

Ernst-Moritz-Arndt-Universität Greifswald
Fachrichtung Mathematik/Informatik

Algorithmische Berechnung von ersten Integralen und speziellen Lösungen bei Differentialgleichungen

Wissenschaftliche Arbeit zur Erlangung
des akademischen Grades

“Diplom-Mathematiker“

an der Fachrichtung Mathematik/Informatik
der Ernst-Moritz-Arndt-Universität Greifswald

eingereicht von
Holger Irrgang

im Auftrag von
Prof. Dr. habil. G. Czichowski

Greifswald, 12. Januar 1996

Haftungsausschluß:

Es wird keinerlei Haftung übernommen für irgendwelche Schäden, die aus der Nutzung der in dieser Arbeit beschriebenen Programme folgen.

Diese Diplomarbeit wurde mit dem Textsatzsystem \TeX erstellt.

Inhaltsverzeichnis

1. Einleitung	3
2. Elementare Erste Integrale - Algorithmische Betrachtungen	5
2.1. Problemstellung und einführendes Beispiel	5
2.2. Umformulierungen der Problemstellung	6
2.3. Algorithmus	7
2.4. Differentialkörper und elementare Erweiterungen	9
2.5. Grundlagen des Algorithmus	11
2.5.1. Die Resultate von Darboux	20
2.6. Das Problem $f \mid Df$	22
3. Spezielle Lösungen bei Systemen von linearen partiellen Differentialgleichungen	28
3.1. Motivation und Vorbereitung	28
3.2. Reduktion der Problemstellung	29
3.3. Standardformen	30
3.4. Algorithmische Behandlung	33
3.5. Algorithmische Berechnung von rationalen Lösungen linearer gewöhnlicher Differentialgleichungen	36
3.5.1. Problemvorstellung und vorbereitende Aussagen	36
3.5.2. Der Algorithmus	39
3.5.3. Rechnungen und Erklärungen zum Algorithmus	42
3.5.4. Ein Beispiel	44
A. Vorstellung der Programme	48

1. Einleitung

Bei der Behandlung von Differentialgleichungen ist man häufig nicht in der Lage, (mit den bekannten Methoden) die gesamte Lösungsmenge bzw. eine brauchbare Darstellung der Lösung in geschlossener Form anzugeben. Es erweist sich nun als sinnvoll, gezielt nach speziellen Lösungen bzw. nach speziellen „ersten Integralen“ zu suchen. Mit „ersten Integralen“ bezeichnet man Funktionen in den Variablen und Ableitungen, die auf den Lösungen einer Differentialgleichung konstant sind. Der Vorteil bei diesem Vorgehen liegt darin, daß man durch die Einschränkung der Form der gesuchten Funktionen von zusätzlichen Bedingungen ausgehen kann, die bei der Lösung der Differentialgleichung weiterhelfen könnten. Außerdem ist es sehr von Nutzen, wenn das angewandte Verfahren algorithmisch ist. Diese beiden Aspekte wurden in der vorliegenden Arbeit hauptsächlich berücksichtigt, wobei rationale Lösungen und elementare erste Integrale im Mittelpunkt stehen. Betrachtet werden zwei verschiedene Arten von Differentialgleichungssystemen. Zum einen sind dies Systeme der Form

$$\frac{dx}{dt} = P \quad \frac{dy}{dt} = Q,$$

wobei P und Q Polynome in x und y sind. Von **M. J. Prella** und **M. F. Singer** ist in der Arbeit [1] ein Algorithmus dargestellt, der Ergebnisse von **Darboux**, die in [5] aufgeführt sind, benutzt und mit dem sich elementare erste Integrale des obigen Systems finden lassen. Bei der Entwicklung des Algorithmus stößt man auf das folgende Problem: Man bestimme zu dem Differentialoperator $D = P\partial_x + Q\partial_y$ alle irreduziblen Polynome f mit der Eigenschaft $f \mid Df$. Als Lösungsvarianten dieses Problems werden zwei Möglichkeiten aufgezeigt. Allerdings geht man in beiden von einer vorgegebenen Schranke für den „Totalgrad“ der Polynome f aus. Zur Berechnung einer solchen Schranke ist bisher kein Verfahren bekannt. Somit startet man den Algorithmus mit einer heuristischen Vorgabe. Aus den gefundenen Polynomen wird dann zunächst versucht, ein rationales erstes Integral zu konstruieren. Ist das nicht möglich, so wird ein elementares erstes Integral der Form $w = w_0 + \sum c_i \ln w_i$ gebildet, wobei die w_i algebraisch über $\mathbb{C}(x, y)$ und die c_i komplexe Zahlen sind.

Für beide Möglichkeiten zum Finden der irreduziblen Polynome f mit $f \mid Df$ sind in REDUCE zwei Programme erstellt worden. Der Quelltext mit einigen

Erläuterungen ist im Anhang aufgeführt. Aus den Beispielrechnungen mit diesen Programmen wurde folgendes ersichtlich: Für einige spezielle Formen der Koeffizientenpolynome P und Q wurden bereits bei einer niedrigen vorgegebenen Schranke irreduzible Polynome mit der obigen Eigenschaft gefunden. Andererseits traten aber auch bei konkreten P und Q mit kleinem „Totalgrad“ enorme Rechenzeiten auf, zudem wurde viel Speicherkapazität benötigt. Möglicherweise entstehen diese Probleme dadurch, daß sehr große algebraische Gleichungssysteme zu lösen sind.

Als zweite Art von Differentialgleichungssystemen werden lineare partielle Differentialgleichungssysteme mit endlich-dimensionalem Lösungsraum betrachtet. Dabei interessieren wir uns für rationale Lösungen. Durch diese starken Einschränkungen wird das Vorgehen in allen Punkten algorithmisch ablaufen. Wichtige Hilfsmittel bei dem hier dargestellten Algorithmus sind „Termordnungen“, der „Standardform-Algorithmus“ [9, 10] sowie die algorithmische Suche von rationalen Lösungen bei linearen gewöhnlichen Differentialgleichungen. Der „Standardform-Algorithmus“ überführt das Ausgangsdifferentialgleichungssystem in ein äquivalentes Differentialgleichungssystem, das bei endlich-dimensionalem Lösungsraum eine gewöhnliche Differentialgleichung enthält. Für die Suche nach rationalen Lösungen bei linearen gewöhnlichen Differentialgleichungen existiert bereits ein Algorithmus, der allerdings den Spezialfall mit parameterfreien Koeffizienten behandelt und in [8] dargestellt ist. Wir werden sehen, daß sich dieser Algorithmus auf den Fall mit parameterabhängigen Koeffizienten verallgemeinern läßt.

Wiederholtes Lösen gewöhnlicher Differentialgleichungen führt zu neuen linearen partiellen Differentialgleichungssystemen, die ebenfalls einen endlich-dimensionalen Lösungsraum besitzen. Der Algorithmus wird rekursiv auf die entstehenden Systeme angewandt, bis die Lösungen der gewöhnlichen Differentialgleichungen nur noch konstante Parameter enthalten.

Bei der algorithmischen Suche nach elementaren ersten Integralen bzw. nach rationalen Lösungen stellt sich demnach folgendes heraus. Die zunächst relativ einfach erscheinenden Systeme im ersten Teil der Arbeit lassen keine vollständig algorithmische Bearbeitung zu. Bei den im zweiten Teil im allgemeinen größeren und komplizierteren Systemen kann dagegen vollkommen algorithmisch vorgegangen werden. Entscheidend dabei sind die wesentlichen Eigenschaften Linearität der Differentialgleichungen und endliche Dimension des Lösungsraumes.

Die in der vorliegenden Arbeit verwendeten algebraischen Verfahren wie Partialbruchzerlegung oder Division mit Rest in mehreren Variablen sind in einer Reihe von Büchern beschrieben und sämtlich konstruktiv.

2. Elementare Erste Integrale - Algorithmische Betrachtungen

2.1. Problemstellung und einführendes Beispiel

Oft ist es nicht möglich oder nicht effektiv, die Lösung eines Systems von Differentialgleichungen explizit mit elementaren Funktionen darzustellen. In vielen Fällen lohnt es sich, nach elementaren Funktionen zu suchen, die konstant auf allen Lösungen des Systems sind. Solche Funktionen heißen „elementare erste Integrale“. Ein Beispiel soll dies verdeutlichen.

Beispiel: Räuber-Beute-Gleichungen

$$\frac{dx}{dt} = ax - bxy \quad \frac{dy}{dt} = -cy + dxy \quad a, b, c, d \in \mathbb{R}^+ \quad (2.1)$$

Die Lösung der Räuber-Beute-Gleichungen läßt sich nicht explizit in elementaren Funktionen schreiben. Man kann jedoch zeigen, daß die Funktion

$$F(x, y) = dx + by - c \ln x - a \ln y$$

konstant auf allen Lösungskurven $(x(t), y(t))$ und damit ein erstes Integral von (2.1) ist. Unter Benutzung von F ist es nun möglich zu beweisen, daß alle Lösungskurven im positiven Quadranten ($x > 0, y > 0$) geschlossen sind, d.h. diese Lösungen sind periodisch.

Das erste Integral F könnte man allgemeiner in der Form

$$F(x, y) = w_0 + \sum_{i=1}^2 c_i \ln w_i$$

schreiben, wobei w_0, w_1 und w_2 algebraische (hier rationale) Funktionen in x und y sind.

Wir wollen in diesem Teil der Arbeit folgendes zeigen: Wenn ein System von Differentialgleichungen ein elementares erstes Integral hat, dann besitzt es auch eines von obiger spezieller Form. Darauf aufbauend wird ein Algorithmus angegeben, der, ausgehend von einem Verfahren von **M. J. Prelle** und **M. F. Singer**, elementare erste Integrale findet. Ein Nachteil ist, daß dieser Algorithmus mit einer heuristischen Vorgabe arbeitet. Er liefert aber trotzdem in vielen Fällen ein Ergebnis.

2.2. Umformulierungen der Problemstellung

Es werden Systeme von Differentialgleichungen der Form

$$\frac{dx}{dt} = P \quad \frac{dy}{dt} = Q \quad (2.2)$$

betrachtet. P und Q sind Polynome in x und y mit komplexen Koeffizienten: $\mathbb{C}[x, y]$.

Definition 2.1 Sei $f = f(x, y)$ eine Funktion, die auf allen Lösungen von (2.2) konstant ist, d.h.

$$\begin{aligned} \frac{df}{dt} &= \partial_x f \frac{dx}{dt} + \partial_y f \frac{dy}{dt} = 0 \\ \text{und somit} \quad P\partial_x f + Q\partial_y f &= 0 \quad \text{in jedem Punkt } (x, y). \end{aligned} \quad (2.3)$$

f heißt dann ein **erstes Integral** des Systems (2.2). Ist f außerdem eine elementare Funktion, so heißt f ein **elementares erstes Integral**.

Unser Ziel ist es, solche elementaren ersten Integrale von Systemen der obigen Form zu finden. Sehen wir uns nun die Gleichung (2.3) aus der Definition an. Man kann sie mit dem linearen partiellen Differentialoperator

$$D := P\partial_x + Q\partial_y \quad (2.4)$$

in der Form $D(f) = 0$ schreiben. Diese Gleichung ist jetzt so zu interpretieren: Wir suchen nach elementaren Funktionen f , die im Kern des Operators D liegen. Umgekehrt gilt: Jede elementare Funktion aus dem Kern von D ist ein elementares erstes Integral des Systems (2.2).

Nun sehen wir uns die charakteristische Gleichung von (2.2) an:

$$\frac{dx}{P} = \frac{dy}{Q}.$$

Daraus erhält man folgende gewöhnliche Differentialgleichung erster Ordnung

$$\frac{dy}{dx} = \frac{Q}{P}.$$

Ein erstes Integral dieser Differentialgleichung ist eine Funktion $f(x, y)$, die längs aller Lösungskurven $(x, y(x))$ konstant ist, d.h.

$$\begin{aligned} \frac{df}{dx} &= \frac{\partial f}{\partial y} \frac{dy}{dx} + \frac{\partial f}{\partial x} = 0 \\ \text{und demnach} \quad \frac{\partial f}{\partial y} \frac{Q}{P} + \frac{\partial f}{\partial x} &= 0 \quad \text{in jedem Punkt } (x, y). \end{aligned}$$

Die gleiche Argumentation wie für das System (2.2) liefert dann, daß sich die ersten Integrale der obigen gewöhnlichen Differentialgleichung im Kern von D befinden und auch in diesem Fall gilt die Umkehrung.

Im Fall einer gewöhnlichen Differentialgleichung erster Ordnung stellt ein erstes Integral eine implizit geschriebene Lösung dar.

2.3. Algorithmus

Die hier verwendete Theorie ist in der Arbeit [1] dargestellt, in der ausgehend von der Existenz eines elementaren ersten Integrals auf die Existenz eines ersten Integrals von „einfacher Form“ geschlossen wird.

In diesem Zusammenhang kann man einen Algorithmus zur Bestimmung von elementaren ersten Integralen des Systems (2.2) angeben. In den anschließenden Abschnitten wird dieser Algorithmus erklärt. Wir werden dann auch feststellen, daß das Grundproblem nur ein heuristisches Herangehen zuläßt.

Es wurde bereits gezeigt, daß das Problem der Suche nach elementaren Lösungen von

$$Dw = 0 \quad \text{mit} \quad D = P\partial_x + Q\partial_y$$

äquivalent zu dem Problem ist, elementare erste Integrale von (2.2) zu finden. Wir werden in unseren Betrachtungen hauptsächlich diesen linearen partiellen Differentialoperator D benutzen.

Zur Vorbereitung des Algorithmus wird folgendes definiert.

Definition 2.2 Sei f ein Polynom in x und y . Wir bezeichnen den Grad von f bzgl. x mit \deg_x und analog den Grad bzgl. y mit \deg_y . Ein Ausdruck der Form $x^i y^j$, $i, j \in \mathbb{N}$ heißt ein **Monom**. Der **Totalgrad** eines Monoms M ist:

$$\text{tdeg } M = \deg_x M + \deg_y M.$$

Der Totalgrad von f ist dann definiert durch:

$$\text{tdeg } f = \max\{\text{tdeg } M \mid M \text{ ist Monom in } f\}.$$

Sei nun ein $N \in \mathbb{N}$ gegeben, das eine Schranke für den Totalgrad von Polynomen, aus denen die ersten Integrale konstruiert werden, darstellt. Der **Input** des Algorithmus besteht aus dieser natürlichen Zahl N und zwei Polynomen P und Q . Der **Output** ist eine Menge von elementaren ersten Integralen für das System (2.2).

1. Schritt Man suche alle Polynome f mit folgenden Eigenschaften:

1. $f \mid Df$, wobei $D = P\partial_x + Q\partial_y$
2. f ist irreduzibel und
3. $\text{tdeg } f \leq N$.
(Wie dies geschieht, wird in einem späteren Abschnitt erläutert.)

Die gefundenen Polynome seien f_1, \dots, f_k . Hat man nur Konstanten als Lösungen erhalten, so gibt es keine nichttrivialen elementaren ersten Integrale, die sich aus irreduziblen Polynomen mit den obigen Eigenschaften konstruieren lassen.

2. Schritt Nun werden die Quotienten $Df_i/f_i, 1 \leq i \leq k$ berechnet. Danach überprüft man, ob sie linear unabhängig über \mathbb{Z} sind. Ist das nicht der Fall, so ist eine Relation

$$\sum_{i=1}^k n_i \frac{Df_i}{f_i} = 0, \quad (2.5)$$

$n_i \in \mathbb{Z}, 1 \leq i \leq k$, wobei nicht alle n_i gleich Null sind, erfüllt. Man ermittle die n_1, \dots, n_k . Die Funktion

$$w = \prod_{i=1}^k f_i^{n_i}$$

ist dann eine elementare, sogar eine rationale Lösung von $Dw = 0$.

3. Schritt Sind die $Df_i/f_i, 1 \leq i \leq k$ linear unabhängig über \mathbb{Z} , so verfahren wir wie folgt: Man suche nach $r_i \in \mathbb{Q}, 1 \leq i \leq k$, mit

$$\sum_{i=1}^k r_i \frac{Df_i}{f_i} = -(\partial_x P + \partial_y Q). \quad (2.6)$$

Man bilde jetzt $S = \prod_{i=1}^k f_i^{r_i}$ und bestimme Funktionen w_0, \dots, w_m , die algebraisch über $\mathbb{C}(x, y)$ sind, und Konstanten $c_1, \dots, c_m \in \mathbb{C}$, so daß gilt:

$$\begin{aligned} \partial_x(w) &= SQ \quad \text{und} \quad \partial_y(w) = -SP, \\ \text{wobei } w &= w_0 + \sum_{i=1}^m c_i \ln w_i. \end{aligned}$$

Hat man $w_0, w_1, \dots, w_m, c_1, \dots, c_m$ gefunden, die die obigen Bedingungen erfüllen, so ist das aus ihnen gebildete w ein elementares erstes Integral von (2.2).

Es bleibt natürlich noch zu klären, woraus sich die n_i, r_i und w_i ergeben. Für die ganzen Zahlen n_i und für die rationalen Zahlen r_i werden wir das Vorgehen im Anschluß beschreiben. Wie man w_0, w_1, \dots, w_m und die Konstanten c_1, \dots, c_m bekommt, wird in einem späteren Abschnitt dargestellt.

Zur Bestimmung der n_i und r_i werden nun zunächst die Ausdrücke $Df_i/f_i, 1 \leq i \leq k$, und $-(\partial_x P + \partial_y Q)$ berechnet. Dann überführt man die Gleichungen (2.5) bzw. (2.6) in eine ganzrationale Form. Koeffizientenvergleich nach verschiedenen Monomen liefert jeweils ein lineares Gleichungssystem, dessen Lösung die gesuchten n_i bzw. r_i sind.

