4. Didaktische Prinzipien als Konstruktionsanleitungen für den Unterricht

"Dem Lehrjungen gebühret zuzuhören und still zu schweigen... Der Lehrjunge soll nichts reden, in wehrender Lection auch nichts sagen. Denn sonst verhindert er beyde, die Lehrmeister und seine Mitschüler, daß die Lection nicht kann zur rechten Zeit vollendet werden. Hat er aber nöthigs zu fragen, so schreyb er beyseit auff, und nach gehaltener Lection hat er zu fragen Zeit genug."

Ratke, Artickel auf welchen fürnehmlich die Ratichianische Lehrkunst beruhet, Leipzig 1617 [vgl. Freudenthal, 1977, S 55].

"Dem Lehrmeister gebühret zuzuhören und still zu schweigen... Der Lehrmeister soll nichts reden, in wehrender Lection auch nichts sagen. Denn sonst verhindert er beyde, die Lernmädchen und Lernjungen, daß die Motivation und Action nicht kann verwirklicht werden. Hat er aber nöthigs zu sagen oder zu fragen, so schreyb er beyseit auff, und nach vollendeter Action hat er zu sagen und zu fragen Zeit genug."

Heugl, Artickel auf welchen fürnehmlich die Kunst beruhet, Lernmädchen und -jungen zu motivieren und zu activieren, Wien 1996.

Die Erziehung zu selbständigem Denken, Urteilen und Handeln ist wohl einer der wichtigsten Aufträge der Schule der Zukunft. Das erfordert aber weniger eine Änderung der Lerninhalte, sondern ein Umdenken bei der didaktischen Konzeption und bei der Organisation des Unterrichts. Nun brauchen wir diesbezüglich das Rad im Jahr 1996 nicht neu zu erfinden. Der Begriff "Projekt" im Sinne einer Lernmethode oder -konzeption wurde erstmals 1904 explizit formuliert und besonders seit Beginn der siebziger Jahre gibt es in der didaktischen Literatur und in den Lehrplänen eine Fülle von Anleitungen für einen stärker schülerzentrierten Unterricht. Trotzdem verläuft der Unterricht statistisch gesehen noch immer sehr stark lehrerzentriert, und die Ratichianische Lehrkunst gibt es auch noch.

Auch wir haben keinen Generallösungen anzubieten. Wir sind nach unseren Untersuchungen allerdings überzeugt davon, daß das Lehr- und Lernmedium Computer, und dabei insbesondere die Computeralgebra-Systeme, die didaktische Konzeption und die Unterrichtsorganisation stark beeinflussen und verändern. Diese These wollen wir in diesem Abschnitt durch Beispiele und Erfahrungen aus Unterrichtsexperimenten begründen.

So wie man beim Bau eines Hauses nicht nur Baumaterialien, sondern auch Hilfsmittel, wie etwa Maschinen oder ein Gerüst, und nicht zuletzt auch Konstruktionsanleitungen und Planungsvorschriften braucht, benötigt man zum Bau des mathematischen Gebäudes:

Baumaterialien, das sind die *mathematischen Inhalte*,

Hilfsmittel, also *Rechenhilfsmittel* wie den numerischen Rechner oder das CAS,

Konstruktionsanleitungen, das sind die *didaktischen Prinzipien*.

Quellen, aus denen didaktische Prinzipien stammen:

1. *Die pädagogische, gesellschaftliche Quelle:* Aus den allgemeinen Bildungszielen des Lehrplans, aus der Bildungs- und Lehraufgabe eines Fachs sowie den fachspezifischen Lernzielen lassen sich Konstruktionsvorschriften ableiten.

   Zu dieser Quelle gehört auch die gesellschaftliche und fachwissenschaftliche Diskussion um die Frage nach der Legitimation des Mathematikunterrichts (siehe etwa Arbeiten von B. Buchberger zum White Box/Black Box-Prinzip [Buchberger, 1993]).

2. *Die lernpsychologische Quelle:* Theorien wie etwa von J. Piaget, J.S. Bruner oder R.M. Gagne, um nur einige zu nennen, sind sowohl als Quelle für die Findung und Formulierung didaktischer Prinzipien als auch für ihre theoretische Absicherung von großer Bedeutung.

3. *Die Quelle der Empirie:* Durch Untersuchung des Unterrichtsgeschehens, des Lehrer- und Schülerverhaltens, der Lehr- und Lernprozesse werden bestimmte Gesetzmäßigkeiten festgestellt.

Wir wollen in diesem Buch nicht auf bekannte didaktische Prinzipien, wie etwa das Spiralprinzip oder das genetische Prinzip näher eingehen (siehe dazu z.B.: [Bruner, 1972], [Claus, 1989] oder [Wittmann, 1981]).

Uns geht es vielmehr um Konstruktionsanleitungen, bei denen das CAS als "Gerüst" oder "Baumaschine" eine wichtige Rolle spielt. Kenner didaktischer Literatur werden allerdings feststellen, daß etliche der in diesem Buch formulierten Prinzipien nur andere Ausprägungen klassischer didakischer Prinzipien sind, was nur die Universalität dieser Konstruktionsanleitungen bestätigt und uns die theoretische Absicherung erleichtert. So kann man etwa im White Box/Black Box-Prinzip deutlich Elemente des Spiralprinzips erkennen.

Unsere Absicht in diesem Kapitel ist es, der Bedeutung der neuen 'bautechnologischen Möglichkeiten', die sich aus der Nutzung von CAS ergeben, durch eine andere Formulierung und eine andere Gewichtung und Sichtweise Rechnung zu tragen.

4.1. Das White Box/Black Box-Prinzip

In impliziter Form wurde dieses Prinzip erstmals auf einem Symposium zum Thema "Symbolic Math in Education" konstruiert, das im Rahmen der ICME-Konferenz 1984 in Adelaide stattfand. Explizit formulierte es Prof. Bruno Buchberger, Vorstand des RISC-Instituts an der Universität Linz (Research Institut for Symbolic Computation) in einer Arbeit im Jahr 1990. Wie schon in der Einleitung zu diesem Kapitel erwähnt, findet man in diesem Prinzip Elemente des Spiralprinzips wieder, aber das Besondere daran ist, daß eine sinnvolle Realisation dieses didaktischen Konzepts erst bei Nutzung des Computers und insbesondere eines CAS möglich ist.

Viele Bereiche der Schulmathematik wurden in letzter Zeit in folgendem Sinn "trivialisiert": Der benötigte Algorithmus kann von Maschinen bearbeitet werden. Es ergibt sich daher die Frage, ob man den Schüler dann noch mit diesen Problemen quälen soll oder ob man nicht besser Maschinen zur Verfügung stellen soll, die die passenden Algorithmen beherrschen [Buchberger, 1992].

In der Diskussion über Segen oder Fluch des Einsatzes von Computern und insbesondere von CAS beobachtet man oft zwei extreme Positionen:

Die 'Traditionalisten': Der Computer ist eine Gefahr für die mathematische Kultur. Das blinde Verwenden von Werkzeugen verhindert das Verstehen mathematischer Konzepte. 'Basteln' ist kein brauchbarer Ersatz für die "Anstrengung des Begriffsbildungsprozesses".

Die 'Progressiven': Verlieren wir doch keine wertvolle Unterrichtszeit mit Tätigkeiten, die die Maschine besser beherrscht. Warum soll der Schüler einen Algorithmus herleiten oder verstehen können, wenn es eine **soLve**-Taste gibt? Wir würden mehr Raum für "kreatives Problemlösen" bekommen.

Wie so oft taugen Extreme nichts. Der sinnvolle Konsens zwischen den beiden extremen Standpunkten lautet: *White Box/Black Box-Prinzip*.

Es handelt sich dabei um ein rekursives Modell des Unterrichts, das jeweils in zwei Phasen abläuft:

4.1.1. White Box-Phase: Phase des verstehenden Lernens

Die Schüler sollen (genetisch oder nach einem wissenschaftsorientierten Ansatz) zu einem Begriff, einem Algorithmus, einem mathematischen Konzept geführt werden. Die in dieser Phase entwickelten Operationen sollen ohne Verwendung des Computers, also "zu Fuß", vom Schüler ausgeführt werden können. Grundtätigkeiten sollen durch Übung automatisiert werden.

Der Computer soll nur dort verwendet werden, wo Bereiche früherer White Box-Phasen als Black Box genutzt werden, oder allgemeiner, wo er zur Erhellung der aktuellen White Box beitragen kann.

Mögliche Aktivitäten in der White Box-Phase:

- Formulieren eines Problems; Finden einer Vermutung; Entwickeln eines Begriffs; Entwickeln eines Algorithmus; Beweisen.

- Rechnen ausreichend vieler Übungsaufgaben ohne CAS; experimentelles Lernen mit Unterstützung des Computers; CAS-unterstütztes Nutzen von Black Boxes, die in früheren White Boxes erforscht wurden.
- Diskussion der Lösungsfälle, der Grenzen und der Verallgemeinerungsmöglichkeiten der Methode; eigenständiges Entwickeln von Modulen, die in der Black Box-Phase als Black Boxes genutzt werden können.

4.1.2. Black Box-Phase: Phase des erkennenden und begründenden Anwendens

Die Schüler sollen die in der White Box-Phase entwickelten Algorithmen und Konzepte bei praktischen Problemen oder bei weiteren White Box-Phasen passend einsetzen. Das Bearbeiten des Algorithmus wird dem Computer als Black Box überlassen. Die Schüler sollen entscheiden, was zu tun ist, eventuell seine Entscheidung auch begründen, er muß es aber nicht mehr selbst tun.

Die *Rekursivität* besteht darin, daß man während einer bestimmten White Box-Phase Bereiche, die in einer in der Hierarchie niedrigeren White Box-Phase verstehend gelernt und entwickelt wurden, als Black Boxes nutzt usw. Das Gebäude der Mathematik entwickelt sich also als ein System ineinandergeschachtelter White und Black Boxes.

Wenn dieser Aufbau gelingt, wäre die Sorge mancher Mathematiker unbegründet, Computernutzung würde nur mehr ein unreflektiertes, irgenwann automatisiertes Nutzen von Black Boxes bedeuten, und viele wichtige mathematische Inhalte würden bedeutungslos. Im Gegenteil! Leider erlebt man im traditionellen Mathematikunterricht, in dem Rechenfertigkeiten oft sehr stark im Mittelpunkt stehen, viel mehr unreflektierte Black Boxes, als man glaubt. Alle Schüler wissen zwar in der Differentialrechnung, daß "x^5 zu $5.x^4$ wird", aber nur wenige können den Begriff des Differentialquotienten erklären, deuten oder die Regeln herleiten. Unsere Untersuchungen haben gezeigt, daß bei Nutzung von CAS mehr Raum für White Box-Phasen bleibt und die Schüler besser zum eigenständigen Lernen in dieser Phase angeleitet werden können. Das CAS macht also wichtige mathematische Inhalte bei Beachtung dieses Prinzips nicht überflüssig. Entlastet wird der Schüler aber bei komplexen Operationen in der Phase der Anwendung des bisher Gelernten, da das CAS dafür Black Boxes anbietet.

Natürlich sollten nicht alle im Sinne dieses Prinzips verwendeten Black Boxes "absolut schwarze Körper" sein (siehe dazu: Modul Prinzip). Wenn der Schüler seine Entscheidung für eine Black Box begründen oder die Ergebnisse der Black Box interpretieren soll, muß dafür gesorgt werden, daß eine gewisse 'White-Box-Kompetenz' erhalten bleibt, wie z. B. elementare Rechenfertigkeit in der Algebra, Strukturerkennungskompetenz, Definition verwendeter Begriffe, Wissen um mögliche Lösungsfälle usw. Diese Kompetenz wird nicht von selbst erhalten bleiben, sondern muß vom Lehrer im Sinne des didaktischen Prinzips der "Stabilisierung" oder im Sinne des "Spiralprinzips" immer wieder gefördert und gefordert werden.

4.1.3. Das White Box/Black Box-Prinzip in der Algebra

Unser ursprünglicher didaktischer Ansatz war anders: In der White Box-Phase sollte der Computer nicht verwendet werden. Die Konsequenzen für die Algebra wären gewesen, daß der Computer frühestens in der 9. Schulstufe eingesetzt würde und daß die elementare Algebra in der 6. und 7. Schulstufe 'CAS-frei' wäre. Durch die Experimente in Versuchsklassen der Stufen 6 und 7 haben wir unsere Meinung grundlegend geändert: Es hat sich gezeigt, daß das CAS viele Hilfen bietet, um die Boxes der elementaren Algebra 'white' zu machen. Ein Großteil der Beispiele im folgenden Artikel wurde im Rahmen der Experimente in den Versuchsklassen am Gymnasium Stockerau von G. Razenberger und W. Klinger entwickelt. Eine Sammlung von Aufgaben für die 7. bis 12. Schulstufe, die im Rahmen des österreichischen DERIVE-Projekts in Versuchsklassen erprobt wurden, wurde als ACDCA-Report Nr. 2 von den Projektlehrern K. Aspetsberger, K. Fuchs und W. Klinger veröffentlicht [Aspetsberger, 1994].

Einteilung des Algebralernens in aufeinanderfolgende white und black Boxes:

WHITE BOX	Genutzte BLACK BOXES
WHITE BOX 1: "Termbox" Aufstellen von Termen. Bearbeiten von Termen. Rechnen mit Termen.	 Nutzen des CAS zum Testen und Üben.
WHITE BOX 2: "Gleichungsbox" Entwickeln von Strategien zum Lösen von Gleichungen. Äquivalenzumformungen.	BLACK BOX: "Termbox" Die Termumformungen übernimmt das CAS als Black Box.
WHITE BOX 3: "Gleichungssysteme" Entwickeln von Strategien zum Lösen von Gleichungssystemen.	BLACK BOXES: "Termbox" "Gleichungsbox" Das Termumformen und das Lösen einzelner Gleichungen übernimmt das CAS als Black Box.
WHITE BOX 4: "Anwendungsbox" Nutzen der Algebrakenntnisse beim Problemlösen. White Box-Aktivitäten: Modellbilden, Interpretieren.	BLACK BOXES: "Termbox" "Gleichungsbox" "Gleichungssysteme" "Differentialrechnungsbox" usw. Das Operieren übernimmt das CAS als Black Box.

