Nichtlineare Dynamische Systeme: Eine Fallstudie für die Anwendung von Computeralgebramethoden

Benno Fuchssteiner

1 Vorbemerkung

Ich bin mir als „Reiner Mathematiker" der potentiell kulturzerstörerischen Wirkung der *Computerisierung* wohl bewußt. Trotzdem führe ich seit einiger Zeit praktische Arbeiten auf dem Gebiet der Computeralgebra durch, und habe als Ziel meiner Gruppe sogar die Neuentwicklung eines parallelverarbeitenden Computeralgebrasystems vor Augen.

Wieso kommt also ein Reiner Mathematiker auf einmal dazu, sich für Hardware, Software, Oberfächen und Interfaces zu interessieren, und einen Teil seiner Zeit mit der praktischen Implementierung von Algorithmen zu verbringen?

Nun dies hat - wie sollte es auch anders sein - mathematische Gründe. Diese Gründe haben mich dazu gebracht, zu glauben, daß die Einbeziehung solcher Hilfsmittel in Zukunft für fast alle Mathematiker einerseits eine Notwendigkeit sein wird, andererseits aber auch große Chancen eröffnen kann.

Ich möchte sowohl diese Notwendigkeit als auch die damit einhergehenden Chancen am Beispiel der Strukturtheorie nichtlinearer vollständig-integrabler Systeme andeuten. Im Folgenden wähle ich zur Illustration, als typisches Exemplar für ein lösbares nichtlineares System, die Korteweg-de Vries-Gleichung. Diese Wahl erfolgt aber nur deshalb, weil dies ein weithin bekanntes Beispiel ist; ich könnte ebenso gut auch ein anderes vollständig-integrables nichtlineares System aus dem Bereich der Integrodifferentialgleichungen oder auch der quantenmechanischen Spinketten nehmen.

Im ersten Teil des Aufsatzes erläutere ich die spezielle geometrische[1] Struktur dieses Systems. Im zweiten Teil versuche ich aus den, bei der Behandlung solcher Systeme gemachten, Erfahrungen Rückschlüsse für den Einsatz und die Entwicklung von Computeralgebra zu ziehen.

2 Struktur eines speziellen nichtlinearen Systems

Wir betrachten das folgende

Problem 2.1: *Löse die nichtlineare partielle Differentialgleichung*

$$u_t = u_{xxx} + 6uu_x \tag{2.1}$$

zu vorgegebenen Anfangswerten $u(x, t = 0)$.

[1] Für den Hang auch analytische Probleme unter geometrischen Gesichtspunkten zu betrachten, bin ich meinem Lehrer D. Laugwitz dankbar.

Solch ein Problem ist im allgemeinen schwer zu lösen, insbesondere wenn es sich, wie hier, um ein nichtlineares System handelt. Wir wollen Abstand davon nehmen, das Problem numerisch-approximativ zu lösen. Denn Approximationen, die numerischen Lösungen zugrunde liegen, könnnen häufig die intrinsische Harmonie und Schönheit eines nichtlinearen Systems zerstören [2]. Natürlich lehne ich approximative Lösungen nicht vollkommen ab, sondern ich plädiere nur dafür, daß man sie unter Einbeziehung struktureller Aspekte betrachtet; dies auch deshalb, weil man dadurch zu besseren Approximationen kommt.

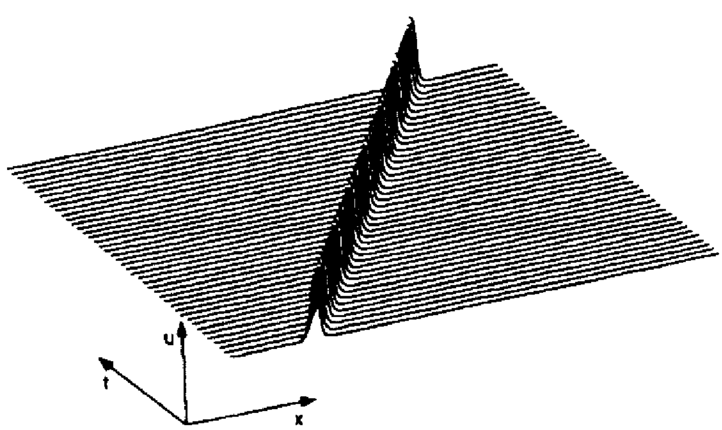
Die Besonderheiten der KdV erkennt man in der Tat recht einfach durch Betrachtung einiger spezieller Lösungen. Die Lösung zum Ansatz

$$u(x,t) = s_c(x - ct) \tag{2.2}$$

läßt sich leicht ausrechnen, da man dadurch auf eine gewöhnliche Differentialgleichung geführt wird. Unter der Annahme des Verschwindens im Unendlichen ist die explizite Form dieser Lösung dann

$$s_c(x + ct) = \frac{c}{2}\cosh^{-2}\left\{\frac{\sqrt{c}}{2}(x - x_0 + ct)\right\}. \tag{2.3}$$

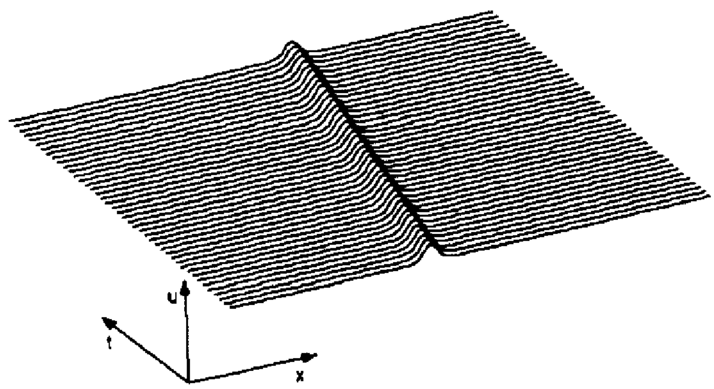
Wir nennen dies eine *Single-Soliton*-Lösung und erhalten im Fall $c = 1.2$ für die t-slices dieser Lösung folgendes Bild:



Figur 1: Single-Soliton der KdV für $c = 1.2$

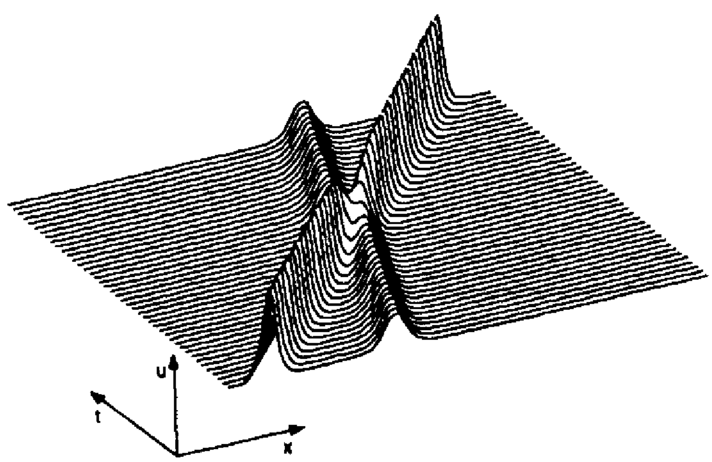
Wenn man nun für große t asymptotisch zwei solche Wellenberge vorgibt, also noch zusätzlich eine Lösung der Form

[2] Ein gutes Beispiel für diese zerstörerische Wirkung von Approximationen liefert das physikalische Pendel. Hätte man da nicht immer die Approximation betrachtet, die für kleine Auslenkungen durch das sogenannte „Mathematische Pendel“ gegeben ist, so hätte man die Theorie doppelt-periodischer Funktionen sicher eher entdeckt. Denn offensichtlich ist die Existenz doppelt-periodischer Lösungen physikalisch ganz einfach einzusehen, da das „Auf-den-Kopf-stellen“ des Pendels einem Übergang zu rein imaginärer Zeit entspricht, und damit die Periodizität entlang der imaginären Achse impliziert.



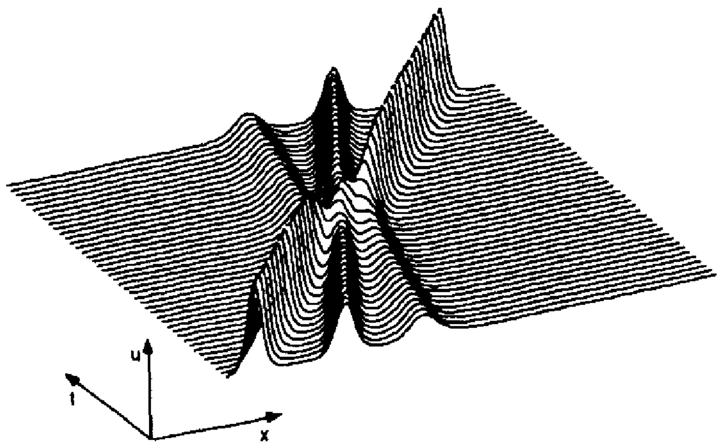
Figur 2: Single-Soliton der KdV für $c = 0.6$

hinzufügt, so gelangt man über einfache numerische Studien zur folgenden Lösung:



