_2} + \sin(\alpha_j) \cdot \frac{s_2}{\|s_2\|_2} \right), \quad \alpha_j \in [0, 2\pi],\ \ell_j > 0.$$

Für ein festes ℓ_j erhält man damit Gelenkpositionen (x_j, y_j, z_j), die sowohl in der von s_1 und s_2 aufgespannten Ebene, als auch auf dem Rand einer Kugel mit Radius ℓ_j um den Punkt $(x_{j-1}, y_{j-1}, z_{j-1})$ liegen. Die Menge aller dafür zulässigen Punkte ist also der Schnitt einer Kugel mit einer Ebene, siehe Abbildung 5.6. Die Varietät dieser Kugel entspricht $\mathfrak{V}(r_j^{(n)})$ und die Varietät der Ebene lässt sich durch $\mathfrak{V}(\eta_x x_j + \eta_y y_j + \eta_z z_j)$ beschreiben, wobei sich der *Normalenvektor* $\eta := (\eta_x, \eta_y, \eta_z)^T$ durch die Orthogonalitätsbeziehungen $(s_1, \eta) = 0 = (s_2, \eta)$ und $\|\eta\|_2 = 1$ ergibt.

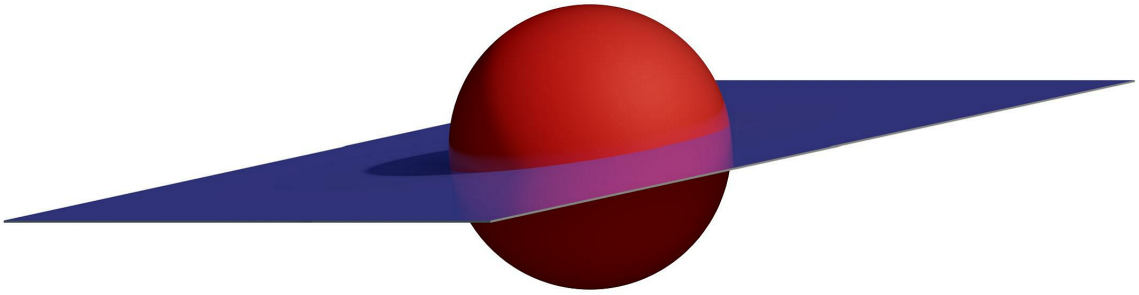


Abbildung 5.6.: Darstellung der Gelenkpositionen eines Drehgelenks mit fester Drehebene als Varietäten: Schnitt einer Kugel und einer Ebene

Die Erkennung der Kugelgleichung aus einer approximativen H-Basis kann, wie im letzten Abschnitt beschrieben, durch Polynomdivision mit Rest realisiert werden. Im Folgenden werden wir nun der Frage nachgehen, ob es auch für diesen Gelenktyp eine Erkennungsmethode gibt, die die H-Basis erhält.

Betrachten wir das Bewegungsprofil $\Xi \subset \mathbb{R}^3$ einer kinematischen Kette, die nur aus einem Drehgelenk mit fester Drehebene besteht (d. h. $n = 1$), so enthält die H-Basis F des approximativen Ideals $\mathfrak{I}_{p,\varepsilon}(\Xi)$ ein lineares und ein quadratisches Polynom zusätzlich zu den Polynomen höheren Grades, die sich aus der Endlichkeit von Ξ ergeben. Dabei sei vorausgesetzt, dass Ξ aus hinreichend vielen Punkten besteht und $\varepsilon > 0$ passend gewählt ist. Das lineare Basispolynom entspricht der Ebenengleichung $\eta_x x_1 + \eta_y y_1 + \eta_z z_1 = 0$ und liefert direkt den Normalenvektor der Ebene.

Das quadratische Polynom lässt sich jedoch nicht direkt einem Kugelgelenk zuordnen. Wir erhalten stattdessen ein Polynom der Form

$$f(x_1, y_1, z_1) = f_0 + f_1 z_1^2 + f_2 y_1 z_1 + f_3 y_1^2 + f_4 x_1 z_1 + f_5 x_1 y_1 + f_6 x_1^2, \quad f_k \in \mathbb{R}. \tag{5.10}$$

Da dieses Polynom einer H-Basis entnommen wurde, muss

$$\Lambda(f) \in \mathcal{W}^0_{2,3}(\eta_x x_1 + \eta_y y_1 + \eta_z z_1) \tag{5.11}$$

gelten. Dies ist im Allgemeinen für das charakterisierende Polynom $r_1^{(1)}$ eines Kugelgelenks nicht erfüllt. Ferner enthält die approximative H-Basis nur dieses eine quadratische Polynom, sodass auch eine Minimierung der 0-Norm aller quadratischen

Basispolynome analog zum planaren Fall nicht zielführend ist. Vielmehr müssen wir zur Rekonstruktion der Kugelgleichung $r_1^{(1)}$ den linearen Raum

$$\begin{aligned}\operatorname{span}\{&\eta_x x_1^2 + \eta_y x_1 y_1 + \eta_z x_1 z_1,\ \eta_x x_1 y_1 + \eta_y y_1^2 + \eta_z y_1 z_1,\ \eta_x x_1 z_1 + \eta_y y_1 z_1 + \eta_z z_1^2,\\ &x_1^2 + y_1^2 + z_1^2 - \ell_1^2\} \ni r_1^{(1)}\end{aligned} \tag{5.12}$$

betrachten. Leider hilft auch in diesem Fall die Minimierung der 0-Norm nicht weiter, da der Raum (5.12) eine Basis besitzen kann, deren Elemente alle in der 0-Norm kleiner sind als $\|r_1^{(1)}\|_0 = 4$. Wir zeigen dies an folgendem Beispiel:

Beispiel 5.6. *Sei $\eta = (1,1,1)^T/\sqrt{3}$, dann bilden die Zeilen der Matrix*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ -\ell_1^2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

eine Basis des linearen Raums aus (5.12). Das Gleiche gilt aber auch für die Zeilen der Matrix

$$\begin{bmatrix} \ell_1^2/2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -\ell_1^2/2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \ell_1^2/2 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix},$$

die jeweils nur drei von Null verschiedene Einträge haben und somit eine bzgl. der 0-Norm kleinere Basis darstellen.

Die Konstruktion des Raums (5.12) zeigt, dass der konstante Term von $r_1^{(1)}(x_1, y_1, z_1)$, also der Radius der Kugel, ausschließlich von $f(x_1, y_1, z_1)$ beeinflusst wird. Allerdings sind beide Polynome unterschiedlich skaliert, sodass wir nur den Zusammenhang $\ell_1^2 = c \cdot f_0$ für ein $c \in \mathbb{R}$ feststellen können. Um den Faktor c zu bestimmen, müssen wir die Normierung von f „rückgängig machen“, d. h. wir zerlegen $\Lambda(r_1^{(1)})$ bzgl. der linearen Räume $\mathcal{V}_{2,3}^0(\eta_x x_1 + \eta_y y_1 + \eta_z z_1)$ und $\mathcal{W}_{2,3}^0(\eta_x x_1 + \eta_y y_1 + \eta_z z_1)$. Dies führt zu einer unnormierten Version der Leitform von $f(x_1, y_1, z_1)$ und liefert so die Skalierung, in der wir ℓ_1^2 aus dem Koeffizienten f_0 des konstanten Terms ablesen können.

Sind alle Komponenten des Normalenvektors $\eta = (\eta_x, \eta_y, \eta_z)^T$ von Null verschieden, so können wir eine Basis von $\mathcal{V}^0_{2,3}(\eta_x x_1 + \eta_y y_1 + \eta_z z_1)$ und $\mathcal{W}^0_{2,3}(\eta_x x_1 + \eta_y y_1 + \eta_z z_1)$ angeben durch die Spalten von

$$V = \begin{bmatrix} 0 & 0 & \eta_z \\ 0 & \eta_z & \eta_y \\ 0 & \eta_y & 0 \\ \eta_z & 0 & \eta_x \\ \eta_y & \eta_x & 0 \\ \eta_x & 0 & 0 \end{bmatrix}, \quad \text{bzw.} \quad W = \begin{bmatrix} -\frac{\eta_y}{\eta_z} & -\frac{\eta_x}{\eta_z} & 0 \\ 1 & 0 & 0 \\ -\frac{\eta_z}{\eta_y} & 0 & -\frac{\eta_x}{\eta_y} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{\eta_z}{\eta_x} & -\frac{\eta_y}{\eta_x} \end{bmatrix}.$$

Dabei sind analog zur Konstruktion der Matrizen $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$ nur die relevanten Koeffizienten der Leitformen – in diesem Fall der quadratische Anteil – angegeben. Für Normalenvektoren $\eta \neq 0$ mit verschwindenden Komponenten ist die Matrix W entsprechend anzupassen.

Bevor nun die Zerlegung der Leitform $\Lambda(r_1^{(1)})$ bzgl. V und W durchgeführt wird, sei noch eine Eigenschaft der Koeffizienten des Polynoms f aus (5.10) bemerkt: Da für die Leitform von f per definitionem die Beziehung (5.11) gilt, lassen sich Koeffizienten $k = [k_1, k_2, k_3]^T \in \mathbb{R}^3$ mit $W \cdot k = [f_1, f_2, f_3, f_4, f_5, f_6]^T$ bestimmen. Dies führt zu dem linearen Gleichungssystem

$$\left|\begin{array}{rrrrcl} \eta_y \cdot k_1 & +\eta_x \cdot k_2 & & +\eta_z \cdot f_1 & = & 0 \\ k_1 & & & -f_2 & = & 0 \\ \eta_z \cdot k_1 & & +\eta_x \cdot k_3 & +\eta_y \cdot f_3 & = & 0 \\ & k_2 & & -f_4 & = & 0 \\ & & k_3 & -f_5 & = & 0 \\ & \eta_z \cdot k_2 & +\eta_y \cdot k_3 & +\eta_x \cdot f_6 & = & 0 \end{array}\right|.$$

Damit ist einerseits $[k_1, k_2, k_3] = [f_2, f_4, f_5]$ und aus den übrigen Gleichungen folgt

$$\begin{bmatrix} f_1 & f_2 & f_4 \\ f_2 & f_3 & f_5 \\ f_4 & f_5 & f_6 \end{bmatrix} \cdot \begin{bmatrix} \eta_z \\ \eta_y \\ \eta_x \end{bmatrix} = 0. \tag{5.13}$$

Die Gleichung (5.13) liefert nun einen direkten Zusammenhang zwischen den Koeffi-

zienten von f und dem Normalenvektor der Drehebene: Hat die Matrix den Rang 2, so erhält man den Normalenvektor n aus dem *Nullraum* der angegebenen Matrix.

Widmen wir uns nun der Darstellung des Polynoms $\Lambda(r_1^{(1)})$ durch V und W und betrachten dazu das lineare Gleichungssystem $[V, W] \cdot k = [1, 0, 1, 0, 0, 1]^T$, $k \in \mathbb{R}^6$, wobei die rechte Seite des Gleichungssystems den Koeffizienten des quadratischen Anteils von $\Lambda(r_1^{(1)})$ entspricht. Durch Multiplikation mit η_z, η_y und η_x lässt sich dieses Gleichungssystem schreiben als

$$\left|\begin{array}{rrrcll} k_3 \cdot \eta_z^2 & -k_4 \cdot \eta_y & -k_5 \cdot \eta_x & = & \eta_z & (*) \\ k_2 \cdot \eta_z & +k_3 \cdot \eta_y & & = & -k_4 & (\star) \\ -k_4 \cdot \eta_z & +k_2 \cdot \eta_y^2 & -k_6 \cdot \eta_x & = & \eta_y & (*) \\ k_1 \cdot \eta_z & & +k_2 \cdot \eta_x & = & -k_5 & (\star) \\ & k_1 \cdot \eta_y & +k_2 \cdot \eta_x & = & -k_6 & (\star) \\ -k_5 \cdot \eta_z & -k_6 \cdot \eta_y & +k_1 \cdot \eta_x^2 & = & \eta_z & (*) \end{array}\right| .$$

Setzen wir die Gleichungen $(\star)$ in die Gleichungen $(*)$ ein, so erhalten wir

$$\begin{bmatrix} -\eta_z\eta_x & -\eta_z\eta_y & \eta_z^2 - \eta_y^2 - \eta_x^2 \\ -\eta_y\eta_x & -\eta_z^2 + \eta_y^2 - \eta_x^2 & -\eta_z\eta_y \\ -\eta_z^2 - \eta_y^2 + \eta_x^2 & -\eta_y\eta_x & -\eta_z\eta_x \end{bmatrix} \cdot \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} \eta_z \\ \eta_y \\ \eta_x \end{bmatrix} . \tag{5.14}$$

Das lineare Gleichungssystem (5.14) kann nun symbolisch gelöst werden und liefert die Zerlegung $\Lambda(r_1^{(1)}) = r_V + r_W$ durch

$$r_V = (V \cdot [k_1, k_2, k_3]^T)^T = \frac{\begin{bmatrix} \eta_x^6 + \eta_x^4\eta_y^2 + \eta_x^4\eta_z^2 + \eta_x^2\eta_y^2\eta_z^2 \\ \eta_x^5\eta_y + \eta_x\eta_y^5 + 2(\eta_x^2\eta_y^3 + \eta_x^3\eta_y\eta_z^2 + \eta_x\eta_y^3\eta_z^2) \\ \eta_y^6 + \eta_x^2\eta_y^4 + \eta_y^4\eta_z^2 + \eta_x^2\eta_y^2\eta_z^2 \\ \eta_x^5\eta_z + \eta_x\eta_z^5 + 2(\eta_x^2\eta_z^3 + \eta_x^3\eta_y^2\eta_z + \eta_x\eta_y^2\eta_z^3) \\ \eta_y^5\eta_z + \eta_y\eta_z^5 + 2(\eta_y^2\eta_z^3 + \eta_x^2\eta_y^3\eta_z + \eta_x^2\eta_y\eta_z^3) \\ \eta_z^6 + \eta_x^2\eta_z^4 + \eta_y^2\eta_z^4 + \eta_x^2\eta_y^2\eta_z^2 \end{bmatrix}^T}{\frac{2}{3}\|\eta\|_2^6 + \frac{1}{3}(\eta_x^6 + \eta_y^6 + \eta_z^6) + \eta_x^2\eta_y^2\eta_z^2} \tag{5.15}$$

und $r_W = [1, 0, 1, 0, 0, 1] - r_V$. Weiter ist $\Lambda(f) \cdot c = r_W$ und damit folgt $\ell_1^2 = c \cdot f_0$.

Auch wenn durch (5.15) eine direkte Lösung des Problems möglich ist, sollte man

in numerischer Rechnung aufgrund der hohen Potenzen in (5.15) einen anderen Ansatz wählen. Die Zerlegung von $\Lambda(r_1^{(1)}) = r_V + r_W$ lässt sich ebenfalls mit Hilfe der Singulärwertzerlegung im Sinne der homogenen Polynomdivision aus Abschnitt 2.4.5 durchführen. Das folgende Beispiel verdeutlicht die Konstruktion in (5.13) und zeigt die Bestimmung des Faktors $c \in \mathbb{R}$ mit Hilfe der Zerlegung des charakterisierenden Gelenkpolynoms $\Lambda(r_1^{(1)}) = r_V + r_W$.