2.4. Differentialkörper und elementare Erweiterungen

Der Zugang zu elementaren ersten Integralen wird imfolgenden durch die Betrachtung von „Differentialkörpern“ und ihren „Erweiterungen“ ermöglicht.

Definition 2.3 *Ein Differentialkörper F ist ein Körper K der Charakteristik Null, ausgestattet mit einer Menge $\Delta = \{\partial_1, \dots, \partial_k\}$ von Differentiationen, die sämtlich den üblichen Regeln der Differentiation von Summen und Produkten genügen, d.h. für alle $\partial_i \in \Delta$ gilt:*

- (A) $\partial_i(f + g) = \partial_i(f) + \partial_i(g)$ und
- (B) $\partial_i(fg) = \partial_i(f)g + f\partial_i(g)$ mit $f, g \in K$.

Besteht Δ aus nur einer Differentiation, so nennt man F einen **gewöhnlichen Differentialkörper**.

Beispiel: Ist der Grundkörper K ein Körper von Funktionen in den Variablen x und y , dann können die Differentiationen die partiellen Ableitungen nach x und y sein.

Definition 2.4 Gegeben sei ein Differentialkörper $F = (K, \Delta)$. Ein $c \in K$ heißt **Konstante in F** , wenn für alle $\partial \in \Delta$ gilt: $\partial c = 0$.

Die Konstanten eines Differentialkörpers besitzen die folgende Eigenschaft, die man durch Anwendung der Regeln (A) und (B) leicht nachrechnet.

Satz 2.5 Sei $F = (K, \Delta)$ ein Differentialkörper. Die Menge der Konstanten in F bildet einen Differentialunterkörper $C(K, \Delta)$ von F , den Konstantenkörper.

Beispiel: Die Menge der rationalen Funktionen $\mathbb{C}(x, y)$ mit den bekannten partiellen Ableitungen ∂_x und ∂_y bilden einen Differentialkörper $F = (\mathbb{C}(x, y), \{\partial_x, \partial_y\})$. Der Konstantenkörper $C(\mathbb{C}(x, y), \{\partial_x, \partial_y\})$ ist in diesem Falle natürlich \mathbb{C} .

Definition 2.6 Sei F ein Differentialkörper. Eine **Differentialerweiterung** L von F ist ein Erweiterungskörper von F mit Differentiationen, die die Differentiationen von F fortsetzen.

Eine **einfache elementare Erweiterung** eines Differentialkörpers F ist ein Differentialkörper $F(\theta)$, welcher eine Erweiterung von F so ist, daß θ eine der folgenden Bedingungen erfüllt, wobei $'$ eine beliebige Differentiation von F darstellt.

- 1) θ ist algebraisch über F .
- 2) Es existiert ein $f \in F$, für das gilt: $\theta' = \frac{f'}{f}$. Das Element θ heißt dann ein **Logarithmus** über F .
- 3) Es existiert ein $f \in F$, für das gilt: $\theta' = f'\theta$. Das Element θ heißt dann ein **Exponential** über F .

In den Fällen 2) und 3) heißt $F(\theta)$ eine **transzendente elementare Erweiterung** von F . Eine endliche Anzahl von einfachen elementaren Erweiterungen nennt man eine **Elementarerweiterung** von F .

Bemerkung: Elementare Funktionen im üblichen Sinn bzw. elementare erste Integrale sind jetzt Funktionen, die in Elementarerweiterungen von Differentialkörpern liegen.

Beispiel: Differentialkörper: $F = (\mathbb{Q}(x), \{\partial_x\})$

algebraische Erweiterung: $F_1 = (\mathbb{Q}(x, \sqrt{x}), \{\partial_x\})$ mit $\partial_x \sqrt{x} = 1/(2\sqrt{x})$

Logarithmus-Erweiterung: $F_2 = (\mathbb{Q}(x, \ln x), \{\partial_x\})$ mit $\partial_x \ln x = 1/x$

Exponential-Erweiterung: $F_3 = (\mathbb{Q}(x, e^x), \{\partial_x\})$ mit $\partial_x e^x = e^x$

Mit den definierten algebraischen Begriffen sind wir nun in der Lage, eine Aussage über die Existenz eines elementaren ersten Integrals von (2.2) zu treffen.

Satz 2.7 *Es bezeichne (K, Δ) den Differentialkörper $(\mathbb{C}(x, y), \{\partial_x, \partial_y\})$ und sei (L, Δ) eine elementare Erweiterung von (K, Δ) , für die gilt: $C(K, \Delta) = C(L, \Delta)$. Sei D der lineare Differentialoperator aus (2.4). Das System (2.2) hat ein nicht-triviales elementares erstes Integral genau dann, wenn $C(L, \Delta)$ eine echte Teilmenge von $C(L, \{D\})$ ist.*

Beweis: Sei $f \notin \mathbb{C}$ und elementares erstes Integral von (2.2).

$$\Rightarrow \partial_x f \neq 0 \text{ oder } \partial_y f \neq 0 \text{ und } Df = P\partial_x f + Q\partial_y f = 0$$

Sei $L = K(f)$. Dann gilt $C(L, \Delta) = C(K, \Delta)$ und $f \notin C(L, \Delta)$, aber $f \in C(L, \{D\})$.

Daraus folgt $C(L, \Delta) \subset C(L, \{D\})$, weil offensichtlich für alle $c \in C(L, \Delta)$ gilt: $c \in C(L, \{D\})$.

Nun sei $C(L, \Delta) \subset C(L, \{D\})$, wobei (L, Δ) eine elementare Erweiterung von (K, Δ) ist. Daraus folgt: Es existiert ein $f \in L$ mit

$$\partial_x f \neq 0 \text{ oder } \partial_y f \neq 0 \text{ und } Df = P\partial_x f + Q\partial_y f = 0.$$

Dies ist gerade die Definition eines nichttrivialen elementaren ersten Integrals des Systems (2.2).

□

2.5. Grundlagen des Algorithmus

Seien das System (2.2) und der lineare Operator D aus (2.4) gegeben. In diesem Abschnitt stellen wir uns nun die Fragen: Wie gelangt man von der Suche nach elementaren ersten Integralen für das obige System zu dem Problem, irreduzible Polynome f zu finden, die $f \mid Df$ erfüllen? Wie entsteht daraus der angegebene Algorithmus?

Dazu werden zu Beginn einige Aussagen angegeben, die aus [1] entnommen sind und etwas über die Existenz von ersten Integralen einer bestimmten Form aussagen. Folgende Bezeichnungen werden beibehalten:

$$K = \mathbb{C}(x, y) \quad \text{und} \quad \Delta = \{\partial_x, \partial_y\}.$$

Satz 2.8 Sei (L, Δ) eine elementare Erweiterung des Differentialkörpers (K, Δ) und seien die zugehörigen Konstantenkörper gleich: $C(L, \Delta) = C(K, \Delta)$. Sei D der Differentialoperator aus (2.4) und sei vorausgesetzt, daß $C(L, \Delta)$ eine echte Teilmenge von $C(L, \{D\})$ ist. Dann gibt es in L Elemente w_0, w_1, \dots, w_m , die algebraisch über K sind, und c_1, c_2, \dots, c_m in $C(K, \Delta)$, so daß

$$Dw_0 + \sum_{i=1}^m c_i \frac{Dw_i}{w_i} = 0 \quad \text{und}$$

$$\partial w_0 + \sum_{i=1}^m c_i \frac{\partial w_i}{w_i} \neq 0 \quad \text{für ein } \partial \in \Delta.$$

Der Beweis des Satzes ergibt sich aus dem folgenden Lemma, das hier ohne Beweis angegeben wird. Er ist in [1] zu finden.

Lemma 2.9 Seien (K, Δ) , (L, Δ) und D wie im Satz 2.8. Wir setzen voraus, daß u_0, u_1, \dots, u_m in L und d_1, d_2, \dots, d_m in $C(L, \Delta)$ existieren, so daß

$$Du_0 + \sum_{i=1}^m d_i \frac{Du_i}{u_i} = 0 \quad \text{und}$$

$$\partial u_0 + \sum_{i=1}^m d_i \frac{\partial u_i}{u_i} \neq 0 \quad \text{für ein } \partial \in \Delta.$$

Dann gibt es in L Elemente w_0, w_1, \dots, w_m , die algebraisch über K sind, und $c_1, c_2, \dots, c_m \in C(K, \Delta)$, so daß

$$Dw_0 + \sum_{i=1}^m c_i \frac{Dw_i}{w_i} = 0 \quad \text{und}$$

$$\partial w_0 + \sum_{i=1}^m d_i \frac{\partial w_i}{w_i} \neq 0 \quad \text{für ein } \partial \in \Delta.$$

Zum **Beweis** des Satzes: Die Voraussetzungen des Lemmas ergeben sich aus denen des Satzes wie folgt. Seien (K, Δ) , (L, Δ) und D wie im Satz gegeben und sei $C(L, \Delta)$ echte Teilmenge von $C(L, \{D\})$. Letzteres bedeutet, es existiert ein $v \in L$ mit $Dv = 0$ und $\partial v \neq 0$ für ein $\partial \in \Delta$. Wir setzen nun $u_0 = v$ und $d_1 = d_2 = \dots = d_m = 0$. Die Voraussetzung des Lemmas ist damit erfüllt. Aus der Aussage des Lemmas folgt dann die Aussage des Satzes.

□

Den Satz 2.8 nutzen wir nun zum Beweis der folgenden Aussagen.

Satz 2.10 *Haben die Gleichungen (2.2) ein elementares erstes Integral, dann existiert ein über K algebraisches Element $R \neq 0$, so daß*

$$DR = -(\partial_x P + \partial_y Q)R. \quad (2.7)$$

Beweis: Anwendung des Satzes 2.8 ergibt: Es existieren w_0, \dots, w_m , die algebraisch über K sind, und $c_1, \dots, c_m \in \mathbb{C}$, so daß $Dw_0 + \sum_{i=1}^m c_i \frac{Dw_i}{w_i} = 0$ und $\partial w_0 + \sum_{i=1}^m d_i \frac{\partial w_i}{w_i} \neq 0$ für ein $\partial \in \Delta$. Sei

$$\begin{aligned} R_1 &= \partial_x w_0 + \sum_{i=1}^m d_i \frac{\partial_x w_i}{w_i}, & R_2 &= \partial_y w_0 + \sum_{i=1}^m d_i \frac{\partial_y w_i}{w_i}. \\ & \implies PR_1 + QR_2 = 0 \end{aligned}$$

Wir können o.B.d.A. Q ungleich Null voraussetzen. Sei nun $R := R_1/Q$. Daraus folgen $R_1 = QR$ und $R_2 = -PR$. Da ∂_x und ∂_y in jeder algebraischen Erweiterung von K vertauschbar sind, folgt $\partial_y(QR) = \partial_x(-PR)$. Nach Ausführung der Differentiationen bekommt man die Aussage des Satzes.

□

Bemerkung: Für die sich anschließenden Erläuterungen wird eigentlich die negierte Aussage von Satz 2.10 benötigt: Existiert kein über K algebraisches Element $R \neq 0$, das (2.7) erfüllt, dann besitzt das System (2.2) kein elementares erstes Integral.

Eine Art Umkehrung des obigen Satzes ist die folgende Aussage.

Satz 2.11 *Es existiere ein Element $S \neq 0$, das algebraisch über K ist und (2.7) erfüllt. Dann gilt eine der beiden Aussagen:*

- (i) *Es gibt ein $w \in K$, so daß $Dw = 0$ und $\partial w \neq 0$ für ein $\partial \in \Delta$.*
- (ii) *Für jedes $R \neq 0$, das algebraisch über K ist und (2.7) erfüllt, gibt es ein $c \in \mathbb{C}$, so daß $R = cS$, und ein $l \in \mathbb{Z}$, so daß $R^l \in K$.*

Gilt (i), dann hat das System (2.2) bereits nach Definition ein elementares erstes Integral. Gilt (ii), so hat das System (2.2) ein elementares erstes Integral genau dann, wenn w_0, \dots, w_m , die wiederum algebraisch über K sind, und $c_1, \dots, c_m \in \mathbb{C}$ existieren, so daß

$$\partial_x w_0 + \sum_{i=1}^m c_i \frac{\partial_x w_i}{w_i} = SQ \quad \partial_y w_0 + \sum_{i=1}^m c_i \frac{\partial_y w_i}{w_i} = -SP.$$

Zum Beweis des Satzes benötigen wir das folgende Lemma.

Lemma 2.12 Sei $K = \mathbb{C}(x, y)$ und seien R und S zwei über K algebraische Elemente ungleich Null. Es gelte: Aus $Dw = 0$ und $w \in K$ folgt, daß w eine Konstante ist. Ist dann $D(R/S) = 0$, so ist R ein konstantes Vielfaches von S .

Beweis: Sei $T := R/S$. Wir nehmen an, daß T algebraisch über K ist. Dann existiert ein eindeutig bestimmtes Minimalpolynom für T , so daß

$$T^n + \sum_{i=0}^{n-1} t_i T^i = 0,$$

mit $t_i \in K$, $i = 0, \dots, n-1$, erfüllt ist. Auf diese Gleichung wenden wir den Operator D an. Daraus folgt:

$$nT^{n-1}D(T) + \sum_{i=1}^{n-1} (D(t_i)T^i + t_i i T^{i-1} D(T)) = 0.$$

Da $D(T) = 0$ gilt, erhält man

$$\sum_{i=0}^{n-1} D(t_i)T^i = 0.$$

Gilt für ein $i > 0$ $D(t_i) \neq 0$, so steht diese algebraische Gleichung im Widerspruch zur Eindeutigkeit des Minimalpolynoms. Daraus folgt: $T \in K$ und damit nach Voraussetzung $T \in \mathbb{C}$. Also existiert ein $c \in \mathbb{C}$ mit $R = cS$.

Im Fall $D(t_i) = 0$ für alle i , folgt, daß T im algebraischen Abschluß von \mathbb{C} liegt, woraus wieder $T \in \mathbb{C}$ folgt.

□

Beweis des Satzes 2.11: Sei $R \neq 0$ algebraisch über K und erfülle (2.7). Außerdem setzen wir voraus, daß (i) nicht gilt, d.h. aus $Dw = 0$ für $w \in K$ folgt $w \in \mathbb{C}$. R/S erfüllt

$$\begin{aligned} D(R/S) &= \frac{D(R)S - RD(S)}{S^2} = \\ &= \frac{-(\partial_x P + \partial_y Q)RS + R(\partial_x P + \partial_y Q)S}{S^2} = 0. \end{aligned}$$

Aus dem Lemma 2.12 folgt $R = cS$ für ein $c \in \mathbb{C}$.

Sei nun E eine algebraische Erweiterung von K , die R enthält. Für jeden K -Automorphismus σ von E bekommen wir

$$\frac{D(\sigma R)}{\sigma R} = -(\partial_x P + \partial_y Q),$$

da die Anwendung von D mit der Anwendung eines Automorphismus vertauscht werden kann. Summieren wir diese Gleichung über alle σ aus der Galois-Gruppe G von E über K , so erhalten wir:

$$\frac{D(\text{Norm}(R))}{\text{Norm}(R)} = -l(\partial_x P + \partial_y Q) \quad \text{für ein } l \in \mathbb{Z},$$

dabei ist $\text{Norm}(z) := \prod_{\sigma \in G} \sigma(z)$. Aus $D(R^l) = lD(R)R^{l-1}$ folgt:

$$D\left(\frac{\text{Norm}(R)}{R^l}\right) = 0.$$

Es folgt wieder aus dem Lemma 2.12, daß R^n ein konstantes Vielfaches von $\text{Norm}(R)$ und damit ein Element von K ist.

Nun sei vorausgesetzt, daß (2.2) ein elementares erstes Integral hat. Im Beweis des vorigen Satzes wurde bereits gezeigt: $\exists w_0, \dots, w_m$ und R , die algebraisch über K sind, und $c_1, \dots, c_m \in \mathbb{C}$, so daß

$$\partial_x w_0 + \sum_{i=1}^m c_i \frac{\partial_x w_i}{w_i} = RQ, \quad \partial_y w_0 + \sum_{i=1}^m c_i \frac{\partial_y w_i}{w_i} = -RP$$

und $DR = -(\partial_x P + \partial_y Q)R$. Gilt (i) nun nicht, so haben wir durch (ii): $R = cS$ für ein $c \in \mathbb{C}$, so daß

$$\frac{\partial_x w_0}{c} + \sum_{i=1}^m c_i \frac{\partial_x w_i}{cw_i} = SQ, \quad \frac{\partial_y w_0}{c} + \sum_{i=1}^m c_i \frac{\partial_y w_i}{cw_i} = -SP.$$

Die andere Richtung beweisen wir wie folgt. Wenn w_0, \dots, w_m , die algebraisch über K sind, und $c_1, \dots, c_m \in \mathbb{C}$ existieren, so daß

$$\partial_x w_0 + \sum_{i=1}^m c_i \frac{\partial_x w_i}{w_i} = SQ \quad \partial_y w_0 + \sum_{i=1}^m c_i \frac{\partial_y w_i}{w_i} = -SP,$$

dann gilt

$$D(w_0) + \sum_{i=1}^m c_i \frac{D(w_i)}{w_i} = P(SQ) - Q(SP) = 0$$

und, wegen $S \neq 0$, P oder $Q \neq 0$,

$$\partial w_0 + \sum_{i=1}^m c_i \frac{\partial w_i}{w_i} \neq 0 \quad \text{für ein } \partial \in \Delta.$$

Wir können dann eine elementare Erweiterung (L, Δ) von (K, Δ) finden mit $C(L, \Delta) = C(K, \Delta)$, wobei L ein w enthält mit $Dw = 0$ und $\partial w \neq 0$ für ein $\partial \in \Delta$, was bedeutet, daß ein elementares erstes Integral existiert.