4.1.4. Die Termbox

Wie Roland Fischer in seinem Buch "Mensch und Mathematik" [Fischer, 1985, S 47f] ausführt, hat die elementare Algebra sowohl einen formalen als auch einen inhaltlichen Aspekt. Mathematik lebt sowohl von der Trennung als auch von der Verknüpfung dieser Aspekte. Auch wenn die temporäre Trennung des Formalen vom Inhaltlichen eine charakteristische Methode der Mathematik und eine ihrer Stärken ist, muß doch kritisiert werden, daß ein Kennzeichen der klassischen Schulmathematik eine Überbetonung des formalen Aspekts und die fehlende Verknüpfung mit dem inhaltlichen Aspekt ist.

Die Verwendung von CAS bietet die Chance, den Lernenden im formalen Bereich Tätigkeiten abzunehmen, durch Experimentieren und Testen mehr Verständnis zu erreichen und somit Kapazitäten für den inhaltlichen Bereich und für die Verknüpfung des Formalen mit dem Inhaltlichen freizumachen.

Beispiel 4.1: Strukturerkennungsübungen

Verschiedene Untersuchungen, insbesondere die von Günther Malle [Fischer, 1985, S 59f] bestätigen, daß zu den häufigsten Fehlerquellen beim Termrechnen Strukturerkennungsfehler gehören. Durch das Arbeiten mit dem CAS gewinnt die Strukturerkennungskompetenz noch mehr an Bedeutung, da der Schüler vom Ausführer zum Planer wird, sich also aufgrund der erkannten Termstruktur für eine Lösungsstrategie entscheiden muß, bzw. basierend auf seinem Wissen um Termstrukturen einen Term entwikeln muß. Die bisher durch intensives Üben von Rechenfertigkeiten erworbene implizite Strukturerkennungskompetenz ist aber bei Verzicht auf Rechendrill nicht mehr gegeben (oder war, wie viele Untersuchungen zeigen, nie vorhanden), daher muß man die Möglichkeiten des CAS zum Erwerb dieser Kompetenz nutzen.

Eine Möglichkeit ergibt sich bei DERIVE durch Markieren von logisch korrekten Teiltermen mit Hilfe der Cursorsteuerung. Der Schüler kann durch experimentelles Arbeiten den Aufbau des Terms erforschen und außerdem wird diese Strukturerkennungskompetenz auch durch eine größere Zahl von Übungsaufgaben, die am Computer leicht möglich sind, verbessert.

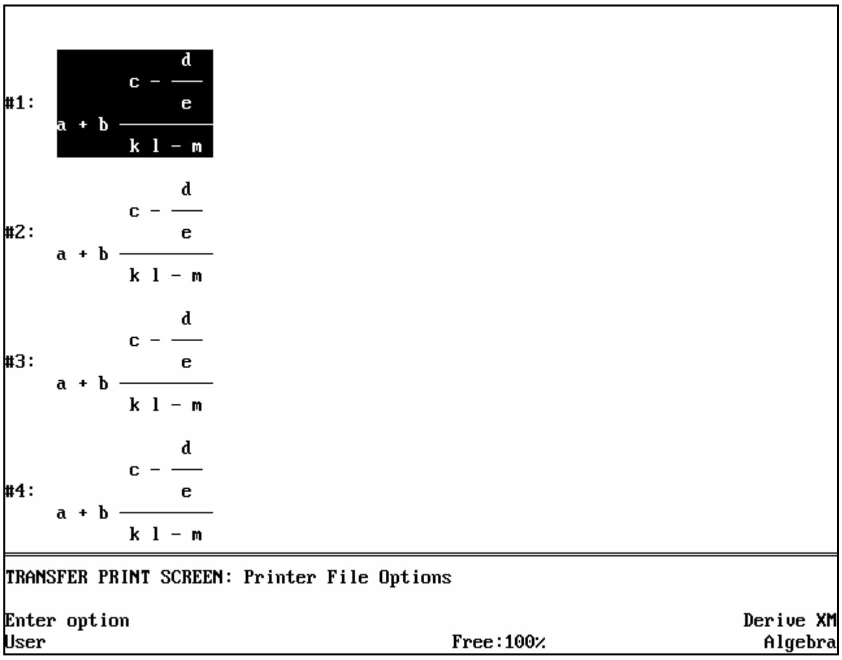

Abb. 4.1:Strukturerkennungsübungen

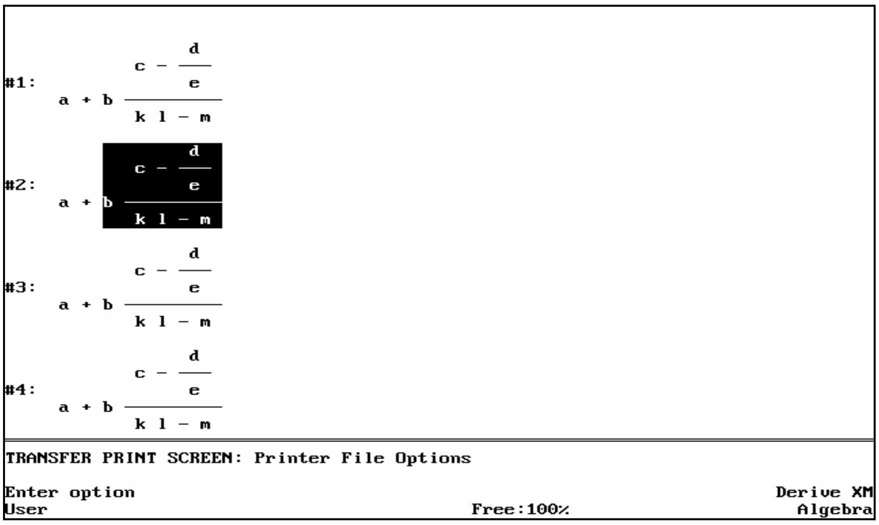

Abb. 4.2:Strukturerkennungsübungen

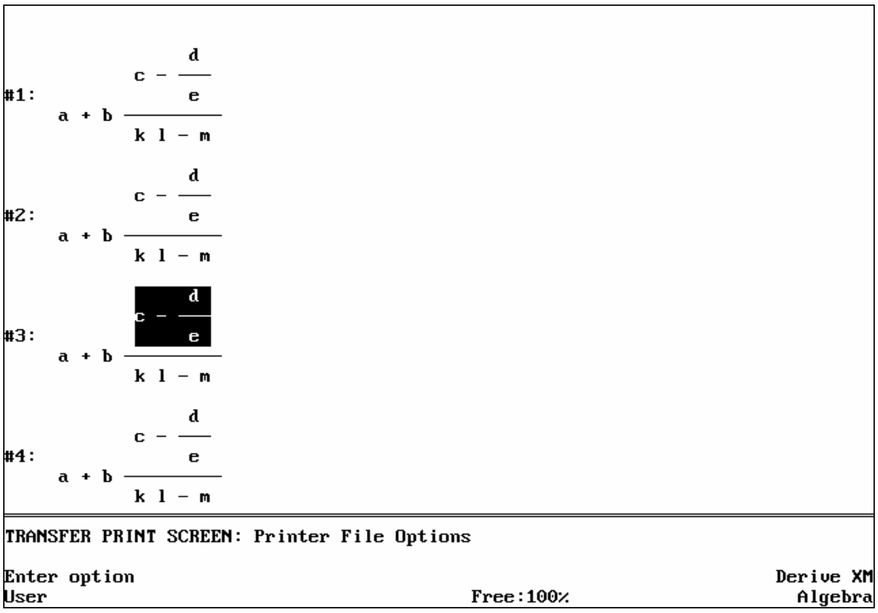

Abb. 4.3:Strukturerkennungsübungen

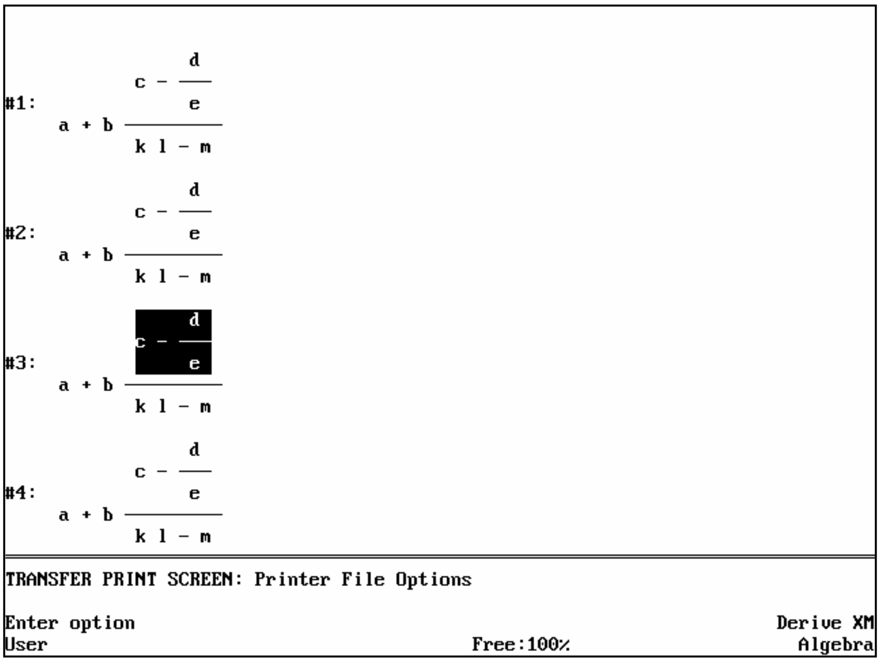

Abb. 4.3:Strukturerkennungsübungen

Didaktisch gesehen es empfiehlt sich, die Terme nicht in einem File anzubieten, sondern vom Schüler selbst eingeben zu lassen. Aus der Not der linearen Eingabe wird dabei insofern eine Tugend, als der Schüler schon vor der Eingabe eine Strukturentscheidung treffen muß. Während er bei der Eingabe in einen numerischen Taschenrechner seine Strukturerkennungsfehler nicht mehr entdeckt, sieht er bei der Eingabe in DERIVE sofort die falsche Struktur. Eine weitere bewährte Begleitmaßnahme, die das verstehende Lernen fördert und eine Verküpfung zwischen der Strukturerkennungskompetenz und der Grundgesetzerkennungskompetenz herstellt, ist der Auftrag, die beim Eindringen in die Termstruktur auftretenden Rechengesetze und Grundgesetze als Begründung anzugeben.

4.1.5. Termumformungen

Der Vorteil des CAS in der White Box-Phase besteht darin, daß der Schüler selbsttätig Bearbeitungstechniken erforschen und individuelle Techniken entwickeln kann. Folgende Techniken werden bei der Nutzung des CAS eingesetzt:

- Unterlegen von Ausdrücken und Teilausdrücken (mit den Cursortasten im Algebrafenster).
- Anwenden von Befehlen auf Ausdrücke und Teilausdrücke (**Simplify, Factor, Expand**).
- Ersetzen komplizierterer Ausdrücke durch einfachere (**Manange Substitute**).
- Rückführen von vereinfachten Ausdrücken in die komplexe Form (**Manage Substitute**).

Beispiel 4.2: Umformen in ein Produkt ("Faktorisieren")

Zerlege in Faktoren: $(3.m + 1)^2 - (2.m - 3)^2$

Im Sinne des Black Box/White Box-Prinzips könnte der Schüler die Aufgabe zuerst dem Computer übertragen und das CAS als Black Box nutzen:

```
#1:  InputMode := Word

#2:  CASeMode := Sensitive
                2          2
#3:  (3 m + 1)  - (2 m - 3)
```

Mit **Factor** entsteht folgender Ausdsruck:

```
#4:  (m + 4) (5 m - 2)
```

Nun versucht der Schüler, die Black Box "white" zu machen. Die erste Tätigkeit ist eine Strukturerkennung. Der gegebene Term scheint die Struktur des Typs $a^2 - b^2$ zu haben. Mit **Manage Substitute** wird nun der erste Klammerausdruck durch die Variable *AUSDRUCK1* erstetzt.

```
                   2             2
#5:  AUSDRUCK1  - (2 m - 3)
```

Nun wird (2 *m* - 3) unterlegt und durch *AUSDRUCK2* ersetzt.

```
                   2             2
#6:  AUSDRUCK1  - AUSDRUCK2
```

Mit **Factor** wird folgende Formel angewendet: $a^2 - b^2 = (a + b)(a - b)$

```
#7:  (AUSDRUCK1 + AUSDRUCK2) (AUSDRUCK1 - AUSDRUCK2)
```

Die Ausrücke aus #4 werden wieder substituiert.

```
#8:  ((3 m + 1) + (2 m - 3)) ((3 m + 1) - (2 m - 3))
```

Durch schrittweise Vereinfachung mit **Simplify** entsteht:

```
#9:  (5 m - 2) ((3 m + 1) - (2 m - 3))

#10: (5 m - 2) (m + 4)
```

Jetzt müßten natürlich ausreichend viele Übungen zur Festigung des Gelernten eingesetzt werden, wobei das CAS den Vorteil bietet, daß in kurzer Zeit viele solcher Beispiele geübt werden können und daß das CAS zum Testen und für Proben genutzt werden kann.

Beispiel 4.3: Anwenden von Formeln

Übungen zum Lernziel: Ergänzen zu einem vollständigen Quadrat.