Beispiel 5.7. *Wir betrachten das Bewegungsprofil eines Drehgelenks mit fester Drehebene im Raum um den Ursprung. Dabei sei dessen Drehebene durch den Normalenvektor $\eta = (1,1,1)^T/\sqrt{3}$ gegeben sowie die Länge $\ell_1 = 3$. Bestimmen wir eine approximative H-Basis, so enthält diese die folgenden normierten Polynome:*

$$\begin{aligned} f_1(x_1,y_1,z_1) &= 0.57735x_1 + 0.57735y_1 + 0.57735z_1, \\ f_2(x_1,y_1,z_1) &= 0.08760x_1^2 - 0.04380x_1y_1 - 0.04380x_1z_1 + 0.08760y_1^2 - 0.04380y_1z_1 \\ &\quad + 0.08760z_1^2 - 0.98551. \end{aligned}$$

Das lineare Polynom f_1 liefert den Normalenvektor $\eta = (1,1,1)^T/\sqrt{3}$. Alternativ erhalten wir diesen Vektor auch aus dem quadratischen Polynom f_2 durch

$$\mathcal{N}\left(\begin{bmatrix} 0.08760 & -0.04380 & -0.043800 \\ -0.04380 & 0.08760 & -0.043800 \\ -0.04380 & -0.04380 & 0.087601 \end{bmatrix}\right) = \begin{bmatrix} 0.57735 \\ 0.57735 \\ 0.57735 \end{bmatrix}.$$

Um die Länge ℓ_1 zu bestimmen, zerlegen wir $\Lambda(r_1^{(1)})$ bzgl. η:

$$\begin{aligned} \Lambda(r_1^{(1)}) = [1,0,1,0,0,1]^T &= [0.2,0.4,0.2,0.4,0.4,0.2]^T + [0.8,-0.4,0.8,-0.4,-0.4,0.8]^T \\ &= r_V + r_W. \end{aligned}$$

Nun ist $0.8\Lambda(f_2) = 0.08760 r_W$ und damit folgt $-\ell_1^2 = \frac{0.8}{0.08760} \cdot (-0.98551) \approx -9$, was die gesuchte Länge $\ell_1 = 3$ liefert. Die H-Basis bleibt dabei erhalten, da wir nur ein Polynom umskaliert haben.

Man beachte, dass diese Überlegungen nur für kinematische Ketten mit *einem* Drehgelenk mit fester Drehebene gelten. Für längere Ketten entstehen mehrere lineare und quadratische Polynome in der approximativen H-Basis, sodass aus praktischen

Gesichtspunkten die Gelenkerkennung mittels Polynomdivision mit Rest verwendet werden sollte. Dabei muss jedoch berücksichtigt werden, dass sich nun auch lineare Polynome in der H-Basis befinden, was sich auf die Lösung des approximativen Ideal-Membership-Problems bzgl. $r_j^{(n)}$ auswirkt. In der H-Darstellung von $r_j^{(n)}$ treten auch die Produkte dieser linearen Polynome mit allen Termen vom Grad 1 auf, was zu $r_j^{(n)} \in \mathfrak{I}_{\infty,\tilde{\varepsilon}}(\Xi)$ mit

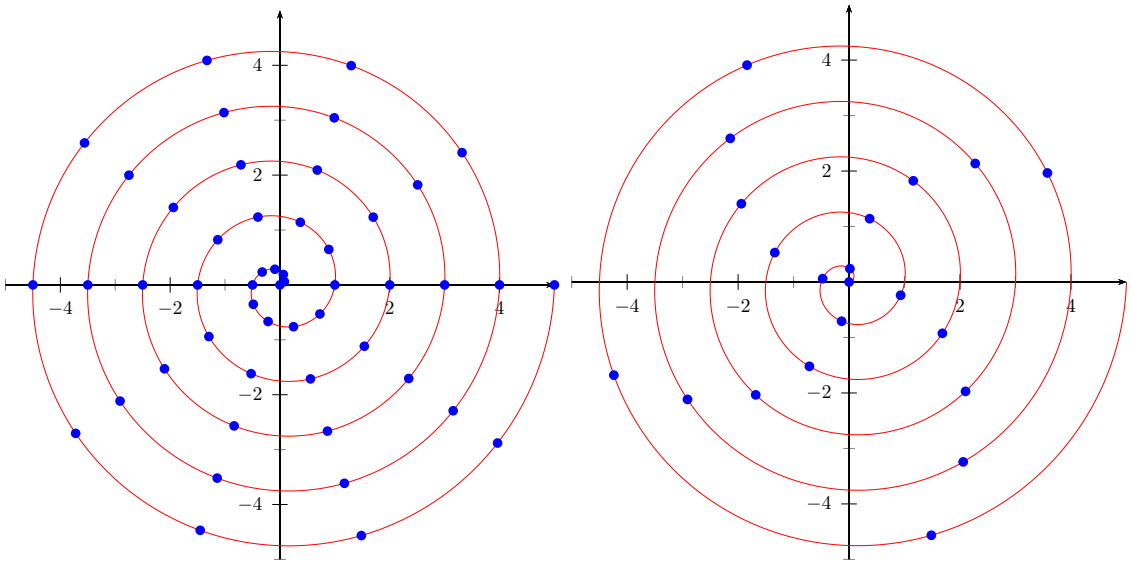
$$\tilde{\varepsilon} = \varepsilon \cdot \sqrt{2n \cdot \#F_1 + \#F_2}$$

führt. Für eine kinematische Kette, die aus n Drehgelenken mit fester Drehebene besteht, gilt, hinreichend viele Abtastungspunkte vorausgesetzt, $\#F_1 = \#F_2 = n$, also erhalten wir mit $\tilde{\varepsilon} = \varepsilon \cdot \sqrt{n} \cdot \sqrt{2n+1}$ eine Verschlechterung der Abschätzung um den Faktor $\sqrt{2n+1}$ im Vergleich zu Kugel- oder Schubgelenken.

Eine wichtige Voraussetzung für diese Überlegungen ist die Festlegung des Aufhängungspunktes an den Koordinatenursprung. Ist dies nicht gegeben, so haben die linearen Polynome einen von Null verschiedenen konstanten Anteil. Dies sorgt dafür, dass die Produkte der linearen Polynome mit den Termen vom Grad 1 nicht mehr orthogonal zu den quadratischen Polynomen der H-Basis sein müssen. Dadurch wird eine genauere Analyse – beispielsweise mit Hilfe der Singulärwerte der Faltungsmatrix – notwendig. Man vergleiche dazu die Resultate aus Abschnitt 4.2.3.

5.2.2. Weitere Gelenktypen

In diesem Abschnitt werden weitere Gelenktypen kinematischer Ketten angegeben, für die eine Erkennung aus einem Bewegungsprofil mit Hilfe von approximativen H-Basen möglich ist. Hierbei handelt es sich um Gelenke, die prinzipiell auch durch eine Kombination der bereits vorgestellten Gelenktypen realisiert werden können. Somit lässt sich die Gelenkerkennung auf diese Typen zurückführen. Eine direkte Erkennung von zusammengesetzten Gelenken hat jedoch den Vorteil, dass weniger Positionen benötigt werden und sich so die zu analysierende Datenmenge sowie der daraus resultierende Polynomgrad verringern. Dabei hat die hier dargestellte Auflistung keinen Anspruch auf Vollständigkeit, sondern soll vielmehr einen Ausblick auf weitere Einsatzmöglichkeiten von approximativen H-Basen geben. Für Informationen über andere Gelenkarten und deren Realisierung sei auf [HKSS97] verwiesen.



(a) $a = 1$, $\Delta t = 0.1$ **(b)** $a = 1$, $\Delta t = 0.24$

Abbildung 5.7.: Archimedische Spiralen in der Ebene, $0 \leq t \leq 5$.

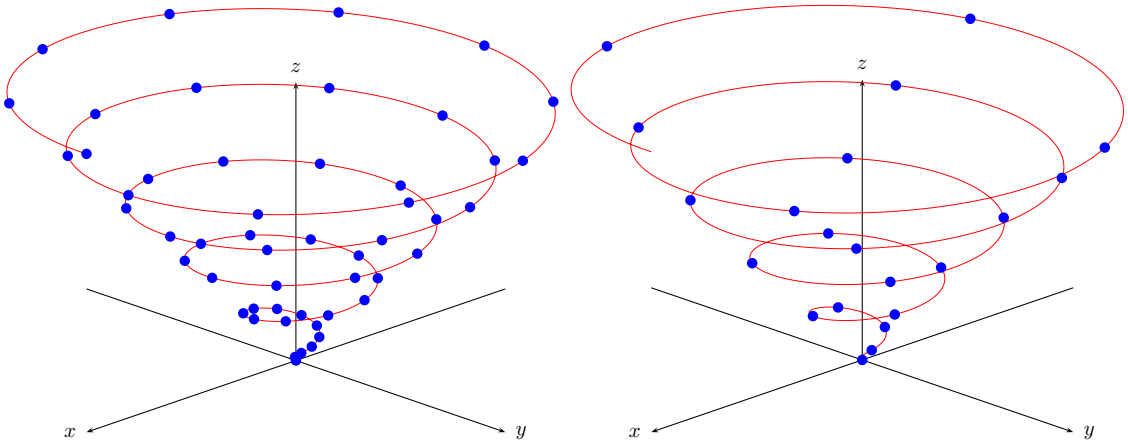
Spirale

Eine *archimedische* bzw. *arithmetische Spirale* in der Ebene ist definiert als die parametrische Kurve

$$\mathbb{R} \ni t \mapsto \begin{pmatrix} at \cdot \cos(2\pi t) \\ at \cdot \sin(2\pi t) \end{pmatrix} \in \mathbb{R}^2, \quad a > 0, \tag{5.16}$$

d. h. der Radius der Spirale ändert sich proportional zum Drehwinkel. Die Abbildungen 5.7a und 5.7b zeigen Abtastungen einer Spirale für $a = 1$ und äquidistante Drehwinkel mit $\Delta t = 0.1$ bzw. $\Delta t = 0.24$.

Wenden wir die Algorithmen aus Kapitel 4 auf die beiden Punktmengen an, so erhalten wir im ersten Fall ein Polynom vom Grad 5 im Geometrieanteil des approximativen Ideals. Dieses Polynom erklärt sich durch die Tatsache, dass die Punktmenge auch als Abtastung einer Vereinigung von fünf Geraden aufgefasst werden kann (vgl. Abbildung 5.7a), was auf der Idealseite dem Produkt von fünf linearen Polynomen entspricht. Im zweiten Fall besteht das approximative Ideal aus sieben Polynomen vom Grad 6. Berücksichtigen wir, dass diese Punktmenge aus 21 Punkten besteht, so



(a) $a = 1,\ b = 1,\ c = 0,\ \Delta t = 0.1$ **(b)** $a = 1,\ b = 1,\ c = 0,\ \Delta t = 0.24$

Abbildung 5.8.: Archimedische Spiralen im Raum, $0 \leq t \leq 5$.

liefert ein Nachschlagen in Tabelle 3.1, dass es sich hierbei um den Endlichkeitsanteil des Ideals handelt. Demnach wird in diesem Fall keine Geometrie erkannt.

Schließen wir den Sonderfall mehrerer kollinearer Punkte durch ungleichmäßig verteilte Abtastungen oder Ähnliches aus, so können wir keine geometrischen Informationen aus der Diskretisierung einer ebenen archimedischen Spirale gewinnen. Dies ändert sich durch den Übergang zu Spiralen im Raum, die analog zu (5.16) als parametrische Kurven

$$\mathbb{R} \ni t \mapsto \begin{pmatrix} at \cdot \cos(2\pi t) \\ at \cdot \sin(2\pi t) \\ bt + c \end{pmatrix} \in \mathbb{R}^3, \quad a > 0,\ b > 0,\ c > 0, \tag{5.17}$$

beschrieben werden können. Die Abbildungen 5.8a und 5.8b zeigen analog zum planaren Fall zwei Abtastungen. Bestimmen wir nun approximative H-Basen dieser beiden Punktmengen, so ist in beiden Fällen jeweils ein quadratisches Polynom der Form

$$x^2 + y^2 - \frac{a^2}{b^2} z^2 - \frac{a^2}{b^2} c^2, \tag{5.18}$$

enthalten, aus dem wir die Werte für $\frac{a}{b}$ und c bestimmen können. Natürlich liegen auch hier die Punkte aus Abbildung 5.8a in der Vereinigung von fünf Ursprungs-

geraden. Da das quadratische Polynom aus (5.18) jedoch einen kleineren Grad als das Produkt von fünf linearen Polynomen hat, wird die Gelenkerkennung in diesem Fall nicht von der Verteilung der Punkte beeinflusst. Geometrisch beschreibt dieses Polynom, bzw. dessen Varietät, die Mantelfläche eines Doppelkegels um die z-Achse mit einer Verschiebung um c.

Dieses Wissen können wir uns nun auch bei der Erkennung von archimedischen Spiralen in der Ebene zunutze machen, sofern die Drehwinkel äquidistant gewählt sind. Wir betrachten dazu erneut die Punktmenge $\Xi \subset \mathbb{R}^2$ aus Abbildung 5.7b und ergänzen die Punkte $\xi \in \Xi$ wie folgt:

$$\widetilde{\xi}^{(j)} = (\xi_1^{(j)}, \xi_2^{(j)}, j)^T.$$

Anschaulich bedeutet diese Ergänzung ein gleichmäßiges *Auseinanderziehen* der Spirale in Richtung der z-Koordinate, sodass die Punkte in der Mantelfläche eines Kegels liegen, vgl. Abbildung 5.8. Bestimmen wir nun eine approximative H-Basis zu den modifizierten Punkten, so enthält diese, bis auf Normierung, ein Polynom der Form

$$x^2 + y^2 - \frac{\widetilde{a}^2}{\widetilde{b}^2} z^2$$

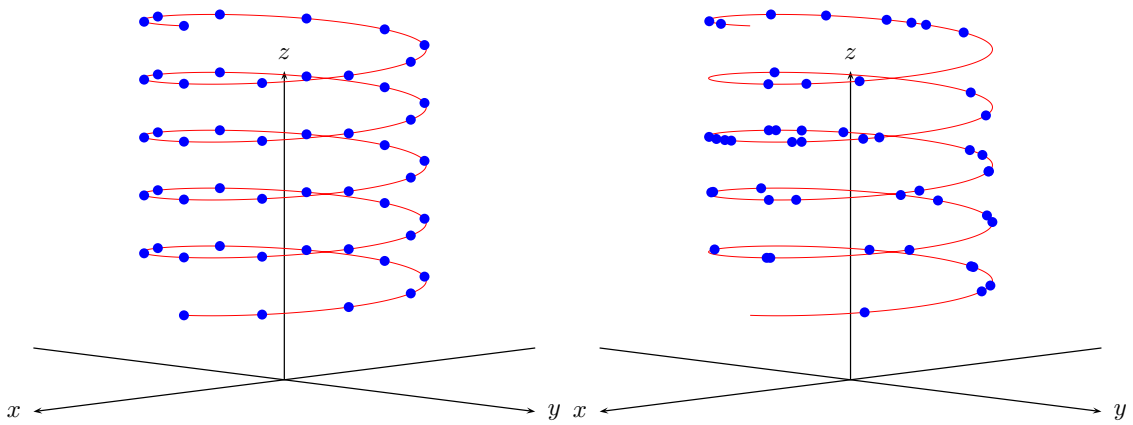
und es gilt $\widetilde{a}/\widetilde{b} = a\Delta t$. In unserem Beispiel aus Abbildung 5.7b erhalten wir so $x^2 + y^2 - 0.0576z^2$ was mit $a = 1$ zu $\Delta t = 0.24$ führt. Dieses Verfahren lässt sich insbesondere bei getakteten Messungen von Bewegungen mit gleichbleibender Winkelgeschwindigkeit anwenden.

Helix

Eine *Helix* oder *Schraubenlinie* ist durch folgende parametrische Kurve definiert:

$$\mathbb{R} \ni t \mapsto \begin{pmatrix} r \cdot \cos(2\pi t) \\ r \cdot \sin(2\pi t) \\ ht + c \end{pmatrix} \in \mathbb{R}^3, \quad r > 0,\ h > 0,\ c > 0. \tag{5.19}$$

Dabei bezeichnet man r als *Radius*, h als *Ganghöhe* und c als *Offset*. Im Vergleich zu (5.17) fällt auf, dass hier der Radius konstant und damit unabhängig vom Dreh-



(a) Gleichmäßig verteilte Drehwinkel mit $\Delta t = 0.1$

(b) Zufällig verteilte Drehwinkel

Abbildung 5.9.: Abtastung einer Helix mit $r = 2$, $h = 3$, $c = 4$ für $0 \leq t \leq 5$.

winkel ist. In den Abbildungen 5.9a und 5.9b sind verschiedene Abtastungen einer Helix dargestellt. Dabei wurden im ersten Fall gleichmäßig verteilte Drehwinkel und im zweiten Fall zufällig verteilte Drehwinkel verwendet. Bestimmen wir zu beiden Punktmengen je eine approximative H-Basis, so enthalten diese in beiden Fällen ein quadratisches Polynom der Form $x^2 + y^2 - r^2$, dessen Varietät einem Zylinder um die z-Achse mit Radius r entspricht. Daraus können wir nun auf den Radius der Helix schließen, nicht aber auf deren Ganghöhe und Offset. Um auch diese Parameter zu bestimmen, benötigt man Kenntnis über die Verteilung von t. Sind die Werte für t gegeben, so kann man diese in (5.19) als zusätzliche Komponente anfügen und erhält damit in der approximativen H-Basis ein lineares Polynom, aus dem sich, nach Normierung, die Werte für h und c ablesen lassen. Die Bemerkung über die Unabhängigkeit der Gelenkerkennung von der Verteilung der Punkte, die im Zusammenhang mit der Spirale im Raum gemacht wurde, gilt hier entsprechend.