□

Bemerkung: Die negierte Aussage von Satz 2.11 wird ebenfalls benötigt: Sind die Punkte (i) und (ii) aus dem Satz nicht erfüllt, so existiert kein über K algebraisches Element $S \neq 0$, das der Gleichung 2.7 genügt.

Aus dem Satz 2.11 lassen sich jetzt die folgenden Schritte zum Vorgehen bei der Suche nach elementaren ersten Integralen für das System 2.2 ablesen.

- (A) Man entscheide, ob $Dw = 0$ nichtkonstante Lösungen in $\mathbb{C}(x, y)$ hat. Ist das der Fall, so suche man eine dieser Lösungen. Sie ist ein rationales erstes Integral von 2.2 und man ist fertig.
- (B) Wenn $Dw = 0$ nur konstante Lösungen in $\mathbb{C}(x, y)$ hat, dann stelle man fest, ob $D(\ln S) = -(\partial_x P + \partial_y Q)$ eine nichtkonstante, über $\mathbb{C}(x, y)$ algebraische Lösung S mit $S^n \in \mathbb{C}(x, y)$ für ein $n \in \mathbb{Z}$ hat. Existiert eine solche Lösung, so suche man diese.
- (C) Falls (B) gilt, frage man nach der Existenz von w_0, \dots, w_m , die algebraisch über $\mathbb{C}(x, y)$ sind, und nach der Existenz von $c_1, \dots, c_m \in \mathbb{C}$, die die Gleichungen

$$\partial_x(w) = SQ \quad \text{und} \quad \partial_y(w) = -SP$$

mit $w = w_0 + \sum_{i=1}^m c_i \ln w_i$ erfüllen. Existieren die w_0, \dots, w_m und c_1, \dots, c_m , so müssen auch diese gefunden werden. Das gerade dargestellte w ist dann ein elementares erstes Integral.

Bemerkung: Sofern eine der in (B) und (C) auftretenden Existenzentscheidungen negativ ausfällt, gibt es zu 2.2 keine elementaren ersten Integrale. Dies ergibt sich aus den negierten Aussagen der Sätze 2.11 und 2.12.

(C) wird nun komplett gelöst.

Sei k der algebraische Abschluß von $\mathbb{C}(y)$ und $F = k(x, S)$. Unter Betrachtung von F als einen gewöhnlichen Differentialkörper mit der Ableitung ∂_x haben wir $C(F, \{\partial_x\}) = k$.

Der erste Schritt zur Lösung von (C) ist zu entscheiden, ob $u_0, \dots, u_n \in F$ und $c_1, \dots, c_n \in k$ existieren mit $SQ = \partial_x u_0 + \sum_{i=1}^n c_i \frac{\partial_x u_i}{u_i}$. F ist algebraisch über $k(x)$ (weil nach (B) S algebraisch ist), also kann hier der sogenannte *Risch-Algorithmus* angewendet werden.

Bemerkung: Der Risch-Algorithmus gestattet, die Integration von elementaren (in diesem Fall algebraischen) Funktionen algorithmisch auszuführen. Insbesondere ist es möglich, mit ihm zu entscheiden, ob das Integral durch elementare

Funktionen ausgedrückt werden kann. Eine ausführliche Beschreibung des Algorithmus findet der Leser in [6].

Hat man nun bei der Anwendung des Risch-Algorithmus festgestellt, daß keine u_i und c_i existieren, ist man fertig. Existieren sie, dann kann man voraussetzen, daß die c_i linear unabhängig über \mathbb{Q} sind und daß gilt: $\partial_x u_i \neq 0, i = 1, \dots, n$.

Als nächstes wird entschieden, ob $c_i \in \mathbb{C}$, d.h. $\partial_y c_i = 0$, für alle i erfüllt ist. Mit Hilfe eines indirekten Beweises kann gezeigt werden: Gilt für ein i $c_i \notin \mathbb{C}$, dann hat das Differentialgleichungssystem 2.2 kein elementares erstes Integral, und wir sind fertig. Damit können wir voraussetzen, daß wir über F algebraische u_0, \dots, u_n und $c_1, \dots, c_n \in \mathbb{C}$ gefunden haben, die

$$SQ = \partial_x u_0 + \sum_{i=1}^n c_i \frac{\partial_x u_i}{u_i}$$

erfüllen.

Jetzt betrachten wir den folgenden Ausdruck:

$$I = \partial_y u_0 + \sum_{i=1}^n c_i \frac{\partial_y u_i}{u_i} + SP.$$

Da ∂_x und ∂_y vertauschbar sind, haben wir

$$\partial_x I = \partial_y (\partial_x u_0 + \sum_{i=1}^n c_i \frac{\partial_x u_i}{u_i} - SQ) = 0.$$

Dies folgt aus (B), weil $DS = -(\partial_x P + \partial_y Q)S$ die Gleichung $\partial_x SP = -\partial_y SQ$ liefert. Somit ist I in k und auch in einer elementaren Erweiterung von $\mathbb{C}(y)$. Nun wird wieder der oben genannte Risch-Algorithmus angewendet, um zu entscheiden, ob über $\mathbb{C}(y)$ algebraische Elemente v_0, \dots, v_m , und $d_1, \dots, d_m \in \mathbb{C}$ existieren, so daß

$$I = \partial_y v_0 + \sum_{i=1}^m d_i \frac{\partial_y v_i}{v_i}.$$

Falls ja, haben wir:

$$\begin{aligned} \partial_x I &= \partial_y \left(\partial_x v_0 + \sum_{i=1}^m d_i \frac{\partial_x v_i}{v_i} \right) \\ \Rightarrow \partial_x u_0 - \partial_x v_0 + \sum_{i=1}^n c_i \frac{\partial_x u_i}{u_i} - \sum_{i=1}^m d_i \frac{\partial_x v_i}{v_i} &= SQ \\ \partial_y u_0 - \partial_y v_0 + \sum_{i=1}^n c_i \frac{\partial_y u_i}{u_i} - \sum_{i=1}^m d_i \frac{\partial_y v_i}{v_i} &= -SP. \end{aligned}$$

Somit sind wir fertig. Existieren keine solchen Elemente, dann kann wiederum indirekt gezeigt werden, daß keine w_0, \dots, w_k , die algebraisch über $\mathbb{C}(x, y)$ sind, und

$e_1, \dots, e_k \in \mathbb{C}$ existieren, so daß

$$\partial_y w_0 + \sum e_i \frac{\partial_y w_i}{w_i} = -SP.$$

Falls nämlich solche Elemente existierten, hätten wir

$$I = \partial_y u_0 + \sum_{i=1}^k c_i \frac{\partial_y u_i}{u_i} + \partial_y w_0 + \sum_{i=1}^n e_i \frac{\partial_y w_i}{w_i}.$$

Das bedeutet, daß I ein Integral bzgl. ∂_y in einer elementaren Erweiterung von $\mathbb{C}(x, y)$ hätte. Da I algebraisch über $\mathbb{C}(y)$ und $\mathbb{C}(x, y)$ eine elementare Erweiterung bzgl. ∂_y von $\mathbb{C}(y)$ mit neuen Konstanten ist, liefert das *Starke Liouvillesche Theorem* (dargestellt in [4]) folgendes: Es existieren v_0, \dots, v_m , die algebraisch über $\mathbb{C}(y)$ sind, und $d_1, \dots, d_m \in \mathbb{C}$, so daß

$$I = \partial_y v_0 + \sum_{i=1}^m d_i \frac{\partial_y v_i}{v_i}.$$

Das steht im Widerspruch zur Voraussetzung.

Damit ist das Problem (C) vollständig behandelt worden. Wir haben herausbekommen, daß entweder keine elementaren ersten Integrale existieren, oder

$$w = u_0 - v_0 + \sum_{i=1}^n c_i \ln u_i - \sum_{i=1}^m d_i \ln v_i$$

ist ein elementares erstes Integral.

Bemerkung: Liegt S selbst in $\mathbb{C}(x, y)$, dann hat das Problem (C) immer eine Lösung.

Mit den Ausführungen zu (C) wissen wir jetzt, wie im zweiten Teil des 3. Schrittes vom Algorithmus vorzugehen ist.

Wir behandeln nun Punkt (A).

Sei $w \in \mathbb{C}(x, y)$. Wir schreiben w als:

$$w = \prod_{i=1}^m f_i^{n_i}$$

mit f_i irreduzibel in $\mathbb{C}[x, y]$, $n_i \in \mathbb{Z}$.

Da die Gleichung $Dw = 0$ erfüllt ist, muß für alle f_i gelten: $f_i \mid Df_i$, wegen:

$$\ln w = \sum n_i \ln f_i \quad \Rightarrow \quad D(\ln w) = 0$$

$$\begin{aligned}
&\Rightarrow \sum_{i=1}^m n_i \frac{Df_i}{f_i} = 0 \\
&\Rightarrow 0 = n_1 \frac{Df_1}{f_1} + \frac{h}{f_2 \cdots f_m} \quad \text{mit} \\
&h = n_2 Df_2 f_3 \cdots f_m + \cdots + n_m f_2 \cdots f_{m-1} Df_m \\
&\Rightarrow -n_1 Df_1 (f_2 \cdots f_m) = h f_1 \\
&\text{Es gilt: } f_1 \nmid (f_2 \cdots f_m),
\end{aligned}$$

da alle f_i paarweise verschieden und irreduzibel sind.

$$\Rightarrow f_1 \mid Df_1$$

Für die restlichen f_i kann die gleiche Rechnung geführt werden.

Wir gehen zur Behandlung von Punkt (B) über.

Nun sei S ein Element, das algebraisch über $\mathbb{C}(x, y)$ ist, mit $S^n \in \mathbb{C}(x, y)$, so daß

$$D(\ln S) = -(\partial_x P + \partial_y Q).$$

Gleicher Ansatz wie für w liefert:

$$S = \prod_{i=1}^m f_i^{r_i}$$

mit f_i irreduzibel in $\mathbb{C}[x, y]$, $r_i \in \mathbb{Q}$, $i = 1, \dots, m$.

Wie bei (A) kann folgende Rechnung durchgeführt werden:

$$\begin{aligned}
D(\ln S) &= \frac{DS}{S} = \sum_{i=1}^m r_i \frac{Df_i}{f_i} = -(\partial_x P + \partial_y Q) \\
&\Rightarrow r_1 \frac{Df_1}{f_1} + \cdots + r_m \frac{Df_m}{f_m} = h \in \mathbb{C}[x, y] \\
&\Rightarrow r_1 \frac{Df_1}{f_1} + \frac{l}{f_2 \cdots f_m} = h \\
&\Rightarrow r_1 f_2 \cdots f_m Df_1 = h f_1 \cdots f_m - l f_1 \\
&\Rightarrow f_1 \mid Df_1.
\end{aligned}$$

Die gleiche Rechnung liefert für alle anderen f_i ebenfalls diese Bedingung.

Damit sind die Probleme (A) und (B) auf das Problem zurückgeführt worden, irreduzible Polynome f zu finden, für die $f \mid Df$ gilt. Zwei Möglichkeiten solche Polynome zu finden, werden im Abschnitt 2.6 beschrieben.

2.5.1. Die Resultate von Darboux

Fragt man nach irreduziblen Polynomen f , für die $f \mid Df$ gilt, so kann man von folgendem ausgehen: Der Totalgrad aller solcher Polynome für einen festen Operator D ist beschränkt. Dies besagen Ergebnisse von **Darboux**, die in [5] dargestellt sind. Allerdings ist noch kein Verfahren zur Berechnung einer effektiven Schranke bekannt. Darum ist die bisherige Vorgehensweise in diesem Punkt heuristisch. Bei der algorithmischen Behandlung der Suche nach elementaren ersten Integralen von 2.2 bleibt demnach folgendes Problem offen:

Wird ein linearer partieller Differentialoperator D wie in 2.4 gegeben, dann bestimme man ein

$N \in \mathbb{N}$, für das gilt: Ist ein $f \in \mathbb{C}(x, y)$ irreduzibel und erfüllt $f \mid Df$, so ist sein Totalgrad nicht größer als N .

Die erwähnten Ergebnisse von Darboux beinhalten nun folgendes:

Hat man bereits irreduzible Polynome f_1, \dots, f_m gefunden, die $f_i \mid Df_i$, $i = 1, \dots, m$, erfüllen, dann gilt eine der beiden nachstehenden Aussagen:

(i) $m \leq ((d+1)d/2) + 2$ mit $d = \max\{\text{tdeg } P, \text{tdeg } Q\}$

(ii) $\exists n_1, \dots, n_m \in \mathbb{Z}$, die nicht alle Null sind und die Gleichung

$$\sum_{i=1}^m n_i \frac{Df_i}{f_i} = 0$$

erfüllen.

Der Fall (ii) liefert den 2. Schritt des Algorithmus.

Wir gehen nun davon aus, daß wir zu einem gegebenen Operator D ein N mit den obigen Eigenschaften kennen. Jetzt lassen sich alle irreduziblen Polynome f mit $f \mid Df$ finden. Ist ihre Anzahl größer als $d(d+1)/2 + 2$, so können wir ein rationales erstes Integral zu 2.2 konstruieren. Ist die Anzahl nicht größer als $d(d+1)/2 + 2$, dann seien f_1, \dots, f_m diese Polynome. Wir suchen nun nach rationalen Zahlen r_1, \dots, r_m , die die Relation

$$\sum_{i=1}^m r_i \frac{Df_i}{f_i} = -(\partial_x P + \partial_y Q)$$

erfüllen. Dies ist Gegenstand im ersten Teil des 3. Schrittes vom Algorithmus.

Es hat sich herausgestellt, daß mit Hilfe des Algorithmus aus irreduziblen Polynomen f mit $f \mid Df$, deren Totalgrade eine gegebene Zahl N nicht übersteigen, elementare erste Integrale konstruiert werden können. Liefert der Algorithmus

nur Konstanten als Ergebnis, so können noch nichttriviale elementare erste Integrale existieren, die sich aus irreduziblen Polynomen mit der obigen Eigenschaft und mit Totalgraden größer als N konstruieren lassen. Trotz dieser starken Einschränkung kann in vielen Fällen mit dem Algorithmus ein nichttriviales elementares erstes Integral gefunden werden.

Beispiel 1: Wir wollen noch einmal die Räuber-Beute-Gleichungen

$$\frac{dx}{dt} = ax - bxy \quad \frac{dy}{dt} = -cy + dxy \quad a, b, c, d \in \mathbb{R}^+$$

betrachten. Der zugehörige Differentialoperator D hat das Aussehen:

$$D = (ax - bxy)\partial_x + (-cy + dxy)\partial_y.$$

Wir setzen $N = 1$ und bekommen als Polynome f , die $f \mid Df$ erfüllen, x und y . Desweiteren gilt:

$$\frac{Dx}{x} = -by + a \quad \text{und} \quad \frac{Dy}{y} = dx - c.$$

Falls nun b oder d ungleich Null ist, sind die beiden obigen Brüche linear unabhängig über \mathbb{Z} . Also muß man die Gleichung

$$r_1 \frac{Dx}{x} + r_2 \frac{Dy}{y} = r_1(-by + a) + r_2(dx - c) = -a + by - c + dx$$

lösen. Man erhält: $r_1 = r_2 = -1$.

Sei jetzt $S = x^{-1}y^{-1}$ und wir suchen nach w_0, \dots, w_n , die algebraisch über $\mathbb{C}(x, y)$ sind, und $c_1, \dots, c_n \in \mathbb{C}$, so daß

$$\begin{aligned} \partial_x w_0 + \sum_{i=1}^n c_i \frac{\partial w_i}{w_i} &= SQ = \frac{c}{x} + d \\ \partial_y w_0 + \sum_{i=1}^n c_i \frac{\partial w_i}{w_i} &= -SP = \frac{a}{y} + b. \end{aligned}$$

Als Lösung bekommen wir:

$$w_0 = dx + by, \quad w_1 = x, \quad w_2 = y, \quad c_1 = -c, \quad c_2 = -a,$$

und damit ist

$$w = dx + by - c \ln x - a \ln y$$

ein elementares erstes Integral der Räuber-Beute-Gleichungen.

Beispiel 2: Nun betrachten wir das System

$$\frac{dx}{dt} = x \quad \frac{dy}{dt} = y.$$

Der zugehörige Differentialoperator ist

$$D = x\partial_x + y\partial_y.$$

Wir setzen wieder $N = 1$ und bekommen als Polynome f mit $f \mid Df$

$$f = \alpha x + \beta y,$$

wobei α und β beliebige komplexe Zahlen sein können. Da in diesem Fall wegen $d = \max\{\text{tdeg } P, \text{tdeg } Q\} = \max\{1, 1\} = 1$

$$\frac{(d+1)d}{2} + 2 = 3$$

gilt, nehmen wir uns z.B. die folgenden drei irreduziblen Polynome her:

$$f_1 = x, \quad f_2 = y, \quad f_3 = x - y.$$

Damit bekommt man

$$s_1 \frac{Dx}{x} + s_2 \frac{Dy}{y} + s_3 \frac{D(x-y)}{x-y} = 0.$$

Diese Gleichung wird z.B. durch das Tripel $s_1 = 1, s_2 = 1, s_3 = -2$ erfüllt. Das heißt,

$$w = xy(x-y)^{-2}$$

ist ein rationales erstes Integral.

2.6. Das Problem $f \mid Df$

Um die Erläuterungen des Algorithmus zu vervollständigen, muß noch das folgende Problem geklärt werden:

Man suche zu einem gegebenen linearen partiellen Differentialoperator

$$D = P\partial_x + Q\partial_y \quad \text{mit } P, Q \in \mathbb{C}[x, y]$$

alle irreduziblen Polynome $f \in \mathbb{C}[x, y]$, für die $f \mid Df$ gilt.