Es geht in dieser Lernsequenz um die folgenden Formeln:

```
                 2    2             2
#1:  (u + v)  = u  + 2 u v + v                         User

                 2    2             2
#2:  (u - v)  = u  - 2 u v + v                         User

      2    2
#3:  u  - v  = (u + v) (u - v)                         User
```

Auf Arbeitsblättern werden den Schülern folgende Aufgaben gestellt, die sie dann mit dem CAS durch experimentelles Lernen lösen sollen:

Ermittle *a,b,c* so, daß eine der obigen Formeln paßt:

```
          2                   2
#4:  4 x  + a + 25 = (b + c)                           User
```

Mit Hilfe von **Manage Substitute** werden die Variablen durch die vermuteten Ausdrücke ersetzt. Natürlich geht es bei selbständigem Arbeiten nicht so rasch wie hier, aber gerade die Möglichkeit des Fehlersuchens mit dem CAS, das Diskutieren über Strategien und das Finden eigener Strategien ist in dieser Form nur mit dem CAS möglich und rechtfertigt den Einsatz in der White Box-Phase.

$$\#5: \quad 4x^2 + 20x + 25 = (2x + 5)^2 \qquad \text{Sub(\#4)}$$

Einige Testmöglichkeiten: Faktorisieren des linken Ausdrucks:

$$\#6: \quad (2x + 5)^2 = (2x + 5)^2 \qquad \text{Fctr(\#5')}$$

Lösen der Gleichung nach der Variablen *x:*

$$\#7: \quad x = @1 \qquad \text{Solve(\#6)}$$

Natürlich sollte man auch Aufgaben stellen, die nicht so reibungslos funktionieren:

$$\#13: \quad 4x^2 + 2xy + a = (b + c)^2 \qquad \text{User}$$

$$\#14: \quad 4x^2 + 2xy + y^2 = (2x + y)^2 \qquad \text{Sub(\#13)}$$

Die Testmöglichkeiten mit **Factor** oder **Expand** zeigen, daß die Vermutung falsch ist:

$$\#15: \quad 4x^2 + 2xy + y^2 = (2x + y)^2 \qquad \text{Fctr(\#14)}$$

$$\#16: \quad 4x^2 + 2xy + y^2 = 4x^2 + 4xy + y^2 \qquad \text{Expd(\#15')}$$

Das CAS hilft, die richtige Lösung zu finden:

$$\#17: \quad 2xy = 2(2x)c \qquad \text{User}$$

$$\#18: \quad c = \frac{y}{2} \qquad \text{Solve(\#17)}$$

Man beobachtet wieder: Die Haupttätigkeit des Lernenden besteht im Formulieren von Vermutungen und im Testen.

$$\#19: \quad 4x^2 + 2xy + \frac{y^2}{4} = \left[2x + \frac{y}{2}\right]^2 \qquad \text{Sub(\#13)}$$

$$\#20: \quad 4x^2 + 2xy + \frac{y^2}{4} = 4x^2 + 2xy + \frac{y^2}{4} \qquad \text{Expd(\#19)}$$

4.1.6. Die Gleichungsbox

Wenn man im Sinne des von Roland Fischer [Fischer, 1985, S 63ff] vorgeschlagenen Wegs in den Schulstufen 5 und 6 den Formalismus des Gleichungslösens vorsichtig aus inhaltlichen Überlegungen heraus entwickelt hat, so ist es spätestens in der 7. Schulstufe notwendig, jenen Schritt zu tun, der eine der großen Stärken der Mathematik ist, nämlich die Trennung des Formalen vom Inhaltlichen. Ich halte dabei die Idee der Äquivalenzumformung nach wie vor für die wirkungsvollste Strategie, weil sie für die Schüler eine weit über das Lösen linearer Gleichungen hinaus bedeutsame heuristische Strategie darstellt: "Unter Einhaltung bestimmter Spielregeln auf beiden Seiten dasselbe tun."

Die große Chance, die sich aus der Verwendung von CAS ergibt, besteht darin, daß der Schüler seine individuellen Strategien durch experimentelles Lernen selbst entdecken kann. Untersucht man Schülerfiles, so kann man selbst in der Prüfungssituation erkennen, daß von einem "algorithmischen Gehorsam", wo jeder Schüler den vom Lehrer vorgestellten Weg nachvollzieht, bei Einsatz des CAS nichts mehr zu sehen ist.

Einsatzmöglichkeiten des CAS in der Gleichungs-White-Box:

- Schrittweise Durchführung von Umformungen. Dabei können die beiden Seiten der Gleichung entweder einzeln oder gleichzeitig bearbeitet werden. Bei DERIVE kann die Gleichung oder Teile davon mit **F3** oder **F4** in den Editor geholt und bearbeitet werden.
- Analyse und Interpretation der dabei entstehenden neuen äquivalenten Gleichung. Eventuell Rückführen in die ursprüngliche Form als Probe.
- Vergleichen verschiedener Äquivalenzumformungen, Finden der individuell am geeignetsten Strategie. Erkennen falscher bzw. unwirksamer Strategien.
- Ständiges begleitendes Testen, Probe durch Einsetzen vermuteter Lösungen ohne großen Rechenaufwand.
- Herstellen einer Querverbindung zum Funktionsbegriff: Visualisierung der Äquivalenzumformung. Veranschaulichen der Konsequenz nicht äquivalenter Umformungen.

Beispiel 4.4: Vergleichen von Umformungsstrategien

Die Window-Shuttle-Technik (siehe Kap.4.4) erlaubt noch dazu das parallele Arbeiten in verschiedenen Fenstern, in diesem Fall das Vergleichen der Wirksamkeit verschiedener Lösungsstrategien.

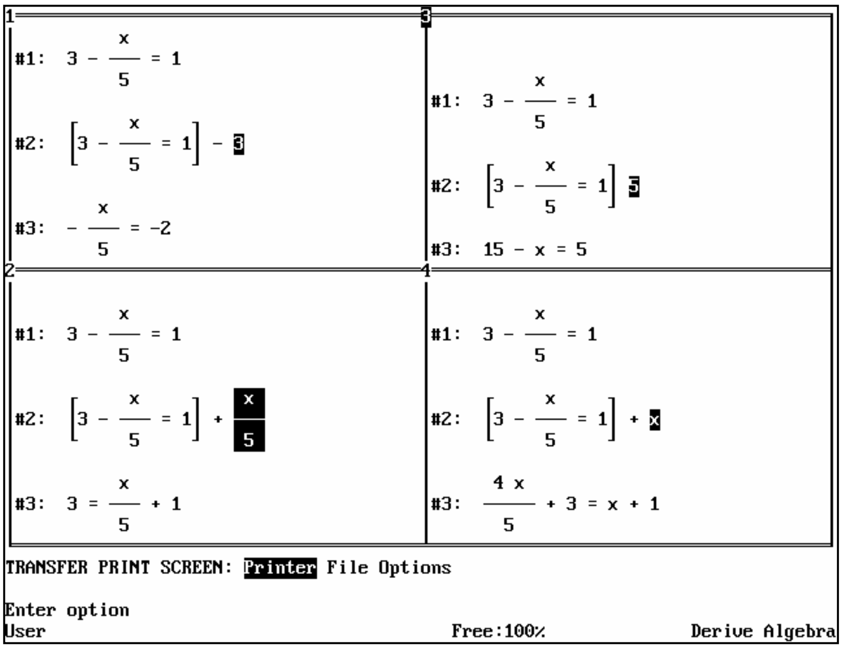

Abb. 4.4:Gleichungsbox

Wie im Fenster 4 zu sehen ist, kann der Schüler auch die Unwirksamkeit falscher Strategien erkennen, was er beim Arbeiten auf Papier wahrscheinlich nicht bemerken würde. Dort führt z.B. bei der Gleichung $3.x = 12$ die Strategie "auf beiden Seiten 3 subtrahieren" zum falschen Ergebnis $x = 9$, beim Arbeiten mit dem CAS dagegen zu $3.x - 3 = 9$.

Beispiel 4.5: Visualisierung von Äquivalenzumformungen

Deutet man eine Gleichung der Form L(x) = R(x) als Schnitt zweier Funktionen L und R, so kann man durch "Shutteln" zwischen einem Algebra- und einem Grafikfenster die Konsequenzen von Äquivalenzumformungen beobachten.

Durch die Beobachtung gemeinsamer Eigenschaften der Graphen der Zeilen #2 und #4 in der folgenden Abbildung kommt man zum Schluß: Der Schnitt erfolgt stets an der Stelle x=1. Die Überprüfung erfolgt auch durch Zeichnen von [1,t] (Zeile #5). Im Algebrafenster könnte man dann noch die Probe für die aus dem Grafikfenster gewonnene Vermutung machen (Abb.4.6 und Abb.4.7).

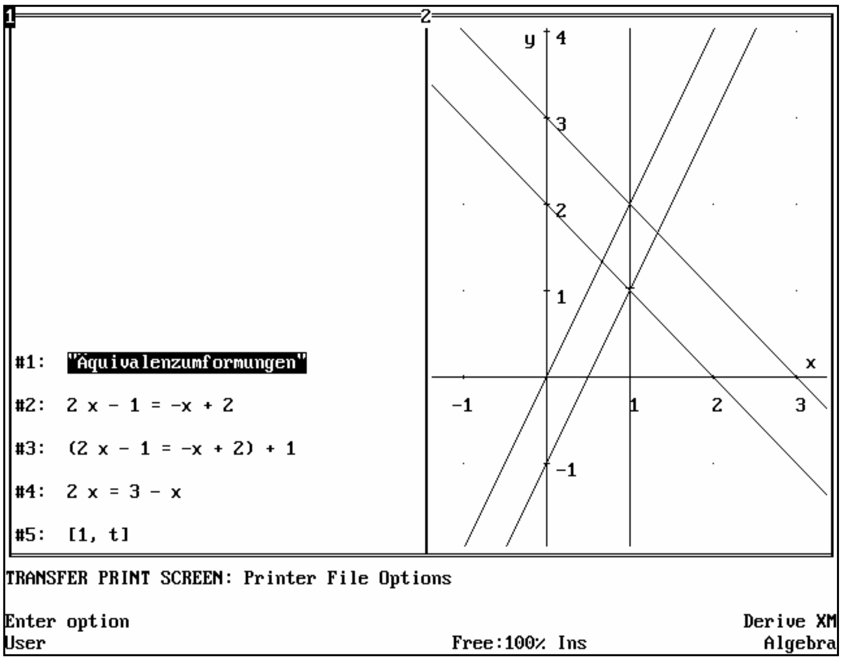

Abb. 4.5:Visualisierung von Äquivalenzumformungen

Bei weiteren Äquivalenzumformungen kommt man zum selben Ergebnis:

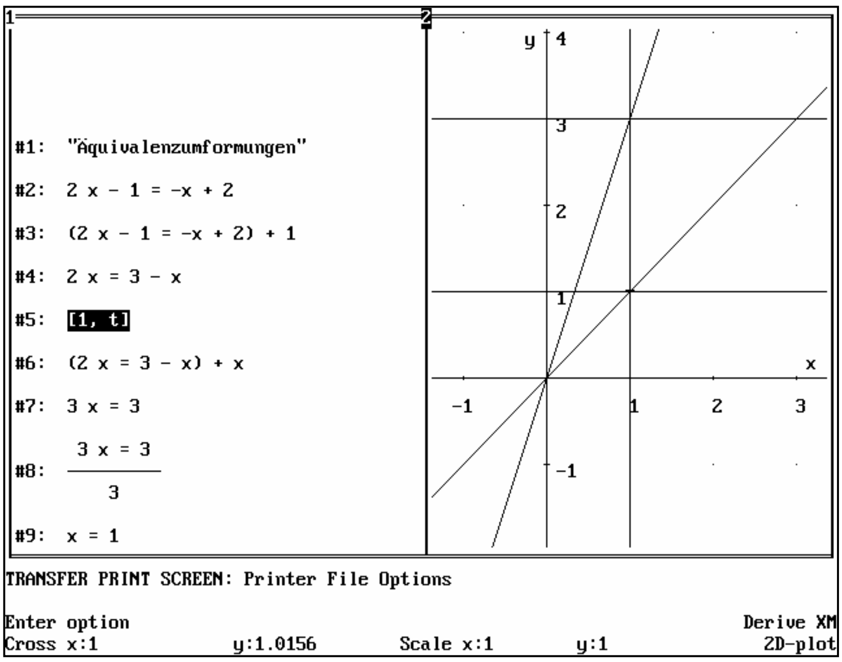

Abb. 4.6:Visualisierung von Äquivalenzumformungen

Auch die Folgen nicht äquivalenter Umformungen können im Grafikfenster beobachtet werden, wie etwa die Multiplikation mit x oder das Quadrieren der Gleichung (Abb.4.8 und Abb.4.9).

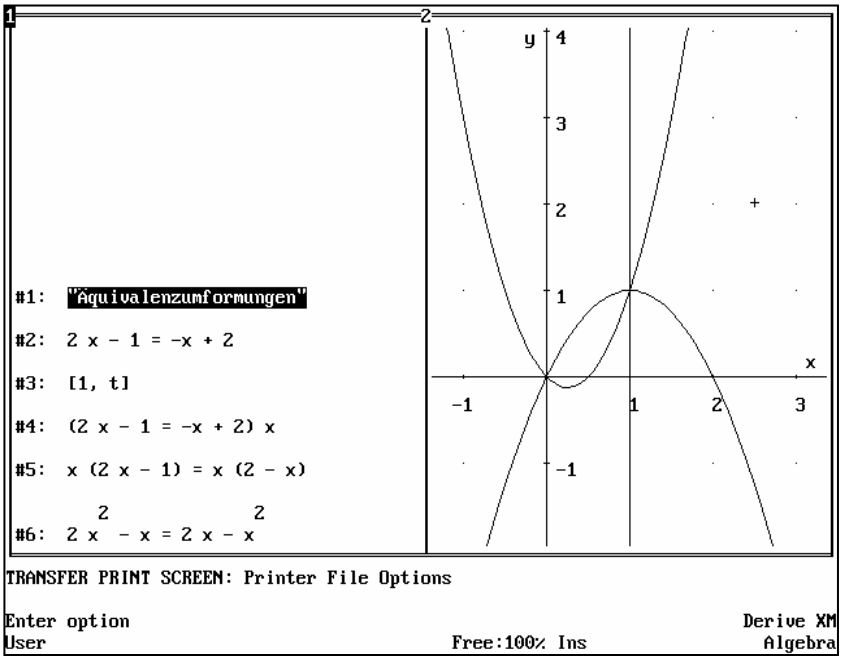

Abb. 4.7:Visualisierung von Äquivalenzumformungen

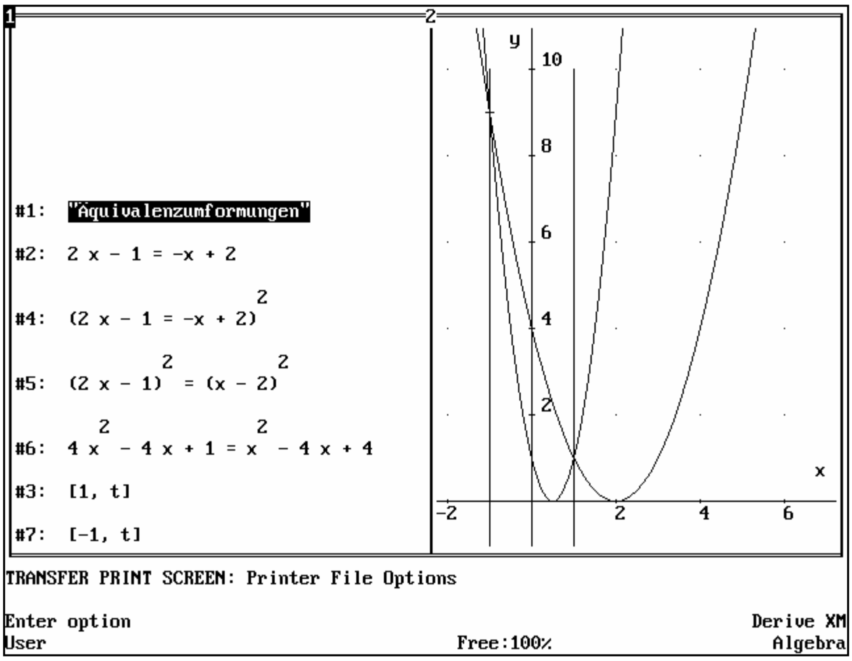

Abb. 4.8:Visualisierung von Äquivalenzumformungen

Von didaktischer Bedeutung ist bei dieser Vorgangsweise auch die bei den neuen DERIVE-Versionen mögliche "*Lehrerschnittstelle*":

Der Lehrer kann das Menü auf die Lernentwicklung der Schüler abstimmen und entscheidet, welche Einstellungen im Hauptmenü erscheinen. So bietet man in der White Box-Phase der elementaren Algebra etwa nur die Befehle **Simplify**, **Expand**, **Factor**, **Manage Substitute**, **Approx** und **Plot** an, nicht aber **soLve**. An manchen Schulen, die am CAS-Projekt beteiligt sind, gibt es im Netz des EDV-Raums schon Menüvoreinstellungen für die einzelnen Lernstufen von der 7. bis zur 12. Schulstufe. Im Kapitel 5.2.2 wird darauf noch näher eingegangen.

