

Diplomarbeit

Analyse symbolischer Lösungsverfahren für kinematische Transformationen

Dominik Wagenführ

Justus-Liebig-Universität Gießen

Prof. Dr. Tomas Sauer

Mathematisches Institut der
Justus-Liebig-Universität Gießen

2. August 2005

Inhaltsverzeichnis

I	Einleitung	1
II	Mathematische Hintergründe	3
1	Einführung	3
2	Gröbner-Basen	6
2.1	Ordnungsrelationen und Gradbegriff	6
2.2	Divisionsalgorithmus	9
2.3	Hilbert-Basis-Theorem	12
2.4	Gröbner-Basen und S-Polynome	14
2.5	Buchbergers Algorithmus	18
3	H-Basen	24
3.1	Ein weiterer Gradbegriff	24
3.2	Lineare Räume	25
3.3	Divisionsalgorithmus	28
3.4	H-Basen	30
4	Lösung von polynomialen Gleichungen	33
4.1	Dimension eines Ideals	33
4.2	Normalformraum	33
4.3	Multiplikationstabellen	36
4.4	Berechnung der Varietät	38
III	Algorithmen und Analyse	42
1	Einführung	42
2	Gröbner-Basen	43
2.1	Divisionsalgorithmus	43
2.2	Buchbergers Algorithmus	44
2.3	Reduktionsalgorithmus	46
3	Berechnung der linearen Räume	51
3.1	Der Raum P_d	51
3.2	Der Raum $V_d(I)$	52
3.3	Skalarprodukt von Polynomen	56
3.4	Der Raum $W_d(I)$	59

4	Berechnung der H-Basis	62
4.1	H-Basis	62
4.2	Syzygienmodul	63
4.3	Divisionsalgorithmus	63
4.4	Laufzeit	65
5	Lösungsverfahren	67
5.1	Normalformraum	67
5.2	Multiplikationstabellen	68
5.3	Berechnung der Varietät	70
IV	Anwendungen	74
1	Einführung	74
2	2D-Kinematik 1 - Spezialfall	75
2.1	Konstruktion	75
2.2	Idealbestimmung	75
2.3	Inverse kinematische Transformation	76
2.4	Kinematische Transformation	76
3	2D-Kinematik 1 - Allgemein	78
3.1	Konstruktion	78
3.2	Idealbestimmung	78
3.3	Inverse kinematische Transformation	79
3.4	Kinematische Transformation	79
4	2D-Kinematik 2	80
4.1	Konstruktion	80
4.2	Idealbestimmung	80
4.3	Inverse kinematische Transformation	82
4.4	Kinematische Transformation	83
4.5	Trigonometrischer Ansatz	85
5	3D-Kinematik 1 - Spezialfall	92
5.1	Konstruktion	92
5.2	Idealbestimmung	92
5.3	Inverse kinematische Transformation	93
5.4	Kinematische Transformation	94
6	3D-Kinematik 1 - Allgemein	95
6.1	Konstruktion	95
6.2	Idealbestimmung	95

6.3	Inverse kinematische Transformation	95
6.4	Kinematische Transformation	95
7	3D-Kinematik 2	97
7.1	Konstruktion	97
7.2	Idealbestimmung	97
7.3	Inverse kinematische Transformation	101
7.4	Kinematische Transformation	101
7.5	Trigonometrischer Ansatz	103
V	Zusammenfassung	110
VI	Anhang	112
1	Grafische Darstellung der Kinematiken in Maple	112
2	Ordnungsrelationen in Maple	116
3	Lösung der allgemeinen 2D-Kinematik 1	119
4	Lösung der speziellen 2D-Kinematik 2	121
5	Lösung der speziellen 3D-Kinematik 2	122
6	Befehlsverzeichnis	129
	Literatur	138

Teil I

Einleitung

Bei vielen Fragestellungen in Technik, Mathematik oder Physik hat man es mit polynomialen Gleichungen zu tun, die gelöst werden müssen. So auch in der Robotertechnik, in der man Kinematiken (sich bewegende Systeme) hat, deren Konfiguration man bestimmen will. Die Siemens AG entwickelt solche parallelen Kinematiken für Werkzeugmaschinen, die numerisch gesteuert werden. Dabei muß man mit Hilfe der Koordinatentransformation die Maschinenkoordinaten in Umweltkoordinaten umrechnen, um die Kraft zu berechnen, die notwendig ist, um die Maschine an einen bestimmten Punkt zu bewegen. Für diese Kraftberechnung benötigt man die exakte Lösung der Transformationen.

Hier könnte man nun auf numerische Methoden zurückgreifen, die Homotopien oder die Newton-Methoden benutzen, wie zum Beispiel in *Solving polynomial systems using continuation for Engineering and Scientific Problems* von A. Morgan (1987) oder in *Iterative solution of nonlinear equations in several variables* von J.M.Ortega und W.C.Rheinboldt (1970) beschrieben. Neben einer numerischen Lösung hat sich im Laufe der Zeit durch die Computer Algebra Systeme (CAS) eine neue Methode etabliert: Man betrachtet anstelle der Nullstellen der Polynomgleichungen die Varietät (Nullstellenmenge) des zugehörigen Ideals. Für diese Idee gibt es schon seit langer Zeit gute Ansätze, die auf Gröbner-Basen basieren; eine Idealbasis mit besonders nützlichen Eigenschaften. Der Nachteil von Gröbner-Basen, die 1965 von Buchberger "erfunden" wurden, ist, daß diese bei multivariaten Polynomen in mehreren Unbekannten grundsätzlich eine Variable etwas bevorzugen, was oft das Gleichgewicht und die Symmetrien in einem Problem stört. Hier setzen nun H-Basen an, die schon von Macaulay (1916) erwähnt wurden, auch wenn es sich bei ihm damals in Wirklichkeit nur um spezielle Gröbner-Basen gehandelt hat. Diese behandeln alle Unbekannten gleich und beachten nur den Totalgrad eines Polynoms. Dadurch werden Symmetrien nicht gestört, zusätzlich behält diese Basis aber die guten Eigenschaften einer Gröbner-Basis bei.

Wir wollen in dieser Arbeit Gröbner-Basen und H-Basen einführen und erstmals Algorithmen für H-Basen in einer Programmiersprache implementieren und vorstellen. Hierfür behandeln wir zunächst im zweiten Teil die mathematischen Grundlagen, um dann im dritten Teil die Algorithmen in Pseudocode vorstellen zu können. Dabei wurde bei der Programmierung auf das CAS `Maple` von Maplesoft zurückgegriffen, da dieses durch seine

vielen Routinen und Funktionen und dem Pascal-ähnlichen Programmierstil leicht zu bedienen ist. Aufgrund einiger Probleme wurde zusätzlich das CAS **Singular** der Universität Kaiserslautern verwendet, um spezielle Ergebnisse berechnen und in **Maple** verwenden zu können.

Der vierte und letzte Teil dieser Arbeit zeigt die Anwendung von H-Basen und der Algorithmen an besonderen Kinematiken, für welche die Siemens AG numerische Steuerungen entwickelt. Es hat sich aber im Laufe der Berechnungen herausgestellt, daß H-Basen - oder auch Gröbner-Basen - leider keine adäquate Leistung bei solch komplexen Konstruktionen bringen können. Aufgrund der sehr hohen Anzahl von Unbekannten und des exponentiellen Wachstums der Berechnungszeit können die Ergebnisse oft nur nach längerer Zeit oder - im schlimmsten Fall - gar nicht berechnet werden. Desweiteren stören oft Symmetrien, wie Spiegelungen, die Berechnungen, so daß man zwar korrekte mathematische Ergebnisse für ein Problem erhält, diese technisch aber nicht mehr verwertbar sind.

Dennoch sind die Methoden und Algorithmen für "kleinere" Probleme sehr gut zu verwenden und sollten auch in Zukunft eine Rolle bei der Lösung polynomialer Gleichungssysteme spielen.

Teil II

Mathematische Hintergründe

1 Einführung

Bevor wir in die höhere Mathematik einsteigen, definieren wir einige Begriffe, die im Laufe der weiteren Kapitel mehrfach benutzt werden.

Definition 1.1. Ein *Ring* ist eine Menge R , auf der zwei Operationen $+$ und \cdot mit den folgenden Eigenschaften definiert sind¹:

- Für alle $a, b \in R$ gilt $a + b = b + a$.
- Es existiert ein Nullelement in R , das heißt, es existiert $0 \in R$, so daß $0 + a = a + 0 = a$. Dieses Element wird auch *neutrales Element* bezüglich Addition genannt.
- Für alle $a \in R$ existiert ein $b \in R$ mit $a + b = 0$.
- Für alle $a, b, c \in R$ gilt $(a + b) + c = a + (b + c)$ und $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- Für alle $a, b, c \in R$ gilt $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ und $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Gilt zusätzlich $a \cdot b = b \cdot a$ für alle $a, b \in R$, so heißt der Ring R *kommutativ* beziehungsweise *abelsch*. Existiert ein Element $1 \in R$ mit $a \cdot 1 = 1 \cdot a = a$ für alle $a \in R$, so heißt dieses Element *Einselement* beziehungsweise *neutrales Element* bezüglich Multiplikation. R heißt dann *unitär*. Gibt es außerdem keine $a, b \in R$ mit $a \neq 0 \neq b$ und $a \cdot b = 0$, so heißt R *nullteilerfrei*.

Im weiteren werden wir nur noch mit kommutativen, nullteilerfreien, unitären Ringen arbeiten, weswegen wir hierzu im folgenden nur noch Ring sagen werden.

Definition 1.2. Ist die Teilmenge U eines Ringes R wieder ein Ring, so heißt U ein *Unterring* von R . Ein Unterring I heißt (zweiseitiges) *Ideal* von R , falls gilt:

$$f \in R \text{ und } g \in I \Rightarrow f \cdot g = g \cdot f \in I.$$

¹In der Regel sind dies Addition und Multiplikation.

Beispiel. Die Menge aller geraden Zahlen ist ein Ideal von \mathbb{Z} .

Definition 1.3. Erzeugt die Menge $F \subset R$ ein Ideal I , das heißt, ist $F = \{f_1, \dots, f_r\}$ und

$$I = \left\{ \sum_{i=1}^r f_i r_i \mid r_i \in R \text{ beliebig} \right\},$$

so heißt F *Idealbasis* von I . Man schreibt dann $I = \langle F \rangle$.

Bemerkung. Die Bezeichnung *Basis* sollte man hier nicht mit der herkömmlichen Bezeichnung einer Vektorraumbasis verwechseln. So muß eine Idealbasis weder minimal sein, noch läßt sich von der Anzahl der Basiselemente auf die Dimension des Ideals schließen.

Beispiel. Die Menge $F = \{2\} \subset \mathbb{Z}$ erzeugt die Menge der geraden Zahlen, da

$$\begin{aligned} I &= \{2 \cdot r \mid r \in \mathbb{Z} \text{ beliebig}\} \\ &= \{\dots, 2 \cdot (-2), 2 \cdot (-1), 2 \cdot 0, 2 \cdot 1, 2 \cdot 2, \dots\} \\ &= \{\dots, -4, -2, 0, 2, 4, \dots\}. \end{aligned}$$

Definition 1.4. Sei R ein Ring. Die Menge $\mathbb{P} := R[x_1, \dots, x_n]$ heißt *Polynomring* über R in den Unbekannten x_1, \dots, x_n . Die Elemente in \mathbb{P} heißen *Polynome*. Ein Polynom p ist eine endliche Linearkombination in R von *Monomen* a , die sich wiederum schreiben lassen als

$$a = x^\alpha$$

mit $x = (x_1, \dots, x_n)$ und $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$. Man nennt α *Multiindex*. Das bedeutet nun

$$\begin{aligned} p &= \sum_{j=1}^m c_j x^{\alpha_j} \\ &= \sum_{j=1}^m c_j (x_1, \dots, x_n)^{(\alpha_{1,j}, \dots, \alpha_{n,j})} \\ &= \sum_{j=1}^m c_j x_1^{\alpha_{1,j}} \dots x_n^{\alpha_{n,j}}, c_j \in R \end{aligned}$$

Bemerkung. Wie der Name *Polynomring* schon sagt, ist \mathbb{P} ebenfalls ein Ring.

Beispiele.

1. Das (univariate) Polynom $p = x^2 + x + 1 \in \mathbb{Z}[x]$ ist bereits in obiger Darstellung.
2. Das (multivariate) Polynom $p = 2x_1^2 - x_1x_2^2 + 3x_2 - 1 \in \mathbb{Z}[x_1, x_2]$ läßt sich auch schreiben als

$$p = 2x^{(2,0)} - x^{(1,2)} + 3x^{(0,1)} - x^{(0,0)}$$

mit $x = (x_1, x_2)$.

Definition 1.5. Sei $F = \{f_1, \dots, f_s\} \subset \mathbb{P}$. Die Menge der gemeinsamen Nullstellen von F

$$V(F) = \{z \in R^n \mid f_i(z) = 0 \forall i = 1, \dots, s\}$$

bezeichnet man als *affine Varietät* von F .

Satz 1.6. Sind F und G zwei Idealbasen mit $\langle F \rangle = \langle G \rangle$, dann ist $V(F) = V(G)$.

Beweis: Dies folgt aus der Definition der Varietät. Ist $z \in V(F)$, dann ist $f(z) = 0$ für alle $f \in \langle F \rangle = \langle G \rangle$. Daraus folgt $z \in V(G)$. Analog gilt die Umkehrung. \square

Dies bedeutet, daß die Varietät, also die Menge der gemeinsamen Nullstellen, nur vom Ideal selbst abhängt, aber nicht von einer bestimmten Idealbasis. Dies ist die Grundlage für die Verwendung von Gröbner- und H-Basen zur Lösung von polynomialen Gleichungssystemen.

2 Gröbner-Basen

2.1 Ordnungsrelationen und Gradbegriff

Essentiell für Gröbner-Basen sind Ordnungen. Im Falle von Polynomen in einer Unbekannten besteht hier kein Problem, da man einfach den normalen Gradbegriff verwenden kann und die Monome dementsprechend ordnet.

Nur was macht man bei multivariaten Polynomen? Wie sieht es zum Beispiel mit den Monomen x^2y und xy^2 aus? Welches von beiden ist größer?

Da wir jedes Monom a als

$$a = x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$$

mit $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$ schreiben können, erhalten wir eine "Eins-zu-Eins"-Beziehung zwischen \mathbb{P} und \mathbb{N}_0^n , das heißt,

$$x^\alpha \succ x^\beta \Leftrightarrow \alpha \succ \beta.$$

Definition 2.1. Eine *monomiale Ordnung* auf \mathbb{P} ist eine Relation \succ auf \mathbb{N}_0^n , die folgende Eigenschaften erfüllt:

1. \succ ist eine totale Ordnung auf \mathbb{N}_0^n .
2. Ist $\alpha \succ \beta$ und $\gamma \in \mathbb{N}_0^n$, dann gilt $\alpha + \gamma \succ \beta + \gamma$.
3. \succ ist wohlgeordnet auf \mathbb{N}_0^n , das heißt, jede nicht-leere Teilmenge von \mathbb{N}_0^n hat ein kleinstes Element bezüglich \succ .

Die dritte Bedingung wird häufig genutzt, um die Terminierung eines Algorithmus zu zeigen.

Definition 2.2. (Lexikographische Ordnung, engl. *Lex Order*)

Sei $\alpha = (\alpha_1, \dots, \alpha_n), \beta = (\beta_1, \dots, \beta_n) \in \mathbb{N}_0^n$. Wir sagen $\alpha >_{lex} \beta$, falls in der Vektordifferenz $\alpha - \beta \in \mathbb{Z}^n$ gilt:

$$(\alpha - \beta)_{\min_{i=1}^n \{i \mid \alpha_i - \beta_i \neq 0\}} > 0,$$

oder mit anderen Worten, wenn der am weitesten links-stehende, von Null verschiedene Eintrag positiv ist.

Beispiele.

1. $(2, 5, 2) >_{lex} (2, 1, 4)$, da $\alpha - \beta = (0, 4, -2)$.
2. $(2, 0, 2) >_{lex} (0, 3, 1)$, da $\alpha - \beta = (2, -3, 1)$.

Definition 2.3. (Graduiert lexikographische Ordnung, engl. *Graded Lex Order*)

Sei $\alpha, \beta \in \mathbb{N}_0^n$. Dann heißt $\alpha >_{grlex} \beta$, falls

$$\sum_{i=1}^n \alpha_i = |\alpha| > |\beta| = \sum_{i=1}^n \beta_i, \quad \text{oder} \quad |\alpha| = |\beta| \text{ und } \alpha >_{lex} \beta.$$

Beispiele.

1. $(2, 5, 2) >_{grlex} (2, 0, 4)$, da $|\alpha| = 9 > |\beta| = 6$.
2. $(2, 0, 2) >_{grlex} (0, 3, 1)$, da $|\alpha| = |\beta| = 4$ und $\alpha >_{lex} \beta$ (siehe oben).

Eine nicht gleich ersichtliche Ordnungsdefinition ist die *Graded Reverse Lex Order*, die in vielen Anwendungen genutzt wird, da sich gezeigt hat, daß sich eine Basis bezüglich dieser Ordnung oft am effizientesten berechnen läßt.

Definition 2.4. (Graduiert umgekehrt lexikographische Ordnung, engl. *Graded Reverse Lex Order*)

Sei $\alpha, \beta \in \mathbb{N}_0^n$. Dann heißt $\alpha >_{grevlex} \beta$, falls

$$|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i,$$

oder $|\alpha| = |\beta|$ und in $\alpha - \beta \in \mathbb{Z}^n$ gilt:

$$(\alpha - \beta)_{\max_{i=1}^n \{i \mid \alpha_i - \beta_i \neq 0\}} < 0,$$

oder mit anderen Worten, wenn der am weitesten rechts-stehende, von Null verschiedene Eintrag negativ ist.

Beispiele.

1. $(2, 5, 2) >_{\text{grevlex}} (2, 1, 3)$, da $|\alpha| = 9 > |\beta| = 6$.
2. $(0, 3, 1) >_{\text{grevlex}} (2, 0, 2)$, da $|\alpha| = |\beta| = 4$ und $\alpha - \beta = (-2, 3, -1)$.

Bemerkung. Es ist leicht zu zeigen, daß diese drei Ordnungen tatsächlich monomiale Ordnungen nach obiger Definition sind.

Abhängig von einer festgelegten monomialen Ordnung wollen wir nun Begriffe wie *Grad* und *Leitkoeffizient* definieren.

Definition 2.5. Sei $0 \neq f = \sum_{\alpha \in \mathbb{N}_0^n} a_\alpha x^\alpha$ ein Polynom in \mathbb{P} und sei $>$ eine monomiale Ordnung.

1. Der *Grad* von f ist

$$\deg(f) = \max\{\alpha \in \mathbb{N}_0^n \mid a_\alpha \neq 0\},$$

wobei das Maximum bezüglich $>$ gemeint ist.

2. Der *Leitkoeffizient* von f ist

$$LK(f) = a_{\deg(f)} \in R.$$

3. Das *Leitmonom* von f ist

$$LM(f) = x^{\deg(f)} \quad (\text{mit Koeffizient } 1).$$

4. Der *Leitterm* von f ist

$$LT(f) = LK(f) \cdot LM(f).$$

Beispiel. Sei $f = -y^4z + 6xy - 5x^4y^2z + x^5 + 4y^3z^2 + y^2 - x^3y^4 \in \mathbb{Z}[x, y, z]$. Das Polynom ist nun wie folgt geordnet:

$$\begin{array}{ll} \text{lex:} & f = x^5 - 5x^4y^2z - x^3y^4 + 6xy - y^4z + 4y^3z^2 + y^2 \quad \text{mit } \deg(f) = (5, 0, 0), \\ \text{grlex:} & f = -5x^4y^2z - x^3y^4 + x^5 - y^4z + 4y^3z^2 + 6xy + y^2 \quad \text{mit } \deg(f) = (4, 2, 1), \\ \text{grevlex:} & f = -x^3y^4 - 5x^4y^2z + x^5 + 4y^3z^2 - y^4z + 6xy + y^2 \quad \text{mit } \deg(f) = (3, 4, 0). \end{array}$$

Lemma 2.6. (ohne Beweis) Seien $f, g \in \mathbb{P}$ mit $f \neq 0 \neq g$. Dann gilt:

1. $\deg(fg) = \deg(f) + \deg(g)$.
2. Ist $f + g \neq 0$, dann ist $\deg(f + g) \leq \max\{\deg(f), \deg(g)\}$.
Gleichheit folgt, wenn $\deg(f) \neq \deg(g)$.

2.2 Divisionsalgorithmus

Wir sind nun auf der Suche nach einem Algorithmus, der uns zu gegebenem $f \in \mathbb{P}$ und $F = \{f_1, \dots, f_s\} \subset \mathbb{P}$ die Darstellung

$$f = a_1 f_1 + \dots + a_s f_s + r,$$

liefert, wobei die Quotienten a_1, \dots, a_s und der Rest r wieder in \mathbb{P} liegen. Außerdem soll gelten, falls $a_i f_i \neq 0$:

$$\deg(f) \geq \deg(a_i f_i).$$

Theorem 2.7. (Divisionsalgorithmus in \mathbb{P})

Sei eine monomiale Ordnung $>$ auf \mathbb{N}_0^n festgelegt und sei $F = (f_1, \dots, f_s)$ ein geordnetes s -Tupel von Polynomen in \mathbb{P} . Dann kann jedes $f \in \mathbb{P}$ geschrieben werden als

$$f = a_1 f_1 + \dots + a_s f_s + r,$$

wobei $a_1, \dots, a_s, r \in \mathbb{P}$ und entweder ist $r = 0$ oder r ist ein Polynom, dessen Monome durch keines der $LM(f_1), \dots, LM(f_s)$ teilbar sind. Wir bezeichnen r als *Rest* von f bei Division durch F .

Beweis: Wir beweisen die Existenz von a_1, \dots, a_s und r , indem wir einen Algorithmus angeben, um diese zu berechnen.

Algorithmus 2.8 nach Cox, Little und O'Shea [5, S.61 ff]

1. Setze $a_1 := 0, \dots, a_s := 0, r := 0$ und $p := f$.
2. Solange $p \neq 0$:
 - Setze $i := 1$ und $\text{divisionaufgetreten} := \text{falsch}$.

- Solange $i \leq s$ und divisionaufgetreten = falsch:
 - Wenn $LM(f_i) \mid LM(p)$, dann setze

$$a_i := a_i + LT(p)/LT(f_i),$$

$$p := p - (LT(p)/LT(f_i))f_i,$$
 divisionaufgetreten := wahr.
 - Sonst setze $i := i + 1$.
- Wenn divisionaufgetreten = falsch, dann setze $r := r + LT(p)$ und $p := p - LT(p)$.

Als erstes bemerken wir, daß bei jedem Durchlauf der äußeren Schleife einer der beiden Fälle eintritt:

- (Divisionsschritt) Falls $LM(p)$ von einem der $LM(f_i)$ geteilt wird, dann wird der Algorithmus wie im univariaten Fall fortgeführt, das heißt, der Quotient $LT(p)/LT(f_i)$ wird zu den a_i hinzugefügt.
- (Restbestimmungsschritt) Falls $LM(p)$ von keinem der $LM(f_i)$ geteilt wird, dann addiert der Algorithmus $LT(p)$ zum Rest r dazu.

Um die Korrektheit des Algorithmus zu beweisen, zeigen wir zuerst, daß die Gleichung

$$f = a_1 f_1 + \dots + a_s f_s + p + r \tag{1}$$

zu jeder Zeit des Algorithmus erfüllt ist. Dies ist natürlich für die Startwerte von a_1, \dots, a_s, p und r korrekt. Befinden wir uns im Divisionsschritt, dann wird $LM(f_i)$ durch $LM(p)$ geteilt und die Gleichung

$$a_i f_i + p = (a_i + LT(p)/LT(f_i))f_i + (p - (LT(p)/LT(f_i))f_i)$$

zeigt, daß $a_i f_i + p$ unverändert bleibt. Da alle anderen Variablen unberührt sind, ist (1) somit erfüllt. Falls wir uns im Restbestimmungsschritt befinden, werden nur p und r verändert, die Summe $p + r$ aber bleibt gleich, da

$$p + r = (p - LT(p)) + (r + LT(p)).$$

Wiederum ist Gleichung (1) erfüllt.

Der Algorithmus endet, sobald $p = 0$ ist. Dann wird (1) zu

$$f = a_1 f_1 + \dots + a_s f_s + r.$$

Da nur Terme zu r addiert werden, die durch keines der $LM(f_i)$ teilbar sind, erhalten wir für a_i, \dots, a_s und r die gewünschten Eigenschaften, falls der Algorithmus terminiert.

Der Abbruch des Algorithmus ist leicht nachzuvollziehen. Durch $p' = p - LT(p)$, wobei p' das neue p ist, ergibt sich also, daß p' einen geringeren Grad hat, also $\deg(p') < \deg(p)$, $p' \neq 0$. Da $<$ nach Definition 2.1 wohlgeordnet ist, muß nach endlich vielen Schritten $p = 0$ gelten.

Jeder Term in a_i ist von der Form $LT(p)/LT(f_i)$ für einen beliebigen Wert der Variable p . Der Algorithmus beginnt mit $p = f$ und wie wir eben gezeigt haben, ergibt sich daraus, daß $LT(p) \leq LT(f)$. Hieraus folgt leicht $\deg(a_i f_i) \leq \deg(f)$, falls $a_i f_i \neq 0$. \square

Beispiele.

Seien $f = x^3 y - x^2 y^2$, $f_1 = x^2 - y$, $f_2 = x^3 - y^2 \in \mathbb{Z}[x, y]$ und als monomiale Ordnung wählen wir die Relation $>_{lex}$.

1. Setze $a_1 = a_2 = r = 0$ und $p = f = x^3 y - x^2 y^2$.
2. $LM(f_1) = x^2 \mid x^3 y = LM(p) \Rightarrow a_1 := xy$, $p := -x^2 y^2 + xy^2$
3. $LM(f_1) = x^2 \mid x^2 y^2 = LM(p) \Rightarrow a_1 := xy - y^2$, $p := xy^2 - y^3$
4. $LM(f_1) = x^2 \nmid xy^2 = LM(p) \Rightarrow$ weiter mit f_2
5. $LM(f_2) = x^3 \nmid xy^2 = LM(p) \Rightarrow r := xy^2$, $p := -y^3$
6. $LM(f_1) = x^2 \nmid y^3 = LM(p) \Rightarrow$ weiter mit f_2
7. $LM(f_2) = x^3 \nmid y^3 = LM(p) \Rightarrow r := xy^2 - y^3$, $p := 0$

Ergebnis: $f = a_1 f_1 + a_2 f_2 + r$ beziehungsweise $x^3 y - x^2 y^2 = (xy - y^2)(x^2 - y) + 0 \cdot (x^3 - y^2) + (xy^2 - y^3)$.

Wir wissen, daß ein Polynom f in einem Ideal I liegt, wenn der Divisionsalgorithmus den Rest 0 liefert. Aber was ist mit der Umkehrung? Sehen wir uns daher die gleiche Aufgabe

an und vertauschen jetzt dabei die Rollen von f_1 und f_2 .²

Seien also wiederum $f = x^3y - x^2y^2$, aber $f_1 = x^3 - y^2, f_2 = x^2 - y$.

1. Setze $a_1 = a_2 = r = 0$ und $p = f = x^3y - x^2y^2$.
2. $LM(f_1) = x^3 \mid x^3y = LM(p) \Rightarrow a_1 := y, p := -x^2y^2 + y^3$
3. $LM(f_1) = x^3 \nmid x^2y^2 = LM(p) \Rightarrow$ weiter mit f_2
4. $LM(f_2) = x^2 \mid x^2y^2 = LM(p) \Rightarrow a_2 := -y^2, p := 0$

Ergebnis: $f = a_1f_1 + a_2f_2 + r$ beziehungsweise $x^3y - x^2y^2 = y(x^3 - y^2) - y^2(x^2 - y)$.

Man sieht also, daß die Endwerte für a_i und r stark von der Wahl und der Reihenfolge der f_i abhängen. Wenn man eine "gute" Basis hat, erreicht man sein gewünschtes Ziel, daß der Rest unabhängig von der Reihenfolge der f_i ist. Wir werden sehen, daß Gröbner-Basen zu solchen "guten" Basen gehören. Dementsprechend würde dann auch die Umkehrung gelten: Wenn ein Polynom p in einem von F erzeugtem Ideal liegt, muß der Rest 0 sein.

2.3 Hilbert-Basis-Theorem

Bevor wir wirklich zu den Gröbner-Basen kommen, müssen wir noch einiges an Vorarbeit leisten, was Monome und Ideale angeht.

Definition 2.9. Ein Ideal $I \subset \mathbb{P}$ heißt *monomiales Ideal*, wenn es eine Teilmenge $A \subset \mathbb{N}_0^n$ gibt, so daß I aus allen Polynomen besteht, die als endliche Summen der Form $\sum_{\alpha \in A} h_\alpha x^\alpha$ geschrieben werden können, wobei $h_\alpha \in \mathbb{P}$. In diesem Fall schreiben wir $I = \langle x^\alpha \mid \alpha \in A \rangle$.

Ein Beispiel hierfür ist $I = \langle x^3y^2, x^2y^4, xy \rangle \subset \mathbb{Z}[x, y]$.

Lemma 2.10. Sei $I = \langle x^\alpha \mid \alpha \in A \rangle$ ein monomiales Ideal. Dann liegt ein Monom x^β genau dann in I , wenn x^β mindestens durch ein $x^\alpha, \alpha \in A$, teilbar ist.

Beweis: Dies ist leicht zu sehen. Gilt $x^\alpha \mid x^\beta$, dann folgt $x^\beta = h_\alpha x^\alpha \in I$ nach Definition, $h_\alpha \in \mathbb{P}$. Ist umgekehrt $0 \neq x^\beta \in I$, so läßt sich dies als $x^\beta = \sum_{\alpha \in \mathbb{N}_0^n} h_\alpha x^\alpha$ schreiben, $h_\alpha \in \mathbb{P}$.

²Man kann sich sicher sein, daß das Ideal immer noch das gleiche ist, welches aus den Polynomen erzeugt wird.

Multipliziert man diese Summe aus und sortiert nach den x_α neu, so sieht man, daß auf der rechten Seite alle Terme durch bestimmte x_α teilbar sind. Dies gilt dann natürlich auch für die linke Seite. \square

Bemerkung. Eine wichtige Eigenschaft von monomialen Idealen ist, daß sie grundsätzlich endlich erzeugt werden (auch bekannt als *Dicksons Lemma*, siehe Cox, Little und O’Shea [5, S.69 f], welches eine Vereinfachung des Hilbert-Theorems ist).

Definition 2.11. Sei $I \subset \mathbb{P}$ ein Ideal verschieden von $\{0\}$.

1. Wir bezeichnen mit $LT(I)$ die Menge der Leitterme der Elemente aus I , das heißt,

$$LT(I) = \{cx^\alpha \mid \text{ex. } f \in I \text{ mit } LT(f) = cx^\alpha\}.$$

2. Wir bezeichnen mit $\langle LT(I) \rangle$ das Ideal, welches aus den Elementen von $LT(I)$ erzeugt wird.

Bemerkung. Es ist klar, daß aus $I = \langle f_1, \dots, f_s \rangle$ nur die Inklusion $\langle LT(f_1), \dots, LT(f_s) \rangle \subseteq \langle LT(I) \rangle$ folgt. Für $I = \langle f_1, f_2 \rangle$ mit $f_1 = x^2 - y$, $f_2 = x^3 - y \in \mathbb{Z}[x, y]$ und *Lex Order* ergibt sich zum Beispiel

$$x(x^2 - y) - (x^3 - y) = xy + y \in I \Rightarrow xy \in \langle LT(I) \rangle.$$

Aber es ist leicht zu sehen, daß $xy \notin \langle x^2, x^3 \rangle = \langle LT(f_1), LT(f_2) \rangle$.

Satz 2.12. Sei $I \subset \mathbb{P}$ ein Ideal. Dann gilt:

1. $\langle LT(I) \rangle$ ist ein monomiales Ideal.
2. Es existieren $g_1, \dots, g_t \in I$, so daß $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$.

Beweis:

1. Die Leitmonome von Elementen aus I bilden sicher ein monomiales Ideal. Da sich diese von den Leittermen aber nur durch eine von Null verschiedene Konstante unterscheiden, gilt $\langle LM(g) \mid g \in I \rangle = \langle LT(g) \mid g \in I \rangle$ und damit ist dies auch ein monomiales Ideal.

2. Da $\langle LT(I) \rangle = \langle LM(g) \mid g \in I \rangle$ und sich monomiale Ideale nach Dicksons Lemma endlich erzeugen lassen, gilt $\langle LT(I) \rangle = \langle LM(g_i) \mid i = 1, \dots, s \rangle$. Aus den gleichen Gründen wie in

Punkt 1 erhalten wir so die gewünschte Aussage $\langle LT(I) \rangle = \langle LT(g_i) \mid i = 1, \dots, s \rangle$. \square

Damit kommen wir auch schon zur Kernaussage dieses Abschnitts, daß sich jedes Polynomideal als endliches Erzeugnis schreiben läßt.

Theorem 2.13. (Hilbert-Basis-Theorem)

Jedes Ideal $I \subset \mathbb{P}$ hat ein endliches Erzeugnis, das heißt, $I = \langle g_1, \dots, g_t \rangle$ mit $g_1, \dots, g_t \in I$.

Beweis: Falls $I = \{0\}$, nehmen wir als Erzeugnis $\{0\}$, was sicher endlich ist. Enthält I ein Polynom ungleich Null, kann man ein endliches Erzeugnis g_1, \dots, g_t wie folgt konstruieren.

Nach obigem Satz 2.12 existieren $g_1, \dots, g_t \in I$, so daß $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$. Wir behaupten nun $I = \langle g_1, \dots, g_t \rangle$.

Es ist klar, daß $\langle g_1, \dots, g_t \rangle \subseteq I$, da $g_i \in I$. Sei $f \in I$ ein beliebiges Polynom. Mit Hilfe des Divisionsalgorithmus aus Kapitel 2.2 erhalten wir

$$f = a_1 g_1 + \dots + a_t g_t + r,$$

wobei kein Term von r durch eines der $LT(g_i)$ teilbar ist. Wir müssen zeigen, daß $r = 0$. Es ist

$$r = f - a_1 g_1 - \dots - a_t g_t \in I.$$

Wäre $r \neq 0$, dann gilt $LT(r) \in \langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$ und nach Lemma 2.10 muß $LT(r)$ durch eines der $LT(g_i)$ teilbar sein. Dies ist ein Widerspruch zur Annahme. Das zeigt

$$f = a_1 g_1 + \dots + a_t g_t + 0 \in \langle g_1, \dots, g_t \rangle,$$

also $I \subseteq \langle g_1, \dots, g_t \rangle$ und damit $I = \langle g_1, \dots, g_t \rangle$. \square

2.4 Gröbner-Basen und S-Polynome

Die Basis im Hilbert-Basis-Theorem hat die besondere Eigenschaft, daß $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$. Da dies nicht für alle Basen eines Ideals gilt, wollen wir dieser Basis einen speziellen Namen geben.

Definition 2.14. Es sei eine monomiale Ordnung festgelegt. Eine endliche Teilmenge $G = \{g_1, \dots, g_t\}$ eines Ideals I heißt genau dann *Gröbner-Basis*, falls gilt:

$$\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle.$$

Äquivalent dazu ist die Aussage, daß eine Menge $\{g_1, \dots, g_t\} \subseteq I$ genau dann eine Gröbner-Basis ist, wenn der Leitterm jedes Elements von I durch mindestens eines der $LT(g_i)$ teilbar ist.

Folgerung 2.15. Es sei eine monomiale Ordnung festgelegt. Dann hat jedes von Null verschiedene Ideal I eine Gröbner-Basis.

Beweis: Dies folgt sofort aus der Konstruktion von Theorem 2.13. Außerdem ist jede Gröbner-Basis von I auch eine Idealbasis von I . \square

Wir wollen nun die Auswirkung einer Gröbner-Basis auf unseren Divisionalgorithmus aus Kapitel 2.2 betrachten.

Satz 2.16. Sei $G = \{g_1, \dots, g_t\}$ eine Gröbner-Basis eines Ideals $I \subset \mathbb{P}$ und sei $f \in \mathbb{P}$. Dann gibt es einen eindeutigen Rest $r \in \mathbb{P}$, für den gilt:

1. Kein Term von r ist durch eines der $LT(g_i)$ teilbar.
2. Es existiert ein $g \in I$, so daß $f = g + r$.

Vor allem ist r unabhängig von der Reihenfolge der Basiselemente, durch die im Divisionsalgorithmus geteilt wird.

Beweis: Der Divisionsalgorithmus gibt uns $f = a_1g_1 + \dots + a_tg_t + r$, wobei r Bedingung 1 automatisch erfüllt. Bedingung 2 können wir erfüllen, indem wir $g = a_1g_1 + \dots + a_tg_t \in I$ setzen. Dies beweist die Existenz von r .

Für die Eindeutigkeit nehmen wir an, daß $f = g+r = g'+r'$ und daß r und r' Bedingungen 1 erfüllen. Dann ist $r - r' = g - g' \in I$, aber $r \neq r'$. Daraus folgt $LT(r - r') \in \langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$. Nach Lemma 2.10 ist $LT(r - r')$ teilbar durch eines der $LT(g_i)$. Dies ist aber unmöglich, da kein Term von r und r' durch eines der $LT(g_i)$ teilbar ist. Also muß $r - r' = 0$ gelten und die Eindeutigkeit ist gezeigt. \square

Bemerkung. Obwohl der Rest r in $f = a_1g_1 + \dots + a_tg_t + r$ eindeutig ist, gilt dies im Normalfall nicht für die Koeffizienten a_1, \dots, a_t .

Folgerung 2.17. Sei $G = \{g_1, \dots, g_t\}$ eine Gröbner-Basis eines Ideals $I \subset \mathbb{P}$ und sei $f \in \mathbb{P}$. Es gilt $f \in I$ genau dann, wenn bei der Division von f durch G der Rest gleich Null ist.

Beweis: Wir haben schon weiter oben gesehen, daß das Polynom f in $I = \langle G \rangle$ liegt, wenn der Rest gleich Null ist. Sei umgekehrt $f \in I = \langle G \rangle$. Dann erfüllt $f = f + 0$ die Bedingungen aus Satz 2.16 und somit muß der Rest gleich Null sein. \square

Wir haben nun eine Möglichkeit, das *Ideal Membership Problem* zu lösen, bei dem man bestimmen will, ob ein Polynom in einem Ideal liegt. Nun fehlt uns nur noch die Möglichkeit zu erkennen, ob eine Basis eine Gröbner-Basis ist oder nicht.

Definition 2.18. Wir schreiben \overline{f}^F für den Rest bei der Division von f durch das geordnete s -Tupel $F = (f_1, \dots, f_s)$. Ist F eine Gröbner-Basis für $\langle f_1, \dots, f_s \rangle$, dann können wir nach Satz 2.16 F als Menge ansehen, bei der die Ordnung irrelevant ist.

Beispiel. Sei $F = (x^2y - y^2, x^4y^2 - y^2) \subset \mathbb{Z}[x, y]$. Benutzen wir die lexikographische Ordnung, erhalten wir

$$\overline{x^5y}^F = xy^3,$$

da der Divisionsalgorithmus zur Gleichung

$$x^5y = (x^3 + xy)(x^2y - y^2) + 0 \cdot (x^4y^2 - y^2) + xy^3$$

führt.

Wie wir schon angedeutet haben, kann das mögliche Auftreten von ungünstigen Linearkombinationen der f_i dazu führen, daß der entstehende Leiterterm nicht mehr im von $LT(f_i)$ erzeugten Ideal liegt. Dies ist das Hindernis, das wir zum Erreichen einer Gröbner-Basis überwinden müssen.

Definition 2.19. Seien $f, g \in \mathbb{P}$ mit $f \neq 0 \neq g$.

1. Ist $\deg(f) = \alpha$ und $\deg(g) = \beta$, dann sei $\gamma := (\gamma_1, \dots, \gamma_n)$, wobei $\gamma_i = \max\{\alpha_i, \beta_i\}$ für alle i . Wir nennen x^γ das *kleinste gemeinsame Vielfache* von $LT(f)$ und $LT(g)$ und schreiben $x^\gamma = \text{kgV}(LT(f), LT(g))$.
2. Das *Syzygien-Polynom* (oder kurz *S-Polynom*) von f und g ist

$$S(f, g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g.$$

Man beachte, daß die Leitkoeffizienten ebenfalls invertiert werden.

Beispiel. Sei $f = x^3y^2 - x^2y^3 + x$ und $g = 3x^4y + y^2$ in $\mathbb{Z}[x, y]$ mit der *Grlex Order*. Dann ist $\gamma = (4, 2)$ und

$$\begin{aligned} S(f, g) &= \frac{x^4y^2}{x^3y^2} \cdot f - \frac{x^4y^2}{3x^4y} \cdot g \\ &= x \cdot f - \frac{1}{3} \cdot y \cdot g \\ &= -x^3y^3 + x^2 - \frac{1}{3}y^3. \end{aligned}$$

Kommen wir nun zum Haupttheorem dieses Abschnitts, in dem wir erfahren, wann eine Basis für ein Ideal eine Gröbner-Basis ist.

Theorem 2.20. (Buchbergers S-Paar Kriterium) Sei I ein Polynomideal. Dann ist eine Basis $G = \{g_1, \dots, g_t\}$ von I genau dann eine Gröbner-Basis, wenn für alle Paare $i \neq j$ der Rest der Division von $S(g_i, g_j)$ durch G gleich Null ist, das heißt

$$\overline{S(g_i, g_j)}^G = 0 \quad \forall i \neq j.$$

Beweisidee³. Die eine Richtung folgt sofort durch Folgerung 2.17.

Für die Rückrichtung sei f ein Polynom aus I ungleich Null. Zu zeigen ist, daß, falls alle S-Polynome nach Division durch G den Rest Null haben, der Leitterm von f in

³Für den vollen Beweis siehe Cox, Little und O'Shea [5, S.82 ff].