Da keine Schranke des Totalgrades dieser Polynome bekannt ist, geben wir uns eine natürliche Zahl N größer Null vor, so daß wir nun nach irreduziblen Polynomen f mit der obigen Eigenschaft und mit $\text{tdeg } f \leq N$ suchen. Es werden zwei Möglichkeiten vorgestellt, das obige Problem zu behandeln. Zu beiden Möglichkeiten wurden in REDUCE Programme erstellt.

1. Möglichkeit

Lemma 2.13 Sei der lineare partielle Differentialoperator $D = P\partial_x + Q\partial_y$ mit obigen P und Q gegeben. Erfüllt ein Polynom f die Bedingung $f \mid Df$, so ist dies äquivalent der Aussage: Es existiert ein $g \in \mathbb{C}[x, y]$ mit $Df = gf$. Für den Totalgrad eines solchen Polynoms g gilt dann:

$$\text{tdeg } g \leq \max\{\text{tdeg } P, \text{tdeg } Q\} - 1.$$

Beweis: Wir betrachten die Totalgrade in der Gleichung $P\partial_x f + Q\partial_y f = gf$.

$$\begin{aligned} \implies \text{tdeg}(P\partial_x f + Q\partial_y f) &\leq \max\{\text{tdeg } P + \text{tdeg } f - 1, \text{tdeg } Q + \text{tdeg } f - 1\} \\ &= \max\{\text{tdeg } P, \text{tdeg } Q\} + \text{tdeg } f - 1 \end{aligned}$$

$$\text{tdeg}(gf) = \text{tdeg } g + \text{tdeg } f \leq \max\{\text{tdeg } P, \text{tdeg } Q\} + \text{tdeg } f - 1$$

$$\implies \text{tdeg } g \leq \max\{\text{tdeg } P, \text{tdeg } Q\} - 1$$

□

Sei $M := \max\{\text{tdeg } P, \text{tdeg } Q\} - 1$. Wir setzen jetzt g als Polynom mit dem Totalgrad M und f als Polynom mit dem Totalgrad N an. Man kann nun die beiden Seiten der Gleichung

$$P\partial_x f + Q\partial_y f = gf$$

berechnen und Koeffizientenvergleich nach verschiedenen Monomen durchführen. So entsteht ein algebraisches Gleichungssystem, dessen Lösung Bedingungen für die Koeffizienten aus den Ansätzen von f und g liefert.

Beispiel:

$$\begin{aligned} D &= (ax - bxy)\partial_x + (-cy + dxy)\partial_y \\ N = 1 & \quad : \quad f := \alpha_{10}x + \alpha_{01}y + \alpha_{00} \\ \text{tdeg } P = 2, & \quad \text{tdeg } Q = 2 \\ \implies g &:= \beta_{10}x + \beta_{01}y + \beta_{00} \end{aligned}$$

$$\begin{aligned} Df &= gf \\ \implies (ax - bxy)\alpha_{10} + (cy + dxy)\alpha_{01} &= (\alpha_{10}x + \alpha_{01}y + \alpha_{00})(\beta_{10}x + \beta_{01}y + \beta_{00}) \end{aligned}$$

Gleichungssystem:

$$\begin{aligned} \alpha_{01}d - \alpha_{10}b &= \alpha_{01}\beta_{10} + \alpha_{10}\beta_{01} \\ 0 &= \alpha_{10}\beta_{10} \\ 0 &= \alpha_{01}\beta_{01} \\ a\alpha_{10} &= \alpha_{10}\beta_{00} + \alpha_{00}\beta_{10} \\ -c\alpha_{01} &= \alpha_{01}\beta_{00} + \alpha_{00}\beta_{01} \\ 0 &= \alpha_{00}\beta_{00} \end{aligned}$$

$$\Rightarrow \text{1. Lösung: } \alpha_{10} = 0, \beta_{10} = d, \beta_{00} = -c, \alpha_{00} = 0, \beta_{01} = 0 \\ \Rightarrow f = \alpha_{01}y \quad \alpha_{01} \text{ beliebig}$$

$$\text{2. Lösung: } \alpha_{01} = 0, \beta_{01} = -b, \beta_{00} = a, \alpha_{00} = 0, \beta_{10} = 0 \\ \Rightarrow f = \alpha_{10}x \quad \alpha_{10} \text{ beliebig}$$

2. Möglichkeit

Bekannt ist: Gilt $f \mid Df$, so existiert ein $g \in \mathbb{C}[x, y]$ mit $Df = gf$. Dies ist äquivalent zur Aussage: Df liegt in dem von f erzeugten Ideal. Wir setzen f wie in der ersten Möglichkeit an. Mit diesem Ansatz berechnet man Df . Nun wird die Gröbnerbasis des Ideals $\langle f, Df \rangle$ bzgl. der Variablen x, y sowie der Koeffizienten aus dem Ansatz von f in dieser Reihenfolge berechnet. Man bekommt eine Darstellung der Gröbnerbasis, in der f und weitere Polynome in x, y und in den Koeffizienten vorkommen. Da f allein die Basis des Ideals bilden soll, erhält man durch Koeffizientenvergleich wieder Bedingungen für die Koeffizienten. Ein Beispiel soll das Vorgehen veranschaulichen.

Beispiel: $D = y\partial_x + (xy + 9)\partial_y$

$$N = 1 : f := u_{10}x + u_{01}y + u_{00}$$

$$\text{Gröbnerbasis: } \left\{ \begin{array}{l} -u_{10}^2y - 9u_{10}u_{01} + u_{01}^2y^2 + u_{01}u_{00}y, \\ u_{10}y + u_{01}xy + 9u_{01}, \\ u_{10}x + u_{01}y + u_{00} \end{array} \right\}.$$

Da $\langle f, Df \rangle = \langle f \rangle$ gelten soll, müssen u_{00}, u_{01} und u_{10} so gewählt werden, daß die ersten beiden Ausdrücke in der Gröbnerbasis verschwinden. Es folgt demnach:

$u_{10} = u_{01} = 0$ und u_{00} frei wählbar. Das bedeutet, $f = \text{const}$ ist das einzige Polynom mit $f \mid Df$ und $\text{tdeg } f \leq 1$.

Bei der Anwendung der Vorgehensweisen aus den beiden Möglichkeiten steht man vor folgenden Schwierigkeiten:

1. Sind bis zur Schranke N nur konstante Lösungen gefunden worden, heißt das natürlich nicht, daß keine nichttrivialen Lösungen mehr existieren. Um doch noch Lösungen zu bekommen, müßte man N vergrößern und die Suche von vorn beginnen.
2. Die zu lösenden Gleichungssysteme können auch bei kleinen Totalgraden von P und Q so groß werden, daß Zeit- und Speicherprobleme auftreten.

Über die Struktur in der Menge aller irreduziblen Polynome f mit $f \mid Df$ kann folgende Aussage getroffen werden.

Satz 2.14 Sei ein Differentialoperator $D = P\partial_x + Q\partial_y$, $P, Q \in \mathbb{C}[x, y]$, gegeben. Die Polynome $f \in \mathbb{C}[x, y]$ mit der Eigenschaft $f \mid Df$ bilden eine freie Halbgruppe bezüglich der Multiplikation. Existiert zu dem System 2.2 kein rationales erstes Integral, dann ist diese Halbgruppe endlich erzeugt.

Beweis: Seien f und g zwei beliebige Polynome, die die Voraussetzungen des Satzes erfüllen. Das heißt, es existieren zwei weitere Polynome f^* und g^* , so daß gilt:

$$\begin{aligned} ff^* &= Df \quad \text{und} \quad gg^* = Dg. \\ \Rightarrow D(fg) &= D(f)g + fD(g) = \\ &= ff^*g + fg g^* = \\ &= fg(f^* + g^*) \end{aligned}$$

Wir haben zu dem Polynom fg das Polynom $f^* + g^*$ gefunden, wobei $fg(f^* + g^*) = D(fg)$, d.h. $fg \mid Dfg$. Die bereits erläuterten Resultate von Darboux liefern den Beweis der letzten Aussage.

□

Der anschließende Satz stellt einige spezielle Formen des linearen partiellen Differentialoperators D zusammen, für die Polynome f mit der Eigenschaft $f \mid Df$ gefunden wurden.

Satz 2.15 Seien α, β komplexe Zahlen, $a, b, c \in \mathbb{C}[x, y]$, und sei D ein gegebener linearer partieller Differentialoperator der Form $D = P\partial_x + Q\partial_y$ mit $P, Q \in \mathbb{C}[x, y]$.

- (i) Ist eines der beiden Polynome P oder Q gleich Null, so besteht die Menge der Polynome f mit $f \mid Df$ aus allen Primfaktoren des nichtverschwindenden Koeffizientenpolynoms sowie aus allen Polynomen, die nicht von der Differentiationsvariablen abhängen.
- (ii) P und Q seien von folgender Form: $P = ab$ und $Q = ac$. a habe die Primzerlegung $a = \prod_{i=1}^n a_i$. Dann gilt für alle $a_i, 1 \leq i \leq n, a_i \mid Da_i$.
- (iii) Gilt $P = xa(x, y)$ (b.z.w. $Q = yb(x, y)$), so erfüllt $f = \alpha x$ (b.z.w. $f = \alpha y$) die geforderte Bedingung.
- (iiii) Gilt $P = ya$ und $Q = xa$, dann gilt für die Polynome $f_1 = \alpha(x + y)$ und $f_2 = \beta(x - y)$ $f_i \mid Df_i, i = 1, 2$.

Beweis: Es werden hauptsächlich nur die Polynome g angegeben, die $Df = gf$ erfüllen.

- (i) Sei o.B.d.A. $Q = 0$ und $P = \prod_{i=1}^n p_i$. Wir betrachten einen festen Primfaktor $p_j, j \in \{1, \dots, n\}$.

$$\implies g = \partial_x(p_j)p_1 \cdots p_{j-1}p_{j+1} \cdots p_n$$

Sei b ein nichtkonstantes irreduzibles Polynom, das ungleich allen oberen Primfaktoren ist, von x abhängt und für das ebenfalls gilt

$$P\partial_x b = hb.$$

Da b weder P noch $\partial_x b$ teilt, muß b also konstant sein, was im Widerspruch zur Voraussetzung steht.

- (ii) $g_i = a_1 \cdots a_{i-1}a_{i+1} \cdots a_n(b\partial_x a_i + c\partial_y a_i)$
 (iii) $g = a$
 (iiii) $g_1 = a$ und $g_2 = -a$

□

Interessiert man sich also für elementare erste Integrale von Systemen der Art

$$\frac{dx}{dt} = P \quad \frac{dy}{dt} = Q,$$

mit $P, Q \in \mathbb{C}[x, y]$, so sucht man zunächst nach rationalen ersten Integralen und dann nach ersten Integralen der Form

$$w = w_0 + \sum_{i=1}^n c_i \ln w_i,$$

wobei w_0, \dots, w_n algebraisch über $\mathbb{C}(x, y)$ und c_1, \dots, c_n komplexe Zahlen sind. In beiden Fällen gelangt man zu der Aufgabe, irreduzible Polynome $f \in \mathbb{C}[x, y]$ finden zu müssen, die die Eigenschaft $f \mid Df$, mit $D = P\partial_x + Q\partial_y$, haben. Dabei besteht das Problem darin, daß man keine effektive Schranke für den Totalgrad dieser Polynome berechnen kann. Gibt man sich nun eine solche Schranke vor, dann kann man nach Polynomen mit der obigen Eigenschaft suchen, deren Totalgrad diese Schranke nicht übersteigt. In dem Fall, daß die gefundenen Polynome linear abhängig über \mathbb{Z} sind, kann man ein rationales erstes Integral konstruieren, dessen Totalgrad offensichtlich beschränkt ist. Dabei ist der Totalgrad einer rationalen Funktion $f(x, y) = p(x, y)/q(x, y)$, p und q teilerfremd, gegeben durch

$\max\{\text{tdeg } p, \text{tdeg } q\}$. Eine andere Möglichkeit der Suche nach rationalen ersten Integralen bei gegebener Schranke m für den Totalgrad wäre jetzt, nach ersten Integralen der Form

$$f_1 := \sum_{i=0}^m r_i y^i \text{ bzw. } f_2 := \sum_{i=0}^m s_i x^i,$$

$r_i \in \mathbb{C}(x)$ und $s_i \in \mathbb{C}(y)$, zu fragen. Somit könnte man diese Teilaufgabe wie folgt formulieren: Man gebe sich eine natürliche Zahl m vor und finde erste Integrale mit dem obigen Aussehen. Daraus leiten sich Systeme von linearen partiellen Differentialgleichungen in einer gesuchten Funktion f der folgenden Art ab (In unserem Fall sind Lösungen gesucht, die die Form von f_1 haben.):

$$\begin{aligned} P\partial_x f + Q\partial_y f &= 0 \\ \frac{\partial^m f}{\partial y^{m+1}} &= 0. \end{aligned}$$

Diese Systeme haben einen endlich-dimensionalen Lösungsraum. Im anschließenden Teil der Arbeit wird dargestellt, wie für solche speziellen Systeme von Differentialgleichungen vollkommen algorithmisch nach rationalen Lösungen gesucht wird. Dabei verallgemeinern wir die Aufgabenstellung in der Richtung, daß mehrere gesuchte Funktionen zugelassen sind.

3. Spezielle Lösungen bei Systemen von linearen partiellen Differentialgleichungen

3.1. Motivation und Vorbereitung

Eine Grundaufgabe bei der systematischen Behandlung von Differentialgleichungen ist die Bestimmung der Symmetrien zu einer vorgegebenen Differentialgleichung. *Symmetrien* sind Transformationen, die Lösungen einer Differentialgleichung wieder in Lösungen überführen. Kennt man also eine spezielle Lösung der Differentialgleichung, so können aus dieser mit Hilfe der Symmetrien weitere Lösungen konstruiert werden. Bei der Berechnung von Symmetrien steht man irgendwann vor dem Problem, ein System linearer partieller Differentialgleichungen in mehreren Funktionen, *die definierenden Gleichungen*, lösen zu müssen. Ein Weg zur Lösung solcher Systeme besteht darin, geschickt "Termordnungen" zu wählen, um nach Ausführung des "Standardform-Algorithmus" gewöhnliche Differentialgleichungen in dem neuen System zu erhalten. Dieses Vorgehen kann beschleunigt werden, wenn man ständig die entstehenden "kurzen Gleichungen" auswertet. *Kurze Gleichungen* bestehen aus linken Seiten, die nur einen Term enthalten, die rechten Seiten sind Null. Diese Gleichungen werden durch Integration gelöst. Das Ergebnis setzt man dann in die verbleibenden Gleichungen ein. Findet man keine gewöhnlichen Differentialgleichungen, so wähle man eine andere Ordnung und beginne wieder von vorn. Dieses Vorgehen läuft sehr intuitiv ab. Wir wollen in diesem Kapitel einen vollkommen algorithmischen Zugang zur Konstruktion rationaler Lösungen darstellen. Die rationalen Lösungen würden im Fall der definierenden Gleichungen zu speziellen Symmetrien der Differentialgleichung führen.

Eine wesentliche Einschränkung dabei ist, daß wir den Lösungsraum des partiellen Differentialgleichungssystems als endlich-dimensional voraussetzen müssen. Das trifft auf die Systeme definierender Gleichungen bei der Symmetrieberechnung für gewöhnliche Differentialgleichungen zu. Wie sich die Eigenschaft "endlich-dimensionaler Lösungsraum" im Aussehen des Systems widerspiegelt und wie man

den Fall allgemeiner Koeffizienten bei der Suche nach rationalen Lösungen auf den Fall von polynomialen Koeffizienten reduziert, soll zu Beginn gezeigt werden.

3.2. Reduktion der Problemstellung

Zunächst wollen wir die folgenden Bezeichnungen einführen: der Vektor der gesuchten Funktionen sei $\mathbf{u} := (u^1, \dots, u^n)$ und die Menge der Variablen $\mathbf{x} := (x_1, \dots, x_m)$. Gegeben sei nun ein System $\mathbf{S} := (S_1, \dots, S_b)$ von linearen partiellen Differentialgleichungen, wobei jedes $S_i, 0 \leq i \leq b$, die Gestalt

$$\sum_{k,J} c_J^k u_J^k + c = 0$$

hat. c_J^k und c seien die Koeffizienten. Sie können beliebige Funktionen in \mathbf{x} sein. J ist ein Multiindex, der die Ableitungen der Funktionen nach den Variablen darstellt. Die Ableitung u_J^n hat dann z.B. für $J = (j_1, \dots, j_m)$ folgendes Aussehen:

$$u_J^n = \frac{\partial^l u^n}{\partial^{j_1} x_1 \partial^{j_2} x_2 \dots \partial^{j_m} x_m} \quad \text{mit} \quad l = \sum_{k=1}^m j_k.$$

Zusätzlich sei der Lösungsraum von \mathbf{S} als endlich-dimensional vorausgesetzt. Wenn wir von einem endlich-dimensionalen Lösungsraum reden, so ist folgendes gemeint: Der Vektorraum der Lösungen des zugehörigen homogenen Systems bzw. der affine Raum der Lösungen des inhomogenen Systems sei endlich-dimensional.

Beispiel $\mathbf{S} = (S_1, S_2, S_3)$, $\mathbf{u} = (u^1, u^2)$, $\mathbf{x} = (x_1, x_2, x_3)$

$$\begin{aligned} x_1 \sin(x_3) u_{(2,0,3)}^1 + u_{(1,1,1)}^2 &= 0 & S_1 \\ x_3 x_2 u_{(1,0,3)}^1 &= 0 & S_2 \\ \cos(x_1) u_{(2,3,0)}^2 &= 0 & S_3 \end{aligned}$$

Der Multiindex $J = (0, 0, \dots, 0)$ wird üblicherweise nicht geschrieben, da die durch ihn dargestellte Ableitung die Funktion selbst ist.