4.1.7. Die Box: Gleichungssysteme

Im Sinne des White Box/Black Box-Prinzips sollten in der White Box-Phase folgende Ziele angestrebt werden:

- Den Begriff des linearen Gleichungssystems erläutern können. Den Begriff der Lösungsmenge erläutern und verschiedene Lösungsmengen deuten können.
- Lösungsalgorithmen entwickeln und erklären können.
- Die entwickelten Algorihmen für einfache Fälle 'zu Fuß', d.h. ohne Computereisatz anwenden können.
- Modellbilden: Gleichungssysteme aufstellen können.
- Lösungsmengen interpretieren können.
- Den Lösungsalgorithmus des CAS anwenden können.

Lineare Gleichungssysteme in der 8.Schulstufe:

Natürlich kann man die üblichen Lösungsalgorithmen, wie etwa die grafische Lösung, das Einsetzungs- oder das Gleichsetzungsverfahren usw. auch an Hand einfacher Beispiele "zu Fuß" behandeln. Hier sollen an Beispielen einige Argumente angeführt werden, die für den Einsatz von CAS auch in dieser Phase sprechen.

Beispiel 4.6: Lösungsverfahren für Gleichungssysteme

Grafisches Lösen:

Behandeln vieler Aufgaben in relativ kurzer Zeit. Konzentrieren auf das Wesentliche des Lösens, Untersuchen verschiedener Lösungsfälle. Keine Ablenkung durch Nebentätigkeiten, wie Aufstellen von Wertemengen. Experimentelles Entdecken der Lösung durch "Wandern auf den Geraden im *Trace*-Modus von DERIVE". Nutzen der **Zoom**-Möglichkeit. Gemäß der Window-Shuttle-Technik: Pendeln zwischen dem Algebra- und Grafikfenster, Beobachten von Konsequenzen algebraischer Operationen im Grafikfenster.

Einsetzungsverfahren (Substitutionsverfahren):

```
#1:  3.x - y = 5                                          User

#2:  2.x + 3.y = 7                                        User
```

Die Auflösung der Gleichung #1 nach y erfolgt mit dem CAS als Black Box.

```
#3:  y = 3.x - 5                                     Solve(#1)
```

Die Tätigkeit des Substituierens wird durch die Nutzung der entsprechenden DERIVE-Optionen, wie etwa das Verwenden von **Manage Substitute**, bewußt wahrgenommen. Der Schüler entscheidet die Operation, führt sie aber selbst nicht aus.

```
#4:  2.x + 3.(3.x - 5) = 7                             Sub(#2)
```

Wieder übernimmt das CAS Lerninhalte früherer White Boxes als Black Box

```
#5:  x = 2                                           Solve(#4)

#6:  y = 3.2 - 5                                       Sub(#3)
```

```
#7:  y = 1                                             Simp(#6)
```

Gleichsetzungsverfahren:

```
#1:  3.x - y = 5                                          User

#2:  2.x + 3.y = 7                                        User

#3:  y = 3.x - 5                                     Solve(#1)

              7 - 2.x
#4:  y =   ─────────                                 Solve(#2)
                 3
```

Das bewußt durchgeführte "gleich - Setzen" erfolgt bei DERIVE durch Verwendung der Funktionstasten **F3** bzw. **F4**, mit denen markierte Ausdrücke in die **AUTHOR**-Zeile kopiert werden können. Dadurch wird die neue Gleichug durch Gleichsetzen zweier Terme gebaut. Das Lösen dieser Gleichung könnte im Sinne des White Box/Black Box-Prinzips vom CAS als Black Box übernommen werden. Eine andere Möglichkeit wäre die Verwendung des Build-Befehls, dessen Name schon seine Möglichkeiten beschreibt.

```
                    7 - 2.x
#5:  3.x - 5 =   ─────────                                User
                       3

#6:  x = 2                                           Solve(#5)

#7:  y = 3.2 - 5                                       Sub(#3)

#8:  y = 1                                            Simp(#7)
```

Additionsverfahren (Eine Vorform des Gaußschen Eliminationsverfahrens):

Wie schon in der Gleichungsbox ausgeführt könnte sich der Schüler auf das Entscheiden der Strategie konzentrieren und die Ausführung der Äquivalenzumformungen der beiden Gleichungen dem CAS überlassen.

Lösen mit dem CAS:

Bei Verwendung von DERIVE wird das Gleichungssystem als Vektor eingegeben. Mit **soLve** erhält man sofort die gesuchte Lösung.

Lineare Gleichungssysteme in der 10. Schulstufe

Eine wesentliche Idee des Spiralprinzips besagt, daß eine Box, die in der 8. Schulstufe einmal "white" war, danach nicht für immer "black" bleiben muß, oder besser gesagt nicht "black" bleiben soll. In Österreich steht im Lehrplan der 10. Schulstufe des Realgymnasiums das Thema "Matrizen". Bis jetzt wurde es sehr stiefmütterlich behandelt. Durch Nutzung des CAS könnte es einen wesentlich höheren Stellenwert bekommen, da die aufwendige Rechenarbeit beim Rechnen mit Matrizen das CAS übernehmen könnte. Zusätzlich kann dieses Kapitel für eine White Box-Phase zum Erforschen des Gaußschen Eliminationsverfahrens genutzt werden. Somit wird das Thema "Lösen von Gleichungssystemen" in verschiedenen Altersstufen auf verschiedenen Exaktheitsniveaus behandelt. Der Unterschied zum traditionellen Spiralprinzip besteht im Nutzen des CAS einerseits zur Erhellung der neuen White Box und andererseits als Rechenhilfe bei Verwendung von Inhalten früherer White Box-Phasen.

Entsprechend der am Beginn der Box "Gleichungssysteme" formulierten Grobziele wären die Feinziele dieser White Box-Phase [Lehmann, 1990]:

-Kenntnisse über das Rechnen mit Matrizen beim Lösen von Gleichungssystemen anwenden können.

-Die Zielsetzung des Gauß Algorithmus erläutern können und die Lösung schrittweise mit Hilfe von DERIVE-Operationen für Matrizen berechnen können. Zwischenergebnisse deuten können. Die Auswirkung der DERIVE-Befehle und den dazugehörigen Algorithmus deuten können.

- Die Umformung des Gleichungssystems mit dem Derive-Befehl row_reduce durchführen können.
- Das bei den "Gaußschen" Umformungen entstehende Endschema auswerten können.

- Das Ablesen der Lösungsmenge aus dem Endschema begründen können.

4.1.8. Die Anwendungsbox

In dieser Box ist das Thema der White Box: Schulung des Problemlösens. Dabei können die in früheren White Box-Phasen erworbenen Kenntnisse im Bereich von Algebra oder Analysis unter Nutzung des CAS als Black Box eingesetzt werden. Bisher war in der Anwendungsphase (sofern sie überhaupt vorhanden war) der Schwerpunkt der Schülertätigkeit das Rechnen. Nun kann gerade das *Operieren* dem CAS übertragen werden und das *Modellbilden* sowie das *Interpretieren* gewinnen an Bedeutung.

Beispiel 4.7: Eine klassische Aufgabe in traditionellen Schulbüchern

Von einer Polynomfunktion 3. Grads kennt man den Tiefpunkt T(2/-1) und den Wendepunkt W(1,2). Ermittle die Funktionsgleichung, diskutiere die Funktion und fertige eine Zeichnung an.

Bisher war die Hauptarbeit zur Ermittlung der Funktionsgleichung das Lösen des zugehörigen Gleichungssystems. Das übernimmt jetzt das CAS als Black Box. Die Tätigkeit des Schülers konzentriert sich auf das eigentliche Ziel dieser Analysisaufgabe: *Das Modellbilden.*

#1: $F(x) := a\cdot x^3 + b\cdot x^2 + c\cdot x + d$ User

#2: $\left[\frac{d}{dx}\right]^1 F(x) = 3\cdot a\cdot x^2 + 2\cdot b\cdot x + c$ User=Simp(User)

#3: $\left[\frac{d}{dx}\right]^2 F(x) = 6\cdot a\cdot x + 2\cdot b$ User=Simp(User)

#4: $F1(x) := 3\cdot a\cdot x^2 + 2\cdot b\cdot x + c$ User

#5: $F2(x) := 6\cdot a\cdot x + 2\cdot b$ User

Die eigentliche Modellbildungstätigkeit besteht im ermitteln der Zeile #6:

#6: $[F(2) = -1, F(1) = 2, F1(2) = 0, F2(1) = 0]$ User

Das Lösen übernimmt das CAS:

#7: $\left[a = \frac{3}{2}, b = -\frac{9}{2}, c = 0, d = 5\right]$ Solve(#6)

Diese Aufgabe haben wir nicht ausgewählt, weil wir sie in einem computerunterstützten Mathematikunterricht der Zukunft für so wichtig halten, sondern um zu zeigen, wie sich die Schwerpunkte verschieben, und was von den bisherigen Schülertätigkeiten übrig bleibt.

4.1.9. Zusammenfassung: CAS in der Algebra

Im traditionellen Mathematikunterricht ist wahrscheinlich in allen Ländern der Welt die Rechenfertigkeit das dominierende Ziel. Immer dann, wenn in der Geschichte der Mathematik neue Rechenhilfsmittel oder neue Rechenverfahren entwickelt wurden, mußte die bis dahin forcierte Kalkülkompetenz überdacht werden. Nun sind wir wieder an so einem entscheidenden Punkt der mathematischen Entwicklung angelangt: Die Nutzung des Computers und insbesondere der Computeralgebra-Systeme verlangt ein Überdenken der bisher im Mittelpunkt stehenden Rechenfertigkeiten.

Folgende Kompetenzen haben sich bei unseren Experimenten als wichtig erwiesen -man könnte von einer *algebraischen Allgemeinbildung* sprechen:

Strukturerkennungskompetenz:

Sie ist bei der Entwicklung eines Terms, bei der Eingabe und bei der Entscheidung für eine bestimmte Operation aber auch beim Interpretieren und Testen von Ergebnissen notwendig.

Äquivalenzerkennungskompetenz:

Besonders zu beachten ist die Schnittstelle Operieren - Interpretieren: Der Schüler muß Ergebnisse interpretieren, die er nicht selber produziert hat. Oft liefert das CAS unerwartete Resultate und der Schüler weiß nicht, ob sie zu seinen erwarteten Ergebnissen äquivalent oder von diesen verschieden sind. Auch die individuellen Ergebnisse verschiedener Schüler beim experimentellen Lernen müssen oft in bezug auf Äquivalenz überprüft werden. Daher müssen die Schüler Teststrategien zur Steigerung dieser Kompetenz entwikeln.

Grundgesetzanwendungskompetenz:

Gerade das White Box/Black Box-Prinzip erfordert diese Begründungskompetenz in der White Box-Phase als wesentlichen Bestandteil des verstehenden Lernens.

Elementare Kalkülkompetenz:

Einig sind wir nur, daß eine solche Kompetenz nach wie vor auch bei intensiver Nutzung von CAS notwendig ist, da ja sonst auch die Entscheidung für ein bestimmtes Modell, für eine bestimmte Operation oder für einen bestimmten Beweisschritt nicht möglich wäre. Wie aber diese Kompetenz im Detail aussieht, wissen wir noch nicht. Das wird Gegenstand vieler Untersuchungen und Diskussionen sein.

Testkompetenz:

Gerade bei dem durch die Verwendung von CAS möglichen experimentellen Lernen wird das Testen eine zentrale Tätigkeit. Auch die verstärkte Bedeutung des Modellbildens und des Interpretierens erfordert diese Kompetenz. Eine unstrukturierte Versuch und Irrtumsmethode führt aber kaum zum Ziel. Es ist die bewußte Aneignung heuristischer Teststrategien seitens der Schüler notwendig.

Visualisierungskompetenz:

Das Prinzip der Anschaulichkeit ist nicht neu. Ohne Computer war das Veranschaulichen aber oft sehr aufwendig. Die Verwendung von CAS bietet für den Erwerb und die Nutzung dieser Kompetenz völlig neue Möglichkeiten. Visualisieren mit dem CAS ist nicht nur ein Beitrag zu einem besseren Verständnis, sondern stellt auch eine neue Form des Operierens dar und erlaubt ein besseres Interpretieren von Ergebnissen.

4.2. Das Black Box/White Box-Prinzip

Wie schon der Name sagt werden beim Unterrichten nach diesem Prinzip die White und die Black Box-Phase vertauscht. Unterstützt vom CAS als Black Box erforscht der Schüler einen ihm nicht bekannten Bereich der Mathematik durch experimentelles, aktives Lernen und kommt zu Vermutungen, welche in der folgenden White Box-Phase abgesichert werden.

Bei dieser Arbeitsweise erweist sich das CAS als ein völlig neues Medium für den operativen Mathematikunterricht. Der Schüler kann mit Hilfe des CAS als "Expertensystem" real und gedanklich operieren, er kann durch experimentelles Arbeiten zu Einsichten kommen.

4.2.1. Lernphasen bei Anwendung des Black Box/White Box-Prinzips:

Phase 1: Black Box-Phase

Der Schüler erforscht einen ihm unbekannten Bereich der Mathematik mit Hilfe des CAS als Black Box durch Experimentieren, wie zum Beispiel:

- Einen neuen Algorithmus durch Ausführen mit dem CAS als Black Box und durch experimentelles Zurückführen auf Bekanntes.
- Beispiel: Entdecken und Erforschen der Kettenregel beim Differenzieren.