Figur 3: Zwei-Soliton Lösung der KdV

Für große Zeiten zerfällt diese Lösung also in eine Superposition von Lösungen der Form (3.5). Oder anders ausgedrückt: Die Wellen verhalten sich ähnlich wie bei einer elastischen Streuung. Für große Zeiten besteht also der einzige Einfluß der nichtlinearen Wechselwirkung in einer Phasenverschiebung der Single-Solitons. Man kann dieses Phänomen nun, unter vernünftigen Bedingungen an die Erhaltung der, von der Gesamtheit der Solitonen, getragenen Energie, für die Wechselwirkung einer größeren Anzahl von Wellenbergen verallgemeinern. Zum Beispiel würde man bei der Superposition dreier Wellenberge die folgende Lösung erhalten:



Figur 4: Drei-Soliton Lösung der KdV

Was uns an diesen Beobachtungen interessiert, ist die mögliche strukturelle Erkenntnis. Um diese zu gewinnen, betrachten wir die Mannigfaltigkeit aller Lösungen dieser Art, also alle Lösungen für die asymptotisch gilt:

$$u(x,t) \simeq \sum_{i=1}^{N} s_i(x + c_i t + q_i^{\pm}) \quad \text{für } t \to \pm\infty \tag{2.4}$$

wobei die c_i die verschiedenen Geschwindigkeiten der asymptotisch auftretenden Wellen sind, und die $q_i^{\pm}$ die entsprechenden Phasen beschreiben. Offensichtlich ergibt die Gesamtheit der c_i und $q_i = q_i^+$ eine 2N-dimensionale Mannigfaltigkeit. Und diese können wir als Parameterraum für unsere vorher betrachtete Mannigfaltigkeit nehmen indem wir setzen:

$$c_i = c_i(u(x,0)), \quad q_i = q_i(u(x,0)) \ , \tag{2.5}$$

indem wir also der Anfangsbedingung $u(x,0)$ als Koordinaten die Daten zuordnen, die wir durch asymptotische Auswertung erhalten. Den Fluß selbst erhalten wir in den neuen Parametern durch Betrachtung der Dynamik dieser Daten, welche wir mit der in (3.4) eingeführten Funktion leicht über die Definition

$$c_i(t) = c_i(u(x,t)), \quad q_i(t) = q_i(u(x,t)) \tag{2.6}$$

erhalten können. Die Größen $q_i(t), c_i(t)$ sind also die durch (2.5) gegebenen Daten, die man erhält, wenn man $u(x,t)$ an Stelle von $u(0,t)$ als Anfangsbedingung verwendet. Die explizite Form dieser Dynamik ist nun leicht gefunden, wenn man bedenkt, daß sich die c_i überhaupt nicht ändern, und daß, per Definitionem die Phasen durch eine einfache Verschiebung des Ursprungs der t-Achse als

$$q_i^{\pm}(u(x,t)) = q_i^{\pm}(u(x,0)) + c_i t \ , \tag{2.7}$$

bestimmt werden können. Also wachsen die Phasen linear mit der Zeit t, und unser Fluß hat in den neuen Koordinaten die einfache Darstellung

$$\frac{d}{dt}\begin{pmatrix} c_1 \\ c_2 \\ q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ +1 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 & -c_1 & 0 \\ 0 & 0 & 0 & -c_2 \\ c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.8}$$

Hier haben wir, der Einfachheit halber, den Fall $N = 2$ gewählt. Wir bemerken, daß dies ein linerares System ist, außerdem ist es ein Hamiltonsystem, denn auf der rechten Seite steht ein Gradient

$$\begin{pmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{2}\mathrm{grad}(c_1^2 + c_1^2). \tag{2.9}$$

Dies ist aber nicht die einzige Hamiltonformulierung, denn auch der Vektor in der zweiten Zeile ist ein Gradient

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \mathrm{grad}(c_1 + c_1). \tag{2.10}$$

Da außerdem die Matrix, die vor diesem Gradienten steht, eine symplektische Form definiert, haben wir damit eine weitere Hamiltonformulierung. Im allgemeinen Fall haben wir also

$$\frac{d}{dt}\begin{pmatrix} c_1 \\ \vdots \\ c_N \\ q_1 \\ \vdots \\ q_N \end{pmatrix} = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}\frac{1}{2}\mathrm{grad}(c_1^2 + \ldots + c_N^2)$$
$$= \begin{pmatrix} 0 & -\Lambda \\ \Lambda & 0 \end{pmatrix}\mathrm{grad}(c_1 + \ldots + c_N)\,, \tag{2.11}$$

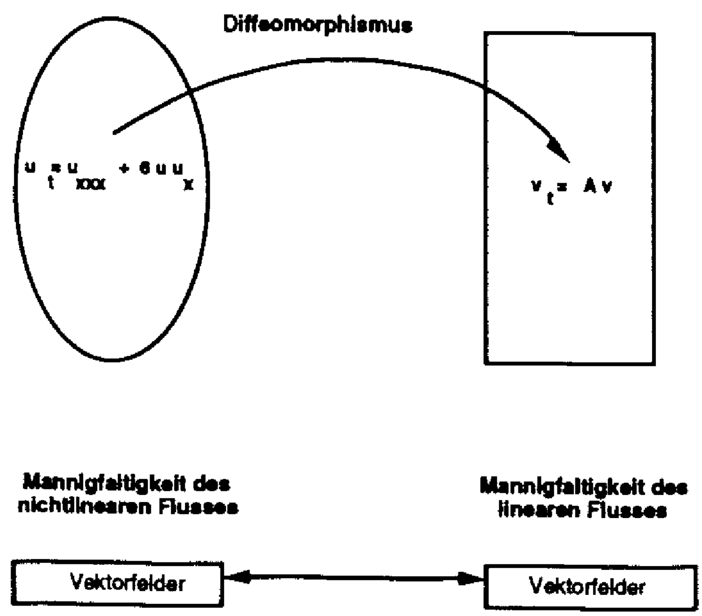
wobei I die $N \times N$-Einheitsmatrix und Λ die Matrix bezeichnet, welche in der Diagonalen folgende Einträge hat

$$\Lambda = \begin{pmatrix} c_1 & 0 & .. & 0 \\ 0 & c_2 & .. & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & .. & 0 & c_N \end{pmatrix}. \tag{2.12}$$

Die gewählte Parametrisierung ist übrigens eine sogenannte ***Wirkungs- und Winkelvariablen-Darstellung*** [1]. Vom mathematischen Standpunkt aus ist unser Anfangswertproblem gelöst, und wir halten unsere strukturelle Einsicht in der folgenden Figur 3 fest. Natürlich ist diese ganze Betrachtung nur ein Trick, und gelöst haben wir, zumindest vom ***praktischen*** Standpunkt aus, gar nichts, denn die eingeführten neuen Koordinaten (in Abhängigkeit vom Anfangswert) sind gerade das, was der Praktiker berechnen möchte. Er will also den Diffeomorphismus aus Figur 3 berechnen, und den können wir ihm jetzt noch nicht geben. Wir haben die Betrachtung eigentlich auch nur deshalb durchgeführt, um das Ziel der folgenden Rechnungen abstrakt formulieren zu können.

Problem 2.2: *Für die speziellen Lösungen (2.4), welche durch Wechselwirkung beliebig vieler Wellenberge des vorgegebenen Systems entstehen, wird ein vollständiger Satz* c_i, q_i; $i = 1, 2, 3, \ldots$ *von Koordinaten mit den folgenden Eigenschaften gesucht:*

- *der Fluß läuft entlang der Koordinatenlinien* $c_i = const$,
- *die Koordinaten* q_i *wachsen linear mit dem Fluß.*



Figur 3: Linearisierung

3 Symmetrien

Die in Problem 2.2 geforderte Koordinatisierung der Dynamik ist deshalb besonders geeignet, weil in diesem neuen Koordinatensystem die Lösung recht einfach wird. Ein Standardbeispiel für so eine Koordinatisierung ist durch die Einführung der Polarkoordinaten beim harmonischen Oszillator gegeben. Man nennt solche Koordinaten, in Anlehnung an dieses Beispiel, die *Wirkungs- und Winkelvariablen* des Systems. Bei nichtlinearen Systemen sollte man sich darüber im klaren sein, daß nur speziell strukturierte Systeme eine solche Koordinatisierung zulassen. Die Auffindung und Klassifizierung solcher Systeme ist eine der Hauptaufgaben des Gebiets. Die theoretische Durchdringung des Problems wird durch die Beobachtung erleichtert, daß die Verschiebung von Lösungskurven (*orbits*) um einen festen Koordinatenwert (einer beliebigen Koordinate) wieder eine Lösungskurve ergibt.