6 Weitere Anwendungen und numerische Ergebnisse

Inhalt

6.1. Fehlerbehaftete Messdaten aus einfachen geometrischen Strukturen 166

6.2. H-Basen vs. Randbasen 170

6.3. Implizite Funktionen 173

6.4. Datensätze mit Ausreißern 176

In diesem Kapitel werden weitere Anwendungsmöglichkeiten von approximativen H-Basen aufgezeigt. Zudem werden die hier untersuchten bzw. neu entwickelten Verfahren mit anderen bereits bekannten Methoden verglichen. Dazu stehen folgende Referenzen zur Verfügung:

1. Der *Approximate Buchberger-Möller* Algorithmus (ABM) von Limbeck aus [Lim13] sowie
2. der *Low-degree Polynomial* Algorithmus (LPA), der von Fassino und Torrente in [FT13] beschrieben wird und
3. der *Almost Vanishing Ideal* Algorithmus (AVI) von Heldt, Kreuzer, Poulisse und Pokutta aus [HKPP09].

6.1. Fehlerbehaftete Messdaten aus einfachen geometrischen Strukturen

In [Lim13] wurden bereits erste Vergleiche zwischen den Ergebnissen des ABM Verfahrens und approximativen H-Basen durchgeführt. Dazu konstruierte man Testdatensätze, die einfache geometrische Zusammenhänge repräsentieren. Diese wurden dann durch Addition von zufälligen Werten gestört, sodass die definierenden Gleichungen der Ausgangsmenge nicht mehr erfüllt sind. Ein exaktes Verfahren zur Bestimmung des Verschwindungsideals dieser Punkte, wie etwa der bekannte Buchberger-Möller Algorithmus aus [MB82], kann somit den geometrischen Zusammenhang nicht mehr erkennen.

Wir betrachten nun die Daten aus [Lim13] erneut und vergleichen die dort erzielten Ergebnisse mit den hier vorgestellten Verfahren. Dazu wurden jeweils drei approximative H-Basen unter folgenden Vorgaben konstruiert:

1. $p = \infty$ mit einem heuristisch ermittelten, *guten Startwert*,
2. $p = \infty$ *ohne Startwert* sowie
3. $p = 2$.

Die Resultate des ABM Algorithmus wurden mit ApCoCoA berechnet und entsprechen im Wesentlichen den Werten aus [Lim13] bis auf die dort angegebene Genauigkeit. Zur Vergleichbarkeit der Ergebnisse wurden alle Koeffizientenvektoren bzgl. des graduiert-lexikographisch größten Terms normiert. Dies entspricht der Standardvorgabe der ABM Implementierung in ApCoCoA.

Man beachte, dass bei den folgenden Beispielen von Limbeck die Skalierung auf den d-dimensionalen Einheitswürfel vernachlässigt wurde. Da dies eine Voraussetzung für den AVI Algorithmus bzw. dessen Implementierung in ApCoCoA ist, können wir hier keinen Vergleich mit diesem Verfahren durchführen.

Beispiel 6.1. *Wir betrachten die durch das univariate quadratische Polynom*

$$f(x) = \frac{15}{2}x^2 - 6x + 8$$

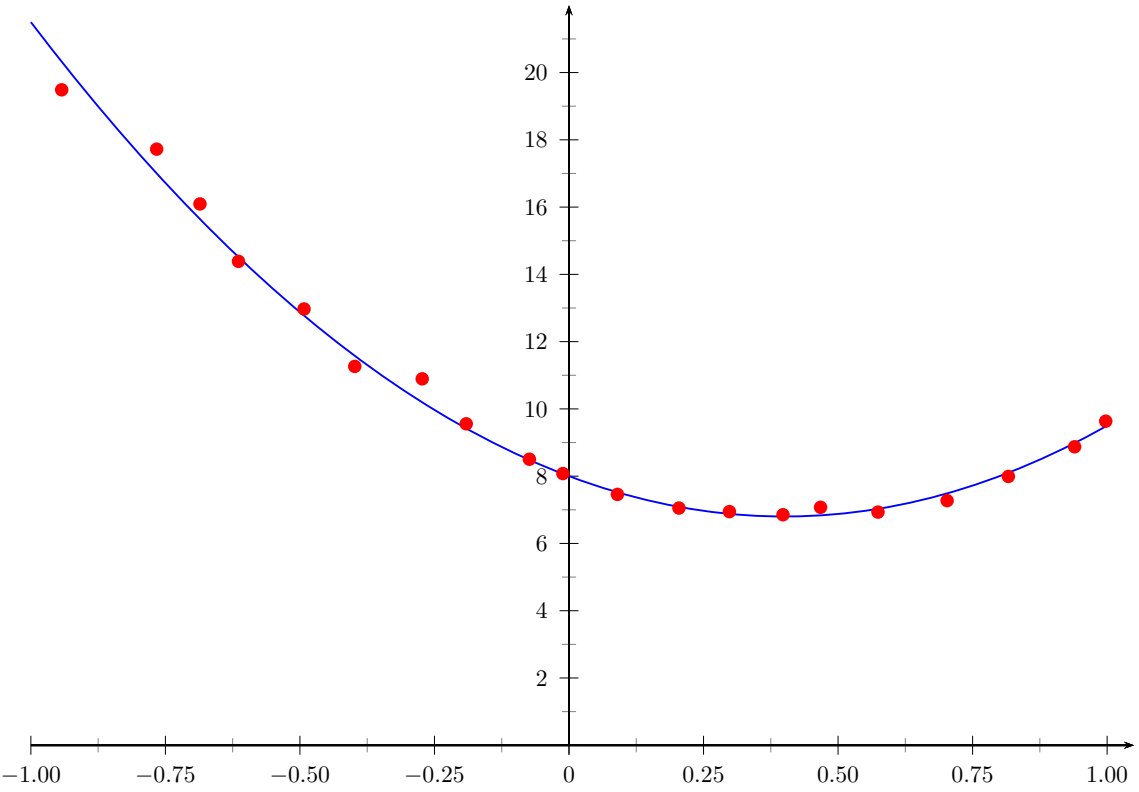


Abbildung 6.1.: Parabel aus Beispiel 6.1 und eine fehlerbehaftete Abtastung der Funktionswerte.

	1	x_2	x_1
exakt	1.0666667	-0.1333333	-0.8000000
H-Basis, $p = \infty$, guter Startwert	1.0495620	-0.1286615	-0.9513714
H-Basis, $p = \infty$, ohne Startwert	-231.6246174	42.3208063	189.4934899
H-Basis, $p = 2$	1.7472798	-0.2737205	-1.6777696
ABM	1.7472798	-0.2737205	-1.6777696
	x_2^2	x_1x_2	x_1^2
exakt	0.0000000	0.0000000	1.0000000
H-Basis, $p = \infty$, guter Startwert	-0.0000942	0.0158880	1.0000000
H-Basis, $p = \infty$, ohne Startwert	-2.0015148	-18.7664426	1.0000000
H-Basis, $p = 2$	0.0077453	0.0939056	1.0000000
ABM	0.0077453	0.0939056	1.0000000

Tabelle 6.1.: Koeffizientenvektoren des quadratischen Basispolynoms verschiedener approximativer Basen aus Beispiel 6.1.

gegebene Parabel auf dem Intervall $-1 \leq x \leq 1$. Durch Abtastung und anschließende Störung der Werte erhält man die in Abbildung 6.1 dargestellte Punktmenge. Die Zahlenwerte sind näherungsweise in [Lim13] angegeben. Wenden wir nun die verschiedenen Verfahren zur Berechnung einer approximativen H-Basis an, so erhalten wir die in Tabelle 6.1 angegebenen Koeffizientenvektoren.

An Beispiel 6.1 werden zwei wichtige Punkte deutlich:

- Die Qualität der Näherung an den exakten Koeffizientenvektor hängt entscheidend von der Wahl des Startwerts ab. Für einen guten Startwert erhalten wir das beste Ergebnis aller Verfahren. Legen wir jedoch keinen Startwert fest, so besteht die Gefahr, dass das Verfahren einen Datenausreißer auswählt und somit zwar richtige, aber unbrauchbare Resultate liefert. Auf diese Problematik werden wir in Abschnitt 6.4 noch genauer eingehen.
- Der Koeffizientenvektor aus dem ABM Verfahren stimmt exakt mit dem Resultat aus der approximativen H-Basis für $p = 2$ überein. Tatsächlich sind die Resultate bis auf Maschinengenauigkeit identisch.

Beispiel 6.2. *Ein weiteres Beispiel aus [Lim13] beschreibt eine Punktmenge, die durch Abtastung einer Ebene im Raum gegeben ist. Dabei ist die Ebene durch die Normalengleichung*

$$f(x_1, x_2, x_3) = x_1 - 3x_2 + \frac{5}{2}x_3 - 4 = 0$$

festgelegt. Auch diese abgetasteten Werte wurden gestört. Für näherungsweise Zahlenwerte sei erneut auf [Lim13] verwiesen. Durch Anwendung des ABM Verfahrens und der Methoden zur Bestimmung einer approximativen H-Basis erhalten wir die in Tabelle 6.2 angegebenen Koeffizientenvektoren.

Für diese Resultate gelten ebenfalls die Bemerkungen aus dem letzten Beispiel: Im Fall $p = \infty$ ohne Startwert erhält man das schlechteste Ergebnis, auch wenn der Koeffizientenvektor deutlich näher an den exakten Werten liegt. Für einen guten Startwert liefert das Verfahren eine recht gute Näherung der Normalengleichung. Im Fall $p = 2$ sind die H-Basis und die vom ABM Algorithmus bestimmte Randbasis wieder identisch.

	1	x_3	x_2	x_1
exakt	-4.000000	2.500000	-3.000000	1.000000
H-Basis, $p = \infty$, guter Startwert	-4.087483	2.521508	-2.966077	1.000000
H-Basis, $p = \infty$, ohne Startwert	-3.688516	2.175606	-2.506551	1.000000
H-Basis, $p = 2$	-4.025118	2.433679	-2.860425	1.000000
ABM	-4.025118	2.433679	-2.860425	1.000000

Tabelle 6.2.: Koeffizientenvektoren des linearen Basispolynoms verschiedener approximativer Basen aus Beispiel 6.2.

Beispiel 6.3. *Das letzte Beispiel, das von Limbeck zum Vergleich von approximativen Randbasen und H-Basen untersucht wurde, basiert auf einer Geraden im Raum, deren Parameterdarstellung durch*

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda \in \mathbb{R},$$

gegeben ist. Analog zu den letzten beiden Beispielen wurde zu dieser Geraden eine Punktmenge durch Abtastung und Störung der Daten generiert, die wieder in [Lim13] angegeben ist. Die Varietät einer Geraden führt zu einer H-Basis mit zwei linearen Polynomen, da die Gerade im geometrischen Sinne als Schnitt zweier Ebenen dargestellt werden kann. Daher reicht an dieser Stelle eine Normierung der Resultate aus den verschiedenen Verfahren nicht aus. Vielmehr müssen wir eine Vorüberlegung zum Aufbau der vom ABM Algorithmus berechneten Randbasen treffen:

- *Die berechneten Randbasen berücksichtigen eine Termordnung.*
- *Die Basispolynome der berechneten Randbasis haben einen minimalen Träger.*

Eine genauere Untersuchung der Unterschiede zwischen H-Basen und Randbasen folgt in Abschnitt 6.2. Um die zweite Voraussetzungen für approximative H-Basen herzustellen, kann beispielsweise die Methode zur Minimierung der approximativen 0-Norm aus Abschnitt 4.4 verwendet werden. So ergeben sich die in Tabelle 6.3 angegebenen Koeffizientenvektoren.

In diesem Beispiel liefert das ABM Verfahren bzw. die approximative H-Basis für $p = 2$ das beste Ergebnis. Dies lässt darauf schließen, dass sich die Störung sehr

	1	x_3	x_2	x_1
exakt	0.000000	-1.000000	1.000000	0.000000
	0.000000	-1.000000	0.000000	1.000000
H-Basis, $p = \infty$, guter Startwert	-0.052949	-0.995077	1.000000	0.000000
	-0.000570	-0.999386	0.000000	1.000000
H-Basis, $p = \infty$, ohne Startwert	0.089080	-1.002179	1.000000	0.000000
	0.042618	-1.001546	0.000000	1.000000
H-Basis, $p = 2$	0.012693	-1.000771	1.000000	0.000000
	0.004427	-1.000016	0.000000	1.000000
ABM	0.012693	-1.000771	1.000000	0.000000
	0.004427	-1.000016	0.000000	1.000000

Tabelle 6.3.: Koeffizientenvektoren der beiden linearen Basispolynome verschiedener approximativer Basen aus Beispiel 6.3. Für die H-Basen wurde zusätzlich eine Minimierung der approximativen 0-Norm durchgeführt.

gleichmäßig auf die Punkte verteilt. Die Tatsache, dass auch in diesem Beispiel die Koeffizientenvektoren von Randbasis und H-Basis für $p = 2$ übereinstimmen, zeigt, dass sich auch durch eine nachträgliche Minimierung *der approximativen* 0*-Norm dünn besetzte Basispolynome* gleicher Qualität *erzeugen lassen.*

6.2. H-Basen vs. Randbasen

Dieser Abschnitt beschreibt die Unterschiede zwischen approximativen H-Basen und approximativen Randbasen im praktischen Einsatz. Eine erste Gegenüberstellung wurde bereits von Limbeck in [Lim13] durchgeführt, die an dieser Stelle unter Berücksichtigung der hier vorgestellten Verfahren fortgesetzt wird. Limbeck hat im Wesentlichen die folgenden Punkte angegeben:

1. **Interpolation der Daten:** Der Algorithmus zur Bestimmung einer approximativen H-Basis generiert eine Folge von Polynommengen $F_k^0 \subset \Pi_{k,d} \setminus \Pi_{k-1,d}$ mit der Eigenschaft $F_k^0(\Xi_k) = 0$ für eine Menge von Punkten $\Xi_0 \subset \Xi_1 \subset \cdots \subset \Xi_k \subset \Xi$. Das heißt, die vorgegebenen Datenpunkte werden schrittweise interpoliert. Dies gilt nicht im ABM Verfahren, da die erzeugten Basispolynome im Allgemeinen keine gemeinsamen Nullstellen haben.

2. **Verwendete Auswertungsnorm:** Während der ABM Algorithmus den Auswertungsvektor eines Polynoms an allen vorgegebenen Stellen $\Xi \subset \mathbb{R}^d$ bzgl. der euklidischen Norm bewertet, wird im H-Basen Algorithmus zu diesem Zweck die Maximumsnorm verwendet. Dies führt zu Basen unterschiedlicher approximativer Ideale, nämlich $\mathfrak{I}_{2,\varepsilon}(\Xi)$ bzw. $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$.

3. **Besetztheit der Basispolynome:** Das ABM Verfahren zielt darauf ab, Basispolynome mit möglichst kleinem Träger zu generieren. Im Gegensatz dazu sind die Basispolynome, die für approximative H-Basen bestimmt werden, in der Regel voll besetzt.

4. **Abhängigkeit von einer Termordnung:** Eine Voraussetzung des ABM Algorithmus ist eine vorgegebene Termordnung. Approximative H-Basen sind dagegen von Termordnungen unabhängig.