- Eine Operation oder eine Funktion, die das CAS zur Verfügung stellt (z.B.: DIFFERENTIATE, LIMES, TAYLOR, VECTOR, usw.).
- Beispiel: Erforschen der Eigenschaften und Auswirkungen der Taylorfunktion von DERIVE in der Black Box-Phase als Vorbereitung auf die Taylorreihenentwicklung in der White Box-Phase [vgl. Barzel, 1991].
- Ein Black Box Modul, den der Lehrer entwickelt hat.
- Ein vom Schüler bearbeitetes File, in dem die vom CAS als Black Box ausgeführten Operationen hinterfragt werden ("was der Computer kann, können wir auch").
- Beispiel: Siehe "White Box/Black Box-Prinzip in der Algebra": Am Beginn der White Box-Phase verwendet der Schüler die Befehle **Factor**, **Expand** oder **soLve** und erforscht danach die Auswirkung der Befehle durch Experimentieren.

Typische Arbeitsweisen:

- Ausführen von noch unbekannten CAS-Befehlen
- Untersuchen der Auswirkungen der Befehle
- Wechsel der Darstellungsform (Algebra-, Grafikfenster)
- Zurückführen auf Bekanntes
- Betrachten von Sonderfällen
- experimentelles schrittweises Nachvollziehen von BLACK BOX - Operationen, wie etwa **soLve** oder **Simplify**, durch Nutzen bekannter Operationen, wie **Expand** oder **Factor**.

Phase 2: White Box-Phase

Durch das operative Bearbeiten der Black Box soll der Schüler zu Einsichten über die zugrundeliegenden mathematischen Begriffe, Algorithmen oder Theorien kommen. Die in der Black Box-Phase gewonnenen Vermutungen sollen nun in der White-Phase begründet und abgesichert werden. Ausreichendes Üben, vielfältiges Anwenden, Begründen und Beweisen sollen die White Box stabilisieren. In weiteren, in der Spirale folgenden Phasen können dann die Inhalte dieser White Box als Black Box genutzt werden.

Beispiel 4.8: Die Kettenregel beim Differenzieren

Voraussetzungen sind Kenntnis der Begriffe: "Verkettung von Funktionen", Grenzwert reeller Funktionen, Differenzenquotient, Differentialquotient. Differentiationsregeln für Potenzfunktionen und Polynomfunktionen.

Black Box-Phase

#1: $(3.x^2 + 7)^5$ User

Läßt man Schüler "demokratisch" Ergebnisse der Ableitung dieser Funktion vermuten, so finden sich am häufigsten die Antworten: $5.(3.x^2 + 7)^4$ und $5.(6.x + 7)^4$.

#2: $\frac{d}{dx}(3\cdot x^2 + 7)^5$ Dif(#1,x)

Bei Nutzung des CAS als Black Box ergibt sich für viele Schüler ein unerwartetes Ergebnis:

#3: $30.x.(3.x^2 + 7)^4$ Simp(#2)

Nun stellt sich die Frage, wie dieses Ergebnis zu erklären ist, oder besser gesagt, welche Regel dahinter steckt. Ein erster Schritt ist häufig die Anwendung der heuristischen Regel: "Führe auf Bekanntes zurück". Bekannt ist die Ableitung von Polynomfunktionen. Mit **Expand** erreicht man diese Zurückführung:

Expd(#1)

$$\#4:\ 243\cdot x^{10} + 2835\cdot x^{8} + 13230\cdot x^{6} + 30870\cdot x^{4} + 36015\cdot x^{2} + 16807$$

Die Ableitung könnte der Schüler selbst ausführen, im Sinne des White Box/Black Box-Prinzips kann die Arbeit aber auch dem CAS überlassen werden.

Dif(#4,x)

$$\#5:\ \frac{d}{dx}\left(243\cdot x^{10} + 2835\cdot x^{8} + 13230\cdot x^{6} + 30870\cdot x^{4} + 36015\cdot x^{2} + 16807\right)$$

$$\#6:\ 2430\cdot x^{9} + 22680\cdot x^{7} + 79380\cdot x^{5} + 123480\cdot x^{3} + 72030\cdot x \qquad \text{Simp(\#5)}$$

Dieser Ausdruck läßt natürlich noch keine Schlüsse auf Regeln zu. Erst durch Faktorisieren erhält man die Bestätigung, daß auch bei Zurückführen auf Bekanntes dasselbe Ergebnis erhalten wird.

$$\#7:\ 30.x.(3.x^{2} + 7)^{4} \qquad \text{Fctr(\#6)}$$

Um zu einer Vermutung für eine Regel zu kommen, definiert man eine "termlose" Funktion *G*(*x*)

$$\#8:\ G(x) := \qquad \text{User}$$

und differenziert mehrere zusammengesetzte Funktionen:

$$\#9:\ \frac{d}{dx} G(x)^{2} \qquad \text{User}$$

$$\#10:\ 2\cdot G(x)\cdot\frac{d}{dx} G(x) \qquad \text{Simp(\#9)}$$

$$\#11:\ \frac{d}{dx} \text{SIN}(G(x)) \qquad \text{User}$$

$$\#12:\ \text{COS}(G(x))\cdot\frac{d}{dx} G(x) \qquad \text{Simp(\#11)}$$

$$\#13:\ \frac{d}{dx} \sqrt{(G(x))} \qquad \text{User}$$

$$\#14:\ \frac{\frac{d}{dx} G(x)}{2\cdot\sqrt{(G(x))}} \qquad \text{Simp(\#13)}$$

Die Ergebnisse haben alle etwas gemeinsam: Das Produkt aus der Ableitung der"äußeren" Funktion und der Ableitung der "inneren" Funktion *G*.

Versucht man noch weiter zu verallgemeinern, das heißt, definiert man auch die äußere Funktion *F* "termlos", so erhält man wieder ein überraschendes Ergebnis in Form eines Limes, der dann in der White Box-Phase hinterfragt werden kann:

$$\#15:\ F(x) := \qquad \text{User}$$

$$\#16:\ \frac{d}{dx} F(G(x)) \qquad \text{User}$$

$$\#17:\ \left[\frac{d}{dx} G(x)\right] \cdot \lim_{@1 \to G(x)} \frac{d}{d\ @1} F(@1) \qquad \text{Simp(\#16)}$$

Auch der Ersetzen von $G(x)$ durch die Variable z und das Ableiten von $F(x)$ nach x kann als Vorbereitung auf die White Box-Phase dienen:

$$\#18:\ z := G(x) \qquad \text{User}$$

$$\#19:\ \frac{d}{dx} F(z) \qquad \text{User}$$

$$\#20:\ \left[\frac{d}{dx} G(x)\right] \cdot \lim_{@2 \to G(x)} \frac{d}{d\ @2} F(@2) \qquad \text{Simp(\#19)}$$

White Box-Phase

Nun ist der Boden bereitet für die theoretische Absicherung der gewonnenen Vermutung. Dafür gibt es in der Schulbuchliteratur genügend Vorschläge. Der Unterschied zum traditionellen Unterricht besteht eben darin, daß der Schüler schon durch aktives Lernen zu einer Vermutung gekommen ist, also in diese White Box-Phase schon etwas mitbringt und somit die Idee in der exaktifizierenden Phase viel besser verstehen wird wie wenn die theoretische Phase am Anfang des Lernprozesses stehen würde. Ein weiterer Unterschied ergibt sich aus der Möglichkeit, das CAS auch in dieser Phase beim Operieren als Black Box zu nutzen, wodurch sich der Schüler auf das Wesentliche, nämlich auf die logischen Schritte zur Herleitung der Regel konzentrieren kann.

4.3. Das Modulprinzip

Der Modulgedanke ist eine fundamentale Idee der Informatik, die von den CAS übernommen wurde. CAS werden selbst meist modulhaft geliefert: als Kernel und Packages. Man kann weitere Packages/Module/Utility-Files dazuschreiben, in Zeitschriften werden solche ständig veröffentlicht, auf öffentlichen Fileservern sind sie massenhaft zu finden.

Ein CA-Modul kann als die konsequente Weiterentwicklung jener kurzen Programme (*short programs* – meist in Basic, Pascal oder Logo) angesehen werden, die etwa seit der zweiten Hälfte der achziger Jahre verstärkt Einzug in den Mathematikunterricht gehalten haben. Diese Art von Programmen ist auch auf den Begleitdisketten fast aller gängigen österreichischen Mathematiklehrbücher zu finden, und oft sind Quelltexte auch direkt in den Lehrbuchtext eingearbeitet. Sie dienen vielfach Visualisierungszwecken oder zur Implementationen der meist kurzen Algorithmen, die aber für den Taschenrechner zu aufwendig wären.

4.3.1. Was ist ein Modul?

Unter Modularität versteht man, grob gesprochen, die Anwendung des Baukasten-Prinzips bei Problemlöseprozessen. Der Modulgedanke hat viele Quellen. Die zwei bekanntesten seien hier angeführt. R.Descartes faßt im "Discours de la méthode" das Wesentliche an der damals entstehenden Naturwissenschaft in vier Vorschriften zusammen:

- Die erste besagt, niemals eine Sache als wahr anzuerkennen, von der ich nicht evidentermaßen erkenne, daß sie wahr ist: d.h. Übereilung und Vorurteile sorgfältig zu vermeiden und über nichts zu urteilen, was sich meinem Denken nicht so klar und deutlich darstellte, daß ich keinen Anlaß hätte, daran zu zweifeln.
- Die zweite, jedes Problem, das ich untersuchen würde, in so viele Teile zu teilen, wie es angeht und wie es nötig ist, um es leichter zu lösen.
- Die dritte, in der gehörigen Ordnung zu denken, d.h. mit den einfachsten und am leichtesten zu durchschauenden Dingen zu beginnnen, um so nach und nach, gleichsam über Stufen, bis zur Erkenntnis der zusammengesetztesten aufzusteigen, ja selbst in Dinge Ordnung zu bringen, die natürlicherweise nicht aufeinander folgen.
- Die letzte, überall so vollständige Aufzählungen und so allgemeine Übersichten aufzustellen, daß ich versichert wäre, nichts zu vergessen. [Descartes, S.31f]

Verbirgt sich nicht in Punkt zwei und drei der Descartesschen analytischen Methode bereits vollständig das, was wir heute Top-Down-Methode bzw. Bottom-Up-Methode nennen? Und stehen nicht am Ende dieser Teilung eines Problems kleinste Einheiten, einzelne Bausteine, die wir als *Module* bezeichnen können? Diese Module sind also einerseits Endprodukt einer intensiven Auseinandersetzung mit einem Problem, einem Ausschnitt aus der uns umgebenden Realität. Andererseits sind sie aber auch Ausgangspunkt eines konstruktiven Prozesses, bei dem aus diesen kleinsten Wissenspaketen wieder größere Einheiten, größere Module zusammengesetzt werden können.

H.J.Vollrath sieht auch in den Elementen des Euklid bereits Ansätze zum Modulgedanken, wenn er schreibt: "Hier wird also gar nicht mehr auf die Konstruktionen der Postulate zurückgegriffen, sondern nur noch auf bereits gelöste Aufgaben, die man als "Bausteine" verwendet. Betrachtet man Konstruktionsbeschreibungen als Algorithmen, dann benutzt Euklid also bereits Module" [Postel/Kirsch/Blum, 1991, S.220].

Von besonderer Bedeutung und zu außerordentlicher Blüte gekommen ist die Methode der Modularisierung natürlich in der Informatik. Nach J.Ziegenbalg [Ziegenbalg,1984,S.404ff] meint hier Modularisierung insbesondere:

- Zerlegung eines komplexen Problems in einfache, überschaubare Bestandteile (Module). Jedes Modul sollte nach Möglichkeit eine gewisse eigenständige Bedeutung haben.
- Bearbeitung und Lösung der einzelnen Teilprobleme.
- Zusammensetzung der Teillösungen zu einer Gesamtlösung des Ausgangsproblems.

Für die Mathematik hat W. Dörfler [Dörfler, 1991, S.71f] zwei wesentliche Aspekte des Modulgedankens herausgearbeitet und dabei auch auf die weitreichenden Konsequenzen für den Mathematikunterricht hingewiesen.

Demnach können wir Module auffassen als *Wissenseinheiten*,

- in denen (komplexes) Wissen komprimiert wird, und
- in denen Operationen durch diese Kapselung als Ganzes abrufbar und einsetzbar werden.

"Das Modul entspricht einem als kognitive Einheit aufrufbaren kognitiven Schema, das zu seiner Verwendung nicht weiter aufgelöst werden muß, weil genügend "kondensierte" Erfahrung mit der kognitiven Tätigkeit des Schemas vorliegt". [Dörfler, a.a.O.]

Module haben sicher kognitive Entlastungsfunktion, sie tragen zur Reduktion der Komplexität bei, indem sie Abläufe, Tätigkeiten, komplexes Wissen als Einheit handhabbar machen und wie H.G.Weigand [Weigand,1993, S.431] ergänzt, wird "erst durch modulares Arbeiten ... ein effizientes kalkülhaftes Arbeiten ermöglicht".

Dörfler weist auch darauf hin, daß "der »erwachsene« Mathematiker ... gewisse Wissensblöcke als Muster, Vorlagen (templates) in seinem Denken zur Verfügung (hat), er kennt von diesen die Einsatz- und Verwendungsmöglichkeiten und kann sie flexibel in Problemlöseprozessen einsetzen. Als Beispiele solcher Module kann man benennen: Diagonalverfahren, vollständige Induktion, Lemma von Zorn, Auswahlaxiom; aber auch: Ableitung, Riemann Integral, lineare Abbildung usf."

Es muß hier angefügt werden, daß nicht nur bei den erwachsenen Mathmatikern, sondern auch im Unterricht bei den »jungen« Mathematikern immer schon solche Module verwendet wurden und werden. In ganz selbstverständlicher - ja trivialer Weise - etwa, wenn die fortgesetzte Addition zum Modul der Multiplikation zusammengefaßt oder die fortgesetzte Multiplikation zum Modul der Potenz erhoben wird. Aber natürlich kann auch das Lösen einer Gleichung, der Ableitungskalkül oder die Bestimmung von Extremwerten als Modul angesehen werden.

Durch die "Kapselung" von Wissen kann dieses in Form von Elementen, Einheiten, Objekten (vgl. 2.5.2, Bereitstellung neuer Sprachelemente) oder "Bausteinen" in den Problemlöseprozeß eingebaut werden. Die besondere Bedeutung der "Einkapselung" von erworbenem Wissen für den Mathematikunterricht liegt in der Veränderung und Reorganisation mathematischer Tätigkeiten. Diese Reorganisation läßt sich in drei Schritten beschreiben.