$\langle LT(g_1), \dots, LT(g_t) \rangle$ liegt. Da $f \in I$, kann man auch

$$f = \sum_{i=1}^t h_i g_i \quad (2)$$

mit Koeffizienten $h_i \in \mathbb{P}$ schreiben. Da wir eine Wohlordnung haben, gilt

$$\deg(f) \leq \max \{ \deg(h_i g_i) \}. \quad (3)$$

Gilt hier keine Gleichheit, haben sich einige Leitterme in (2) aufgehoben. Man kann nun diese Gleichung auch als Linearkombination der S-Polynome schreiben. Durch die Annahme, daß alle diese S-Polynome bei Division durch G den Rest Null haben, können wir eine Darstellung von f mittels der S-Polynome suchen, in der sich so wenig Terme wie möglich aufheben. Fortgesetzt bedeutet dies, daß man einen Ausdruck für (2) findet, für den Gleichheit in (3) gilt. Das heißt, $\deg(f) = \max \{ \deg(h_i g_i) \}$ für ein i und somit ist $LT(f)$ teilbar durch $LT(g_i)$. Daraus folgt dann $LT(f) \in \langle LT(g_1), \dots, LT(g_t) \rangle$ und das Theorem ist bewiesen. \square

Beispiel. Sei $I = \langle y - x^2, z - x^3 \rangle$. Wir behaupten, daß $G = \{y - x^2, z - x^3\}$ eine Gröbner-Basis für *Lex Order* mit $y > z > x$ ist.

Dazu betrachten wir das S-Polynom

$$S(y - x^2, z - x^3) = \frac{yz}{y}(y - x^2) - \frac{yz}{z}(z - x^3) = -zx^2 + yx^3.$$

Mittels des Divisionsalgorithmus erhalten wir

$$-zx^2 + yx^3 = x^3 \cdot (y - x^2) + (-x^2) \cdot (z - x^3) + 0,$$

also ist $\overline{S(y - x^2, z - x^3)}^G = 0$. Nach Theorem 2.20 bedeutet dies, daß G eine Gröbner-Basis von I ist. Man kann übrigens leicht zeigen, daß G für *Lex Order* mit $x > y > z$ keine Gröbner-Basis mehr ist.

2.5 Buchbergers Algorithmus

Aus Folgerung 2.15 wissen wir zwar, daß es für jedes von Null verschiedene Ideal I eine Gröbner-Basis gibt, aber bisher wir konnten keinen Algorithmus angeben, um aus einer

bestehenden Idealbasis eine Gröbner-Basis zu erzeugen. Mit Hilfe des S-Paar Kriteriums ist dies nun möglich.

Theorem 2.21. Sei $I = \langle F \rangle \neq \{0\}$ ein Polynom-Ideal mit $F = (f_1, \dots, f_s)$. Dann kann man für I eine Gröbner-Basis auf folgende Art und Weise konstruieren:

Algorithmus 2.22 nach Cox, Little und O’Shea [5, S.87 f]

1. Setze $G := F$ und $G' := \emptyset$.
2. Solange $G \neq G'$:
 - Setze $G' := G$.
 - Für jedes Paar $\{p, q\}$, $p \neq q$ in G' :
 - Setze $S := \overline{S(p, q)}^{G'}$.
 - Ist $S \neq 0$, dann $G := G \cup \{S\}$.

Beweis: Wir zeigen zuerst, daß zu jeder Zeit des Algorithmus $G \subseteq I$ gilt. Dies ist zu Beginn sicher wahr. Während des Algorithmus wird G nur durch den Rest $S = \overline{S(p, q)}^{G'}$ mit $p, q \in G$ erweitert. Wenn $G \subseteq I$, dann liegen auch p, q und somit auch $S(p, q)$ in I . Da wir nur durch $G' \subseteq I$ teilen, erhalten wir $G \cup \{S\} \subseteq I$. Wir bemerken außerdem, daß G immer eine Basis von I ist, da zu jeder Zeit $F \subseteq G$ gilt.

Der Algorithmus wird beendet, falls $G = G'$ ist, was bedeutet, daß $\overline{S(p, q)}^{G'} = 0$ für alle $p, q \in G$. Folglich ist G eine Gröbner-Basis nach Theorem 2.20.

Es bleibt noch zu zeigen, daß der Algorithmus terminiert. Die Menge G besteht immer aus G' und dem Rest der S-Polynome von Elementen aus G' . Da $G' \subseteq G$, gilt außerdem $\langle LT(G') \rangle \subseteq \langle LT(G) \rangle$. Wir behaupten, daß $\langle LT(G') \rangle \subset \langle LT(G) \rangle$, solange $G \neq G'$. Nehmen wir also an, der Rest $r \neq 0$ eines S-Polynoms wurde zu G hinzugefügt. Da r ein Rest der Division durch G' ist, ist $LT(r)$ nicht teilbar durch die Leiterterme der Elemente von G' und folglich ist $LT(r) \notin \langle LT(G') \rangle$. Da aber $LT(r) \in \langle LT(G) \rangle$, zeigt dies, daß G' immer kleiner ist als G .

Wegen $\langle LT(G') \rangle \subseteq \langle LT(G) \rangle$ bilden $\langle LT(G') \rangle$ eine ansteigende Folge von Idealen in \mathbb{P} . Aufgrund der Eigenschaft, daß diese Kette nach endlich vielen Schritten stationär werden muß⁴, da \mathbb{P} ein Noetherscher Ring ist, gilt daher irgendwann $\langle LT(G') \rangle = \langle LT(G) \rangle$, was ebenfalls $G = G'$ impliziert. \square

⁴Siehe dazu Cox, Little und O’Shea [5, S.76 f (Ascending Chain Condition)]

Beispiel. Betrachten wir Algorithmus 2.22 anhand eines Beispiels genauer. Sei $I = \langle f_1, f_2 \rangle$ mit $f_1 = x^2 - y, f_2 = x^3 - y^2 \in K[x, y]$ und *Lex Order*. Da $LT(S(f_1, f_2)) = -xy \notin \langle LT(f_1), LT(f_2) \rangle$, ist $\{f_1, f_2\}$ keine Gröbner-Basis von I .

Der Algorithmus läuft nun wie folgt:

1. $G := (x^2 - y, x^3 - y^2) = (g_1, g_2)$
2. $G' := G$
3. $\overline{S(g_1, g_2)}^{G'} = -xy + y^2 \Rightarrow G := (x^2 - y, x^3 - y^2, -xy + y^2) = (g_1, g_2, g_3)$
4. $G \neq G' \Rightarrow$ von vorn
5. $G' := G$
6. $\overline{S(g_1, g_2)}^{G'} = 0$
7. $\overline{S(g_1, g_3)}^{G'} = y^3 - y^2 \Rightarrow G := (x^2 - y, x^3 - y^2, -xy + y^2, y^3 - y^2) = (g_1, g_2, g_3, g_4)$
8. $\overline{S(g_2, g_3)}^{G'} = 0$
9. $G \neq G' \Rightarrow$ von vorn
10. $G' := G$
11. $\overline{S(g_i, g_j)}^{G'} = 0 \forall i, j \in \{1, \dots, 4\}$ mit $i \neq j$
12. $G' = G \Rightarrow$ Ende

Wir erhalten damit, daß $G = \{x^2 - y, x^3 - y^2, -xy + y^2, y^3 - y^2\}$ eine Gröbner-Basis von I ist.

Bemerkung. Drehen wir im übrigen die Reihenfolge der Polynome um, erhalten wir das gleiche Ergebnis, da $S(g_1, g_2) = S(g_2, g_1)$ ist.

Ein großes Problem ist, daß Gröbner-Basen nach obiger Berechnung oft größer als nötig sind. Dies wollen wir jetzt ändern.

Lemma 2.23. Sei G eine Gröbner-Basis für das Polynom-Ideal I . Sei $p \in G$, so daß $LT(p) \in \langle LT(G \setminus \{p\}) \rangle$. Dann ist auch $G \setminus \{p\}$ eine Gröbner-Basis von I .

Beweis: Wir wissen, daß $\langle LT(G) \rangle = \langle (LT(I)) \rangle$. Ist $LT(p) \in \langle LT(G \setminus \{p\}) \rangle$, dann ist $\langle LT(G \setminus \{p\}) \rangle = \langle LT(G) \rangle$. Daraus folgt die Behauptung. \square

Definition 2.24. Eine Gröbner-Basis G für ein Polynom-Ideal I heißt *minimal*, falls gilt:

1. $LK(p) = 1$ für alle $p \in G$.
2. Für alle $p \in G$ gilt $LT(p) \notin \langle LT(G \setminus \{p\}) \rangle$.

Beispiel. Setzen wir unser obiges Beispiel fort. Nach Normalisierung (auf Leitkoeffizient 1) erhalten wir $G = \{x^2 - y, x^3 - y^2, xy - y^2, y^3 - y^2\}$. Nun ist $LT(g_2) = x^3$ ein Vielfaches von $LT(g_1) = x^2$. Damit erhalten wir als minimale Gröbner-Basis $G = \{x^2 - y, xy - y^2, y^3 - y^2\}$.

Da diese minimale Gröbner-Basis aber leider nicht eindeutig ist, benötigen wir eine besondere Eigenschaft.

Definition 2.25. Eine Gröbner-Basis G für ein Polynom-Ideal I heißt *reduziert*, falls gilt:

1. $LK(p) = 1$ für alle $p \in G$.
2. Für alle $p \in G$ liegt kein Monom von p in $\langle LT(G \setminus \{p\}) \rangle$.

Bemerkung. Bei den Algorithmen werden wir auf Eigenschaft (1) der reduzierten Basis verzichten, da die Normierung zu Leitkoeffizient 1 oft zu komplizierten Brüchen in den Koeffizienten der Polynome führt. Polynome werden stattdessen so normiert, daß der größte gemeinsame Teiler aller Koeffizienten 1 ergibt.

Wir wollen nun hier noch einen zweiten Algorithmus für die Berechnung einer Gröbner-Basis präsentieren (nach Buchberger [3, S.192 f]):

Algorithmus 2.26

1. Setze $G := F$.
2. Setze $B := \{(f_i, f_j) \mid f_i, f_j \in G \text{ mit } i < j\}$.
3. Solange $B \neq \emptyset$:
 - Wähle ein $b = (b_1, b_2) \in B$ und setze $B := B \setminus \{b\}$.

- Setze $S := \overline{S(b_1, b_2)}^{G'}$.
- Ist $S \neq 0$, dann $G := G \cup \{S\}$ und $B := B \cup \{(g, S) \mid g \in G\}$.

Algorithmus 2.26 hat gegenüber Algorithmus 2.22 den Vorteil, daß nicht ständig alle Paare von Polynomen betrachtet werden, sondern diese einfach nur aus der Menge B genommen werden müssen. Dennoch haben Tests während dieser Arbeit gezeigt, daß der Algorithmus nur unwesentlich schneller ist.

Verbesserungen des Algorithmus:

1. Da wir am Ende eine reduzierte Gröbner-Basis erhalten möchten, ist es praktischer, nach jedem Hinzufügen von Polynomen zur Basis zu schauen, ob andere Polynome wegfallen. Dadurch hält man die Anzahl der Polynome und auch der Berechnungen der S-Polynome klein und erhält am Ende automatisch eine reduzierte Gröbner-Basis. Die Idee dahinter kann in Buchberger [3, S.195 ff] nachgelesen werden.
2. Nach Cox, Little und O'Shea [5, S.8] läßt sich die Anzahl der Divisionen bei der Bestimmung des Restes etwas reduzieren, wenn man die Elemente in der Basis bezüglich der gegebenen monomialen Ordnung der Größe nach ordnet.
3. Die Komplexität des Algorithmus hängt sehr stark von der Reihenfolge ab, in der man die S-Polynome berechnet. Normalerweise sollte man zuerst das S-Polynom des Polynom-Paares ausrechnen, deren kleinstes gemeinsames Vielfaches der beiden Leitmonome am kleinsten bezüglich der monomialen Ordnung ist (aus Buchberger [3, S.194 f]).

Punkt (1) wurde im endgültigen Algorithmus (siehe Kapitel 2.3 im nächsten Teil auf Seite 46) in abgewandelter Form implementiert. Punkte (2) und (3) wurden aber weggelassen, da diese nicht zur Verbesserung der Laufzeit geführt haben. Man betrachte dazu folgendes Beispiel.

Beispiel. Sei eine Idealbasis $F := \langle x_1^2 - 2ax_1 + a^2 + y_1^2 - l_1^2, x_2^2 + 2ax_2 + a^2 + y_2^2 - l_2^2, x_1^2 + y_1^2 - x_2^2 - y_2^2, x_1^2 - 2x_1x_2 + x_2^2 + y_1^2 - 2y_1y_2 + y_2^2 - 4b^2 \rangle$ mit Unbekannten x_1, x_2, y_1, y_2 und Konstanten a, b, l_1, l_2 gegeben. Wir wollen die Gröbner-Basis hiervon berechnen beziehungsweise die Zeit zur Berechnung messen.

Algorithmus	Dauer in sec
Alg. 2.22	45,797
Alg. 2.22 mit Verbesserung 2	81,247
Alg. 2.26	61,203
Alg. 2.26 mit Verbesserung 3	56,749

Auch wenn Verbesserung 3 eine Leistungssteigerung gebracht hat, so ist Algorithmus 2.26 dennoch langsamer als Buchbergers ursprüngliche Variante, welche wir auch im Fortlauf nutzen wollen⁵.

⁵Weitere Messungen haben gezeigt, daß Algorithmus 2.22 auch für größere Gröbner-Basen schneller ist.

3 H-Basen

Die Frage, die sich jetzt stellt, ist: Warum soll man etwas anderes als Gröbner-Basen benutzen, wenn man diese bereits zur Verfügung hat und sie sich mitunter auch schneller berechnen lassen? Eine Antwort ist hier unter anderem: Weil Symmetrien bei Gröbner-Basen oft zerstört werden.

Viele Probleme in der Technik beruhen auf Symmetrien. Betrachten wir das Polynom $x^2 + y^2$. Wenn man nun den Leitterm hiervon unter Gröbner-Basen betrachtet, so wird entweder x oder y bevorzugt. Eine Variable ist also immer dominant. Mit H-Basen kann dies vermieden werden, da man grundsätzlich nur homogene Leitterme betrachtet, das heißt alle Monome mit höchstem Grad.

Bemerkung. Jede Gröbner-Basis, die auf einer graduierten Ordnung basiert, ist dabei immer eine H-Basis, welche die Symmetrien aber unter Umständen nicht beachtet.

Beispiel. Sei $I = \langle x^2z + y^3, xy^2 - z^3 \rangle$. Die Gröbner-Basis mit graduiert lexikographischer Ordnung ist $\langle x^2z + y^3, xy^2 - z^3, xz^4 + y^5, y^7 + z^7 \rangle$, was ebenfalls eine H-Basis nach späterer Definition sein wird.

3.1 Ein weiterer Gradbegriff

Auch bei H-Basen spielen Ordnungen eine wichtige Rolle. Es sind hier aber grundsätzlich nur die Ordnungen wichtig, in denen der Totalgrad an erster Stelle steht. Wir können nun also einen weiteren Gradbegriff einführen.

Definition 3.1. Sei $0 \neq f = \sum_{\alpha \in \mathbb{N}_0^n} a_\alpha x^\alpha$ ein Polynom in \mathbb{P} und sei $>$ die natürliche Ordnung auf \mathbb{N}_0 .

1. Der *Grad* von f ist

$$\delta(f) = \max\{|\alpha| \in \mathbb{N}_0 \mid a_\alpha \neq 0\},$$

wobei das Maximum bezüglich $>$ gemeint ist und $|\alpha| = \sum_{i=1}^n \alpha_i$.

2. Der *Leitterm* von f ist

$$\Lambda(f) = \sum_{|\alpha|=\delta(f)} a_\alpha x^\alpha.$$

Beispiel. Sei $f = 2xy + 4 - x^2y - y^3 \in \mathbb{Z}[x, y]$. Das Polynom ist nun wie folgt geordnet:

$$f = -x^2y - y^3 + 2xy + 4 \text{ mit } \delta(f) = 3.$$

Der Leitterm von f ist

$$\Lambda(f) = -x^2y - y^3.$$

Wie man sieht, kann der Leitterm nun ein homogenes Polynom sein und ist nicht mehr zwingend ein Monom wie bei Gröbner-Basen.

Bemerkung. Es sollte klar sein, daß es damit auch keine Definition mehr für Leitmonom oder Leitkoeffizient geben kann.

3.2 Lineare Räume

Um einen Divisionsalgorithmus für H-Basen zu formulieren, können wir leider nicht einfach nur den Algorithmus aus Kapitel 2.2 umformulieren, denn bei H-Basen ist es möglich, daß ein Polynom das andere "ein bißchen" teilt.

Beispiel. Sei $f = x^2 + y^2$ und $g = x^3 + y^3$. Bei Gröbner-Basen betrachtet man die Leittermine und errechnet den Teiler hiervon. Es ergibt sich, daß

$$g = f \cdot x + (-xy^2 + y^3).$$

Man sieht hier schon, daß der Rest $-xy^2 + y^3 = (-x + y) \cdot y^2$ das Monom y^2 beinhaltet. Und genau das wird bei H-Basen nicht ignoriert. Es ist hier:

$$g = f \cdot \frac{3}{4}(x + y) + \frac{x^3 - 3x^2y - 3xy^2 + y^3}{4}.$$

Desweiteren fordern wir zusätzlich die stärkere Eigenschaft, daß der Rest r bei der Division von f durch eine Idealbasis F orthogonal zu $f - r$ ist.

Definition 3.2. Es ist

$$P_d = \{p \mid p \text{ ist ein homogenes Polynom aus } \mathbb{P} \text{ vom Grad } d \}$$

der lineare Raum aller homogenen Polynome vom Grad d . Eine Basis dieses Raumes ist

$L := \{x^\alpha \text{ mit } |\alpha| = d\}$. Per Definition sei $P_d = \{0\}$ für alle $d < 0$.

Wir schreiben $\langle\langle L \rangle\rangle = P_d$ für das (lineare) Erzeugnis von L .

Beispiel. Sei $\mathbb{P} = \mathbb{Z}[x, y]$. Dann ist

1. $P_0 = \langle\langle 1 \rangle\rangle = \mathbb{Z}$.
2. $P_1 = \langle\langle x, y \rangle\rangle$.
3. $P_2 = \langle\langle x^2, xy, y^2 \rangle\rangle$.

Von nun an sei $F := (f_1, \dots, f_s) \subset \mathbb{P}$ eine (geordnete) Basis für das Ideal I , das heißt $I = \langle F \rangle$.

Definition 3.3. Es ist

$$V_d(F) = \left\{ \sum_{i=1}^s h_i \Lambda(f_i) \mid h_i \in P_{d-\delta(f_i)} \right\}.$$

Dieser Raum gibt uns vor, woher wir die Koeffizienten beim Divisionsalgorithmus wählen können.

Beispiel. Sei $F = (x^2 + xy, -y + 1) \subset \mathbb{Z}[x, y]$. Es ist nun

$$\begin{aligned} V_d(F) &= \left\{ \sum_{i=1}^s h_i \Lambda(f_i) \mid h_i \in P_{d-\delta(f_i)} \right\} \\ &= \{h_1 \Lambda(f_1) + h_2 \Lambda(f_2) \mid h_i \in P_{d-\delta(f_i)}\} \\ &= \{h_1(x^2 + xy) + h_2 y \mid h_1 \in P_{d-2}, h_2 \in P_{d-1}\} \end{aligned}$$

Da wir grundsätzlich lieber mit endlichen Mengen rechnen, betrachten wir anstelle von P_d als ganzen Raum immer nur die Basis hiervon. Wir erhalten dann zum Beispiel

$$\begin{aligned} V_2(F) &= \langle\langle h_1(x^2 + xy) + h_2 y \mid h_1 \in P_0, h_2 \in P_1 \rangle\rangle \\ &= \langle\langle (x^2 + xy) + xy, (x^2 + xy) + yy \rangle\rangle \\ &= \langle\langle x^2 + 2xy, x^2 + xy + y^2 \rangle\rangle, \end{aligned}$$

was ein Unterraum von $P_2 = \langle\langle x^2, xy, y^2 \rangle\rangle$ ist.

Bemerkung. Grundsätzlich gilt: $V_d(F) \subseteq P_d$ für alle $d \in \mathbb{Z}$.

Definition 3.4. Für ein Ideal I gilt wie oben

$$V_d(I) := \left\{ \sum_{f \in I} h_f \Lambda(f) \mid h_f \in P_{d-\delta(f)} \right\},$$

was ebenfalls ein linearer Unterraum von P_d ist.

Jetzt benötigen wir noch das orthogonale Komplement, um den Rest bei der Division zu berechnen. Hierzu brauchen wir aber zuerst ein geeignetes Skalarprodukt, mit dem wir die Orthogonalität zweier Polynome bestimmen können.

Definition 3.5. Es seien f und g Polynome in \mathbb{P} . Das *monomiale Skalarprodukt* ist dann

$$(f, g) = \sum_{\alpha \in \mathbb{N}_0^n} f_\alpha \cdot g_\alpha,$$

wobei α der Multiindex in den Polynomen ist.

Ein weiteres Skalarprodukt, welches wir hier auch grundsätzlich benutzen werden, ist mit Hilfe des linearen Differentialoperators definiert:

$$(f, g) = \sum_{\alpha \in \mathbb{N}_0^n} f_\alpha \cdot g_\alpha \cdot \alpha!.$$

Wir nennen dies das *Differential-Skalarprodukt*.

Bemerkung. Diese Definition liefert uns: $(f, g) = 0$, wenn $\delta(f) \neq \delta(g)$.

Beispiele. Wir betrachten das Differential-Skalarprodukt.

1. $(x^2 - y, xy + 1) = 0$, da keine zwei gleichen Monome vorkommen.

2. $(x^2 - 3xy, 2xy + 1) = -6$, da

$$\begin{aligned} (x^2 - 3xy, 2xy + 1) &= 1 \cdot 0 \cdot (2, 0)! + -3 \cdot 2 \cdot (1, 1)! + 0 \cdot 1 \cdot (0, 0)! \\ &= -6 \cdot 1! \cdot 1! = -6 \end{aligned}$$

3. $(x^2y - 3xy, -3x^2y - 2xy) = 0$, da

$$\begin{aligned} (x^2y - 3xy, -3x^2y - 2xy) &= 1 \cdot (-3) \cdot (2, 1)! + (-3) \cdot (-2) \cdot (1, 1)! \\ &= -3 \cdot 2 + 6 \cdot 1 \\ &= -6 + 6 = 0 \end{aligned}$$

Nun können wir den zu $V_d(I)$ orthogonalen Raum definieren.

Definition 3.6. Sei $V_d(I)$ wie oben. Dann ist

$$W_d(I) := \{p \in P_d \mid (V_d(I), p) = 0\}$$

der zu $V_d(I)$ orthogonale Raum. Die Definition von $W_d(F)$ ist analog.

3.3 Divisionsalgorithmus

Theorem 3.7. Sei $F = (f_1, \dots, f_s) \subset \mathbb{P}$, $I = \langle F \rangle$ und $f \in \mathbb{P}$. Dann gibt es $h_i \in \mathbb{P}$, so daß $h_i f_i \in V(I)$ und $r \in W(I)$ mit

$$f = \sum_{i=1}^s h_i f_i + r$$

und $h_i f_i \perp r \forall i = 1, \dots, s$. Es ist hier $V(I) = \bigcup_{d \in \mathbb{N}_0} V_d(I)$ und $W(I) = \bigcup_{d \in \mathbb{N}_0} W_d(I)$.

Beweis: Für den Beweis verwenden wir den Algorithmus von Sauer [11, S.2295 f].

Algorithmus 3.8

1. Setze $g := f$.
2. Solange $g \neq 0$:
 - Setze $d := \delta(g)$.
 - Finde Polynome $h_{d,i} \in P_{d-\delta(f_i)}$, $i = 1, \dots, s$, so daß gilt:
 $r_d := \Lambda(g) - \sum_{i=1}^s h_{d,i} \Lambda(f_i) \perp V_d(F)$.
 - Setze $g := g - \sum_{i=1}^s h_{d,i} f_i - r_d$.
3. Setze $h_i := \sum_{d=0}^{\delta(f)} h_{d,i}$ und $r := \sum_{d=0}^{\delta(f)} r_d$.

Der Algorithmus terminiert, sobald $g = 0$ ist. Dies ist auf alle Fälle gegeben, denn in jedem Schritt wird g durch

$$\begin{aligned} g' &= g - \sum_{i=1}^s h_{d,i} f_i - r_d \\ &= g - \sum_{i=1}^s h_{d,i} f_i - \Lambda(g) + \sum_{i=1}^s h_{d,i} \Lambda(f_i) \\ &= g - \Lambda(g) - \sum_{i=1}^s h_{d,i} (f_i - \Lambda(f_i)) \end{aligned}$$

ersetzt. Dabei hat g' einen Totalgrad $\delta(g') < \delta(g)$, da sich die Leitterme $\Lambda(g)$ und $\Lambda(f_i)$ wegheben.

Es ist unmittelbar klar, daß $h_i f_i \perp r$ für alle $i = 1, \dots, s$, da die Definition unseres Skalarproduktes $h_{k,i} f_i \perp r_l$ für alle $k \neq l$ gibt und zusätzlich der Algorithmus die Aussage für $k = l$ liefert. \square

Bemerkung. Dieser Algorithmus ist aufgrund der Unterbestimmtheit des Systems nicht eindeutig. Die Wahl der h_i ist vollkommen frei. Wir betrachten dazu das folgende Beispiel.

Beispiel. Sei $F := (x + y, x)$ und $f := x - y^2$. Es ist dann

$$r_2 := -y^2 - (c_{1,1}x + c_{1,2}y)(x + y) - (c_{2,1}x + c_{2,2}y)x$$

mit Unbekannten $c_{i,j}$. Wir können nun $c_{1,1} = 1, c_{1,2} = -1, c_{2,1} = -1, c_{2,2} = 0$ setzen und erhalten so

$$\begin{aligned} r_2 &= -y^2 - (x - y)(x + y) - (-x)x \\ &= -y^2 - x^2 + y^2 + x^2 \\ &= 0. \end{aligned}$$

Die alternative Wahl $c_{1,1} = 0, c_{1,2} = -1, c_{2,1} = 0, c_{2,2} = 1$ ergibt aber ebenso

$$\begin{aligned} r_2 &= -y^2 - (-y)(x + y) - (y)x \\ &= -y^2 + xy + y^2 - xy \\ &= 0. \end{aligned}$$

Es ist nun in beiden Fällen $g' = g - \Lambda(g) - r_2 = x$, was wir wieder dividieren können. Wir erhalten am Ende zwei Darstellungen

1. $g = (x - y)f_1 + (-x + 1)f_2 = (x - y)(x + y) + (-x + 1)x = -y^2 + x$ und
2. $g = (-y)f_1 + (y + 1)f_2 = (-y)(x + y) + (y + 1)x = -y^2 + x$,

die beide die gewünschten Bedingungen erfüllen.

3.4 H-Basen

Wir wollen zuerst definieren, was eine H-Basis ist.

Definition 3.9. Sei $F := (f_1, \dots, f_s) \subset \mathbb{P}$. F heißt *H-Basis* des Ideals $I = \langle F \rangle$, wenn für alle $f \in I$ Polynome $h_1, \dots, h_s \in \mathbb{P}$ existieren mit

$$f = \sum_{i=1}^s h_i f_i \quad \text{und} \quad \delta(h_i f_i) \leq \delta(f).$$

Bemerkung. Obiger Divisionsalgorithmus liefert uns genau diese Eigenschaft.

Man kann eine H-Basis auch auf andere Arten definieren.

Theorem 3.10. Seien $F := (f_1, \dots, f_s) \subset \mathbb{P}$ und $I = \langle F \rangle$. Dann sind folgende Aussagen äquivalent:

1. F ist eine H-Basis von I .
2. $\langle \Lambda(f_1), \dots, \Lambda(f_s) \rangle = \langle \Lambda(I) \rangle$.
3. Für jedes $f \in I$ gilt: f hat Rest 0 bei Division durch F .

Beweis:

"1 \Rightarrow 2": Die Inklusion $\langle \Lambda(f_1), \dots, \Lambda(f_s) \rangle \subset \langle \Lambda(I) \rangle$ ist klar.

Für die andere Richtung sei $f \in I$. Dann ist $f = \sum_{i=1}^s h_i f_i$ und $\delta(h_i f_i) \leq \delta(f)$, $h_i \in \mathbb{P}$. Wir definieren $J := \{j \mid \delta(h_j f_j) = \delta(f)\}$. Dann läßt sich der Leiternorm von f als $\Lambda(f) = \sum_{j \in J} \Lambda(h_j f_j) = \sum_{j \in J} \Lambda(h_j) \Lambda(f_j)$ schreiben. Das bedeutet $\langle \Lambda(f_1), \dots, \Lambda(f_s) \rangle \supset \langle \Lambda(I) \rangle$ und daraus folgt 2.

"2 \Rightarrow 3": Dies folgt per Induktion über die Leiterterme beziehungsweise den Grad von f . Gilt 2, dann kann man den Leiterterm von f als $\Lambda(f) = \sum_{j \in J} \Lambda(h_j) \Lambda(f_j)$ schreiben, J wie oben. Reduzieren wir f nun zu $f' = f - \sum_{j \in J} \Lambda(h_j) \Lambda(f_j)$, können wir dies induktiv fortsetzen.

"3 \Rightarrow 1": Mehrmaliges Anwenden des Divisionsalgorithmus gibt uns eine Kette von Reduktionen von f , das heißt, $f =: g_0$ läßt sich unter F zu g_1 reduzieren, läßt sich unter F zu g_2 reduzieren, ..., läßt sich unter F zu $g_t = 0$ reduzieren. Es ist dann $\Lambda(g_{i-1}) = \sum_{j=1}^s \Lambda(h_{ij}) \Lambda(f_j) \forall i = 1, \dots, t$. Somit erhalten wir für f die Darstellung

$$f = \sum_{j=1}^s \sum_{i=1}^t \Lambda(h_{ij}) f_j,$$

was der Definition einer H-Basis entspricht. \square

Wie schon bei Gröbner-Basen (vgl. Seite 17) bekommen wir auch hier wieder Probleme, wenn sich bei zwei Polynomen die Leiterterme aufheben, so daß das Ergebnis möglicherweise nicht mehr im Ideal liegt. Leider lassen sich bei H-Basen die Syzygien nicht so leicht berechnen wie bei Gröbner-Basen. Zuerst wollen wir aber den Begriff Syzygie definieren.

Definition 3.11. Sei $F := (f_1, \dots, f_s) \subset \mathbb{P}$. Die Menge

$$S(F) := \left\{ (h_1, \dots, h_s)^T \in \mathbb{P}^s \mid \sum_{i=1}^s h_i f_i = 0 \right\}$$

heißt *Syzygienmodul* von F . Eine *Syzygie* zu F ist ein Element aus $S(F)$.

Die Verbindung zwischen diesem Modul und der H-Basis folgt sofort.

Theorem 3.12. (ohne Beweis) Sei $F = (f_1, \dots, f_s) \subset \mathbb{P}$ eine Basis des Ideals I . Dann ist F genau dann eine H-Basis, wenn sich jedes Element

$$p := \sum_{i=1}^s h_i f_i, \quad (h_1, \dots, h_s) \in S(\Lambda(F))$$

zu Null reduzieren läßt, das heißt, wenn jedes Element bei Division durch F den Rest 0 läßt.

Über dieses Theorem erhält man auch sofort einen Algorithmus zur Berechnung einer H-Basis.

Algorithmus 3.13 nach Sauer [11, S. 2300]

1. Setze $H := F = (f_1, \dots, f_s)$ und reduziere H dabei⁶.
2. Berechne eine Basis G von $S(\Lambda(F))$.
3. Schleife über jedes Element $g = (g_1, \dots, g_s)$ in G :
 - Berechne den Rest r von $p = \sum_{i=1}^s g_i f_i$ bei Division durch F .
 - Ist $r \neq 0$, füge r zu H hinzu.
 - Reduziere H soweit wie möglich.
4. Ist $F \neq H$, setze $F := H$ und gehe zu 1.

Als Ergebnis erhält man eine H-Basis H von $\langle F \rangle$. Man kann auf die gleiche Art und Weise wie bei Gröbner-Basen zeigen (vgl. Kapitel 2.5, Seite 18), daß der Algorithmus korrekt ist.

Zum Ende stellt sich noch die Frage, wie wir einer Idealbasis F ansehen, ob diese vielleicht schon eine H-Basis ist. Hierzu gibt es folgende Methoden:

1. Wenn F bereits eine graduierte Gröbner-Basis ist, was man leicht durch die Berechnung der Syzygienpolynome überprüfen kann und die homogenen Leiterterme von F nur Monome sind, dann ist F bereits eine H-Basis. Dies ist sofort klar, da die homogenen Leiterterme im Falle von Monomen gleich den Leitertermen bei graduierten Ordnungen sind und für diese ist F bereits eine Gröbner-Basis.
2. Wenn die homogenen Leiterterme von F nur den Punkt 0 als gemeinsame Nullstelle haben, ist F bereits eine H-Basis. (Beweis siehe Möller und Sauer [8, S.347 f].)
3. Das dritte Kriterium ist recht simpel. Man berechnet eine Basis des Syzygien-Moduls und überprüft, ob alle Syzygienpolynome modulo F zu Null reduziert werden.

⁶Die Reduktion geschieht dabei wie bei Gröbner-Basen (siehe Kapitel III.2.3, Seite 46).

4 Lösung von polynomialen Gleichungen

Wie schon in der Einführung in Kapitel 1 auf Seite 5 gesagt, können wir die Lösung mehrerer polynomialer Gleichungen bestimmen, in dem wir die Varietät des zugehörigen Polynomideals betrachten. Hierfür gibt es einen sehr guten Ansatz von Möller und Stetter [9], welche die Berechnung der Varietät auf ein Eigenwertproblem von Matrizen zurückführen.

4.1 Dimension eines Ideals

Definition 4.1. Sei $I \subset \mathbb{P} = R[X]$ ein Ideal mit $X = [x_1, \dots, x_n]$. Für jede (geordnete) Teilmenge $U \subseteq X$ bezeichnet I_U das *Eliminationsideal* $I \cap R[U]$.

Beispiele. Seien $X = [x, y, z]$ und $I = \langle x^2 - yz^2, y^2 - x, 2y^2 \rangle$.

1. Sei $U = [z] \subset X$. Dann ist $I_U = \langle \rangle$, da z in keinem Basispolynom isoliert vorkommt.
2. Sei $U = [x, y] \subset X$. Dann ist $I_U = \langle y^2 - x, 2y^2 \rangle$.
3. Sei $U = X$. Dann gilt immer $I_U = I_X = I$.

Definition 4.2. (aus Becker und Weispfennig [2, S.270]) Sei $I \subset \mathbb{P} = R[X]$ ein Ideal und $U \subseteq X$. Dann heißt U *unabhängig* modulo I , falls $I_U = \{0\}$ gilt.

Die *Dimension* von I ist dann definiert als

$$\dim(I) = \max \{ |U| \mid U \subseteq X \text{ unabhängig modulo } I \}.$$

Bemerkung. Die Dimension eines Ideals ist bei der Bestimmung der Varietät wichtig, da wir mit den Methoden in diesem Text nur nulldimensionale Ideale lösen können.

4.2 Normalformraum

Definition 4.3. Sei I ein Ideal von Polynomen aus \mathbb{P} . Man nennt \mathbb{P}/I einen *Faktorring* oder auch *Quotientenring*, auf dem die Operationen $+$ und \cdot mit

$$\begin{aligned} [a] + [b] &:= [a + b], \\ [a] \cdot [b] &:= [a \cdot b] \end{aligned}$$

definiert sind, wobei $[a], [b] \in \mathbb{P}/I$.

Anschaulich betrachtet ist dies der Ring, in dem alle Reste liegen, die bei der Division eines beliebigen Polynoms aus \mathbb{P} durch I beziehungsweise F mit $I = \langle F \rangle$ entstehen.

Definition 4.4. Sei F eine Idealbasis und G die zugehörige Gröbner- beziehungsweise H-Basis. Dann heißt $NF(F) := \mathbb{P}/G$ der *Normalformraum* von F beziehungsweise G .

Da jedes Ideal I endlich erzeugt werden kann, sagen wir auch $NF(I)$ anstelle von $NF(F)$, wobei F die Idealbasis von I ist.

Bei Gröbner-Basen ist dieser Raumes wie folgt definiert:

$$NF(G) := \langle \langle p \in \mathbb{P}^* \mid p \notin \langle LT(G) \rangle \rangle \rangle,$$

wobei \mathbb{P}^* die Menge aller Monome ist.

Für H-Basen ergibt sich eine einfachere Definition:

$$NF(G) := W(G) = \bigcup_{d \in \mathbb{N}_0} W_d(G).$$

Das Problem bei der Berechnung einer Basis des Normalformraumes ist, daß diese nur für nulldimensionale Ideale endlich ist. Bisher wurde noch kein Algorithmus gefunden, der diese Basis für nicht-nulldimensionale Ideale berechnen kann.

Beispiel. Sei $F = (x - z^2, x - y^2 + z, y - x) \subset \mathbb{Z}[x, y, z]$. Eine H-Basis hiervon ist $H = (-z^2 + y, -y^2 + y + z, -x + y)$. Wenn wir uns nun $W(H)$ anschauen, erhalten wir:

$$\begin{aligned} d < 0 : W_d &= \emptyset \\ d = 0 : W_d &= \langle 1 \rangle \\ d = 1 : W_d &= \langle z, x + y \rangle \\ d = 2 : W_d &= \langle xz + yz \rangle \\ d > 2 : W_d &= \emptyset \end{aligned}$$

Es ist also $NF(F) = (1, z, x + y, xz + yz)$.⁷

⁷Die Zuordnung von d nach $\dim W_d$ beschreibt die homogene Hilberfunktion $H_I(d)$ von I (siehe Möller und Sauer [8, S.341 f]).

Punkt (2) zur Erkennung einer H-Basis aus Kapitel 3.4 führt zu folgender Hypothese:

Hypothese. Ist G eine Gröbner- oder H-Basis und haben die (homogenen) Leiterterme von G nur den Punkt 0 gemeinsam, dann ist das von G erzeugte Ideal $\langle G \rangle$ nulldimensional. Weiter gilt, daß die maximale Anzahl der unabhängigen Variablen der Lösung von $LT(G) = 0$ beziehungsweise $\Lambda(G) = 0$ der Dimension des Ideals $\langle G \rangle$ entsprechen.

Beispiele.

1. Sei $F := (x^2 - y, y + z, 2xy - z^2)$. Eine Gröbner-Basis hiervon ist $G = (x^2 + z, y + z, -2xz - z^2, z^3 + 4z^2)$. Die Menge der Leiterterme ist $LT(G) = (x^2, y, -2xz, z^3)$. Setzt man diese gleich 0, erhält man als einzige Lösung $\{x = 0, y = 0, z = 0\}$. Das heißt, das Ideal $\langle F \rangle$ ist nulldimensional.
2. Sei $F := (x - y, y^2 + z^2)$, was schon eine Gröbner-Basis ist. Es ist $LT(F) = (x, y^2)$. Setzt man dies gleich 0, erhält man die Lösung $\{x = 0, y = 0, z = z\}$ mit der freien Variable z . Das heißt, das Ideal $\langle F \rangle$ ist eindimensional.
3. Sei $F := (x^2 - wy, z^2 + y^2)$. Dies entspricht der H-Basis und zusätzlich der Menge der homogenen Leiterterme. Setzt man dies gleich 0, erhält man drei Lösungen:

$$\{w = w, x = 0, y = 0, z = 0\},$$

$$\left\{w = \frac{-ix^2}{z}, x = x, y = iz, z = z\right\}, \left\{w = \frac{ix^2}{z}, x = x, y = -iz, z = z\right\}.$$

In der ersten Menge gibt es eine freie Variable. In den anderen beiden je zwei. Das bedeutet, daß das Ideal $\langle F \rangle$ zweidimensional ist, da $\max(1, 2, 2) = 2$.

Bemerkung. Für Gröbner-Basen wurde zumindest der Zusammenhang mit der Nulldimensionalität in Becker und Weispfennig [2, S.271 ff] bewiesen. Für H-Basen und die allgemeine Dimension des Ideals fehlt der Beweis leider noch. Man hätte damit aber eine leichte Überprüfung der Dimension eines Ideals, falls man eine Gröbner- oder H-Basis hierzu bestimmen kann.

Gehen wir also davon aus, daß das Erzeugnis von F ein nulldimensionales Ideal ist. Dann hat es die folgende Eigenschaft:

- Die Anzahl der Basiselemente in des Normalformraums $NF(F)$ ist größer oder gleich der Anzahl der Lösungen in der Varietät von $\langle F \rangle$.

Wir können also leicht im Voraus bestimmen, wie viele Lösungen wir bei einem bestimmten Problem maximal erhalten werden. Die Relation "größer" ist erforderlich, da die Varietät auch mehrfache Nullstellen haben kann. Gleichheit erhält man nur bei einfachen Nullstellen.

Beispiel. Nehmen wir $F := (x^3) \subset \mathbb{Z}[x]$ an. Dann ist die einzige Lösung in der Varietät sicherlich nur 0, dies aber als dreifache Nullstelle. Der Normalformraum $\langle\langle 1, x, x^2 \rangle\rangle$ hat dementsprechend drei Basiselemente.

Bemerkung. Um dem Auftreten mehrfacher Nullstellen entgegenzuwirken, kann man zuerst das *Radikal*

$$\sqrt{\langle F \rangle} = \{f \in \mathbb{P} \mid f^k \in \langle F \rangle \text{ für ein } k \in \mathbb{N}\}$$

berechnen, bei dem nur noch einfache Nullstellen übrig bleiben. Hierauf wollen wir zwar nicht näher eingehen, zur Vereinfachung betrachten wir ab sofort aber nur noch Radikale.⁸

Nun wollen wir uns langsam an das Bestimmen der Varietät eines Ideals heranwagen. Hierfür benötigen wir sogenannte "Multiplikationstabellen".