Natürlich beziehen sich diese Aussagen nur auf die ursprüngliche Variante der Konstruktion approximativer H-Basen von Sauer, vgl. Algorithmus 4.12. In Abschnitt 4.1 wurde jedoch gezeigt, dass sich dieses Verfahren in den ersten beiden Punkten modifizieren lässt. Weiterhin kann die Methode zur Minimierung der approximativen 0-Norm aus Abschnitt 4.4 verwendet werden, um H-Basen mit dünn besetzten Basispolynomen zu erzeugen. Dies ermöglicht eine Reproduktion der Resultate des ABM Verfahrens für einfache Varietäten. Drei Beispiele solcher Varietäten wurden im letzten Abschnitt besprochen.

Die vom ABM Algorithmus vorausgesetzte Termordnung unterscheidet die Verfahren jedoch nach wie vor. Auch wenn in [Lim13] darauf hingewiesen wurde, dass von Termordnungen unabhängige Randbasen existieren und ein Verfahren zur Konvertierung von Randbasen angegeben wurde, bleibt die Frage nach der numerischen Stabilität dieses Vorgehens. Das folgende Beispiel zeigt eine kritische Situation aus dem Kontext der in Kapitel 5 untersuchten kinematischen Systeme.

Beispiel 6.4. *Wir betrachten folgende planare kinematische Kette: Ein Schubgelenk sei am Aufhängungspunkt, d. h. dem Ursprung, befestigt. Die Richtung sei dabei durch den festen, aber nicht bekannten Winkel $\alpha \in [0, 2\pi]$ gegeben, der von Gelenk und x_1-Achse eingeschlossen wird. Zudem sei die Länge durch $\ell \leq 1$ beschränkt. Ein*

Bewegungsprofil dieser Kette ist also von der Form

$$\Xi \subset \left\{ \ell \cdot \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} : 0 \leq \ell \leq 1 \right\}.$$

Je nach Termordnung erhalten wir dann folgende Polynome in der approximativen Randbasis:

$$f(x_1, x_2) = x_1 - \frac{\cos(\alpha)}{\sin(\alpha)} x_2, \quad \textit{bzw.} \quad f(x_1, x_2) = x_2 - \frac{\sin(\alpha)}{\cos(\alpha)} x_1. \tag{6.1}$$

Für $\alpha \to 0$ bzw. $\alpha \to \pi/2$ werden die Nenner in (6.1) sehr klein, was die Berechnung numerisch instabil macht. Die Berechnung einer approximativen H-Basis betrifft dies nicht, da wir ohne eine vorgegebene Termordnung das Basispolynom

$$f(x_1, x_2) = \frac{\sin(\alpha)}{\sqrt{2}} x_1 - \frac{\cos(\alpha)}{\sqrt{2}} x_2$$

erhalten. Die Implementierung des ABM Verfahrens in ApCoCoA liefert zwar gute Ergebnisse, dies ist aber darin begründet, dass ApCoCoA Polynome über $\mathbb{Q}$, bzw. über den im endlichen Speicher darstellbaren rationalen Zahlen, bestimmt und damit eine wesentlich höhere Rechengenauigkeit ermöglicht. Wählen wir beispielsweise $\alpha \approx 10^{-7}$ und eine Datengenauigkeit von 10^{-8}, so erhalten wir für $\varepsilon = 10^{-8}$ eine approximative Randbasis, die das Polynom

$$f(x_1, x_2) = x_1 - \frac{699609849313982676992}{232779278373101} x_2 - \frac{366340852349675}{44693621447635392} \tag{6.2}$$

enthält. Dabei übersteigt bereits der Zähler des Koeffizienten von x_2 die Darstellung einer Gleitkommazahl in doppelter Genauigkeit. Tatsächlich entspricht (6.2) nach Rundung und passender Normierung auch dem linearen Polynom in der approximativen H-Basis für $p = 2$. Es ist jedoch davon auszugehen, dass dieses Ergebnis in einer rein numerischen Umgebung deutlich stärker von Rundungsfehlern beeinflusst wird.

Ein wesentlicher Unterschied zwischen approximativen H-Basen und approximativen Randbasen wurde jedoch noch nicht genannt: die Orthogonalitätsbeziehung zwischen den Basispolynomen gleichen Grades bzw. den Leitformen $x^\alpha \Lambda(f)$ und $\Lambda(g)$,

$f, g \in F$, $|\alpha| = \deg(g) - \deg(f) \geq 0$. Diese Eigenschaft ist eine elementare Voraussetzung für die Eindeutigkeit des Divisionsalgorithmus, der in Abschnitt 2.4.5 beschrieben wurde. In Abschnitt 5.2 wurde im Zusammenhang mit Drehgelenken im Raum eine Anwendung vorgestellt, für die der Divisionsalgorithmus unbedingt notwendig ist, da ein explizites Ideal-Membership-Problem gelöst werden muss. Es existiert auch ein Divisionsalgorithmus für Randbasen, der von Kehrein und Kreuzer in [KK05] beschrieben wurde. Die numerische Stabilität dieses Verfahrens ist jedoch noch zu untersuchen, ebenso wie die Verwendung von approximativen Randbasen anstelle exakter Randbasen.

6.3. Implizite Funktionen

Im Folgenden untersuchen wir eine weitere Anwendung der Verfahren zur Bestimmung approximativer H-Basen: die numerische Bestimmung *impliziter Darstellungen* von parametrischen Kurven, die in vielen graphischen Anwendungen, wie z. B. dem CAGD, verwendet werden. Als wichtige Beispiele sind hier unter anderem *Bézierkurven* oder *Spline-Kurven* zu nennen. Auch die in Abschnitt 5.2.2 beschriebenen Kurven der archimedischen Spirale und der Helix gehören zu dieser Kategorie.

Interessiert man sich für eine *implizite Darstellung* einer parametrischen Kurve, d. h.

$$f(x_1, \ldots, x_d) = 0,$$

so ist in der Regel ein hoher symbolischer Rechenaufwand erforderlich. Tastet man hingegen die Kurve an einzelnen Punkten hinreichend genau ab und berechnet zu dieser Punktmenge eine approximative H-Basis, so enthält diese eine gute Näherung an die implizite Darstellung der Kurve. Das folgende Beispiel zeigt das Vorgehen anhand einer *Bézierkurve*:

Beispiel 6.5. *Wir betrachten eine Situation, die von Fassino und Torrente in [FT13] untersucht wurde. Dabei ist eine Bézierkurve durch die Parameterdarstellung*

$$x_1(t) = 4t(2t^5 - 3t^4 + 8t^2 + 6t + 3)/(t^6 - 3t^5 + 3t^4 + 3t^2 + 1), \tag{6.3}$$
$$x_2(t) = 6t(4t^4 + 9t^3 - 9t^2 - 9t + 5)/(t^6 - 3t^5 + 3t^4 + 3t^2 + 1) \tag{6.4}$$

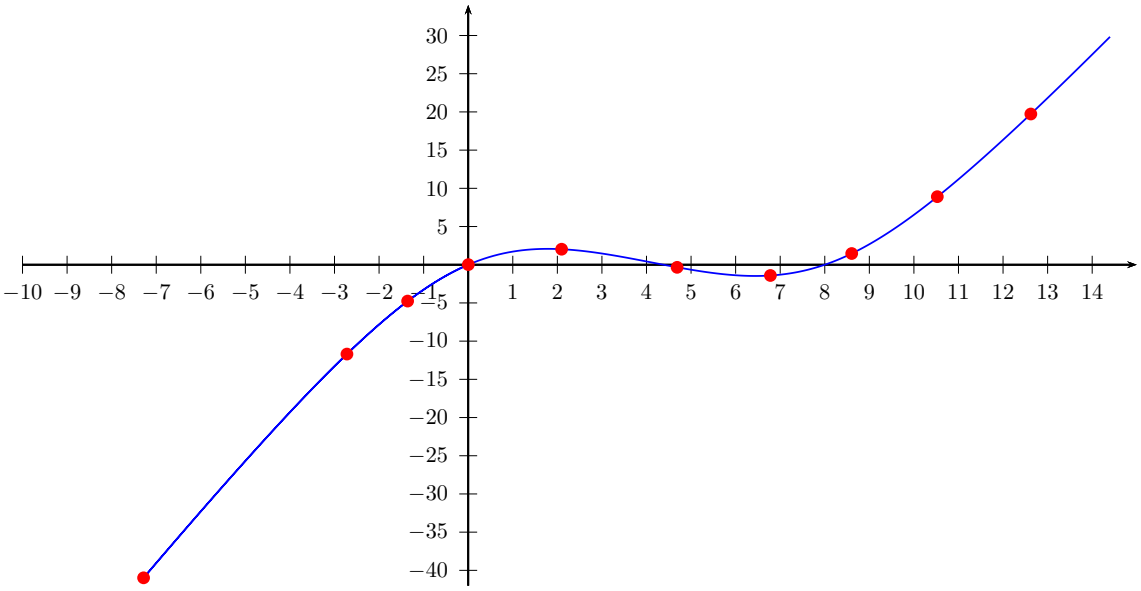


Abbildung 6.2.: Bézierkurve aus Beispiel 6.5 für $1 \leq t \leq 2$. Die Markierungen zeigen die abgetasteten und gerundeten Werte Ξ.

und die zugehörige exakte implizite Darstellung $g(x_1, x_2) = 0$ *mit*

$$g(x_1, x_2) = x_1^3 - \frac{2}{1269}x_1^2x_2 - \frac{28}{423}x_1x_2^2 + \frac{224}{34263}x_2^3 - \frac{15712}{1269}x_1^2 - \frac{56}{1269}x_1x_2 + \frac{848}{3807}x_2^2 + \frac{44480}{1269}x_1 - \frac{17792}{1269}x_2$$

angegeben. Durch Auswertung der Gleichungen (6.3) *und* (6.4) *an 10 zufälligen Stellen* $-1 < t < 2$ *erhält man die in [FT13] angegebene Punktmenge*

$$\begin{aligned}\Xi = \{&(0,0), (-1.3581, -4.7661), (2.0956, 2.0315), (4.6884, -0.3349),\\ &(-2.7205, -11.6848), (-7.2835, -40.9773), (6.7793, -1.4114),\\ &(8.6024, 1.4575), (10.52500, 8.8937), (12.6213, 19.7217)\} \subset \mathbb{R}^2.\end{aligned}$$

Diese Punkte sowie die Bézierkurve für $-1 < t < 2$ *sind in Abbildung 6.2 dargestellt. Alle Werte wurden dabei auf eine Genauigkeit von* 10^{-4} *gerundet, daher liegt es nahe, nach einer approximativen H-Basis von* $\mathfrak{I}_{p,\varepsilon}(\Xi)$ *mit* $\varepsilon = 10^{-4}$ *zu suchen. Wenden wir die in Abschnitt 4.1 besprochenen Verfahren auf die Menge* Ξ *an, so enthält die berechnete H-Basis die in Tabelle 6.4 angegebenen Koeffizientenvektoren*

	1	x_2	x_1	x_2^2	x_1x_2
exakt	0.00000	-14.0205	35.0512	0.22275	-0.04413
H-Basis, $p = \infty$, $\xi^{(0)} = (0,0)$	0.00000	-14.0204	35.0510	0.22275	-0.04414
H-Basis, $p = \infty$, ohne Startw.	0.00000	-14.0205	35.0516	0.22276	-0.04423
H-Basis, $p = 2$	-0.00032	-14.0203	35.0511	0.22267	-0.04383
[FT13], Example 5.3	-0.00033	-14.0206	35.0514	0.22269	-0.04396
[FT13], Example 5.3, Var. 2	0.00000	-14.0205	35.0513	0.22271	-0.04407
	x_1^2	x_2^3	$x_1x_2^2$	x_1^2y	x_1^3
exakt	-12.3814	0.00654	-0.06619	-0.00158	1.00000
H-Basis, $p = \infty$, $\xi^{(0)} = (0,0)$	-12.3814	0.00654	-0.06618	-0.00159	1.00000
H-Basis, $p = \infty$, ohne Startw.	-12.3815	0.00654	-0.06620	-0.00156	1.00000
H-Basis, $p = 2$	-12.3814	0.00653	-0.06617	-0.00162	1.00000
[FT13], Example 5.3	-12.3814	0.00654	-0.06618	-0.00159	1.00000
[FT13], Example 5.3, Var. 2	-12.3814	0.00653	-0.06619	-0.00158	1.00000

Tabelle 6.4.: Koeffizientenvektoren einer impliziten Darstellung der Bézierkurve aus Beispiel 6.5. Alle Werte wurden zur Vergleichbarkeit auf die in [FT13] angegebene Genauigkeit gerundet.

als Elemente kleinsten Grades.

Zunächst lässt sich feststellen, dass alle untersuchten Verfahren vergleichbar gute Näherungen liefern. Es fällt jedoch auf, dass die H-Basen für $p = \infty$ ebenso wie das zweite Ergebnis von Fassino keinen konstanten Term bzw. zugehörige Koeffizienten im Bereich der Maschinengenauigkeit enthalten, was dem korrekten Verhalten der Bézierkurve entspricht. Fassino und Torrente erreichen dies durch eine Modifikation im Algorithmus, die den konstanten Term zwingend auf Null setzt. Eine ähnliche Forderung machen wir im H-Basen Algorithmus durch die Wahl des Startwertes als $\xi^{(0)} = (0,0)$. Auch dadurch wird das Polynom gezwungen an dieser Stelle zu interpolieren und damit muss der konstante Term Null sein. Interessanterweise führt auch die Variante *ohne* einen festgelegten Startwert zu diesem Ergebnis bzgl. des konstanten Terms, obwohl die Interpolation an $(0,0)$ nicht notwendig ist. Damit wird eine wichtige Eigenschaft der Kurve automatisch erkannt. Für $p = 2$ erkennt der H-Basen Algorithmus diese Eigenschaft ebenso wenig wie das LPA Verfahren von Fassino und Torrente und beide liefern dadurch ein Polynom mit anderem Träger als die gesuchte Funktion g.

6.4. Datensätze mit Ausreißern

In diesem Abschnitt betrachten wir ein typisches Problem aus Anwendungen, die auf inexakten Daten, wie z. B. Messungen, basieren: die Behandlung von *Datenausreißern*, d. h. einzelnen Punkten, die *stark fehlerbehaftet* sind. Dabei sei vereinfachend angenommen, dass alle Daten bis auf einen Datenausreißer exakt sind. Im Kontext von Varietäten kann dies konkretisiert werden:

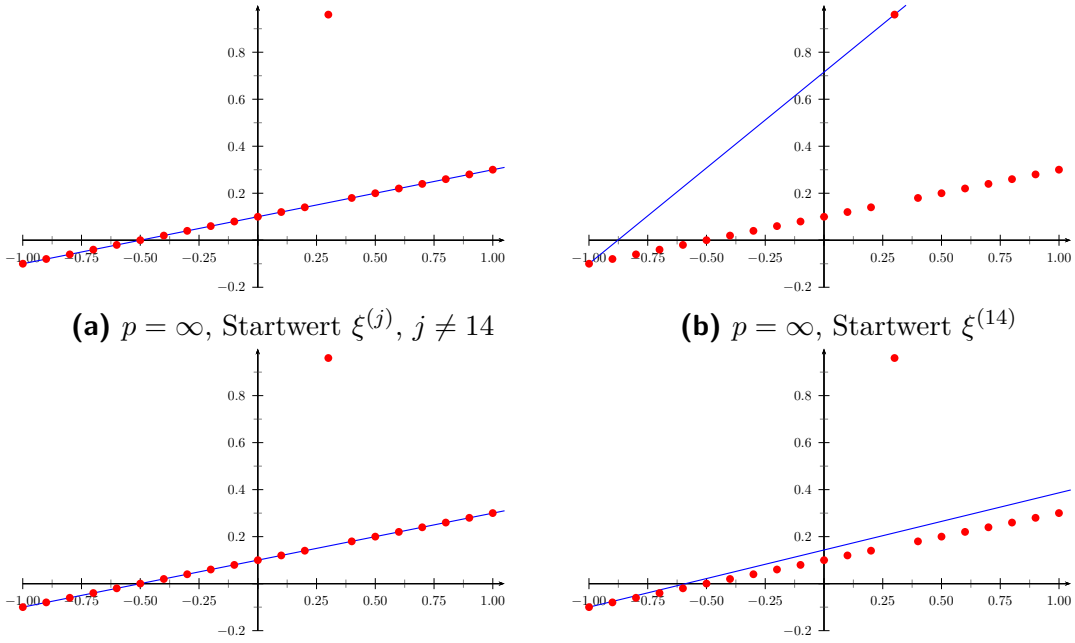
> Gegeben sei eine endliche Punktmenge $\Xi \subset \mathbb{R}^d$, die bis auf einen einzelnen Datenausreißer $\widetilde{\xi} \in \Xi$ in der Varietät eines Polynoms f liegt. Es gilt also $\Xi \not\subseteq \mathfrak{V}(f) \supseteq (\Xi \setminus \widetilde{\xi})$, $f \in \Pi_d$.