Schritt 1: Verstehen, Strukturieren, Aufspüren von Wissenseinheiten

Am Beginn steht das Verstehen. Ein bearbeiteter Problembereich, ein Unterrichtsthema, ein Projekt, ein mathematisches Teilgebiet muß soweit durchdrungen und verstanden sein, daß Wissenseinheiten aufgespürt werden und diese Wissenseinheiten zueinander in eine (hierarchische) Ordnung gebracht werden können (Stichworte: "lokales Ordnen", strukturierte Darstellung). "Es sei allerdings darauf hingewiesen,daß ein inhaltliches Begriffsverständnis dieser Module unverzichtbar ist, da nur so ein "didaktisches Prinzip der Balance zwischen Bedeutungshaltigkeit und operativer Aktion" beim Arbeiten mit dem Computer im Unterricht entwickelt werden kann." [H.G.Weigand, a.a.O.,S.432]

Schritt 2: Implementieren, Testen und Dokumentieren

Durch den Einsatz von CAS (die selbst als umfangreiche Sammlung von Modulen betrachtet werden können), wird nun ein weiterer Schritt möglich. Wir können nun über eine statische Zusammenfassung des erworbenen Wissens hinausgehen, indem wir es durch eine Implementation im CAS (das dann als Speicher und Prozessor dient) interaktiv nutzbar machen. Durch die Implementation verbindet sich die Ökonomie der Mathematisierung ("die Kraft des Formalen", über das so viele Kalküle, Techniken und Verfahren zugänglich werden) mit der numerischen, symbolischen und grafischen Leistungsfähigkeit des CAS.

Im Prozeß der Implementierung berühren sich Mathematik und Informatik: Implementierung heißt *mathematisch*: Definieren, Exaktifizieren, konstruktives Beschreiben, Strukturieren, Algorithmisieren und heißt *informatisch*: Formulieren in der Sprache des CAS, Nutzen von vorangehenden Modulen, Nutzen der Möglichkeiten des CAS, Beachten der Grenzen des CAS, Nutzbarmachen der Systemresourcen, eventuell Programmieren.

Beim Prozeß der Implementation (bei dem die "Kapselung" des Wissens stattfindet), ist es natürlich notwendig, in erster Linie den inneren Aufbau von Modulen im Auge zu haben. Demgegenüber steht beim Prozeß des Testens und der Dokumentation bereits das Wissen um die Funktionalität, die Anwendbarkeit und die Bedingungen der Anwendung im Vordergrund.

Schritt 3: Problemlösen mit Hilfe von Modulen

Durch die Verwendung der Module erfolgt nun die eigentliche Reorganisation mathematischer Tätigkeit, von der die Rede ist: Terme, Gleichungen, Funktionen, Algorithmen, Kalküle stehen als Ganzes zur Verfügung und können als Ganzes in den Problemlöseprozeß eingebaut werden, der damit überschaubarer wird. Denken und Arbeiten in der Problemlöseebene werden dadurch leichter möglich. Damit verbunden ist natürlich eine Verschiebung des Schwerpunkts von der Tätigkeit des Ausführens hin zur Tätigkeiten wie Planen, Reflektieren und Anwenden von Strategien. Damit ermöglicht der Modulgedanke letztendlich eine neue Qualität des Problemlösens.

4.3.2. Zur Genese von Modulen

Wir wollen hier eine Forderung H.G.Weigands voranstellen und gleichzeitig mit diesem Abschnitt einen kleinen Beitrag zu ihrer Erfüllung leisten: "Bei der Verwendung moderner Softwaresysteme ist es eine wichtige Aufgabe der Mathematikdidaktik, den Aufbau, die Konstruktion und den Umgang mit Modulen unter mathematischen und didaktischen Gesichtspunkten zu analysieren sowie Vorschläge und Strategien für die Einordnung von Modulen in den Mathematiklehrgang zu entwickeln" [a.a.O, S.432].

Die Implementation von Modulen kann von verschieden Seiten erfolgen. Die drei in bezug auf den Mathematikunterricht wesentlichsten sollen hier betrachtet werden:

Von Schülern erstellte Module

Module können das Produkt der Einzelarbeit von Schülern sein oder von diesen gemeinsam in Gruppen entwickelt werden. Für eine Umsetzung im Unterricht ist es allerdings notwendig, daß den Schülern der Umgang mit DERIVE hinreichend vertraut ist, wenngleich sich auch der "Programmieraufwand" - soweit überhaupt von einem solchen die Rede sein kann - meist in Grenzen hält, so daß er auch Schülern, die nicht einschlägig informatisch vorbelastet sind, zuzumuten ist. Dieser notwendige Grad an Vertrautheit stellt sich unserer Erfahrung nach dann ein, wenn das CAS einige Zeit im Unterricht konsequent eingesetzt wurde. Dabei spielt weniger die Zeit eine Rolle als vielmehr die Vielfalt der Probleme, die mit dem CA-Werkzeug behandelt wurden. Günstig ist sicher Vertrautheit mit der Definition von CA-Funktionen, Bekanntheit mit den wesentlichen Steuerstrukturen und, wenn möglich, mit rekursiven und iterativen Ausdrüken.

Zu Schritt 1: "Aufspüren" von Modulen

Für Schüler ist es sicher nicht einfach, ein mathematisches Teilgebiet zu strukturieren, frühestens kann dies nach einer eingehenden Bearbeitung dieses Gebiets von ihnen verlangt werden. Eine solche Strukturierung erfolgte bisher z.B. in Form einer Kurzfassung, einer Zusammenfassung des Gebiets, einer schematischen Darstellung und dem Erstellen einer Formelsammlung. Es ist aber in der Praxis nicht notwendig, mit der Implementation von Modulen wirklich bis zum Abschluß einer Unterrichtseinheit zu warten. Unserer Beobachtung nach sind Schüler sehr kreativ im Aufspüren von Wisseneinheiten.

Beispiel 4.9: Häufig wiederkehrende Abläufe als Module - das Modul als Rechenhilfe

Ein häufig wiederkehrender Ablauf im Mathematikunterricht ist das Erstellen von Tabellen, z.B. das Erstellen einer Wertetabelle. Dazu kann ein Modul entwickelt werden, das den immer wiederkehrenden Vorgang des Eintippens und Berechnen eines Terms für eine bestimmte Belegung in einem Schritt ausführen läßt.

```
#1:  WERTETABELLE(f, ug, og, sw) :=
                          VECTOR([x, f], x, ug, og, sw)          User
```

Auftretende Parameter: *f* Funktionsterm, *ug, og* untere und obere Grenze der Wertetabelle, *sw* Schrittweite. Dieses einfache Modul kann dann mit folgendem Aufruf verwendet werden:

#2: $\text{WERTETABELLE}\left[\dfrac{1}{x}, -2, 2, 0.5\right]$ User

#3: $\begin{bmatrix} -2 & -0.5 \\ -1.5 & -0.666666 \\ -1 & -1 \\ -0.5 & -2 \\ 0 & \pm\infty \\ 0.5 & 2 \\ 1 & 1 \\ 1.5 & 0.666666 \\ 2 & 0.5 \end{bmatrix}$ Approx(#2)

Das immer Wiederkehrende ist damit zu einem einzigen Schritt, zur Anweisung "Generiere eine Tabelle mit den angegebenen Spezifikationen" geworden.

Beispiel 4.10: Die Formel als Modul

Auch jede Formel kann als ein Modul betrachtet werden. Bei der Behandlung von Aufgaben in der Analytischen Geometrie etwa tritt oft das Problem auf, den Winkel zwischen zwei Vektoren - sowohl in der Ebene als auch im Raum - ermitteln zu müssen.

Die Formel $\cos\varphi = \dfrac{\vec{a}\cdot\vec{b}}{|a|\cdot|b|}$ [22] stellt hier eine Lösung dieses Problems dar.

Das Modul WINKEL (#1) stellt dann die (sogar dimensionsunabhängige) Implementation (in DERIVE) dar:

```
#1:  [Angle := Degree ]                                      User
```

Die in #1 durchgeführte Programmvoreinstellung ermöglich des Rechnen im Gradmaß.

```
                             ┌     a · b     ┐
#2:  WINKEL(a, b) := ACOS│ ───────────── │                 User
                             └  |a| · |b|    ┘
```

Auftretende Parameter: *a, b* beliebig-dimensionale Vektoren

Ein Test dieser Implementation besteht im einfachsten Fall darin, mit dem implementierten Modul zuvor behandelte Aufgaben nochmals zu lösen:

Beispiele aus der Ebene:

```
#3:  WINKEL([1, 0], [0, 1])                                  User
#4:  90                                               Approx(#3)

#5:  WINKEL([1, 1], [0, 1])                                  User
#6:  45                                               Approx(#5)
```

Oder aus dem $\mathbb{R}^3$: Wie groß ist der Winkel zwischen Raumdiagonale und Grundfläche eines Würfels?

```
#7:  WINKEL([1, 1, 0], [1, 1, 1])                            User
#8:  35.2643                                          Approx(#7)
```

Beispiel 4.11: Anpassung des Systems an die Wünsche des Schülers - Erweiterung der Leistungsfähigkeit durch Module

Beim Rechnen mit komplexen Zahlen tritt auch das Problem auf, Gleichungen der Art

$$x^n - z = 0$$

lösen zu müssen, d.h., es stellt sich die Frage nach den Wurzeln aus komplexen Zahlen. Mit diesem Problem wird man auch bereits im Zusammenhang mit der Definition von Wurzeln aus reellen Zahlen konfrontiert. Läßt sich die Wurzel aus einer negativen Zahl angeben?

Ist nicht $\sqrt[3]{-8} = -2$ [23]? Was ist eigentlich nicht korrekt an der Umformung

$$-2 = \sqrt[3]{-8} = (-8)^{\frac{1}{3}} = (-8)^{\frac{2}{6}} = \sqrt[6]{(-8)^2} = \sqrt[6]{64} = \sqrt[6]{2^6} = 2 \ ?$$

An Ende der Klärung dieses versteckten Begriffsproblems steht die Definition der n-ten komplexen Wurzeln, die sich mittels Satz von Moivre berechnen lassen. Man erhält alle n-ten Wurzeln aus $z = r.$ (cos φ + i. sin φ) mit

$$\zeta_k = \sqrt[n]{r} \cdot (\cos\frac{\varphi + k \cdot 2\pi}{n} + i \cdot \sin\frac{\varphi + k \cdot 2\pi}{n})\ .$$

Das CAS liefert nun aber - je nach Voreinstellung - entweder die Hauptwurzel (k=0) oder eine reelle Wurzel, falls eine existiert, auch wenn sie nicht die Hauptwurzel ist, oder einfach nur irgendeine Wurzel. Wollen wir wirklich alle *n*-ten Wurzeln ermitteln, so läßt sich diese Systemanpassung durch ein Modul bewerkstelligen:

```
                                       1/n   ┌    ┌ PHASE(z) + k·2·π ┐
#1:  C_ROOTS(z,n):= VECTOR ┌ |z|     · │COS│ ──────────────────  │ +
                            └            └    └         n          ┘

                    ┌ PHASE(z) + k·2·π ┐┐
             î·SIN │ ──────────────────  ││, k, 0, n - 1┐  User
                    └         n          ┘┘              ┘
```

Auftretende Parameter: *z* komplexer Radikand, *n* Wurzelexponent.

Die drei dritten Wurzeln von -8 sind damit:

```
#2:  C_ROOTS(-8,3) = [1+√3·î, -2, 1-√3·î]      User=Simp(User)
```

Sie erfüllen alle die Gleichung $x^3 + 8 = 0$

```
User=Simp(User)
                 3
#3:  VECTOR(rz  , rz, [1 + √3·î, -2, 1 - √3·î]) = [-8, -8, -8]
```

Beispiel 4.12: Zusammenfassung eines Kapitels und interaktive Formelsammlung

Die Formelsammlung im CA-unterstützten Unterricht ist eine Sammlung von Modulen. Im folgenden wird ein Ausschnitt aus einer bei der schriftlichen Abschlußarbeit (Matura) tatsächlich verwendeten Formelsammlung (vgl. dazu 4.2.4) wiedergegeben. (Es wurde lediglich die Syntax an die Version 3.0 von DERIVE angepaßt). Dieser Ausschnitt besteht aus einer Reihe von Modulen, die im Laufe des Lehrplankapitels "Wachstumsprozesse und vernetzte Systeme" von Schüleren und Lehrer gemeinsam erarbeitet wurden. Ihr Sinn liegt hauptsächlich darin, Module zur Generierung von Tabellen und Module zur grafischen Darstellung bereitzustellen.

```
#1:  "Wachstumsprozesse und vernetzte Systemen"          User

#2:  "Wertetabelle für explizite Funktionen"             User

#3:  WERTETAB_E(f, ug, og, sw) :=
          VECTOR([x, f], x, ug, og, sw)                  User
```

Auftretende Parameter: f F*u*nktionsterm, *ug, og* untere und obere Grenze der Wertetabelle, sw Schrittweite.

```
#4:  "Wertetabelle für rekursive Funktionen"             User

#5:  WERTETAB_R(f, x0, y0, sz) :=
          ITERATES([n + 1, f], [n, x],[x0, y0], sz)      User
```

Auftretende Parameter: *f* rekursiver Funktionsterm, *x*0, *y*0 Startwerte, *sz* Schrittzahl.

```
#6:  "Einige nützliche Funktionen bei vernetzten Systemen"

#7   "Modul zur Simulation eines Systems mit einer
      Bestandsgröße"                                 User

#8:  SYSTEM1(r, y, y0, t0, δt, sz) :=
     ITERATES([t + δt, y + r·δt],[t, y], [t0, y0], sz)   User
```

Auftretende Parameter: *r* Änderungsrate, *y* Bestandsgröße, *y*0 Anfangsbestand, *t*0 Anfangszeit, *δt* Zeitintervall, *sz* Schrittzahl.

```
#9   "Modul zur Simulation eines Systems mit zwei
      Bestandsgrößen"                              User

#10: SYSTEM2(r1, r2, y1, y2, y10, y20, t0, δt, sz) :=
     ITERATES([t + δt, y1 + r1·δt, y2 + r2·δt], [t, y1, y2],
   [t0, y10, y20],sz)                       User
```

Auftretende Parameter: *r*1,*r*2 Änderungsraten, *y*1,*y*2 Bestandsgrößen, *y*10, y20 Anfangsbestände, *t*0 Anfangszeit, *δt* Zeitintervall, *sz* Schrittzahl.

```
#11: "Hilfsmodul zum Extrahieren einzelner Spalten aus
      einer Tabelle"                                   User

#12: ZWEISPALTEN(m, s1, s2) := [m`  , m`  ]`
                                 s1    s2              User
```

Auftretende Parameter: *m* Tabelle (Matrix), *s*1,*s*2 zu extrahierende Spalten

Zum Schritt 2: Umsetzung im CAS

Eine Implementation von Modulen durch die Schüler selbst ist nur dann möglich, wenn solche Einheiten von diesen überhaupt erkannt werden, z.B. in Form von wiederkehrenden -algorithmischen - Abläufen, in Formeln, als gewünschte Systemerweiterungen oder als Zusammenfassungen von Unterrichtseinheiten. Eine weitere Voraussetzung ist - wie bereits erwähnt - die nötige Vertrautheit im Umgang mit dem CAS.