4.3 Multiplikationstabellen

Betrachten wir zuerst die Funktion

$$\phi_f(p) = (p \cdot f) \pmod I$$

mit $p \in NF(I)$. Das Bild von ϕ_f liegt also wieder in $NF(I)$. Außerdem ist die Abbildung linear.

Die Multiplikationstabellen sind wie folgt definiert.

⁸Eine genauere Beschreibung und Bestimmung des Radikals findet man in Sauer [12, S.123 ff].

Definition 4.5. Ist $P = (p_1, \dots, p_r)$ eine Basis des Normalformraumes von I , dann ist die quadratische Matrix $M(f) = (m_{ij})_{i,j=1}^r$, die durch die Gleichungen

$$\phi_f(p_i) = \sum_{j=1}^r m_{ij} \cdot p_j, \quad i = 1, \dots, r,$$

bestimmt wird, die *Multiplikationstabelle* von P bezüglich Multiplikation mit f .

Beispiel. Sei wieder $F = (x - z^2, x - y^2 + z, y - x) \subset \mathbb{Z}[x, y, z]$ und die H-Basis $H = (-z^2 + y, -y^2 + y + z, -x + y)$. Wir wollen nun die Multiplikationstabelle von $P = NF(F) = (1, z, x + y, xz + yz)$ bezüglich $f = x^2 - 2z$ berechnen.

Durch die Funktion ϕ_f erhalten wir vier Gleichungen:

$$\begin{aligned} \phi_f(1) &= 1 \cdot f \pmod H = \frac{x+y}{2} + 3z \\ \phi_f(z) &= z \cdot f \pmod H = \frac{xz + yz + 3x + 3y}{2} \\ \phi_f(x+y) &= (x+y) \cdot f \pmod H = 3xz + 3yz + x + y + 2z \\ \phi_f(xz + yz) &= (xz + yz) \cdot f \pmod H = xz + yz + 4x + 4y + 6z \end{aligned}$$

Hierüber erhalten wir dann die Koeffizienten für die Matrix $M(f)$ aus den Gleichungen

$$\begin{aligned} \frac{x+y}{2} + 3z &= m_{1,1} + m_{1,2} \cdot (x+y) + m_{1,3} \cdot z + m_{1,4} \cdot (xz + yz) \\ &\Rightarrow m_{1,1} = 0, m_{1,2} = 3, m_{1,3} = 1/2, m_{1,4} = 0 \\ \frac{xz + yz + 3x + 3y}{2} &= m_{2,1} + m_{2,2} \cdot (x+y) + m_{2,3} \cdot z + m_{2,4} \cdot (xz + yz) \\ &\Rightarrow m_{2,1} = 0, m_{2,2} = 0, m_{2,3} = 3/2, m_{2,4} = 1/2 \\ 3xz + 3yz + x + y + 2z &= m_{3,1} + m_{3,2} \cdot (x+y) + m_{3,3} \cdot z + m_{3,4} \cdot (xz + yz) \\ &\Rightarrow m_{3,1} = 0, m_{3,2} = 2, m_{3,3} = 1, m_{3,4} = 3 \\ xz + yz + 4x + 4y + 6z &= m_{4,1} + m_{4,2} \cdot (x+y) + m_{4,3} \cdot z + m_{4,4} \cdot (xz + yz) \\ &\Rightarrow m_{4,1} = 0, m_{4,2} = 6, m_{4,3} = 4, m_{4,4} = 1 \end{aligned}$$

Damit erhalten wir die Multiplikationstabelle $M(f) = \begin{pmatrix} 0 & 3 & 1/2 & 0 \\ 0 & 0 & 3/2 & 1/2 \\ 0 & 2 & 1 & 3 \\ 0 & 6 & 4 & 1 \end{pmatrix}$.

4.4 Berechnung der Varietät

Das Wichtigste bei der Bestimmung der Varietät ist das folgende (abgewandelte) Theorem von Möller und Stetter [9, S.322].

Theorem 4.6. Sei I ein nulldimensionales Radikal, $P = (p_1, \dots, p_m)$ eine Basis des Normalformraums $NF(I)$, $V(I) = \{v_1, \dots, v_m\}$ die Varietät von I und $M(f)$ die Multiplikationstabelle bezüglich $f \in \mathbb{P}$. Dann sind $f(v_1), \dots, f(v_m)$ die Eigenwerte von $M(f)$.

Um dieses Theorem zu beweisen, benötigen wir aber noch eine kleine interpolatorische Hilfe, die über folgenden Satz aus Sauer [12, S.120 f] gegeben ist.

Satz 4.7. Seien I , P , $NF(I)$ und $V(I)$ wie in Theorem 4.6 gegeben. Dann gelten folgende Aussagen:

1. $NF(I)$ ist ein Interpolationsraum. Das heißt für jedes $f \in \mathbb{P}$ existiert genau ein $g \in NF(I)$ mit $f(v_i) = g(v_i)$ für alle $i = 1, \dots, m$.
2. Es existieren $l_i \in NF(I)$, so daß $l_i(v_j) = \delta_{i,j}$ für alle $i, j = 1, \dots, m$.
3. Für $f \in \mathbb{P}$ ist

$$L_I(f) := \sum_{i=1}^m f(v_i) \cdot l_i$$

das (Lagrange-)Interpolationspolynom von f .

4. Die l_i bilden eine Basis von $NF(I)$.

Beweis:

1. Sei $f \in \mathbb{P}$ und $r := f \bmod I \in NF(I)$, das heißt r ist der Rest bei Division von f durch I . Es ist dann $f - r \in I$ und damit gilt $(f - r)(v_i) = 0$ für alle $i = 1, \dots, m$. Damit wird f durch r an den Stellen v_i interpoliert.

Nehmen wir nun an, daß es $r, r' \in NF(I)$ gibt mit $(f - r)(v_i) = 0 = (f - r')(v_i)$ für alle $i = 1, \dots, m$. Das bedeutet also, daß $(r - r')(v_i) = 0$. Damit ist aber $r - r' \in I$ und muß bei Division durch I den Rest 0 lassen. Da die Division eine lineare Abbildung ist, gilt nun $0 = (r - r') \bmod I = r \bmod I - r' \bmod I = r - r'$ und daraus folgt $r = r'$, was die Eindeutigkeit zeigt.

2. Wir setzen

$$l_i(x) := \prod_{j=1, i \neq j}^m \frac{x - v_j}{v_i - v_j}$$

für alle $i = 1, \dots, m$. Die Polynome l_i interpolieren damit eindeutig $\delta_{i,j}$.

3. Dies folgt sofort aus der Definition der l_i .

4. Sei $g \in NF(I)$. Dann läßt sich dies darstellen durch

$$g = L_I(g) = \sum_{i=1}^m g(v_i) \cdot l_i,$$

was aber nichts anderes heißt, daß die l_i eine Basis von $NF(I)$ bilden. \square

Damit können wir leicht obiges Theorem beweisen.

Beweis des Theorems: Wir müssen zeigen, daß

$$M(f) \cdot l_i = f(v_i) \cdot l_i \quad \forall i = 1, \dots, m,$$

wobei l_i die Eigenvektoren⁹ zu den Eigenwerten $f(v_i)$ sind. Aus Punkt 4 des obigen Satzes wissen wir, daß die l_i (als Polynome) eine Basis von $\langle\langle P \rangle\rangle$ bilden. Da die Multiplikationstabelle $M(f)$ also durch $\phi_f(l_i)$ bestimmt ist, reicht es zu zeigen, daß

$$\phi_f(l_i) = f(v_i) \cdot l_i \quad \forall i = 1, \dots, m.$$

Betrachten wir nun $\phi_{f-f(v_i)}(l_i)$ für ein $i \in \{1, \dots, m\}$. Es ist dann

$$\begin{aligned} \phi_{f-f(v_i)}(l_i) &= (f - f(v_i)) \cdot l_i \pmod I \\ &= (f \cdot l_i - f(v_i) \cdot l_i) \pmod I \\ &= f \cdot l_i \pmod I - f(v_i) \cdot l_i && \text{- da } f(v_i) \cdot l_i \in NF(I) \\ &= \sum_{j=1}^m (f \cdot l_i)(v_j) \cdot l_j - f(v_i) \cdot l_i && \text{- nach Satz 4.7, Punkt 3} \\ &= f(v_i) \cdot l_i - f(v_i) \cdot l_i && \text{- nach Satz 4.7, Punkt 2} \\ &= 0. \end{aligned}$$

⁹Dies ist gleichbedeutend mit den Koeffizientenvektoren der Polynome l_i aus Satz 4.7.

Das heißt also $\phi_{f-f(v_i)}(l_i) = \phi_f(l_i) - f(v_i) \cdot (l_i) = 0$ und daraus folgt $\phi_f(l_i) = f(v_i) \cdot (l_i)$ wie gewünscht. \square

Setzt man nun für f die Unbekannten $x_i, i = 1, \dots, n$, ein, so erhält man die Aussage, daß die Eigenwerte der Matrix $M(x_i)$ die Lösungen aus der Varietät sind.

In unserem Fall sind die verschiedenen Lösungen/Eigenwerte aller Matrizen $M(x_i)$ über folgende Eigenschaft verbunden:

- Seien $M(x_i)$ die Multiplikationstabellen bezüglich der Unbekannten $x_i, i = 1, \dots, n$. Seien $e_i^{(j)}, j = 1, \dots, k_i$ die Eigenwerte zu $M(x_i)$ und seien $E_i^{(j)}, j = 1, \dots, k_i$, die zugehörigen Eigenräume. Dann gilt:

$$X := (e_1^{m_1}, \dots, e_n^{m_n}) \text{ ist eine Lösung von } F(X) = 0$$

$$\Leftrightarrow \dim(E_1^{m_1} \cup \dots \cup E_n^{m_n}) < \dim(E_1^{m_1}) + \dots + \dim(E_n^{m_n}),$$

mit $m_i \in \{1, \dots, k_i\} \forall i = 1, \dots, n$.

Beispiel. Sei $F := (xy - x, x^2 - y^2) \subset \mathbb{Z}[x, y]$. Das Radikal und gleichzeitig eine H-Basis ist mit $H = (xy - x, x^2 - x, y^2 - x)$ gegeben. Wir wollen die Multiplikationstabellen von $P = NF(H) = (1, x, y)$ bezüglich der Unbekannten x und y berechnen.

$$\text{Wir erhalten } M(x) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \text{ und } M(y) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Die Eigenwerte und Eigenräume von $M(x)$ sind

$$e_x^1 = -1, E_x^1 = \left\langle \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \right\rangle,$$

$$e_x^2 = 0, E_x^2 = \left\langle \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle,$$

$$e_x^3 = 1, E_x^3 = \left\langle \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\rangle.$$

Die Eigenwerte und Eigenräume von $M(y)$ sind

$$e_y^1 = 0, E_y^1 = \left\langle \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle,$$

$$e_y^2 = 1, E_y^2 = \left\langle \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\rangle.$$

Wenn wir die Eigenwerte von $M(x)$ und $M(y)$ kreuzweise verbinden, erhalten wir mit

$$S' := ([-1, 0], [-1, 1], [0, 0], [0, 1], [1, 0], [1, 1])$$

die Menge aller potentiellen Lösungen von $H(X) = 0$.

Zum Filtern der wirklichen Lösungen vergleichen wir die Dimension der zugehörigen vereinigten Eigenräume und die Summe der Einzeldimensionen. Es ergibt sich

$$\begin{aligned} \dim(E_x^1 \cup E_y^1) &= 2 = 2 = \dim(E_x^1) + \dim(E_y^1), \\ \dim(E_x^1 \cup E_y^2) &= 2 < 3 = \dim(E_x^1) + \dim(E_y^2), \\ \dim(E_x^2 \cup E_y^1) &= 1 < 2 = \dim(E_x^2) + \dim(E_y^1), \\ \dim(E_x^2 \cup E_y^2) &= 3 = 3 = \dim(E_x^2) + \dim(E_y^2), \\ \dim(E_x^3 \cup E_y^1) &= 2 = 2 = \dim(E_x^3) + \dim(E_y^1), \\ \dim(E_x^3 \cup E_y^2) &= 2 < 3 = \dim(E_x^3) + \dim(E_y^2). \end{aligned}$$

Durch das Kriterium von oben haben wir also mit $S := ([-1, 1], [0, 0], [1, 1])$ unsere drei wirklichen Lösungen gefunden.

Hiermit haben wir also alle mathematischen Werkzeuge zusammen, um die Varietät eines (nulldimensionalen) Ideals zu berechnen.

Teil III

Algorithmen und Analyse

1 Einführung

Zuerst wollen wir einige Definitionen und Bezeichnungen wiederholen beziehungsweise einführen, die wir als Kurzschreibweise in den Algorithmen benutzen werden.

Fortan bezeichne $\{\dots\}$ ungeordnete Mengen und $[\dots]$ geordnete Mengen. Oft wird hier auch der Begriff "Liste" für geordnete Mengen benutzt.

Sei $U := [x_1, \dots, x_n]$ die Menge aller n Unbekannten, mit denen wir rechnen wollen. Diese Menge ist geordnet, da sie wiedergibt, in welcher Reihenfolge die Unbekannten geordnet und sortiert werden sollen. Im Programm wird diese Menge mit `List_Undets` bezeichnet und mittels `setUndets([x1, ..., xn])` zugewiesen¹⁰.

Analog muß man feststehende Parameter a_1, \dots, a_n vor der Berechnung als Konstanten deklarieren. Dies geschieht mittels `setConsts([a1, ..., an])`.

Sei R ein Ring, so bezeichne $\mathbb{P} := R[U] = R[x_1, \dots, x_n]$ wieder den Polynomring über R in den Unbekannten x_1, \dots, x_n . Sei $F := [f_1, \dots, f_s]$ eine Menge von Polynomen aus \mathbb{P} . Das von F erzeugte Ideal $I = \langle F \rangle$ ist dann $I = \{\sum_{i=1}^s f_i \cdot p_i \mid p_i \in \mathbb{P} \text{ beliebig}\}$.

Ist $f \in \mathbb{P} \setminus \{0\}$, so bezeichne $\delta(f)$ den Totalgrad von f und $\Lambda(f)$ den homogenen Leitterm von f . Ist I ein Ideal, so ist $\Lambda(I) = \{\Lambda(f) \mid f \in I\}$. Ist F eine Menge von Polynomen, so ist $\Lambda(F) := [\Lambda(f_1), \dots, \Lambda(f_s)]$.

Für Gröbner-Basen wiederholen wir die Definition. Es ist

- $\text{deg}(f)$ = Grad von f geschrieben als Multiindex α ,
- $LM(f)$ = Leitmonom von f ,
- $LK(f)$ = Leitkoeffizient von f und
- $LT(f)$ = Leitterm von f , das heißt, $LT(f) = LK(f) \cdot LM(f)$.

¹⁰Dabei sollte man natürlich keine "... " verwenden, sondern die Unbekannten komplett angeben.

2 Gröbner-Basen

In diesem Abschnitt werden wir nur die wichtigsten drei Algorithmen für Gröbner-Basen aufzeigen, da der Schwerpunkt der Arbeit bei H-Basen liegt. Das Kapitel umfaßt den Divisionalgorithmus, den Buchberger-Algorithmus und einen Reduktionsalgorithmus.

2.1 Divisionsalgorithmus

Der Divisionalgorithmus aus Teil II.2.2 von Seite 9 wird eigentlich "Eins-zu-Eins" übernommen. Mit der Eingabe einer Idealbasis $F = [f_1, \dots, f_s]$ und einem Polynom g erhalten wir so eine Liste mit Koeffizienten $h_i, i = 1, \dots, s$, und Rest r , so daß

$$g = \sum_{i=1}^s h_i f_i + r.$$

Aufruf: $\text{gDivAlg}(F, g)$;

Algorithmus 2.1

1. Erstelle Liste $C := [0, \dots, 0]$ mit $s+1$ Einträgen.
2. Setze Polynom $t := g$.
3. Solange $t \neq 0$:
 - Setze $k := 1$ und $dva := false$.
 - Solange $k \leq s$ und $dva = false$:
 - Wenn $LM(f_i) \mid LM(t)$, dann setze

$$C[i] := C[i] + LT(t)/LT(f_i),$$

$$t := t - (LT(t)/LT(f_i)) f_i,$$

$$dva := true.$$
 - Sonst setze $k := k + 1$.
 - Wenn $dva = false$, dann setze

$$C[s + 1] := C[s + 1] + LT(p) \text{ und } t := t - LT(t)$$

Beispiele hierzu haben wir schon in Teil II gezeigt, weswegen wir diese hier nicht erneut aufgreifen wollen.

Bemerkung. Die implementierten Funktionen `gcfs(F,g)` und `grmd(F,G)` geben dabei nur die Koeffizienten beziehungsweise nur den Rest zurück.

2.2 Buchbergers Algorithmus

Zuerst wollen wir nur kurz daran erinnern, wie wir das S-Polynom zu zwei gegebenen Polynomen f und g berechnen können. Der Algorithmus dafür wird hier nicht extra aufgeführt, da er recht simpel ist.

Aufruf: `gSyzSet(f,g)`;

Wir erhalten damit zwei Polynome h_1 und h_2 , so daß gilt:

$$S(f,g) = h_1 \cdot f - h_2 \cdot g,$$

wobei einfach $h_1 := \frac{x^\gamma}{LT(f)}$ und $h_2 := -\frac{x^\gamma}{LT(g)}$ mit $x^\gamma = kgV(LT(f), LT(g))$.

Um uns die Arbeit etwas zu vereinfachen, definieren wir noch eine Menge S , die alle diese Paare von Polynomen enthält.

Aufruf: `gSyzSets(F)`;

Nun können wir die Gröbner-Basis berechnen. Aus einem speziellen Grund, den wir später genauer ausführen werden¹¹, werden wir hier nicht den Algorithmus 2.22 aus Teil II übernehmen. Wir berechnen stattdessen eine Matrix M , so daß

$$G = F \cdot M,$$

wobei F unsere Ausgangsbasis für das Ideal und G die zugehörige reduzierte Gröbner-Basis ist.

¹¹Siehe dazu Kapitel 4.2 auf Seite 63.

Aufruf: `IGBMatrix(F)`;

Algorithmus 2.2

1. Setze $G := F$ und $H := \emptyset$.
2. Erstelle Einheitsmatrix $M \in R^{s \times s}$.
3. Reduziere G und M passend.¹²
4. Solange $G \neq H$:
 - Setze $H := G = [g_1, \dots, g_t]$ und $i := 0$.
 - Solange $i < |G|$:
 - Setze $i := i + 1$ und $j := i$.
 - Solange $j < |G|$:
 - * Setze $j := j + 1$ und Matrix $N := ()$.
 - * Setze $g := S(g_i, g_j)$.
 - * Berechne Menge $R := [r_1, \dots, r_t, r] = \mathbf{gDivAlg}(G, h)$ mit

$$h = \sum_{i=1}^t r_i g_i + r.$$

- * Wenn $r \neq 0$, dann:
 - Berechne eine Vektor v mit $v_i = \mathbf{gSyzSet}(g_{-i}, g_{-j})_1 - r_i$,
 - $v_j = \mathbf{gSyzSet}(g_{-i}, g_{-j})_2 - r_j$ und $v_k = -r_k$ für $i \neq k \neq j$.
 - Füge den Vektor $M \cdot v$ als Spaltenvektor zu M und r zu G hinzu.
 - * Reduziere G und M passend.

Bemerkung. Die Menge G in unserem Algorithmus ist dabei natürlich immer die gesuchte Gröbner-Basis, die man aber mittels der Matrix M leicht durch eine einzelne Matrix/Vektor-Multiplikation berechnen kann. Der Befehl dafür lautet `GBasis(F)`.

Nun berechnen wir das ganze noch einmal umgekehrt, das heißt, wir suchen eine Matrix N mit

$$F = G \cdot N,$$

¹²Siehe dazu Algorithmus 2.5 auf Seite 47.

wobei G die gegebene Gröbner-Basis ist und $F = [f_1, \dots, f_s]$ die Ausgangsbasis unseres Ideals.

Aufruf: `GBMatrix(F)`;

Algorithmus 2.3 (nach [3, Buchberger, S.217])

1. Berechne $G = \text{GBasis}(F)$.
2. Berechne Matrix N mit Spaltenvektoren $n_i = \text{gcfs}(G, f_i)$ für $i = 1, \dots, s$

Die letzte Matrix R , die wir hier berechnen wollen, soll die lineare, homogene Gleichung

$$R \cdot G = 0$$

lösen, wobei G eine Gröbner-Basis zu einer Idealbasis F ist.

Aufruf: `gSMatrix(F)`;

Algorithmus 2.4 nach Buchberger [3, S.218]

1. Berechne $G := \text{GBasis}(F) = [g_1, \dots, g_t]$ und $S := \text{gSyzSets}(G)$.
2. Setze $R := \emptyset$ und $k := 0$.
3. Schleife i von 1 bis t :
 - Schleife j von $i + 1$ bis t :
 - Setze $k := k + 1$.
 - Berechne $h := \text{gSyzPoly}(g_i, g_j)$.
 - Setze Vektor $v := S[k] - \text{gcfs}(G, h)$.
 - Füge v als Zeilenvektor zu R hinzu.

2.3 Reduktionsalgorithmus

Nun fehlt uns eigentlich nur noch der Reduktionsalgorithmus. Dieser besteht aus zwei Teilen: Zuerst wird festgestellt, ob eine Menge $F = [f_1, \dots, f_s]$ überhaupt reduziert werden

kann und wenn ja, welches Element auf welche Art und Weise reduziert wird. Danach findet die eigentliche Reduktion statt.

Aufruf: `canReduce(F, t);`

Algorithmus 2.5

1. Setze $i := t + 1$ und erstelle einen Vektor $r = 0$ mit s Einträgen¹³.
2. Solange $1 < i$ und $r_1 = 0$:
 - Setze $i := i - 1$.
 - Setze $r := 0$.
 - Setze $H := [f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_s]$.
 - Setze $(r_j)_{j=2}^s := \text{gcfs}(H, f_i)$.
 - Wenn $H^T \cdot (r_j)_{j=2}^s \neq 0$, dann setze $r_1 := i$.

Bemerkung. Die übergebene Variable t gibt dabei nur an, ab welchem Element von hinten nach vorne reduziert werden soll. Bei mehrfacher Reduktion erspart dies die erneute Überprüfung von bereits reduzierten Elementen.

Da dieser Algorithmus in Teil II nicht vorgestellt wurde, wird er etwas genauer erklärt:

Der Vektor $r \in \mathbb{N}_0 \times \mathbb{P}^{s-1}$ wird am Ende des Algorithmus zurückgegeben. Komponente r_1 gibt an, welches Element in einer Menge reduziert werden kann. Die Werte r_2, \dots, r_s sind die Koeffizienten der Division des zu reduzierenden Polynoms f_i mit der Menge $H := F \setminus \{f_i\}$.

Es gilt nun

$$0 \neq H^T \cdot \text{gcfs}(H, f) = f - \text{grmd}(H, f) \quad \text{für alle } f \in \mathbb{P}$$

$$\Leftrightarrow \text{grmd}(H, f) \neq f.$$

Wir überprüfen also nur, ob der Rest von f_i bei Division durch H ungleich f_i ist, was bedeutet, daß wir den Grad von f_i etwas reduzieren können ohne das Ideal zu verändern. Es ist außerdem klar, daß der Algorithmus terminiert, da durch die Dekrementierung von i irgendwann eine untere Grenze erreicht ist.

¹³Dabei ist 0 hier als Nullvektor zu verstehen.

Beispiel. Sei $F := [x^2y, xy + 1, y] \subset \mathbb{Z}[x, y]$. $\text{canReduce}(F, 3)$ ergibt dann

- $i := 4, r = (0, 0, 0)^T$
- $1 < i = 4$ und $r_1 = 0 \Rightarrow$
 - $i := 3, r := 0, H := [x^2y, xy + 1, y] \setminus [y] = [x^2y, xy + 1]$
 - $r := (0, \text{gcf}(H, y)) = (0, 0, 0)$, da $LT(y)$ sich nicht mit den Leittermen der beiden anderen Polynomen darstellen läßt.
 - $H^T \cdot (r_j)_{j=2}^s = x^2y \cdot 0 + (xy + 1) \cdot 0 = 0$
- $1 < i = 3$ und $r_1 = 0 \Rightarrow$
 - $i := 2, r := 0, H := [x^2y, xy + 1, y] \setminus [xy + 1] = [x^2y, y]$
 - $r := (0, \text{gcf}(H, xy+1)) = (0, 0, x)$, da $LT(xy + 1) = xy = 0 \cdot x^2y + x \cdot y$.
 - $H^T \cdot (r_j)_{j=2}^s = x^2y \cdot 0 + y \cdot x = xy \neq 0 \Rightarrow$
 $r_1 := 2$, also $r = (2, 0, x)$.
- $1 < i = 2$ und $r_1 = 2 \neq 0 \Rightarrow$ Ende.

Wir wissen also nun, daß wir Element 2 in unserer Menge F reduzieren können und haben sogar die Koeffizienten für die Reduktion. Dies muß man nun so oft wiederholen, bis nichts mehr reduziert werden kann. Dazu hilft uns der nächste Algorithmus.

Aufruf: $\text{reduceSet}(F)$;

Algorithmus 2.6

1. Setze $H := F$ mit $H = [h_1, \dots, h_t]$ und bestimme Vektor $r = \text{canReduce}(H, t)$.
2. Setze $i := r_1$.
3. Solange $i \neq 0$:
 - Setze Vektor $v := [h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_t]$.
 - Berechne Polynom $m := h_i - v^T \cdot (r_j)_{j=2}^t$.
 - Ist $m \neq 0$, setze $H := [h_1, \dots, h_{i-1}, m, h_{i+1}, \dots, h_t]$,
sonst setze $H := [h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_t]$.

- Setze $i := i - 1$.
- Ist $i = 0$, setze $i := t$.
- Bestimme Vektor $r = \text{canReduce}(H, t)$ und setze $i := r_1$.

Zuerst müssen wir uns klar werden, daß dieser Algorithmus terminiert, das heißt, wir müssen zeigen, daß i irgendwann Null sein muß. Nehmen wir also an, dies ist nicht so. Aus der Gradreduktion des Reduktionsalgorithmus 2.5 folgt unmittelbar, daß $\deg(m) < \deg(h_i)$, da grundsätzlich mindestens der Leitern von h_i subtrahiert wird. Da $i \neq 0$ für jeden Schritt, muß irgendwann $r = 0$ gelten. Da dies ebenfalls für alle Polynome in H gilt, ist irgendwann $H := [h]$. Der Divisionsalgorithmus von h durch $H \setminus [h]$ liefert aber nur Koeffizienten 0 (beziehungsweise liefert gar keine Koeffizienten mehr). Daraus folgt $i = 0$, was im Widerspruch zur Annahme steht.

Es ist klar, daß der Algorithmus zu einer reduzierten Menge führt.

Bemerkung. Es sei hier noch gesagt, daß es sinnvoll ist, nach jedem Hinzufügen eines Polynoms zur Gröbner-Basis diesen Reduktionsalgorithmus anzuwenden. Beispiele haben gezeigt, daß in den Zwischenschritten die Basis das Dreifache der endgültigen Anzahl an Polynomen haben kann. Dadurch muß man circa das Neunfache an Polynomdivisionen durchführen, was den GB-Algorithmus entsprechend verlangsamt. Eine Reduktion nach jedem Hinzufügen hält die Basis dagegen immer in relativer Nähe zur endgültigen Anzahl.

Beispiel. Wir setzen unser Beispiel von oben fort. Sei also wieder $F := [x^2y, xy + 1, y]$.

- $H := F = [x^2y, xy + 1, y]$, $r := \text{canReduce}(H, 3) = [2, 0, x]^T$, $i := r_1 = 2$.
- $i \neq 0 \Rightarrow$
 - $v := H \setminus [h_2] = [x^2y, y]^T$ und $m = xy + 1 - [x^2y, y] \cdot [0, x]^T = xy + 1 - xy = 1$
 - $m \neq 0 \Rightarrow H := [x^2y, 1, y]$
 - $i := 3 - 1 = 2$ und $r := \text{canReduce}(H, 2) = [1, x^2y, 0]^T$, $i := r_1 = 1$
- $i \neq 0 \Rightarrow$
 - $v := H \setminus [h_1] = [1, y]^T$ und $m = x^2y - [1, y] \cdot [x^2y, 0]^T = x^2y - x^2y = 0$
 - $m = 0 \Rightarrow H := [1, y]$
 - $i := 1 - 1 = 0 \Rightarrow i := 2$ und $r := \text{canReduce}(H, 2) = [2, y]^T$, $i := r_1 = 2$

- $i \neq 0 \Rightarrow$
 - $v := H \setminus [h_2] = [1]^T$ und $m = y - [1] \cdot [y]^T = y - y = 0$
 - $m = 0 \Rightarrow H := [1]$
 - $i := 2 - 1 = 1$ und $r := \text{canReduce}(H, 1) = [0]^T$, $i := r_1 = 0$
- $i = 0 \Rightarrow$ Ende.

Nun haben wir die wichtigen Algorithmen für Gröbner-Basen erfaßt und wenden uns den H-Basen zu.

3 Berechnung der linearen Räume

3.1 Der Raum P_d

Zur Erinnerung: Es ist $P_d := \{p \mid p \text{ ist ein homogenes Polynom aus } \mathbb{P} \text{ vom Grad } d\}$. Da wir hier natürlich nicht alle Polynome, sondern nur eine Basis dieses Raums berechnen, sei im Folgenden P_d immer als Basis zu verstehen.

Aufruf: Pspace(d);

Algorithmus 3.1

1. Ist $d < 0$ setze $G := \emptyset$, gehe zum letzten Punkt.
2. Wurde P_d schon einmal berechnet, setze $G := P_d$ und gehe zum letzten Punkt.
3. Ist $d = 0$, setze $G := \{1\}$ und gehe zum letzten Punkt.
4. Setze $G := P_{d-1}$.
5. Setze $H := \emptyset$.
6. Schleife j von 1 bis n :
 - Schleife k von 1 bis $|G|$:
 - Setze $p := g_k \cdot x_j$, wobei g_k das k -te Element in G ist.
 - Wenn $p \notin H$, dann $H := H \cup \{p\}$.
7. $G := H$.
8. $P_d := G$.

Beispiel. Sei $d = 2$ und $U = [x, y, z]$. Wir nehmen an, daß für $i < d$ P_i schon berechnet wurde.

- $G := P_{d-1} = P_1 = \{x, y, z\}$
- $H := \emptyset$

- $j = 1/3$:
 - $p := x \cdot x = x^2 \notin H \Rightarrow H := \{x^2\}$
 - $p := y \cdot x = xy \notin H \Rightarrow H := \{x^2, xy\}$
 - $p := z \cdot x = xz \notin H \Rightarrow H := \{x^2, xy, xz\}$
- $j = 2/3$:
 - $p := x \cdot y = xy \in H$
 - $p := y \cdot y = y^2 \notin H \Rightarrow H := \{x^2, xy, xz, y^2\}$
 - $p := z \cdot y = yz \notin H \Rightarrow H := \{x^2, xy, xz, y^2, yz\}$
- $j = 3/3$:
 - $p := x \cdot z = xz \in H$
 - $p := y \cdot z = yz \in H$
 - $p := z \cdot z = z^2 \notin H \Rightarrow H := \{x^2, xy, xz, y^2, yz, z^2\}$
- $G := H = \{x^2, xy, xz, y^2, yz, z^2\}$.
- $P_2 := G = \{x^2, xy, xz, y^2, yz, z^2\}$

Laufzeit. Wichtig ist hierbei natürlich die Rekursion. Alle Versuche die Menge P_d direkt zu bestimmen, benötigten eine wesentlich längere Zeit zur Berechnung. So konnte die Dauer der Berechnung von P_{10} mit acht Unbekannten von 436 Sekunden auf 25 Sekunden verringert werden. Desweiteren wird in der Prozedur P_d global gespeichert. Das bedeutet, wird die Berechnung ein zweites Mal aufgerufen, wird auf das bereits bekannte Ergebnis zurückgegriffen, was keine Zeit mehr für die Berechnung erfordert. Man sollte natürlich beachten, daß sich P_d bei einer Änderung der Unbekannten ebenfalls ändert.

3.2 Der Raum $V_d(I)$

Es ist

$$V_d(I) := \left\{ \sum_{f \in I} q_f \Lambda(f) \mid q_f \in P_{d-\delta(f)} \right\}.$$

Wir berechnen wieder nur eine Basis beziehungsweise zuerst ein Erzeugendensystem von $V_d(I)$, weswegen V_d von nun an die Basis für diesen Raum beschreibt.

Aufruf: $V_{\text{space}}(d, F)$;

Algorithmus 3.2

1. Setze $T := \Lambda(F) = [t_1, \dots, t_s]$.
2. Setze $M := \max\{\delta(t_i) \mid i = 1, \dots, s\}$ und $m := \min\{\delta(t_i) \mid i = 1, \dots, s\}$.
3. Setze $G := \emptyset$.
4. Berechne $H_i := P_{d-i} = \{h_1^{(i)}, \dots, h_{k_i}^{(i)}\}$ für $i = m, \dots, M$.
5. Schleife i von 1 bis s :
 - $G := G \cup \bigcup_{j=1}^{k_{\delta(t_i)}} t_i \cdot h_{k_j}^{\delta(t_i)}$.
6. $V_d := G$.

Beispiel. Sei $F = [x^3 - xy^2 + 2x, y^2 - y + x]$, $U = [x, y]$ und $d = 3$.

- $T := \Lambda(F) = [x^3 - xy^2, y^2]$
- $M := 3 = \max\{3, 2\}$, $m := 2 = \min\{3, 2\}$
- $G := \emptyset$
- $H_2 := P_1 = \{x, y\}$, $H_3 := P_0 = \{1\}$
- $i = 1/2$:
 - $G := G \cup \{(x^3 - xy^2) \cdot 1\} = \{x^3 - xy^2\}$
- $i = 2/2$:
 - $G := G \cup \{y^2 \cdot x\} = \{x^3 - xy^2, xy^2\}$
 - $G := G \cup \{y^2 \cdot y\} = \{x^3 - xy^2, xy^2, y^3\}$
- $V_3 := G = \{x^3 - xy^2, xy^2, y^3\}$.

Laufzeit. Da die oben aufgeführten Mengen intern als Listen gespeichert werden, muß aufpaßt werden, daß keine Redundanzen entstehen. Sei F zum Beispiel eine Menge aus 100 Polynomen in den Unbekannten x, y , die alle Totalgrad 10 haben und aus jeweils 3 Monomen bestehen. Betrachtet man hier $\Lambda(F)$ als Liste, so erhält man logischerweise wieder 100 Polynome. Betrachtet man aber die Menge, was aber noch keine Basis ist, so sind es nur noch ca. 50. Dies wirkt sich natürlich bei allen nachfolgenden Berechnungen aus. Die Berechnung für $V_{200}(F)$ als Liste dauert so 13 Sekunden, als Menge nur 3,5 Sekunden, was eine Ersparnis von 76 Prozent ausmacht. Auch der Unterschied bei der Anzahl der Elemente ist beachtlich, so sind es in zweiten Fall nur 3670 Polynome, im ersten dagegen 19215.

Man kann die Anzahl der Elemente noch weiter reduzieren, in dem man weitere Abhängigkeiten sucht. Wenn V die Polynome $x^3 - xy$ und xy enthält, so kann man das zweite vom ersten subtrahieren und erhält immer noch den gleichen Raum. Der Algorithmus hierfür folgt.

Aufruf: `LinSpace(V)`;¹⁴

Algorithmus 3.3

1. Setze $l := 0, P := \emptyset, L := \emptyset, Z := \emptyset$.¹⁵
2. Schleife i von 1 bis k :
 - Setze $t := v_i, w := []$.¹⁶
 - Solange $t \neq 0$:
 - Setze $h := LM(t)$ und $c := LK(t)$ bezüglich einer monomialen Ordnung.
 - Ist $h \in P$, dann sei pos die Position von h in P . Setze dann $w_j := 0$ für $j = |w| + 1, \dots, pos - 1$ und $w_{pos} := c$.
 - Wenn $h \notin P$, dann setze $l := l + 1$. Setze $w_j := 0$ für $j = |w| + 1, \dots, l - 1$ und $w_l := c$. Außerdem setze $P := P \cup h$.
 - Setze $t := t - c \cdot h$.
 - Setze $Z := Z \cup \{w\}$.

¹⁴Wobei nun $V := V_d = \{v_1, \dots, v_k\}$ gilt.

¹⁵ P, L und Z sind dabei geordnete Menge.

¹⁶ w ist ein Vektor, der dynamisch wächst in Abhängigkeit von l .

3. Schleife i von 1 bis k :

- Setze $w := z_i$ und dann $w_j := 0$ für $j = |w| + 1, \dots, l$.
- Setze $z_i := w$.

4. Berechne eine Vektorraumbasis B von Z .

5. Setze $R := \bigcup_{i=1}^{|B|} (b_i^T \cdot P)$.

Die Menge R ist dann eine Basis von $\langle\langle V \rangle\rangle$. Dies gilt im übrigen nicht nur für $V = V_d$, sondern für beliebige Vektorräume $V \subseteq \mathbb{P}$.

Die Idee hinter dem Algorithmus sollte klar sein: Man identifiziert in Schritt 2 jedes auftretende multivariate Monom mit einem Einheitsvektor in einem Vektorraum. Mittels der Koeffizientenvektoren aller Polynome berechnet man eine Basis und transferiert in Schritt 5 das Ganze wieder zurück in multivariate Polynome.

Beispiel. Sei $V = \{xy - 2x, 2x - y, 3xy - 3y\}$. Wir haben drei unterschiedliche Monome, das heißt $P = [xy, x, y]$. Diesen Monomen weisen wir entsprechend die Einheitsvektoren $e_i \in K^3, i = 1, 2, 3$, zu. Wir erhalten dann folgende Vektormenge:

$$M = \{e_1 - 2e_2, 2e_2 - e_3, 3e_1 - 3e_3\} = \left\{ \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \\ -3 \end{pmatrix} \right\}.$$

Eine Vektorraumbasis hiervon ist

$$B = \left\{ \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix} \right\}.$$

Formen wir dies wieder durch die eindeutige Zuordnung der Einheitsvektoren zu P um, erhalten wir $R = \{xy - 2x, 2x - y\}$ als Basis von $\langle\langle V \rangle\rangle$.

Laufzeit. Wir greifen unsere Beispiel von oben wieder auf. Wie schon gesagt, wurden zur Berechnung von $V_{200}(F)$ ca. 3,5 Sekunden benötigt, wobei die Menge 3670 Polynome enthielt. Berechnen wir dies nun einmal mit `LinSpace` inklusive. Die Dauer steigt nun auf

135 Sekunden an, dafür hat die Menge $V_{200}(F)$ nur noch 201 Elemente und ist wirklich eine Vektorraumbasis. Die geringe Anzahl der Basiselemente wird sich später sehr positiv auswirken, selbst wenn die Berechnung der Basis länger dauert.¹⁷

Bemerkung. Man kann auch mittels des Orthonormalisierungsverfahrens von Schmidt eine Orthonormalbasis der Vektoren berechnen und erhält somit eine Orthogonalbasis für die Polynome. Dabei muß man nur berücksichtigen, daß wir hier meist ein anderes Skalarprodukt benutzen als das Standardskalarprodukt für Vektoren. Man sollte zusätzlich beachten, daß die Berechnung bei einer großen Anzahl von Vektoren erheblich mehr Zeit benötigt, als die einfache Basisberechnung.

Man kann auch versuchen, direkt einen Basisalgorithmus für Polynome zu implementieren. Damit würde man sich das mehrfache Konvertieren in Vektoren und zurück ersparen.

Eine Alternative zu obigem Ansatz wird von Peña und Sauer [10] beschrieben, die zur Berechnung des Raumes V_d auf den "darunter" liegenden Raum V_{d-1} zurückgreifen. Durch diese rekursive Definition erhält man eine erhebliche Zeitersparnis und der gesamte Algorithmus wird effizienter.

3.3 Skalarprodukt von Polynomen

Wir erinnern uns noch einmal kurz an die Definition der beiden Skalarprodukte aus Kapitel 3.2 von Teil II:

- Monomiales Skalarprodukt:

$$(f, g) = \sum_{\alpha \in \mathbb{N}_0^n} f_\alpha \cdot g_\alpha$$

- Differential-Skalarprodukt:

$$(f, g) = \sum_{\alpha \in \mathbb{N}_0^n} f_\alpha \cdot g_\alpha \cdot \alpha!$$

Bemerkung. Welches Skalarprodukt in den Algorithmen benutzt wird, hängt von der globalen Variable `sp_type` ab, die man mittels `setSP("M")` beziehungsweise `setSP("D")` zuweisen kann. Die Zuweisung vor der Berechnung ist dabei optional. Ist `sp_type` unbe-

¹⁷Siehe Seite 61.

stimmt, wird automatisch das Differential-Skalarprodukt benutzt.

Für die Berechnung des Skalarproduktes muß man also den Koeffizienten zu einem Multiindex und mitunter die Fakultät eines Multiindex bestimmen. Die Fakultät läßt sich sehr leicht mittels $\prod_{i=1}^n a_i!$ berechnen, wenn der Multiindex $a := [a_1, \dots, a_n]$ gegeben ist.

Daher gehen wir gleich zur Berechnung des Koeffizienten über. Seien dazu $f \in \mathbb{P}$ und ein Multiindex a gegeben.

Aufruf: `coeffmi(f, a);`

Algorithmus 3.4

1. Setze $c := 0$ und $p := x^a = \prod_{i=1}^n x_i^{a_i}$.
2. Schleife i von 1 bis Anzahl der Monome in f :
 - Setze $t := f[i]$.¹⁸
 - Ist $LM(t) = p$, dann setze $c := LK(t)$ und Ende.

Beispiel. Sei $U := [x, y]$, $f = 2xy - 3x$ und $a = [1, 0]$.

- $c := 0, p := x^1 \cdot y^0 = x$
- $i = 1/2$:
 - $t := 2xy$
 - $LM(t) = xy \neq x = p$
- $i = 2/2$:
 - $t := -3x$
 - $LM(t) = x = p \Rightarrow c := -3$

Weiter benötigen wir noch die Umwandlung eines Monoms in einen Multiindex.