Bemerkung: Der Einfachheit halber werden von nun an alle Rechnungen über dem Zahlkörper \mathbf{Q} ausgeführt.

Die folgende Aussage gibt an, wie sich \mathbf{S} reduzieren lässt, wenn wir nach rationalen Lösungen fragen.

Satz 3.1 *Sind für ein System linearer partieller Differentialgleichungen \mathbf{S} rationale Lösungen gesucht, so lässt sich \mathbf{S} in ein äquivalentes lineares System umformen, in dem sämtliche Koeffizientenfunktionen Polynome in den Variablen und die Koeffizienten innerhalb einer Differentialgleichung außerdem teilerfremd sind.*

Beweis: Wir betrachten den von den Koeffizienten aufgespannten Vektorraum $V = \text{span}(c_J^k, c)$ über dem Körper $K = \mathbf{Q}(x_1, \dots, x_m)$. Die Basis von V über K sei d_1, \dots, d_t . Daraus folgt, daß alle c_J^k und c die Darstellungen

$$c_J^k = \sum_{i=1}^t b_J^{ik} d_i \quad \text{mit} \quad b_J^{ik} \in K, 0 \leq i \leq t,$$

$$c = \sum_{i=1}^t b_i d_i \quad \text{mit} \quad b_i \in K, 0 \leq i \leq t,$$

zulassen. Wir nutzen diese Darstellungen nun zur Umformung der Gleichungen des Systems und betrachten eine beliebige Differentialgleichung aus **S**. Sie kann folgendermaßen geschrieben werden:

$$\sum_{k,J} c_J^k u_J^k + c = \sum_{k,J} \left(\sum_{i=1}^t b_J^{ik} d_i \right) u_J^k + \sum_{i=1}^t b_i d_i = \sum_{i=1}^t \left(\sum_{k,J} b_J^{ik} u_J^k + b_i \right) d_i.$$

Die b_J^{ik} und b_i sind nach Voraussetzung rationale Funktionen. Auch die u_J^k sind rationale Funktionen, da wir nach rationalen Lösungen fragen und beliebige Ableitungen von rationalen Funktionen wieder rationale Funktionen ergeben. Die letzte Gleichung ist genau dann erfüllt, wenn alle Koeffizienten vor den Basisfunktionen verschwinden. Wir setzen also diese Koeffizienten Null und wiederholen die Aufspaltung für die restlichen Gleichungen des Systems **S**. So erhalten wir ein äquivalentes lineares System von Differentialgleichungen, das nur noch rationale Koeffizientenfunktionen enthält. Wenn wir jetzt jede Differentialgleichung mit dem Hauptnenner der Koeffizientenfunktionen multiplizieren, dann haben wir zunächst polynomiale Koeffizienten. (Im folgenden nennen wir dieses Vorgehen kurz "Überführung in eine ganz-rationale Form".) Wir dividieren noch jede Gleichung durch den größten gemeinsamen Teiler ihrer Koeffizientenpolynome und erhalten die letzte Aussage des Satzes.

□

Somit haben wir das lineare partielle Differentialgleichungssystem **S** mit beliebigen Koeffizientenfunktionen umgewandelt in ein System mit Koeffizientenpolynomen, die dazu noch innerhalb einer jeden Gleichung teilerfremd sind. Das hier dargelegte Vorgehen zur Spezialisierung von Systemen linearer Differentialgleichungen bei der Suche nach rationalen Lösungen ist aus [8] entnommen, wo es für den Fall gewöhnlicher Differentialgleichungen beschrieben wurde.

3.3. Standardformen

Die *Standardform* eines linearen partiellen Differentialgleichungssystems **S** ist ein äquivalentes lineares partielles Differentialgleichungssystem, das gewisse Eigen-

schaften hat, auf die wir noch eingehen werden. Zur Überführung eines solchen Systems in seine Standardform benutzt man den "Standardform-Algorithmus". Der Standardform-Algorithmus besteht im wesentlichen aus drei Teilen: "Termordnungen", "Reduktionen" und "Bildung von S-Gleichungen". O.B.d.A. wählen wir die lexikographische Ordnung als Ordnung in den Funktionen und in den Variablen:

$$u^1 > u^2 > \dots > u^n \quad \text{und} \quad x_1 > x_2 > \dots > x_m. \quad (3.1)$$

Eine Ableitung einer Funktion u^i nach den Variablen heißt *Monom*. Mit den beiden Ordnungen aus 3.1 hat man automatisch eine Ordnung in den Monomen gegeben, indem man erst die Funktionen und dann die Grade der Ableitungen nach den einzelnen Variablen vergleicht.

Eine *Termordnung* ist eine Totalordnung " $<$ " auf der Menge aller Monome, für die gilt:

$$1 < M \quad \text{für alle } M \quad \text{und} \quad M_1 < M_2 \Rightarrow MM_1 < MM_2 \quad \text{für alle } M, M_1, M_2.$$

Da das größte Monom einer Differentialgleichung und sein Koeffizient eine besondere Rolle spielen, definieren wir:

Definition 3.2 *Das größte Monom in einer Differentialgleichung heißt führendes Monom und wird mit LM bezeichnet. Der Koeffizient vor dem führenden Monom heißt führender Koeffizient. Wir bezeichnen ihn mit LC . Das Produkt aus führendem Monom und führendem Koeffizienten ist der führende Term. Er bekommt die Bezeichnung LT .*

Mit einer gegebenen Termordnung läßt sich dann auch eine Ordnung zwischen den Gleichungen von S einführen, indem zunächst die führenden Monome verglichen werden. Falls diese gleich sind, vergleicht man die nächstgrößeren u.s.w. Das System S erhält durch die Ordnung in seinen Gleichungen eine Blockstruktur. Das heißt, es teilt sich in n Blöcke bzw. Untersysteme. Die führenden Monome eines Untersystems sind ausschließlich Ableitungen einer gesuchten Funktion u^i . Man kann die Blöcke nun z.B. so anordnen, daß der Block mit der größten Funktion u^1 ganz oben und der Block mit der kleinsten Funktion u^n ganz unten steht. Die Operationen "Reduktion" und "Bildung von S-Gleichungen" können nun entweder blockweise oder im gesamten System z.B. von unten beginnend ausgeführt werden.

Reduktion einer Differentialgleichung S_i durch eine Differentialgleichung S_j bedeutet, daß S_j in geeigneter Weise differenziert und mit einem Polynom multipliziert wird. Der neue führende Term von S_j annulliert bei Bildung der Differenz $S_i - S_j$ einen Term aus S_i . Wir nennen ein System S linearer partieller Differentialgleichungen *reduziert*, wenn sich kein S_i von S durch ein $S_j, i \neq j$, von S reduzieren

läßt.

Für die Gleichungen $S_i, i = 1, \dots, b$, aus \mathbf{S} wollen wir entsprechend dem Gleichungsindex die führenden Monome, Koeffizienten und Terme mit LM_i, LC_i und LT_i in dieser Reihenfolge bezeichnen.

Die S -Gleichung zu S_i und S_j bildet sich folgendermaßen:

$$S(S_i, S_j) := D_i S_i - \frac{LC_i}{LC_j} D_j S_j,$$

wobei D_i und D_j die kleinsten Ableitungen sind, die folgende Relation erfüllen: $D_i LM_i = D_j LM_j$. (Kleinste Ableitung bedeutet minimale Anzahl von Differentiationen.) Die Bildung von S -Gleichungen kann als Hinzunahme von Gleichungen, die sich als Integrabilitätsbedingungen ergeben, interpretiert werden.

Definition 3.3 Ein System $\mathbf{S} = (S_1, S_2, \dots, S_b)$ von linearen partiellen Differentialgleichungen ist ein System in **Standardform**, wenn jede S -Gleichung $S(S_i, S_j)$ durch die Gleichungen von \mathbf{S} zu Null reduziert werden kann.

Im folgenden gehen wir davon aus, daß \mathbf{S} ein System in Standardform bzgl. lexikographischer Ordnung (siehe vorn) ist. Wir schreiben die Gleichungen S_i nun so um, daß auf der linken Seite nur LT_i stehen. In der Standardform \mathbf{S} nennt man die Monome, die sich nicht durch Differentiationen aus den LM_i ergeben, *Parametrische Ableitungen*. Um die Dimension des Lösungsraumes festzustellen, zählt man nun einfach alle parametrischen Ableitungen. Somit hat \mathbf{S} einen endlich-dimensionalen Lösungsraum genau dann, wenn die Anzahl der parametrischen Ableitungen endlich ist. Der nächste Satz ist ein Kriterium, mit dem man nur anhand der $LM_i, 1 \leq i \leq b$, feststellen kann, ob \mathbf{S} einen endlich-dimensionalen Lösungsraum hat.

Satz 3.4 Sei \mathbf{S} ein lineares partielles Differentialgleichungssystem in Standardform bzgl. obiger Ordnung. \mathbf{S} hat genau dann einen endlich-dimensionalen Lösungsraum, wenn gilt:

Zu jeder Funktion u^i und zu jeder Variablen x_j gibt es eine Gleichung in \mathbf{S} mit führendem Monom der Form $\partial^k u^i / \partial x_j^k$.

Wir formulieren noch weitere Eigenschaften einer Standardform, die wir später benutzen werden.

Satz 3.5 Sei \mathbf{S} ein lineares partielles Differentialgleichungssystem in Standardform mit endlich-dimensionalem Lösungsraum und seien die Ordnungen in den Variablen und Funktionen wie in 3.1 gegeben. Dann hat das Untersystem für u^n

ebenfalls einen endlich-dimensionalen Lösungsraum. Des weiteren ist die letzte Gleichung S_b eine gewöhnliche Differentialgleichung für u^n , wobei x_m die Differentiationsvariable ist.

Die Beweise der beiden Sätze kann der Leser aus [9] entnehmen.

3.4. Algorithmische Behandlung

In diesem Abschnitt gehen wir von einem linearen partiellen Differentialgleichungssystem mit polynomialen Koeffizientenfunktionen und endlich-dimensionalem Lösungsraum aus, welches wir von nun an ein *L-System* nennen und auch weiterhin mit \mathbf{S} bezeichnen. Für solche L-Systeme geben wir jetzt einen Algorithmus an, mit dem sich rationale Lösungen finden lassen. Der Algorithmus besteht darin, die ersten drei der folgenden Schritte nacheinander für alle Funktionen u^1, \dots, u^n auszuführen (O.B.d.A. wird es hier für u^n demonstriert.) und abschließend die Ergebnisse im 4. Schritt noch einmal zu verarbeiten.

Der **Input** des Algorithmus setzt sich also aus einem System von linearen partiellen Differentialgleichungen \mathbf{S} , einem Vektor von gesuchten Funktionen \mathbf{u} und einem Variablen-Vektor \mathbf{x} zusammen. Der **Output** ist ein Vektor, dessen Komponenten rationale Funktionen in x_1, \dots, x_m sind, die noch linear von frei wählbaren konstanten Parametern abhängen.

1. Schritt: Man wähle eine Ordnung (Wir nutzen die Ordnung aus 3.1.) in den gesuchten Funktionen so, daß das gewählte u^i (hier u^n) die kleinste Funktion ist. Bzgl. der daraus entstehenden Termordnung wird mit \mathbf{S} der Standardform-Algorithmus durchgeführt. Man bekommt ein Untersystem SS , in dem nur noch u^n als gesuchte Funktion auftritt und dessen letzte Gleichung eine gewöhnliche Differentialgleichung bzgl. der kleinsten Variablen x_m ist.

2. Schritt: Wir suchen nach rationalen Lösungen der letzten, gewöhnlichen Differentialgleichung. Wie das geschieht, wird noch behandelt. Die gefundene allgemeine rationale Lösung ist linear in frei wählbaren Parametern $\alpha_0, \dots, \alpha_{r-1}$, die noch von x_1, \dots, x_{m-1} abhängen können. Wir testen nun, ob alle auftretenden Parameterfunktionen konstant sind. Wenn dies der Fall ist, sind wir fertig. Sonst gehen wir zum 3. Schritt.

3. Schritt: Die Lösung aus dem 2. Schritt setzt man in die restlichen Gleichungen von SS ein. Die Variable x_m wird jetzt im folgenden Sinn zum Aufsplitten benutzt: Man überführt die Gleichungen in eine ganzrationale Form und macht dann Koeffizientenvergleich nach den Potenzen von x_m . Es entsteht ein neues L-System (Im Anschluß wird gezeigt, daß wirklich wieder ein

L-System herauskommt.), in dem jetzt die $\alpha_0, \dots, \alpha_{r-1}$ die gesuchten Funktionen sind. Die Variablen haben sich um das x_m vermindert. Die Ordnung in den x_1, \dots, x_{m-1} bleibt erhalten. Diese Daten bilden einen neuen Input, mit dem man den Algorithmus erneut startet.

4. Schritt: Wir bekommen aus den ersten drei Schritten für jedes $u^i, i = 1, \dots, n$, eine Parameterdarstellung, die ausschließlich konstante Parameter enthält. Diese Darstellungen setzen wir in eine Standardform von **S** ein. Die Gleichungen werden noch einmal in eine ganzrationale Form überführt. Koeffizientenvergleich nach verschiedenen Monomen der Variablen liefert ein lineares Gleichungssystem in den Parametern, das dann gelöst wird. Dieses Gleichungssystem ist im allgemeinen nicht eindeutig lösbar, d.h. seine Lösung enthält noch unabhängige Parameter, die dann auch als frei wählbare Parameter in der Lösung (u^1, \dots, u^n) auftreten.

Die Vorgehensweise im 1. Schritt ist mit den Ausführungen über die Standardformen klar. Zum 2. Schritt werde ich im nächsten Abschnitt einen Algorithmus angeben, der rationale Lösungen von gewöhnlichen Differentialgleichungen mit Koeffizienten, die Polynome in mehreren Variablen sind, bestimmt. Die Lösungen enthalten Parameter, die Funktionen in allen Variablen außer der Differentiationsvariablen sind.

Entscheidend ist, daß im 3. Schritt wieder ein L-System entsteht:

Lemma 3.6 *Gegeben sei ein L-System **S** und eine Ordnung in den Variablen. Werden dann die ersten drei Schritte des obigen Algorithmus ausgeführt, so ist mit den vorherigen Bezeichnungen das sich ergebende Differentialgleichungssystem für die $\alpha_0, \dots, \alpha_{r-1}$ wieder ein L-System.*

Beweis: Der Kürze halber wollen wir die letzte Gleichung von **S**, also die gewöhnliche Differentialgleichung, wieder mit S_b bezeichnen. Die Lösung von S_b aus dem 2. Schritt ist linear in den Parameterfunktionen (dies wird im nächsten Abschnitt klar). Diese Linearität bleibt natürlich auch beim Differenzieren und bei der Überführung in eine ganzrationale Form erhalten.

Das Differentialgleichungssystem in den Parameterfunktionen entsteht durch Koeffizientenvergleich nach x_m im 3. Schritt. Da die Gleichungen bereits eine ganzrationale Form haben, sind die Koeffizienten Polynome in x_1, \dots, x_{m-1} und den Parametern. Bleibt noch zu zeigen, daß der Lösungsraum des Differentialgleichungssystem für die Parameterfunktionen endlich-dimensional ist.

Wir haben bereits im Satz 3.5 gesehen: Wenn **S** einen endlich-dimensionalen Lösungsraum besitzt, dann besitzt auch das Untersystem SS für u^n einen endlich-dimensionalen Lösungsraum, d.h. die Menge

$$\{u^n \mid u^n \text{ ist Lösung von } SS\}$$

bildet einen endlich-dimensionalen Vektorraum über \mathbf{Q} . Aus S_b bekommen wir folgenden Ansatz für u^n :

$$u^n = \alpha_0 \varphi_0 + \cdots + \alpha_{r-1} \varphi_{r-1},$$

wobei die $\alpha_i \in \mathbf{Q}(x_1, \dots, x_{m-1})$, $0 \leq i \leq r-1$, und die φ_i , $0 \leq i \leq r-1$, linear unabhängig über $\mathbf{Q}(x_1, \dots, x_{m-1})$ sind. Das bedeutet, jeder Lösung u^n von S_b läßt sich eineindeutig ein r -Tupel $(\alpha_0, \dots, \alpha_{r-1})$ zuordnen. Daraus folgt, es existiert ein \mathbf{Q} -Isomorphismus Φ , der jeder Lösung von S_b ein r -Tupel von rationalen Funktionen in x_1, \dots, x_{m-1} zuordnet. Also ist die Menge

$$\{(\alpha_0, \dots, \alpha_{r-1}) \in \mathbf{Q}(x_1, \dots, x_{m-1})^r \mid u^n = \alpha_0 \varphi_0 + \cdots + \alpha_{r-1} \varphi_{r-1} \text{ ist Lösung von } S_b\}$$

ein r -dimensionaler Vektorraum über \mathbf{Q} . In diesem Vektorraum suchen wir nun nach Lösungen für das neue Differentialgleichungssystem aus dem 3. Schritt. Da jeder Teilraum eines endlich-dimensionalen Vektorraumes wieder endlich-dimensional ist, sind wir fertig.

□

Die Suche nach rationalen Lösungen für ein festes u^i wird also im Algorithmus umgewandelt in die Suche nach rationalen Lösungen für ein L-System in mehreren Funktionen. D.h. die Anzahl der gesuchten Funktionen nimmt zu. Jedoch verringert sich die Zahl der unabhängigen Variablen bei jedem Durchlauf der Schritte 1 und 2 um 1. Die Abbildung 3.1 soll dies veranschaulichen.