Nun stellt sich die Frage, wie robust die hier untersuchten Verfahren in der Erkennung von f sind, wenn nur die Menge Ξ bekannt ist. Wir konstruieren dazu exemplarisch eine Punktmenge, die der oben beschriebenen Situation entspricht:

$$\Xi = (\{(x, 0.2x + 0.1) : x = -1, -0.9, \ldots, 0.9, 1\} \setminus \{(0.3, 0.16)\}) \cup \{(0.3, 0.96)\} \subset \mathbb{R}^2.$$

Der Punkt $\xi^{(14)} = (0.3, 0.96)$ liegt dabei nicht mehr auf der durch $f(x) = 0.2x + 0.1$ definierten Gerade und beschreibt somit den Datenausreißer.

Nun konstruieren wir approximative H-Basen für $p = \infty$ und $p = 2$ und wählen dabei die Toleranz ε hinreichend groß, sodass die H-Basis ein lineares Polynom enthält. Die Resultate für diese linearen Polynome sind in Abbildung 6.3 graphisch dargestellt. Man sieht, dass die Gerade für $p = \infty$ nur dann von dem Datenausreißer beeinflusst wird, wenn dieser als Startpunkt gewählt wurde. Diese Beobachtung ist jedoch nicht allgemein gültig, sondern hängt vielmehr von der Lage des Datenausreißers ab. Die Begründung dafür liefern unsere Untersuchungen aus Abschnitt 4.1.3: Das lineare Polynom wird durch den Startpunkt und einen weiteren Punkt aus Ξ eindeutig festgelegt. Dieser Punkt ist so gewählt, dass er nach Drehung, Verschiebung und Skalierung der Menge Ξ den größten Abstand zum Koordinatenursprung bzgl. der euklidischen Norm hat. In unserem Beispiel liegt der Datenausreißer weit genug innen, sodass nach der Transformation der Punktmenge Ξ stets einer der Randpunkte einen größeren Abstand zum Koordinatenursprung aufweist. Dies ist natürlich im Allgemeinen nicht der Fall, insbesondere wenn der Datenausreißer am Rand der Abtastung liegt.



(a) $p = \infty$, Startwert $\xi^{(j)}$, $j \neq 14$

(b) $p = \infty$, Startwert $\xi^{(14)}$

(c) $p = \infty$, ohne Startwert

(d) $p = 2$, ohne Startwert

Abbildung 6.3.: Abtastung einer Gerade mit einem Datenausreißer an $\xi^{(14)}$ und Rekonstruktion der Gerade mit approximativen H-Basen.

Wir können jedoch festhalten, dass es für $p = \infty$ unter bestimmten Voraussetzungen möglich ist, die Gerade *ohne* eine Verfälschung durch den Datenausreißer zu rekonstruieren. Dies trifft für $p = 2$ nicht zu, da in diesem Fall *alle* Punkte für die Konstruktion des linearen Polynoms berücksichtigt werden. Letztlich bleibt noch zu bemerken, dass auch hier das lineare Polynom aus der approximativen H-Basis für $p = 2$ wieder exakt mit dem Resultat aus der Randbasis übereinstimmt, die der ABM Algorithmus liefert.

7 Zusammenfassung und Ausblick

Der Fokus dieser Arbeit liegt auf der Frage, wie man mit Hilfe rein numerischer Methoden alle Polynome bestimmen kann, die an einer gegebenen endlichen Punktmenge $\Xi \subset \mathbb{R}^d$ verschwinden. Dies führte zur Definition eines approximativen Ideals und des Verfahrens zur Konstruktion einer approximativen H-Basis von Sauer, vgl. [Sau07]. Zur Untersuchung dieses Verfahrens wurden in Kapitel 2 zunächst die Grundlagen zur Verarbeitung multivariater Polynome in einer numerischen Umgebung geschaffen. Dazu zählen die verschiedenen Ansätze zur Speicherung multivariater Polynome ebenso wie die Algorithmen zur Auswertung sowie der Durchführung elementarer Rechenoperationen. Im Falle der Multiplikation wurde gezeigt, dass die Verwendung des Kronecker-Tricks und der schnellen univariaten Faltung in vielen Fällen nicht sinnvoll ist.

Kapitel 3 lieferte eine Zusammenfassung der wichtigsten Resultate zum Thema H-Basen und mit Satz 3.19 ein bekanntes Invarianzkriterium, das die Konstruktion von H-Basen mit dünn besetzten Koeffizientenvektoren ermöglicht, vgl. Abschnitt 4.4. Ebenso wurde mit Satz 3.32 ein einfaches Kriterium zur Entscheidung gegeben, ob die Punktmenge einen niederdimensionalen geometrischen Zusammenhang beschreibt. Dies ist ein wichtiges Hilfsmittel für die Wahl des Toleranzparameters $\varepsilon > 0$ aller Algorithmen zur Konstruktion approximativer H-Basen, vgl. Abschnitt 4.1.2. An dieser Stelle ist noch zu untersuchen, wie sich die Polynome des Endlichkeitsanteils verhalten, wenn die Punktmenge in einer niederdimensionalen Untermannigfaltigkeit liegt. Damit wäre eine vollständige Abgrenzung beider Teile

möglich.

Neben dem bekannten Verfahren zur Konstruktion approximativer H-Basen von Sauer, vgl. [Sau07], wurden in Kapitel 4.1 zwei neue Algorithmen eingeführt. Dabei entfällt die Abhängigkeit von einem Startwert und es können approximative H-Basen bzgl. der euklidischen Norm berechnet werden. Eine mögliche Anpassung des Verfahrens für die Betragssummennorm $p = 1$ ist noch zu untersuchen.

Da in einer approximativen H-Basis zwar alle Basiselemente die Bedingungen des approximativen Ideals erfüllen, dies aber im Allgemeinen nicht für das von der Basis aufgespannte Ideal gilt, wurde in Abschnitt 4.2 untersucht, wie sich die Rechenoperationen des Polynomrings auf approximative Ideale auswirken. Dabei konnte ein Resultat von Sauer über die Addition zweier orthogonaler Polynome eines approximativen Ideals auf Linearkombinationen paarweise orthogonaler Polynome erweitert werden. Dies ermöglichte unter anderem eine Fehleranalyse der in Abschnitt 4.4 präsentierten Konstruktion dünn besetzter H-Basen. Weiterhin wurde eine Abschätzung für die Multiplikation von Polynomen approximativer Ideale mit beliebigen anderen Polynomen hergeleitet. Um die dazu benötigte Berechnung des kleinsten Singulärwerts einer Faltungsmatrix zu vermeiden, konnte eine neue Schranke für diesen Singulärwert angegeben werden, die ein bereits bekanntes Resultat verbessert. Da es dennoch Fälle gibt, in denen diese Schranke nur eine triviale Abschätzung liefert, besteht die Möglichkeit, die Aussage noch weiter zu verbessern. Dazu wurde eine entsprechende Vermutung formuliert.

Als Anwendungsmöglichkeit der hier präsentierten Methoden wurden in Kapitel 5 kinematische Ketten untersucht. Es konnte gezeigt werden, dass eine bzgl. der approximativen 0-Norm minimale approximative H-Basis eines planaren Bewegungsprofils die Rekonstruktion des strukturellen Aufbaus einer kinematischen Kette erlaubt. Alternativ dazu wurde die Fragestellung als explizites *Ideal-Membership-Problem* formuliert und die Lösung mittels approximativer H-Basen besprochen. Dieser Ansatz ermöglichte beim Übergang zu kinematischen Ketten im Raum auch die Erkennung von Drehgelenken mit einer festen Drehebene.

Um die hier besprochenen Algorithmen in den Kontext bereits bestehender Verfahren einzuordnen, wurden in Kapitel 6 Vergleiche mit den Verfahren von Heldt, Kreuzer, Poulisse und Pokutta, Limbeck sowie Fassino und Torente durchgeführt.

Dabei hat sich gezeigt, dass approximative H-Basen im Vergleich zu Randbasen, die mit Hilfe des ABM Algorithmus berechnet wurden, vergleichbar gute Ergebnisse erzielen. Dies wurde bereits von Limbeck in [Lim13] bemerkt. Neu sind jedoch die Erkenntnisse, dass für die gezeigten Beispiele die Randbasis und die approximative H-Basis mit $p = 2$ exakt übereinstimmen. Ebenfalls konnte an einfachen Beispielen gezeigt werden, dass die nachträgliche Minimierung der approximativen 0-Norm einer approximativen H-Basis zu den gleichen Ergebnissen führt, wie die bereits nach Konstruktion dünn besetzte Randbasis. Ein noch zu untersuchender Punkt ist das Verhalten von approximativen H-Basen bzgl. anderer Skalarprodukte. Dazu eignet sich beispielsweise das in Abschnitt 2.1 angesprochene Skalarprodukt $(f, g)_D$.

Es wurde gezeigt, dass sich approximative H-Basen auch zur Bestimmung impliziter Darstellungen von parametrischen Kurven eignen und die Ergebnisse mit den Resultaten von Fassino und Torente vergleichbar sind. Unter konstruktiven Gesichtspunkten eignen sich die hier vorgestellten Verfahren sogar besser, da durch die Auswahl eines Startwerts eine geometrische Modellierung möglich ist, die in dem Verfahren von Fassino und Torente nur durch die nicht intuitive Anpassung eines Koeffizienten erreicht wird. An dieser Stelle ist auch eine Erweiterung der hier vorgestellten Verfahren denkbar, die es ermöglicht, eine exakte Interpolation an mehreren vorgegebenen Punkten zu fordern und somit Nebenbedingungen an das approximative Ideal zu modellieren.

A

Funktionsübersicht

Inhalt

A.1. Datenstrukturen 183
A.2. Polynomoperationen 185
A.3. H-Basen 188
A.4. Approximative H-Basen 189
A.5. Wichtige Hilfsfunktionen 190
A.6. Planare kinematische Ketten 195

Dieses Kapitel gibt eine Übersicht über die wichtigsten im Rahmen dieser Arbeit entwickelten und implementierten Funktionen. Alle Funktionen wurden für die frei verfügbare Software GNU Octave geschrieben und verwenden die Syntax von Matlab. Für weitere Informationen zur Verwendung von GNU Octave sei auf [EBH08] verwiesen. Wir beschreiben im Folgenden jede Funktion bezüglich ihrer Aufgabe und geben die dafür benötigten Parameter an. Anschließend wird die Verwendung an einem kurzen Beispiel illustriert.

A.1. Datenstrukturen

In den hier vorgestellten Programmcodes werden vier verschiedene Datentypen von GNU Octave bzw. Matlab genutzt. Dabei unterscheiden wir nur die Form der

Datenstuktur und nicht deren Inhalt, wie beispielsweise die in C/C++ üblichen Datentypen `int`, `float` oder `double`. Alle hier verwendeten Datenstrukturen beinhalten Gleitkommazahlen in doppelter Genauigkeit, wobei jedoch teilweise implizit vorausgesetzt wird, dass die Werte ganzzahlig und nichtnegativ sind. Ein typisches Beispiel dafür ist der Grad eines Polynoms.

Die folgenden Datenstrukturen sind für uns von Bedeutung:

- **Skalare Variablen:** Dies ist die einfachste Datenstuktur, die hauptsächlich für Parameter wie Grad des Polynoms oder die Toleranzschwelle benutzt werden. Die Zuweisung lautet beispielsweise: `epsilon = 0.00123`.
- **Vektorwertige Variablen:** Diese Datenstruktur bildet multivariate Polynome bzw. deren Koeffizientenvektoren ab, wobei wir stets Zeilenvektoren verwenden werden. Dabei ist zu beachten, dass die Länge des Vektors zu der Variablenanzahl und dem Polynomgrad passt. Wir haben in Abschnitt 2.2 gesehen, dass man von einem Koeffizientenvektor nicht auf diese beiden Werte schließen kann. Ebenso setzen wir die Anordnung der Koeffizienten in *glex*-Ordnung voraus. Das Polynom $f(x_1, x_2) = 2x_1^2 - 3x_1x_2 + x_1 - 5$ wird also zu `f = [-5,0,1,0,-3,2]`.
- **Matrixwertige Variablen:** Matrizen werden unter anderem für Mengen von Polynomen oder Punkten verwendet. Die Matrix zu $F = \{x_1 - 2, 2x_1 + x_2 - 3\}$ lautet daher `F = [[-2,0,1];[-3,1,2]]`, die Punktmenge

  $$\Xi = \{(0,1), (2,3), (-1,5)\} \subset \mathbb{R}^2$$

  können wir durch `XI = [[0,1];[2,3];[-1,5]]` darstellen.
- **Cell-Arrays:** Ein *Cell-Array* ist ein Datentyp, der in jeder *Zelle* einen anderen Datentyp beinhalten kann. Wir werden Cell-Arrays hauptsächlich zur Darstellung von H-Basen verwenden und mit Hilfe der einzelnen Zellen dabei die Basispolynome bzgl. ihres Totalgrades partitionieren. Dabei ist zu beachten, dass die Zählung der Zellindizes bei 1 beginnt, der kleinste mögliche Polynomgrad jedoch 0 ist. Dementsprechend wird etwa die H-Basis $F = \{x_1, x_2, x_3^2\}$ dargestellt als `F = { [], [[0,0,0,1];[0,0,1,0]], [0,0,0,0,1,0,0,0,0,0] }`.

A.2. Polynomoperationen

Polynomaddition

Zur Addition zweier Polynome $f, g \in \Pi_d$ wird die Funktion `polyAdd(f,g)` verwendet, deren Argumente die Koeffizientenvektoren von f und g sind. Eine Anpassung des Grades beider Polynome auf $\deg(f+g)$ wird automatisch durchgeführt. Zur Subtraktion zweier Polynome ist der zweite Summand zu negieren, d. h. wir wenden `polyAdd(f,-g)` an.

```
octave:1> f = [1,2,3];
octave:2> g = [0,1,-1,4,0,2];
octave:3> polyAdd(f,g)
ans =   1   3   2   4   0   2
```

Polynommultiplikation

Zur Multiplikation zweier Polynome $f, g \in \Pi_d$ stehen drei unterschiedliche Funktionen zur Verfügung, die den in dieser Arbeit vorgestellten Methoden entsprechen. Alle Funktionen stimmen dabei in den Ein- und Ausgabeparametern überein.