¹⁸Wobei mit $f[i]$ das i -te Monom von f gemeint ist.

Aufruf: pmi (m) ;

Algorithmus 3.5

1. Setze Multiindex $a := [0, \dots, 0] \in \mathbb{N}_0^m$, wobei $m = |U|$.
2. Ist $\delta(m) \leq 0$ dann Ende.
3. Schleife i von 1 bis Anzahl der Unbekannten y_i in m :
 - Setze $t := m[i] = y_i^{e_i}$ mit e_i als Exponent.¹⁹
 - Bestimme pos als Position der Unbekannten y_i in U .
 - Setze $a[pos] := e_i$.

Beispiel. Sei $U := [x, y, z]$ und $m := x^2z$.

- $a := [0, 0, 0]$
- $i = 1/2$:
 - $t := x^2$
 - $U[pos] \stackrel{!}{=} x \Rightarrow pos = 1$
 - $a[1] := 2$
- $i = 2/2$:
 - $t := z$
 - $U[pos] \stackrel{!}{=} z \Rightarrow pos = 3$
 - $a[3] := 1$

Ergebnis: $a = [2, 0, 1]$.

Jetzt kommen wir schließlich zu der Berechnung des Skalarproduktes zwischen zwei Polynomen f und g ²⁰.

¹⁹Wobei mit $m[i]$ die i -te Unbekannte von m gemeint ist.

²⁰Hier nur am Beispiel des monomialen Skalarproduktes.

Aufruf: `scalarproduct(f, g)`;

Algorithmus 3.6

1. Setze $s := 0$.
2. Ist $\delta(f) \leq 0$ oder $\delta(g) \leq 0$ dann:
 - Ist $\delta(f) = 0$ and $\delta(g) = 0$, setze $s := fg$.
 - Ende.
3. Ist f ist ein Monom, dann setze a als Multiindex von f .
4. Ist g ist ein Monom, dann setze a als Multiindex von g .
5. Wenn Punkte 3 und 4 verneint wurden, dann sind f und g also keine Monome. Überspringe daher den nächsten Punkt.
6. Setze $s := s + \text{coeffmi}(f, a) \cdot \text{coeffmi}(g, a)$ und Ende.
7. Bestimme das "kürzere" Polynom (Anzahl der Monome) und tausche es gegebenenfalls an erste Stelle.
8. Schleife i von 1 bis Anzahl der Monome von f :
 - Setze $t := f[i]$.
 - Bestimme Multiindex a von t , das heißt $a := \text{pmi}(t)$.
 - Setze $s := s + \text{coeffmi}(f, a) \cdot \text{coeffmi}(g, a)$.

3.4 Der Raum $W_d(I)$

Sei $V_d(I)$ wie oben. Dann ist

$$W_d(I) := \{p \in P_d \mid (V_d(I), p) = 0\}$$

der zu $V_d(I)$ orthogonale Raum. Wir betrachten hier wieder nur eine Basis dieses Raumes und schreiben W_d anstelle $W_d(I)$.²¹

²¹Zur Erinnerung: Dieser Raum wird benötigt, um die Basis des Normalformraumes zu bestimmen.

Aufruf: Wspace(d,F);

Algorithmus 3.7

1. Setze $P := P_d = \{p_1, \dots, p_u\}$ und $V := V_d = \{v_1, \dots, v_r\}$.
2. Setze Polynom $t := \sum_{i=1}^u c_i \cdot p_i$, mit $c_i \in \mathbb{K}$ als unbekannte Koeffizienten.
3. Setze $T := \{(t, v_i) \mid i = 1, \dots, r\} = \{t_1, \dots, t_r\}$ als die Menge aller Skalarprodukte
4. Generiere aus T eine $r \times u$ -Matrix M .
5. Bestimme den homogenen Lösungsraum $S = \{s_1, \dots, s_k\}$ von $M \cdot x = 0$.
6. $W_d := \{P \cdot s_i \mid i = 1, \dots, k\}$.

Beispiel. Sei $F = [x^2 - xy + y^3, x - y]$, $U = [x, y]$, $d = 2$ und wir benutzen das Differential-Skalarprodukt.

- $W_d := \emptyset$, $P := P_d = [x^3, x^2y, xy^2, y^3]$ und $V := V_d = [xy^2, y^3]$.
- $t := \sum_{i=1}^3 c_i \cdot p_i = c_1x^3 + c_2x^2y + c_3xy^2 + c_4y^3$
- $T := \{(t, v_i) \mid i = 1, \dots, 4\} = \{6c_4, 2c_3\}$

- 2×4 -Matrix

$$M = \begin{pmatrix} 0 & 0 & 0 & 6 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

- Lösungsraum

$$S = \left\{ \left(\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) \right\}$$

von M

- Setze Vektor $p = [x^3, x^2y, xy^2, y^3]$.
- $W_d := \{p \cdot s_1, p \cdot s_2\} = \{x^2y, x^3\}$.

Laufzeit. Wichtig hierbei ist natürlich die Größe von V_d , denn danach richtet sich die Anzahl der Skalarprodukte, die berechnet werden müssen. Bei der Laufzeitanalyse der Berechnung von V_d erwähnten wir bereits, daß eine Verkleinerung sich positiv auswirkt, obwohl die Basisberechnung an und für sich länger dauert. Um dies zu bestätigen betrachten wir eine Idealbasis $F \in \mathbb{Z}[x, y, z]$ aus 10 Polynomen mit Maximalgrad 5, die jeweils aus 5 Monomen bestehen.

In den folgenden drei Tabellen, die jeweils für eine andere Berechnungsart von $V := V_{15}(F)$ stehen, sieht man die Laufzeit einzelner Funktionen²², wenn wir $W := W_{15}(F)$ berechnen²³.

Berechnung	Zeit in sec	
V	1,75	
Skalarprodukte	58,7	
LGS lösen	37,35	
Gesamtzeit	97,97	ohne Optimierungen von V mit $ V = 672$

Berechnung	Zeit in sec	
V	1,58	
Skalarprodukte	45,49	
LGS lösen	37,38	
Gesamtzeit	84,6	Erzeugendensystem von V mit $ V = 403$

Berechnung	Zeit in sec	
V	16,5	
Skalarprodukte	16,14	
LGS lösen	7,94	
Gesamtzeit	40,66	Basis von V mit $ V = 133$

Wie man sieht, ist es sinnvoll eine Basis von V_d zu berechnen, da die Skalarproduktberechnung und das Lösen des linearen Gleichungssystems den Großteil der Zeit bei der Berechnung von W_d ausmachen.

²²Bei der gesamten Berechnung fallen noch einige andere Funktionen ins Gewicht, so daß die Summen in den Tabellen nicht exakt mit den Gesamtwerten übereinstimmt.

²³Die Berechnung von W besteht hauptsächlich aus der Berechnung der Skalarprodukte und dem Lösen des linearen Gleichungssystems.

4 Berechnung der H-Basis

Für die Berechnung der H-Basis gibt es leider noch keinen Algorithmus, der auf die Berechnung von Gröbner-Basen verzichten kann, ohne zu komplex zu werden, das heißt sich einfach implementieren ließe. Wie schon bei Gröbner-Basen berechnet man eine Basis des Syzygienmoduls, erhält so die Syzygienpolynome und reduziert diese modulo des Ideals. Erhält man ein Polynom ungleich Null, fügt man dieses zum Ideal hinzu und fängt von vorne an.

Ein wesentlicher Bestandteil - und auch Komplexitätsfaktor - ist die Reduktion der Polynome. Hierfür muß man Polynome aus $V(I) := \bigcup_{d \in \mathbb{N}_0} V_d(I)$, die als Koeffizienten dienen, und ein Polynom aus $W(I) := \bigcup_{d \in \mathbb{N}_0} W_d(I)$ für den Rest finden.

4.1 H-Basis

Wir beschäftigen uns zuerst mit der H-Basis-Berechnung. Hierfür übernehmen wir den Algorithmus aus Kapitel 3.4 von Seite 32.

Aufruf: `HBasis(F)`;

Algorithmus 4.1

1. Ist `isHBasis(F) = true`, dann Ende.²⁴
2. Setze $H := F$ und $G := \emptyset$.
3. Solange $G \neq H$:
 - Setze $G := H$ und berechne eine Basis $S = \{s_1, \dots, s_r\}$ des Syzygienmoduls von $\Lambda(H)$.
 - Schleife i von 1 bis r :
 - Setze $h := s_i^T \cdot G$.
 - Berechne den Rest r von h modulo G .
 - Ist $r \neq 0$ dann füge r zu H hinzu.

Bemerkung. Mittels `isHBasis(F, true)` beziehungsweise `isHBasis(F, false)` kann überprüft werden, ob F eine H-Basis ist. Der boolsche Operator gibt dabei an, ob Punkt 3 der Kriterien von Seite 32 mitüberprüft werden soll (`true`) oder nicht (`false`).

²⁴Siehe hierfür die Kriterien in Kapitel II.3.4 auf Seite 32.

4.2 Syzygienmodul

Wie schon oben erwähnt, muß bei der Berechnung einer Basis des Syzygienmoduls leider noch auf Gröbner-Basen zurückgegriffen werden, so daß H-Basen kein völliger Ersatz sind. Der Algorithmus wurde von Buchberger [3, S. 219] beschrieben. Man greift dabei auf die Matrizen zurück, die wir in Kapitel 2.2 auf Seite 44 bestimmt haben. Dies ist auch der Grund, warum wir dort nicht den "normalen" Buchberger-Algorithmus, wie er in Teil II vorgestellt wurde, übernommen haben.

Aufruf: aSyzMod(F) ;

Algorithmus 4.2

1. Berechne Gröbner-Basis $G = \text{GBasis}(F)$.
2. Berechne Matrix $M = \text{IGBMatrix}(F)$ mit $G = F \cdot M$.
3. Berechne Matrix $N = \text{GBMatrix}(F)$ mit $F = G \cdot N$.
4. Berechne Matrix $R = \text{gSMatrix}(F)$ mit $R \cdot F = 0$.
5. Berechne Blockmatrix Q mit

$$Q = \begin{pmatrix} I - N^T \cdot M^T \\ R \cdot M^T \end{pmatrix},$$

wobei I die passende Einheitsmatrix ist.

6. Transponiere Q und extrahiere die Spaltenvektoren.

Als Ergebnis erhält man eine Menge $S = [s_1, \dots, s_r]$ von Vektoren, so daß gilt:

$$s^T \cdot F = 0 \quad \text{für alle } s \in \langle\langle S \rangle\rangle.$$

4.3 Divisionsalgorithmus

Wenn wir $F = [f_1, \dots, f_s]$ und $f \in \mathbb{P}$ gegeben haben, dann sind $h_i \in V(I)$ und $r \in W(I)$ mit

$$f = \sum_{i=1}^s h_i \cdot f_i + r$$

gesucht, wobei $V(I) = \bigcup_{d \in \mathbb{N}_0} V_d(I)$ und $W(I) = \bigcup_{d \in \mathbb{N}_0} W_d(I)$.

Aufruf: $\text{hDivAlg}(F, f)$;

Algorithmus 4.3

1. Setze alle $h_i := 0, r := 0, j := 0$ und $t := f$.
2. Solange $t \neq 0$:
 - Setze $j := j + 1$.
 - Berechne $h_i^{(j)}$ und $r^{(j)}$ als Koeffizienten und Rest bei der Reduktion des homogenen Leittermes von t modulo $\Lambda(F)$.
 - Setze $h_i := h_i + h_i^{(j)}$ und $r := r + r^{(j)}$.
 - Setze $t := t - \sum_{i=1}^s h_i^{(j)} \cdot f_i$.

Der Divisionsalgorithmus läßt sich somit auf einen Divisionsalgorithmus für homogene Leitterme zurückführen.

Aufruf: $\text{hcoeffPoly}(F, f)$;

Algorithmus 4.4

1. Bestimme $d := \delta(f)$, den Totalgrad von f .
2. Schleife i von 1 bis s :
 - Berechne $P := P_{d-\delta(f_i)} = \{p_1, \dots, p_r\}$.
 - Setze $h_i := \sum_{j=1}^r c_{i,j} \cdot p_j$ mit Unbekannten $c_{i,j}$.
3. Berechne $V := V_d(\cdot)$.
4. Setze $r := \Lambda(f) - \sum_{i=1}^s h_i \cdot \Lambda(f_i)$ als Rest.
5. Setze $L := \{(v, r) = 0 \mid v \in V\}$.
6. Berechne Koeffizientenmatrix M und -vektor b aus L von $c_{i,j}$.
7. Bestimme den Lösungsraum von $M \cdot x = b$, wähle einen Repräsentanten aus und setze die Lösung in die $h_i, i = 1, \dots, s$, ein.

Bemerkung. Die Wahl des Repräsentanten ist derzeit willkürlich. Durch eine geschickte Wahl (viele Einträge gleich Null) kann man die Berechnungszeit wahrscheinlich verkürzen.

4.4 Laufzeit

Die Laufzeit der gesamten H-Basis-Berechnung hängt essentiell von der Berechnung des Restes und der Skalarprodukte ab, wie oben schon erwähnt wurde. Einfache Optimierungen können aber auch hier schon viel bewirken.

Wenn man sich obigen Algorithmus zur Reduktion der Polynome genau anschaut, fällt auf, daß man sehr oft Redundanzen berechnet. So berechnet man zum Beispiel V_d immer aufs Neue, selbst wenn der Grad d sich wiederholt. Deshalb speichern wir nun V_d in einer globalen Variable ab. Auch die Polynome h_i hängen nur von F und d ab, daher speichern wir diese ebenso.

Außerdem sollte das Augenmerk auf die Berechnung des Skalarproduktes gerichtet werden. Es ist:

$$L := \{(v, \Lambda(f) - \sum_{i=1}^s h_i \cdot \Lambda(f_i)) = 0 \mid v \in V\}.$$

Wie man sieht, ist der Ausdruck $-\sum_{i=1}^s h_i \cdot \Lambda(f_i)$ immer wieder der Gleiche und nur abhängig von d , aber nicht von f selbst. Da das Skalarprodukt bilinear ist, gilt:

$$(v, \Lambda(f) - \sum_{i=1}^s h_i \cdot \Lambda(f_i)) = (v, \Lambda(f)) + (v, -\sum_{i=1}^s h_i \cdot \Lambda(f_i)).$$

Der zweite Summand ist unabhängig von f , diesen können wir dadurch global berechnen und speichern. Wir müssen damit nur noch in jedem Schritt $(v, \Lambda(f))$ für alle $v \in V$ berechnen. Dies lohnt vor allem, da $\Lambda(f)$ in der Regel nur ein Monom oder zumindest ein relativ kleines Polynom ist.

Schauen wir uns nun die Laufzeit des obigen Algorithmus an einem Beispiel an. Wir benutzen dazu einen Teil der Ausgangsgleichungen der 2D-Kinematik 2 aus Kapitel 4.2 von Seite 80. Sei also $U := [x_1, y_1, x_2, y_2]$ und $F = [x_1^2 - 2ax_1 + a^2 + y_1^2 - l_1^2, x_2^2 + 2ax_2 + a^2 + y_2^2 - l_2^2, x_1^2 + y_1^2 - x_2^2 - y_2^2, x_1^2 - 2x_1x_2 + x_2^2 + y_1^2 - 2y_1y_2 + y_2^2 - 4b^2]$ mit Konstanten a, b, l_1, l_2 .

Funktion	Anzahl der Aufrufe	Gesamtzeit in sec
aSyzMod	3	6,92
hDivAlg	28	54,21
scalarproduct	1649	103,87
Pspace	77	0,00
Vspace	12	0,87
LinSpace	12	4,40
GenerateMatrix	87	0,56
LinearSolve	87	45,51

Gesamtzeit: 3:47 min

Speicherverbrauch: 15,8 MB

Wenn man das Ganze mit der ersten Implementierung ohne Optimierungen vergleicht, bei der die Berechnung 23:44 Minuten dauerte und 41,2 MB verbrauchte, ist dies eine Laufzeiterparnis von ca. 84 Prozent und eine Speicherersparnis von ca. 62 Prozent. Im Allgemeinen hängt dies natürlich immer von der Komplexität des Ideals ab.

Bemerkung. Durch die Bilinearität des Skalarproduktes erhalten wir zusätzlich folgende Hilfe: Wenn wir die Matrix gebildet haben, suchen wir ein x mit $Mx = b$, wobei b in unserem Fall durch $(v, \Lambda(f))$ bestimmt wird. Da M für alle Polynome vom Grad d immer gleich bleibt, könnten wir hier eine LU-Zerlegung der Matrix M in eine rechte obere Dreiecksmatrix U und eine linke untere Dreiecksmatrix L , die auf der Hauptdiagonalen nur Einträge 1 hat, vornehmen und müssen zuerst nur $Lc = b$ und danach $Ux = c$ lösen. Dies kann einige Zeit sparen und die Dauer der `LinearSolve`-Funktion verkleinern.

5 Lösungsverfahren

Im folgenden Kapitel kann man alle Prozeduren bezüglich Gröbner-Basen oder H-Basen berechnen. Welche Basis benutzt wird, hängt von der globalen Variable `basis_type` ab, die man mittels `setBasisType("G")` beziehungsweise `setBasisType("H")` zuweisen kann. Die Zuweisung vor der Berechnung ist dabei optional. Ist `basis_type` unbestimmt, wird automatisch die H-Basis-Berechnung benutzt.

5.1 Normalformraum

Zur Erinnerung: Der Normalformraum ist wie folgt definiert:

- Gröbner-Basen:

$$NF(G) := \langle\langle p \in \mathbb{P}^* \mid p \notin \langle LT(G) \rangle \rangle\rangle,$$

wobei $\mathbb{P}^* := \bigcup_{d \in \mathbb{N}_0} P_d$ die Menge aller Monome ist.

- H-Basen:

$$NF(G) := W(G) = \bigcup_{d \in \mathbb{N}_0} W_d(G).$$

Die Anzahl der Basiselemente verrät uns zum einen, wie viele (unter Umständen mehrfache) Lösungen man maximal erhalten kann. Zusätzlich kann man mittels der Basiselemente die Varietät des Ideals I bestimmen.

Aufruf: : `NForm(F)`;

Algorithmus 5.1

1. Ist `isZeroDim(F) = false`, dann Fehler und Ende.
2. Setze $i := 0$ und $B_i := \{1\}$.
3. Ist `basis_type="G"`:
 - Setze $H := LT(\text{GBasis}(F))$.
 - Solange $B_i \neq \emptyset$:
 - Schleife j von 1 bis $|P_i|$:
Ist `grmd(H,P_ij) ≠ 0`, dann setze $B_i = B_i \cup \{P_{ij}\}$.

– Setze $i := i + 1$.

4. Ist `basis_type="H"`:

- Setze $H := \text{HBasis}(F)$.
- Solange $B_i \neq \emptyset$:
 - Bestimme $B_i = W_i(F)$.
 - Setze $i := i + 1$.

5. Setze $NF := \bigcup_{j=1}^{i-1} B_j$.

Da wir mittels `isZeroDim(F)` überprüfen, ob das von F erzeugte Ideal nulldimensional ist, erhalten wir so auch die Gewissheit, daß der Algorithmus terminiert.

5.2 Multiplikationstabellen

Die genaue Definition der Multiplikationstabellen kann in Teil II in Kapitel 4.3 auf Seite 36 nachgelesen werden. Wir wenden uns sofort dem Algorithmus zur Berechnung zu.

Aufruf: `MultiTable(F,h)`;

Algorithmus 5.2

1. Berechne eine Basis $P = [p_1, \dots, p_s]$ des Normalformraums von F .
2. Setze $p := \sum_{j=1}^s c_j \cdot p_j$ mit Unbekannten c_j .
3. Schleife i von 1 bis s :
 - Setze r als Rest von $p_i \cdot h$ bei Division durch F .²⁵
 - Berechne $r - p$ und löse mittels Koeffizientenvergleich die c_j auf.
 - Setze die i -te Spalte von M als $(c_1, \dots, c_r)^T$.

²⁵Es ist also $r = \phi_h(p_i)$.

Beispiel. Sei hierzu $U := [x, y]$, $F := [xy^2 - 2x, y - 4x]$ und $h = xy$.

- Eine H-Basis von F ist $[y^3 - 2y, -4x + y]$ und eine Basis des Normalformraums hierzu ist $P = [1, x + 4y, x^2 + 8xy + 16y^2]$, das heißt, die Varietät zum Ideal $\langle F \rangle$ hat genau drei Elemente.
- $p := \sum_{j=1}^3 c_j \cdot p_j = c_3(x^2 + 8xy + 16y^2) + c_2(x + 4y) + c_1$
- $i = 1/3$:
 - $r := 1 \cdot xy \pmod{F} = \frac{4}{289}(x^2 + 8xy + 16y^2)$
 - Es ergibt sich, daß $c_1 = c_2 = 0$ und $c_3 = \frac{4}{289}$.
- $i = 2/3$:
 - $r := (x + 4y)xy \pmod{F} = \frac{x}{2} + 2y$
 - Es ergibt sich, daß $c_1 = c_3 = 0$ und $c_2 = \frac{1}{2}$.
- $i = 3/3$:
 - $r := (x^2 + 8xy + 16y^2)xy \pmod{F} = \frac{x^2}{2} + xy + 8y^2$
 - Es ergibt sich, daß $c_1 = c_2 = 0$ und $c_3 = \frac{1}{2}$.
- Daraus ergibt sich die Multiplikationstabelle

$$M := \begin{pmatrix} 0 & 0 & \frac{4}{289} \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

Um Rechenzeit zu sparen, definieren wir nun noch eine zweite Funktion, die uns die Berechnung der Multiplikationstabellen zu mehreren h s gibt.

Aufruf: `MultiTableSet(F,U)`;

Laufzeit. Folgendes Beispiel zeigt, daß dieser Zusammenschluß sinnvoll ist. Sei $U = [x, y]$ und $F = [x^4 + 2x^2y^2 + y^4 - 1, xy(x^2 - y^2)]$. Wir wollen jetzt für x und y die Multiplikationstabellen berechnen. Einmal erledigen wir dies mit einer normalen Schleife über die Menge U und berechnen immer `MultiTable(F, U[i])`. Der Vorgang dauert 3.66 Sekunden. Verwenden wir aber den eben erwähnten Algorithmus mittels `MultiTableSet(F, U)`, dauert

es nur noch 2.59 Sekunden. Dies ist nur ein kleiner Gewinn, aber je mehr Unbekannte man hat und je größer die Normalformraumbasis ist, desto größer ist die Zeitersparnis²⁶.

5.3 Berechnung der Varietät

Der Algorithmus zur Bestimmung der Varietät zu einem zugehörigen Ideal ist etwas größer und erhält daher ein gesamtes Kapitel. Zuerst umreißen wir noch einmal kurz den Algorithmus des Lösungsverfahrens:

1. Bestimme eine Basis P des Normalformraumes eines Ideals I .
2. Berechne die Multiplikationstabellen $M_i = M(x_i)$ für $i = 1, \dots, n$.
3. Bestimme die Eigenwerte der M_i und die zugehörigen Eigenvektoren und "verbinde diese passend".

Punkte 1 und 2 haben wir mit den Algorithmen oben erledigt. Nun kommt aber der komplizierteste Teil, die Berechnung der Eigenräume und das Verbinden der Eigenwerte zur Lösung.

Die Eigenvektor-Berechnung führen wir hier nicht näher aus. Es sei nur gesagt, daß die Funktion `Eigenspaces(F)` als Rückgabewert in einer mehrfach verschachtelten Liste die Eigenwerte und Eigenvektoren aller Multiplikationstabellen für die Unbekannten in `List_Undets` enthält.

Beispiel. Sei $U = [x, y]$ und $F = [y^2 - y, x - 1]$. Dann wäre das Resultat von `Eigenspaces(F)` folgende Liste:

$$\left[\left[\left[[1], \left[\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \right] \right] \right], \left[\left[[0], \left[\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \right] \right], \left[[1], \left[\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \right] \right] \right] \right].$$

Was auf den ersten Blick etwas unstrukturiert wirkt, ist recht logisch aufgebaut: Die äußere Klammer umfaßt die beiden Inhalte (Eigenwerte und Eigenvektoren) der Multiplikationstabellen zu den beiden Unbekannten x und y . In jeder dieser Listen befinden sich wieder mehrere Listen mit je einem Eigenwert und den zugehörigen Eigenvektoren.

²⁶Ein Beispiel mit 6 Unbekannten, dessen Normalformraum die Dimension 6 hat, benötigte so nur 7,5 anstatt 40 Sekunden.

Bemerkung. In dem Beispiel gibt es auch schon einen Sonderfall. Wenn wir uns die Multiplikationstabelle von x und die zugehörigen Eigenwerte und -vektoren anschauen, fällt auf, daß es nur einen Eigenwert gibt und der Eigenraum den ganzen R^2 aufspannt. Das bedeutet, für x gibt es nur diese eine Lösung, die dazu unabhängig von allen anderen Variablen ist. Es ist daher sinnvoll, solche Sonderfälle schon vor der folgenden Vereinigung der Menge zu extrahieren.

Man sieht nun auch schon, wie man die zugehörigen Eigenwerte und Eigenvektoren verknüpfen kann. Dies geschieht über die Vereinigung aller Eigenräume und das Überprüfen der Dimension. Bevor wir diese Überprüfung aber tätigen können, müssen wir die obige Liste geschickt verknüpfen.

Dazu benutzen wir die Funktion `CartUnion(M)`, wobei M die Form von obiger Liste haben muß²⁷. Der Algorithmus ist natürlich rekursiv definiert, wird hier aber nicht aufgeführt oder näher erläutert.

Der Befehl `CartUnion(Eigenspaces(F))` mit obigem F und U liefert:

$$\left[\left[[1, 0], \left[\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \right] \right], \left[[1, 1], \left[\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \right] \right] \right].$$

Es wurden einfach zu den zugehörigen Mengen von Eigenwerten und Eigenvektoren der beiden Multiplikationstabellen kreuzweise²⁸ die Eigenwerte verbunden und die Eigenräume vereinigt. Man erhält so eine Liste von mehreren potentiellen Lösungen, die man noch überprüfen muß.

Das Überprüfen einer möglichen Lösung (dies sind die n -Tupel in der Liste) geschieht mittels der zugehörigen Liste von Vektoren. Dabei gehören die Lösungen für die einzelnen Variablen genau dann "zueinander" - sprich liegen wirklich in der Varietät -, wenn die zugehörigen Vektoren linear abhängig sind. Um dies zu überprüfen, bildet man einfach eine Vektorraumbasis und schaut sich die Dimension an. Ist diese kleiner als die Anzahl der Vektoren am Anfang, hat man eine Lösung (siehe Teil II, Kapitel 7, Seite 40).

Damit haben wir auch den dritten Teil erledigt, der zu folgendem Algorithmus führt.

²⁷Dies ist genau die Form, welche die Funktion `Eigenspaces()` schließlich liefert.

²⁸Daher auch der Name `CartUnion` der Prozedur, der sich so am "Cartesian Product" orientiert.

Aufruf: : solveH(F);

Algorithmus 5.3

1. Setze $S := \emptyset$, $V := \emptyset$ und $W := \emptyset$.
2. Berechne $E := \text{Eigenspaces}(F)$.
3. Schleife i von 1 bis $|E|$:
 - Gibt es nur einen Eigenwert in $E[i]$, dann füge die Gleichung $x_i = \text{Eigenwert}(E[i])$ zu V hinzu und lösche $E[i]$.
 - Ansonsten füge x_i zu W hinzu.
4. Berechne $C := \text{CartUnion}(E)$.
5. Schleife j von 1 bis $|C|$:
 - Berechne eine Basis B aller Eigenvektoren von $C[j]$.
 - Ist $|B| < |C[j]|$, füge die Menge $W = \text{Lösungstupel}(C[j]) \cup V$ zu S hinzu.

Das Ergebnis ist die gesuchte Varietät, also eine Menge S aller Lösungen des Ideals F .

Beispiele: Wir werden den Algorithmus nicht im Einzelnen durchgehen, sondern einfach einige nulldimensionale Ideale betrachten und von diesen die H-Basis, Normalformraumbasis und Lösung angeben.

1. Sei $U = [x, y, z]$, $F = [x^2 - 1, y^2 - x, z - x + y]$. Die H-Basis ist dann $[2yz + z^2 + y + z - 1, y^2 - y - z, -x + y + z]$ und die Basis des Normalformraumes ist $[1, x + z, x + y, -xy + xz - yz + 4z^2]$, das heißt, es gibt die folgenden vier Lösungen: $[[x = -1, y = -i, z = -1 + i], [x = -1, y = i, z = -1 - i], [x = 1, y = -1, z = 2], [x = 1, y = 1, z = 0]]$.
2. Sei $U = [x, y]$, $F = [3x^2y + 2xy + y + 9x^2 + 5x - 3, 2x^3y - xy - y + 6x^3 - 2x^2 - 3x + 3, x^3y + x^2y + 3x^3 + 2x^2]$. Die H-Basis ist $[xy + x - y + 3, -2x^2 + 3x - 2y + 6, 2y^2 - 8x - 5y - 3]$ und die Normalformraumbasis $[1, y, x]$. Das führt uns zu der Lösungsmenge $[[x = 0, y = 3], [x = \frac{5+\sqrt{65}}{4}, y = \frac{-3-\sqrt{65}}{4}], [x = \frac{5-\sqrt{65}}{4}, y = \frac{-3+\sqrt{65}}{4}]]$.

3. Sei $U = [x, y]$, $F = [x^2 - y, x^3 - y^2, -xy + y^2, y^3 - y^2]$. Dann ist die H-Basis $[x^2 - y, -xy + y^2, y^3 - y^2]$ und die Normalformraumbasis $[1, y, x, xy + 2y^2]$. Die Lösungsmenge, per `solveH(F)` ermittelt, ist $[[x = 0, y = 0], [x = 1, y = 1]]$.

Bemerkung. Das dritte Beispiel ist ein Sonderfall, der schon in Kapitel II.4.2 auf Seite 36 beschrieben wurde. Es handelt sich hier um mehrfache Nullstellen, weswegen die Anzahl der Lösungen kleiner als die Basis des Normalformraums ist.

Teil IV

Anwendungen

1 Einführung

In diesem Teil IV betrachten wir nun einige Anwendungsbeispiele für Gröbner- und H-Basen anhand diverser Kinematiken. Bei all diesen Beispielen geht es darum, aus einer bestimmten Längenkonfiguration von Stangen die Position des Kopfes zu bestimmen (kinematische Transformation) beziehungsweise aus der Position des Kopfes die Längen der Stangen abzuleiten (inverse kinematische Transformation). Wir werden hierfür zuerst die Beispiele genau beschreiben und in ein polynomiales Gleichungssystem umschreiben. Diese Polynome werden dann dem Ideal hinzugefügt, von welchem wir die Gröbner- beziehungsweise H-Basis berechnen werden. Mittels der Methoden aus Teil II werden wir dann versuchen, die zum Ideal gehörige Varietät zu bestimmen.

2 2D-Kinematik 1 - Spezialfall

2.1 Konstruktion

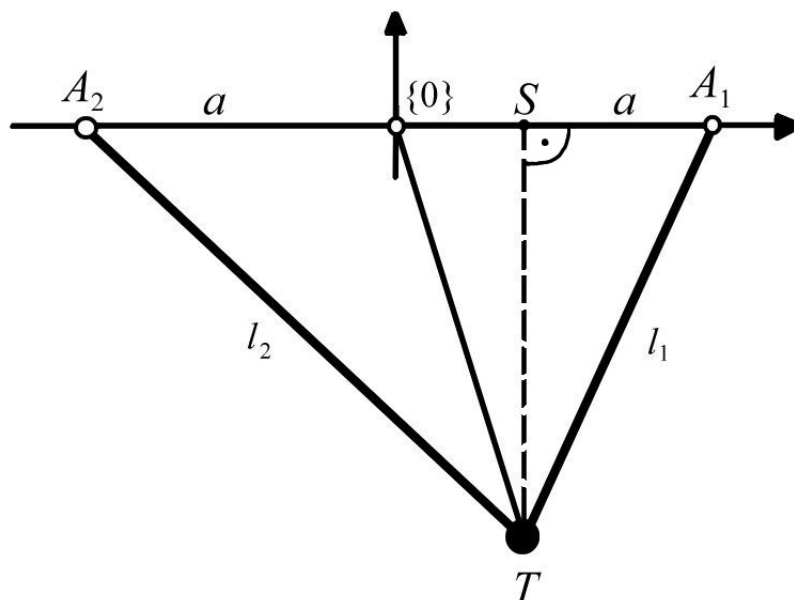


Abbildung 2.1: Spezielle 2D-Kinematik 1

Die Abbildung 2.1 beschreibt den Spezialfall der 2D-Kinematik 1, in dem die zwei Aufhängepunkte A_1 und A_2 auf der x -Achse liegen und zusätzlich die Abstände von beiden Punkten zum Ursprung $\{0\}$ gleich groß sind.

2.2 Idealbestimmung

Der Abstand vom Ursprung zu den beiden Punkten A_1 und A_2 beträgt jeweils a . Daher haben diese die Form

$$A_1 = (a, 0)^T,$$

$$A_2 = (-a, 0)^T.$$

Betrachten wir nun den Punkt $S = (x, 0)^T$, der die Projektion von T auf die x -Achse ist. Nach dem Satz von Pythagoras gilt nun

$$l_i^2 = y^2 + \|A_i - S\|_2^2, \quad i = 1, 2.$$

Dies führt zu folgendem Gleichungssystem:

$$\begin{aligned}y^2 + (a - x)^2 - l_1^2 &= 0, \\y^2 + (a + x)^2 - l_2^2 &= 0.\end{aligned}$$

Wir können nun also unser Ideal I beziehungsweise die erzeugende Idealbasis F angeben mit

$$F := [y^2 + (a - x)^2 - l_1^2, y^2 + (a + x)^2 - l_2^2].$$

2.3 Inverse kinematische Transformation

Für die Bestimmung der inversen kinematischen Transformation benötigen wir keine H-Basen, sondern können das Ergebnis direkt ablesen. Es ist

$$\begin{aligned}l_1 &= \sqrt{y^2 + (a - x)^2}, \\l_2 &= \sqrt{y^2 + (a + x)^2}.\end{aligned}$$

2.4 Kinematische Transformation

Zuerst müssen wir mittels `setUndets([x,y])` und `setConsts([a,l1,l2])` die Variablen und Konstanten deklarieren.²⁹ Berechnen wir nun die H-Basis von F , erhalten wir

$$H = [16a^2y^2 - 8l_1^2a^2 + 16a^4 - 8a^2l_2^2 + l_2^4 - 2l_2^2l_1^2 + l_1^4, -4ax - l_1^2 + l_2^2].$$

Da x und y bereits getrennt in den Gleichungen vorkommen, kann man diese sofort auflösen. Aber auch ein `solveH(F)` führt uns zu dem gewünschten Ergebnis, so daß mit

$$\begin{aligned}x &= \frac{l_2^2 - l_1^2}{4a}, \\y &= -\frac{\sqrt{2l_2^2l_1^2 + 8l_1^2a^2 - 16a^4 - l_2^4 + 8a^2l_2^2 - l_1^4}}{4a}\end{aligned}$$

das System gelöst wird.

²⁹Zur Deklaration siehe Teil III, Kapitel 1.

Bemerkung. Beachtenswert ist, daß die Lösung für x relativ leicht zu berechnen ist, wegen der Wert für y um einiges komplizierter aussieht. Wir können annehmen, daß dies bei der speziellen 3D-Kinematik in Kapitel 5 ebenfalls so sein wird.

Werfen wir aber noch einen Blick auf die Normalform. Die Basis hiervon ist $P = [1, y]$. Da wir zwei Elemente in der Basis haben, sollten wir eigentlich auch zwei Lösungen mittels `solveH` erhalten, außer die Lösung oben wäre eine mehrfache Nullstelle. Natürlich handelt es sich hier nicht um eine mehrfache Nullstelle, die zweite Lösung wurde nur unterschlagen. Man bekommt nämlich zusätzlich zu obigen Werten noch

$$\begin{aligned} x &= \frac{l_2^2 - l_1^2}{4a}, \\ y &= \frac{\sqrt{2l_2^2l_1^2 + 8l_1^2a^2 - 16a^4 - l_2^4 + 8a^2l_2^2 - l_1^4}}{4a} \end{aligned}$$

als Lösung. Dies ist die Spiegelung der Konstruktion an der x -Achse, was laut unseren Gleichungen nicht verboten ist. Leider wurde keine polynomiale Gleichung gefunden, welche es schafft, diesen positiven y -Wert zu eliminieren, was uns später Probleme bereiten wird.

Bemerkung. Man beachte auch, daß für manche l_1 - und l_2 -Werte der Wert für y imaginär wird; nämlich immer dann, wenn $l_1 + l_2 < 2a$ ist, da die Stangen dann keinen Berührungspunkt mehr haben. Mathematisch gesehen gibt es aber immer noch (komplexe) Lösungen hierfür. Auch dieser Aspekt, das heißt die Elimination komplexer Lösungen, konnte nicht durch das Hinzufügen polynomialer Gleichungen ausgeglichen werden, wodurch wir später ebenfalls Probleme bekommen werden.

3 2D-Kinematik 1 - Allgemein

3.1 Konstruktion

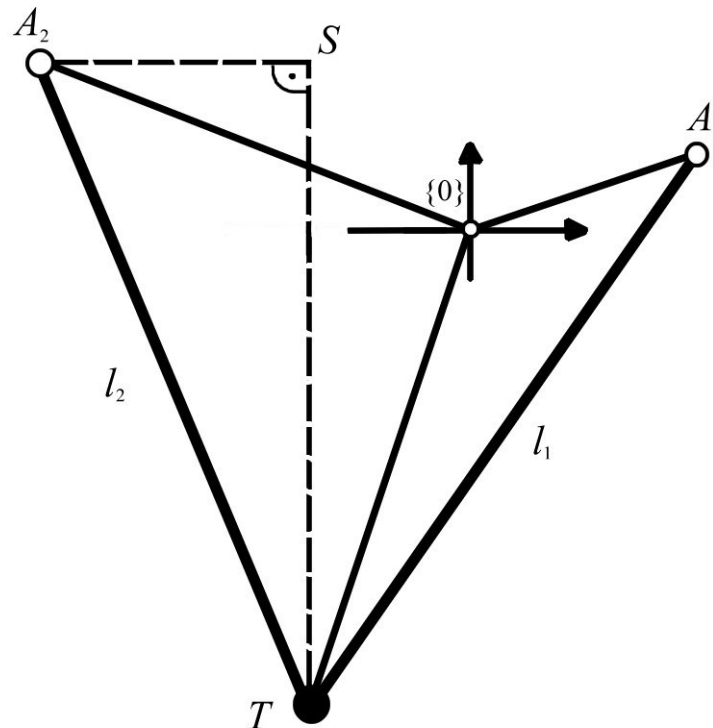


Abbildung 3.1: Allgemeine 2D-Kinematik 1

In der speziellen 2D-Kinematik 1 liegen die beiden Aufhängpunkte A_1 und A_2 auf der x -Achse und zusätzlich ist der Abstand der beiden Punkte zum Ursprung identisch. In dieser Kinematik wollen wir keine Bedingungen mehr an die Aufhängpunkte $A_1 = (x_1, y_1)$ und $A_2 = (x_2, y_2)$ stellen.

3.2 Idealbestimmung

Man betrachte den Schnittpunkt S in Abbildung 3.1. Mit dem Satz des Pythagoras erhalten wir

$$(y + y_2)^2 + (x_2 - x)^2 - l_2^2 = 0.$$

Ähnlich fährt man mit Punkt A_1 fort und erhält die zweite Gleichung

$$(y + y_1)^2 + (x_1 - x)^2 - l_1^2 = 0.$$

Unsere Idealbasis besteht also wieder nur aus zwei Gleichungen. Es ist

$$F := [(x_1 - x)^2 + (y + y_1)^2 - l_1^2, (x_2 - x)^2 + (y + y_2)^2 - l_2^2].$$

3.3 Inverse kinematische Transformation

Wir können wieder sofort ohne die Benutzung von H-Basen die Lösung für l_1 und l_2 ablesen. Es ist:

$$\begin{aligned} l_1 &= \sqrt{(y + y_1)^2 + (x_1 - x)^2}, \\ l_2 &= \sqrt{(y + y_2)^2 + (x_2 - x)^2}. \end{aligned}$$

3.4 Kinematische Transformation

Zuerst legen wir wieder mit `setUndets([x,y])` die beiden Unbekannten fest. Dafür gibt es dieses Mal mehr Konstanten, die wir mit `setConsts([x1,y1,x2,y2,l1,l2])` hinzufügen.

Die H-Basis von F läßt sich berechnen als

$$\begin{aligned} H := & [(-8y_1y_2 - 8x_2x_1 + 4y_2^2 + 4x_1^2 + 4y_1^2 + 4x_2^2)y^2 + (-8y_1x_1x_2 + 4y_2l_1^2 - 4y_1y_2^2 + 4y_1x_2^2 + \\ & 4y_2x_2^2 - 4y_2l_2^2 - 4y_1l_1^2 + 4y_2x_1^2 - 4y_2y_1^2 + 4y_1l_2^2 + 4y_1^3 + 4y_1x_1^2 + 4y_2^3 - 8x_1x_2y_2)y + x_1^4 + x_2^4 + \\ & y_1^4 + l_2^4 + 4l_2^2x_2x_1 + 4l_1^2x_2x_1 - 4y_1^2x_2x_1 - 4y_2^2x_2x_1 + 2y_2^2x_2^2 - 4x_1^3x_2 + 2l_1^2y_2^2 - 2l_2^2x_2^2 + y_2^4 + l_1^4 - \\ & 2l_2^2x_1^2 - 2y_2^2l_2^2 - 2l_1^2l_2^2 + 6x_2^2x_1^2 + 2x_2^2y_1^2 - 2y_2^2y_1^2 + 2y_2^2x_1^2 - 2l_1^2x_1^2 + 2y_1^2x_1^2 - 4x_1x_2^3 - 2l_1^2x_2^2 - \\ & 2y_1^2l_1^2 + 2y_1^2l_2^2, (2x_1 - 2x_2)x + (-2y_1 + 2y_2)y - l_2^2 + l_1^2 - y_1^2 - x_1^2 + x_2^2 + y_2^2]. \end{aligned}$$

Die Basis des Normalformraumes hat die Form

$$P = [1, (y_1 - y_2)x + (x_1 - x_2)y],$$

das heißt, es gibt wieder nur zwei Lösungen für unser Problem³⁰. Durch die vielen Konstanten sind die Lösungen für x und y aber ziemlich lang, diese werden daher nicht hier angegeben, sondern können in Anhang 3 auf Seite 119 nachgelesen werden.