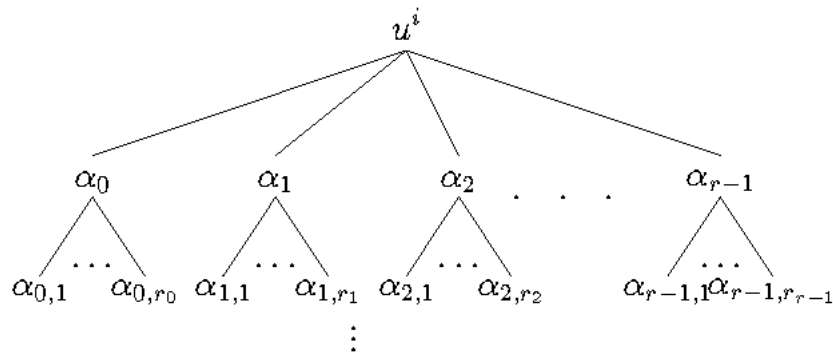


Abbildung 3.1.: Parameterbaum

Die Hierarchie in den verschiedenen Parameterfunktionen wird durch einen Baum beschrieben, in dem die α 's mit einem gemeinsamen 'Vorgänger' immer zusammen in einem L-System auftreten, wobei die Anzahl der Variablen in jeder Stufe mindestens um 1 niedriger ist als in der darüberliegenden Stufe. Im 2. Schritt des obigen Algorithmus ist mit dem Test, ob alle Parameter konstant sind, eine Abbruchbedingung gegeben.

Satz 3.7 *Gegeben sei ein L-System \mathbf{S} . Der Algorithmus zum Finden von rationalen Lösungen von \mathbf{S} bricht nach endlich vielen Schritten ab.*

Beweis: Wie wir bereits gezeigt haben, verringert sich bei jedem Durchlauf der ersten beiden Schritte die Anzahl der Variablen mindestens um 1. Das heißt: Es verschwindet wenigstens die kleinste Variable eines Systems bei dem Übergang zu einem Teilsystem. Da wir nur endlich viele Variablen, nämlich m , im Ausgangsdifferentialgleichungssystem \mathbf{S} voraussetzen, bricht der obige Algorithmus nach endlich vielen Schritten ab.

□

In den bisherigen Ausführungen haben wir gesehen, daß man bei der algorithmischen Suche nach rationalen Lösungen eines L-Systems \mathbf{S} neben der Durchführung von Umformungen, Vereinfachungen und Koeffizientenvergleichen immer wieder gewöhnliche Differentialgleichungen lösen muß, die in den Koeffizienten Variablen aus \mathbf{S} als Parameter enthalten. Sätze über Existenz und Eindeutigkeit sagen nun aus, daß zu einer linearen gewöhnlichen Differentialgleichung entsprechend der Ordnung der Differentialgleichung Lösungen existieren. Allerdings ist das nicht gleichbedeutend mit der Existenz von rationalen Lösungen. Man kann lediglich folgendes feststellen. Falls eine der immer wieder im Algorithmus zu lösenden gewöhnlichen Differentialgleichungen keine rationalen Lösungen besitzt, so besitzt auch das gesamte System keine rationalen Lösungen.

Andererseits kann der Fall eintreten, daß man relativ einfach nichtrationale Lösungen erhält, dafür hat man jedoch im allgemeinen keinen Algorithmus.

3.5. Algorithmische Berechnung von rationalen Lösungen linearer gewöhnlicher Differentialgleichungen

3.5.1. Problemvorstellung und vorbereitende Aussagen

Die Differentialgleichungen, die hier behandelt werden sollen, haben die Form:

$$p_r \frac{\partial^r}{\partial x^r} f + p_{r-1} \frac{\partial^{r-1}}{\partial x^{r-1}} f + \cdots + p_0 f + q = 0. \quad (3.2)$$

Es sind also lineare Differentialgleichungen bezüglich der Variablen x und wir setzen voraus: $p_i, q \in \mathbf{Q}[x_1, \dots, x_{m-1}, x]$, $i = 0, \dots, r$, d.h. die Koeffizientenfunktionen sind Polynome in x_1, \dots, x_{m-1}, x . Man kann auch beliebige Koeffizientenfunktionen voraussetzen, doch da wir uns erneut für rationale Lösungen interessieren, gelangt man mit der Vorgehensweise aus den vorbereitenden Bemerkungen

wieder zu dieser speziellen Form. Gesucht sind Lösungen $f \in \mathbf{Q}(x_1, \dots, x_{m-1}, x)$, also rationale Funktionen über dem Koeffizientenkörper \mathbf{Q} . Es existiert bereits ein Algorithmus für den einfachsten Fall, d.h. für lineare gewöhnliche Differentialgleichungen, in denen die Koeffizienten nicht von Parametern abhängen. Er ist ausführlich in [8] beschrieben.

Fragen wir nach rationalen Lösungen von 3.2, so heißt das, wir suchen ein f der Form

$$f = \frac{Z}{N} \quad \text{mit} \quad Z, N \in \mathbf{Q}[x_1, \dots, x_{m-1}, x]. \quad (3.3)$$

Da x die einzige Differentiationsvariable ist, erweist es sich für die folgenden Konstruktionen als günstiger, Z und N als Elemente von $\mathbf{Q}(x_1, \dots, x_{m-1})[x]$ zu betrachten und auch weiterhin in dem Körper $\mathbf{Q}(x_1, \dots, x_{m-1})$ zu rechnen. Wir können dabei Z und N als relativ prim zueinander voraussetzen.

Der Algorithmus zerfällt in zwei Hauptteile. Wir werden feststellen, daß der Nenner einer möglichen Lösung f nur die Primfaktoren des führenden Koeffizienten p_r enthalten kann. Der erste Teil besteht nun darin, Schranken für die Exponenten der irreduziblen Faktoren zu bestimmen. Danach wird unter Benutzung des bereits bekannten Nenners eine Gradschranke des Zählers für den Grad in x berechnet. Da wir häufig den Grad bzgl. x verwenden, sei dieser mit \deg_x bezeichnet.

Wir geben uns nun o.B.d.A. folgende Darstellung der Primzerlegung des p_r vor:

$$p_r = \prod_{i=1}^t a_i^{l_i} \quad (3.4)$$

und wollen die nachstehende Aussage über die Struktur des Nenners N einer möglichen Lösung f beweisen.

Satz 3.8 *Sei $f = Z/N$, $Z, N \in \mathbf{Q}(x_1, \dots, x_{m-1})[x]$, eine rationale Lösung der Differentialgleichung 3.2 und sei der führende Koeffizient p_r wie in 3.4 faktorisiert. Dann ist jeder Primfaktor des Nenners N von f auch ein Teiler von p_r .*

Vor dem Beweis des Satzes werden wir noch einige hilfreiche Aussagen treffen. Als erstes ist hier der Satz über die eindeutige Entwicklung einer rationalen Funktion in Partialbrüche zu nennen. Wenden wir diesen Satz auf unseren Fall an, so erhalten wir folgendes: Ist eine Funktion f in der Darstellung

$$f = \frac{Z}{N} \quad \text{und} \quad N = \prod_i a_i^{k_i}, \quad Z, N \in \mathbf{Q}(x_1, \dots, x_{m-1})[x]$$

gegeben, so bekommt man nach Partialbruchzerlegung die Darstellung

$$f = a + \sum_{i,j} \frac{e_{ij}}{a_i^j} \quad \text{mit} \quad \deg_x e_{ij} < \deg_x a_i.$$

Nun interessieren wir uns dafür, wie sich der Exponent des Nenners eines Partialbruches nach mehrfachem Ableiten entwickelt.

Lemma 3.9 Sei $g = \frac{a}{b^k}$ ein Partialbruch in $\mathbf{Q}(x_1, \dots, x_{m-1})[x]$. Dann hat $g^{(r)} = \frac{\partial^r g}{\partial x^r}$ die Form

$$g^{(r)} = \frac{a^*}{b^{k+r}} + \dots,$$

wobei die Punkte andeuten sollen, daß nur noch Partialbrüche geringerer Ordnung bzgl. b vorkommen, d.h. a^* und b sind teilerfremd und $\deg_x a^* < \deg_x b$.

Wir bezeichnen die erste Ableitung nach x von nun an mit $'$. Es sei auch darauf hingewiesen, daß für ein irreduzibles Polynom p , das von x abhängt, gilt: $p \nmid p'$.

Beweis des Lemmas: Wir führen Induktion über den Grad der Ableitung durch. Wie im Lemma wird mit drei aufeinander folgenden Punkten angedeutet, daß nur noch Partialbrüche geringerer Ordnung bzgl. b folgen.

Induktionsanfang:

$$\begin{aligned} g' &= \frac{a'b^k - akb^{k-1}b'}{b^{2k}} = \\ &= \frac{a'}{b^k} - \frac{akb'}{b^{k+1}} = \frac{a^*}{b^{k+1}} + \dots \end{aligned}$$

Dabei gilt $a^* \equiv -akb'$ modulo b . Da a und b' teilerfremd zu b sind, ist $a^* \neq 0$.

Induktionsbeweis: Wir starten mit der r -ten Ableitung von g und leiten ein weiteres Mal ab.

$$\begin{aligned} (g^{(r)})' &= \left(\frac{a^*}{b^{k+r}} + \dots \right)' = \\ &= \frac{a^{*'}}{b^{k+r}} - \frac{a^*(k+r)b'}{b^{k+r+1}} + \dots = \frac{a^{**}}{b^{k+r+1}} \end{aligned}$$

Man kann leicht überprüfen, daß die Terme, die durch die Punkte in der obigen Klammer angedeutet sind, nach Ableitung nicht zum Partialbruch höchster Ordnung bzgl. b beitragen können. Die gleiche Argumentation wie für a^* liefert: $\deg_x a^{**} < b$ und $a^{**} \neq 0$.

□

Beweis des Satzes 3.8: Wir wollen in der Darstellung $f = Z/N$ für einen Primfaktor b des Nenners N $b \nmid p_r$ annehmen. Einsetzen der Partialbruchzerlegung von f in die Differentialgleichung liefert dann nach dem Lemma 3.9 einen Ausdruck, der den höchsten Term bzgl. b (höchster Term bzgl. b bedeutet hier und im folgenden Partialbruch höchster Ordnung bzgl. b) enthält. Er hat die Form

$$p_r \frac{a}{b^{k+r}}.$$

Nun zerlegen wir diesen Ausdruck noch einmal in Partialbrüche und erhalten die Differentialgleichung in der Darstellung

$$\frac{a^*}{b^{k+r}} + \dots = 0.$$

Der obige Term ist der höchste Term bzgl. b in der Differentialgleichung und die Punkte stehen für Partialbrüche geringerer Ordnung bzgl. b . Das bedeutet, daß der obige höchste Term bzgl. b nicht annulliert wird, damit kann die Differentialgleichung nicht erfüllt sein.

Wenn b also im Nenner N der Lösung f auftritt, muß b auch ein Teiler des führenden Koeffizienten p_r sein.

□

3.5.2. Der Algorithmus

Sei eine Differentialgleichung wie in 3.2 gegeben. Wir suchen nach rationalen Lösungen f , die die Form aus 3.3 haben.

(N) Bestimmung des Nenners

Wir faktorisieren p_r, \dots, p_0, q und p_r habe die Primzerlegung aus 3.4. O.B.d.A. wird zuerst der Primfaktor a_1 von p_r betrachtet. Sei m_1 der Exponent, mit dem a_1 im Nenner enthalten ist, d.h. sei

$$N = a_1^{m_1} \dots$$

eine Darstellung des Nenners einer möglichen rationalen Lösung f . Unter der Annahme, daß m_1 größer als Null ist, wird eine Schranke M_1 für m_1 gesucht. Dazu muß der kleinste Exponent von a_1 in der Differentialgleichung gefunden werden, wobei man die Exponenten im Zähler positiv und die im Nenner negativ zählt.

(N1) Wir bestimmen die Vielfachheiten von a_1 in den Koeffizienten p_r, \dots, p_0 und bezeichnen sie mit v_r, \dots, v_0 . Berücksichtigt werden nur die Koeffizienten ungleich Null. Zunächst berechnet man:

$$\min_{0 \leq k \leq r} (v_k - k) = V_1.$$

Der kleinste Exponent von a_1 in der Differentialgleichung ergibt sich dann zu

$V_1 - m_1$. Die Indizes der Koeffizienten außer q , in denen a_1 mit diesem Exponenten auftritt, werden in der Menge

$$I_1 = \{k \mid V_1 = v_k - k\}$$

zusammengefaßt.

(N2) Der Exponent von a_1 in q sei v . Dann sind zwei Fälle zu unterscheiden:

(a) $v \leq V_1 - m_1 \implies m_1 \leq V_1 - v$

In diesem Fall haben wir eine Schranke M^* für m_1 . Ist die Differenz $V_1 - v$ echt kleiner als Null bzw. ist $q = 0$, so setzt man $M^* := 0$, andernfalls gilt $M^* = V_1 - v$.

(b) $v > V_1 - m_1$

Eine Schranke M^{**} für m_1 bekommt man hier, indem man alle höchsten Terme bzgl. a_1 zusammenfaßt, die Summe muß Null ergeben,

$$Q(m_1) = 0. \tag{3.5}$$

Im Anschluß an den Algorithmus werden wir diese Relation genauer untersuchen. Die Gleichung 3.5 ist eine Polynomgleichung in x_1, \dots, x_{m-1}, x , in der wir nach verschiedenen Monomen der x_1, \dots, x_{m-1}, x sortieren. Die Koeffizienten vor diesen Monomen werden Null gesetzt. Daraus ergibt sich ein Gleichungssystem, das m_1 erfüllen muß. Wir suchen die größte ganzzahlige Nullstelle. Ist sie größer oder gleich Null, wird sie mit M^{**} bezeichnet, sonst setzt man $M^{**} := 0$.

Die endgültige Schranke M für den Exponenten m_1 von a_1 ist dann

$$M = \max\{M^*, M^{**}\}.$$

Wir wiederholen die Schritte (N1) und (N2) für die restlichen Primfaktoren a_2, \dots, a_t von p_r . Damit haben wir folgenden Nenner N für eine mögliche Lösung f der Differentialgleichung 3.2 erhalten:

$$N = a_1^{M_1} a_2^{M_2} \dots a_t^{M_t}.$$

(Z) Bestimmung des Zählers

Mit Hilfe eines Ansatzes wird eine Schranke S für $\deg_x Z$ bestimmt.

(Z1) Wir setzen f mit dem nun bekannten Nenner in 3.2 ein und erhalten, nachdem wir die Gleichung in eine ganzrationale Form überführt haben, eine lineare gewöhnliche Differentialgleichung für Z :

$$C_r \frac{\partial^r}{\partial x^r} Z + \dots + C_0 Z + C = 0. \tag{3.6}$$

(Induktion über den Ableitungsgrad liefert, daß jede beliebige Ableitung von $\frac{Z}{N}$ linear in den Ableitungen von Z ist.) Man macht folgenden Ansatz für Z :

$$Z = \alpha_s x^s + \alpha_{s-1} x^{s-1} + \dots + \alpha_1 x + \alpha_0$$

mit $\alpha_i = \alpha_i(x_1, \dots, x_{m-1}), \quad 0 \leq i \leq s, \quad s$ beliebig aber fest.

(Z2) Es ist offensichtlich, daß die k -te Ableitung nach x , $0 \leq k \leq r$, den x -Grad von Z um k verringert. Seien die x -Grade der Koeffizientenpolynome gegeben durch l_s, \dots, l_0 . l sei der Grad des Absolutgliedes C_0 bzgl. x . Nachdem wir den obigen Ansatz eingesetzt haben, interessieren wir uns nun für den höchsten Exponenten von x in 3.6, C_0 wird dabei vorerst außer Acht gelassen. Dazu vergleichen wir alle Ausdrücke der Form

$$s + l_i - i, \quad 0 \leq i \leq r,$$

wobei nur die l_i berücksichtigt werden müssen, für die der zugehörige Koeffizient nicht Null ist, d.h. $C_i \neq 0$. Da s nach Voraussetzung fest ist, muß folgendes berechnet werden:

$$V_Z = \max_{0 \leq i \leq r} \{l_i - i\}.$$

Der höchste Exponent von x in 3.6 ist demzufolge $V_Z + s$. Jetzt sucht man sich aus allen Termen, außer dem Absolutglied C_0 , die heraus, die x^{V_Z+s} enthalten:

$$I_Z = \{i \mid l_i - i = V_Z\}.$$

Wieder sind zwei Fälle möglich:

(a) $l \geq V_Z + s, \quad \implies \quad s \leq l - V_Z$

In diesem Fall ergibt sich sofort eine Schranke: $S^* := l - V_Z$ für s , falls diese Differenz größer oder gleich Null und C_0 ungleich Null ist. Sonst setzt man $S^* := 0$.

(b) $l < V_Z + s$

Eine Schranke S^{**} für s bekommt man in diesem Fall dadurch, daß man alle Terme, deren Koeffizienten-Indizes in I_Z enthalten sind, nach Potenzen von x sortiert und dann die Koeffizienten vor x^{V_Z+s} zusammenfaßt und Null setzt. Wir schreiben diese Gleichung zunächst als

$$Q(s) := 0 \tag{3.7}$$

und werden auch im Anschluß an den Algorithmus die Schlußweise angeben, wie diese Relation zustande kommt und wie sie exakt aussieht. Man erhält aus 3.7 eine polynomiale Gleichung für x_1, \dots, x_{m-1} und macht Koeffizientenvergleich. Aus dem gleichen Grund wie in (N2)(b) bekommt man ein Gleichungssystem für s . Von diesem Gleichungssystem suchen wir die größte ganzzahlige Nullstelle. Ist diese Nullstelle echt kleiner als Null, dann setzen wir $S^{**} := 0$, andernfalls bezeichne S^{**} diese Nullstelle.