Beobachtung 3.1: *Die Winkel- und Wirkungsvariablen des Systems ergeben auf natürliche Weise eine kontinuierliche Symmetriegruppe des Systems. Zumindest der* *zu den c_i gehörende Teil dieser Symmetriegruppe muß abelsch sein.*

Diese Beobachtung gibt uns einen erheblichen methodischen Vorteil, da kontinuierliche Symmetriegruppen eine infinitesimale Beschreibung (durch die sogenannte Vektorfeld-Liealgebra) erlauben. Wenn wir also bisher globale Größen gesucht haben, so können wir uns nach dieser Umformulierung auf das Auffinden von lokalen Größen beschränken. Das Problem ist dadurch viel einfacher geworden, weil wir (einstweilen) nur noch die *infinitesimalen Generatoren* einer entsprechend großen Symmetriegruppe zu suchen haben:

Problem 3.2: *Man betrachte die rechte Seite von (2.1)*

$$G(u) = u_{xxx} + 6uu_x$$

als Vektorfeld und suche andere Vektorfelder K, welche damit in der Vektorfeld-Liealgebra vertauschen, die also erfüllen:

$$[K, G] := G'[K] - K'[G] = 0 \qquad (3.1)\,.$$

Dabei bezeichnen die $K'[G]$ die Variationsableitungen von K in Richtung von G:

$$K'[G] \ = \frac{\partial}{\partial\varepsilon}_{|\,\varepsilon=0} K(u + \varepsilon G(u)) \quad .$$

Man sollte bemerken, daß diese Suche keinesfalls ein einfaches Problem ist, denn schon bei gegebenem K ist die Verifikation der Eigenschaft (3.1) recht mühsam. Dies sieht man sofort, wenn man sich den zweit-einfachsten Symmetriegruppengenerator

$$\begin{aligned} K_3(u) &= u_{xxxxxxx} + 14u_{xxxxx}u + 42u_xu_{xxxx} \\ &+ 70u_{xxx}(u_{xx} + u^2) + 280u_{xx}u_xu + 70u_x(u_x^2 + u^3) \end{aligned} \qquad (3.2)$$

der KdV ansieht. Der ungeübte Leser wird wohl einige Zeit brauchen, um zu sehen, daß die Beziehung (3.1) erfüllt ist.

Um Symmetrien zu finden, betrachten wir zuerst die Gleichung (2.10). Da die Größe $[K, G]$ differentialgeometrisch invariant ist, also nicht vom gewählten Koordinatensystem abhängt, sind die Symmetriegeneratoren hierfür leicht gefunden, es sind die Vektorfelder der Form

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ c_1^m \\ \vdots \\ c_N^m \end{pmatrix} . \qquad (3.3)$$

Diese Felder werden rekursiv erzeugt durch Anwendung des Operators

$$\Psi = \begin{pmatrix} \Lambda & 0 \\ 0 & \Lambda \end{pmatrix} , \qquad (3.4)$$

welcher die Matrix Λ (2.12) als Eingänge hat. Man sollte bemerken, daß dieser Operator Quotient der beiden Operatoren ist, die in den Hamiltonformulierungen des Systems auftreten. Dies ist kein Zufall sondern eine sehr allgemeine Schlußfolgerung aus der bi-hamiltonischen Formulierung des Systems, da durch jede Hamiltonformulierung ein Übergang von invarianten Kovektorfeldern zu invarianten Vektorfeldern gegeben wird. Liegen also verschiedene Hamiltonformulierungen vor, so erhält man durch Quotientenbildung eine Abbildung, welche einen Übergang von invarianten Vektorfeldern zu anderen invarianten Vektorfeldern erlaubt. Durch die Matrix Ψ ist also ein ***Rekursionsoperator*** zur Erzeugung von Symmetriegruppengeneratoren gegeben. Wir müssen diese Matrix jetzt nur noch auf die linke Seite von Figur 3 hinüberziehen, um die Symmetriegeneratoren zu finden. Dies stößt natürlich auf Schwierigkeiten, weil wir den für diesen „pull-back" verantwortlichen Operator gar nicht kennen. Wir betrachten deshalb einstweilen nur die ***Struktur*** solcher Operatoren. Die wichtigste Eigenschaft scheint zu sein, daß es sich um einen Diagonaloperator handelt. Wenn wir dies in Bezug auf den jeweiligen Mannigfaltigkeitspunkt ausdrücken, so bedeutet dies, daß für die Variationsableitung des Tensors in beliebiger Richtung B gelten muß

$$\Psi\Psi'[B] = \Psi'[\Psi B]\,. \tag{3.5}$$

Diese Eigenschaft läßt sich immer noch nicht so einfach auf die linke Seite von Figur 3 rüberziehen, da ja die Variationsableitung keine ***differentialgeometrisch invariante*** Konstruktion ist. Um zu einer solch invarianten Konstruktion zu kommen, müssen wir alles durch entsprechende ***Lieableitungen*** ausdrücken (Siehe [3] oder [5] für eine Einführung in Lie Ableitungen.). Dies führt zu folgender

Beobachtung 3.3: *Das (1,1)-Tensor Feld A, welches durch (3.4) gegeben ist, erfüllt*

$$\Psi L_K(\Psi) = L_{\Psi K}(\Psi) \tag{3.6}$$

für alle Vektorfelder K. Hierbei bezeichnet L_K die Lieableitung in Richtung des Vektorfeldes K.

Dies ist nun in der Tat eine differentialgeometrisch invariante Formulierung. Wir nennen Operatoren mit dieser Eigenschaft *hereditäre Operatoren.* Die so gewonnene Eigenschaft ist genau das, was wir suchen:

Satz 3.4: *Sei Φ ein hereditärer Operator, der in Bezug auf das Vektorfeld K invariant ist, für den also gilt*

$$L_K\Phi = 0\,. \tag{3.7}$$

Dann bilden die Vektorfelder

$$K, \Phi K, \Phi^2 K, \Phi^3 K, \ldots \tag{3.8}$$

die Basis einer abelschen Liealgebra.

Beweis: Sei $K_n = \Phi^n K$. Wir betrachten den Kommutator $[K_n, K_m]$, der ja mit der Lieableitung $L_{K_n}K_m$ übereinstimmt. Mit der Eigenschaft der Hereditarität und der Produktregel für Lieableitungen erhalten wir nun leicht aus der Invarianz von Φ bezüglich K

$$\begin{aligned}
[K_n, K_m] &= L_{K_n}(\Phi^m K) = (L_{\Phi^n K}\Phi^m)K + \Phi^m(L_{K_n}K) \\
&= \Phi^n(L_K\Phi^m)K - \Phi^m L_K(\Phi^n K) \\
&= \Phi^{n+m}L_K K = \Phi^{n+m}[K,K] = 0\,.
\end{aligned} \tag{3.9}$$

Wir brauchen also nur noch einen geeigneten hereditären Operator Φ zu raten, um einen Satz von ablschen Symmetriegruppengeneratoren rekursiv zu erzeugen. Daß dies ein recht effizientes Verfahren ist, sehen wir an folgendem

Beispiel 3.5: D^{-1} bezeichne die Integration von $-\infty$ bis x. Durch direkte Rechnung zeigt man, daß der Operator

$$\Phi(u) = D^2 + 2DuD^{-1} + 2u \tag{3.10}$$

die hereditäre Eigenschaft hat. Da er nicht explizit von x abhängt, muß er invariant in Bezug auf das Vektorfeld $K(u) = u_x$ sein. Also kommutieren alle Vektorfelder, die nach der Vorschrift des Satzes gebildet werden. Man sollte beachten, daß das erste dieser Felder die rechte Seite der KdV ist. Damit haben wir durch rekursive Anwendung von Φ unendlich viele Symmetriegruppengeneratoren dieser Gleichung erzeugt. Problem 3.2 ist also glöst. Mit Hilfe der Hamiltonformulierung, und etwas Einfallsreichtum, kann man das Problem 2.2 ebenfalls lösen.

Wir wollen nun die Strukturtheorie nichtlinearer Systeme verlassen und uns damit beschäftigen, welche Möglichkeiten Computeralgebra beim Umgang mit den eingeführten Strukturen eröffnet. Der Leser, der sich für diesen Themenkreis weiter interessiert, sei auf die Überblicksarbeiten [4], [5] und [6] verwiesen.

4 Computeralgebra

Computeralgebrasysteme erlauben das interaktive formelmäßige Rechnen mit mathematischen Objekten, wie sie etwa in der täglichen Arbeit eines Ingenieurs oder Physikers mit anspruchsvollem Hintergrund vorkommen. Im Unterschied zur numerischen Behandlung mathematischer Sachverhalte, manipuliert Computeralgebra Zeichen und Symbole. Eine ganz natürliche Sache, da Computer zu allem eher geeignet sind, als zum Rechnen mit reellen Zahlen, denn eine beliebige reelle Zahl ist ja bekanntermaßen ein außerordentlich kompliziertes Gebilde, wohingehen ein Symbol, wie etwa der Buchstabe π, ein sehr einfach strukturiertes Objekt ist. Daß Computeralgebra möglich sein würde, ist schon recht lange bekannt. Ada Augusta, Countess of Lovelace, die, den romantischen Schwärmereien des Vaters abhold, eher den scharfen analytischen Verstand ihrer Mutter geerbt hatte, schreibt etwa im Jahre 1842

Many persons ... imagine that the business of the engine [Babbage's engine] is to give results in numerical notation, the nature of its processes must consequently be arithmetical and numerical rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or other general symbols; and in fact it might bring out its results in algebraical notation were provisions made accordingly (zitiert nach [9, p. 10]).