1. Die Funktion `polyMult(f,g,d,degf,degg)` beinhaltet die Polynommultiplikation durch Multiplikation mit Termen und anschließender mit Koeffizienten gewichteter Summation.
2. In der Funktion `polyMultConv(f,g,d,degf,degg)` wird die Polynommultiplikation in Form einer Matrix-Vektor Multiplikation der Faltungsmatrix von `f` mit dem Koeffizientenvektor von `g` durchgeführt. Zur Berechnung der Faltungsmatrix wird die in A.5 beschriebene Funktion `convmat` benutzt.
3. Die Funktion `polyMultFFT(f,g,d,degf,degg)` multipliziert zwei Polynome mit Hilfe der schnellen Fouriertransformation unter Verwendung des Kronecker-Tricks. Die schnelle Faltung durch die FFT wird dabei von der Octave-internen Funktion `fftconv` berechnet.

```
octave:1> f=[1,0,2];
octave:2> g=[-3,2,-1];
octave:3> polyMult(f,g,2,1,1)
ans = -3   2  -7   0   4  -2
octave:4> polyMultConv(f,g,2,1,1)
ans = -3   2  -7   0   4  -2
octave:5> polyMultFFT(f,g,2,1,1)
ans = -3   2  -7   0   4  -2
```

Dies entspricht der Berechnung von

$$(2x_1 + 1)(-x_1 + 2x_2 - 3) = -2x_1^2 + 4x_1x_2 - 7x_1 + 2x_2 - 3.$$

Polynomdivision

Die Division eines Polynoms $g \in \Pi_d$, $\deg(g) = k$, durch eine endliche Menge von Polynomen $F \subset \Pi_d$ im Sinne von Algorithmus 2.49 ist in der Funktion `[gF,r] = polyDiv(F,g,k,d,epsilon)` realisiert. Dabei beschreibt `F` ein Cell-Array, das die Koeffizientenvektoren der Polynome F bzgl. ihres Totalgrades gruppiert zusammenfasst, vgl. Abschnitt A.1. Zudem stellt `g` den Koeffizientenvektor von g dar und `k` bzw. `d` entsprechen wie üblich dem Polynomgrad von g sowie der Anzahl an Variablen. Das Argument `epsilon` wird für die Bestimmung der *approximativen Normalform*, vgl. Abschnitt 4.3 verwendet.

Als Resultat erhält man ein Cell-Array `gF` und einen Koeffizientenvektor `r`, der den Rest der Polynomdivision beschreibt. Das Cell-Array `gF` enthält die Faktoren g_f zu den Polynomen $f \in F$ mit $\deg(g_f) = k - \deg(f)$. Die Reihenfolge in `gF` entspricht dabei der Reihenfolge der zugehörigen Polynome im Cell-Array `F`.

```
octave:1> F={ [], [0,1,0], [1,0,0,0,0,1] };
octave:2> g=[-4,0,0,1,0,1];
octave:3> [gF,r] = polyDiv(F,g,2,2,10^-8)
gF =
{
```

```
  [1,1] = [](0x0)
  [1,2] = 0  1  0
  [1,3] = 1
}
r = -5   0   0   0   0   0
```

Wir erhalten also für $g(x_1, x_2) = x_1^2 + x_2^2 - 4$ und $f_1(x_1, x_2) = x_2$, $f_2(x_1, x_2) = x_1^2 + 1$ eine Darstellung $g(x_1, x_2) = x_2 \cdot f_1(x_1, x_2) + 1 \cdot f_2(x_1, x_2) - 5$.

Polynomauswertung

Zur Auswertung eines multivariaten Polynoms stehen zwei Methoden zur Verfügung:

- Die Funktion `HornerEval(f,k,d,XI)` wertet das Polynom mit dem Koeffizientenvektor `f` an den Stellen `XI` aus. Dabei wird das in Abschnitt 2.3 beschriebene multivariate *Horner-Schema* in der Variante von de Boor verwendet.
- Die Funktion `HornerEval2(f,k,d,XI)` wendet ebenfalls das Horner-Schema an, allerdings in der Variante von Sauer und Peña. Die notwendige Indexkonvertierung ist bereits in die Funktion integriert, sodass beide Funktionen einen Koeffizientenvektor in graduiert-lexikographischer Ordnung voraussetzen. In einem Laufzeitvergleich der Verfahren sollte die Indexkonvertierung berücksichtigt werden. Details dazu sind in [CS14] beschrieben.

Beide Methoden benötigen zur Interpretation des Koeffizientenvektors wie üblich den Grad `k` des Polynoms sowie die Anzahl der Variablen als Parameter `d`.

```
octave:1> f=[4,2,0,3,1,5];
octave:2> XI=[[1,2];[0,3];[-1,1]];
octave:3> HornerEval(f,2,2,XI)'
ans =   27   37   13
octave:4> HornerEval2(f,2,2,XI)'
ans =   27   37   13
```

Da in unseren Algorithmen in der Regel mehr als ein Polynom an der Punktmenge `XI` ausgewertet werden muss, wurde die Funktion `VanderMat(F,k,d,XI)` erstellt, die unter Verwendung von `HornerEval(f,k,d,XI)` die *Vandermonde-Matrix* berechnet. Dabei beschreibt `F` die Matrix der Koeffizientenvektoren einer Menge $F \subset \Pi_{k,d} \setminus \Pi_{k-1,d}$. Alternativ berechnet die Funktion `VanderMat2(F,k,d,XI)` die gleiche Vandermonde-Matrix *ohne* Verwendung des Horner-Schemas. Allerdings dient diese Funktion nur dem Vergleich und sollte aus Gründen der numerischen Stabilität sowie der Effizienz nicht verwendet werden.

A.3. H-Basen

Das in Abschnitt 3.2 beschriebene Verfahren zur Berechnung einer H-Basis aus einem endlichen Erzeugendensystem eines nulldimensionalen Ideals (vgl. Algorithmus 3.16) von Möller und Sauer ist in der Funktion `F=makeHBasis(F,d,epsilon)` umgesetzt. Dabei wird ein Erzeugendensystem `F` als Cell-Array sowie die Variablenanzahl `d` und eine Toleranz `epsilon` benötigt, die für die notwendigen Reduktionen von Polynomen verwendet wird. Als Rückgabewert erhält man eine H-Basis in Form des Cell-Arrays `F`. Zur Verdeutlichung wird die Funktion exemplarisch auf das in Beispiel 3.17 gegebene Problem angewendet. Wir betrachten das Erzeugendensystem $F = \{x_1 - x_2, x_2x_3 - 5x_3 + 4x_1, x_1^2 - x_3\}$:

```
octave:1> F{3}=[0,0,-1,0,0,0,0,0,0,1;0,-5,0,4,0,1,0,0,0,0];
octave:2> F{4}=[0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1];
octave:3> F=makeHBasis(F,3,10^-10)
F = {
[1,1] = [](0x0)
[1,2] = [](0x0)
[1,3] =
 0   0.00000  -1  0  0.00000  0   0.00000   0.00000  0.00000  1.00000
 0  -5.00000   0  4  0.00000  1   0.00000   0.00000  0.00000  0.00000
 0  -0.70711   0  0  0.00000  0   0.00000   0.00000  0.70711  0.00000
 0   0.00000   0  0  0.00000  0  -0.57735   0.57735  0.00000  0.00000
 0   0.00000   0  0  0.57735  0   0.00000  -2.88675  0.00000  2.30940
```

```
[1,4] = [](0x20)
}
```

Dies liefert bis auf Normierung die reduzierte H-Basis

$$F = \{x_1 - x_2, x_2x_3 - 5x_3 + 4, x_1x_2 - x_3, x_1x_3 - x_3^2, 4x_1^2 - 5x_1x_3 + x_3^2\}.$$

A.4. Approximative H-Basen

Sämtliche in Abschnitt 4.1 vorgestellten Verfahren zur Berechnung einer *approximativen H-Basis* sind in der Funktion

```
[F0,Fp] = approxHBasis(XI,epsilon,startwert,nrm)
```

implementiert. Allen Verfahren gemeinsam ist die Abhängigkeit von einer Matrix `XI`, die eine endliche Punktmenge $\Xi \subset \mathbb{R}^d$ beschreibt, und einer Toleranz `epsilon` als Schwellenwert für das approximative Ideal. Die Parameter `startwert` und `nrm` bestimmen schließlich, welcher Algorithmus zur Konstruktion der approximativen H-Basis verwendet wird:

- Wählt man `nrm = Inf` und für `startwert` einen Wert aus $\{1, \ldots, \#\Xi\}$, so wird die approximative H-Basis nach Algorithmus 4.12 bestimmt.
- Für `nrm = Inf` und `startwert = NaN` wird dem Verfahren kein Startwert mehr vorgegeben. Damit wird zur Bestimmung der approximativen H-Basis auf Algorithmus 4.23 zurückgegriffen.
- Im Fall `nrm = 2` und `startwert = NaN` wird eine Implementierung von Algorithmus 4.26 verwendet. Da in diesem Verfahren keine Wahl des Startwerts möglich ist, muss für `nrm = 2` auch stets `startwert = NaN` gesetzt werden.

Für alle drei Fälle erhalten wir mit dem *Cell-Array* `F0` eine approximative H-Basis von $\mathfrak{I}_{p,\infty}(\Xi)$, $p \in \{2, \infty\}$. Werden die letzten beiden Parameter nicht angegeben, so wird automatisch `nrm = Inf` und `startwert=1` gesetzt.

```
octave:1> XI = [-1,-1;-1,1;1,-1;1,1];
octave:2> [F0,Fp] = approxHBasis(XI,10^-10);
octave:3> F0
F0 = {
[1,1] = [](0x0)
[1,2] = [](0x3)
[1,3] =
-8.1650e-01 -5.5511e-17  5.5511e-17  4.0825e-01 -5.5511e-17 4.0825e-01
 1.1102e-16 -2.7756e-17 -2.7756e-17 -7.0711e-01 -4.1633e-17 7.0711e-01
}
```

So erhalten wir nach Entfernen der *numerischen Nullen* und passender Umnormierung die H-Basis $F = \{x_1^2 + x_2^2 - 2, x_1^2 - x_2^2\}$ und es gilt $\langle F \rangle = \mathfrak{I}(\Xi)$ mit $\Xi = \{(\pm 1, \pm 1), (\pm 1, \mp 1)\}$.

A.5. Wichtige Hilfsfunktionen

Approximative 0-Norm Minimierung

Zur Bestimmung einer Basis des Spaltenraums der Matrix $A \in \mathbb{R}^{n \times m}$, $\operatorname{rank}(A) = m$, mit minimaler 0-Norm wird die Funktion `A = min0norm(A,epsilon)` verwendet. Der Parameter `epsilon` gibt dabei die Toleranz $\varepsilon > 0$ der approximativen 0-Norm an.

```
octave:1> A=[3,1;-4,2;5,6];
octave:2> min0norm(A,10^-10)
ans =
   0.29412  -0.38235
  -0.00000   1.00000
   1.00000  -0.00000
```

Man beachte, dass die Funktion eine Basis des *Spaltenraums* der Matrix berechnet. Da wir in Abschnitt A.1 vereinbart haben, dass Polynome stets als Zeilenvektoren

gespeichert werden, müssen wir eine Koeffizientenmatrix entsprechend transponieren. Das folgende Beispiel greift dazu die in Abschnitt A.4 berechnete H-Basis auf und minimiert die quadratischen Basispolynome:

```
octave:1> XI = [-1,-1;-1,1;1,-1;1,1];
octave:2> [F0,Fp] = approxHBasis(XI,10^-10);
octave:3> F0{3} = min0norm(F0{3}',10^-10)'
F0 = {
  [1,1] = [](0x0)
  [1,2] = [](0x3)
  [1,3] =
    -1.00000  -0.00000   0.00000   0.00000  -0.00000   1.00000
    -1.00000  -0.00000   0.00000   1.00000  -0.00000   0.00000
}
```

Somit erhalten wir die bzgl. der 0-Norm minimale Basis $F = \{x_1^2 - 1, x_2^2 - 1\}$.

Termmatrizen

Die in Abschnitt 2.2 beschriebenen *Termmatrizen* $\boldsymbol{T}_{k,d}$ bzw. die *homogenen Termmatrizen* $\boldsymbol{T}^0_{k,d}$ können mit den Funktionen `T=termMat(k,d)` und `T=termMatH(k,d)` bestimmt werden. Dabei werden die Matrizen $\boldsymbol{T}_{k,d}$ blockweise aus den homogenen Termmatrizen $\boldsymbol{T}^0_{j,d}$, $j \leq k$, zusammengesetzt, die wiederum nach der Rekursionsformel aus Lemma 2.20 berechnet werden.

```
octave:1> T=termMatH(2,3)
T =
   0   0   2
   0   1   1
   0   2   0
   1   0   1
   1   1   0
   2   0   0
```

```
octave:2> T=termMat(2,2)
T =
   0   0
   0   1
   1   0
   0   2
   1   1
   2   0
```

Homogene Teilräume

Die Funktion `[Vk,Wk] = GetVkWk(F,k,d)` berechnet zu einer Menge von Polynomen $F \subset \Pi_d$ je eine Basis der homogenen Teilräume $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$. Dabei sind die Polynome bzgl. ihres Grades geordnet als *Cell-Array* `F` anzugeben, vgl. Abschnitt A.1.

```
octave:1> F = { [], [[0,-1,0,1];[0,0,-1,1]], [0,0,0,0,1,0,1,0,0,1] };
octave:2> [Vk,Wk] = GetVkWk(F,1,3)
Vk =
   0.00000   0.40825   0.40825  -0.81650
   0.00000   0.70711  -0.70711   0.00000
Wk =
   0.00000   0.57735   0.57735   0.57735
octave:3> [Vk,Wk] = GetVkWk(F,2,3)
Vk =
   0 0 0 0   0.36273   0.22595   0.36273  -0.42307  -0.42307   0.57257
   0 0 0 0  -0.27060   0.00000   0.27060   0.65328  -0.65328   0.00000
   0 0 0 0   0.33333  -0.66667   0.33333   0.33333   0.33333   0.33333
   0 0 0 0  -0.50524  -0.21705  -0.50524  -0.04297  -0.04297   0.66231
   0 0 0 0   0.04525   0.67631   0.04525   0.45613   0.45613   0.34986
   0 0 0 0  -0.65328   0.00000   0.65328  -0.27060   0.27060  -0.00000
Wk = [](0x10)
```

Dies bedeutet insbesondere, dass $\mathcal{W}^0_{2,3} = \text{span}\{\emptyset\} = \{0\}$ und damit die Menge

$$F = \{x_1 - x_3, x_1 - x_2, x_1^2 + x_2^2 + x_3^2\}$$

eine H-Basis ist.

Faltungsmatrix

Die *Faltungsmatrix* $\boldsymbol{C}_{k,d}(f)$ zu einem Polynom $f \in \Pi_d$, $\deg(f) \leq k$, kann mit der Funktion `Cf = convMat(f,degf,k,d)` berechnet werden.

```
octave:1> f=[3,-1,2];
octave:2> Cf = convMat(f,1,3,2)
Cf =
   3   0   0   0   0   0
  -1   3   0   0   0   0
   2   0   3   0   0   0
   0  -1   0   3   0   0
   0   2  -1   0   3   0
   0   0   2   0   0   3
   0   0   0  -1   0   0
   0   0   0   2  -1   0
   0   0   0   0   2  -1
   0   0   0   0   0   2
```

Die Matrix $\boldsymbol{C}_{k,d}(f)$ wird dabei blockweise aus den *homogenen Faltungsmatrizen* $\boldsymbol{C}^0_{j,d}(f)$, $\deg(f) \leq j \leq k$, zusammengesetzt, die wiederum mit Hilfe der Funktion `Cf = convMatHom(F,j,d)` bestimmt werden. Im Gegensatz zu der Funktion `convMat` wird dabei ein *Cell-Array* `F` benötigt, sodass auch die homogene Faltungsmatrix einer Menge von Polynomen $F \subset \Pi_d$ berechnet werden kann. Schließlich liefert die Funktion `Cf = convMatHomLT(F,k,d)` die homogene Faltungsmatrix der Leitformen $\Lambda(F)$ in der abgeschnittenen Variante $\widetilde{\boldsymbol{C}^0_{k,d}(\Lambda(F))}$, $\deg(f) \leq k$, die in Abschnitt 2.4.4 beschrieben wurde. Die Funktionen `convMat` und `convMatLT` erzeugen die Faltungsmatrix dabei in transponierter Form.

```
octave:1> F{2}=[1,2,3];
octave:2> F{3}=[-2,0,2,3,2,4,1,0,5];
octave:3> convMatHom(F,2,3)
ans =
   0   1   0   0   2   3   0   0   0   0
   0   0   1   0   0   2   3   0   0   0
   0   0   0   1   0   0   0   2   3   0
  -2   0   2   3   2   4   1   0   5   0
octave:4> convMatHomLT(F,2,3)
ans =
   1   2   0   3   0   0
   0   1   2   0   3   0
   0   0   0   1   2   3
   3   2   4   1   0   5
```

Zeilenstufenform mittels QR-Zerlegung

Um eine Matrix $A \in \mathbb{R}^{n \times m}$ in *Zeilenstufenform* zu bringen, können wir die Funktion `A = refQR(A,epsilon)` verwenden. Dabei beschreibt `epsilon` die Toleranzschwelle für zulässige Pivotelemente in einer Zeile. Das folgende Beispiel zeigt eine Matrix, für die in der QR-Zerlegung zwar eine Dreiecksmatrix generiert wird, die aber *nicht* in Zeilenstufenform ist. Die Funktion `refQR` liefert die gewünschte Form.

```
octave:1> A=[1,2,3,4;0,0,5,6;0,0,7,8];
octave:2> [Q,R]=qr(A); R
R =
   1   2   3   4
   0   0   5   6
   0   0   7   8
octave:3> R=refQR(A,10^-10)
R =
  -1.00000  -2.00000  -3.00000  -4.00000
   0.00000   0.00000  -8.60233  -9.99730
   0.00000   0.00000  -0.00000   0.23250
```

A.6. Planare kinematische Ketten

Zur Erkennung von Gelenken einer *planaren kinematischen Kette* aus einem *Bewegungsprofil* $\Xi \subset \mathbb{R}^{N\times 2n}$ wurde die Funktion `l = detectJoint(F0,n,tol)` entwickelt. Diese hängt von einer approximativen H-Basis `F0`, der Anzahl der Gelenke `n` und einer Toleranzschwelle `tol`, die für die Lösung des approximativen Ideal-Membership-Problems benötigt wird, ab. Als Rückgabewert erhält man – im Falle von Drehgelenken – einen Vektor `l`, der die Längen der Verbindungen beschreibt. Zusätzlich wird eine Textausgabe erzeugt, die die erkannten Gelenke auflistet.