Die endgültige Gradschranke für den Grad des Zählers Z ist

$$S = \max\{S^*, S^{**}\}.$$

Mit dieser Schranke haben wir nun folgenden Ansatz für Z :

$$Z = \alpha_S x^S + \alpha_{S-1} x^{S-1} + \cdots + \alpha_1 x + \alpha_0$$

mit $\alpha_i = \alpha_i(x_1, \dots, x_{m-1}), \quad 0 \leq i \leq S.$

Dieser Ansatz wird in die Gleichung 3.6 eingesetzt. Koeffizientenvergleich nach verschiedenen Potenzen von x liefert ein System von linearen Gleichungen in $\alpha_0, \dots, \alpha_S$ über $\mathbf{Q}(x_1, \dots, x_{m-1})$. Dieses lineare Gleichungssystem wird z.B. durch Gaußelimination gelöst. Als unabhängige Parameter in der Lösung f bleiben $\alpha_{i_0}, \dots, \alpha_{i_{r-1}}, i_0, \dots, i_{r-1} \in \{0, \dots, S\}$, die frei wählbare rationale Funktionen in der Lösung des Gleichungssystem sind. Die rationale Lösung von 3.2 hat demnach folgendes Aussehen

$$f = \frac{\beta_S x^S + \cdots + \beta_1 x + \beta_0}{a_1^{m_1} a_2^{m_2} \cdots a_t^{m_t}},$$

wobei β_0, \dots, β_S lineare Funktionen in den $\alpha_{i_0}, \dots, \alpha_{i_{r-1}}$ sind.

3.5.3. Rechnungen und Erklärungen zum Algorithmus

Zuerst wollen wir die Relation 3.5 betrachten. Dazu möchte ich ein Lemma zur Struktur der r -ten Ableitung einer rationalen Funktion angeben. Dieses Lemma ist eine Konkretisierung des Lemmas 3.9 und wird auch genauso bewiesen.

Gegeben sei eine rationale Funktion f in Partialbruchdarstellung

$$f = \frac{a}{b^k} + \cdots,$$

die Punkte stehen für weitere Partialbrüche, unter denen bzgl. b nur solche geringerer Ordnung auftreten. Die erste Ableitung sei wieder mit $'$ bezeichnet.

Lemma 3.10 *Unter den obigen Voraussetzungen und mit den festgelegten Bezeichnungen gilt:*

Die r -te Ableitung der Funktion f hat die Gestalt:

$$f^{(r)} = \frac{(-1)^r k \cdots (k+r-1) a b^r}{b^{k+r}} + \cdots,$$

wobei die Punkte für Terme geringerer Ordnung bzgl. b stehen.

Im Schritt (N2)(b) werden nun alle in der Menge I_1 ausgewiesenen Koeffizienten mit den zugehörigen Ableitungen zusammengefaßt. Da nach dem kleinsten Exponenten $V_1 - m_1$ von a_1 gefragt ist, genügt es, die höchsten Terme bzgl. a_1 aus den Ableitungen der gesuchten Funktion zu berücksichtigen. Daraus folgt

$$\sum_{i \in I_1, i \neq 0} \frac{(-1)^i m_1 \cdots (m_1 + i - 1) \left(\frac{\partial}{\partial x} a_1\right)^r}{a_1^{m_1 + V_1}} \hat{p}_i + \left[\frac{\hat{p}_0}{a_1^{m_1 + V_1}} \right] = 0.$$

Das $\hat{}$ auf den p_i , $i \in I_Z$ bedeutet folgendes: $\hat{p}_i = p_i/(a_i^{v_i})$. Der Term mit p_0 steht in eckigen Klammern, weil er nur dann als Summand auftritt, wenn $0 \in I_1$. Polynomdivisionen der einzelnen $(\frac{\partial}{\partial x} a_1)^i \hat{p}_i$ nach a_1 liefern nun Restpolynome R_i . Damit erhält man die Beziehung

$$Q(m_1) = \sum_{i \in I_1, i \neq 0} (-1)^i m_1 \cdots (m_1 + i - 1) R_i + [R_0] = 0.$$

Dies ist die Gleichung 3.5. Die linke Seite dieser Gleichung stellt nun den Zähler des höchsten Terms bzgl. a_1 in der Differentialgleichung dar, d.h. er wird nicht durch andere Terme aus der Differentialgleichung annulliert. Also sortiert man nach Monomen in x und den Parametern x_1, \dots, x_{m-1} und setzt die Koeffizienten Null.

Anhand einer kurzen Rechnung möchte ich nun auch zeigen, wie man zur Gleichung 3.7 gelangt und welche Form sie hat. Gegeben war die Differentialgleichung

$$p_r \frac{\partial^r}{\partial x^r} f + p_{r-1} \frac{\partial^{r-1}}{\partial x^{r-1}} f + \cdots + p_0 f + q = 0$$

und bei der Suche nach rationalen Lösungen f haben wir den Nenner N bereits bestimmt. Wir erhalten bekanntlich eine lineare gewöhnliche Differentialgleichung für Z :

$$C_r \frac{\partial^r}{\partial x^r} Z + \cdots + C_0 Z + C = 0, \quad (3.8)$$

wobei $C_i = C_i(p_r, \dots, p_0, N)$, $0 \leq i \leq r$, $C = C(q, N)$ bekannt sind.

Gesucht sind polynomiale Lösungen für Z .

$$\implies Z := \alpha_s x^s + \alpha_{s-1} x^{s-1} + \cdots + \alpha_0$$

Der größte x -Exponent ist $V_Z + s$. Offensichtlich kann nur $\alpha_s x^s$ zu diesem Exponenten beitragen.

$$\implies \sum_{i \in I_Z, i \neq 0} s \cdots (s - i + 1) \hat{C}_i + [\hat{C}_0] = 0$$

Diese Gleichung entspricht der Gleichung 3.7. Hieraus bekommen wir das S^{**} . Das $\hat{}$ bedeutet wieder folgendes: $\hat{C}_i = C_i/(x^{l_i})$. C_0 steht in eckigen Klammern, da es nur dann erscheint, wenn $0 \in I_Z$. Einsetzen in die Differentialgleichung 3.8 liefert

$$\sum_{i=0}^r C_i (\alpha_s S \cdots (S - i + 1) x^{S-i} + \cdots) + C = 0$$

Nun wird die Gleichung in eine ganzrationale Form überführt. Koeffizientenvergleich nach verschiedenen Potenzen von x liefert ein lineares Gleichungssystem für die α_i über $\mathbf{Q}(x_1, \dots, x_{m-1})$. Die Lösungen sind rationale Funktionen, die in der gesuchten Funktion als frei wählbare Parameter vorkommen.

Der Zähler Z der Lösung f der ursprünglichen Differentialgleichung ist dann linear in diesen Parametern.

3.5.4. Ein Beispiel

An einem linearen partiellen Differentialgleichungssystem in zwei gesuchten Funktionen wollen wir nun den Algorithmus durchführen. Wir haben damit: $\mathbf{u} := (u^1, u^2)$. Diese Funktionen mögen von zwei Variablen abhängen, also: $\mathbf{x} := (x, y)$. Nach den vorigen Ausführungen muß unser System \mathbf{S} mindestens aus vier Gleichungen bestehen, damit sein Lösungsraum endlich-dimensional ist. Wir geben uns nun \mathbf{S} vor und werden dann punktweise den Algorithmus abarbeiten. Da die Rechnungen bei voller Ausführung zu umfangreich werden würden, gebe ich für manche Gleichungen nur das Ergebnis an. In dem Beispiel treten nur Ableitungen erster und zweiter Ordnung auf. Üblicherweise stellt man dann die Ableitungen dadurch dar, daß die Differentiationsvariablen als unterer Index geschrieben werden.

Sei das lineare partielle Differentialgleichungssystem $\mathbf{S} = (S_1, S_2, S_3, S_4)$ gegeben durch:

$$\begin{aligned} y^2(y+2)(x+1)u_y^2 - (x+1)2yu^2 + x^2(y+1)u_x^1 - x(y+1)u^1 &= 0 & S_1 \\ y(y+2)(x+1)u_x^2 + y(y+2)u^2 - x(y+1)u_x^1 + (y+1)u^1 &= 0 & S_2 \\ y(y+1)u_y^1 + u^1 + x(y+1)u_x^1 &= 0 & S_3 \\ (y+1)u_{xx}^1 &= 0 & S_4 \end{aligned}$$

Wir wählen als Ordnung in den Variablen:

$$y > x.$$

Nun zeichnen wir zunächst u^2 als gesuchte Funktion aus.

1. Schritt: Die Ordnung in den gesuchten Funktionen ist sofort klar:

$$u^1 > u^2.$$

Der Standardform-Algorithmus liefert:

$$\begin{aligned} (1+y)u_y^1 + u^1 &= 0 \\ -xu_x^1 + u^1 &= 0 \\ \left. \begin{aligned} y(y+2)u_y^2 - 2u^2 &= 0 \\ -(x+1)u_x^2 - u^2 &= 0 \end{aligned} \right\} SS \end{aligned}$$

2. Schritt: Der zweite Algorithmus wird angewandt, um die letzte Gleichung der Standardform zu lösen. Wir suchen nach einer Lösung $u^2 = Z/N$.

(N) Der einzige Primfaktor des führenden Koeffizienten ist $a = x + 1$. Daraus folgt: $N = a^m$.

(N1) Vielfachheiten: $v_1 = 1$ $v_0 = 0$

$$V = \min\{1 - 1, 0 - 0\} = 0 \quad \text{und} \quad I = \{1, 0\}$$

(N1) (a) $M^* := 0$, da in unserem Beispiel das Absolutglied Null ist.

(b)

$$\begin{aligned} \frac{(-1)ma_x}{a^m}(-1) + \frac{-1}{a^m} &= 0 \\ \implies m = 1 \quad \text{und} \quad M^{**} &= 1 \end{aligned}$$

$$\implies M = \max\{M^*, M^{**}\} = 1$$

$$\implies N = a = x + 1$$

(Z) $u^2 = Z/(x + 1)$

(Z1) Die lineare Differentialgleichung für Z hat nun die Gestalt:

$$-(x + 1)^2 Z_x = 0. \tag{3.9}$$

Ansatz für Z :

$$Z := \alpha_s x^s + \dots + \alpha_0$$

Die α 's hängen hier noch von y ab.

(Z2)

$$V_Z = \max\{2 - 1\} = 1 \quad \text{und} \quad I_Z = \{1\}$$

(a) Da auch in diesem Fall ein Absolutglied nicht vorhanden ist, setzt man $S^* := 0$.

(b)

$$\begin{aligned} -(x^2 + 2x + 1)\alpha_s s x^{s-1} + \dots &= 0 \\ \implies -\alpha_s s x^{s+1} + \dots &= 0 \\ \implies s = 0 \quad \implies S^{**} &= 0 \end{aligned}$$

$$\implies S = \max\{S^*, S^{**}\} = 0$$

$$\implies Z = \alpha_0$$

Einsetzen in 3.9 liefert $0 = 0$, also bleibt $\alpha_0(y)$ als unabhängiger Parameter.

Die rationale Lösung der gewöhnlichen Differentialgleichung lautet (Wir schreiben für α_0 kurz α):

$$u^2 = \frac{\alpha}{x+1}.$$

Da das α noch keine Konstante ist, gehen wir zum 3. Schritt.

3. Schritt: Das Restsystem von SS besteht nur noch aus einer Gleichung. In diese setzen wir die Lösung von u^2 ein. Wir erhalten die folgende Gleichung:

$$y(y+2)\alpha_y - 2\alpha = 0.$$

Dies ist auch wieder eine gewöhnliche Differentialgleichung, deshalb kann der erste Schritt übersprungen werden, und wir lösen sofort diese Gleichung. Die Lösung ist:

$$\alpha = \frac{\beta y}{y+2},$$

dabei ist β eine Konstante. Wir haben demnach für u^2 die folgende Lösung erhalten:

$$u^2 = \frac{\beta y}{(y+2)(x+1)}.$$

Jetzt ändern wir die Ordnung in den gesuchten Funktionen:

$$u^2 > u^1.$$

Die ersten 3 Schritte werden noch einmal in Kurzform angegeben. Der erste Schritt liefert die Standardform:

$$\begin{aligned} y(y+2)u_y^2 - 2u^2 &= 0 \\ (x+1)u_x^2 + u^2 &= 0 \\ (y+1)u_y^1 + u^1 &= 0 \\ xu_x^1 - u^1 &= 0 \quad SS. \end{aligned}$$

Im 2. Schritt bekommen wir die Lösung der letzten Gleichung

$$u^1 = \alpha(y)x.$$

Einsetzen und Koeffizientenvergleich im 3. Schritt führen zu der Gleichung

$$(1+y)\alpha_y + \alpha = 0.$$

Die Lösung dieser Gleichung ist

$$\alpha = \frac{\beta^*}{y+1},$$

wobei β^* nur noch eine Konstante ist. Folgende Lösung haben wir für u^1 bekommen:

$$u^1 = \frac{\beta^* x}{y+1}.$$

4. Schritt: Setzen wir die Lösungen für u^1 und u^2 in eine der Standardformen ein, so sieht man lediglich, daß alle Gleichungen erfüllt sind. Also sind β und β^* frei wählbare Parameter in der Lösung von \mathbf{S} . Das heißt, der Lösungsraum ist zweidimensional.

Diese Aussage bekommt man auch, wenn in einer der Standardformen die parametrischen Ableitungen gezählt werden.

A. Vorstellung der Programme

Erläuterungen

Es sollen nun Anwendung und Funktionsweise der REDUCE-Programme beschrieben werden. Wozu man die einzelnen Prozeduren in den Programmen benutzt und welche Parameter man übergeben muß, ist in den Prozeduren selbst hinter dem Schlüsselwort *comment* vermerkt.

Das Ziel war es, ein Programm zu erstellen, das Polynome f in den Variablen x und y bestimmt, die die Eigenschaft $f \mid Df$ haben, wobei D der lineare partielle Differentialoperator $D = P\partial_x + Q\partial_y$ mit $P, Q \in \mathbb{C}[x, y]$ ist. So entstanden die beiden im Abschnitt 2.6 dargestellten Möglichkeiten. Der ersten Möglichkeit entspricht das Programm **glsystem.red**, der zweiten Möglichkeit das Programm **groebbas.red**.

Beide Programme lassen eine Betrachtung von mehreren Operatoren zu. Es ist dem Nutzer überlassen, ob eine Datei eingelesen werden soll oder die Eingabe von der REDUCE-Oberfläche erfolgt. Die Eingabedatei muß aus zwei Listen bestehen, die jeweils die Koeffizientenpolynome P bzw. Q enthalten.

Im Programm **glsystem.red** wird die Suche nach den Polynomen f mit der Prozedur *oper*(ma, me) gestartet. ma und me sind untere bzw. obere Schranken für den Totalgrad. Das bedeutet, die gesuchten Polynome werden mit einem Totalgrad m angesetzt, der $ma \leq m \leq me$ erfüllt. Desweiteren wird für ein Polynom g ein Ansatz mit Totalgrad $\max\{\text{tdeg } P, \text{tdeg } Q\} - 1$ gemacht. Um solche Ansätze aufzustellen, wird die Prozedur *compoly* genutzt. Die verwendeten Koeffizienten sind für f $alpha(i, j)$ und für g $beta(i, j)$. Auf den Ansatz von f wird mit der Prozedur *opuse* der Operator D angewendet. Das Resultat wird mit fg gleichgesetzt. Koeffizientenvergleich liefert ein Gleichungssystem für die $alpha(i, j)$ und $beta(i, j)$. Dieses Gleichungssystem wird mit der Systemprozedur *groebner* aus dem Paket *groebner* [14], [15] in eine Standardform überführt und gelöst. Die Lösung stellt dann ein System von Bedingungen für die $alpha(i, j)$ und $beta(i, j)$ dar.

Die Prozedur *test0* entfernt nach jeder abgeschlossenen Suche für einen Grad und einen Operator die triviale Lösung aus der Lösungsmenge. Die gesamte Suche zu

einem Operator wird mit der Prozedur *poly* durchgeführt.

Die Ausgabe der Ergebnisse erfolgt im Programm in die Datei **ergebnis**. Dabei muß folgendes beachtet werden: Möchte man zu bereits vorhandenen Ergebnissen neue Lösungen hinzufügen, so muß entweder die Datei **ergebnis** umbenannt werden oder man ändert den im Programm angegebenen Namen der Ausgabedatei. Dieses komplizierte Vorgehen ist notwendig, da man in REDUCE keine Ergebnisse an den Inhalt einer Datei anfügen, sondern vorhandene Inhalt der Ausgabedatei wird überschrieben. Soll die Ausgabe auf den Bildschirm erfolgen, so ist in der Prozedur *oper* vor den Schlüsselworten *out* und *shut* das %-Zeichen zu setzen.

Die gerade angeführten Erläuterungen zur Ausgabe sind in gleicher Weise für die Datei **groebbas.red** gültig.

Das Ergebnis ist eine Liste von Lösungen. Die Lösungen sind Listen von Gleichungen, die die Bedingungen für die Koeffizienten von *f* und *g* darstellen. Ist eine Liste leer, so existiert bis zu diesem Totalgrad nur die triviale Lösung $f = \text{const}$. Ist in einer Liste ein Koeffizient nicht enthalten, so ist dieser frei wählbar in der Lösung und kein weiterer Koeffizient hängt von ihm ab. Ist dagegen ein Koeffizient frei wählbar in der Lösung und ein anderer Koeffizient abhängig von diesem, so setzt REDUCE die Variable *arbcomplex* ein.