Computeralgebrasysteme erfüllen die Erfordernisse einer deklarativen[3] Hochsprache mit moderner Syntax zum Programmieren anspruchsvoller mathematischer Sachverhalte. Um einen anschaulichen Einblick in die Möglichkeiten solcher Systeme zu

[3] Deklarative Programmiersprachen spezifizieren die Eigenschaften des gewünschten Resultats und nicht, wie man es bei imperativen Sprachen gewohnt ist, den Weg wie man zu dem Resultat gelangt.

geben, hier einige *sehr einfache* Beispiele[4]:

- Zur Berechnung der 800-stelligen Zahl 341! braucht man genau diesen Befehl und etwa null Sekunden Rechenzeit.
- Die Zerlegung der 34-stelligen Zahl (31!+1) in ihre vier Primfaktoren braucht weniger als zwei Sekunden und als Befehl *ifactor(31!+1)* .
- Durch Eingabe von

  $$dsolve(diff(\varphi(t), t, t) + sin(\varphi(t)) = 0, \varphi(t))$$

  erhält man nach etwa einer Sekunde die explizite Lösung des Pendels mit nichtlinearer Rückstellkraft (physikalisches Pendel). Hat man noch etwas Geduld, so kann man gleich ein Plot der Bahnen dieses nichtlinearen Systems in druckfertiger Form mitnehmen:

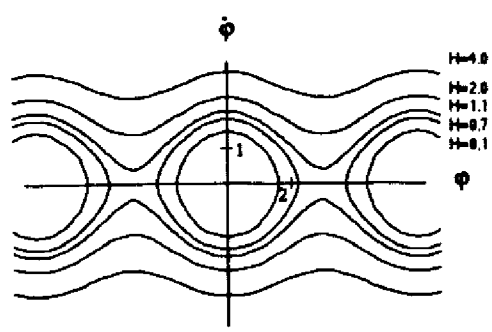


Fig. 4: Phasenraumbahnen des Pendels

- Ein lauffähiges Maple-Programm für die Berechnung der Fibonacci Zahlen beträgt etwas weniger als 1 Zeile.
- Für die exakte Berechnung der ersten 50 Taylorkoeffizienten von

  $$\sin(3x^2 + e^{\sqrt{1+6x^2}})$$

  benötigt man einen halbzeiligen Befehl und etwa null Sekunden.
- Die Integration von

  $$\frac{\sin(x) - 2\cos(x)}{\cos(x) - 3\sin(x)}$$

  braucht nach halbzeiligem Befehl ca. zwei Sekunden.

Bewußt wurden nur einfache, ja triviale Beispiele angegeben, doch auch diese Beispiele verdeutlichen das vorhandene Potential.

Es wird klar, daß das, was manchem als komplizierte mathematische Fragestellung erscheinen mag, durch algorithmische Analyse so umgewandelt wird, daß eine Maschine die Lösung finden kann. Dies demonstriert die grundsätzliche Stärke und Vitalität der Mathematik, welche die eigenen Fragestellungen so lange analysiert, bis diese so schematisiert sind, daß sie auch von Anwendern mit geringem mathematischen Verständnis, oder sogar von Automaten beantwortet werden können. Wer darin

[4]Gerechnet mit MAPLE auf einem Macintosh IIfx

allerdings die Gefahr sieht, daß die Computerisierung eines Tages die Mathemtik überfüssig machen könnte,[5] muß schon einer sehr oberflächlichen Betrachtungsweise fähig sein. Das Gegenteil ist der Fall: Die Trivialisierung von Problemen und die Automatisierung von Routinefragen ist ein immerwährender Prozess in der Entwicklung der Mathematik, der uns und anderen das Feld freimacht, um tatkräftig Neuland zu betreten, und dort neue Probleme anzufassen.

Dieser Prozess vollzieht sich nicht immer mit gleicher Geschwindigkeit. Erfreulicherweise gibt es mitunter dramatische Sprünge und Schübe. So zum Beispiel bei der Erfindung des Differentialkalküls, bei dem Fragestellungen, welche in Spezialfällen von Archimedes bis Fermat tiefe Gedanken erforderten, so automatisiert wurden, daß sie heute von jedem mäßig begabten Anwender oder Mathematiker erfolgreich im Handumdrehen gelöst werden können.

Meiner Meinung nach ist es nicht ausgemacht, ob wir nicht durch die Verfügbarkeit von Computeralgebra vor einem neuen Sprung dieser Art stehen. Denn war bisher in unserem Jahrhundert die mathematische Forschung geprägt durch *„ein weg vom Quantitativen und hin zum Qualitativen“* [12, p. 1], so können wir jetzt daran gehen, das Quantitative wieder einzubeziehen, um zu einer neuen Qualität zu gelangen.

Mathematische Beispiele neuer Größenordnung werden uns erschlossen, dadurch daß wir komplizierte Berechnungen mit großem Formelaufwand scheinbar mühelos durchführen können. Eine neue Disziplin *„Experimentelle Mathematik“* tut sich auf, neu nicht in der Idee sondern in der Dimension, denn große experimentelle Mathematiker hat es immer gegeben. Eine eindrucksvolle Sammlung dessen, was die Vielfalt von experimenteller Mathematik ausmachen kann, hat Michiel Hazewinkel zusammengestellt ([8] und [7]).

Selbst wenn man dieser Bewertung nicht zustimmt, so muß man doch einräumen, daß Computeralgebra Auswirkungen darauf haben wird, wie Anwender in Zukunft mit mathematischen Formeln arbeiten und umgehen werden. Computeralgebrasysteme sind heute schon so mächtig[6], daß jeder, der zum erstenmal mit diesen Werkzeugen zu tun hat, R. Pavelle et al. zustimmen wird, wenn diese schreiben:

These programs do in a few brief minutes virtually all mathematics that most engineers and scientists know. [11]

Eine Bewertung, die manchen Beobachter zu einer Überbewertung der zukünftigen Bedeutung von Mathematik verleiten mag:

Mathematics is a basis of technological progress, and technological progress is a key for international competetivness. Automating an important part of the mathematical problem-solving progress is a key technology for a nation that wishes to control, structure and accelerate technological progress. The automation of the solution of mathematical problems is a powerful lever by which human productivity and expertise can be amplified many times. [9, p. 16]

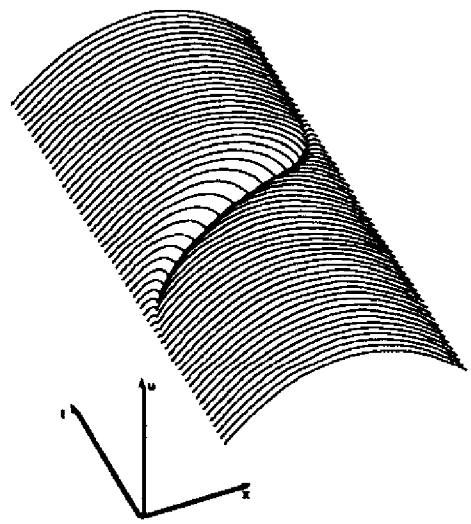
Ich möchte dieses Kapitel aber nicht mit dem Anschein von übermäßiger Zufriedenheit schließen. Einerseits werden, wie so oft, durch sich ankündigenden Fortschritt liebgewonnene und wertvolle Gedanken und Methoden verschüttet, und eine neue Oberflächlichkeit droht Einzug zu halten. Zum anderen muß kritisch bemerkt werden, daß Computeralgebra noch einen weiten Weg gehen muß, um dem anspruchsvollen und kritischen Mathematiker beim Aufbruch zu neuen Horizonten zu helfen.

[5] Ein Gedankengang, dem die öffentliche Meinung allerdings ständig erliegt.

[6] Für einen Überblick über bestehende Systeme vergleiche man [10]

4.1 Über den Nutzen von Computeralgebra: Ein Beispiel

Um ein überzeugendes Beispiel dafür zu präsentieren, daß Computeralgebra wirklich den Aufbruch zu neuen Horizonten erlauben mag, zeige ich hier eine Lösung

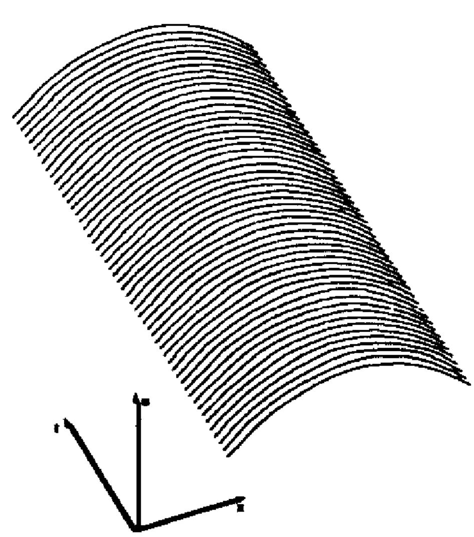


Figur 5: Zwei-Soliton der Harry-Dym-Gleichung

der sogenannten Harry-Dym-Gleichung:

$$u_t = u^3 u_{xxx} \ . \tag{4.1}$$

Die Lösung selbst ist für Physiker interessant wegen der charakteristischen *Delle*. Das Besondere der dargestellten Lösung dieser nichtlinearen partiellen Differentialgleichung fällt einem auf, wenn man eine zweite Lösung daneben betrachtet:



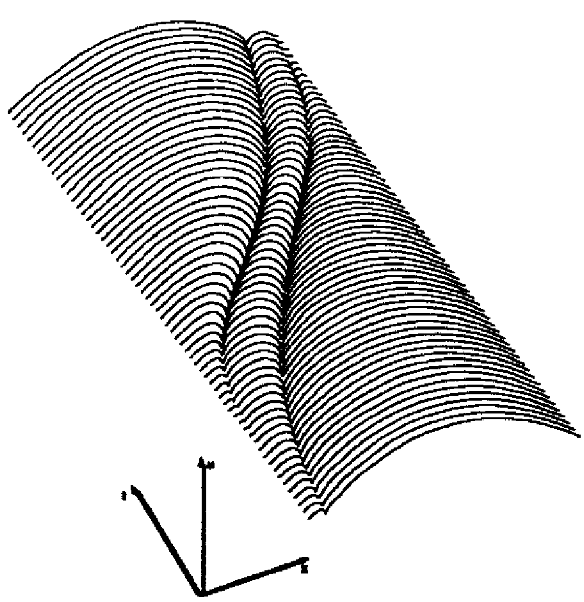
Figur 6: Ein-Soliton der Harry-Dym-Gleichung

Beim Vergleich beider Plots zeigt sich, daß die charakteristische Delle in der ersten Lösung mit der Zeit t mindestens exponentiell schnell verschwindet. Konsequenz davon ist, daß sehr kleine, nahezu *infinitesimale* Störungen des Anfangswertes zu diesem charakteristischen Phänomen führen können. Eine numerische Lösung des Anfangswertproblems ist deshalb wegen dieser Instabilität unmöglich, dies obwohl das System keineswegs chaotisch ist. Daher kann man diesen charakteristischen Effekt nur durch die Angabe der expliziten Lösungen herausfinden. In der Tat, dies ist auch der Weg, auf dem die angegebenen Lösungen, durch Verwendung der im zweiten Kapitel geschilderten Methoden, gefunden wurden.