³⁰Was man aber auch der H-Basis wieder direkt ansieht, da y in der ersten quadratischen Gleichung allein vorkommt.

4 2D-Kinematik 2

4.1 Konstruktion

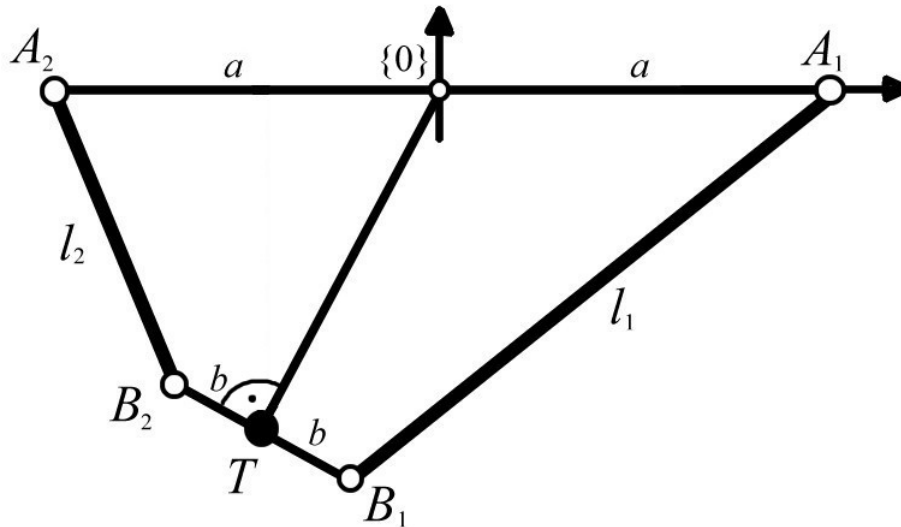


Abbildung 4.1: Spezielle 2D-Kinematik 2

In Abbildung 4.1 sehen wir eine ähnliche Kinematik wie in Kapitel 2, nur laufen die beiden Stangen nicht in der Mitte zusammen, sondern treffen in den Punkten B_1 und B_2 auf ein Tableau, in dessen Mitte der gesuchte Punkt T befestigt ist. Außerdem gibt eine mittlere Stange, die den Ursprung $\{0\}$ mit T verbindet und dazu noch senkrecht auf dem unteren Tableau steht, der Konstruktion zusätzlichen Halt.

4.2 Idealbestimmung

Zuerst untersuchen wir wieder den Einfluß der Längen l_1 und l_2 . Es ist leider nicht möglich, x und y direkt zu berechnen. Daher werden wir einen Umweg über die beiden Endpunkte des Tableaus gehen. Dafür betrachten wir die (nicht eingezeichneten) Projektionen von $B_1 = (x_1, y_1)$ und $B_2 = (x_2, y_2)$ auf die x -Achse. Wir erhalten wie schon beim Spezialfall der 2D-Kinematik 1 über den Satz des Pythagoras die Gleichungen

$$\begin{aligned} y_1^2 + (a - x_1)^2 - l_1^2 &= 0 \\ y_2^2 + (a + x_2)^2 - l_2^2 &= 0 \end{aligned}$$

Weiterhin wissen wir, daß durch den gleichen Abstand b der beiden Endpunkte zu T , das Dreieck B_1OB_2 gleichschenkelig ist (Abbildung 4.2). Dadurch erhalten wir

$$x_1^2 + y_1^2 - x_2^2 - y_2^2 = 0.$$

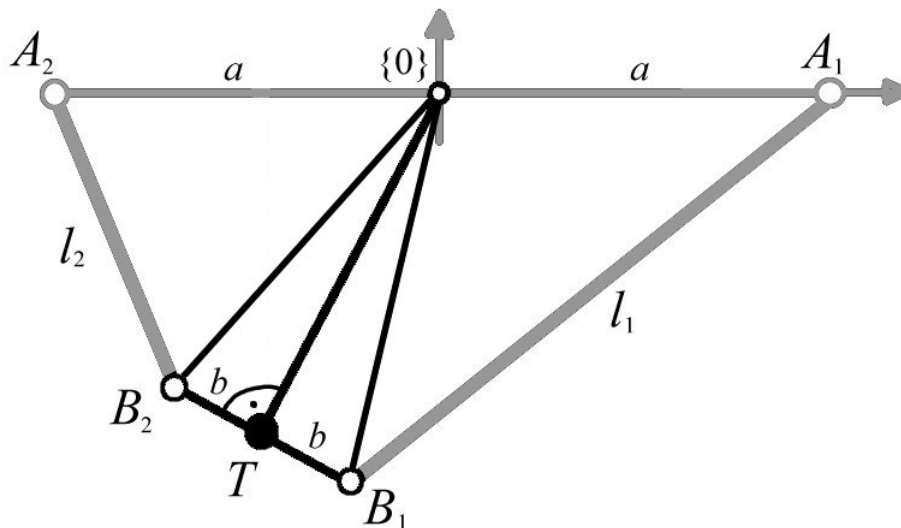


Abbildung 4.2: Das gleichschenkelige Dreieck in der Konstruktion.

Außerdem kann man hier auch leicht die Orthogonalität der mittleren Stange zum Tableau ablesen. Es ist nämlich

$$\begin{aligned} x^2 + y^2 + b^2 - x_1^2 - y_1^2 &= 0, \\ x^2 + y^2 + b^2 - x_2^2 - y_2^2 &= 0. \end{aligned}$$

Auch wenn wir den Umstand schon betrachtet haben, daß der Vektor $B_2 - B_1$ senkrecht auf dem Vektor T steht, können wir dies noch mittels des Standard-Skalarproduktes

$$(B_2 - B_1, T) = 0 \Leftrightarrow \left(\begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right) = 0$$

ausdrücken, was zu der (redundanten) Gleichung

$$(x_2 - x_1)x + (y_2 - y_1)y = 0$$

führt.

Zum Schluß wollen wir endlich die für uns interessanten Variablen x und y mit in die Gleichungen aufnehmen. Wenn man die Punkte B_1 und B_2 berechnet hat, kann man T leicht als

$$T = \frac{B_1 + B_2}{2}$$

angeben. Dies führt dann zu folgenden polynomialen Gleichungen

$$\begin{aligned} x_1 + x_2 - 2x &= 0, \\ y_1 + y_2 - 2y &= 0. \end{aligned}$$

Wir haben nun alle Gleichungen zusammen, um die Idealbasis für diese Kinematik aufzustellen. Es ist also nun

$$F := [y_1^2 + (a - x_1)^2 - l_1^2, y_2^2 + (a + x_2)^2 - l_2^2, x_1^2 + y_1^2 - x_2^2 - y_2^2, x^2 + y^2 + b^2 - x_1^2 - y_1^2, x^2 + y^2 + b^2 - x_2^2 - y_2^2, (x_2 - x_1)x + (y_2 - y_1)y, x_1 + x_2 - 2x, y_1 + y_2 - 2y].$$

4.3 Inverse kinematische Transformation

Wir setzen mit `setUndets([x1,y1,x2,y2,l1,l2])` und `setConsts([x,y,a,b])` die Unbekannten und Konstanten fest.

Die H-Basis von F ist dann $[(x^2 + y^2)y_2^2 + (-2y^3 - 2yx^2)y_2 + y^2x^2 - b^2x^2 + y^4, -xl_1^2 - 2aay_2 + x^3 + a^2x + y^2x + b^2x + 2ay^2 - 2x^2a, -xl_2^2 - 2aay_2 + x^3 + a^2x + y^2x + b^2x + 2x^2a + 2ay^2, -xx_1 + yy_2 - y^2 + x^2, xx_2 + yy_2 - x^2 - y^2, y_1 + y_2 - 2y]$.

Die Normalformraumbasis hat acht Elemente, so daß sich für l_1 und l_2 acht verschiedene Lösungen ergeben. Filtert man nachträglich die Lösungen heraus, die in negative Längen resultieren würden, erhält man die folgende eindeutige Lösung:

$$\begin{aligned} l_1 &= \sqrt{\frac{-2axy^2 - 2x^3a + 2aby\sqrt{y^2 + x^2} + 2x^2y^2 + x^4 + y^2b^2 + x^2b^2 + x^2a^2 + y^4 + y^2a^2}{y^2 + x^2}}, \\ l_2 &= \sqrt{\frac{2axy^2 + 2x^3a + 2aby\sqrt{y^2 + x^2} + 2x^2y^2 + x^4 + y^2b^2 + x^2b^2 + x^2a^2 + y^4 + y^2a^2}{y^2 + x^2}}. \end{aligned}$$

4.4 Kinematische Transformation

Zuerst setzen wir mit `setUndets([x1,y1,x2,y2,x,y])` und `setConsts([l1,l2,a,b])` die Unbekannten und Konstanten fest.

Die H-Basis können wir sofort und ohne große Probleme berechnen:

$$H := [x_2^2 + y_2^2 + 2ax_2 + a^2 - l_2^2, 4ax + l_1^2 - l_2^2, -16a^2y^2 - 32a^3x_2 - 16b^2a^2 - 16a^4 + 16l_2^2a^2 - l_1^4 + 2l_1^2l_2^2 - l_2^4, 4ay_2y + (l_2^2 + 8a^2 - l_1^2)x_2 + 4b^2a + 4a^3 - 4l_2^2a, 2ax_1 + 2ax_2 + l_1^2 - l_2^2, y_1 + y_2 - 2y, -32a^3x_2y_2 + (-4l_1^2a + 4l_2^2a + 32a^3)x_2y + (16l_2^2a^2 - l_1^4 - 16b^2a^2 - 16a^4 + 2l_1^2l_2^2 - l_2^4)y_2 + (16b^2a^2 - 16l_2^2a^2 + 16a^4)y].$$

Werfen wir nun einen Blick auf den Normalformraum beziehungsweise die Basis hiervon. Diese ist

$$P = [1, -y_1 + y_2, -x_1 + x_2, 2y_1 + y, -2x_1^2 + x_1x_2 + 2y_1^2 - y_1y_2 - 2x_2^2 + 2y_2^2, (-l_2^2 + 8a^2 + l_1^2)x_1y_1 + (l_2^2 + 8a^2 - l_1^2)x_1y_2 + 8a^2x_1y + (l_2^2 - 8a^2 - l_1^2)y_1x_2 + (-l_2^2 - 8a^2 + l_1^2)x_2y_2 - 8a^2x_2y].$$

Die Dimension des Raumes ist sechs, das heißt, wir werden sechs Lösungen für unser Problem erhalten. Dies ist insofern etwas verwunderlich, da es aus Ingenieurssicht eigentlich nur eine Lösung geben kann.

Leider ist es nicht mehr möglich mittels `solveH` die symbolischen Lösungen zu berechnen. Daher schauen wir uns nur zwei Beispiele an.

Beispiel 1: Wir substituieren die Konstanten in H mit $\{a = 10, b = 2, l_1 = 17, l_2 = 17\}$.

Die H-Basis ist dann:

$$H = [x_2^2 + 20x_2 - 189 + y_2^2, 20x_1 + 20x_2, y_1 + y_2 - 2y, 40x, -1600y^2 - 32000x_2 + 296000, 40y_2y + 800x_2 - 7400, -32000x_2y_2 + 32000x_2y + 296000y_2 - 296000y].$$

Wie oben schon gesagt, hat der Normalformraum die Dimension sechs; die sechs Lösungen - mit `solveH` berechnet - sind also

$$x_1 = 2, y_1 = 15, x_2 = -2, y_2 = 15, x = 0, y = 15,$$

$$x_1 = 2, y_1 = -15, x_2 = -2, y_2 = -15, x = 0, y = -15,$$

$$x_1 = -2, y_1 = \sqrt{145}, x_2 = 2, y_2 = \sqrt{145}, x = 0, y = \sqrt{145},$$

$$\begin{aligned}
 x_1 &= -2, y_1 = -\sqrt{145}, x_2 = 2, y_2 = -\sqrt{145}, x = 0, y = -\sqrt{145}, \\
 x_1 &= \frac{-37}{4}, y_1 = \frac{3}{4}\sqrt{145}i, x_2 = \frac{37}{4}, y_2 = -\frac{3}{4}\sqrt{145}i, x = 0, y = 0 \text{ und} \\
 x_1 &= \frac{-37}{4}, y_1 = -\frac{3}{4}\sqrt{145}i, x_2 = \frac{37}{4}, y_2 = \frac{3}{4}\sqrt{145}i, x = 0, y = 0.
 \end{aligned}$$

Zuerst einmal sieht man, daß sich die Lösungen 2, 4 und 6 dadurch ergeben, daß man die Lösungen 1, 3 und 5 einfach an der x -Achse spiegelt. Lösung 5 ist die komplexe Lösung, aus der sich der uninteressante Fall ergibt, daß $x = y = 0$ ist. Lösung 1 wäre die "richtige" beziehungsweise gute Lösung für unser Problem. Beachtung schenken sollte man aber vor allem Lösung 3, die den Fall löst, daß sich beide Gelenkstangen überkreuzen. Hierfür müßte man zusätzliche Gleichungen finden, die man dem Gleichungssystem hinzufügt, damit dies nicht mehr geschehen kann.

Beispiel 2: Diesmal substituieren wir mit $\{a = 10, b = 2, l_1 = 22, l_2 = 10\}$.

Die H-Basis ist dann:

$$H = [x_2^2 + y_2^2 + 20x_2, 40x + 384, -1600y^2 - 32000x_2 - 153856, 40y_2y + 416x_2 + 160, 20x_1 + 20x_2 + 384, y_1 + y_2 - 2y, -32000x_2y_2 + 16640x_2y - 153856y_2 + 6400y].$$

Aber obwohl die Startwerte für die Konstanten a, b, l_1, l_2 relativ einfach aussehen, lassen sich die Lösungen schon nicht mehr (in annehmbarer) Zeit berechnen. Dies liegt unter anderem daran, daß bei der Basis-Berechnung zum Überprüfen der Lösung kein Ergebnis erzielt wird, da die Werte durch die komplexen Wurzel­ausdrücke nicht mehr exakt berechnet werden können.

Auf Grundlage des "scientific guessing" kann man die Werte für x und y dennoch in ihrer derzeitigen Form angeben. Diese kann man in Anhang 4 auf Seite 121 nachschlagen. Es sei hierzu gesagt, daß der Wert für x immer noch die Form der Lösung der einfachen 2D-Kinematik 1 hat. Die Lösung für y dagegen ist um einiges komplizierter und enthält sechste Wurzeln. Dies kommt daher, daß wir sechs potentielle Lösungen erhalten haben. Durch diesen komplizierten Ausdruck erhält man leider nach der Substitution mit festen Werten eine komplexe Zahl, deren imaginärer Anteil aber fast Null ist, so daß man zumindest eine sehr gute Näherung an die exakte Lösung erhält.

Auch wenn dieses Ergebnis etwas ernüchternd ist, wollen wir hier einen Ansatz zeigen, der uns bei der Elimination der Lösung mit gekreuzten Stangen helfen soll.

4.5 Trigonometrischer Ansatz

Einleitung

Zuerst einmal wollen wir alle trigonometrischen Gleichungen zusammenfassen. Sei dazu ein Dreieck ABC gegeben mit den Seiten a, b und c und den Winkeln α, β und γ .

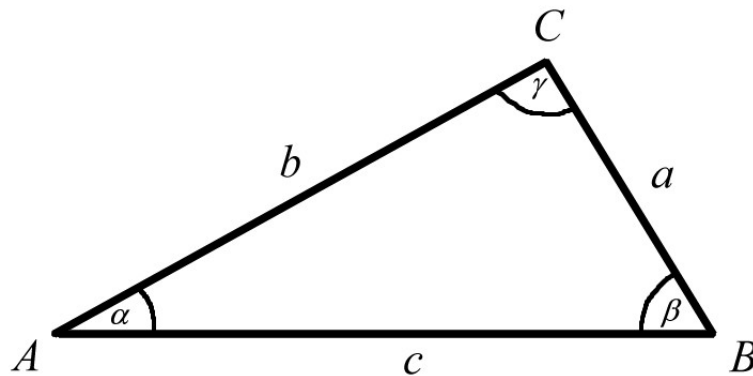


Abbildung 4.3: Dreieck ABC

In Abbildung 4.3 gelten folgenden Gleichungen³¹:

- $2bc \cdot \cos(\alpha) = b^2 + c^2 - a^2$
- $b \cdot \sin(\alpha) = a \cdot \sin(\beta)$
- $\sin(\alpha + \beta) = \sin(\alpha) \cdot \cos(\beta) + \cos(\alpha) \cdot \sin(\beta)$
- $\cos(\alpha + \beta) = \cos(\alpha) \cdot \cos(\beta) - \sin(\alpha) \cdot \sin(\beta)$
- $\cos^2(\alpha) + \sin^2(\alpha) = 1$
- $\cos(360^\circ) = 1$

³¹Die Aussagen gelten für die anderen Winkel und Seiten analog.

Idealbestimmung

Mit diesem Wissen wollen wir nun den guten Fall für die 2D-Kinematik 2, daß die Gelenkstangen sich nicht überkreuzen, eindeutig in mehreren Gleichungen beschreiben.

Hierzu sei gesagt, daß im guten Fall die vier Winkel $\alpha_1, \alpha_2, \beta_1$ und β_2 an den Gelenkpunkten $A_1 = (a, 0), A_2 = (-a, 0), B_1 = (x_1, y_1)$ und $B_2 = (x_2, y_2)$ zusammen addiert genau 360° ergeben. Im schlechten Fall würden wir ein entartetes Viereck erhalten, dessen Winkel sich nur zu einem Winkel kleiner 360° addieren würden.

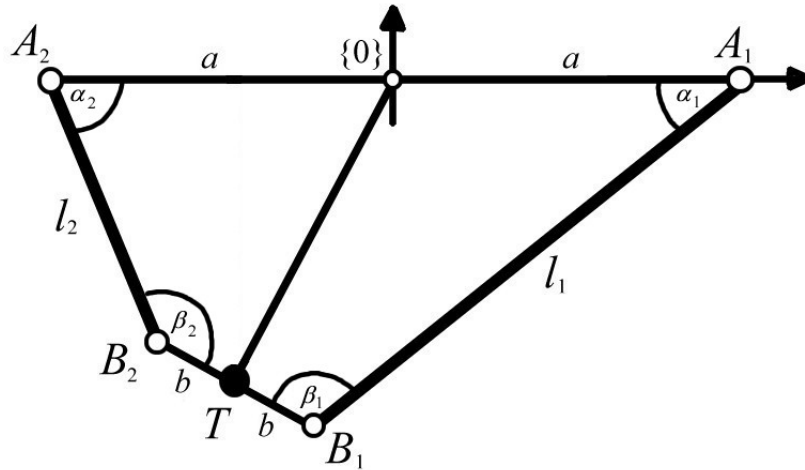


Abbildung 4.4: 2D-Kinematik 2

Es ist also $1 = \cos(360^\circ) = \cos(\alpha_1 + \alpha_2 + \beta_1 + \beta_2)$. Dies wiederum können wir über die trigonometrischen Gleichungen umformen in:

$$\begin{aligned}
 1 &= \cos(360^\circ) = \cos(\alpha_1 + \alpha_2 + \beta_1 + \beta_2) = \cos(\alpha_1 + \alpha_2) \cdot \cos(\beta_1 + \beta_2) - \sin(\alpha_1 + \alpha_2) \cdot \sin(\beta_1 + \beta_2) = \\
 &= (\cos(\alpha_1) \cdot \cos(\alpha_2) - \sin(\alpha_1) \cdot \sin(\alpha_2)) \cdot (\cos(\beta_1) \cdot \cos(\beta_2) - \sin(\beta_1) \cdot \sin(\beta_2)) - (\sin(\alpha_1) \cdot \cos(\alpha_2) + \\
 &\cos(\alpha_1) \cdot \sin(\alpha_2)) \cdot (\sin(\beta_1) \cdot \cos(\beta_2) + \cos(\beta_1) \cdot \sin(\beta_2)) = \cos(\alpha_1) \cdot \cos(\alpha_2) \cdot \cos(\beta_1) \cdot \cos(\beta_2) - \\
 &\cos(\alpha_1) \cdot \cos(\alpha_2) \cdot \sin(\beta_1) \cdot \sin(\beta_2) - \sin(\alpha_1) \cdot \sin(\alpha_2) \cdot \cos(\beta_1) \cdot \cos(\beta_2) + \sin(\alpha_1) \cdot \sin(\alpha_2) \cdot \\
 &\sin(\beta_1) \cdot \sin(\beta_2) - \sin(\alpha_1) \cdot \cos(\alpha_2) \cdot \sin(\beta_1) \cdot \cos(\beta_2) - \sin(\alpha_1) \cdot \cos(\alpha_2) \cdot \cos(\beta_1) \cdot \sin(\beta_2) - \\
 &\cos(\alpha_1) \cdot \sin(\alpha_2) \cdot \sin(\beta_1) \cdot \cos(\beta_2) - \cos(\alpha_1) \cdot \sin(\alpha_2) \cdot \cos(\beta_1) \cdot \sin(\beta_2).
 \end{aligned}$$

Betrachten wir nun nur den linken Teil der 2D-Konstruktion.³²

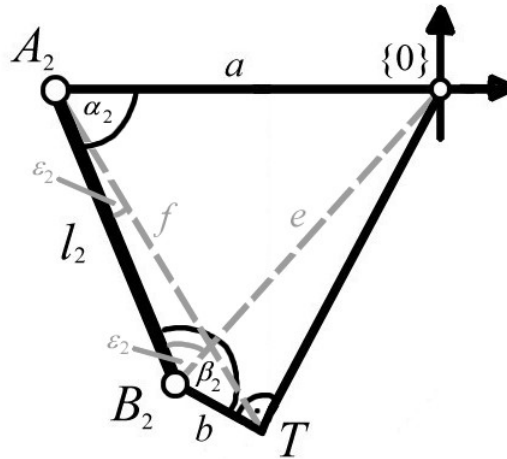


Abbildung 4.5: 2D-Kinematik 2 - linke Seite

Zuerst wollen wir $\cos(\alpha_2)$ und $\cos(\beta_2)$ ersetzen, da dies mittels des Kosinussatzes recht leicht geht. Dieser Satz liefert uns nämlich:

$$\begin{aligned} 2al_2 \cdot \cos(\alpha_2) &= a^2 + l_2^2 - e^2 \text{ mit } e = \|(x_2, y_2)\|_2 \text{ und} \\ 2bl_2 \cdot \cos(\beta_2) &= b^2 + l_2^2 - f^2 \text{ mit } f = \|(x, y) - (-a, 0)\|_2. \end{aligned}$$

Durch Einsetzen von e und f ergeben sich die Gleichungen

$$\begin{aligned} 2al_2 \cdot \cos(\alpha_2) &= a^2 + l_2^2 - x_2^2 - y_2^2, \\ 2bl_2 \cdot \cos(\beta_2) &= b^2 + l_2^2 - (x + a)^2 - y^2, \end{aligned}$$

beziehungsweise

$$\begin{aligned} \cos(\alpha_2) &= \frac{a^2 + l_2^2 - x_2^2 - y_2^2}{2al_2}, \\ \cos(\beta_2) &= \frac{b^2 + l_2^2 - (x + a)^2 - y^2}{2bl_2}. \end{aligned}$$

Nun wenden wir uns $\sin(\alpha_2)$ und $\sin(\beta_2)$ zu. Durch den Sinussatz erhält man

$$\begin{aligned} \sin(\epsilon_2) \cdot e &= \sin(\alpha_2) \cdot a \text{ beziehungsweise} \\ \sin(\epsilon_1) \cdot f &= \sin(\beta_2) \cdot b \end{aligned}$$

³²Die Berechnungen für den rechten Teil gehen analog.

mit e und f wie oben.

Da wir durch die Norm in e und f eine Wurzel in unseren Ausdruck bekommen würden, quadrieren wir die Gleichungen und erhalten

$$\begin{aligned}\sin^2(\epsilon_2) \cdot e^2 &= \sin^2(\alpha_2) \cdot a^2, \\ \sin^2(\epsilon_1) \cdot f^2 &= \sin^2(\beta_2) \cdot b^2.\end{aligned}$$

Wegen $\sin^2(\epsilon_i) + \cos^2(\epsilon_i) = 1$ für $i = 1, 2$ und mittels Einsetzen von e und f kann man dies auch schreiben als

$$\begin{aligned}(1 - \cos^2(\epsilon_2)) \cdot (x_2^2 + y_2^2) &= \sin^2(\alpha_2) \cdot a^2, \\ (1 - \cos^2(\epsilon_1)) \cdot ((x + a)^2 + y^2) &= \sin^2(\beta_2) \cdot b^2.\end{aligned}$$

Dies können wir leider noch nicht ohne Weiteres als Gleichung zu unserem Ausgangsideal hinzunehmen, da $\cos(\epsilon_2)$ und $\cos(\epsilon_1)$ noch unbekannt sind. Wie aber oben schon benutzt, erhalten wir hier wieder durch den Kosinussatz

$$\begin{aligned}2l_2e \cdot \cos(\epsilon_2) &= l_2^2 + e^2 - a^2, \\ 2l_2f \cdot \cos(\epsilon_1) &= l_2^2 + f^2 - b^2.\end{aligned}$$

Durch Umstellen nach $\cos(\epsilon_i)$, $i = 1, 2$ und Quadrieren³³ erhält man so die Gleichungen

$$\begin{aligned}\cos^2(\epsilon_2) &= \frac{(l_2^2 + e^2 - a^2)^2}{4l_2^2e^2}, \\ \cos^2(\epsilon_1) &= \frac{(l_2^2 + f^2 - b^2)^2}{4l_2^2f^2}.\end{aligned}$$

Nun können wir wieder e und f einsetzen und erhalten eine endgültige Darstellung

$$\begin{aligned}\cos^2(\epsilon_2) &= \frac{(l_2^2 + x_2^2 + y_2^2 - a^2)^2}{4l_2^2(x_2^2 + y_2^2)}, \\ \cos^2(\epsilon_1) &= \frac{(l_2^2 + (x + a)^2 + y^2 - b^2)^2}{4l_2^2((x + a)^2 + y^2)}.\end{aligned}$$

³³Wir sind schließlich nur an $\cos^2(\epsilon_i)$ interessiert.

Dieses Ergebnis setzen wir nun in unseren Gleichungen oben ein und erhalten

$$\left(1 - \frac{(l_2^2 + x_2^2 + y_2^2 - a^2)^2}{4l_2^2(x_2^2 + y_2^2)}\right) \cdot (x_2^2 + y_2^2) = \sin^2(\alpha_2) \cdot a^2,$$

$$\left(1 - \frac{(l_2^2 + (x+a)^2 + y^2 - b^2)^2}{4l_2^2((x+a)^2 + y^2)}\right) \cdot ((x+a)^2 + y^2) = \sin^2(\beta_2) \cdot b^2.$$

Ausmultiplizieren und kürzen ergibt

$$(x_2^2 + y_2^2) - \frac{(l_2^2 + x_2^2 + y_2^2 - a^2)^2}{4l_2^2} = \sin^2(\alpha_2) \cdot a^2,$$

$$((x+a)^2 + y^2) - \frac{(l_2^2 + (x+a)^2 + y^2 - b^2)^2}{4l_2^2} = \sin^2(\beta_2) \cdot b^2.$$

Analog führt man diese Berechnungen nun für die rechte Seite der Kinematik durch.

Wir fügen nun also zuerst das Polynom

$$ca_1ca_2cb_1cb_2 - ca_1ca_2sb_1sb_2 - sa_1sa_2cb_1cb_2 + sa_1sa_2sb_1sb_2 - sa_1ca_2sb_1cb_2 - sa_1ca_2cb_1sb_2 - ca_1sa_2sb_1cb_2 - ca_1sa_2cb_1sb_2 - 1$$

zu unserer Idealbasis F hinzu. Da wir aber alle Kosinus explizit angeben konnten, ersetzen wir diese hier sofort, damit wir vier Unbekannte weniger in unseren Gleichungen haben.

Es ergibt sich also das Polynom

$$\frac{(a^2+l_1^2-x_1^2-y_1^2)(a^2+l_2^2-x_2^2-y_2^2)(b^2+l_1^2-(x-a)^2-y^2)(b^2+l_2^2-(x+a)^2-y^2)}{16a^2l_1^2l_2^2b^2} - \frac{(a^2+l_1^2-x_1^2-y_1^2)(a^2+l_2^2-x_2^2-y_2^2)sb_1sb_2}{4a^2l_1l_2}$$

$$- \frac{sa_1sa_2(b^2+l_2^2-(x-a)^2-y^2)(b^2+l_2^2-(x+a)^2-y^2)}{4b^2l_1l_2} + sa_2sa_2sb_1sb_2 - \frac{sa_1(a^2+l_2^2-x_2^2-y_2^2)sb_1(b^2+l_2^2-(x+a)^2-y^2)}{4al_2^2b}$$

$$\frac{sa_1(a^2+l_2^2-x_2^2-y_2^2)(b^2+l_1^2-(x-a)^2-y^2)sb_2}{4al_2bl_1} - \frac{(a^2+l_1^2-x_1^2-y_1^2)sa_2sb_1(b^2+l_2^2-(x+a)^2-y^2)}{4al_2bl_1}$$

$$- \frac{(a^2+l_1^2-x_1^2-y_1^2)sa_2(b^2+l_1^2-(x-a)^2-y^2)sb_2}{4al_2^2b} - 1.$$

Die Unbekannten sa_1, sa_2, sb_1, sb_2 können wir leider nicht direkt angeben. Hierfür haben wir aber zumindest die Polynome

$$sa_1^2a^2 - x_1^2 - y_1^2 + \frac{(l_1^2 + x_1^2 + y_1^2 - a^2)^2}{4l_1^2},$$

$$sa_2^2a^2 - x_2^2 - y_2^2 + \frac{(l_2^2 + x_2^2 + y_2^2 - a^2)^2}{4l_2^2},$$

$$sb_1^2b^2 - (x-a)^2 - y^2 + \frac{(l_1^2 + (x-a)^2 + y^2 - b^2)^2}{4l_1^2}$$

$$sb_2^2b^2 - (x+a)^2 - y^2 + \frac{(l_2^2 + (x+a)^2 + y^2 - b^2)^2}{4l_2^2},$$

die wir auch zu F hinzufügen, was nun wie folgt aussieht:

$$F := [y_1^2 + (a-x_1)^2 - l_1^2, y_2^2 + (a+x_2)^2 - l_2^2, x_1^2 + y_1^2 - x_2^2 - y_2^2, x^2 + y^2 + b^2 - x_1^2 - y_1^2, x^2 + y^2 + b^2 - x_2^2 - y_2^2, (x_2 - x_1)x + (y_2 - y_1)y, x_1 + x_2 - 2x, y_1 + y_2 - 2y, ((a^2 + l_1^2 - x_1^2 - y_1^2)(a^2 + l_2^2 - x_2^2 - y_2^2)(b^2 + l_1^2 - (x-a)^2 - y^2)(b^2 + l_2^2 - (x+a)^2 - y^2))/(16a^2l_1^2l_2^2b^2) - \frac{(a^2+l_1^2-x_1^2-y_1^2)(a^2+l_2^2-x_2^2-y_2^2)sb_1sb_2}{4a^2l_1l_2} - (sa_1sa_2(b^2+l_1^2-(x-a)^2-y^2)(b^2+l_2^2-(x+a)^2-y^2))/(4b^2l_1l_2) + sa_2sa_2sb_1sb_2 - (sa_1(a^2+l_2^2-x_2^2-y_2^2)sb_1(b^2+l_2^2-(x+a)^2-y^2))/(4al_2^2b) - (sa_1(a^2+l_2^2-x_2^2-y_2^2)(b^2+l_1^2-(x-a)^2-y^2)sb_2)/(4al_2bl_1) - ((a^2+l_1^2-x_1^2-y_1^2)sa_2sb_1(b^2+l_2^2-(x+a)^2-y^2))/(4al_2bl_1) - ((a^2+l_1^2-x_1^2-y_1^2)sa_2(b^2+l_1^2-(x-a)^2-y^2)sb_2)/(4al_1^2b) - 1, sa_1^2a^2 - x_1^2 - y_1^2 + \frac{(l_1^2+x_1^2+y_1^2-a^2)^2}{4l_1^2}, sa_2^2a^2 - x_2^2 - y_2^2 + \frac{(l_2^2+x_2^2+y_2^2-a^2)^2}{4l_2^2}, sb_1^2b^2 - (x-a)^2 - y^2 + \frac{(l_1^2+(x-a)^2+y^2-b^2)^2}{4l_1^2}, sb_2^2b^2 - (x+a)^2 - y^2 + \frac{(l_2^2+(x+a)^2+y^2-b^2)^2}{4l_2^2}].$$

Transformation

Wir setzen zuerst wieder die Konstanten mit `setConsts([l1,l2,a,b])` und die Unbekannten mit `setUndets([x1,y1,x2,y2,x,y,sa1,sa2,sb1,sb2])` fest.

Leider ist es nicht mehr möglich, die symbolische H-Basis oder Gröbner-Basis von F in annehmbarer Zeit zu berechnen. Wir wollen aber zumindest einen Blick auf die Anzahl der Lösungen werfen. Da die Dimension des Normalformraumes, welche die Anzahl der Lösungen bestimmt, gleich bleibt, egal ob man H-Basen oder Gröbner-Basen benutzt, berechnen wir zu unserem obigen Beispiel 1 die Gröbner-Basis von F .

Da **Maple** leider etwas kompliziert aufgebaut ist, was Ordnungen betrifft³⁴, wurde auf das Computeralgebra-System **Singular** zurückgegriffen. Dieses hat einen konsistenten Ordnungsbegriff und der Gröbner-Algorithmus ist um einiges schneller implementiert als in **Maple**.

Nach der Substituierung mit $\{a = 10, b = 2, l_1 = 17, l_2 = 17\}$ können wir nun in **Singular** eine Gröbner-Basis berechnen und zurück an **Maple** übergeben. Berechnen wir hierzu mit-

³⁴Genaue Erklärung in Anhang 2 auf Seite 116.

tels $\text{NForm}(\mathbf{F})$ die Normalformraumbasis, erhalten wir eine Menge mit 32 Elementen. Das bedeutet, unsere Hoffnung, aus den vorherigen sechs Lösungen die gespiegelten und komplexen Lösungen zu eliminieren, hat sich nicht bestätigt. Wir haben im Gegensatz dazu mehr als fünfmal so viele Lösungen erhalten. Die Erklärung hierzu ist aber ziemlich leicht, wenn man sich die Lösungen für die Winkel anschaut. Diese dürfen nämlich sowohl negativ als auch komplex werden. Dadurch haben wir also keinen Gewinn erzielt und müssen uns mit der etwas komplizierten Lösung aus Kapitel 4.4 von Seite 84 zufrieden geben.

5 3D-Kinematik 1 - Spezialfall

5.1 Konstruktion

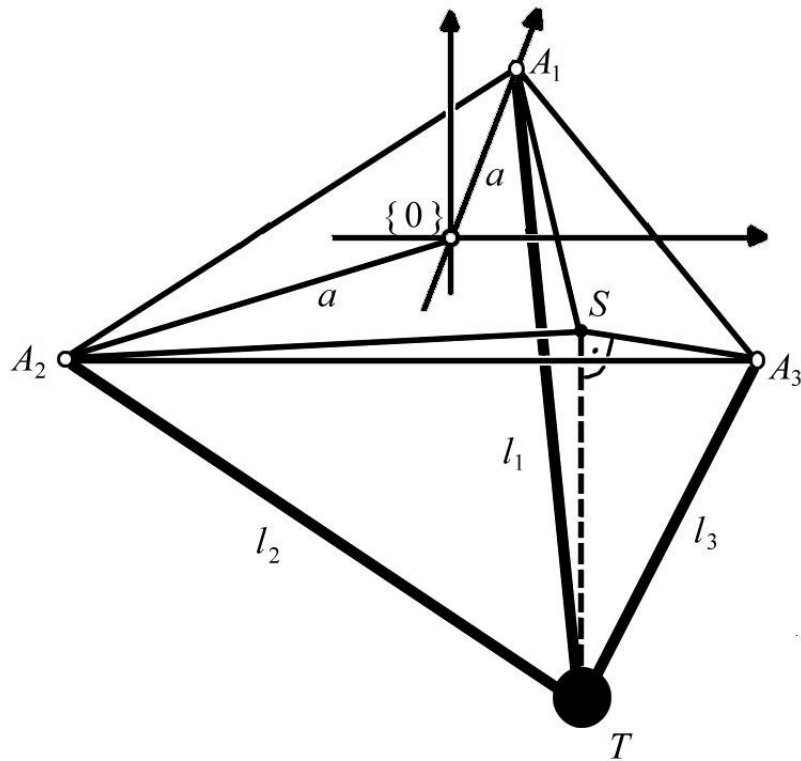


Abbildung 5.1: Spezielle 3D-Kinematik 1

Die Abbildung 5.1 beschreibt die 3D-Kinematik 1 in dem Spezialfall, in dem die drei Aufhängepunkte ein gleichseitiges Dreieck bilden, welches in der (x, z) -Ebene liegt. Zusätzlich wird der Abstand der drei Aufhängepunkte zum Ursprung 0 als konstant festgelegt, das heißt die Konstruktion ist auch im Ursprung aufgehängt.

5.2 Idealbestimmung

Da das Dreieck $A_1A_2A_3$ gleichseitig ist, ist der Abstand a vom Ursprung zu den einzelnen Punkten A_1 , A_2 und A_3 jeweils gleich groß.³⁵

³⁵Man kann die Konstruktion auch anders parametrisieren, zum Beispiel den Abstand zwischen zwei Punkten A_i und A_j , $i \neq j$, als a definieren.

Da a als Konstante vorgegeben ist, lassen sich die Punkte $A_i, i = 1, 2, 3$, leicht in Abhängigkeit von a berechnen. Es ist dann:

$$\begin{aligned} A_1 &:= (0, 0, a)^T, \\ A_2 &:= \left(-\frac{\sqrt{3}}{2}a, 0, -\frac{1}{2}a \right)^T, \\ A_3 &:= \left(\frac{\sqrt{3}}{2}a, 0, -\frac{1}{2}a \right)^T. \end{aligned}$$

Man betrachtet nun die Projektion S von T in die (x, z) -Ebene. Das bedeutet, wenn $T = (x_1, y_1, z_1)^T$, so ist $S = (x_1, 0, z_1)^T$. Nach dem Satz von Pythagoras gilt nun

$$l_i = y^2 + \|A_i - S\|_2^2, \quad i = 1, 2, 3.$$

Dies führt zu folgendem Gleichungssystem:

$$\begin{aligned} y^2 + x^2 + (a - z)^2 - l_1^2 &= 0 \\ y^2 + \left(-\frac{\sqrt{3}}{2}a - x \right)^2 + \left(-\frac{1}{2}a - z \right)^2 - l_2^2 &= 0 \\ y^2 + \left(\frac{\sqrt{3}}{2}a - x \right)^2 + \left(-\frac{1}{2}a - z \right)^2 - l_3^2 &= 0 \end{aligned}$$

Unser Ideal wird nun von der Idealbasis

$$F := [x^2 + y^2 + (a - z)^2 - l_1^2, y^2 + (-\frac{\sqrt{3}}{2}a - x)^2 + (-\frac{1}{2}a - z)^2 - l_2^2, y^2 + (\frac{\sqrt{3}}{2}a - x)^2 + (-\frac{1}{2}a - z)^2 - l_3^2]$$

erzeugt.

5.3 Inverse kinematische Transformation

Da wir für die Bestimmung des Ideals die drei Längen l_1, l_2 und l_3 explizit angeben mußten, können wir die inverse kinetische Transformation leicht bestimmen.

Es ist:

$$\begin{aligned} l_1 &= \sqrt{y^2 + x^2 + (a - z)^2}, \\ l_2 &= \sqrt{y^2 + \left(-\frac{\sqrt{3}}{2}a - x\right)^2 + \left(\frac{-1}{2}a - z\right)^2}, \\ l_3 &= \sqrt{y^2 + \left(\frac{\sqrt{3}}{2}a - x\right)^2 + \left(\frac{-1}{2}a - z\right)^2}. \end{aligned}$$

5.4 Kinematische Transformation

Wir benutzen zur Deklaration der Unbekannten und Konstanten wieder die Prozeduren `setUndets([x,y,z])` und `setConsts([a,b,l1,l2,l3])`.

Berechnen wir nun die H-Basis von F , erhalten wir:

$$H = [9a^2y^2 - 3l_1^2a^2 + l_3^4 - l_3^2l_2^2 + l_2^4 + 9a^4 - l_2^2l_1^2 - 3a^2l_2^2 - 3a^2l_3^2 + l_1^4 - l_1^2l_3^2, 6az - l_2^2 + 2l_1^2 - l_3^2, 12ax + 2\sqrt{3}l_3^2 - 2\sqrt{3}l_2^2].$$

Mittels der Multiplikationstabellen können wir dies nun leicht lösen³⁶. Es ergibt sich, daß für

$$\begin{aligned} x &= \frac{\sqrt{3}(l_2^2 - l_3^2)}{6a}, \\ y &= \frac{\sqrt{-l_2^4 + 3l_1^2a^2 - l_3^4 + l_3^2l_2^2 + 3a^2l_3^2 - 9a^4 + l_2^2l_1^2 + 3a^2l_2^2 - l_1^4 + l_1^2l_3^2}}{-3a}, \\ z &= \frac{-2l_1^2 + l_2^2 + l_3^2}{6a} \end{aligned}$$

das System gelöst wird³⁷.