Die Funktion `XI = planeRobot(N,conf,len)` dient der Erstellung von Testdatensätzen zur Gelenkerkennung. Dabei wird ein *Bewegungsprofil* einer planaren kinematischen Kette generiert, wobei die Drehwinkel bzw. Schublängen zufällig verteilt sind. Als Parameter werden die Anzahl der Datenpunkte `N`, ein String `conf`, der die Struktur der kinematischen Kette beschreibt („r“ für Drehgelenke, „p“ für Schubgelenke) und ein Vektor `len` für die (maximalen) Längen der Verbindungen benötigt.

Das folgende Beispiel beschreibt die *Gelenkerkennung* einer planaren kinematischen Kette, die aus zwei Drehgelenken mit Längen 3 und 4 und einem Schubgelenk mit maximaler Länge 2 besteht. Der erzeugte Testdatensatz beinhaltet dabei $N = 200$ Datenpunkte und die Toleranzschwelle ist als $\varepsilon = 10^{-10}$ gewählt.

```
octave:1> XI=planeRobot(100,"rrp",[3,4,2]);
octave:2> [F0,Fp] = approxHBasis(XI,10^-10);
octave:3> l=detectJoint(F0,3,10^-10)
Drehgelenk: Aufhaengung -> (x1,y1), l=3.000000.
Drehgelenk: (x1,y1) -> (x2,y2), l=4.000000.
Schubgelenk: (x2,y2) -> (x3,y3).
l =
   3.0000   4.0000
```

Symbolverzeichnis

Bezeichnung	Beschreibung	Seiten
$(\cdot,\cdot)$	Monomiales Skalarprodukt	11
$(\cdot,\cdot)_D$	Gewichtetes monomiales Skalarprodukt	13
$\oplus$	Direkte Summe	10
$\|f(\Xi)\|_p$	Auswertungsnorm von $f \in \Pi_d$ an $\Xi \subset \mathbb{R}^d$	76
$\|f\|_0$	0-Norm von $f \in \Pi_d$, $\#\operatorname{supp}(f)$	113
$\|f\|_{0,\varepsilon}$	Approximative 0-Norm von $f \in \Pi_d$	130
$\|f\|_2$	Koeffizientennorm von $f \in \Pi_d$, $\sqrt{(f,f)}$	76
$\mathbb{B}_\infty^d$	d-dimensionale Einheitskugel bzgl. der Maximumsnorm	111
$\mathbb{C}$	Menge der komplexen Zahlen	62
$\boldsymbol{C}_{k,d}(f)$	Faltungsmatrix vom Grad $k \geq \deg(f)$ des Polynoms $f \in \Pi_d$ in glex Ordnung	31
$\boldsymbol{C}_{k,d}^0(f)$	Homogene Faltungsmatrix vom Grad $k \geq \deg(f)$ des Polynoms $f \in \Pi_d$ in glex Ordnung	39
$\widetilde{\boldsymbol{C}_{k,d}^0(\Lambda(F))}$	Abgeschnittene homogene Faltungsmatrix vom Grad $k \geq \deg(f)$ der Leitformen $\Lambda(F) \subset \Pi_{k,d}^0$ in glex Ordnung	40

Bezeichnung	Beschreibung	Seiten
$\boldsymbol{C}_{k,d}^{\prec}(f)$	Faltungsmatrix vom Grad $k \geq \deg(f)$ des Polynoms $f \in \Pi_d$ bzgl. der Termordnung $\prec$	121
$\deg(f)$	Totalgrad eines Polynoms $f \in \Pi_d$	10
e_j	j-ter Einheitsvektor $(\underbrace{0,\ldots,0}_{j-1},1,0,\ldots,0)^T$	79
ε_i	i-ter Einheitsmultiindex $(\underbrace{0,\ldots,0}_{i-1},1,0,\ldots,0) \in \mathbb{N}_0^d$	23
$\langle F \rangle$	Von $F \subseteq \Pi_d$ erzeugtes Ideal	51
$\mathfrak{I}(\Xi)$	Verschwindungsideal der Punktmenge $\Xi \subset \mathbb{R}^d$	63
$\mathfrak{I}_{p,\varepsilon}(\Xi)$	p-approximatives Ideal der Punktmenge $\Xi \subset \mathbb{R}^d$ mit Toleranz $\varepsilon > 0$	76
$\Lambda(f)$	Leitform eines Polynoms $f \in \Pi_d$	11
$\mathcal{N}(A)$	Nullraum der linearen Abbildung A	41
$\mathbb{N}$	Menge der natürlichen Zahlen $\{1,2,\ldots\}$	
$\mathbb{N}_0$	Menge der natürlichen Zahlen *mit* Null $\{0,1,2,\ldots\}$	
$\mathbb{N}_0^d$	Menge aller Multiindizes in d Variablen	8, 9
$\nu_{\mathcal{F}}(g)$	Normalform von g bzgl. des Ideals $\mathcal{F}$	56
$\nu_{\mathcal{F},\varepsilon}(g)$	Approximative Normalform von g bzgl. des Ideals $\mathcal{F}$	126
Π_d	Ring der Polynome über $\mathbb{R}$ in d Variablen	9
Π_d^0	Menge aller homogenen Polynome über $\mathbb{R}$ in d Variablen	10
$\overline{\Pi}_d$	Ring der Polynome über $\mathbb{C}$ in d Variablen	62
$\Pi_{k,d}$	Raum aller Polynome über $\mathbb{R}$ in d Variablen mit Grad höchstens k	10

Bezeichnung	Beschreibung	Seiten
$\Pi^0_{k,d}$	Raum aller homogenen Polynome über $\mathbb{R}$ in d Variablen mit Grad k	10
$\boldsymbol{P}^{\prec}_{k,d}$	Permutationsmatrix zur Umordnung der Terme aus $\boldsymbol{T}_{k,d}$ bzgl. $\prec$	119
$\mathbb{Q}$	Menge der rationalen Zahlen	7
$\mathbb{R}$	Menge der reellen Zahlen	7
$\mathcal{R}(A)$	Bildraum der linearen Abbildung A	41
$\operatorname{supp}(f)$	Träger des Koeffizientenvektors von $f \in \Pi_d$	112
$\boldsymbol{T}_{k,d}$	Termmatrix in glex Ordnung	16, 17
$\boldsymbol{T}^0_{k,d}$	Homogene Termmatrix in glex Ordnung	17
$\mathfrak{V}(F)$	Menge aller gemeinsamen Nullstellen (Varietät) der Polynome $F \subset \Pi_d$	62
$\mathcal{V}_d(F)$	Von $\Lambda(F)$ erzeugter linearer Teilraum von Π_d	12
$\mathcal{V}_{k,d}(F)$	Von $\Lambda(F)$ erzeugter linearer Teilraum von $\Pi_{k,d}$	12
$\mathcal{V}^0_{k,d}(F)$	Von $\Lambda(F)$ erzeugter linearer Teilraum von $\Pi^0_{k,d}$	12
$\mathcal{W}_d(F)$	Orthogonales Komplement von $\mathcal{V}_d(F)$	12
$\mathcal{W}_{k,d}(F)$	Orthogonales Komplement von $\mathcal{V}_{k,d}(F)$	12
$\mathcal{W}^0_{k,d}(F)$	Orthogonales Komplement von $\mathcal{V}^0_{k,d}(F)$	12
$\overline{\Xi}$	Zariski-Abschluss der Menge $\Xi \subseteq \mathbb{C}^d$	66
$\mathbb{Z}$	Menge der ganzen Zahlen $\{\dots, -2, -1, 0, 1, 2, \dots\}$	

Liste der Algorithmen

2.23. Multivariate Polynomauswertung mittels Horner-Schema 23

2.24. *glex*-Version von Algorithmus 2.23 23

2.30. Multiplikation eines multivariaten Polynoms mit einem Term 28

2.38. Polynommultiplikation mittels univariater Faltung 36

2.46. Bestimmung einer Basis von $\mathcal{V}^0_{k,d}(F)$ und $\mathcal{W}^0_{k,d}(F)$ 43

2.49. Verallgemeinerte Polynomdivision 47

3.16. Bestimmung einer H-Basis 58

4.6. Abgebrochene QR-Zerlegung mit Spaltenpivotisierung 79

4.12. Bestimmung einer approximativen H-Basis 84

4.20. Zeilenstufenform via QR-Zerlegung 100

4.23. Bestimmung einer approximativen H-Basis von $\mathfrak{I}_{\infty,\varepsilon}(\Xi)$ *ohne* Interpolation 101

4.26. Bestimmung einer approximativen H-Basis von $\mathfrak{I}_{2,\varepsilon}(\Xi)$ 106

4.62. Approximative 0-Norm Minimierung 131

4.65. Vollständige approximative 0-Norm Minimierung 133

Abbildungsverzeichnis

2.1. Auswertungsschema für multivariate Polynome 22
2.2. Schematische Darstellung der Multiplikation eines Polynoms mit einem Term 29
2.3. Faltungsmatrizen eines linearen Polynoms 33
2.4. Vergleich der Länge von Koeffizientenvektoren vor und nach der Substitution durch den Kronecker-Trick. 38

3.1. Graphische Darstellung eines monischen Ideals. 52
3.2. Varietät des Ideals $\langle 2x_1 - x_2 + 1, 2x_1^3 - x_2 + 1\rangle$ als Schnittpunkte von einer Geraden und einer Hyperbel. 65
3.3. Reelle Anteile der Varietäten zu den Idealen des Gegenbeispiels aus Satz 3.31 69

4.1. Varietäten der Basispolynome aus Beispiel 4.17. 90
4.2. Punkte aus Beispiel 4.18 und Polynom kleinsten Grades der approximativen H-Basis. 92
4.3. Aufbau einer Matrix L aus Algorithmus 4.23. 103
4.4. Minimale Singulärwerte der Faltungsmatrix $\boldsymbol{C}_{1+k,2}(4x_1 + 2x_2 + 1)$. . . 124

5.1. Schematische Darstellung der beiden Gelenktypen einer planaren kinematischen Kette. 140

5.2. Bewegungsprofile einer planaren kinematischen Kette aus zwei Drehgelenken mit $\ell_1 = 3$ und $\ell_2 = 1$. 143
5.3. Bewegungsprofil eines Schubgelenks in Richtung v in exakter und fehlerbehafteter Darstellung. 143
5.4. Laufzeit der Gelenkerkennung mittels Minimierung der 0-Norm bzw. Polynomdivision in logarithmischer Skalierung. 150
5.5. Humanoides Modell als Anwendungsbeispiel der Gelenkerkennung . . 151
5.6. Darstellung der Gelenkpositionen eines Drehgelenks mit fester Drehebene. 155
5.7. Archimedische Spiralen in der Ebene 161
5.8. Archimedische Spiralen im Raum 162
5.9. Abtastung einer Helix 164

6.1. Parabel aus Beispiel 6.1 und eine fehlerbehaftete Abtastung der Funktionswerte. 167
6.2. Bézierkurve aus Beispiel 6.5 für $1 \leq t \leq 2$. Die Markierungen zeigen die abgetasteten und gerundeten Werte Ξ. 174
6.3. Abtastung einer Gerade mit einem Datenausreißer an $\xi^{(14)}$ und Rekonstruktion der Gerade mit approximativen H-Basen. 177

Literaturverzeichnis

[Aig06] Aigner, Martin: *Diskrete Mathematik*. 6., korr. Aufl. Vieweg+Teubner Verlag, 2006 (Aufbaukurs Mathematik). – ISBN 978–3–8348–0084–8

[AT13] ApCoCoA-Team: *ApCoCoA: Applied Computations in Commutative Algebra*. http://www.apcocoa.org, 2013. – zuletzt abgerufen: 03.03.2015

[Bat13] Batselier, Kim: *A Numerical Linear Algebra Framework for Solving Problems with Multivariate Polynomials*, University College Cork, Dissertation, 2013

[BG65] Businger, Peter ; Golub, Gene H.: Linear least squares solutions by Householder transformations. In: *Numerische Mathematik* 7 (1965), Nr. 3, S. 269–276

[Bjö94] Björck, Åke: Numerics of gram-schmidt orthogonalization. In: *Linear Algebra and Its Applications* 197 (1994), S. 297–316

[Bla00] Blanck, Jens: Domain representations of topological spaces. In: *Theoretical Computer Science* 247 (2000), Nr. 1, S. 229–255

[Boo00] Boor, Carl de: Computational aspects of multivariate polynomial interpolation: Indexing the coefficients. In: *Advances in Computational Mathematics* 12 (2000), Nr. 4, S. 289–301

[BP92] Brawer, Robert ; Pirovino, Magnus: The linear algebra of the Pascal matrix. In: *Linear Algebra and Its Applications* 174 (1992), S. 13–23

[BR90] Boor, Carl de ; Ron, Amos: On multivariate polynomial interpolation. In: *Constructive Approximation* 6 (1990), Nr. 3, S. 287–302. DOI: 10.1007/BF01890412. – ISSN 0176–4276

[Buc65] Buchberger, Bruno: *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Universität Innsbruck, Dissertation, 1965

[BW93] Becker, Thomas ; Weispfenning, Volker: *Gröbner bases*. Springer-Verlag New York, 1993 (Graduate Texts in Mathematics). – ISBN 978–0–387–97971–7

[CLO98] Cox, David A. ; Little, John B. ; O'Shea, Donal: *Using algebraic geometry*. Springer-Verlag New York, 1998 (Graduate Texts in Mathematics). – ISBN 978–0–387–98492–6

[CLO07] Cox, David A. ; Little, John B. ; O'Shea, Donal: *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. 3rd. Springer-Verlag New York, 2007 (Undergraduate Texts in Mathematics). – ISBN 978–0–387–35650–1

[CS14] Czekansky, Johannes ; Sauer, Tomas: The multivariate Horner scheme revisited. In: *BIT Numerical Mathematics* (2014). DOI: 10.1007/s10543-014-0533-x. – ISSN 1572–9125

[CT65] Cooley, James W. ; Tukey, John W.: An Algorithm for the Machine Calculation of Complex Fourier Series. In: *Mathematics of Computation* 19 (1965), Nr. 90, S. 297–301

[CT14] CoCoA-Team: *CoCoA: a system for doing Computations in Commutative Algebra*. http://cocoa.dima.unige.it, 2014. – zuletzt abgerufen: 03.03.2015