Im Prinzip besteht Angabe solcher expliziten Lösung aus einer Reihe einfacher Schritte:

1. Finde ein *reflektionsfreies* Potential des klassischen eindimensionalen Schrödinger-Operators.
2. Bestimme explizit das Quadrat des *Eigenvektors* des Operators mit diesem Potential.
3. Verwende die gefundene Funktion als Anfangsbedingung eines gewissen nichtlinearen, vollständig-integrablen, unendlichdimensionalen Systems, der sogenannten Singularitäten-Mannigfaltigkeitsgleichung der KdV.
4. Führe mit der so erhaltenen Lösung eine Art *Hodographtransformation* durch. Das ist eine Transformation bei welcher abhängige und unabhängige Variable vertauscht werden.
5. Als Resultat erhält man dann die geplottete Lösung.

Eine kleine Schwierigkeit ist dabei die explizite Lösung des Eigenvektorproblems für den Schrödinger-Operator zu finden. Auch hier weiß man sich mit ein wenig Theorie zu helfen. Aufgrund der Tatsache, daß das vorgelegte Potential reflektionsfrei ist, kann man in der Tat einen potentialabhängigen Integro-Differentialoperator finden, welcher das vorgegebene Potential auf die Quadrate seiner Eigenvektoren abbildet. Auf Grund der Symmetriegruppenanalyse findet man diesen Operator immer als Produkt einer Integration mit einem Polynom des Ausdrucks Φ, welcher in (3.10) angegeben wurde. Die Ordnung dieses Operators steigt allerdings linear mit der Anzahl der Eigenwerte an.

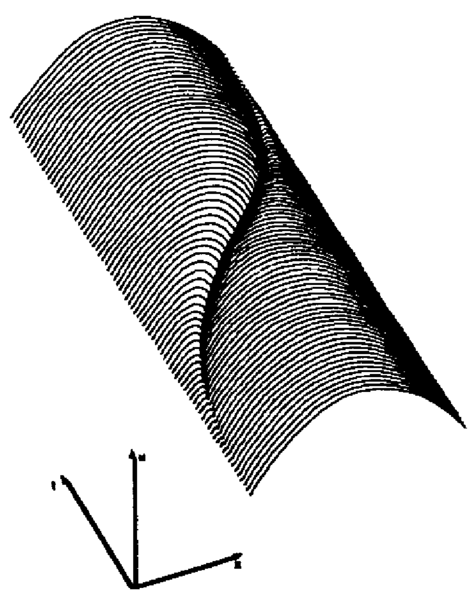


Figur 7: Drei-Soliton der Harry-Dym-Gleichung

Der angegebene, frappierend einfache, Lösungsweg hat den leichten Nachteil, daß man, obwohl alle Schritte mathematisch leicht zu durchschauen sind, für die Lösung eine explizite Formel erhält, die sich über nahezu 50 Seiten erstreckt.

Mit der Bearbeitung per Hand, und anschließender numerischer Auswertung, kann man die kostbare Zeit eines kreativen Mathematikers blockieren. Verwendet man hingegen an dieser Stelle Computeralgebra, so kann einem die Länge der auftretenden Formel gleichgültig sein, denn man sieht sie gar nicht, und man erhält das Resultat in Sekundenschnelle.

Dies war aber noch ein sehr bescheidenes Beispiel zur Illustration des Nutzens von Computeralgebra . Erhöht man etwa die Zahl der interessierenden Dellen (Figur 7), oder erhöht man die Ordnung der zugrunde liegenden nichtlinearen Differentialgleichung, bei welcher man solche Phänomene erwartet, betrachtet man also zum Beispiel eine explizite Lösung



Figur 8: Zwei-Soliton der Kawamoto-Gleichung

der sogenannten Kawamoto-Gleichung,

$$u_t = 10u^4 u_{xx} u_{xxx} + 5u^4 u_x u_{xxxx} + u^5 u_{xxxxx} \ ,$$

so erhält man mit ähnlichen Methoden, die natürlich erst mathematisch durchschaut werden müssen, Formeln über eine Länge von Hunderten von Seiten. Auch in diesen Fällen werden die gezeigten Lösungen, trotz der Länge der Formeln, in recht kurzer Zeit ermittelt und geplottet.

Interessant ist, daß man es dabei mit Problemen der *Reinen Mathematik* zu tun hat, welche die Grenzen dessen, was ein Mathematiker mit Kopf und Hand tun kann, bei weitem überschreiten. Trotzdem sind dies strukturell einfache Probleme, deren Lösungen nicht etwa numerische Einsichten, sondern *theoretische Erkenntnisse* über die Natur der vorgelegten nichtlinearen Systeme vermitteln.

4.2 Zum Unvermögen von Computeralgebra

Die im letzten Abschnitt gemachten Ausführungen täuschen mit ihrer einfachen Formulierung etwas über die wirklichen Schwierigkeiten der Verifizierung entsprechender Eigenschaften hinweg. Wir wollen ein einfaches aber schlagkräftiges Beispiel dafür vorführen.

Wir betrachten die schon erwähnte Kawamoto Gleichung, die wir aus einem Heft des „Journal of the Physical Society of Japan" entnommen haben:

$$u_t = 10u^4 u_{xx} u_{xxx} + 5u^4 u_x u_{xxxx} + u^5 u_{xxxxx} \,. \tag{4.2}$$

Aus Strukturüberlegungen findet man für diese Gleichung als Kandidaten [2] für einen geigneten invarianten hereditären Operator die folgende Größe:

$$\Phi(u) = \varrho^2 D J(v) \Theta(v) D^{-1} u^{-2} \,, \tag{4.3}$$

wobei v folgende Hilfsvariable

$$v = u u_{xx} - \frac{1}{2}(u_x)^2$$

ist, und J und Θ Abkürzungen für folgende Operatoren sind

$$\begin{aligned} \Theta(v) &= uDuDuv_x + 3uvv_x \\ J(v) &= uDuDuv_x + 3(uvv_x + uDv^2) \\ &\quad +2[uDuDvD^{-1}vu^{-1} + D^{-1}vDvDuv] \\ &\quad +8[v^2 D^{-1} vu^{-1} + D^{-1} v^3 u^{-1}] \,. \end{aligned}$$

Die Gültigkeit von (3.6) prüft man am besten durch Nachrechnung in einer geeigneten Karte. In den Koordinaten der Karte ausgedrückt, ist die Eigenschaft der Hereditarität äquivalent dazu, daß der folgende Ausdruck für beliebige Vektorfelder A und B, symmetrisch zu sein hat

$$\Phi\Phi'[A]B - \Phi'[\Phi A]B \,.$$

Wegen des Vorkommens der D^{-1} ist es leider gar nicht offensichtlich, welche der Terme sich gegenseitig aufheben. Man muß zum Beispiel, um nur einen einfachen Fall anzuführen, entscheiden ob Terme der Art

$$D^{-1}AD^{-1}uD^{-1}B_{xx} + D^{-1}u_x AB + D^{-1}AD^{-1}u_x B_x - uAB \tag{4.4}$$

sich zu Null aufheben oder nicht. Bei diesem Term kann man das allerdings noch leicht mit dem bloßen Auge sehen, aber die wirklich vorkommenden Ausdrücke sind bei diesem Beispiel noch viel komplizierter. Man muß also durch trickreiche, partielle Integration Terme in geeigneter Weise ineinander überführen.

Partielle Integration ist aber eine syntaktische *Regel* und führt, zusammen mit den anderen arithmetischen Regeln, zu einer *formalen Sprache*. Dieses ist eine kontextsensitive Sprache, bei der die Regeln zudem nicht einmal *beschränkt* sind, da ja die durch partielle Integration entstehenden Ausdrücke keineswegs immer kürzer als ihre Vorgänger sind. Die Beschränktheit ist aber ein Erfordernis, welches man im allgemeinen zur Terminierung der Ableitungen braucht. Und diese Terminierung von Ableitungen ist es, die einem erlaubt, Normalformen zu finden. Nun man kann sich vielleicht vorstellen, daß es an Stelle der Beschränktheit ein anderes strategisches Prinzip gibt, welches zur Entscheidbarkeit der Sprache führt.