Bemerkung. Wie man schön sehen kann, sind - wie schon bei der 2D-Kinematik 1 vermutet - die Lösungen für x und z relativ leicht zu bekommen. Die dritte Variable y dagegen ist um einiges komplizierter.

³⁶Natürlich kann man das Ergebnis auch sofort ablesen, da die Unbekannten in der H-Basis alle getrennt wurden.

³⁷Desweiteren gibt es wieder die gespiegelte Lösung $(x, -y, z)^T$.

6 3D-Kinematik 1 - Allgemein

6.1 Konstruktion

Im Gegensatz zur Kinematik in Kapitel 5 sind die drei Aufhängpunkte der 3D-Kinematik nun frei wählbar in allen drei Dimensionsrichtungen (ohne Abbildung).

6.2 Idealbestimmung

Der Ansatz ist der gleiche wie schon bei der allgemeinen 2D-Kinematik 1 in Kapitel 3.2. Man betrachtet jetzt aber die Projektion von T in die verschobene (x, z) -Ebene durch die drei Eckpunkte A_1, A_2 und A_3 . Damit erhält man mit Pythagoras:

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = l_i^2, \quad i = 1, 2, 3.$$

Die zugehörige Idealbasis ist dann

$$F := [(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 - l_1^2, (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 - l_2^2, (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 - l_3^2].$$

6.3 Inverse kinematische Transformation

Wieder können wir die Werte der inversen kinematischen Transformation sofort angeben:

$$\begin{aligned} l_1 &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2}, \\ l_2 &= \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2}, \\ l_3 &= \sqrt{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2}. \end{aligned}$$

6.4 Kinematische Transformation

Wir setzen vorerst nur mit `setConsts([l1,l2,l3,x1,y1,z1,x2,y2,z2,x3,y3,z3])` die Konstanten fest.

Normalerweise würden wir mit `setUndets([x,y,z])` ebenfalls die Unbekannten festsetzen, nur tritt dann folgendes Phänomen auf: Ordnet man die Unbekannten mit $x > y > z$, $y > x > z$, $z > y > x$ oder $y > z > x$ und berechnet die H-Basis, erhält man nach Substitution der Aufhängpunkte aus der speziellen 3D-Kinematik 1 ein falsches Ergebnis. Hieran

erkennt man also, daß die Reihenfolge der Unbekannten oft eine wichtige Rolle spielt. Die Erklärung, wie es dazu kommt, ist leicht. Da wir bei der speziellen 3D-Kinematik 1 den Spezialfall haben, daß $y_i = 0$ für alle $i = 1, 2, 3$, wird y bei den obigen Relationen zu stark bevorzugt. Einzig $x > z > y$ und $z > x > y$ liefern korrekte Ergebnisse³⁸. Dies könnte natürlich auch auf einen Fehler von `Maple` bei der Berechnung der symbolischen Lösung zurückzuführen sein.

Wir entscheiden uns, die Unbekannten mit `setUndets([x,z,y])` festzulegen.

Die H-Basis läßt sich nun einfach berechnen, zumal die Unbekannten wieder automatisch getrennt werden. Da die Lösung durch die viele Konstanten aber recht lang ist, konnte man diese nicht aus `Maple` heraus kopieren, so daß die Lösung hier nicht dargestellt werden kann. Es ist in dieser Hinsicht aber prinzipiell sinnvoller, zuerst die Aufhängpunkte zu substituieren und danach das System mittels `solveH()` zu lösen.

³⁸Natürlich wäre es auch hier möglich, diesen Fehler zu erzwingen, dies wäre dann aber keine sinnvolle Kinematik mehr.

7 3D-Kinematik 2

7.1 Konstruktion

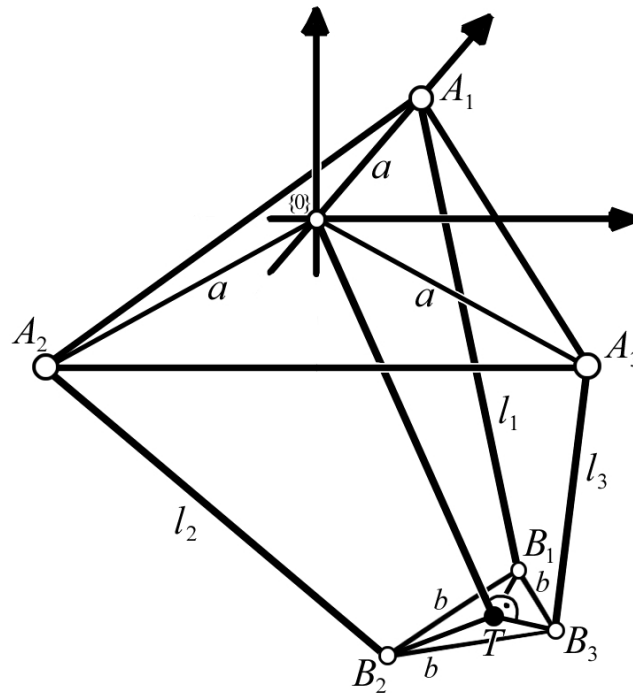


Abbildung 7.1: Spezielle 3D-Kinematik 2

Dies ist nun die komplizierteste Konstruktion von allen. Die Aufhängpunkte A_1, A_2, A_3 liegen alle in der (x, z) -Ebene und bilden ein gleichseitiges Dreieck. Dadurch haben sie alle den gleichen Abstand a zum Ursprung. Ebenso ist das Tableau der Punkte B_1, B_2, B_3 ein gleichseitiges Dreieck, welches an den äußeren Stangen mit den Längen l_1, l_2, l_3 aufgehängt ist. Zur Versteifung des Konstrukts ist noch eine Stange vom Ursprung senkrecht auf dem Tableau im Mittelpunkt T befestigt.

7.2 Idealbestimmung

Die Punkte A_1, A_2 und A_3 lassen sich, wie schon bei der speziellen 3D-Konstruktion 1 in Kapitel 5, leicht bestimmen.

Es ist

$$A_1 := (0, 0, a)^T,$$

$$A_2 := \left(-\frac{\sqrt{3}}{2}a, 0, -\frac{1}{2}a \right)^T,$$

$$A_3 := \left(\frac{\sqrt{3}}{2}a, 0, -\frac{1}{2}a \right)^T.$$

Da wir den Punkt T nicht direkt berechnen können, bestimmen wir zuerst die äußeren Punkte $B_i = (x_i, y_i, z_i)$, $i = 1, 2, 3$, und setzen dann T als Mittelpunkt dieses Dreiecks fest.

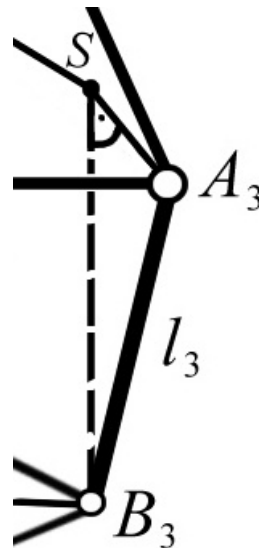


Abbildung 7.2: Eckpunkte der 3D-Kinematik 2

Zuerst wollen wir aber den Einfluß der äußeren Stäbe auf die Konstruktion betrachten. Wir schauen uns dazu nur eine Verbindungsstrecke von A_3 nach B_3 an (Abbildung 7.2). Der Punkt $S = (x_3, 0, z_3)$ ist hier die Projektion von B_3 in die (x, z) -Ebene. Mittels Pythagoras gilt nun ganz einfach

$$\|S - A_3\|_2^2 + \|S - B_3\|_2^2 = \|B_3 - A_3\|_2^2 \quad \text{beziehungsweise}$$

$$\left(x_3 - \frac{\sqrt{3}a}{2} \right)^2 + \left(z_3 + \frac{a}{2} \right)^2 + y_3^2 = l_3^2.$$

Analog erhalten wir so für die beiden anderen Stangen die Gleichungen

$$\begin{aligned} x_1^2 + (z_1 - a)^2 + y_1^2 &= l_1^2, \\ \left(x_2 + \frac{\sqrt{3}a}{2}\right)^2 + \left(z_2 + \frac{a}{2}\right)^2 + y_2^2 &= l_2^2. \end{aligned}$$

Wie in vorherigen Abschnitt erwähnt, ist das aufgehängte Tableau durch ein gleichseitiges Dreieck bestimmt, deren Eckpunkte den Abstand b voneinander haben.

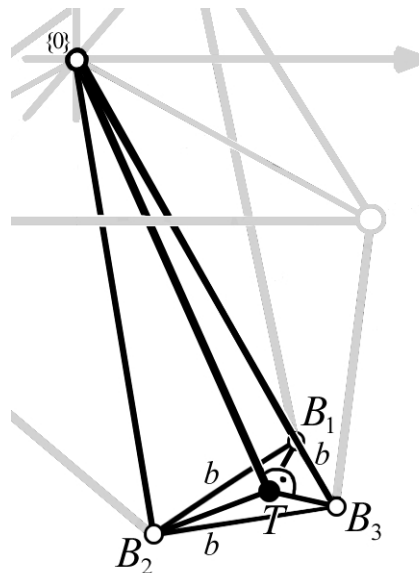


Abbildung 7.3: Tetraeder $0B_1B_2B_3$

Wir können also leicht drei neue Gleichungen einführen, nämlich

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = b^2, \quad 1 \leq i < j \leq 3.$$

Die Orthogonalitätsbeziehung können wir durch das Standard-Skalarprodukt ablesen, welches uns die Beziehung

$$(T - B_i, T) = 0, \quad i = 1, \dots, 3,$$

gibt. In Gleichungen gefaßt, erhalten wir so

$$(x - x_1)x + (y - y_1)y + (z - z_1)z = 0,$$

$$(x - x_2)x + (y - y_2)y + (z - z_2)z = 0,$$

$$(x - x_3)x + (y - y_3)y + (z - z_3)z = 0.$$

Zum Schluß müssen wir natürlich noch T in Abhängigkeit der Eckpunkte B_1, B_2, B_3 berechnen. Betrachten wir hierzu nur das Dreieck $B_1B_2B_3$ mit Mittelpunkt T in der jeweiligen Ebene, in der es liegt.

Wir wissen aus der Geometrie, daß der Mittelpunkt eines gleichseitigen Dreiecks ebenfalls der Schwerpunkt ist, der sich als Schnittpunkt der drei Seitenhalbierenden bilden läßt. Weiterhin wissen wir, daß sich die Seitenhalbierenden im Verhältnis 1:2 schneiden. In unserem Dreieck ist der Abstand vom Mittelpunkt T zum Eckpunkt B_1 genau b .

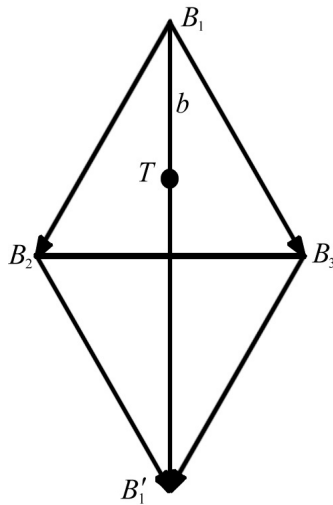


Abbildung 7.4: Dreieck $B_1B_2B_3$

Fassen wir nun die Punkte als Vektoren auf. Dann ist $B_1' = (B_2 - B_1) + (B_3 - B_1)$. Der Abstand von B_1 zu B_1' ist somit $3b$. Dadurch erhalten wir $T - B_1 = \frac{B_2 + B_3 - 2B_1}{3}$ beziehungsweise $T = B_1 + \frac{B_2 + B_3 - 2B_1}{3} = \frac{B_1 + B_2 + B_3}{3}$ und unsere letzten drei Gleichungen

$$(x_1 + x_2 + x_3) = 3x,$$

$$(y_1 + y_2 + y_3) = 3y,$$

$$(z_1 + z_2 + z_3) = 3z.$$

Wir haben nun insgesamt zwölf Gleichungen beisammen und unsere Idealbasis sieht wie folgt aus:

$$F := [x_1^2 + (z_1 - a)^2 + y_1^2 - l_1^2, \left(x_2 + \frac{\sqrt{3}a}{2}\right)^2 + \left(z_2 + \frac{a}{2}\right)^2 + y_2^2 - l_2^2, \left(x_3 - \frac{\sqrt{3}a}{2}\right)^2 + \left(z_3 + \frac{a}{2}\right)^2 + y_3^2 - l_3^2, (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - b^2, (x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 - b^2, (x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 - b^2, (x - x_1)x + (y - y_1)y + (z - z_1)z, (x - x_2)x + (y - y_2)y + (z - z_2)z, (x - x_3)x + (y - y_3)y + (z - z_3)z, (x_1 + x_2 + x_3) - 3x, (y_1 + y_2 + y_3) - 3y, (z_1 + z_2 + z_3) - 3z].$$

7.3 Inverse kinematische Transformation

Es ist nicht mehr möglich, die Größen l_1, l_2 und l_3 so leicht wie oben zu bestimmen. Es ist genau genommen unmöglich, die Größen genau zu bestimmen, wenn wir nur die Informationen aus dem Ideal zur Verfügung haben. Wenn wir feste Werte für die Konstanten x, y, z, a und b einsetzen, dann gibt es keine eindeutige Lösung für die Eckpunkte B_1, B_2, B_3 und damit auch nicht für l_1, l_2 und l_3 . Das Dreieck $B_1B_2B_3$ kann sich nämlich um die Gerade OT drehen, ohne eine der obigen Bedingungen beziehungsweise polynomialen Gleichungen zu verletzen. Dieses Problem tritt bei der kinematischen Transformation ebenfalls auf.

7.4 Kinematische Transformation

Wir berechnen wieder nur eine Gröbner-Basis mit substituierten Werten in `Maple`, um die Dimension des Normalformraumes zu bestimmen. Eine Überprüfung der Nulldimensionalität zeigt aber, daß es sich hier um ein eindimensionales Ideal handelt, weswegen wir keine Multiplikationstabellen verwenden können, um die Lösung zu berechnen.

Die Frage, wieso hier kein nulldimensionales Ideal vorliegt, wurde oben schon beantwortet und wird anhand eines Beispiels verdeutlicht.

Wir stellen uns die einfachste Substitution vor, die es bei dieser Konstruktion geben kann (Abbildung 7.5). Dies ist der Fall, wenn das gleichseitige Dreieck im Tableau identisch zum Dreieck der Aufhängung ist. Dann verlaufen die drei Stangen senkrecht nach unten und sind gleich lang. Zur Einfachheit substituieren wir hier nun mit $\{a = \sqrt{3}, b = 3, l_i = 4 \mid i = 1, 2, 3\}$.

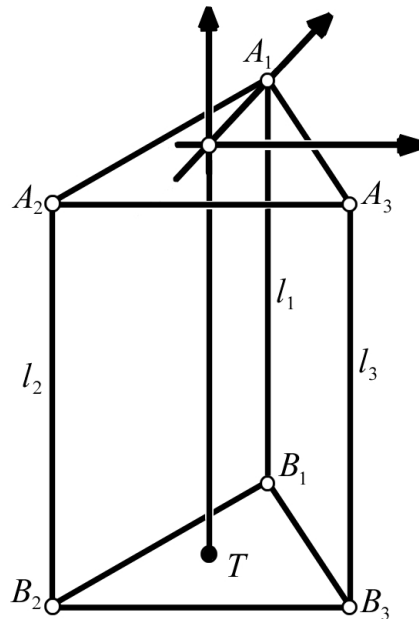


Abbildung 7.5: einfache Konstruktion A

Die Lösung, die wir verdeutlichen wollen, befindet sich dann in den Punkten

$$T = (x, y, z)^T = (0, 4, 0)^T,$$

$$B_1 = (x_1, y_1, z_1)^T = (0, 4, \sqrt{3})^T,$$

$$B_2 = (x_2, y_2, z_2)^T = \left(-\frac{3}{2}, 4, -\frac{\sqrt{3}}{2}\right)^T,$$

$$B_3 = (x_3, y_3, z_3)^T = \left(\frac{3}{2}, 4, -\frac{\sqrt{3}}{2}\right)^T.$$

Nun drehen wir das untere Dreieck im Uhrzeigersinn um den Ursprung, so daß Punkt B_2 unter A_3 , B_1 unter A_2 und B_3 unter A_1 liegt (Abbildung 7.6). Die Stangen verlaufen nun nicht mehr vertikal nach unten, sondern diagonal zum "nächsten" Punkt. Man kann nun leicht die Lösung dieser Konstruktion berechnen.

Es ist

$$T' = (0, \sqrt{7}, 0)^T,$$

$$B1' = \left(-\frac{3}{2}, \sqrt{7}, -\frac{\sqrt{3}}{2}\right)^T,$$

$$B2' = \left(\frac{3}{2}, \sqrt{7}, -\frac{\sqrt{3}}{2} \right)^T,$$

$$B3' = \left(0, \sqrt{7}, \sqrt{3} \right)^T.$$

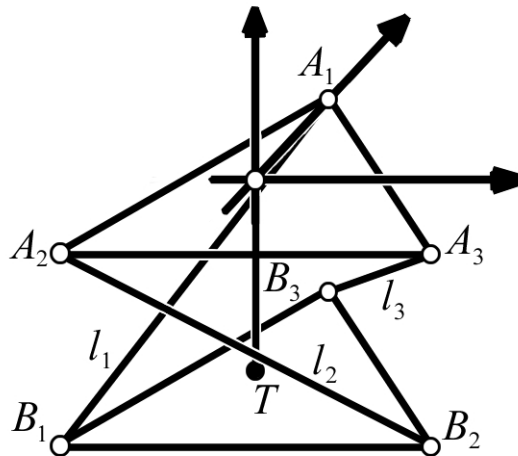


Abbildung 7.6: einfache Konstruktion B

Setzen wir diese beiden Lösungen in unser Ausgangsideal ein, so ergibt sich, daß beide unser Gleichungssystem lösen.

Durch ein einfaches Drehen des Tableaus erreicht man also, daß es zur gleichen Konfiguration der Parameter unendlich viele Lösungen gibt, was in einem nicht-nulldimensionales Ideal resultiert.

Wir wollen nun Gleichungen zum Ideal hinzufügen, die das nicht mehr zulassen.

7.5 Trigonometrischer Ansatz

Einleitung

Wieder versuchen wir mittels der trigonometrischen Funktionen Sinus und Kosinus eine Lösung zu finden.

Die Werkzeugmaschine zur Kinematik ist im Ursprung mit einem Kardan-Gelenk befestigt, welches sich nur um die x - und die z -Achse drehen beziehungsweise kippen kann. Eine Drehung um die y -Achse ist damit verboten. Diese Eigenschaft können wir mittels sogenannter Drehmatrizen darstellen. Eine Drehung um die x -Achse mit dem Winkel ϕ entspricht der

Matrix

$$M_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}.$$

Eine Drehung um die z -Achse um den Winkel ψ entspricht der Matrix

$$M_z(\psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Natürlich kann man eine Drehung um die z - und danach um die x -Achse auch als eine Matrix schreiben, indem man das Produkt von $M_x(\phi) \cdot M_z(\psi)$ berechnet. Es ist dann

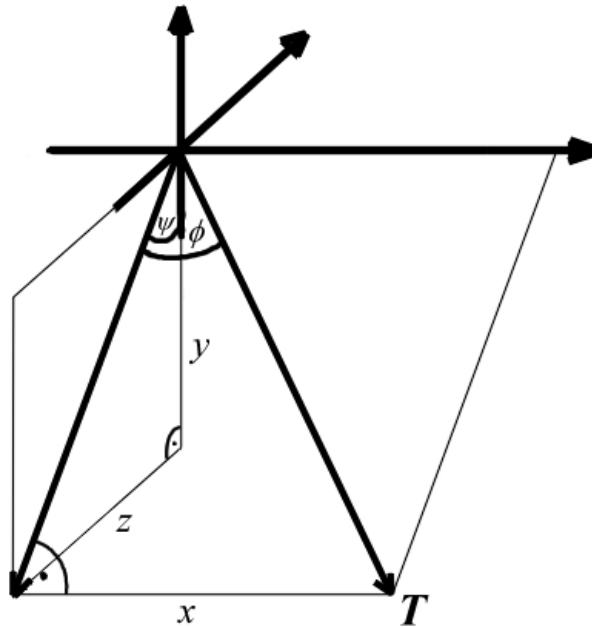
$$M(\phi, \psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) \cos(\phi) & \sin(\psi) \sin(\phi) \\ -\sin(\psi) & \cos(\psi) \cos(\phi) & \cos(\psi) \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}.$$

Idealbestimmung

Die Idee ist nun folgende: Wenn wir den Punkt $T = (x, y, z)^T$ gegeben haben, so ist die Position der Außenpunkte B_1, B_2, B_3 strikt festgelegt. Man betrachtet den Vektor $S := (0, -\sqrt{x^2 + y^2 + z^2}, 0)^T$ und berechnet die Winkel zur Drehung von S nach T . Nun betrachtet man die Vektoren B'_1, B'_2, B'_3 , welche das Dreieck $B_1B_2B_3$ in Ruhelage beschreiben. Diese Vektoren kann man explizit angeben. Mittels der Drehmatrizen und der zuvor berechneten Winkel erhält man so die explizite Lösung für die Punkte B_1, B_2, B_3 .

Zuvor müssen wir aber die Winkel berechnen. Sei nun $T = (x, y, z)^T$ gegeben. Die Winkel ψ und ϕ gehören zu den Kugelkoordinaten (Abbildung 7.7). Mit $d = \sqrt{x^2 + y^2}$ und $l = \sqrt{x^2 + y^2 + z^2}$ gilt nun

$$\begin{aligned} \sin(\psi) &= -\frac{x}{d}, \\ \cos(\psi) &= -\frac{y}{d}, \\ \sin(\phi) &= \frac{z}{l}, \\ \cos(\phi) &= \frac{d}{l}. \end{aligned}$$

Abbildung 7.7: Kugelkoordinatendarstellung von T

Dies führt nach Substitution zu folgender Drehmatrix

$$M := \begin{pmatrix} -\frac{y}{d} & -\frac{x}{l} & -\frac{xz}{dl} \\ \frac{x}{d} & -\frac{y}{l} & -\frac{yz}{dl} \\ 0 & -\frac{z}{l} & \frac{d}{l} \end{pmatrix}.$$

Nun betrachten wir das Dreieck $B'_1 B'_2 B'_3$. Da b als Konstante fest ist, können wir die Eckpunkte genau bestimmen.³⁹

Es ist

$$\begin{aligned} B'_1 &:= \left(0, -l, \frac{b}{\sqrt{3}}\right)^T, \\ B'_2 &:= \left(-\frac{b}{2}, -l, -\frac{b}{2\sqrt{3}}\right)^T, \\ B'_3 &:= \left(\frac{b}{2}, -l, -\frac{b}{2\sqrt{3}}\right)^T. \end{aligned}$$

³⁹Die y -Werte ergeben sich dabei aus dem Vektor S .

Diese drei Vektoren werden jetzt mittels der Drehmatrix M gedreht und man erhält folgende Gleichungen

$$M \cdot B'_1 = \begin{pmatrix} x - \frac{bxz}{\sqrt{3dl}} \\ y - \frac{byz}{\sqrt{3dl}} \\ z + \frac{bd}{\sqrt{3l}} \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = B_1,$$

$$M \cdot B'_2 = \begin{pmatrix} \frac{by}{2d} + x + \frac{bxz}{2\sqrt{3dl}} \\ -\frac{bx}{2d} + y + \frac{byz}{2\sqrt{3dl}} \\ z - \frac{bd}{2\sqrt{3l}} \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = B_2,$$

$$M \cdot B'_3 = \begin{pmatrix} -\frac{by}{2d} + x + \frac{bxz}{2\sqrt{3dl}} \\ \frac{bx}{2d} + y + \frac{byz}{2\sqrt{3dl}} \\ z - \frac{bd}{2\sqrt{3l}} \end{pmatrix} = \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} = B_3.$$

Aus diesen elf Gleichungen bilden wir nun Polynome, die wir zu unserer Idealbasis hinzufügen können. Wir erhalten als Basis also

$$F := [x_1^2 + (z_1 - a)^2 + y_1^2 - l_1^2, \left(x_2 + \frac{\sqrt{3}a}{2}\right)^2 + (z_2 + \frac{a}{2})^2 + y_2^2 - l_2^2, \left(x_3 - \frac{\sqrt{3}a}{2}\right)^2 + (z_3 + \frac{a}{2})^2 + y_3^2 - l_3^2, (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - b^2, (x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 - b^2, (x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 - b^2, (x - x_1)x + (y - y_1)y + (z - z_1)z, (x - x_2)x + (y - y_2)y + (z - z_2)z, (x - x_3)x + (y - y_3)y + (z - z_3)z, (x_1 + x_2 + x_3) - 3x, (y_1 + y_2 + y_3) - 3y, (z_1 + z_2 + z_3) - 3z, \sqrt{3}dl(x - x_1) - bxz, \sqrt{3}dl(y - y_1) - byz, \sqrt{3}l(z - z_1) + bd, \sqrt{3}lby + 2\sqrt{3}dl(x - x_2) + bxz, -\sqrt{3}lby + 2\sqrt{3}dl(x - x_3) + bxz, -\sqrt{3}lby + 2\sqrt{3}dl(y - y_2) + byz, 2\sqrt{3}l(z - z_2) - bd, -\sqrt{3}lby + 2\sqrt{3}dl(y - y_3) + byz, 2\sqrt{3}l(z - z_3) - bd, x^2 + y^2 - d^2, x^2 + y^2 + z^2 - l^2].$$

Inverse kinematische Transformation

Wir setzen zuerst wieder die Konstanten mit `setConsts([x,y,z,a,b])` und die Unbekannten mit `setUndets([x1,y1,z1,x2,y2,z2,x3,y3,z3,l,d,l1,l2,l3])` fest.

Die H-Basis von F läßt sich nun leicht berechnen:

$$H = [(y^2 + x^2)x_1 - 2xzz_3 - xy^2 + 2z^2x - x^3, z_1 + 2z_3 - 3z, (2x^2 + 2y^2)y_2 + 2zyz_3 + xbd - 2y^3 - 2yx^2 - 2yz^2, z_2 - z_3, (2x^2 + 2y^2)x_3 + 2xzz_3 + ybd - 2z^2x - 2xy^2 - 2x^3, (2x^2 + 2y^2)y_3 + 2zyz_3 - xbd - 2y^3 - 2yx^2 - 2yz^2, (2x^2 + 2y^2)x_2 + 2xzz_3 - ybd - 2z^2x - 2x^3 - 2xy^2, y^2 + x^2)y_1 - 2zyz_3 - yx^2 - y^3 + 2yz^2, (z^2 + y^2 + x^2)d^2 - 2y^2x^2 - x^4 - x^2z^2 - y^4 - z^2y^2, (6z^2 + 6y^2 + 6x^2)z_3d + (b\sqrt{3}x^2 + b\sqrt{3}y^2)l + (-6z^3 - 6zy^2 - 6zx^2)d, (12z^2 + 12y^2 + 12x^2)z_3^2 + (-24zy^2 -$$

$$24zx^2 - 24z^3)z_3 + 12z^4 + 12x^2z^2 - b^2x^2 - y^2b^2 + 12z^2y^2, 3bld + (6x^2\sqrt{3} + 6\sqrt{3}z^2 + 6y^2\sqrt{3})z_3 - 6\sqrt{3}z^3 - 6\sqrt{3}zx^2 - 6y^2z\sqrt{3}, 6z_3l - 6zl + \sqrt{3}bd, l^2 - x^2 - y^2 - z^2, 3l_1^2 - 12az_3 - 3x^2 - b^2 - 3z^2 + 18za - 3y^2 - 3a^2, 6y^2 + 6x^2)l_2^2 + (6xz\sqrt{3}a - 6ax^2 - 6ay^2)z_3 - 3ayb\sqrt{3}d - 6x^2a^2 - 6xz^2a\sqrt{3} - 6y^4 - 6x^3a\sqrt{3} - 6xa\sqrt{3}y^2 - 6x^4 - 12y^2x^2 - 2b^2x^2 - 6a^2y^2 - 2y^2b^2 - 6z^2y^2 - 6x^2z^2, (6y^2 + 6x^2)l_3^2 + (-6ay^2 - 6ax^2 - 6xz\sqrt{3}a)z_3 - 3ayb\sqrt{3}d - 2y^2b^2 - 6a^2y^2 - 6z^2y^2 - 6x^4 + 6x^3a\sqrt{3} - 6y^4 - 6x^2a^2 - 12y^2x^2 - 6x^2z^2 - 2b^2x^2 + 6xz^2a\sqrt{3} + 6xa\sqrt{3}y^2].$$

Die Normalformraumbasis hierzu hat 32 Elemente, das heißt, das Ideal ist nulldimensional wie gewünscht und es gibt 32 verschiedene Lösungen.

Wenn wir feste Werte für die Konstanten einsetzen, sieht man, weshalb es so viele Lösungen gibt. Durch das Quadrieren der Unbekannten l und d erhält man sowohl negative als auch positive Lösungen. Da l und d aber für die Längen von Vektoren stehen, dürfen diese natürlich nicht negativ sein. Da wir diesen Umstand leider nicht von vornherein ausschließen können, müssen wir dieses Ergebnis am Ende herausfiltern. Selbiges gilt natürlich auch für l_1, l_2 und l_3 . Diese liefert uns dann folgendes Ergebnis:

$$l_1 = (-(-6z^2y^2 - y^2b^2 - b^2x^2 + 6z^3a + 6zay^2 + 6zax^2 + 2la\sqrt{3}bd - 3a^2y^2 - 3x^2a^2 - b^2z^2 - 6x^2z^2 - 6y^2x^2 - 3x^4 - 3y^4 - 3a^2z^2 - 3z^4)/(3z^2 + 3y^2 + 3x^2))^{1/2},$$

$$l_2 = (-(-6z^4x^2 - 6y^2z^4 - 6xa\sqrt{3}y^2z^2 - 6z^3ax^2 - 6zax^4 - 6z^3ay^2 - 6zay^4 - 6y^6 - 6x^6 - 6xa\sqrt{3}y^4 - 6x^2a^2z^2 - 12x^2a^2y^2 - 12x^3a\sqrt{3}y^2 - 24y^2x^2z^2 - 2b^2x^2z^2 - 4b^2x^2y^2 - 6a^2y^2z^2 - 2y^2b^2z^2 - 6x^4a^2 - 12y^4z^2 - 18y^4x^2 - 12x^4z^2 - 18x^4y^2 - 2b^2x^4 - 6a^2y^4 - 2y^4b^2 - 12zax^2y^2 - 6x^5a\sqrt{3} - 3ay^3b\sqrt{3}d - 3ayb\sqrt{3}dz^2 - 3ayb\sqrt{3}dx^2 - 6x^3a\sqrt{3}z^2 - 3laxzbd + lax^2\sqrt{3}bd + lay^2\sqrt{3}bd)/(6z^2y^2 + 6y^4 + 12y^2x^2 + 6x^2z^2 + 6x^4))^{1/2},$$

$$l_3 = (-(-6z^4x^2 - 6y^2z^4 + 6xa\sqrt{3}y^2z^2 - 6z^3ax^2 - 6zax^4 - 6z^3ay^2 - 6zay^4 - 6y^6 - 6x^6 + 6xa\sqrt{3}y^4 - 6x^2a^2z^2 - 12x^2a^2y^2 + 12x^3a\sqrt{3}y^2 - 24y^2x^2z^2 - 2b^2x^2z^2 - 4b^2x^2y^2 - 6a^2y^2z^2 - 2y^2b^2z^2 - 6x^4a^2 - 12y^4z^2 - 18y^4x^2 - 12x^4z^2 - 18x^4y^2 - 2b^2x^4 - 6a^2y^4 - 2y^4b^2 - 12zax^2y^2 + 6x^5a\sqrt{3} - 3ay^3b\sqrt{3}d - 3ayb\sqrt{3}dz^2 - 3ayb\sqrt{3}dx^2 + 6x^3a\sqrt{3}z^2 + 3laxzbd + lax^2\sqrt{3}bd + lay^2\sqrt{3}bd)/(6z^2y^2 + 6y^4 + 12y^2x^2 + 6x^2z^2 + 6x^4))^{1/2}$$

$$\text{mit } d = \sqrt{x^2 + y^2}, l = \sqrt{x^2 + y^2 + z^2}.$$

Kinematische Transformation

Die Konstanten und Unbekannten werden dieses Mal mit `setConsts(11,12,13,a,b)` und `setUndets(x1,y1,z1,x2,y2,z2,x3,y3,z3,x,y,l,d,x,y,z)` festgesetzt.

Da sowohl `Maple` als auch `Singular` Probleme mit der Berechnung der Basis hatten, wenn die Eckpunkte der Aufhängung ein gleichseitiges Dreieck bilden, wurden die Punkte so verschoben, daß nur ganzzahlige Einträge in den Vektoren vorkommen. Die Aufhängung befindet sich nun in den Punkten

$$A_1 := (0, 0, a)^T,$$

$$A_2 := (-a, 0, -a/2)^T,$$

$$A_3 := (a, 0, -a/2)^T.$$

Die gleichen Änderungen wurden im unteren Tableau vorgenommen. Obwohl nun beide Dreiecke nicht mehr gleichseitig, sondern nur gleichschenkelig sind, hat sich in den Gleichungen, welche die Kinematik beschreiben, kaum etwas verändert. Einzig die Abstandsbedingungen der Punkte B_1, B_2, B_3 müssen etwas modifiziert werden.

Da es für beide CAS immer noch zu schwer wäre, die symbolische Lösung zu berechnen, substituieren wir wieder mit einfachen Werten. Wir setzen also $\{a = 2, b = 4, l_i = 3 \mid i = 1, 2, 3\}$.

Unser Ideal F hat nun die folgende Gestalt:

$$F = [x_1^2 + y_1^2 + (2 - z_1)^2 - 9, (-2 - x_2)^2 + y_2^2 + (-1 - z_2)^2 - 9, (2 - x_3)^2 + y_3^2 + (-1 - z_3)^2 - 9, (x - x_1)x + (y - y_1)y + (z - z_1)z, (x - x_2)x + (y - y_2)y + (z - z_2)z, (x - x_3)x + (y - y_3)y + (z - z_3)z, (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - 13, (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 - 16, (x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 - 13, x_1 + x_2 + x_3 - 3x, y_1 + y_2 + y_3 - 3y, z_1 + z_2 + z_3 - 3z, 2dl(x - x_1) - 4xz, 2dl(y - y_1) - 4yz, 2l(z - z_1) + 4d, 8ly + 4dl(x - x_2) + 4xz, -8lx + 4dl(y - y_2) + 4yz, 4l(z - z_2) - 4d, -8ly + 4dl(x - x_3) + 4xz, 8lx + 4dl(y - y_3) + 4yz, 4l(z - z_3) - 4d, x^2 + y^2 - d^2, x^2 + y^2 + z^2 - l^2].$$

Die Gröbner-Basis G können wir sowohl in `Maple` (graduiert umgekehrt lexikographisch) als auch in `Singular` (graduiert lexikographisch) berechnen. Für die weitere Verwendung beschränken wir uns auf die Basis aus `Maple`. Da die Basis mit 83 Polynomen recht groß ist,

wollen wir diese hier nicht darstellen⁴⁰. Unser Interesse gilt auch vorerst nur der Dimension des Ideals und der Basis des Normalformraumes.

Die Dimension des Ideals - mit `dimIdeal(G)` bestimmt - ergibt, daß wir es, wie gewünscht, mit einem nulldimensionalen Ideal zu tun haben. Die Berechnung der Normalformraumbasis zeigt danach, daß es vierzig Lösungen für unser Problem gibt.

Mittels der Funktion `realSol()`⁴¹ können wir bestimmen, wie viele reelle Lösungen es gibt. Es stellt sich heraus, daß von den vierzig Lösungen nur vier reell sind. Wenn wir davon ausgehen, daß die Hälfte hiervon nur wieder gespiegelt ist, so bleiben effektiv nur die gesuchte Lösung und wahrscheinlich eine Lösung mit gekreuzten Stangen übrig.

Man sollte nun weitere Gleichungen zum Ideal hinzufügen, um die komplexen Lösungen zu entfernen.

⁴⁰Sie kann aber in Anhang 5 auf Seite 122 nachgeschlagen werden.

⁴¹Nach Algorithmen von Cohen, Cuyper und Sterk [4, Real Root Counting, S.134 ff] und May [7].

Teil V

Zusammenfassung

Die Anwendungen aus Teil IV haben gezeigt, daß die Methoden zur Berechnung der Lösungen noch nicht optimal sind. Dabei bestehen folgende Probleme:

- Die Berechnung der H-Basis erfordert immer noch Gröbner-Basen.
- Die Komplexität bei wachsender Anzahl von Unbekannten steigt drastisch an.
- Aufgrund der Dimensionsberechnung bei der Bestimmung der Varietät (siehe II.7, Seite 40) ist nur exaktes Rechnen in \mathbb{Z} beziehungsweise \mathbb{Q} möglich.
- Eine Lösung ist nur für nulldimensionale Ideale berechenbar.

Punkt 3 ist insofern störend, da das exakte Rechnen die Koeffizienten der Polynombasis sehr schnell ziemlich groß werden läßt, wodurch sich die Berechnung verlangsamt. Hier müßte man gegebenenfalls auf numerische Methoden zurückgreifen und einen kleinen Fehler beim Ergebnis in Kauf nehmen. H-Basen sollten dann aber die bessere Wahl sein!

Es stellt sich jetzt die Frage, wieso man überhaupt auf H-Basen (oder Gröbner-Basen) zurückgreifen soll, wenn man ein CAS wie `Maple` hat, welches über eine eigene `solve`-Routine verfügt. Die Antwort ist, daß es nicht ganz ersichtlich ist, wie `Maple` hier vorgeht. Bei den obigen Algorithmen wurde auf tieferliegende Funktionen von `Maple` verzichtet und nur auf Standardroutinen wie Basisberechnung zurückgegriffen. Dies gewährleistet, daß die Algorithmen mit geringem Aufwand auch in andere CAS wie beispielsweise `Singular` transferiert werden könnten. Außerdem schafft es `Maple` nicht mehr eine symbolische Lösung für obige Probleme zu berechnen.

Natürlich haben H-Basen und das Berechnen der Varietät mittels Multiplikationstabellen auch noch andere Vorteile:

- Exakte Berechnung im Gegensatz zu numerischen Methoden.
- Für "kleinere" Probleme mit wenigen Unbekannten und Konstanten schnell berechenbar.

Es gibt derzeit noch keine Abschätzung, die etwas über die Größe einer H-Basis, den Grad der Polynome oder die Größe der Koeffizienten verrät, wenn man nur das Ausgangsideal kennt. Daher ist es schwierig zu entscheiden, wann man es mit einem "kleinen" Problem zu tun hat. Prinzipiell sollte man aber versuchen, die Anzahl der Konstanten gering zu halten, da diese bei der exakten Berechnung die Koeffizienten der Polynome nur unnötig vergrößern und unter Umständen das Ergebnis negativ beeinflussen.

Weiterhin hat es sich als großer Vorteil herausgestellt, vor der Berechnung der Gröbner- oder H-Basis mittels des Gröbner-Paketes von Maple eine graduiert umgekehrt lexikographische Gröbner-Basis zu berechnen. Hier dient der Befehl `Groebner[gbasis](F, tdeg(op(List_Undets)))`, wobei F unser Ideal ist und `List_Undets` zuvor mit dem Befehl `setUndets(U)` zugewiesen sein muß. Auch wenn wir die Gröbner-Basis von Maple nicht direkt verwenden können, so entsprechen sich die graduiert umgekehrt lexikographische Gröbner-Basis und die graduiert lexikographische Gröbner-Basis aus obigen Algorithmen sehr oft, was viel Berechnungszeit erspart.

Für eine Lösung unserer Kinematik müßte man sich nun speziell um folgende Probleme kümmern:

- Entfernung der komplexen Lösungen durch Betrachtung reeller Ideale.
- Entfernung von gespiegelten beziehungsweise technisch unmöglichen Lösungen (zum Beispiel mit negativen Längenparametern) durch Polynomgleichungen und/oder -ungleichungen.

Hierfür müßte man nun Ansätze finden, um das Ideal zu vergrößern oder so zu verändern, daß die Varietät des Ideals kleiner wird.

Teil VI

Anhang

1 Grafische Darstellung der Kinematiken in Maple

Für die grafische Darstellung der Kinematiken, wie auch für die Animation dient die Funktion `drawKin(K, S, T, v)`, deren Argumente im folgenden erklärt werden.

- K steht für die Kinematik und muß als String angegeben werden. Folgende Angaben sind zulässig:

"2D1s" spezielle 2D-Kinematik 1
 "2D1a" allgemeine 2D-Kinematik 1
 "2D2" spezielle 2D-Kinematik 2
 "3D1s" spezielle 3D-Kinematik 1
 "3D1a" allgemeine 3D-Kinematik 1
 "3D2" spezielle 3D-Kinematik 2

- S ist die Menge der Konstanten, die substituiert werden. Um die Kinematik zu zeichnen, müssen die konstanten Parameter festgelegt werden. S muß dabei die Form

$$\{param_1 = wert_1, param_2 = wert_2, \dots, param_n = wert_n\}$$

haben. Es folgen nun die Konstanten, denen in der jeweiligen Kinematik ein Wert zugewiesen sein muß.

"2D1s" { a }
 "2D1a" { x_1, y_1, x_2, y_2 }
 "2D2" { a }
 "3D1s" { a, b }
 "3D1a" { $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$ }
 "3D2" { a, b }

- v gibt an, ob die inverse kinematische Transformation bestimmt werden soll oder die "normale" kinematische Transformation.

'inverse' Die Liste T enthält die Koordinaten des Gelenkkopfes in x, y - beziehungsweise x, y, z -Koordinaten. Diese werden dann ohne Umrechnung auf dem Bildschirm dargestellt.

'standard' Die Liste T enthält die Längen l_1, l_2 beziehungsweise l_1, l_2, l_3 , aus der mittels der oben berechneten Lösungen die Koordinaten des Gelenkkopfes der Kinematik berechnet werden.

Die Angabe von v sollte dabei ebenfalls in Hochkommas stehen.