Zur Dokumentation ist es erforderlich, einen Ausdruck des implementierten Moduls bereitzustellen, einige Testbeispiele und vor allem die Bedingungen, die für einen Aufruf notwendig sind. Gemeint sind spezielle Einstellungen im System (in Bsp.4.10 etwa erfolgt durch Umstellung auf das Gradmaß `Angle := Degree`), die Beantwortung möglicher Fragen (etwa: Wie müßte die Implementation abgeändert werden, damit sie auch im Bogenmaß funktioniert?) und sonstiger wichtiger Voraussetzungen für einen erfolgreichen Einsatz des Moduls.

Zum Schritt 3: Integration in den Problemlöseprozeß

Gerade bei selbsterstellten Modulen sollte deren Anwendung beim Lösen von Aufgaben und als Hilfe zur Problemlösung keine besondere Schwierigkeiten bereiten. Allerdings ist bei der Erstellung solcher Module in Partnerarbeit oder durch ein Team von Schülern eine gewisse Vorsicht sicher am Platz. Module können unverstanden übernommen werden, selbsterstellte Module nach einiger Zeit nicht mehr richtig eingesetzt werden, weil der Zusammenhang, in dem sie entstanden sind, nicht mehr erinnert wird. K.Aspetsberger [Müller, 1995, S.77] führt verschiedene Vor- und Nachteile solcher Schüler-generierter Module an:

Vorteile der Modulkonzepts:

"Gute Schüler sind motiviert, sich durch das Definieren von Funktionen ein eigenes System zu schaffen. Dies ist umso mehr der Fall, als auch der Nutzen von Funktionen für Schüler leicht einsehbar ist. Das Schreiben von Funktionen ist eine Art des konstruktiven Exaktifizierens und bildet den Abschluß einer intensiven Erarbeitungsphase"

Nachteile des Modulkonzepts:

"Leistungsschwache Schüler verwenden Module "blind". Sie verwenden Funktionen, ohne ihre Wirkungsweisen zu verstehen und zu hinterfragen. Unverstandenes kann durch die Verwendung von Funktionen noch mehr verschleiert werden.

Die Fehlersuche bei unerwünschten Ergebnissen wird für den Schüler noch schwerer, falls die Funktionsweise der verwendeten Module vom Schüler nicht wirklich nachvollzogen werden kann."

Vom Lehrer zur Verfügung gestellte Module

Vom Lehrer zur Verfügung gestellte Module haben in erster Linie dann eine Berechtigung, wenn sie als Hilfsmittel zur Visualisierung eingesetzt werden, Simulationen ermöglichen oder wenn es sich um speziellen Systemanpassungen handelt, um Module zur Steigerung der Leistungsfähigkeit des Systems in einem bestimmten Problembereich. Es können hier Module, deren innerer Aufbau dem Schüler zugänglich gemacht werden soll, wie Beisp.4.13 von Modulen, deren innerer Aufbau für den Schüler eine Black Box bleiben kann, unterschieden werden. Zugänglich gemacht werden sollen solche Module, deren innere Struktur für das Verständnis eines Begriffs einen Betrag leisten kann. Bei beiden dargestellten Beispielen geht es um Visualisierung, wobei im ersten die "Öffnung" des Moduls Sinn machen kann, im zweiten eher nicht.

Beispiel 4.13: Erzeugung eines Stabdiagramms - ein zugängliches Modul

Das hier vorgestellte Modul STAB nützt die Rechenleistung des Systems zur Erzeugung von Diagrammen zur Visualisierung von Aufgaben in der Wahrscheinlichkeitsrechnung.

Die Wahrscheinlichkeit, daß bei einem n-stufigen Bernoulli-Experiment (=aus einer Folge von *n* Versuchen bestehendes Experiment, bei dem jeder Versuch genau zwei mögliche Versuchsausgänge besitzt und jeder einzelne Versuch unter gleichen Bedingungen abläuft) mit der "Erfolgs"-Wahscheinlichkeit P die Anzahl *X* der "Erfolge" genau *k* ist, beträgt

$$P(X=k)=\binom{n}{k}\cdot p^k\cdot(1-p)^{n-k} \qquad (k=0,1,....,n)$$

Der DERIVE-Befehl COMB(n,k) berechnet den Binomialkoeffizienten (#2), STRECKE ist ein Hilfsmodul, das an der Stelle k eine Strecke mit der Höhe h zeichnet..

```
#1:  "Binomialverteilte Zufallsvariable mit den
      Parametern n und p"                                    User
                              k         n - k
#2:  B(n, p, k) := COMB(n, k)·p  ·(1 - p)                     User

                        ┌ k  0 ┐
#3:  STRECKE(k, h) :=   │      │                              User
                        └ k  h ┘

#4:  STAB(n,p) := VECTOR(STRECKE(k,B(n,p,k)),k,0,n)           User
```

Auftretende Parameter: *n* Umfang der Stichprobe, *p* Wahrscheinlichkeit für das Auftreten des Ereignisses *E*.

Mit diesem Modul können nun konkrete Verteilungen leicht visualisiert werden. Es kann dadurch zu einem besseren Verständnis des Begriffs "Wahrscheinlichkeitsverteilung" beitragen.

```
#5:  STAB(20, 0.5)                                            User

#6:  STAB(40, 0.5)                                            User

#7:  STAB(60, 0.5)                                            User

#8:  STAB(80, 0.5)                                            User
```

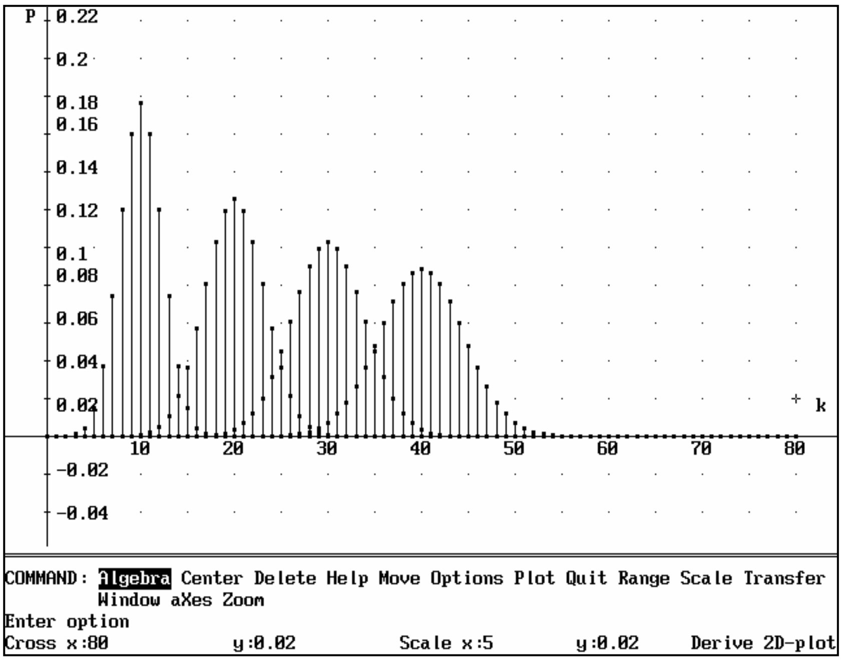

Abb. 4.9: Wahrscheinlichkeitsverteilungen binomialverteilter Zufallsvariablen

Beispiel 4.14: Modul zur Erzeugung von Webdiagrammen - ein geschlossenes Modul

Im Zusammenhang mit Simulationsaufgaben tritt häufig der Wunsch auf, zusätzlich zur Darstellung der Abhängigkeit von der Zeit (*n*-x_n-Diagramm) auch ein sogenanntes Web-Diagramm zu erstellen, um etwa Konvergenz oder chaotische Entwicklung besser verfolgen zu können.

Nach Angabe einer wichtigen Voreinstellung (in #1) und der Definition einer Leerfunktion (`NEXT(xn)` dient zur Aufnahme des iterativen Folgenterms) werden in #3 und #4 zwei Hilfsmodule definiert, die zum Zeichnen des ersten Striches bzw. der "Treppe" im gewünschten Diagramm dienen.

```
#1:  InputMode := Word                                          User

#2:  NEXT(xn) :=                                                User
                  ┌ x0       0        ┐
#3:  AUFTAKT(x0) := [                  ]                        User
                  └ x0   NEXT(x0)     ┘

                  ┌    xn          NEXT(xn)       ┐
#4:  NEXTP(xn) := [ NEXT(xn)       NEXT(xn)       ]             User
                  └ NEXT(xn)   NEXT(NEXT(xn))     ┘
```

Nun kann das gesuchte Modul zusammengebaut werden. Append(Liste1,Liste2) dient dabei zur Vereinigung zweier Listen zu einer:

```
#5:  WEB(x0, sz) :=

       APPEND([AUFTAKT(x0)], VECTOR(NEXTP(xn), xn,
               ITERATES(NEXT(xn), xn, x0, sz - 1)))             User
```

Auftretende Parameter: *x*0 Startwert, *sz* Schrittzahl.

Damit sind wir z.B. in der Lage, ein Webdiagramm für eine Simulation eines Erwärmungsprozesses [Szirucsek u.a., 1990, S.178] zu erstellen.

Ein Gegenstand hat im Kühlschrank eine Temperatur von 6 °C. Er wird in eine Umgebung mit 20 °C gebracht. Sein Temperaturzuwachs beträgt pro Minute 30% des Unterschiedes zwischen der Umgebungstemperatur von 20 °C und seiner Temperatur am Beginn dieser Minute.

Die Aufgabe läßt sich durch folgende Iterationsgleichung beschreiben:

$$x_{n+1} = x_n + 0.3\,(20 - x_n)$$

$$x_{n+1} = 0.7\,x_n + 6 \quad \text{mit } x_0 = 6\,°\text{C}$$

Der Einsatz des Moduls erfolgt durch #6 und #7.

```
#6:   NEXT(xn) := 0.7·xn + 6                              User

#7:   WEB(6, 10)                                          User
```

In #8 und #9 wird die Iterationsgleichung aus Beispiel 2.?? verwendet:

```
#8:   NEXT(xn) := xn + 0.00625·xn·(400 - xn)              User

#9:   WEB(20, 10)                                         User
```

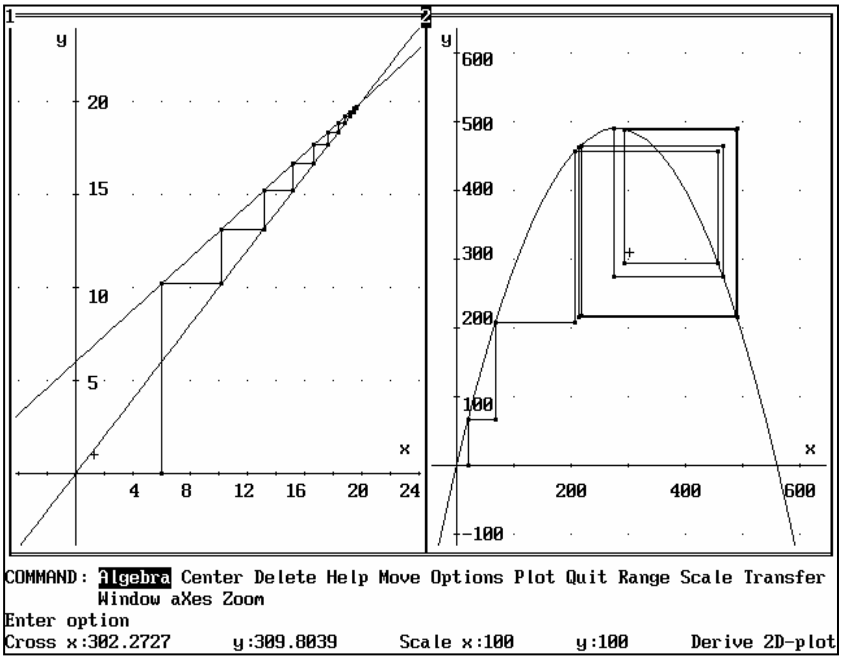

Abb. 4.10: Zwei Spinnwebdiagramme

Wird das Ergebnis von #7 und #8 (die entsprechenden Zahlenkolonnen sind aus Gründen der Übersichtlichkeit hier nicht wiedergegeben) geplottet, so erhält man die folgenden Diagramme (Abb. 4.11).

Anmerkung: Der erste CA-fähige Taschenrechner (TI-92) hat dieses Modul bereits fest im System integriert.

Module, die vom Systementwickler zur Verfügung gestellt werden

Module werden bei praktisch allen CAS (in Form von Utilityfiles, Hilfsdateien, Pakages etc.) von den Systementwicklern beigestellt, um bausteinartig entweder Systemerweiterungen zur Verfügung zu stellen, die nicht dauernd im Kern des Systems zur Verfügung stehen müssen, oder um für bestimmte mathematische Gebiete Problemlösungen zu liefern.