Am Ende der Ergebnisdatei werden die Lösungen zu allen eingegebenen Operatoren in einer Liste zusammengefaßt. Die Datei **ergebnis** hat zum Beispiel für den Operator $D = y\partial_x + x\partial_y$ das folgende Aussehen:

```
D = (y)*dx + (x)*dy
Liste der Koeffizienten:{beta(0,0),
  alpha(0,0),
  alpha(0,1),
  alpha(1,0)}
linke Seiten: {beta(0,0)*alpha(0,0),
  beta(0,0)*alpha(0,1) - alpha(1,0),
  beta(0,0)*alpha(1,0) - alpha(0,1)}
Loesung= {{alpha(1,0)=arbcomplex(2),alpha(0,1)=alpha(1,0),
  alpha(0,0)=0,beta(0,0)=1},
  {alpha(1,0)=arbcomplex(1),alpha(0,1)= - alpha(1,0),
  alpha(0,0)=0,beta(0,0)=-1}}
Liste aller Loesungen der Operatoren: {{{alpha(1,0)
  =arbcomplex(2),alpha(0,1)=alpha(1,0),alpha(0,0)=0,
  beta(0,0)=1},
  {alpha(1,0)=arbcomplex(1),alpha(0,1)= - alpha(1,0),
  alpha(0,0)=0,beta(0,0)=-1}}}
```

In der Datei **groebbas.red** sind Prozeduren implementiert, mit denen sich eine Gröbnerbasis des Ideals $\langle f, Df \rangle$ berechnen läßt. Dazu rufe man die Prozedur *oper(m)* auf, wobei m wieder eine Schranke für den Totalgrad der gesuchten Polynome f darstellt. Es wird mit der Prozedur *compoly* ein Ansatz für f vom Totalgrad m gebildet und *opuse* wendet den Operator D auf diesen Ansatz an. Die Prozedur *groepol* berechnet nun zu einem Operator D die Gröbnerbasis des obigen Ideals. Dabei nutzt man wieder die Systemprozedur *groebner*. Wie aus einer Gröbnerbasis das Polynom f errechnet wird, welches $f \mid Df$ erfüllt, ist bereits im Abschnitt 2.6 beschrieben.

Die erhaltenen Gröbnerbasen werden in die Datei **groeberg** ausgegeben. Für den Operator $D = x\partial_x + y\partial_y$ hat die Ausgabedatei folgendes Aussehen:

```
D = (x)*dx + (y)*dy
Polynom 1. Grades: f = u(1,0)*x + u(0,1)*y + u(0,0)
Df = u(1,0)*x + u(0,1)*y
Loesung: {{u(1,0)*x + u(0,1)*y,u(0,0)}}
Liste der Loesungen fuer alle Operatoren:
{{{u(1,0)*x + u(0,1)*y,u(0,0)}}}
```

Für eine Reihe von Beispieloperatoren wurden Lösungen mit den beiden Programmen gefunden. Allerdings reichte häufig auch bei kleinen Totalgraden der Koeffizientenpolynome P und Q die Speicherkapazität nicht aus bzw. war die Laufzeit nicht vertretbar. Dies liegt möglicherweise daran, daß bei wachsendem Totalgrad des Ansatzes für f „große“ algebraische Gleichungssysteme zu lösen sind, da die Anzahl der Koeffizienten im Ansatz quadratisch mit dem Totalgrad wächst.

Quelltexte

Datei gsystem.red

```
load_package groebner;

operator D ;

operator dx ;
  for all p1 let dx(p1) = df(p1,x) ;

operator dy ;
  for all q1 let dy(q1) = df(q1,y) ;
```

```

procedure compoly2(n,b) ;
comment Mit dieser Prozedur wird ein vollstaendiges Polynom in den
      Variablen x und y mit dem Totalgrad n erzeugt. Der Parameter b stellt
      die Bezeichnung fuer die Koeffizienten dar. ;
begin scalar f;
  clear b;
  operator b;
  if n=0 then f:=b(0,0);
  if fixp(n) and (n > 0) then
    begin
      for i:=0 : n do
        for j:=0 : i do f := f +b(i-j,j)*y**j*x**(i-j) ;
    end;
return f;
end ;

procedure opin;
comment Mit Hilfe von opin wird der Operator D eingelesen. Da D die
      Form P*dx+Q*dy haben soll, brauchen nur P und Q
      eingegeben werden.;
begin
  write("Bitte vom Operator D nur P und Q eingeben,
        da D folgende Gestalt hat:
        P*dx + Q*dy, wobei P und Q Polynome in x und y sind!");
  write("P :");
  P:=xread(t);
  write("Q :");
  Q:=xread(t);
  return P;
  return Q;
end;

procedure oplistin(m);
comment Es ist mit dieser Prozedur moeglich, mehrere
      Operatoren einzugeben.;
begin
  pop:={};
  qop:={};
  for i:=1 : m do
    begin
      clear P, Q;
      opin();
    end;
end;

```

```

        pop:=append(pop,{P});
        qop:=append(qop,{Q});
    end;
write(pop,qop);
return pop;
return qop;
end;

procedure oper(ma,me);
comment Mit oper kann fuer eine selbst gewaehlte Anzahl von
        Operatoren nach Polynomen f gesucht werden, die
        f | Df erfuellen, wobei D=R*dx+S*dy. Dabei wird die
        Suche bei Polynomen mit dem Totalgrad ma begonnen. Gibt
        es nur triviale Loesungen, so wird der Totalgrad um 1
        erhoehrt. Die Suche endet bei Polynomen mit Totalgrad me.;
begin scalar zw1, zw2, zw3, losung;
write("Operatoren per Hand eingeben(1) oder Datei einlesen(0)? ");
zw3:=xread(t);
if zw3=1 then
    begin
        write("Wie viele Operatoren sollen eingelesen werden?");
        zw1:=xread(t);
        oplistin(zw1);
    end
    else begin
        in "c:\reduce\meins\eingabe.red";
        zw1:=length(pop);
    end;
losung:={};
out ergebnis;
for i:=1 : zw1 do
    begin
        write("D = (",part(pop,i),")*dx + (",part(qop,i),")*dy");
        zw2:=poly(ma,me,part(pop,i),part(qop,i));
        losung:=append(losung,{zw2});
    end;
write("Liste aller Loesungen der Operatoren: ",losung);
shut ergebnis;
end;

procedure poly(ma, me, R, S) ;
comment Mit dieser Prozedur sucht man nach Polynomen f, die

```

```

    f | Df erfuellen, wobei  $D=R*dx+S*dy$ . Dabei wird die
    Suche bei Polynomen mit dem Totalgrad  $ma$  begonnen. Gibt
    es nur triviale Loesungen, so wird der Totalgrad um 1
    erhoehrt. Die Suche endet bei Polynomen mit Totalgrad  $me$ .;
begin scalar z, zw1, zw2, zw3, zw4, zw5, loesung,
    zw7, zw8, equations, loes;
unknowns:={x,y};
zw7:=fulldeg(R,unknowns);
zw8:=fulldeg(S,unknowns);
if zw7 > zw8
    then zw3:=fulldeg(R,unknowns)-1
    else zw3:=fulldeg(S,unknowns)-1;
g:=compoly2(zw3,beta);
name1:=cof(g,unknowns);
z:=ma;
while z<me+1 do
    begin
        zw1:=compoly2(z,alpha);
        zw2:=opuse(zw1,R,S);
        zw5:=zw1*g-zw2;
        name2:=cof(zw1,unknowns);
        zw6:=append(name1,name2);
        write("Liste der Koeffizienten:",zw6);
        equations:=cof(zw5,unknowns);
        write("linke Seiten: ",equations);
        torder Gradlex ;
        equations:=groebner(equations) ;
        loes:=solve(equations,zw6);
        zw4:=trivial(rest(name2));
        loes:=test0(loes,zw4);
        if loes={} then z:=z+1 else z:=me+1;
    end;
write("Loesung= ",loes);
return loes;
end ;

procedure opuse(Pol, P, Q) ;
comment Anwendung des Operators  $D=P*dx+Q*dy$ 
    auf ein Polynom Pol.;
begin scalar zw1, zw2, zw3;
    zw1:=P*dx(Pol);
    zw3:=Q*dy(Pol);
    zw2:=zw1+zw3;

```

```

    return zw2;
end;

procedure fulldeg(R, variab);
comment Es laesst sich mit fulldeg der Totalgrad eines Polynoms
      R bzgl. der Variablen, die Elemente der Liste variab sind.;
begin scalar grad;
clear c;
operator c;
if (length(variab) neq 0) then
  for i:=1 : length(variab) do R:=sub({part(variab,i)=c(i)*c(0)},R);
grad:=deg(R,c(0));
return grad;
end;

procedure test0(glliste, search);
comment glliste ist eine Liste, deren Elemente Listen
      sind. Es wird getestet, ob jedes Element der Liste
      search in den einzelnen Elementen von glliste enthalten ist.
      Kommen nun alle Elemente von search in einem Element von glliste
      vor, so wird dieses Element aus glliste entfernt.
      Es wird die reduzierte Liste zurueckgegeben.;
begin scalar zw1, zw2, nliste;
  zw1:=glliste;
  if (search neq {}) and (glliste neq {}) then
    for each x in glliste do
      begin
        zw2:=0;
        nliste:=search;
        while (zw2 = 0) and (nliste neq {}) do
          if elem(first(nliste),x) = 0 then zw2:=1
          else nliste:=rest(nliste);
          if zw2=0 then zw1:=rest(zw1)
          else zw1:=append(rest(zw1),{first(zw1)});
        end;
      end;
return zw1;
end;

procedure trivial(neu);
comment Diese Prozedur wandelt die Liste neu in eine
      Liste von Gleichungen pliste um, deren linke Seiten die
      Elemente aus neu und deren rechte Seiten saemtlich Null sind.;

```

```

begin scalar pllist;
  pllist:={};
  if length(nliste) neq 0 then
    for i:=1 : length(neu) do
      pllist:=append(pllist,{part(neu,i) = 0});
return pllist;
end;

procedure elem(y,L);
comment Hier wird getestet, ob ein Element in einer Liste enthalten
      ist. Ist das der Fall, so ist der Rueckgabewert 1, andernfalls 0.;
begin
if L={} then return 0;
if y = first(L) then return 1 else return elem(y,rest(L));
end;

procedure cof(O,unknowns);
comment Mit der Prozedur cof werden die Koeffizienten des Polynoms O
      bzgl. der Variablen, die Elemente der Liste unknowns sind.;
begin scalar z1, z2, name, name1;
name:={0};
z1:=length(unknowns);
for i:=1 : z1 do
  begin
    name1:={};
    z2:=length(name);
    for j:=1 : z2 do
      name1:=append(name1,coeff(part(name,j),part(unknowns,i)));
    name:={};
    for j:=1 : length(name1) do
      if (part(name1,j) neq 0) then name:=append(name,{part(name1,j)});
  end;
return name;
end;

; end ;

```

Datei groebbas.red

```
load_package groebner;
```



```

operator D ;

operator dx ;
  for all p1 let dx(p1) = df(p1,x) ;

operator dy ;
  for all q1 let dy(q1) = df(q1,y) ;

procedure compoly2(n,b) ;
comment Mit dieser Prozedur wird ein vollstaendiges Polynom in den
  Variablen x und y mit dem Totalgrad n erzeugt. Der Parameter b stellt
  die Bezeichnung fuer die Koeffizienten dar. ;
begin scalar f;
  clear b;
  operator b;
  if n=0 then f:=b(0,0);
  if fixp(n) and (n > 0) then
    begin
      for i:=0 : n do
        for j:=0 : i do f := f +b(i-j,j)*y**j*x**(i-j) ;
    end;
return f;
end ;

procedure opin;
comment Mit Hilfe von opin wird der Operator D eingelesen. Da D die
  Form P*dx+Q*dy haben soll, brauchen nur P und Q
  eingegeben werden.;
begin
  write("Bitte vom Operator D nur P und Q eingeben,
    da D folgende Gestalt hat:
    P*dx + Q*dy, wobei P und Q Polynome in x und y sind!");
  write("P :");
  P:=xread(t);
  write("Q :");
  Q:=xread(t);
  return P;
  return Q;
end;

procedure oplistin(m);
comment Es ist mit dieser Prozedur moeglich, mehrere
  Operatoren einzugeben.;

```

```

begin
pop:={};
qop:={};
for i:=1 : m do
  begin
    clear P, Q;
    opin();
    pop:=append(pop,{P});
    qop:=append(qop,{Q});
  end;
write(pop,qop);
return pop;
return qop;
end;

procedure oper(m);
comment Mit oper wird fuer eine selbst gewaehlte Anzahl von
  Operatoren D die Groebnerbasis des Ideals angegeben, das
  von einem Polynom f und Df erzeugt wird, wobei die Operatoren
  D noch eingegeben werden muessen. f wird dabei als vollstaendiges
  Polynom vom Totalgrad m angesetzt. Es besteht auch die
  Moeglichkeit, eine Datei einzulesen, in der Listen von
  Koeffizienten fuer D enthalten sind.;
begin scalar zw1, zw2, zw3, i;
write("Operatoren per Hand eingeben(1) oder Datei einlesen(0)? ");
zw3:=xread(t);
if zw3=1 then
  begin
    write("Wie viele Operatoren sollen eingegeben werden?");
    zw1:=xread(t);
    oplistin(zw1);
  end
else begin
  in "d:\reduce\meins\eingabe.red";
  zw1:=length(pop);
  end;
losung:={};
out groeberg;
for i:=1 : zw1 do
  begin
    write("D = (",part(pop,i),")*dx + (",part(qop,i),")*dy");
    zw2:=groepol(m,part(pop,i),part(qop,i));
    write("Loesung: ",zw2);
  end;
end;

```

```

        losung:=append(losung,{zw2});
        clear v, w;
    end;
write("Liste der Loesungen fuer alle Operatoren: ");
return losung;
shut groeberg;
end;

procedure groepol(m, R, S);
comment Mit dieser Prozedur wird die Groebnerbasis des Ideals <f,Df>
        berechnet. f ist vollstaendiges Polynom vom Grad m und D=R*dx+S*dy.;
begin scalar zw1, zw2, zw3, bas, baslist;
baslist:={};
for i:=1 : m do
    begin
        clear u;
        operator u;
        zw3:={};
        zw1:=compoly2(m,u);
        write("Polynom ",i,". Grades: f = ",zw1);
        zw2:=opuse(zw1, R, S);
        write("Df = ", zw2);
        zw3:=append({x,y},cof(zw1,{x,y}));
        torder Gradlex;
        bas:=groebner({zw1,zw2},zw3);
        baslist:=append(baslist,{bas});
    end;
return baslist;
end;

procedure opuse(Pol, P, Q) ;
comment Anwendung des Operators D=P*dx+Q*dy
        auf ein Polynom Pol.;
begin scalar zw1, zw2, zw3;
    zw1:=P*dx(Pol);
    zw3:=Q*dy(Pol);
    zw2:=zw1+zw3;
    return zw2;
end;

procedure cof(O,unknowns);
comment Mit der Prozedur cof werden die Koeffizienten des Polynoms O
        bzgl. der Variablen, die Elemente der Liste unknowns sind.;

```

```

begin scalar z1, z2, name, name1;
name:={0};
z1:=length(unknowns);
for i:=1 : z1 do
  begin
    name1:={};
    z2:=length(name);
    for j:=1 : z2 do
      name1:=append(name1,coeff(part(name,j),part(unknowns,i)));
    name:={};
    for j:=1 : length(name1) do
      if (part(name1,j) neq 0) then name:=append(name,{part(name1,j)});
    end;
  return name;
end;

; end ;

```

Literaturverzeichnis

- [1] M. J. P. P. P. Singer: *Elementary first integrals of differential equations*.
Transactions of the American Mathematical Society, Volume 279, Number 1, 1983.
- [2] C. Mack: *Integrations of affine forms over elementary functions*.
Computer Science Department, University of Utah, Technical Report, VCP-39, 1976.
- [3] M. Rosenlicht: *On Liouville 's theorie of elementary functions*.
Pacific J. Math. 65, 1976.
- [4] R. H. Risch: *The problem of integration in finite terms*.
Transactions of the American Mathematical Society, Volume 139, 1969.
- [5] J.P. Jouanolou: *Equations de Pfaff algébriques*.
Lecture Notes in Mathematics, Volume 708, Berlin, New York 1979.
- [6] K. O. Geddes, S. R. Czapar, G. Labahn: *Algorithms for Computer Algebra*.
Boston, Dordrecht, London 1995.
- [7] M. Mignotte : *Mathematics for Computer Algebra*.
New York, Berlin, Heidelberg 1992.
- [8] G. Czichowski: *Vorlesung: Computeralgebra und Differentialgleichungen*.
Greifswald, WS 1993/94.
- [9] M. Thiede: *LIE-Methoden für gewöhnliche Differentialgleichungen - zur Symmetrieberechnung*.
Dissertation, Greifswald 1992.
- [10] M. Berth: *Invarianten von Differentialgleichungen und ihre Berechnung mit Mitteln der Computeralgebra*.
Diplomarbeit, Greifswald 1995.

- [11] I. N. Bronstein, K. A. Semendjajew: *Taschenbuch der Mathematik*.
Moskau, Leipzig 1987.
- [12] M. Artin: *Algebra*.
Basel, Boston, Berlin 1993.
- [13] H. Stephani: *Differentialgleichungen Symmetrien und Lösungsmethoden*.
Heidelberg, Berlin, Oxford 1994
- [14] J. Ueberberg: *Einführung in die Computeralgebra mit REDUCE*.
Mannheim, Leipzig, Wien, Zürich 1992.
- [15] M. MacCallum, F. Wright: *Algebraic Computing With REDUCE*.
Oxford 1991.