Nehmen wir einmal an, daß wir das Normalformenproblem dieser Sprache zufriedenstellend lösen können, und fragen wir uns was dann noch an Kompliziertheit verbleibt. Eines steht fest, bevor wir an die Umformung in Normalform herangehen können, müssen wir alle Ableitungen mit der Produktregel expandieren. Schauen wir was das bedeutet. Wenn in $\Theta(v)D^{-1}u^{-2}$ alle Ableitungen ausgeführt werden, so erhalten wir ungefähr 20 Terme, da ja ein D vor einem Term vierter Ordnung und ein anderes vor einem Term fünfter Ordnung steht. Führen wir nun die zusätzlichen Ableitungen in J aus, welche vor Termen achter und neunter Ordnung stehen, so gibt das einen zusätzlichen Faktor 70. Um Φ zu erhalten müssen wir nun eine letzte Ableitung vor einem Term zehnter Ordnung ausführen. Wir haben also schon $20 \times 70 \times 10$ Terme in Φ. Aber jetzt kommt erst die wirkliche Katastrophe: Die Variationsableitung eines Ausdrucks zehnter Ordnung hat ungefähr zehnmal soviel Terme wie dieser Ausdruck selbst, und da Φ quadratisch in (3.5) eingeht, haben wir die Zahl der Terme in Φ noch zu quadrieren. Nach der Entwicklung unseres Ausdrucks liegen also ungefähr

$$2 \times 10 \times (20 \times 70 \times 10)^2 = 4 Milliarden \tag{4.5}$$

Terme vor. Der Ausdruck dieses Operators, etwa um ihn einem Diplomanden zur Handberechnung anzuempfehlen, ergibt einen Stoß Din A 4-Blätter von etwa 18 km Höhe, oder einen Blätterberg von 1000 Kubikmetern.

Schreckt man davor zurück, diesen Operator auszudrucken und versucht ihn im Rechner weiter zu behandeln, so kommt man auf ein algebraisches Datum der Größenordnung von vielen Gigabyte. Herkömmliche Computeralgebra-Systeme versuchen meistens, ein solches algebraisches Datum auf einmal in den Hauptspeicher zu laden, um es dann weiter zu behandeln. Unnötig zu betonen, daß dies zum Widerspruch der meisten Rechner führt. Eine Workstation mit dem Hauptspeicher vieler Cray's wäre dafür notwendig. Schon dieses einfache Problem ist also weit davon entfernt, sich auf einfache Weise mit einem normalen Computeralgebrasystem und einem normalen Computer lösen zu lassen. Und bei schwierigeren Problemen stehen wir erst ganz am Anfang dessen, was wir von effizienten Hilfsmitteln in der Zukunft verlangen werden.

Man kann sich natürlich fragen, ob sich solch eine komplizierte Rechnung überhaupt lohnt. Die Antwort auf diese Frage ist recht einfach, es lohnt sich wirklich. Denn eine einmalige Rechnung würde uns nämlich für diese Gleichung eine *entscheidende Hilfe* für alle erdenklichen Anfangswertprobleme geben. Außerdem wäre die anschließende Berechnung der Symmetriegruppen, wenn man einmal die hereditäre Eigenschaft verifiziert hat, von viel geringerer Schwierigkeit. Hinzu kommt noch, daß die Kenntnis eines solchen Beispiels die Durchdringung vieler weiterer, auch einfacherer Beispiele, nach sich ziehen würde.

5 Auswirkungen von Computeralgebra auf Lehre und Forschung: Eine Herausforderung an die Mathematik

Trotz des, im letzten Abschnitt geschilderten, Unvermögens gegenwärtiger Computeralgebrasysteme, werden, mittelfristig gesehen, solche Systeme weitgehende Auswirkungen darauf haben, wie wir und andere mit Mathematik umgehen.

Schnelle und nachhaltige Änderungen werden sich zuerst, und dies innerhalb der nächsten 10 Jahre, im Bereich der Lehre einstellen. Die Computeralgebrasysteme, die gestern einen Mainframe erforderten und heute auf einer passablen Workstation laufen, werden in 5 Jahren auf dem Notebook eines jeden Ingenieurs und Naturwissenschaftlers zu finden sein.

Bedenkt man, daß eine große Zahl der heute kreativ forschenden Mathematiker den notwendigen Freiraum für ihre Tätigkeit dem Einsatz in der Ausbildung von

Ingenieuren und Naturwissenschaftlern verdanken, so wird klar, daß eine Sicherung und ein Ausbau dieses Freiraums Änderungen im Bereich der akademischen Lehre erfordert. Denn manche Anwender, die häufig nur unter unsäglichen Schwierigkeiten mit mathematischen Formeln umgehen, werden diesen Schwierigkeiten nur allzu gerne durch den Umgang mit Computeralgebra zu entrinnen suchen. Der Umgang damit wird ihnen leichtgemacht werden, einerseits weil sie sowieso mit Computern umgehen müssen und weil manchmal selbst der sinnlose Umgang damit ihnen das Gefühl einer sinnvollen, anspruchsvollen Tätigkeit vermittelt, andererseits weil immer nutzerfreundlichere Interfaces diese Tätigkeit wirklich erleichtern werden. Mit den Computeralgebrasystemen der Zukunft wird, zumindest an der Oberfläche, so einfach umzugehen sein, daß kaum noch mathematische Vorbildung erforderlich ist.

Gelingt es den Mathematikern nicht den sinnvollen und kritischen Umgang mit solchen Werkzeugen in ihren Veranstaltungen zu vermitteln, so wird ein Großteil der künftigen technisch-naturwissenschaftlichen Elite einen mehr rezeptiven Umgang mit diesen Systemen in den Anwendervorlesungen lernen, und dies bei Anwendern, deren Aufgeschlossenheit gegenüber mathematischen Subtilitäten schon heute nicht allzu groß ist. Eine neue Dimension intellektueller Oberflächlichkeit, verbunden mit einem weiteren Zurückdrängen mathematischen Freiraumes, wäre die Folge. Die schon jetzt beobachtbare Tendenz, daß eine immer stärkere Mathematisierung unserer Welt zu einem immer geringeren Ansehen der Mathematiker selbst führt, würde sich verstärken.

Es ist gar keine Frage, daß neue Inhalte für unsere Curricula zu finden sind, die Frage ist nur, wie schmerzhaft uns die Suche danach werden wird. Die Gefahr ist, daß unsere anfänglichen Schwierigkeiten mit diesen Entwicklungen, wenn schon nicht ihr Vorreiter zu sein, so doch zumindest Schritt zu halten, zu einem neuen Abschwung in der Zahl forschender Mathematiker führen kann.

Ich glaube aber nicht, daß wir uns auf diese Entwicklungen nur aus Notwendigkeit, und weil wir nicht anders können, einstellen sollten, sondern daß wir dies aus intellektueller Neugier gegenüber dem Unbekannten, mit Engagement und kritischer Aufgeschlossenheit, tun werden. Ich bin sicher, wir werden auch manche wissenschaftliche Befriedigung daraus gewinnen. Außerdem ist es eine wichtige, kulturhistorische Aufgage, angesichts des technischen Fortschritts, der wie immer dazu führen wird, daß man Inhalte mit äußeren Erscheinungsformen verwechseln wird, für kritisches Bewußtsein im Umgang mit neuen Werkzeugen zu sorgen. Es sollte uns ein ernstes Anliegen sein, dieses kritische Bewußtsein zumindest in unserem Bereich, dem der Mathematik, zu vermitteln, einem Bereich der ja wohl eigentlich im Kern allen technischen Fortschritts liegt.

Mathematiker sind dazu aufgerufen, den naiven Glauben an die Allmacht des Computers, der in unserer Öffentlichkeit herrscht, durch intelligente Kritik, doch verbunden mit kenntnisreicher Aufgeschlossenheit, in eine differenziertere Betrachtungsweise überzuleiten; dies zumindest bei den von ihnen ausgebildeten Angehörigen der künftigen technischen Elite.

Daß die beschriebene Zukunft schon begonnen hat, sieht man daran, daß in unserem Nachbarland Österreich das Unterrichtsministerium eine Lizenz des Computeralgebrasystems *Derive* für *alle* Gymnasiasten angekauft hat [10, p. 21]. Seit Herbst 1991 ist Derive, ein System mit dem Slogan *„2000 Jahre Mathematik auf einem Disk"* Standardwerkzeug der österreichischen Gymnasien.

Beachtliche Umwälzungen werden aber auch in weiten Bereichen der Forschung eingeleitet werden. Dies insbesondere, weil

- durch die Verfügbarkeit von deklarativen Hochsprachen, die in ihrer Syntax den üblichen mathematischen Formeln weitgehend angepaßt sind, und die standardmäßig Zugriff auf mathematisches Expertenwissen ermöglichen, das *Rapid Prototyping* (dann nicht unbedingt *quick-and-dirty*) kompliziertester Sachverhalte für den Mathematiker, den Ingenieur und Naturwissenschaftler zur täglichen Routine werden wird,

- diese Systeme in Zukunft einen fast unbeschränkten Zugriff auf gute (wie wohl auch auf schlechte) Algorithmen gewähren,

- durch das Wegfallen von Routinemanipulationen neue Möglichkeiten geschaffen werden, die auch neue inhaltliche Dimensionen erschließen werden.

Es ist offensichtlich, daß die Etablierung von *Experimenteller Mathematik* dazu fühen wird, daß leicht durchzuführende mathematische Experimente uns neue Anschauungen und Intuitionen vermitteln werden und Schritte in eine ganz neue Dimension mathematischer Forschung ermöglichen können. Die Vielzahl leicht handhabbarer Beispiele wird zu neuen Entdeckungen und Strukturen führen.