[DGPS14] Decker, Wolfram ; Greuel, Gert-Martin ; Pfister, Gerhard ; Schönemann, Hans: Singular *4-0-1 — A computer algebra system for poly-*

nomial computations. http://www.singular.uni-kl.de, 2014. – zuletzt abgerufen: 03.03.2015

[EBH08] Eaton, John W. ; Bateman, David ; Hauberg, Søren: *GNU Octave Manual Version 3*. Network Theory Ltd., 2008 http://www.gnu.org/software/octave/. – ISBN 978–0–9546120–6–1

[EK12] Eldar, Yonina C. ; Kutyniok, Gitta: *Compressed sensing: Theory and Applications*. Cambridge University Press, 2012

[FH07] Freund, Roland W. ; Hoppe, Ronald H. W.: *Stoer/Bulirsch: Numerische Mathematik 1*. Springer Berlin Heidelberg, 2007 (Springer-Lehrbuch). – ISBN 978–3–540–45389–5

[Fis05] Fischer, Gerd: *Lineare Algebra*. Vieweg, 2005 (Vieweg-Studium : Grundkurs Mathematik). – ISBN 978–3–8348–0031–2

[Fre14] Free Software Foundation: *The GNU Multiple Precision Arithmetic Library*. https://gmplib.org, 2014. – zuletzt abgerufen: 03.03.2015

[FT13] Fassino, Claudia ; Torrente, Maria-Laura: Simple varieties for limited precision points. In: *Theoretical Computer Science* 479 (2013), S. 174–186

[Ger31] Gerschgorin, Semjon A.: Über die Abgrenzung der Eigenwerte einer Matrix. In: *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na* 6 (1931), S. 749–754

[GG03] Gathen, Joachim von z. ; Gerhard, Jürgen: *Modern Computer Algebra*. Cambridge University Press, 2003. – ISBN 978–0–521–82646–4

[Grö49] Gröbner, Wolfgang: *Moderne Algebraische Geometrie: Die Idealtheoretischen Grundlagen*. Springer-Verlag, Wien, Insbruck, 1949. – ISBN 978–3–7091–5740–4

[GS00] Gasca, Mariano ; Sauer, Thomas: Polynomial interpolation in several variables. In: *Advances in Computational Mathematics* 12 (2000), Nr. 4, S. 377–410

[GSE14] Grayson, Daniel ; Stillman, Michael ; Eisenbud, David: *Macaulay2 - a software system for research in algebraic geometry*. http://www.math.uiuc.edu/Macaulay2/, 2014. – zuletzt abgerufen: 03.03.2015

[GVL96] Golub, Gene H. ; Van Loan, Charles F.: *Matrix computations*. 3rd edn. The Johns Hopkins University Press, 1996. – ISBN 978–0–8018–5414–9

[HB02] Hanke-Bourgeois, Martin: *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. 1. Auflage. Teubner, Stuttgart/Leipzig/Wiesbaden, 2002. – ISBN 978–3–519–00356–4

[Hig02] Higham, Nicholas J.: *Accuracy and Stability of Numerical Algorithms*. 2nd. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2002. – ISBN 978–0–89871–521–7

[HJ91] Horn, Roger A. ; Johnson, Charles R.: *Topics in Matrix Analysis*. Cambridge University Press, 1991. – ISBN 978–0–511–84037–1. – Cambridge Books Online

[HKPP09] Heldt, Daniel ; Kreuzer, Martin ; Pokutta, Sebastian ; Poulisse, Hennie: Approximate computation of zero-dimensional polynomial ideals. In: *Journal of Symbolic Computation* 44 (2009), Nr. 11, S. 1566–1591

[HKSS97] Husty, Manfred ; Karger, Adolf ; Sachs, Hans ; Steinhilper, Waldemar: *Kinematik und Robotik*. Springer-Verlag Berlin Heidelberg, 1997. – ISBN 978–3–540–63181–1

[JN00] Johnson, Charles R. ; Nylen, Peter M.: Hadamard product submultiplicativity of certain induced norms. In: *Linear and Multilinear Algebra* 48 (2000), Nr. 2, S. 165–178

[Joh89] Johnson, Charles R.: A Gersgorin-type lower bound for the smallest singular value. In: *Linear Algebra and its Applications* 112 (1989), Nr. 0, S. 1 – 7. DOI: 10.1016/0024-3795(89)90583-1. – ISSN 0024–3795

[KK05] Kehrein, Achim ; Kreuzer, Martin: Characterizations of border bases. In: *Journal of Pure and Applied Algebra* 196 (2005), Nr. 2, S. 251–270

[KMYZ08] Kaltofen, Erich ; May, John P. ; Yang, Zhengfeng ; Zhi, Lihong: Approximate factorization of multivariate polynomials using singular value decomposition. In: *Journal of Symbolic Computation* 43 (2008), Nr. 5, S. 359–376

[KR00] Kreuzer, Martin ; Robbiano, Lorenzo: *Computational Commutative Algebra 1*. Springer-Verlag Berlin Heidelberg, 2000. – ISBN 978–3–540–67733–8

[KR05] Kreuzer, Martin ; Robbiano, Lorenzo: *Computational Commutative Algebra 2*. Springer-Verlag Berlin Heidelberg, 2005. – ISBN 978–3–540–25527–7

[LAP13] LAPACK Team: *LAPACK – Linear Algebra PACKage*. http://www.netlib.org/lapack/, 2013. – zuletzt abgerufen: 04.03.2015

[Lau11] Laudahn, Moritz B.: *Ein Modell zur automatischen Bestimmung von Gelenkmittelpunkten anhand von 3D Motion Capture-Daten*, Universität Augsburg - Lehrstuhl für Multimedia Computing, Bachelorarbeit, 2011. http://www.multimedia-computing.de/mediawiki/images/c/cb/BA_MoritzLaudahn.pdf

[Lim13] Limbeck, Jan: *Computation of Approximate Border Bases and Applications*, Passau, Universität Passau, Dissertation, 2013

[Mac16] Macaulay, Francis S.: *The Algebraic Theory of Modular Systems*. Bd. 19. Cambridge Tracts in Mathematics and Mathematical Physics, Cambridge University Press, 1916

[Mal08] Mallat, Stephane: *A wavelet tour of signal processing: the sparse way*. Academic press, 2008. – ISBN 978–0–12–374370–1

[MB82] Möller, H. Michael ; Buchberger, Bruno: The construction of multivariate polynomials with preassigned zeros. In: *Goos, G., Hartmanis, J. (eds.) Computer Algebra, EUROCAM '82, European Computer Algebra Conference. Lecture Notes in Computer Science* 144 (1982), S. 24–31

[Moe76] Moenck, Robert T.: Practical fast polynomial multiplication. In: *Proceedings of the third ACM symposium on Symbolic and algebraic computation* ACM, 1976, S. 136–148

[MS00a] Möller, H. Michael ; Sauer, Thomas: H-bases for polynomial interpolation and system solving. In: *Advances in Computational Mathematics* 12 (2000), Nr. 4, S. 335–362

[MS00b] Möller, H. Michael ; Sauer, Thomas: H-bases I: the foundation. In: *Cohen, A., Rabut, C., Schumaker, L.L. (eds.) Curve and Surface Fitting: Saint–Malo 1999* (2000), S. 325–332

[MS00c] Möller, H. Michael ; Sauer, Thomas: H-bases II: applications to numerical problems. In: *Cohen, A., Rabut, C., Schumaker, L.L. (eds.) Curve and Surface Fitting: Saint-Malo 1999* (2000), S. 333–342

[MS07] Meinel, Kurt ; Schnabel, Günter: *Bewegungslehre - Sportmotorik: Abriss einer Theorie der sportlichen Motorik unter pädagogischem Aspekt.* Meyer & Meyer, 2007. – ISBN 978–3–89899–245–9

[OBBH00] O'Brien, James F. ; Bodenheimer, Robert E. ; Brostow, Gabriel J. ; Hodgins, Jessica K.: Automatic joint parameter estimation from magnetic motion capture data. In: *Graphics Interface* Bd. 2000, 2000, S. 53–60

[PS00] Peña, Juan M. ; Sauer, Thomas: On the multivariate Horner scheme. In: *SIAM journal on numerical analysis* (2000), S. 1186–1197

[PS07] Peña, Juan M. ; Sauer, Tomas: Efficient polynomial reduction. In: *Advances in Computational Mathematics* 26 (2007), Nr. 1, S. 323–336

[Sau01] Sauer, Thomas: Gröbner bases, H–bases and interpolation. In: *Transactions of the American Mathematical Society* 353 (2001), Nr. 6, S. 2293–2308

[Sau07] Sauer, Tomas: Approximate varieties, approximate ideals and dimension reduction. In: *Numerical Algorithms* 45 (2007), Nr. 1, S. 295–313

[Sau10] Sauer, Tomas: *Computeralgebra.* Vorlesungsskript, Justus–Liebig–Universität Gießen. http://www.fim.uni-passau.de/digitale-bildverarbeitung/lehre/. Version: Sommersemester 2010

[SK11] Schwarz, Hans R. ; Köckler, Norbert: *Numerische Mathematik*. Vieweg+Teubner Verlag, 2011. – ISBN 978–3–8348–1551–4

[Ste04] Stetter, Hans J.: *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 2004. – ISBN 978–0–89871–557–6

[Ste12] Steinmüller, Johannes: *Robotik*. Vorlesung an der TU Chemnitz. https://www.tu-chemnitz.de/informatik/KI/edu/robotik/ws2011/. Version: Wintersemester 2011/2012

[SW05] Schaback, Robert ; Wendland, Holger: *Numerische Mathematik*. Springer-Verlag Berlin Heidelberg, 2005 (Springer-Lehrbuch). – ISBN 978–3–540–21394–9

[TM14a] The MathWorks, Inc.: *Matlab*. http://de.mathworks.com/products/matlab/, 2014. – zuletzt abgerufen: 03.03.2015

[TM14b] The MathWorks, Inc.: *Symbolic Math Toolbox*. http://de.mathworks.com/products/symbolic/, 2014. – zuletzt abgerufen: 03.03.2015

[Tüm07] Tümmler, Jörn: *Avatare in Echtzeitsimulationen*, Universität Kassel, Dissertation, 2007

Stichwortverzeichnis

0-Norm 113, 128, 129
approximative **130**, 131
Minimierung 131–135, 190

A

Algebra-Geometry Dictionary . **64**
`approxHBasis` 189
Aufhängungspunkt . 137, **138**, 142
Auswertungsnorm **76**, 110, 171
Avatar 138

B

Basis 12, 40, 51
homogene ..... *siehe* H-Basis
Basissatz von Hilbert 52
Bewegungsprofil **142**, 144, 195
Bézierkurve 173–175
Bildraum **41**, 42
Bilinearität 108

C

Cell-Array **184**, 186, 189, 192, 193
Compressed Sensing 113, 131
Computeralgebrasystem .. 2, 8, 54
`convMat` 193
`convMatHom` 193

D

Datenausreißer 93, 168
Dicksons Lemma 51
Direkte Summe 10
Direkte Zerlegung **11**, 46, 127
Drehgelenk **139**, 140, 144
mit fester Drehebene 154

E

Effektor ..... *siehe* Manipulator
Einheitsrundungsfehler 24

F

Faltung 31
Faltungsmatrix ... **31**, 114–120, 122, 123, 193
homogene **39**, 40, 45, 193

Fast Fourier Transform 32

G

Ganzrationale Funktion 7
Gelenk
-analyse 138
-bedingung **141**, 144
-erkennung **142**, 153, 195
-position **138**, 139–141
-synthese 138
-typ **139**, 154, 160
Gerschgorin-Kreis 117
`GetVkWk` 192
Gleitkomma
-arithmetik 24
-darstellung 8
-zahlen 8
Gröbnerbasis 1, 125

H

Hadamard-Produkt 111
Halbordnung 15
Hauptachsentransformation 96
H-Basis 2, 48, **54**, 55–57, 188
approximative **77**, 78, 100, 189
dünn besetzte 62, **128**
reduzierte . **58**, 60, 61, 70, 189
H-Darstellung 56
Helix 163
Homogener Teilraum **12**, 192
`HornerEval` 187
`HornerEval2` 187
Horner-Schema 20, 22–25, 187, 188
Householder
-matrix 80
Transformation 78, 95, 99

I

Ideal **50**, 63
approximatives **76**, 91, 107
monisches 51
nulldimensionales .. **56**, 57, 59
radikales **53**, 63, 65
Verschwindungs- 63
Ideal-Membership-Problem . 54, 55
approximatives 125, 142
Implizite Darstellung 173
Interpolation 170
inv. kinematisches Problem ... 138

K

Kern ..... *siehe* Nullraum
Kinematische Kette 137
geschlossene 137
offene 137
planare 195
Koeffizientennorm **76**, 81, 107, 109
Koeffizientenvektor **9**, 14, 26
verschobener 30, 31
Kronecker-Trick **32**, 35, 37, 38
Kugelgelenk **152**, 153–155
Kugelkoordinaten 152

L

Leitform **11**, 40, 46, 54, 193

M

Manipulator 138
`minOnorm` 190
Monoid 15
-homomorphismus 9, 28

Monom 8, **9**, 23, 24, 109–111
Motion Capturing 152
Multiindex ... **8**, 10, 14, 17, 19, 20
Multiplikationsmatrix 31

N

Normalenvektor 154, 157
Normalform **56**, 125, 126
  approximative .. **126**, 127, 186
Normalformenraum 56
Nullraum **41**, 42, 158
Nullstellen 49, 62
  gemeinsame 49, 62, 67
Nullstellensatz von Hilbert 63

O

Ordnungsideal 51
Orthogonales Komplement 12
Orthonormale Polynome .. 81, 104

P

Pascal-Matrix .....
p-Norm 76
`polyAdd` 185
`polyDiv` 186
`polyMult` 185
`polyMultConv` 185
`polyMultFFT` 185
Polynom 8, **9**
  -addition 26, 185
  -auswertung 20, 111, 187
  -division ... **43**, 46, 48, 55, 186
    homogene 46
    univariate 43
  homogenes **10**, 11
  -multiplikation ... 29, 112, 185
  -ring **9**, 11, 26, **62**

Q

QRP-Zerlegung 78, 97, 99, 100
  abgebrochene 78
QR-Zerlegung 78, 99

R

Radikal 53
Reduktion 46
`refQR` 194
Rückwärtsfehler 25

S

Scharniergelenk 154
Schraubenlinie ..... *siehe* Helix
Schubgelenk ... **139**, 141–144, 147
  im Raum 152
Singulärvektor 42
Singulärwert **42**, 105, 114, 122
  kleinster 115–118, 123
Singulärwertzerlegung **42**, 104, 159
Skalarprodukt 11, 13
  monomiales **11**, 14, 108
  Standard- 11, 14
Skalierungsinvarianz 107, 129
Sparse Recovery 131
Spirale 161
Startpunkt **90**, 92, 176
Submultiplikativität 111
Syzygie 127

T

Term **9**, 10
  konstanter 118, 156, 175

-matrix . **16**, 17–20, 29, 35, 191
homogene 19, 191
-ordnung **15**, 119, 121
glex **15**, 121
grlex **121**
lex **15**, 122
rlex **15**, 119, 123
Totale Ordnung 15
Totalgrad . **10**, 44, 54, 61, 123, 184
Träger **112**, 147, 169, 171, 175

U

Urnenmodell 17

V

`VanderMat` 188
Vandermonde-Matrix **81**, 188
Varietät 1, **62**, 63, 65–67
Vektorraum 9, 11, 26
-isomorphismus 26
-operation 26, 27

W

Wohlordnung 15

Z

Zariski-Abschluss 66
Zeilenstufenform 99, 100, 194
Zelle 184
Zustand 142
Zweipunkteform 94

Selbstständigkeitserklärung

Ich erkläre: Ich habe die vorgelegte Dissertation selbstständig und ohne unerlaubte fremde Hilfe und nur mit den Hilfen angefertigt, die ich in der Dissertation angegeben habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen sind, und alle Angaben, die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Dissertation erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der „Satzung der Justus-Liebig-Universität Gießen zur Sicherung guter wissenschaftlicher Praxis“ niedergelegt sind, eingehalten.

Gießen, im Mai 2015