- T ist - je nach Angabe von v - eine Liste von Koordinaten oder Längenangaben. Bei den 2D-Kinematiken wird dabei immer eine gerade Anzahl von Werten in T erwartet. Bei den 3D-Kinematiken wird immer eine Anzahl erwartet, die durch drei teilbar ist. T hat zum Beispiel bei den 3D-Kinematiken mit $v = \text{'standard'}$ folgendes Aussehen:

$$T = [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n].$$

Besteht T nur aus zwei beziehungsweise drei Werten, so wird die Kinematik einfach nur grafisch dargestellt. Besteht die Liste aus $2n$ beziehungsweise $3n$ Werten, mit $n > 1$, beschreiben diese einzelnen "Punkte" die Stationen, die von der Kinematik abgefahren werden. Es werden zwischen zwei solchen Punkten Zwischenschritte berechnet, so daß mittels des Menüpunktes "Animate" im Maple-Menu die Kinematik animiert werden kann.

Bemerkung. Bei den inversen kinematischen Transformationen sind für die Längenangaben in T nur positive Zahlen zugelassen. Außerdem darf bei der normalen kinematischen Transformation der y -Wert nicht positiv sein, damit die Kinematik immer "hängend" dargestellt wird.

Beispiel. Wir geben den Befehl

```
drawKin("2D2", {a=10, b=3}, [22,10, 16,16, 16, 20], 'standard');
```

ein. Wir erhalten eine Grafik bei der die 2D-Kinematik 2 mit den Startwerten $l_1 = 22, l_2 = 10$ dargestellt wird.

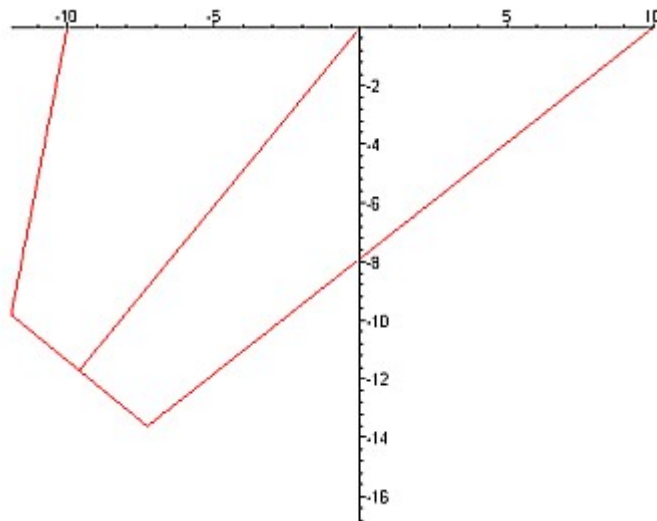


Abbildung 1.1: 2D-Kinematik 2 in Ausgangsstellung

Startet man die Animation, bewegt sich die Kinematik zuerst zu der Position, in der die Stangen die Längen $l_1 = 16, l_2 = 16$ haben und macht dort eine kurze (nicht sichtbare) Pause.

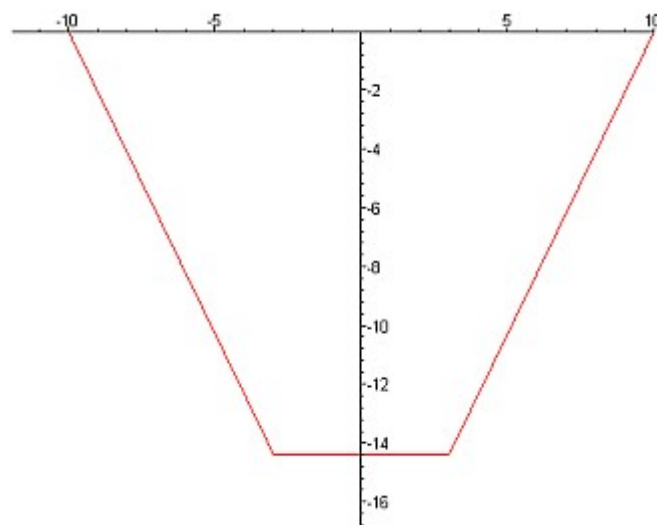


Abbildung 1.2: 2D-Kinematik 2 in Zwischenstellung

Danach bewegt sie sich weiter zu der Position, in der die Stangen die Längen $l_1 = 16, l_2 = 22$ haben.

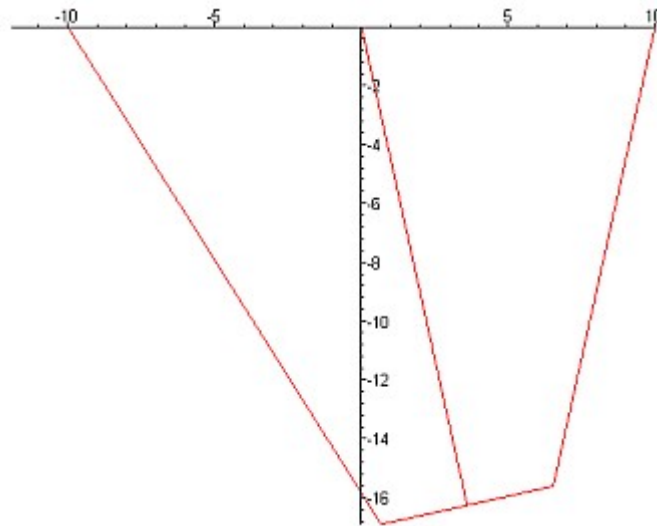


Abbildung 1.3: 2D-Kinematik 2 in Endstellung

2 Ordnungsrelationen in Maple

Maple beherrscht genau genommen drei verschiedene Ordnungen⁴²:

1. lex - Lexikographische Ordnung,
2. grlex - Graduiert lexikographische Ordnung,
3. grevlex - Graduiert umgekehrte lexikographische Ordnung.

Die lex-Order wird in Maple `plex` genannt und ist eindeutig, wodurch es nicht zu Problemen kommen kann.

Bei den graduierten Ordnungen wird es aber etwas schwerer, denn für beide Ordnungen wird in Maple der Begriff `tdeg` benutzt.

Um die Polynome in den obigen Algorithmen zu sortieren, wird die Maple-eigene `sort`-Funktion genutzt, die Polynome grundsätzlich nach der grlex-Ordnung sortiert. Nehmen wir zum Beispiel das Polynom

$$p := xy + y^3z + x^2z^2 \in \mathbb{Z}[x, y, z].$$

Wenn wir dieses mit `sort(p, [x, y, z], tdeg)` sortieren, erhalten wir

$$p = x^2z^2 + y^3z + xy.$$

Der Leitterm in unseren Algorithmen ist also x^2z^2 .

Betrachten wir nun den Leitterm, wenn wir das `Groebner`-Paket von Maple benutzen. Der Befehl `leadterm(p, tdeg([x, y, z]))` gibt y^3z zurück, obwohl man vermuten könnte, daß es der gleiche wie oben ist, da man in beiden Aufrufen die `tdeg`-Order mit der Sortierung $x > y > z$ benutzt.

Die genaue Erklärung liefert die Maple-Hilfe selbst.

⁴²Zur Erklärung der Ordnungen siehe mathematischen Ausführungen in Teil II, Kapitel 2.1 auf Seite 6.

Zitat aus der Beschreibung der Funktion `sort`:

An additional third argument, either the symbol `plex` or `tdeg`, can be given to define the ordering for the multivariate case. If `tdeg` is specified (default), then polynomials in V are sorted in total degree with ties broken by lexicographical order. [...]

Nun noch ein Zitat aus der Beschreibung der Termordnungen bei Gröbner-Basen:

In the previous cases, the leadterm of a polynomial is defined as its highest term with respect to the term order. [...] In the case of the `tdeg` form, a total degree order is used, with ties broken by inverse lexicographical order. [...]

Mit anderen Worten: Bei der reinen Sortierung von Polynomen wird die `grlex`-Ordnung benutzt, bei der Anwendung von Gröbner-Basen werden Polynome aber nach der `grevlex`-Ordnung sortiert.

`Maple` gibt uns zwar die Möglichkeit, eigene Sortierungen anzuwenden, aber leider nur für Listen und nicht für Polynome. Ist ein Polynom sortiert, so bleibt es auch so.

Schauen wir uns ein Beispiel an. Der Aufruf `x+y^2*x+x^5` gibt, wie kaum anders erwartet,

$$x + y^2x + x^5$$

aus. Nun sortieren wir dieses Polynom mittels `sort(x+y^2*x+x^5, [x,y], plex)`. Der erneute Aufruf `x+y^2*x+x^5` gibt nun

$$x^5 + y^2x + x$$

zurück, obwohl wir das gleiche Polynom wie oben eingegeben haben. "Natürlich" gilt diese Ordnung nun aber auch nur für dieses eine Polynom. Ein `y^2*x+x^5` resultiert nach wie vor in

$$y^2x + x^5,$$

obwohl nur der - für die Ordnung unwichtige - niedrigste Term weggelassen wurde.

Das Problem an den Ordnungen ist nun, daß, obwohl der `Maple`-Algorithmus zur Berechnung der Gröbner-Basis mitunter schneller ist als der Algorithmus von Buchberger, den

wir verwenden, man die resultierende Gröbner-Basis nicht zum direkten Weiterrechnen verwenden kann, da sie auf einer anderen Ordnung basiert.

Deshalb wurde auch **Singular** benutzt. Dieses Computeralgebra-System, welches auf Polynomrechnungen spezialisiert ist, ist frei erhältlich unter <http://www.singular.uni-kl.de/>. Man kann hier sowohl die graduiert lexikographische, als auch die graduiert umgekehrt lexikographische Ordnung benutzen. Daher wurde auf **Singular** zur Berechnung der Gröbner-Basis zurückgegriffen und diese Basis konnte dann in **Maple** in den Algorithmen weiter verwendet werden.

3 Lösung der allgemeinen 2D-Kinematik 1

Die zwei Lösungen für die Kinematik sind:

$$\begin{aligned}
 x = & (4x_2^3 + 4x_1y_1^2 + 4x_1y_2^2 - 4x_2x_1^2 - 4x_1x_2^2 + 4x_2y_2^2 + 4x_2y_1^2 + 4x_1l_2^2 - 4x_1l_1^2 - 4x_2l_2^2 + 4x_2l_1^2 - \\
 & 8x_2y_1y_2 + 4x_1^3 - 8x_1y_1y_2 + 4(24y_1^2y_2^2x_2x_1 - 16y_1y_2^3x_2x_1 - 16y_2y_1^3x_2x_1 - 4y_2^2l_1^2x_2x_1 - 4y_2^2l_2^2x_2x_1 - \\
 & 4y_1^2l_1^2x_2x_1 - 4y_1^2l_2^2x_2x_1 - 4y_1l_1^2l_2^2y_2 - 4y_2l_2^2x_2^2y_1 - 8y_2x_1x_2^3y_1 - 4y_2l_1^2x_1^2y_1 + 12y_1x_2^2x_1^2y_2 - 4y_2l_1^2x_2^2y_1 - \\
 & 8y_2x_1^3x_2y_1 - 4y_1l_2^2x_1^2y_2 - y_1^2l_2^4 - y_1^2l_1^4 - y_1^2x_2^4 + 2y_2^4l_2^2 - 2x_1^2y_2^4 + 6y_2y_1^5 - 15y_2^2y_1^4 + 2y_1^4l_2^2 - 15y_1^2y_2^4 + \\
 & 6y_1y_2^5 - 2x_1^2y_1^4 + 20y_2^3y_1^3 + 2y_1^4l_1^2 - 2y_2^4x_2^2 - 2x_2^2y_1^4 - y_2^2l_2^4 - y_2^2l_1^4 - y_2^2x_2^4 - y_2^2x_1^4 + 8x_1^2y_2^3y_1 + \\
 & 2y_1x_1^4y_2 + 2y_1^2l_1^2l_2^2 + 2y_2^2l_2^2x_2^2 + 4y_2^4x_2x_1 + 12y_1^2y_2^2l_2^2 - 8y_1y_2^3l_2^2 + 4y_2^2x_1x_2^3 + 4y_1^4x_2x_1 + 12y_1^2l_1^2y_2^2 - \\
 & 8y_1l_1^2y_2^3 + 8y_2x_1^2y_1^3 - 12y_2^2x_1^2y_1^2 - 8y_2y_1^3l_2^2 - 8y_2y_1^3l_1^2 + 8y_2x_2^2y_1^3 - 12y_2^2x_2^2y_1^2 + 2y_1^2l_2^2x_1^2 + 4y_2^2x_1^3x_2 + \\
 & 2y_2^2l_1^2x_1^2 + 2y_2^2l_2^2l_1^2 - 6y_1^2x_2^2x_1^2 + 2y_2^2l_1^2x_2^2 + 2y_1^2l_1^2x_2^2 + 2y_2^2l_2^2x_1^2 + 2y_1^2l_1^2x_1^2 + 2y_1^2l_2^2x_2^2 + 4y_1^2x_1^3x_2 + \\
 & 4y_1^2x_1x_2^3 - 6y_2^2x_2^2x_1^2 + 8y_2^3x_2^2y_1 + 2y_1l_2^4y_2 + 2y_1l_1^4y_2 + 2y_1x_2^4y_2 + 2l_1^2y_2^4 - y_1^2x_1^4 + 8y_2l_1^2x_2x_1y_1 + \\
 & 8y_2l_2^2x_2x_1y_1 - y_1^6 - y_2^6)^{1/2}/(2c),
 \end{aligned}$$

$$\begin{aligned}
 y = & (8y_1x_1x_2 - 4y_2l_1^2 + 4y_1y_2^2 - 4y_1x_2^2 - 4y_2x_2^2 + 4y_2l_2^2 + 4y_1l_1^2 - 4y_2x_1^2 + 4y_2y_1^2 - 4y_1l_2^2 - 4y_1^3 - \\
 & 4y_1x_1^2 - 4y_2^3 + 8x_1x_2y_2 + 4(20x_2^3x_1^3 - 15x_2^4x_1^2 + 2x_1^4l_2^2 + 6x_2^5x_1 + 6x_1^5x_2 - 15x_1^4x_2^2 + 2l_1^2x_2^4 + \\
 & 2x_1^4l_1^2 - x_1^2l_2^4 - x_1^2l_1^4 - l_1^4x_2^2 + 12y_1^2y_2^2x_2x_1 - 8y_1y_2^3x_2x_1 - 8y_2y_1^3x_2x_1 - 4y_2^2l_1^2x_2x_1 - 4y_2^2l_2^2x_2x_1 - \\
 & 4y_1^2l_1^2x_2x_1 - 4y_1^2l_2^2x_2x_1 - 4y_2l_2^2x_2^2y_1 - 16y_2x_1x_2^3y_1 - 4y_2l_1^2x_1^2y_1 + 24y_1x_2^2x_1^2y_2 - 4y_2l_1^2x_2^2y_1 - \\
 & 16y_2x_1^3x_2y_1 - 4y_1l_2^2x_1^2y_2 - 2y_1^2x_2^4 - x_1^2y_2^4 - x_1^2y_1^4 + 12x_1^2l_2^2x_2^2 + 12x_1^2l_1^2x_2^2 + 2x_1^2l_1^2l_2^2 + 2l_1^2l_2^2x_2^2 - \\
 & 8x_1^3l_2^2x_2 - 8l_1^2x_2^3x_1 - 8x_1^3l_1^2x_2 + 2x_1l_2^4x_2 + 2x_1l_1^4x_2 - 8l_2^2x_2^3x_1 - y_2^4x_2^2 - x_2^2y_1^4 - 2y_2^2x_2^4 - 2y_2^2x_1^4 + \\
 & 4x_1^2y_2^3y_1 + 4y_1x_1^4y_2 + 2y_2^2l_2^2x_2^2 + 2y_2^4x_2x_1 + 8y_2^2x_1x_2^3 + 2y_1^4x_2x_1 + 4y_2x_1^2y_1^3 - 6y_2^2x_1^2y_1^2 + 4y_2x_2^2y_1^3 - \\
 & 6y_2^2x_2^2y_1^2 + 2y_1^2l_2^2x_1^2 + 8y_2^2x_1^3x_2 + 2y_2^2l_1^2x_1^2 - 12y_1^2x_2^2x_1^2 + 2y_2^2l_1^2x_2^2 + 2y_1^2l_1^2x_2^2 + 2y_2^2l_2^2x_1^2 + 2y_1^2l_1^2x_1^2 + \\
 & 2y_1^2l_2^2x_2^2 + 8y_1^2x_1^3x_2 + 8y_1^2x_1x_2^3 - 12y_2^2x_2^2x_1^2 + 4y_2^3x_2^2y_1 + 4y_1x_2^4y_2 + 2l_2^2x_2^4 - l_2^4x_2^2 - x_1^6 - x_2^6 - \\
 & 4l_1^2l_2^2x_2x_1 - 2y_1^2x_1^4 + 8y_2l_1^2x_2x_1y_1 + 8y_2l_2^2x_2x_1y_1)^{1/2}/(2c)
 \end{aligned}$$

und

$$\begin{aligned}
 x = & (4x_2^3 + 4x_1y_1^2 + 4x_1y_2^2 - 4x_2x_1^2 - 4x_1x_2^2 + 4x_2y_2^2 + 4x_2y_1^2 + 4x_1l_2^2 - 4x_1l_1^2 - 4x_2l_2^2 + 4x_2l_1^2 - \\
 & 8x_2y_1y_2 + 4x_1^3 - 8x_1y_1y_2 - 4(24y_1^2y_2^2x_2x_1 - 16y_1y_2^3x_2x_1 - 16y_2y_1^3x_2x_1 - 4y_2^2l_1^2x_2x_1 - 4y_2^2l_2^2x_2x_1 - \\
 & 4y_1^2l_1^2x_2x_1 - 4y_1^2l_2^2x_2x_1 - 4y_1l_1^2l_2^2y_2 - 4y_2l_2^2x_2^2y_1 - 8y_2x_1x_2^3y_1 - 4y_2l_1^2x_1^2y_1 + 12y_1x_2^2x_1^2y_2 - 4y_2l_1^2x_2^2y_1 - \\
 & 8y_2x_1^3x_2y_1 - 4y_1l_2^2x_1^2y_2 - y_1^2l_2^4 - y_1^2l_1^4 - y_1^2x_2^4 + 2y_2^4l_2^2 - 2x_1^2y_2^4 + 6y_2y_1^5 - 15y_2^2y_1^4 + 2y_1^4l_2^2 - 15y_1^2y_2^4 + \\
 & 6y_1y_2^5 - 2x_1^2y_1^4 + 20y_2^3y_1^3 + 2y_1^4l_1^2 - 2y_2^4x_2^2 - 2x_2^2y_1^4 - y_2^2l_2^4 - y_2^2l_1^4 - y_2^2x_2^4 - y_2^2x_1^4 + 8x_1^2y_2^3y_1 + \\
 & 2y_1x_1^4y_2 + 2y_1^2l_1^2l_2^2 + 2y_2^2l_2^2x_2^2 + 4y_2^4x_2x_1 + 12y_1^2y_2^2l_2^2 - 8y_1y_2^3l_2^2 + 4y_2^2x_1x_2^3 + 4y_1^4x_2x_1 + 12y_1^2l_1^2y_2^2 - \\
 & 8y_1l_1^2y_2^3 + 8y_2x_1^2y_1^3 - 12y_2^2x_1^2y_1^2 - 8y_2y_1^3l_2^2 - 8y_2y_1^3l_1^2 + 8y_2x_2^2y_1^3 - 12y_2^2x_2^2y_1^2 + 2y_1^2l_2^2x_1^2 + 4y_2^2x_1^3x_2 + \\
 & 2y_1^2l_1^2x_1^2 + 2y_1^2l_2^2l_1^2 - 6y_1^2x_2^2x_1^2 + 2y_2^2l_1^2x_2^2 + 2y_1^2l_1^2x_2^2 + 2y_2^2l_2^2x_1^2 + 2y_1^2l_1^2x_1^2 + 2y_1^2l_2^2x_2^2 + 4y_1^2x_1^3x_2 + \\
 & 4y_1^2x_1x_2^3 - 6y_2^2x_2^2x_1^2 + 8y_2^3x_2^2y_1 + 2y_1l_2^4y_2 + 2y_1l_1^4y_2 + 2y_1x_2^4y_2 + 2l_1^2y_2^4 - y_1^2x_1^4 + 8y_2l_1^2x_2x_1y_1 + \\
 & 8y_2l_2^2x_2x_1y_1 - y_1^6 - y_2^6)^{1/2}/(2c),
 \end{aligned}$$

$$2y_2^2 l_1^2 x_1^2 + 2y_2^2 l_1^2 l_2^2 - 6y_1^2 x_2^2 x_1^2 + 2y_2^2 l_1^2 x_2^2 + 2y_1^2 l_1^2 x_2^2 + 2y_2^2 l_2^2 x_1^2 + 2y_1^2 l_1^2 x_1^2 + 2y_1^2 l_2^2 x_2^2 + 4y_1^2 x_1^3 x_2 + 4y_1^2 x_1 x_2^3 - 6y_2^2 x_2^2 x_1^2 + 8y_2^3 x_2^2 y_1 + 2y_1 l_2^4 y_2 + 2y_1 l_1^4 y_2 + 2y_1 x_2^4 y_2 + 2l_1^2 y_2^4 - y_1^2 x_1^4 + 8y_2 l_1^2 x_2 x_1 y_1 + 8y_2 l_2^2 x_2 x_1 y_1 - y_1^6 - y_2^6)^{1/2} / (2c),$$

$$y = (8y_1 x_1 x_2 - 4y_2 l_1^2 + 4y_1 y_2^2 - 4y_1 x_2^2 - 4y_2 x_2^2 + 4y_2 l_2^2 + 4y_1 l_1^2 - 4y_2 x_1^2 + 4y_2 y_1^2 - 4y_1 l_2^2 - 4y_1^3 - 4y_1 x_1^2 - 4y_2^3 + 8x_1 x_2 y_2 - 4(20x_2^3 x_1^3 - 15x_2^4 x_1^2 + 2x_1^4 l_2^2 + 6x_2^5 x_1 + 6x_1^5 x_2 - 15x_1^4 x_2^2 + 2l_1^2 x_2^4 + 2x_1^4 l_1^2 - x_1^2 l_2^4 - x_1^2 l_1^4 - l_1^4 x_2^2 + 12y_1^2 y_2^2 x_2 x_1 - 8y_1 y_2^3 x_2 x_1 - 8y_2 y_1^3 x_2 x_1 - 4y_2^2 l_1^2 x_2 x_1 - 4y_2^2 l_2^2 x_2 x_1 - 4y_1^2 l_1^2 x_2 x_1 - 4y_1^2 l_2^2 x_2 x_1 - 4y_2 l_2^2 x_2^2 y_1 - 16y_2 x_1 x_2^3 y_1 - 4y_2 l_1^2 x_1^2 y_1 + 24y_1 x_2^2 x_1^2 y_2 - 4y_2 l_1^2 x_2^2 y_1 - 16y_2 x_1^3 x_2 y_1 - 4y_1 l_2^2 x_1^2 y_2 - 2y_1^2 x_2^4 - x_1^2 y_2^4 - x_1^2 y_1^4 + 12x_1^2 l_2^2 x_2^2 + 12x_1^2 l_1^2 x_2^2 + 2x_1^2 l_1^2 l_2^2 + 2l_1^2 l_2^2 x_2^2 - 8x_1^3 l_2^2 x_2 - 8l_1^2 x_2^3 x_1 - 8x_1^3 l_1^2 x_2 + 2x_1 l_2^4 x_2 + 2x_1 l_1^4 x_2 - 8l_2^2 x_2^3 x_1 - y_2^4 x_2^2 - x_2^2 y_1^4 - 2y_2^2 x_2^4 - 2y_2^2 x_1^4 + 4x_1^2 y_2^3 y_1 + 4y_1 x_1^4 y_2 + 2y_2^2 l_2^2 x_2^2 + 2y_2^4 x_2 x_1 + 8y_2^2 x_1 x_2^3 + 2y_1^4 x_2 x_1 + 4y_2 x_1^2 y_1^3 - 6y_2^2 x_1^2 y_1^2 + 4y_2 x_2^2 y_1^3 - 6y_2^2 x_2^2 y_1^2 + 2y_1^2 l_2^2 x_1^2 + 8y_2^2 x_1^3 x_2 + 2y_2^2 l_1^2 x_1^2 - 12y_1^2 x_2^2 x_1^2 + 2y_2^2 l_1^2 x_2^2 + 2y_1^2 l_1^2 x_2^2 + 2y_2^2 l_2^2 x_1^2 + 2y_1^2 l_1^2 x_1^2 + 2y_1^2 l_2^2 x_2^2 + 8y_1^2 x_1^3 x_2 + 8y_1^2 x_1 x_2^3 - 12y_2^2 x_2^2 x_1^2 + 4y_2^3 x_2^2 y_1 + 4y_1 x_2^4 y_2 + 2l_2^2 x_2^4 - l_2^4 x_2^2 - x_1^6 - x_2^6 - 4l_1^2 l_2^2 x_2 x_1 - 2y_1^2 x_1^4 + 8y_2 l_1^2 x_2 x_1 y_1 + 8y_2 l_2^2 x_2 x_1 y_1)^{1/2} / (2c)$$

$$\text{mit } c = -8y_1 y_2 - 8x_2 x_1 + 4y_2^2 + 4x_1^2 + 4y_1^2 + 4x_2^2.$$

4 Lösung der speziellen 2D-Kinematik 2

Wir werden hier nicht alle sechs Lösungen der 2D-Kinematik 2 angeben, sondern nur die eine Lösung, die für uns relevant ist, das heißt, in der die Stangen nicht überkreuzt sind und die nicht aus komplexen Werten besteht. Es ist dann

$$x = \frac{l_2^2 - l_1^2}{4a} \text{ und}$$

$$y = -(24d - 864(4l_1^2b^2 - 2l_1^2l_2^2 + 4l_2^2b^2 - l_1^4 + 4l_2^2a^2 - 56b^2a^2 + 4a^2l_1^2 - l_2^4 - 4b^4 - 4a^4)/(36d) - 3(32a^4 + 3l_2^4 - 6l_1^2l_2^2 + 3l_1^4 + 32b^2a^2 - 16a^2l_1^2 - 16l_2^2a^2)/a^2)^{1/2}/12 \text{ mit}$$

$$\begin{aligned} d = & (66b^2l_2^2l_1^2 + 8a^6 + 120l_1^2b^2a^2 + 120l_2^2b^2a^2 - 12l_2^2a^4 - 264a^4b^2 - 12l_1^2a^4 - 264a^2b^4 + 8b^6 - \\ & 12b^4l_1^2 - 12b^4l_2^2 - 3l_1^4l_2^2 + 12l_1^2l_2^2a^2 - 21b^2l_2^4 - 21b^2l_1^4 - 3l_1^2l_2^4 + 6l_2^4a^2 - l_1^6 - l_2^6 + 6a^2l_1^4 + \\ & 3(-84l_1^8b^2a^2 - 192l_1^2l_2^6b^2a^2 - 84l_2^8b^2a^2 + 456a^4l_1^6b^2 + 456a^4l_2^6b^2 + 816a^4b^4l_2^4 + 816a^4b^4l_1^4 - \\ & 1200a^6l_2^4b^2 - 1200a^6l_1^4b^2 + 1080a^4l_1^2b^2l_2^4 + 1080a^4l_1^4b^2l_2^2 + 1872l_1^4b^4l_2^2a^2 + 1872l_1^2b^4l_2^4a^2 - 5472l_1^2b^6l_2^2a^2 + \\ & 432l_1^4b^6a^2 - 336l_1^6b^4a^2 + 1536a^8l_1^2b^2 + 1536a^8l_2^2b^2 + 96l_1^2b^8l_2^2 + 1536l_1^2b^8a^2 - 72l_1^2b^6l_2^4 - 72l_1^4b^6l_2^2 - \\ & 324l_1^2b^4l_2^6 - 324l_1^6b^4l_2^2 + 558l_1^4b^4l_2^4 - 1536l_1^2b^6a^4 - 12l_1^6b^2l_2^4 + 6l_1^8b^2l_2^2 + 6l_1^2b^2l_2^8 - 12l_1^4b^2l_2^6 - \\ & 1536a^6b^4l_1^2 - 1536a^6b^4l_2^2 - 336l_2^6b^4a^2 + 1536l_2^2b^8a^2 + 432l_2^4b^6a^2 - 1536l_2^2b^6a^4 - 2208a^6l_1^2l_2^2b^2 - \\ & 216l_1^4l_2^4b^2a^2 - 192l_1^6l_2^2b^2a^2 - 4704a^4b^4l_2^2l_1^2 + 3072a^4b^8 - 768a^2b^{10} - 48l_1^4b^8 + 45l_1^8b^4 + 72l_1^6b^6 + \\ & 6l_1^{10}b^2 - 48l_2^4b^8 + 72l_2^6b^6 + 45l_2^8b^4 - 768a^{10}b^2 + 3072a^8b^4 - 4608a^6b^6 + 6l_2^{10}b^2)^{1/2})^{1/3}. \end{aligned}$$

Da dieser Wert die Nullstelle eines Polynoms vom Grad 6 ist, beinhaltet er leider auch zweite und dritte Wurzeln.

5 Lösung der speziellen 3D-Kinematik 2

Wir haben das Ideal

$$F = [x_1^2 + y_1^2 + (2 - z_1)^2 - 9, (-2 - x_2)^2 + y_2^2 + (-1 - z_2)^2 - 9, (2 - x_3)^2 + y_3^2 + (-1 - z_3)^2 - 9, (x - x_1)x + (y - y_1)y + (z - z_1)z, (x - x_2)x + (y - y_2)y + (z - z_2)z, (x - x_3)x + (y - y_3)y + (z - z_3)z, (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - 13, (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 - 16, (x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 - 13, x_1 + x_2 + x_3 - 3x, y_1 + y_2 + y_3 - 3y, z_1 + z_2 + z_3 - 3z, 2dl(x - x_1) - 4xz, 2dl(y - y_1) - 4yz, 2l(z - z_1) + 4d, 8ly + 4dl(x - x_2) + 4xz, -8lx + 4dl(y - y_2) + 4yz, 4l(z - z_2) - 4d, -8ly + 4dl(x - x_3) + 4xz, 8lx + 4dl(y - y_3) + 4yz, 4l(z - z_3) - 4d, x^2 + y^2 - d^2, x^2 + y^2 + z^2 - l^2].$$

gegeben. Die von Singular berechnete Gröbner-Basis ist dann

$$G := [2z_2 + 2x_3 + z_3 - 6z - 1, 2x_2 + z_2 + 2x_3 - z_3, z_1 + z_2 + z_3 - 3z, y_1 + y_2 + y_3 - 3y, x_1 + x_2 + x_3 - 3x, 3x^2 + 3y^2 + 3z^2 + 4x_2 + 2z_2 - 8x_3 + 4z_3 + 3, d^2 - x^2 - y^2, l^2 - x^2 - y^2 - z^2, z_3l - lz + d, 24x_3z + 12y_3^2 + 15z_3^2 - 12z_3z + 3ld - 2x_2 - 4z_2 - 44x_3 + 22z_3 + 3z - 48, x_3x + y_3y + z_3z - x^2 - y^2 - z^2, 2x_3d + 3z_3d - 6dz - d, 2x_3l + z_3l - 4lz - l - 2d, 4x_3z_3 - 4x_3z - 2z_3^2 + 2z_3z + ld - z_3 + z, x_3^2 + y_3^2 + z_3^2 - 4x_3 + 2z_3 - 4, 1065080586240y_2z + 1786383365376z_2y_3 + 1953348920767494z_2l + 6063734113704z_2d + 206542758248928z_2y + 1786383365376x_3y_3 + 1955996971004934x_3l + 7522765522344x_3d + 209544348991968x_3y + 893191682688y_3z_3 - 1936510156800y_3x - 3325814431488y_3z - 1245889486089463z_3l + 5220414169812z_3d + 101770583752944z_3y + 1936510156800lx - 3639131078997272lz - 22568296567032dz - 622823516505504yz - 893191682688y_3 - 977223881439747l - 2224100572088782d - 103803919417584y, 2y_2y - z_2x + 2z_2z - 2x_3x + z_3x - 2x^2 - 2y^2 - 2z^2, 640y_2x + 320y_2z + 320z_2y_3 - 200z_2l + 40z_2d + 10z_2y + 320x_3y_3 - 200x_3l + 40x_3d + 10x_3y + 160y_3z_3 + 640y_3x - 1280y_3z + 194z_3l + 20z_3d + 5z_3y - 1280lx + 306lz - 120dz - 30yz - 160y_3 + 100l + 274d - 5y, 512y_2l + 64y_2d + 744z_2x_3 - 1044z_2z_3 - 1038z_2x + 2064z_2z - 152x_3^2 + 408x_3z_3 - 800x_3x - 944x_3z - 744y_3^2 - 512y_3l - 64y_3d - 786y_3y - 1768z_3^2 - 1031z_3x + 3766z_3z + 308ld - 2171d^2 + 42xz - 8363z^2 + 14546x_3 - 7169z_3 + 263x - 838z, 2y_2y_3 - z_2x_3 + 2z_2z_3 - 3x_3^2 + x_3z_3 - y_3^2 - z_3^2 + 4x_2 + 2z_2 + 12, 1280y_2x_3 - 640y_2z_3 + 426240y_2x + 213120y_2z + 215120z_2y_3 - 11460z_2l + 21880z_2d - 118130z_2y + 213840x_3y_3 - 11460x_3l + 21880x_3d - 118130x_3y + 108200y_3z_3 + 426240y_3x - 858480y_3z + 97939z_3l + 10940z_3d - 59065z_3y - 849920lx - 69289lz - 65640dz - 10240xy + 354390yz - 320y_2 - 107240y_3 + 5730l + 92729d + 59065y, 8y_2^2 - 16y_2y_3 + 8z_2^2 + 10z_2x_3 - 19z_2z_3 + 6z_2z + 28x_3^2 - 16x_3z_3 + 12x_3z + 8y_3^2 + 11z_3^2 - 6z_3z - 2x_2 - 128, 193651015680z^3 + 291038722955280x_2x_3 - 159746926488376x_2z_3 + 10438472640512x_2x - 10097086428000x_2z +$$

$$\begin{aligned}
& 1789148528640y_2^2 + 9952952647680y_2y_3 + 1604321280y_2l - 253966417920y_2d - \\
& 12259391010816y_2y + 167655404132760z_2x_3 - 69057813006836z_2z_3 - 20729379427112z_2x + \\
& 4906733446650z_2z + 284383602190560x_3^2 - 345178659182608x_3z_3 + 30623837827712x_3x - \\
& 58752789361350x_3z - 46249914076560y_3^2 - 66154659840y_3l - 262436290560y_3d + \\
& 39936526448040y_3y + 60336429187916z_3^2 - 27421966942612z_3x + 34852103597925z_3z - \\
& 19671171072l^2 - 20165984198916ld + 129100677120ly - 158492205495810d^2 + 258201354240dy - \\
& 89920226967552x^2 + 96031656814968xz - 89920226967552y^2 - 294896239361280z^2 + \\
& 990939359675460x_3 - 495026518930110z_3 + 16022215881108x - 9027615531069z, 99532800y^2z + \\
& 2676188160z^3 + 133791067920x_2x_3 - 31567634408x_2z_3 + 28293067120x_2x + 124510981008x_2z - \\
& 9897292800y_2^2 - 55038830400y_2y_3 + 31334400y_2l - 3303060480y_2d + 170118225840y_2y + \\
& 105517157880z_2x_3 - 158705761324z_2z_3 + 156933644666z_2x + 205829459688z_2z + \\
& 233603714880x_3^2 + 14036077456x_3z_3 - 224907465032x_3x - 392384809152x_3z - 86359111920y_3^2 - \\
& 1292083200y_3l - 3468487680y_3d - 312251947938y_3y - 268071482708z_3^2 + 184173982573z_3x + \\
& 178621924062z_3z + 3533910000l^2 + 20686187976ld + 2125854720ly - 27609167232d^2 + \\
& 3261358080dy - 12360902400x^2 - 687508924878xz - 12344313600y^2 - 71395064784z^2 + \\
& 1221061794900x_3 - 646300109106z_3 - 114253966413x + 101552961564z, 733835427840y^3 + \\
& 252255928320yz^2 - 871462836288x_2y_2 + 2515725819648x_2y_3 + 433588826880x_2l + \\
& 95922081718804x_2d - 18015803059752x_2y - 820810460016y_2z_2 - 938911749264y_2x_3 + \\
& 84376832760y_2z_3 + 13508731733760y_2x + 7893180080496y_2z + 8370408614832z_2y_3 - \\
& 55646606088720z_2l + 47550692513282z_2d - 28052185080690z_2y + 9195979610160x_3y_3 - \\
& 56323218088080x_3l + 95777067910924x_3d - 39800921324166x_3y + 2514555899928y_3z_3 + \\
& 13538591573760y_3x - 28123355813904y_3z + 50512902984759z_3l - 48753587945102z_3d + \\
& 856177111269z_3y - 26297401264128lx + 90440906154801lz + 479472865560dz - \\
& 2999688173568xy + 57067457602842yz + 113131988712y_2 - 3448199873880y_3 + \\
& 27648390089160l + 79994945316099d + 10203991221807y, 2211840xz^2 + 458640x_2x_3 - \\
& 216744x_2z_3 + 2958080x_2x - 101296x_2z - 362880y_2d - 119040y_2y + 313560z_2x_3 - 245532z_2z_3 + \\
& 1564378z_2x - 72056z_2z + 430560x_3^2 - 550752x_3z_3 - 5715616x_3x - 190576x_3z - 84240y_3^2 + \\
& 362880y_3d - 63354y_3y - 32604z_3^2 + 2851949z_3x + 414686z_3z + 196920l^2 + 4608ld - 347976d^2 + \\
& 14706xz - 640872z^2 + 1649700x_3 - 767898z_3 + 2175891x - 65268z, 1024xyz + 64x_2y_2 + \\
& 256x_2y_3 + 8316x_2d - 216x_2y + 368y_2z_2 - 368y_2x_3 + 520y_2z_3 + 6400y_2x + 2192y_2z + 3664z_2y_3 - \\
& 1316z_2l + 4342z_2d + 370z_2y + 4560x_3y_3 - 1316x_3l + 8500x_3d + 262x_3y + 1256y_3z_3 + 6400y_3x - \\
& 13808y_3z + 6875z_3l - 4066z_3d + 347z_3y - 12928lx - 3585lz - 1792dx - 552dz - 1434yz + \\
& 24y_2 - 1960y_3 + 658l + 7441d - 239y, 768xy^2 - 1152xz^2 - 620712x_2x_3 + 339788x_2z_3 - \\
& 31572x_2x + 13208x_2z - 35940y_2y - 366300z_2x_3 + 197866z_2z_3 - 13752z_2x - 61304z_2z -
\end{aligned}$$

$$\begin{aligned}
& 612720x_3^2 + 617792x_3z_3 + 21162x_3x + 293312x_3z + 119880y_3^2 + 29946y_3y - 5866z_3^2 - 11544z_3x - \\
& 148678z_3z + 5898ld + 312687d^2 + 179820x^2 + 47760xz + 179820y^2 + 588411z^2 - 2355642x_3 + \\
& 1199895z_3 + 7272x - 20076z, 235008dz^2 - 4025940x^2y + 86016xyz - 4173396y^3 - 4076628yz^2 + \\
& 103488x_2y_2 - 47040x_2y_3 + 31872x_2l + 55790316x_2d - 5561168x_2y + 24528y_2z_2 - 133392y_2x_3 + \\
& 39480y_2z_3 + 878808y_2x + 479472y_2z + 318144z_2y_3 - 37095516z_2l + 28047738z_2d - 11877412z_2y + \\
& 213984x_3y_3 - 37265628x_3l + 55811244x_3d + 1870024x_3y + 234672y_3z_3 + 954408y_3x - 1543776y_3z + \\
& 23786577z_3l - 27661674z_3d - 10031840z_3y - 1843968lx + 68868117lz - 150528dx - 267552dz + \\
& 28064952yz + 66024y_2 - 150672y_3 + 18704046l + 41892015d + 407904y, 552960dyz + 6033252x^3 + \\
& 39156552x^2z + 6033252xy^2 + 13111140xz^2 + 39433032y^2z + 43023816z^3 + 58518816x_2x_3 - \\
& 356520x_2z_3 + 23991590x_2x + 24483808x_2z - 1918080y_2^2 - 10666440y_2y_3 - 5764608y_2d + \\
& 34071198y_2y + 38471484z_2x_3 - 28740930z_2z_3 + 38872300z_2x + 40815776z_2z + 39390684x_3^2 - \\
& 43178442x_3z_3 - 77117473x_3x - 143160074x_3z - 20574396y_3^2 - 3442176y_3d - 58800321y_3y - \\
& 32330847z_3^2 + 53070680z_3x + 69538396z_3z - 563181l^2 + 1195083ld + 1721088ly - 11650473d^2 + \\
& 4396032dy - 7290720x^2 - 131441400xz - 7981920y^2 - 25152543z^2 + 321629130x_3 - 149814963z_3 - \\
& 8918640x + 35341200z, 1105920dy^2 + 207360dz^2 + 1324740x^2y + 798720xyz + 1324740y^3 + \\
& 1393860yz^2 + 8640x_2y_2 + 262080x_2y_3 + 346444x_2d + 1583440x_2y - 248400y_2z_2 - 255600y_2x_3 - \\
& 124920y_2z_3 + 5690280y_2x + 3169680y_2z + 3180480z_2y_3 - 2334660z_2l + 283142z_2d + 592580z_2y + \\
& 4498080x_3y_3 - 2334660x_3l - 3777236x_3d - 4860200x_3y + 800400y_3z_3 + 6478680y_3x - \\
& 12821280y_3z - 1192377z_3l + 1998538z_3d + 2230960z_3y - 12268800lx + 7029027lz - 1397760dx - \\
& 312480dz - 209880yz + 129240y_2 - 1821360y_3 + 1167330l + 978393d + 1572000y, 72dxz + \\
& 8xyz - 8x_2d - 12z_2y_3 - 4z_2d - 12x_3y_3 - 8x_3d - 6y_3z_3 + 36y_3z - 8z_3l + 4z_3d + 47lx + 8lz - \\
& 2dx + 6y_3 - 8d, 72dxy + 9x^3 - 6x^2z + 9xy^2 + 9xz^2 - 6y^2z - 6z^3 - 8x_2z_3 - 4z_2z_3 + 16x_3z_3 - \\
& 8z_3^2 + 6ld + 18xz - 6z_3, 27lz^2 + 96dx^2 - 216dxz + 96dy^2 + 204dz^2 - 36x^2y - 24xyz - 36y^3 - \\
& 36yz^2 + 36x_2l - 8x_2d + 18z_2l + 14z_2d - 72x_3l - 410x_3d - 60z_3l + 223z_3d - 141lx + 96lz + \\
& 6dx - 9dz + 288yz + 27l, 4320lyz - 510201d^2x + 649908d^2z - 1049112dxy + 17280dyz + \\
& 116448x^3 + 180774x^2z + 116448xy^2 - 172569xz^2 + 180774y^2z + 941274z^3 + 1204128x_2x_3 + \\
& 142728x_2z_3 - 9800x_2x + 40064x_2z - 180144y_2d + 38784y_2y + 688752z_2x_3 + 17364z_2z_3 - \\
& 19600z_2x + 80128z_2z + 361152x_3^2 - 2137872x_3z_3 + 156496x_3x - 794272x_3z - 178848y_3^2 - \\
& 107568y_3d + 25452y_3y + 744432z_3^2 - 60830z_3x + 475868z_3z - 20928l^2 - 330642ld + 54144ly - \\
& 403344d^2 + 146016dy - 224640x^2 - 267138xz - 247680y^2 - 728784z^2 + 3721320x_3 - 1250658z_3 - \\
& 46080x + 138240z, 184320ly^2 - 2386215lz^2 + 26556840d^3 - 5046480d^2y - 23252736dx^2 + \\
& 104787000dxz - 26570496dy^2 - 7530996dz^2 + 4114740x^2y + 9185400xyz + 4114740y^3 - \\
& 1208220yz^2 - 99840x_2y_2 - 752640x_2y_3 - 3181620x_2l + 6885160x_2d - 5162240x_2y + 727680y_2z_2 + \\
& 862080y_2x_3 + 346560y_2z_3 - 17698560y_2x - 9851520y_2z - 23890560z_2y_3 + 5647110z_2l +
\end{aligned}$$