Wir alle wissen, daß die Grenzen zwischen den von uns definierten Teildisziplinen, wie zum Beispiel zwischen Analysis und Algebra, fließend sind, und daß wir sie nur zur Ermöglichung einer ersten oberfächlichen Orientierung verwenden. Doch haben diese Grenzen trotzdem in den vergangenen 50 Jahren zu einem beachtlichen Beharrungsvermögen geführt, uns vielleicht mitunter auch den Blick auf die weite Landschaft der Mathematik verstellt. Diese Grenzen werden sicher aufgeweicht werden, und die gemeinsame Faser *Mathematik* wird wieder mehr zum Vorschein kommen. Ganz offensichtlich ist, daß neue Methoden und Werkzeuge unsere mathematische Blickrichtung ändern werden. Computeralgebra sorgt heute schon dafür, daß das was mancher gestern als Analysis ansah, er heute als Algebra erkennt. Algebra und Diskrete Mathematik, weil sie näher der algorithmischen Durchdringung mathematischer Sachverhalte liegen, werden eine neue Blüte erfahren.

Die Einstellung gegenüber dem, was wir als *einfachen* mathematischen Sachverhalt ansehen, wird sich wandeln. Mathematiker wissen natürlich, daß *Einfachheit* allenfalls eine strukturelle Kategorie, oder vielleicht auch eine ästhetische Kategorie ist, und nicht die Frage nach der Größe von Formeln. Trotzdem lassen wir uns in unserem täglichen Arbeiten und Urteilen oft von der Kompliziertheit *unserer* Beschreibung mathematischer Gegenstände, also der entsprechenden Formel, dazu verführen, einen Sachverhalt für kompliziert zu halten. Aber in dem Moment, in dem Formeln gar nicht mehr gesehen werden, sondern nur von einem Interface an das andere weitergereicht werden, klärt sich unser Blick notwendigerweise. Denken wir an die im vierten Abschnitt vorgestellten expliziten Lösungen einiger nichtlinearer partieller Differentialgleichungen. Beim Anschauen der Bilder wird man sich der Einsicht nicht verschließen können, daß es sich dabei um eine sehr einfache strukturelle Eigenschaft handeln muß. Daß dies bisher nicht entdeckt oder durchschaut wurde, lag nur an dem Wust von Formeln der zur Beschreibung nötig war; Formeln, die aber morgen unseren, auf das Ästhetische gerichteten Blick, nicht mehr stören werden.

Welchen Herausforderungen stehen wir wissenschaftlich gegenüber? Ich glaube nicht, daß es sinnvoll ist, daß wir unsere mathematischen Interessen aufgrund des technischen Fortschritts ändern. Aber unsere Aufgeschlossenheit und unsere Neugier, diesen technischen Fortschritt ein Stück Wegs zu begleiten, sind gefragt. Die Herausforderungen, denen wir uns stellen müssen, kommen im Gebiet der Computeralgebra daher, daß eine notwendige drastische Steigerung der Effizienz solcher Systeme nicht ohne Beteiligung der Mathematiker möglich ist. Mathematik ist unter anderem aufgerufen

- das mathematische Wissen, welches die Libraries dieser Systeme bereitstellen, ständig zu verbessern. Auch um die Leistungsfähigkeit und Schnelligkeit dieser Systeme zu steigern.
- Wir sind aufgerufen, über die Standardisierung mathematischer Formeln nachzudenken, darüber was wir als kanonische Darstellung betrachten. Es ist die Frage zu lösen, wie man unzweideutig, syntaktisch einwandfrei Mathematik aufschreiben muß, so daß ein Computer nicht nur die Zeichen lesen kann, sondern sie auch gemäß ihrer Bedeutung weiterverarbeiten kann. Wir brauchen eine abstrakte, mathematische high-level-Programmiersprache, die unabhängig von der Implementierung des sie verstehenden Systems ist.

Die Schaffung einer syntaktisch korrekt beschreibbaren mathematischen Hochsprache ist aus zwei Gründen wichtig. Einmal als Kommunikationsverbindung zwischen verschiedenen Computeralgebrasystemen (und Textverarbeitungssystemen), zum zweiten, um den künftigen Herausforderungen des *Electronic Publishing* gewachsen zu sein. In 20 Jahren wird ein mathematischer oder technischer Artikel nur an der Oberfäche ein auf Papier gedrucktes Werk sein[7]. Dahinter wird in jedem Fall eine hierarchisch aufgebaute Datei liegen, die es dem Nutzer ermöglicht, Formeln vom Bildschirm abzugreifen, dazu aufgestellte Behauptungen weitgehend mechanisch zu verifizieren, und auch, sie in eigene Publikationen einfließen zu lassen.[8] Konsequenz aus dieser Perspektive ist, daß der Unterschied zwischen Textverarbeitungssystem und Computeralgebrasystem ein Anachronismus sein wird. Ansätze zur Überwindung dieses Anachronismus sind vorhanden.

Der Eintritt des geschilderten Zustandes hängt nicht so sehr vom Hardware-Fortschritt sondern vielmehr vom mathematischen Fortschritt ab. Um einen Punkt zu nennen: Computeralgebrasysteme sind eben *Algebrasysteme* und keine *Analysissysteme*, geschweige denn *Mathematik*systeme. Ein wichtiger Grund für dieses Defizit ist darin zu sehen, daß sie mit charakteristischen Schlußweisen, die Analytikern eigen ist, nicht umgehen können. Sie können, bisher jedenfalls, nicht auf intelligente Weise mit Ungleichungen rechnen. Ich bin deshalb davon überzeugt, daß wir in den nächsten 10 Jahren einer Disziplin begegnen werden, die sich die algorithmische Durchdringung unseres intelligenten und intuitiven Umgangs mit Ungleichungen zum Ziel setzt. Diese Disziplin wird es uns ermöglichen, Computeralgebrasysteme zu vergessen und statt dessen *Computermathematiksysteme* zu benutzen.

Um nicht mißverstanden zu werden, ich habe nicht Stellung dazu genommen, ob diese Veränderungen wünschenswert sind. Ich bin aber sicher, daß neben mancher abträglichen Begleiterscheinung, unsere kritische Beteiligung an diesen Veränderungen dazu führen könnte, daß wir ein Großteil unserer täglichen Routinearbeit als Mathematiker abgeben können, um so den Blick frei zu bekommen für neue Entdeckungen im Universum mathematischer Strukturen.

6 Ein Blick in die Zukunft: Mein bevorzugtes System im Jahre 2000

Ich möchte diesen Aufsatz damit abschließen, daß ich dem gegenwärtig unvollkommenen technischen Stand in der Computeralgebra die Leistungsfähigkeit zukünftiger Systeme gegenüberstelle.

[7] Daß er zum Lesen auf Papier gedruckt wird, wird sicher noch lange Zeit die vorherrschende Technik bleiben.

[8] Ich will keineswegs sagen, daß dies ein schöner oder gar wünschenswerter Zustand sein wird.

Beim täglichen Umgang mit Computeralgebra stellt man heute eine Reihe von Unzulänglichkeiten fest. Um nur einige zu nennen:

- Design, Spezifikation und Implementation von Computeralgebra sind kommerziellen Interessen unterworfen. Bei general-purpose Systemen war der Einfluß europäischer Mathematiker bisher eher gering.
- Neue Entwicklungen sind, obwohl häufig das Gegenteil behauptet wird, manchmal schwer zu erhalten,
- Der Quellcode dieser Systeme ist nicht vollständig verfügbar, was angesichts der Tatsache, daß sowohl die Datenstruktur wie auch wesentliche Algorithmen zum Teil im Systemkern implementiert sind, der wissenschaftlichen Ernsthaftigkeit der Nutzer arge Zumutungen auferlegt. Dies ist nicht nur eine Frage der wissenschaftlichen Ethik, sondern auch der täglichen Praktikabilität.

Ich verkenne nicht, daß die Entwicklung solcher Systeme, und bei guten Systemen stimmt dies in besonderem Maße, nur durch große Investitionen im personellen und sächlichen Bereich möglich ist. Deshalb möchte ich mich der Meinung, daß eine Kommerzialisierung der Computeralgebra und damit der diesen Systemen zugrunde liegenden Mathematik, grundsätzlich von Übel ist, nicht ganz anschließen. Aber die Probleme der Kommerzialisierung geistiger Produkte, von denen wir in Zukunft abhängen werden, sollten zumindest nicht übersehen werden. Es sollten mehr Systeme aus dem nichtkommerziellen Forschungsbereich kommen. Dies setzt aber voraus, daß die Entwickler dieses Bereichs sich auch der notwendigen Professionalität bei Interfaces und Dokumentation befleißigen, um nur zwei Bereiche zu nennen. Wegen der gegenseitigen Interdependenz, wäre es aber sicher eine kluge Politik, wenn die nichtkommerziellen wissenschaftlichen Einrichtungen unbegrenzten Zugriff auf alle, auch kommerzielle, Systeme erhalten würden. Dies heißt nicht etwas zu verschenken, sondern nur die Nutzung durch diejenigen anzuregen, deren kritische Rückmeldung für die Entwicklung des *Systems der Zukunft* unerläßlich ist.

Viel entscheidender aber als diese beklagenswerte Situation ist die Tatsache, daß die bisher vorhandenen Computeralgebra-Systeme in ihrer *Leistungsfähigkeit unzureichend* sind.