$$\begin{aligned}
& 1385210z_2d - 2062240z_2y - 27768960x_3y_3 + 13601160x_3l + 7756834x_3d + 15436960x_3y - \\
& 9629760y_3z_3 - 20117760y_3x + 81932160y_3z + 140916z_3l - 5935787z_3d - 7199600z_3y + \\
& 106526145lx - 21417336lz + 1390050dx + 1611405dz - 26470080yz - 434880y_2 + 12632640y_3 - \\
& 6005175l - 5007360y, 18lxz - 24xyz + 4x_2d + 2z_2d + 4x_3d - 2z_3d + 3lx + 24dx, 12lxy + \\
& 9d^2x + 6x^2z - 9xz^2 + 6y^2z + 6z^3 + 8x_2z_3 + 4z_2z_3 - 16x_3z_3 + 8z_3^2 - 6ld - 3xz + 6z_3, 3ldz + ld + \\
& 4ly + 3d^2, 576ldy + 228lx^2 - 2484lxz - 2844ly^2 + 864lz^2 - 1050d^3 + 5898d^2y + 1986dx^2 + \\
& 1986dy^2 + 1944dz^2 - 6114x^2y + 3312xyz - 6114y^3 - 2520yz^2 - 176x_2l - 10737x_2d - 1504z_2l - \\
& 8442z_2d + 2704x_3l - 14949x_3d - 22340z_3l + 594z_3d - 414lx + 32196lz - 3312dx + 14436dz + \\
& 1440yz - 4608l - 4008d, 2ldx - 3ldz - ld - 4ly - 3d^2 + 2xz, 3z_3z^2 + 6d^2z + 4x_2z_3 + 2z_2z_3 - \\
& 8x_3z_3 + 4z_3^2 + d^2 + 4dy + 3z_3, 3z_3yz - 4lx^2 - 6yz^2 - 4dz - yz, 768z_3y^2 + 1218z_3z^2 - 11700ldx + \\
& 9ldz + 2184lxy - 576lyz + 5298d^2x + 4836d^2z + 21600dxy - 240x^3 - 6648x^2z - 240xy^2 + \\
& 1782xz^2 - 6648y^2z - 2712z^3 - 19280x_2x_3 + 11440x_2z_3 + 40x_2x - 1360x_2z - 1080y_2y - \\
& 11320z_2x_3 + 6560z_2z_3 + 80z_2x - 2720z_2z - 17120x_3^2 + 16208x_3z_3 - 1388x_3x + 13088x_3z + \\
& 3600y_3^2 + 852y_3y + 676z_3^2 + 640z_3x - 6652z_3z + 9l^2 - 825ld - 180ly + 9300d^2 + 24dy + 5448x^2 - \\
& 5406xz + 5448y^2 + 17469z^2 - 71220x_3 + 36768z_3 + 960x - 2112z, 2z_3xz + 3z_3z^2 - 2d^2x + \\
& 6d^2z - 2xz^2 + 4x_2z_3 + 2z_2z_3 - 8x_3z_3 + 4z_3^2 + d^2 + 4dy + 3z_3, 96z_3xy - 288z_3yz + 582l^2d + 24l^2y + \\
& 18ld^2 + 96lx^2 + 147lxz - 288ly^2 - 288lz^2 - 720d^3 - 96d^2y - 448dx^2 + 1314dxz - 448dy^2 - \\
& 1240dz^2 + 24x^2y - 144xyz + 24y^3 + 504yz^2 - 384x_2l + 72x_2d - 96x_2y - 192z_2y_3 - 768z_2l + \\
& 24z_2d - 192z_2y - 192x_3y_3 + 192x_3l + 2428x_3d + 48x_3y - 96y_3z_3 + 576y_3z - 86z_3l - 1226z_3d - \\
& 168z_3y + 873lx + 1142lz + 180dx + 390dz - 12xy + 336yz + 96y_3, 24z_3dz - 4l^2y - 3ld^2 - \\
& 36dz^2 + 2z_2d - 2x_3d + 3z_3d - 9dz, 110592z_3dy - 1902942z_3x^2 + 709551z_3xz - 1460574z_3y^2 - \\
& 3089160z_3z^2 + 556416l^2x - 89280l^2z - 5677263ldx + 396900ldz + 695268lxy - 349056lyz - \\
& 2307465d^2x - 1074384d^2z + 10232136dxy - 290304dyz - 110592x^3 - 2865168x^2z - 110592xy^2 + \\
& 77985xz^2 - 2865168y^2z - 1124748z^3 - 8700768x_2x_3 + 194592x_2z_3 + 370712x_2x - 824704x_2z - \\
& 5184y_2d - 683520y_2y - 5090832z_2x_3 + 389184z_2z_3 + 741424z_2x - 1649408z_2z - 7716672x_3^2 + \\
& 16834656x_3z_3 - 793456x_3x + 5489120x_3z + 1625184y_3^2 + 1156032y_3d + 136092y_3y - 4535016z_3^2 + \\
& 679082z_3x - 2926996z_3z + 641712l^2 + 73440ld + 202032ly + 3501702d^2 - 3124152dy + \\
& 2598912x^2 - 3788514xz + 2543616y^2 + 8054640z^2 - 32158296x_3 + 13001472z_3 + 442368x - \\
& 884736z, 48z_3dx - 72z_3dz + 12l^2y + 9ld^2 - 6lxz - 72dxz + 108dz^2 - 4x_2d - 8z_2d + 2x_3d - 7z_3d - \\
& 6dx + 27dz, 768z_3^2z + 66z_3l^2 + 10653z_3ld + 2928z_3ly + 7080z_3d^2 - 3948z_3x^2 - 2220z_3xz - \\
& 3948z_3y^2 - 2712z_3z^2 + 2220ldx - 10533ldz - 1872lxy - 5232lyz + 636d^2x - 6336d^2z - 3744dxy + \\
& 1296x^2z + 3444xz^2 + 1296y^2z + 3096z^3 + 9496x_2x_3 - 5028x_2z_3 + 360x_2x + 4300x_2z - 792y_2y_3 + \\
& 1272y_2y + 7208z_2x_3 - 8976z_2z_3 + 720z_2x + 7832z_2z + 16708x_3^2 - 13382x_3z_3 - 1572x_3x - \\
& 10370x_3z - 2604y_3^2 - 468y_3y - 5379z_3^2 + 1296z_3x + 22849z_3z + 33l^2 - 2589ld - 768ly + 1254d^2 -
\end{aligned}$$

$$\begin{aligned}
& 1152dy - 1128xz - 21261z^2 + 39804x_3 - 19680z_3 - 192x - 1968z, 115200z_3^2y + 857966z_3lx - \\
& 353412z_3lz - 10759920z_3dx - 2664360z_3dz - 520320z_3xy + 900900z_3yz - 1726800l^2d - \\
& 1339008ld^2 + 28800ldy + 1351728lx^2 - 552266lxz + 2280288ly^2 + 1200552lz^2 + 2713770d^3 + \\
& 2321805d^2y + 1904192dx^2 + 6107400dxz + 6811712dy^2 + 14477642dz^2 - 130080x^2y + 780480xyz - \\
& 130080y^3 + 280485yz^2 + 362880x_2y_2 + 362880x_2y_3 + 1129520x_2l + 2597500x_2y + 181440y_2z_2 - \\
& 725760y_2x_3 + 362880y_2z_3 + 387120y_2x + 644640y_2z + 1900560z_2y_3 + 2259040z_2l + 2154320z_2y + \\
& 993360x_3y_3 - 564760x_3l - 31633688x_3d - 2732030x_3y + 1222440y_3z_3 + 2035920y_3x - 5724240y_3z - \\
& 5931762z_3l + 15816844z_3d + 2192785z_3y - 9095100lx + 2825582lz + 984806dx - 596832dz + \\
& 65040xy - 520890yz + 272160y_2 - 587400y_3 - 335280y, 384z_3^2x + 24z_3l^2 + 4368z_3ld + 912z_3ly - \\
& 9825z_3d^2 + 19461z_3x^2 - 882z_3xz + 19461z_3y^2 + 9429z_3z^2 - 12196ldx - 4404ldz - 4264lxy - \\
& 912lyz - 3060d^2x + 4046d^2z - 8160dxy - 6104x^2z + 1608xz^2 - 6104y^2z - 1644z^3 + 7992x_2x_3 + \\
& 6824x_2z_3 - 3034x_2x + 524x_2z + 384y_2^2 + 1344y_2y_3 - 5298y_2y + 5172z_2x_3 + 2320z_2z_3 - 6548z_2x + \\
& 760z_2z + 6288x_3^2 - 37792x_3z_3 + 10199x_3x + 13676x_3z + 360y_3^2 + 11463y_3y + 16742z_3^2 - 7144z_3x + \\
& 11861z_3z + 12l^2 + 234ld - 48ly - 4053d^2 + 636dy + 9942xz - 21060z^2 + 1278x_3 + 10368z_3 + \\
& 1296x - 5088z, 8z_3^2d - 20z_3dz - ld^2 + 12dz^2 - z_3d + dz, 3y_3z^2 - 6z_3lx - 3z_3yz + 6lxz + 3d^2y + \\
& 3yz^2 + 4x_2y_3 + 2z_2y_3 - 8x_3y_3 + 4y_3z_3 + 3y_3, 2y_3yz - 3z_3d^2 - z_3xz + 2z_3z^2 - 2d^2x + 6d^2z - \\
& 2x^2z + 4xz^2 - 2y^2z - 2z^3 + d^2 + 4dy + xz, 16y_3y^2 - 4z_3^2l - 12z_3lx + 4z_3lz + 8z_3xy + 16z_3yz + \\
& 2l^2d + 25lxz + 6dxz - 16x^2y - 16y^3 - 16yz^2 - 8x_3d - 2z_3l + 3lx + 2lz + 12dx + 4xy, 6y_3xz + \\
& 6y_3z^2 + 12z_3lz + 3z_3yz - 12lz^2 + 6d^2y - 12yz^2 + 8x_2y_3 + 4z_2y_3 - 16x_3y_3 + 8y_3z_3 + 12dx - 3yz + \\
& 6y_3, 48y_3xy - 12y_3yz - 1800z_3ld - 360z_3ly - 2133z_3d^2 + 1224z_3x^2 + 144z_3xz + 1296z_3y^2 - \\
& 687z_3z^2 - 1170ldx + 1800ldz + 216lxy + 360lyz + 450d^2x + 714d^2z + 2160dxy - 72x^3 - 486x^2z - \\
& 72xy^2 + 18xz^2 - 630y^2z - 240z^3 - 1928x_2x_3 + 56x_2z_3 + 4x_2x - 136x_2z - 108y_2y - 1132z_2x_3 + \\
& 112z_2z_3 + 8z_2x - 272z_2z - 1712x_3^2 + 3720x_3z_3 - 134x_3x + 1232x_3z + 360y_3^2 + 90y_3y - 982z_3^2 + \\
& 64z_3x - 622z_3z - 6ld - 831d^2 - 144dy + 564x^2 - 546xz + 540y^2 + 1743z^2 - 7122x_3 + 2880z_3 + \\
& 96x - 192z, 16y_3dz + 4y_3yz - 12z_3ld + 20z_3d^2 - 16z_3dy - 24z_3x^2 + 3z_3xz - 24z_3y^2 + 27ldx + \\
& 24ldz + 12lxy + 9d^2x - 4d^2z + 24dxy + 4x^2z + 3xz^2 + 4y^2z + 8x_2x + 16z_2x - 16x_3x - 12y_3y + \\
& 14z_3x - 12z_3z + 12d^2 - 36xz + 12z^2, 2y_3dy - 3z_3dx + 2z_3dz - 2dx^2 + 6dxz - 2dy^2 - 2dz^2 + \\
& dx, 2y_3dx + 4z_3ld + z_3ly + 3z_3dy - 4ldz - lyz - 6dyz, 1548288y_3lz - 368442y_3dx + 1845612y_3dz + \\
& 2253312y_3xy - 3810456y_3yz + 630216z_3l^2 + 264708z_3ld + 1403730z_3ly + 2390103z_3d^2 - \\
& 2474307z_3dy - 494352z_3x^2 + 389052z_3xz - 1621008z_3y^2 - 4060125z_3z^2 + 1495464l^2x + \\
& 428544l^2z + 3582288ldx - 551448ldz - 3880872lxy - 2036394lyz - 800010d^2x - 1555812d^2z + \\
& 1574478dyz - 2253312x^3 + 4105602x^2z - 2253312xy^2 - 2083158xz^2 + 2654082y^2z + 4103478z^3 + \\
& 1496472x_2x_3 - 212184x_2z_3 - 720720x_2x + 851872x_2z - 73728y_2l - 916740y_2d - 146496y_2y + \\
& 1031844z_2x_3 - 424368z_2z_3 - 1441440z_2x + 1703744z_2z + 1544784x_3^2 - 2793360x_3z_3 + 479424x_3x -
\end{aligned}$$

$$\begin{aligned}
& 1423256x_3z - 283608y_3^2 - 147456y_3l + 1233180y_3d - 100680y_3y + 479502z_3^2 - 1444380z_3x + \\
& 2564668z_3z + 315108l^2 + 86868ld - 756540ly + 616077d^2 - 299097dy + 2472030xz - 563328y^2 - \\
& 5481315z^2 + 5518278x_3 - 2268864z_3, 2y_3ly - z_3lx + 2z_3lz - 2lx^2 + 4lxz - 2ly^2 - 2lz^2 + lx + \\
& 2dx, 2y_3lx + z_3ly - z_3d^2 - z_3z^2 - 4lyz + x^2z + y^2z + z^3 - 5ld - ly - 2dy, y_3ld - ldy - \\
& 2lx - yz, 2837760y_3z_3z + 40548360y_3ld + 29106380y_3ly - 18972395y_3d^2 + 31875008y_3dy + \\
& 5251410y_3x^2 + 1602940y_3xz + 4403910y_3y^2 - 20196960y_3z^2 - 105660772z_3^2l - 1117440z_3^2y - \\
& 17204842z_3lx + 148280036z_3lz + 35115888z_3dx + 56841368z_3dz + 1271250z_3xy + 14347585z_3yz + \\
& 21161160l^2d - 18060264ld^2 - 40118280ldy - 23779796lx^2 + 60610852lxz - 22553156ly^2 - \\
& 53199844lz^2 + 24150870d^3 + 12672140d^2y - 87927072dx^2 + 75555744dxz - 26560032dy^2 - \\
& 40122762dz^2 + 1271250x^2y - 2542500xyz + 1271250y^3 - 4703470yz^2 + 6696000x_2y_2 - \\
& 16449600x_2y_3 - 14107440x_2l + 22033740x_2y + 3170640y_2z_2 - 19244880y_2x_3 + 9445080y_2z_3 - \\
& 20089920y_2x + 1515360y_2z - 34318080z_2y_3 - 28214880z_2l + 7610760z_2y + 3968160x_3y_3 + \\
& 7053720x_3l - 211321544x_3d - 9280950x_3y - 28077360y_3z_3 - 2384880y_3x + 90589200y_3z - \\
& 77518406z_3l + 715605z_3y + 25657510lx + 116313866lz + 26348962dx + 5303304dz - 423750xy - \\
& 52745120yz + 6529560y_2 + 1418880y_3 - 1022160y, 4y_3z_3y - 4y_3yz + 2z_3^2x + 4z_3^2z - 4z_3x^2 - \\
& 2z_3xz - 4z_3y^2 - 8z_3z^2 - ldx + 4x^2z + 4y^2z + 4z^3 + z_3x - xz, 12y_3z_3x + 24y_3z_3z - 24y_3x^2 - \\
& 12y_3xz - 24y_3y^2 - 24y_3z^2 + 24z_3^2l - 30z_3^2y - 48z_3lz + 6z_3yz + 3ldy + 24lz^2 + 3d^2y + 3yz^2 + \\
& 4x_2y + 8z_2y + 48x_3d + 88x_3y + 6y_3x + 12z_3l - 44z_3y - 12lz - 15yz + 96y, 8y_3z_3d - 12y_3dz - \\
& 8z_3dy + 2l^2x + lyz + 12dyz - y_3d + dy, 4540416y_3z_3^2 - 32518104y_3z_3x - 30983088y_3z_3z + \\
& 4555512y_3ld + 31756736y_3ly - 6117042y_3d^2 + 22900240y_3dy + 59242800y_3x^2 + 27936142y_3xz + \\
& 56695344y_3y^2 + 24319536y_3z^2 + 87719776z_3^2l + 24203676z_3^2y - 29492704z_3lx - 103526316z_3lz + \\
& 13272216z_3dx + 35660440z_3dz + 3821184z_3xy + 41936307z_3yz + 24978336l^2d + 9412320ld^2 - \\
& 12769134ldy - 2570096lx^2 + 71868544lxz - 55630448ly^2 + 3317372lz^2 + 4928184d^3 + 3501804d^2y - \\
& 41829168dx^2 + 5589552dxz - 116773680dy^2 - 129477000dz^2 + 3821184x^2y - 7642368xyz + \\
& 3821184y^3 + 20445642yz^2 + 12448128x_2y_2 + 23853376x_2y_3 - 16652224x_2l + 49921608x_2y + \\
& 6365952y_2z_2 - 21207168y_2x_3 + 8829984y_2z_3 + 3736368y_2x + 614880y_2z + 9113216z_2y_3 - \\
& 33304448z_2l + 14391312z_2y - 6322112x_3y_3 + 8326112x_3l + 175439552x_3d - 152261280x_3y - \\
& 3980000y_3z_3 - 15492636y_3x + 24468960y_3z + 14718496z_3l + 73107240z_3y + 36758176lx + \\
& 31075120lz + 7584556dx + 27228672dz - 1273728xy + 15959611yz + 8697600y_2 - 9080832y_3 - \\
& 34408128y, 12y_3^2z + 15z_3^2z + 36z_3d^2 - 12z_3z^2 + 3ldz - 72d^2z - 32x_2x_3 - 2x_2z - 16z_2x_3 - 4z_2z + \\
& 64x_3^2 - 32x_3z_3 - 44x_3z + 22z_3z - 12d^2 + 3z^2 - 24x_3 - 48z, 12y_3^2y + 24y_3xz + 15z_3^2y + 48z_3lz - \\
& 12z_3yz + 3ldy - 48lz^2 - 2x_2y - 4z_2y - 44x_3y + 22z_3y + 3yz - 48y, 12y_3^2x - 24y_3yz + 15z_3^2x - \\
& 12z_3xz - 24z_3z^2 + 3ldx + 24x^2z + 24y^2z + 24z^3 - 2x_2x - 4z_2x - 44x_3x + 22z_3x + 3xz - \\
& 48x, 12y_3^2d + 15z_3^2d - 48z_3dz + 3ld^2 + 72dz^2 - 2x_2d - 4z_2d - 44x_3d + 22z_3d + 15dz - 48d, 12y_3^2l +
\end{aligned}$$

$$\begin{aligned}
& 15z_3^2l - 24z_3lz + 3l^2d + 48lz^2 - 2x_2l - 4z_2l - 44x_3l + 22z_3l + 15lz + 24dz - 48l, 12y_3^2z_3 - 12y_3^2z + \\
& 15z_3^3 - 15z_3^2z + 3z_3ld - 9ldz - 2x_2z_3 + 2x_2z - 4z_2z_3 + 4z_2z - 44x_3z_3 + 44x_3z + 22z_3^2 - 13z_3z - \\
& 9z^2 - 48z_3 + 48z, 48y_3^3 - 84y_3^2y + 60y_3z_3^2 + 72y_3z_3x - 48y_3z_3z + 12y_3ld - 144y_3x^2 - 144y_3xz - \\
& 144y_3y^2 + 144y_3z^2 - 93z_3^2y - 216z_3yz - 27ldy - 8x_2y_3 - 178x_2y - 16z_2y_3 - 68z_2y - 176x_3y_3 + \\
& 716x_3y + 88y_3z_3 - 24y_3x + 60y_3z - 388z_3y - 27yz - 192y_3 + 144y, 3x_3y^2 + 3x_3z^2 - 3y_3xy - \\
& 3z_3xz + 3x^3 + 3xy^2 + 3xz^2 + 4x_2x_3 + 2z_2x_3 - 8x_3^2 + 4x_3z_3 + 3x_3, x_3y_3y + x_3z_3z - x_3x^2 - x_3y^2 - \\
& x_3z^2 - y_3^2x - z_3^2x + 4x_3x - 2z_3x + 4x, 12x_3y_3^2 + 15x_3z_3^2 - 12x_3z_3z + 3x_3ld - 24y_3^2z - 24z_3^2z - \\
& 2x_2x_3 - 4z_2x_3 - 44x_3^2 + 22x_3z_3 + 99x_3z - 48z_3z - 48x_3 + 96z, 8y_2z_3d - 12y_2dz + 4x_3dy - 2z_3dy - \\
& 2l^2x + lyz - y_2d - dy, 15y_2z_3^2 + 18y_2z_3x - 12y_2z_3z + 3y_2ld - 36y_2x^2 - 36y_2xz - 36y_2y^2 + 36y_2z^2 - \\
& 24z_2^2y + 6z_2x_3y_3 - 30z_2x_3y - 12z_2y_3z_3 + 30z_2y_3x - 60z_2y_3z + 57z_2z_3y - 18z_2yz + 18x_3^2y_3 - \\
& 84x_3^2y - 6x_3y_3z_3 + 60x_3y_3x + 48x_3z_3y - 36x_3yz + 6y_3^3 - 24y_3^2y + 6y_3z_3^2 - 30y_3z_3x + 60y_3x^2 + \\
& 60y_3y^2 + 60y_3z^2 - 33z_3^2y + 18z_3yz - 2x_2y_2 - 24x_2y_3 + 6x_2y - 4y_2z_2 - 44y_2x_3 + 22y_2z_3 - 6y_2x + \\
& 15y_2z - 12z_2y_3 - 48y_2 - 72y_3 + 384y, 7680z_3^4 - 38016z_3^3x - 38976z_3^3z + 801z_3^2ld - 15048z_3^2d^2 + \\
& 18144z_3^2x^2 + 69948z_3^2xz + 18144z_3^2y^2 + 50520z_3^2z^2 + 3264z_3l^2x + 1314z_3ldx - 13824z_3ldz + \\
& 26160z_3lxy + 30096z_3d^2z - 59544z_3x^2z - 53568z_3xz^2 - 46872z_3y^2z - 5976z_3z^3 - 147l^2d^2 - \\
& 3264l^2xz - 1632ld^2y - 3816ldy^2 + 6696ldz^2 + 1632lx^2y - 28608lxyz + 1632ly^3 - 12672d^2y^2 + \\
& 26784x^2z^2 + 14112y^2z^2 + 26784z^4 - 8576x_2y_2y_3 + 25280x_2x_3z_3 - 23808x_2x_3z - 33344x_2y_3^2 + \\
& 34208x_2y_3y - 1024x_2z_3^2 + 20376x_2z_3x - 42704x_2z_3z - 1246x_2ld - 35712x_2xz + 66960x_2z^2 - \\
& 4480y_2z_2y_3 + 12160y_2x_3y_3 - 3680y_2y_3z_3 + 2976y_2y_3x - 5280y_2y_3z + 12640z_2x_3z_3 - 11904z_2x_3z - \\
& 19648z_2y_3^2 + 21568z_2y_3y - 2048z_2z_3^2 + 13968z_2z_3x - 15712z_2z_3z - 476z_2ld - 17856z_2xz + \\
& 26784z_2z^2 - 50560x_3^2z_3 + 47616x_3^2z + 3904x_3y_3^2 + 33344x_3y_3y + 2752x_3z_3^2 + 53208x_3z_3x + \\
& 157648x_3z_3z + 5012x_3ld - 168x_3d^2 + 4464x_3xz - 249984x_3z^2 + 928y_3^2z_3 - 6528y_3^2x - 2976y_3^2z - \\
& 27436y_3z_3y - 5112y_3lx - 73152y_3dx - 62112y_3dz + 2760y_3yz + 11264z_3^3 - 28350z_3^2x - 91148z_3^2z - \\
& 2758z_3ld + 5016z_3d^2 - 7614z_3xz + 125988z_3z^2 - 147ldz - 8448lxy - 1632dxy - 3816y^2z + \\
& 6696z^3 - 5568y_2y_3 + 18960x_3z_3 - 17856x_3z + 14880y_3^2 - 52128y_3y - 24576z_3^2 + 114048z_3x + \\
& 43776z_3z + 1008ld - 107136xz - 35712z^2].
\end{aligned}$$

6 Befehlsverzeichnis

Zuerst folgt eine Aufzählung aller Konstanten, die benutzt werden.

<code>basis_type</code>	Art der Basis, die benutzt wird ("G"roebner- oder "H"-Basis)
<code>List_Consts</code>	Liste aller Konstanten
<code>List_Undets</code>	Liste aller Unbekannten
<code>sp_type</code>	Art des Skalarproduktes, das benutzt wird ("M"onomiales "D"ifferential-Skalarprodukt)

Eine wichtige Konstante ist `is_basis`, welche aber mit viel Vorsicht benutzt werden sollte. Da viele Prozeduren immer wieder eine Basis aufs neue berechnen müssen, obwohl diese schon im Schritt vorher berechnet wurde, kann man über diese Konstante festlegen, daß man bereits eine Gröbner- oder H-Basis hat. Es ist also sinnvoll, bei einem beliebigen Ideal zuerst die Basis zu berechnen und dann die Konstante `is_basis` mittels `setBasis(true)` festzulegen. Danach kann man mit der Basis weiterrechnen, ohne daß diese auf ihre korrekten Eigenschaften überprüft werden muß. Man sollte dies natürlich wirklich nur mit einer echten Gröbner- oder H-Basis machen, ansonsten erhält man falsche Ergebnisse.

Nun werden alle implementierten Befehle und Funktionen⁴³ samt Kurzbeschreibung, Eingabe und Ausgabe aufgeführt. Mitunter werden Abhängigkeiten aufgeführt, die direkten Einfluß auf das Ergebnis der Funktion haben⁴⁴.

<code>aSyzMod(F)</code>	berechnet den Syzygienmodul für eine Polynommenge (S. 63) Eingabe: Liste F von Polynomen Ausgabe: Syzygienmodul für F als Liste von Vektoren
<code>canReduce(F, i)</code>	überprüft, ob eine Polynommenge reduziert werden kann (S. 47) Eingabe: Liste F von Polynomen und Position i ab der rückwärts gesucht werden soll als Ganzzahl Ausgabe: Liste mit $ F + 1$ Elementen, bei der das erste Element die Position des zu reduzierenden Elementes in F angibt

⁴³In alphabetischer Reihenfolge.

⁴⁴Natürlich ist klar, daß die Liste der Unbekannten grundsätzlich in allen Funktionen indirekt eine Rolle spielt.

- `CartUnion(M)` bildet die spezielle Vereinigung einer Menge (S. 71)
Eingabe: Liste M von Eigenwerten und -räumen, normalerweise ist $M = \text{Eigenspaces}(F)$
Ausgabe: Liste, in der Eigenwerte und -räume speziell verknüpft sind
- `coeffmi(p, a)` liest den Koeffizient zu einem bestimmten Multiindex aus (S. 57)
Eingabe: Polynom p und Multiindex a als Liste
Ausgabe: Koeffizient des Monoms von p , das dem Multiindex entspricht
- `dimId(F)` berechnet die Dimension des Ideals über Eliminationsideale
Eingabe: Liste F von Polynomen
Ausgabe: Ganzzahl i als Dimension
- `dimIdeal(F)` berechnet die Dimension des Ideals (siehe S. 35)
Eingabe: Liste F von Polynomen
Ausgabe: Ganzzahl i als Dimension
Abhängigkeiten: `List_Undets`, `basis_type`
- `drawKin(K, S, T, v)` zeichnet oder animiert eine Kinematik (Näheres in Anhang 1 auf Seite 112)
- `Eigenspaces(F)` berechnet die Eigenwerte und -räume der Multiplikationstabellen (S. 70)
Eingabe: Liste F von Polynomen
Ausgabe: verschachtelte Liste mit allen Eigenwerten und -räumen
Abhängigkeiten: `List_Undets`

facmi(a)	berechnet die Fakultät eines Multiindex Eingabe: Multiindex a als Liste Ausgabe: Fakultät von a als Ganzzahl
GBasis(F)	berechnet die Gröbner-Basis (S. 45) Eingabe: Liste F von Polynomen Ausgabe: Gröbner-Basis G mit $\langle F \rangle = \langle G \rangle$ als Liste von Polynomen
GBMatrix(F)	berechnet eine Matrix zur Gröbner-Basis-Berechnung (S. 46) Eingabe: Liste F von Polynomen Ausgabe: Matrix M mit $F = G \cdot M$, wobei G die Gröbner-Basis zu F ist
gcfs(F,p)	berechnet die Koeffizienten zur Darstellung (S. 44) Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Liste K mit Koeffizienten, so daß $K \cdot F = p - \text{Rest}$
gDivAlg(F,p)	berechnet Koeffizienten zur Darstellung und Rest (S. 43) Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Liste K mit Koeffizienten und Rest r , so daß $K \cdot F + r = p$
gleadcoeff(p)	gibt den Leitkoeffizienten bei Gröbner-Basen zurück Eingabe: Polynom p Ausgabe: Leitkoeffizient von p bezüglich grlex Order
gleadmon(p)	gibt das Leitmonom bei Gröbner-Basen zurück Eingabe: Polynom p Ausgabe: Leitmonom von p bezüglich grlex Order
gleadterm(p)	gibt den Leiterterm bei Gröbner-Basen zurück Eingabe: Polynom p Ausgabe: Leiterterm von p bezüglich grlex Order

-
- `gltSet(F)` berechnet die Leitertme
Eingabe: Liste F von Polynomen
Ausgabe: Liste der Leitertme aller Polynome in F
- `grmd(F,p)` berechnet den Rest bei Division (S. 44)
Eingabe: Liste F von Polynomen und Polynom p
Ausgabe: Polynomrest nach Division von p durch F
- `gSMatrix(F)` Matrix zur Lösung der homogenen linearen Gleichung (S. 46)
Eingabe: Liste F von Polynomen
Ausgabe: Matrix R mit $R \cdot G = 0$, wobei G die Gröbner-Basis zu F ist
- `gSyzPoly(f,g)` berechnet Syzygienpolynom
Eingabe: Polynome f und g
Ausgabe: Syzygienpolynom von f und g
- `gSyzSet(f,g)` berechnet das Syzygienpaar (S. 44)
Eingabe: Polynome f und g
Ausgabe: Vektor h mit $S(f,g) = h^T \cdot (f, -g)$
- `gSyzSets(F)` berechnet das Syzygienset (S. 44)
Eingabe: Liste F von Polynomen
Ausgabe: Liste von Vektoren h
- `HBasis(F)` berechnet die H-Basis (S. 62)
Eingabe: Liste F von Polynomen
Ausgabe: H-Basis H mit $\langle F \rangle = \langle H \rangle$ als Liste von Polynomen
- `hcfs(F,p)` berechnet Koeffizienten zur Darstellung
Eingabe: Liste F von Polynomen und Polynom p
Ausgabe: Liste K mit Koeffizienten, so daß $K \cdot F = p - \text{Rest}$

<code>hcoeffPoly(F,p)</code>	Divisionsalgorithmus für homogene Litterme (S. 64) Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Liste K mit Koeffizienten, so daß $LT(p) = K \cdot F + \text{Rest}$
<code>hDivAlg(F,p)</code>	Divisionsalgorithmus (S. 64) Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Liste K mit Koeffizienten und Rest r , so daß $K \cdot F + r = p$
<code>hleadterm(p)</code>	gibt den Litterterm bei H-Basen zurück Eingabe: Polynom p Ausgabe: homogener Litterterm von p als Polynom
<code>hltSet(F)</code>	berechnet die homogenen Litterme Eingabe: Liste F von Polynomen Ausgabe: Liste der homogenen Litterme aller Polynome in F
<code>hrmd(F,p)</code>	berechnet den Rest bei Division Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Polynomrest nach Division von p durch F
<code>IGBMatrix(F)</code>	berechnet eine Matrix zur Gröbner-Basis-Berechnung (S. 45) Eingabe: Liste F von Polynomen Ausgabe: Matrix M mit $G = F \cdot M$, wobei G die Gröbner-Basis zu F ist
<code>isGBasis(F)</code>	überprüft auf Gröbner-Basis Eingabe: Liste F von Polynomen Ausgabe: <code>true</code> oder <code>false</code>

<code>isHBasis(F)</code>	überprüft auf H-Basis (S. 62) Eingabe: Liste F von Polynomen Ausgabe: <code>true</code> oder <code>false</code>
<code>isZeroDim(F)</code>	überprüft auf Nulldimensionalität (S. 68) Eingabe: Liste F von Polynomen Ausgabe: <code>true</code> oder <code>false</code>
<code>LinSpace(V)</code>	berechnet Vektorraumbasis von Polynomen (S. 54) Eingabe: Liste V von Polynomen als Vektorraumergzeugnis Ausgabe: Vektorraumbasis von V als Liste von Polynomen
<code>mip(a)</code>	erzeugt Polynom aus Multiindex Eingabe: Multiindex a als Liste Ausgabe: Polynom p mit $p = x^a$
<code>MultTable(F,p)</code>	berechnet eine Multiplikationstabelle (S. 68) Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Multiplikationstabelle des Normalformraums von F bezüglich p als Matrix Abhängigkeiten: <code>basis_type</code>
<code>MultTableSet(F,P)</code>	berechnet mehrere Multiplikationstabellen (S. 69) Eingabe: Listen F und P von Polynomen Ausgabe: Liste mit Multiplikationstabellen des Normalformraums von F bezüglich der Polynome in P Abhängigkeiten: <code>basis_type</code>
<code>NForm(F)</code>	berechnet eine Basis des Normalformraums (S. 67) Eingabe: Liste F von Polynomen Ausgabe: Basis des Normalformraumes als Liste von Polynomen Abhängigkeiten: <code>basis_type</code>

<code>pmi(p)</code>	erzeugt Multiindex aus Polynom (S. 58) Eingabe: Polynom p Ausgabe: Multiindex a als Liste mit $p = x^a$ Abhängigkeiten: <code>List_Undets</code>
<code>Pspace(d)</code>	berechnet eine Vektorraumbasis (S. 51) Eingabe: Ganzzahl d Ausgabe: Vektorraumbasis für P_d als Liste von Polynomen Abhängigkeiten: <code>List_Undets</code>
<code>Radical(F)</code>	berechnet das Radikal Eingabe: Liste F von Polynomen Ausgabe: Radikal von F als Liste von Polynomen Abhängigkeiten: <code>basis_type</code>
<code>realSol(F)</code>	berechnet Anzahl der reellen Nullstellen Eingabe: Liste F von Polynomen Ausgabe: Anzahl der reellen Nullstellen Abhängigkeiten: <code>basis_type</code>
<code>reduceSet(F)</code>	reduziert eine Polynommenge (S. 48) Eingabe: Liste F von Polynomen Ausgabe: reduzierte Menge von Polynomen als Liste
<code>scalarproduct(f,g)</code>	berechnet das Skalarprodukt zweier Polynome (S. 59) Eingabe: Polynome f und g Ausgabe: Skalarprodukt von f und g als reelle Zahl Abhängigkeiten: <code>sp_type</code>
<code>setBasis(v)</code>	deklariert eine Basis als Gröbner- oder H-Basis (S. 129) Eingabe: Boolescher Ausdruck v (<code>true</code> oder <code>false</code>) Ausgabe: — (intern wird <code>is_basis</code> gesetzt)

<code>setBasisType(S)</code>	legt die Basis zur Berechnung fest (S. 67) Eingabe: Zeichenkette $S = "G"$ oder $S = "H"$ Ausgabe: — (intern wird <code>basis_type</code> gesetzt)
<code>setConsts(C)</code>	legt die Konstanten vor einer Berechnung fest (S. 42) Eingabe: Liste C mit Konstanten Ausgabe: — (intern wird <code>List_Consts</code> gesetzt)
<code>setSP(S)</code>	legt das zu verwendende Skalarprodukt fest (S. 57) Eingabe: Zeichenkette $S = "M"$ oder $S = "D"$ Ausgabe: — (intern wird <code>sp_type</code> gesetzt)
<code>setUndets(U)</code>	legt die Unbekannten vor einer Berechnung fest (S. 42) Eingabe: Liste U mit Unbekannten Ausgabe: — (intern wird <code>List_Undets</code> gesetzt)
<code>shorten(p)</code>	kürzt ein Polynom auf $ggT = 1$ Eingabe: Polynom p Ausgabe: gekürztes Polynom
<code>shortSet(F)</code>	kürzt jedes Polynom in einer Menge Eingabe: Liste F von Polynomen Ausgabe: Liste von gekürzten Polynomen
<code>solveH(F)</code>	berechnet die Nullstellen (S. 72) Eingabe: Liste F von Polynomen Ausgabe: Liste aller Lösungen X von $F(X) = 0$ Abhängigkeiten: <code>basis_type</code> , <code>List_Undets</code>
<code>traceMatrix(F,p)</code>	berechnet die Spurmatrix Eingabe: Liste F von Polynomen und Polynom p Ausgabe: Spurmatrix bezüglich Normalformraumbasis von F und p Abhängigkeiten: <code>basis_type</code>

<code>TraceMatrix(F)</code>	berechnet Spurmatrizen Eingabe: Liste F von Polynomen Ausgabe: Liste von Spurmatrizen Abhängigkeiten: <code>basis_type</code>
<code>Vspace(d,F)</code>	berechnet Vektorraumbasis (S. 53) Eingabe: Ganzzahl d und Liste F von Polynomen Ausgabe: Vektorraumbasis von $V_d(F)$ als Liste von Polynomen
<code>Wspace(d,F)</code>	berechnet Vektorraumbasis (S. 60) Eingabe: Ganzzahl d und Liste F von Polynomen Ausgabe: Vektorraumbasis von $W_d(F)$ als Liste von Polynomen

Literatur

- [1] Baumann, B.: *Algebra A* (Vorlesungsskript), (Justus-Liebig-Universität Gießen, 2001), http://www.uni-giessen.de/AG_Algebra/ALGA-WS2001.shtml
- [2] Becker, T., Weispfennig, V. (in Cooperation with H. Kredel): *Gröbner Bases - A Computational Approach to Commutative Algebra* (Second Edition), (Springer, 1998)
- [3] Buchberger, B.: Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, in: *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems*, ed. N.K.Bose (D.Reidel, 1985), Seite 184-232
- [4] Cohen, A.M., Cuypers, H., Sterk, H. (Eds.): *Some Tapas of Computer Algebra*, (Springer, 1999)
- [5] Cox, D., Little, J., O'Shea, D.: *Ideal, Varieties and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra* (Second Edition), (Springer, 1997)
- [6] Kowalsky, H.-J., Michler G.O.: *Lineare Algebra* (11. Auflage), (de Gruyter, 1998)
- [7] May, J.P.: *Solutions of Multivariate Polynomial Systems*, (North Carolina State University, 2000), <http://www4.ncsu.edu/~jpmay/>
- [8] Möller, H.M., Sauer, T.: H-bases for polynomial interpolation and system solving, in: *Advances in Computational Mathematics 12* (Baltzer AG, 2000)
- [9] Möller, H.M., Stetter, H.J.: Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems, in: *Numer. Math. 70* (Springer, 1995), Seite 311-329
- [10] Peña, J.M., Sauer, T.: Linear Algebra methods for efficient polynomial reduction, in: *Advances Comput. Math.* (to appear)
- [11] Sauer, T.: Gröbner bases, H-bases and interpolation, in: *Trans. Amer. Math. Soc. 353* (2000), Seite 2293-2308
- [12] Sauer, T.: *Computeralgebra* (Vorlesungsskript), (Justus-Liebig-Universität Gießen, 2003), <http://www.uni-giessen.de/tomas.sauer/>

Erklärung

Ich versichere, daß ich die vorgelegte Arbeit ohne Nutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Gleichwohl versichere ich, ddaß die Arbeit in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat oder von dieser als Teil einer Prüfungsleistung angenommen wurde.

Gießen, den 2. August 2005