Alle vorhandenen Systeme stellen zwar wunderbare Werkzeuge dar, die den effizienten Umgang mit, nach dem heutigen Stand, großen algebraischen Daten bis zu einem Megabyte, oder auch etwas mehr, erlauben. Die vorhandenen Systeme wurden eben entworfen, um den heutigen routinemäßigen täglichen Umgang mit Mathematik und mathematischen Formeln zu erleichtern.

Geht man aber davon aus, daß durch den Einsatz von Werkzeugen dieser Art auch im Bereich der Anwendung von Mathematik auf technische und naturwissenschaftliche Probleme ein Schritt in völlig neue Dimensionen möglich sein sollte, so wird man feststellen, daß es nicht darum geht, bisherige mathematische Tätigkeiten zu vereinfachen, sondern den Umgang mit mathematischen Formeln völlig neuer Größenordnung, meiner Meinung nach im Gigabyte-Bereich und jenseits davon, zu erlauben. Dies ist notwendig, um neue, praktisch relevante Probleme zu behandeln. Diese Perspektive wird von den vorhandenen Systemen nicht genügend unterstützt.

Ich bin sicher, daß, gemessen an den Systemen der Zukunft, die heutigen Computeralgebra-Systeme als bescheidene Spielzeuge belächelt werden. Daß diese Perspektive nicht falsch ist, sieht man unter anderem daran, daß bei meinen ***konkreten Arbeiten*** im Bereich recht einfacher nichtlinearer Systeme durchaus relevante algebraische Daten vorkommen, deren Größe vom mehrstelligen Megabyte-Bereich bis in den zweistelligen Gigabyte-Bereich gehen.

Ich glaube meine Erfahrungen besitzen eine gewisse Allgemeingültigkeit. Ich will sie deshalb kurz zusammenfassen:

- Die landläufige Meinung, daß die Leistungssteigerung bei Computern so rasch vonstatten ginge, daß in Zukunft Speicherplatzerwägungen und Laufzeitaspekte keine Rolle mehr spielen würden, ist falsch, ja sogar grundfalsch. Bei wirklichen Problemen, gerade auch im mathematischen Bereich, wird auch in Zukunft die Effizienz bezüglich des ökonomischen Umgangs mit Laufzeit und Speicherplatz eine entscheidende Rolle spielen. Noch immer ging der Wettlauf zwischen technischem Fortschritt und dem Appetit der Nutzer zu *Un*gunsten des technischen Fortschrittes aus.

- Selbst unter Einbeziehung guter Computeralgebrasysteme sind wir immer noch deutlich davon entfernt, *wirkliche* mathematische bzw. algebraische Probleme zufriedenstellend lösen zu können. Wir werden noch viel mehr Arbeit in die Entwicklung leistungsfähiger Software hereinstecken müssen. Ich glaube, wir sind da erst am Anfang.

- Das immer weitere Ausweichen auf höhere Programmierebenen ist nicht sinnvoll. Ich glaube, man sieht dies zum Beispiel daran, daß CA-Systeme, die in Lisp programmiert sind, denen, die in C programmiert sind, an Effizienz unterlegen sind. Natürlich ist der Nutzen von Hochsprachen für das „Rapid-Prototyping" nicht zu übersehen, aber dauerhafte software-Tools sollten assemblernah geschrieben werden. Es macht sicher wenig Sinn in einer High-level-Sprache (z.B einem Computeralgebrasystem) zu programmieren, um dann bei der Compilierung mehrere andere High-level-Sprachen dazwischen zu schalten. Diese Effizienznachteile kann man sich beim Vorliegen wirklicher Probleme nicht leisten.

- Die meisten Computeralgebrasysteme orientieren sich an einer zu simplen Nutzervorstellung. Natürlich ist man heute leicht begeistert, wenn ein Computer das tun kann, wofür ein ausgewachsener Mathematiker vielleicht zwei Tage braucht, aber unsere Vorstellungen was „einfach" ist, werden sich, wie gesagt, drastisch ändern. Es fehlen deshalb „heavy duty" Systeme mit kleinem aber effizientem Kern, die nicht über zu viele Libraries verfügen und bei denen die wesentlichen Routinen und elementaren Operationen auch auf Prozessorebene realisiert sind.

- Die Entwicklung und Fortentwicklung von Parallelrechnern und den dazu geeigneten Programmierumgebungen ist das Gebot der Stunde. Computeralgebra ist dabei ein vorzügliches, weil einfaches, Anwendungsbeispiel.

6.1 Mein System im Jahre 2000

Aus dieser Situationsschilderung ergibt sich folgender Wunschkatalog für mein *System der Zukunft*:

- Das System ist ein general-purpose System, welches aus wiederverwertbaren Programmodulen besteht. Die Quellcodes stehen jedem interessierten Wissenschaftler offen.

- Das System erlaubt parallele Verarbeitung gemäß einem nutzerdefinierten Profil. Es kann eine automatische Parallelverarbeitung großer algebraischer Daten vornehmen. Die parallelen Aspekte des Systems bauen auf einer rechnerunabhängigen Kommunikationsstruktur auf.
- Das System verfügt über Routinen, mit denen eine effiziente, automatische Parallelisierung von Nutzerprogrammen vorgenommen wird. Bei dieser automatischen Parallelisierung wird nicht nur die syntaktische Abhängigkeit von Programmteilen, sondern deren logische Abhängigkeit berücksichtigt.
- Das System hat eine flexible Programmiersprache, deren Struktur auf einen weitgehenden Konsens innerhalb der Mathematischen Welt aufbaut. Das System stellt Crosscompiler *von und in* alle herkömmlichen Computeralgebrasysteme zur Verfügung.
- Das System stellt entsprechende Interfaces, sowohl zu rechnerunabhängigen Oberflächen wie zu intelligenten Textverarbeitungssystemen, welche in ihrer Formelbehandlung auf der mathematischen Syntax dieser Formeln aufbauen, zur Verfügung. Die Unterscheidung zwischen Computeralgebrasystem und Textbearbeitungssystem ist weitgehend aufgehoben.
- Das System verfügt über eine Vielzahl von Elementen zur automatischen Programmgenerierung. Dies ist nicht nur eine Frage der Praktikabilität, sondern in diesen Aspekten sollte sich auch die Nähe zu algebraischem Arbeiten in der Mathematik widerspiegeln.
- Das System hat sich davon gelöst *Algebra*system zu sein, es ist ein Computer*mathematik*system.
- Das System ist lernfähig, und nimmt während des interaktiven Arbeitens, mit Hilfe seiner Werkzeuge zur Programmgenerierung eine steuerbare automatische Veränderung seines funktionalen Teils vor, es hat damit einen ersten Schritt in Richtung dessen getan, was durch das anrüchige Wort *Künstliche Intelligenz* ausgedrückt wird.
- Die ständige Weiterentwicklung dieses Systems, und seine Portierung auf alle geeigneten Rechner, ist Angelegenheit der *Mathematical Community*, die den damit zusammenhängenden Problemen eine besondere Sorgfalt angedeihen läßt.
- Wesentliche Teile des Kerns des Systems sind auf Prozessorebene realisiert.

References

[1] V. I. Arnold: *Mathematical Methods of Classical Mechanics*, Springer Verlag, Berlin-Heidelberg-New York, 1978

[2] S. Carillo and B. Fuchssteiner: *The abundant symmetry structure of hierarchies of nonlinear equations obtained by reciprocal links*, J. Math. Phys., 30, p.1606-1613, 1989

[3] Y. Choquet-Bruhat and C. De Witt-Morette and M. Dillard-Bleick: *Analysis, Manifolds and Physics*, North-Holland Publ. Co, Amsterdam-New York-Oxford, 1982

[4] B. Fuchssteiner and G. Oevel: *Geometry and action-angle variables of multisoliton systems*, Reviews in Mathematical Physics, 1, p.415-479, 1990

[5] B. Fuchssteiner: *Hamiltonian structure and Integrability*, in: Nonlinear Systems in the Applied Sciences, Mathematics in Science and Engineering, Vol. 185, Academic Press, C. Rogers and W. F. Ames eds., p.211-256, 1991

[6] B. Fuchssteiner: *Linear aspects in the theory of Solitons and nonlinear integrable equations*, Journal of the Phys. Soc. Japan, 60, p.1473-1496, 1991

[7] M. Hazewinkel: *Experimental Mathematics*, Math. Modelling, 6, p.175-211, 1985

[8] M. Hazewinkel: *Experimental Mathematics*, in: Mathematics and Computer Science (J.W. de Bakker, M. Hazewinkel, J.K. Lenstra, eds.) CWI/North Holland Publ., p.193-234, 1986

[9] A. C. Hearn and al.: *Future Directions for Research in Symbolic Computation*, Siam Reports on Issues in the Mathematical Sciences, Philadelphia, 1990

[10] *MathPAD: Eine nutzerorientierte Information aus der MathPAD-Gruppe an der Universität-Gesamthochschule Paderborn*, Vol. 1, Heft 3, September 1991

[11] R. Pavelle, M. Rothstein und J. Fitch: *Computer algebra*, Scientific American, 245, p.102-113, December 1981

[12] F. Winkler, B. Kutzler and F. Lichtenberger: *Computer-Algebra Systems*, Risc-Linz series no 88-10.0, p.33 pages, 1988

Benno Fuchssteiner
Fachbereich Mathematik-Informatik
Gesamthochschule Paderborn
D 4790 Paderborn