Beispiel 4.15: Runge-Kutta-Verfahren zur numerischen Integration von Differenzengleichungen

Wir wollen zuerst ein Beispiel aus der Biologie bzw.Ökologie betrachten, das häufig in der Systemdynamik behandelt wird. Sehr oft geht es bei systemdynamischen Fragestellungen darum, das Langzeitverhalten von voneinander abhängigen Populationen (Räuber-Beute-Beziehung, Konkurrenz, Symbiose etc.) vorherzusagen. Für diese Zwecke ist das Euler-CAuchy-Verfahren oder das verbesserte Euler-CAuchy-Verfahren (Halbschrittverfahren) als Methode der numerischen Integration nicht ausreichend [vgl. Ossimitz/Schlöglhofer 1990,S.253]. In diesem Fall ist es günstig, genauere Verfahren, die in Modulform vorliegen müssen, einsetzen zu können. Bei der numerischen Approximation von Differentialgleichungen ist dies das Runge-Kutta-Verfahren. Im folgenden wird ein Ausschnitt aus der Datei ODE_APPR.MTH wiedergegeben, in der eine Implementation des Runge-Kutta-Verfahrens erfolgt und die wohl ohne intimere Kenntnisse sowohl des besagten Verfahrens wie auch der DERIVE-Programmierweise nicht zu verstehen ist. Bei einem Einsatz - im Zusammenhang mit der Behandlung vernetzter Systeme etwa - ist es daher empfehlenswert, dieses Modul im Hintergrund als sogenanntes Utilitiy-File zu laden.

```
#1:  "File ODE_APPR.MTH, copyright (c) 1990-1994 by Soft
   Warehouse, Inc."                                     User

#2:  RK_AUX3(p, v, u_, c1, c2, c3) :=

    c1 + (   lim     p) + 2·(c2 + c3)
          v->u_ + c3
   ───────────────────────────────────                  User
                    6

#3:  RK_AUX2(p, v, u_, c1, c2) :=

          RK_AUX3(p, v, u_, c1, c2,    lim      p)      User
                                   v->u_ + c2/2

#4: RK_AUX1(p, v, u_, c1) :=

          RK_AUX2(p, v, u_,c1,    lim      p)           User
                          v->u_ + c1/2
#5: RK_AUX0(p, v, v0, n) :=

  ITERATES(u_ + RK_AUX1(p, v, u_, lim p), u_, v0, n)    User
                                  v->u_

#6: RK(r, v, v0, h, n) :=

          RK_AUX0(h·APPEND([1], r), v, v0, n)           User
```

Auftretende Parameter: *r* Liste der Änderungsraten, *v* Liste der Bestände, *v*0 Liste der Anfangsbestände, *h* Schrittweite, *n* Anzahl der Berechnungsschritte.

Bemerkung: Für eine genauere Behandlung der in der Systemdynamik verwendeten numerischen Verfahren mit CA-Unterstützung siehe CA-Report #5 [Lechner, Wien, 1996]

Unter Verwendung dieses Numerik-Moduls sind wir in der Lage, das Langzeitverhalten eínes Systems zu untersuchen, und können dabei Unsicherheiten - zumindest aufgrund eines inadäquaten numerischen Verfahrens - ausschließen. Als Beispiel wollen wir ein Räuber-Beute-Modell betrachten:

Ein Biotop sei von Räuber- und Beutetieren bevölkert. Die Zahl der Beutetiere ist bestimmt durch ihren natürlichen Zuwachs von jährlich 18% und ihre Abnahme durch ihre Feinde. Diese Abnahme ist proportional zur Anzahl der Räuber und zur Anzahl der Beutetiere selbst (Proportionalitätsfaktor b=0.005). Der Zuwachs an Räubern ist proportional zur Zahl der Beutetiere und zur Zahl der Räuber (Proportionalitätsfaktor d=0.0001). Die natürliche Abnahme betrage 15%. Anfangs gibt es 1500 Beutetiere und 25 Räuber im betrachteten Biotop. Beschreibe die Langzeitentwicklung der beiden Populationen!

Die Aufgabe läßt sich in zwei gekoppelte Iterationsgleichungen fassen:

$$\begin{aligned} x_{n+1} &= \quad 0.18 \cdot x_n + 0.005 \cdot x_n \cdot y_n \\ y_{n+1} &= -0.15 \cdot y_n + 0.0001 \cdot x_n \cdot y_n \end{aligned}$$

Unter Verwendung des Moduls RK (#8) und anschließender Einschränkung der entstehenden Tabelle mit der Hilfsfunktion #9 ergibt sich mit den Anfangsbedingungen $x_0 = 25$, $y_0 = 1500$ folgendes Bild der Langzeitentwicklung:

```
#8: RK([0.18·x-0.005·x·y, -0.15·y+0.0001·x·y], [t,x,y],
                  [0,1500,25], 1, 80)      User

       ZWEISPALTEN(m, s1, s2) := ┌m`  , m`  ┐`          User
#9:                               └  s1    s2┘
```

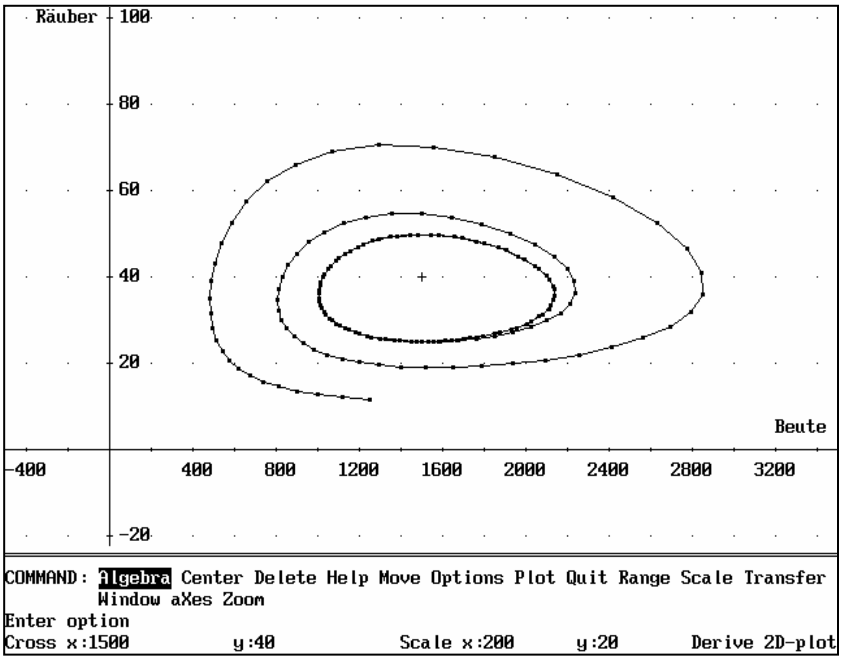

Abb. 4.11: Ein stabiles (?) Räuber-Beute-System (Phasendiagramm)

Vergleichsweise wurde die Aufgabe auch mit dem Euler-CAuchy-Verfahren gerechnet. Das Ergebnis zeigt, daß bei längerfristigen Prognosen das verwendete numerische Verfahren von großer Bedeutung ist (Abb.4.12).

Nach der Behandlung eines diskreten Prozesses wollen wir noch ein Beispiel eines kontinuierlichen Vorgangs aus der Physik betrachten (nach J.Zöchling [Reichel, 1995, S.278ff]), bei dem ein System-Modul zum Einsatz kommt.

Beispiel 4.16: Ein idealer und ein realer Vorgang:

a) Die ideale Bewegung eines Federpendels:

Wollen wir die Bewegung eines Federpendels unter idealen Verhältnissen beschreiben, so können auf Newtons dynamischem Grundgesetz (2.Newtonsches Axiom) aufbauend anschreiben

$$F = m.a = m.\frac{d^2 x}{d t^2}$$

wobei m die Masse des schwingenden Körpers und a die Beschleunigung des Massenstücks an der Feder beschreibt. Unter Verwendung des Hookschen Gesetzes $F = -k.x$ ergibt sich daraus die Bewegungsgleichung:

$$m.\frac{d^2 x}{dt^2} + k.x = 0$$

Durch Einführung der bekannten Beziehung $\omega^2 = \sqrt{\frac{k}{m}}$ 25 für die Kreisfrequenz ω können wir die allgemeinere Bewegungsgleichung anschreiben:

$$\frac{d^2 x}{dt^2} + \omega^2.x = 0$$

Wir wählen nun für die Eigenfrequenz f= 0.5 Hz (⇒ ω = 2 π f = π) und für die Anfangswerte $x(0)=0$ und $v(0) = \pi$ wählen. Unter Einsatz des Moduls DSOLVE2_IV aus dem Utilitiy-File ODE2.MTH erhalten wir eine Lösung, die auch zu erraten gewesen wäre:

```
                           2
#1:   DSOLVE2_IV(0, π , 0, t, 0, 0, π)                         User

#2:   SIN(π·t)                                              Simp(#1)
```

Plotten wir #2, so erhalten wir Abb.4.13 (obere Hälfte).

b) Die reale Bewegung eines Federpendels unter Reibungseinfluß

In der Realität wird aber keine Feder sich derart ideal bewegen, Reibungseinflüsse sind unvermeidlich. Um realtätsnäher zu arbeiten, setzen wir in unserem Modell die Reibung proportional zur Geschwindigkeit an.

$$\frac{d^2 x}{dt^2} + c.\frac{dx}{dt} + \omega^2.x = 0$$

Wenn wir (willkürlich) c = 0.5 setzen und im übrigen obige Bedingungen übernehmen, erhalten wir mit dem Differentialgleichungsmodul

```
                             2
#3:   DSOLVE2_IV(0.5, π , 0, t, 0, 0, π)                       User
```

die Lösung:

```
                                       2
                - t/4       ┌ t·√(16·π  - 1) ┐
       4·π·ê          ·SIN  │ ────────────── │
#4:                         └       4        ┘              Simp(#3)
      ───────────────────────────────────────
                            2
                    √(16·π  - 1)
```

Wenn wir #4 plotten, so erscheint die erwartete gedämpfte Schwingung. Ohne Sinusfunktion ergeben sich die Einhüllenden (Abb. 4.13, unterer Teil).

Schließlich können wir uns noch von der Richtigkeit unserer Lösung überzeugen, wenn wir diese (hier zuerst der Funktion $X(t)$ zugewiesen) in die Ausgangs-Differentialgleichung einsetzen.

```
       ┌d ┐2          1    ┌d ┐1          2
#7:    │──│ X(t)+  ───·  │──│ X(t)+  π ·X(t)  = 0     User=Simp(User)
       └dt┘          2    └dt┘
```

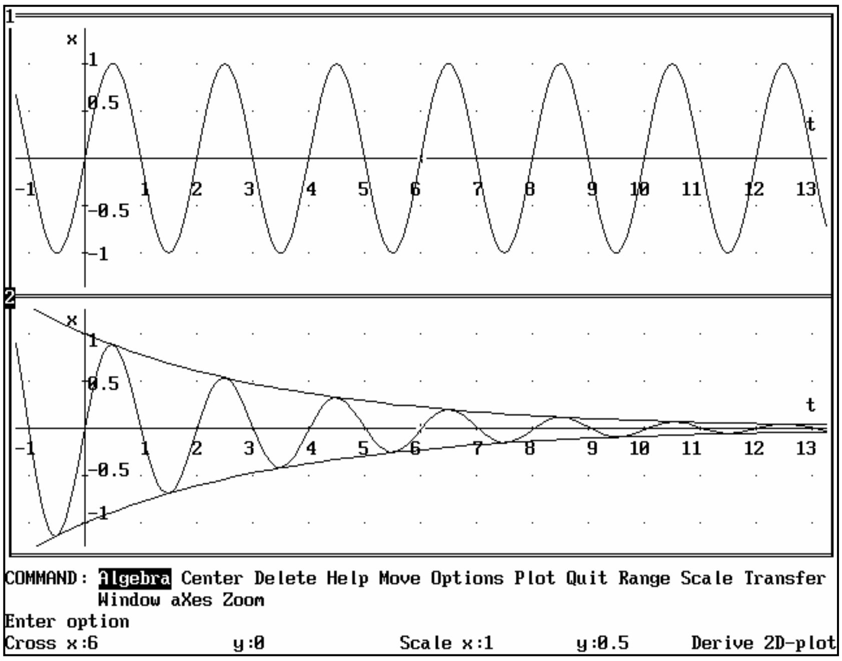

Abb. 4.12: Schwingungsgleichung ohne und mit Reibung

Zusammenfassung:

CAS fördern modulares Denken und Problemlösen: Anzustreben ist, daß die Lernenden die Module auch selber entwickeln und dabei zumindest grundsätzliche Kenntnisse des inneren Aufbaues erwerben. Bei der Nutzung dieser Module steht aber dann das Wissen um Funktionalität, Wirksamkeit, Anwendbarkeit usw. im Vordergrund.

Module stellen eine wesentliche Komponente eines CA-unterstützten, experimentellen Zuganges zur Mathematik dar, sie können helfen, Begriffe vorzubereiten, und sollen zu einer Exaktifizierung hinführen. "Man stellt Teile einer Begriffsbildung, eines Lösungsprozesses, etc. als mögliches Objekt im CAS (z.B. als Funktionsausdruck, Term o.ä.) dar und verwendet es in anderen Zusammenhängen, entwickelt das Objekt weiter, setzt verschiedene Objekte zusammen und gelangt damit zu einer höheren Stufe im Prozeß des mathematischen Arbeitens“ [Köhler, 1995, S.87]

Module können sie *eine wesentliche Rolle im Rahmen eines anwendungsorientierten* Mathematikunterrichts - spielen, bei dem der Schwerpunkt in der Bildung geeigneter Modelle und Mathematisierungen liegt. Werden solche Module geschickt im Unterricht eingesetzt, so können sie zu wachsen anfangen, *eine Eigendynamik entwickeln und eine fruchtbare Weiterentwicklung* erfahren.

Ihre **Sinnhaftigkeit** liegt einerseits

- im Erstellen, wenn damit schrittweises Definieren, Förderung des modularen Arbeitens und strukturierten Vorgehens (Fallunterscheidungen, Übersichtlichkeit, "lokales Ordnen") trainiert wird. Aber auch dann, wenn Problemlösungen und Unterrichtseinheiten zusammengefaßt werden sollen;
- im Endprodukt, wenn das Modul Rechenhilfe, eine Erweiterung der grafischen Möglichkeiten sein soll. Und auch dann, wenn die Sammlung der geschaffenen Module Basis für eine höhere Stufe des Problemlösens sein soll.

4.4. Die Window-Shuttle-Technik

Beispiel 4.17: "Linearisieren von Daten"

Eine Datenmenge ist durch eine lineare Funktion gegeben. Es werden Daten angegeben: f(101)=271.1; f(102.4)=275.6; f(104.6)=280.1; f(106)=283.6.

Durch einen Abschreibfehler ist ein Wert falsch angegeben worden. Stelle anhand einer geeigneten Zeichnung fest, welcher Datenwert falsch ist! Bestimme die Termdarstellung der linear organisierten Datenmenge und korrigiere den falschen Wert! [vgl.Bürger-Fischer-Malle Oberstufe 1]

Ziele: Erkennen von linearen Beziehungen. Lösen von Gleichungssystemen.

Rolle des CAS: Die Grafik dient als Entscheidungshilfe für das Aufsuchen von Zusammenhängen. Die Berechnungen erfolgen im Algebramodus. Ergebnisse werden im Grafikmodus getestet

Schritt 1:

Zuerst wird die Datenmatrix als Vektor eingegeben. Bei einer ersten groben Skalierung erscheint die Punktmenge undifferenziert auf dem Bildschirm (Abb 4.14).

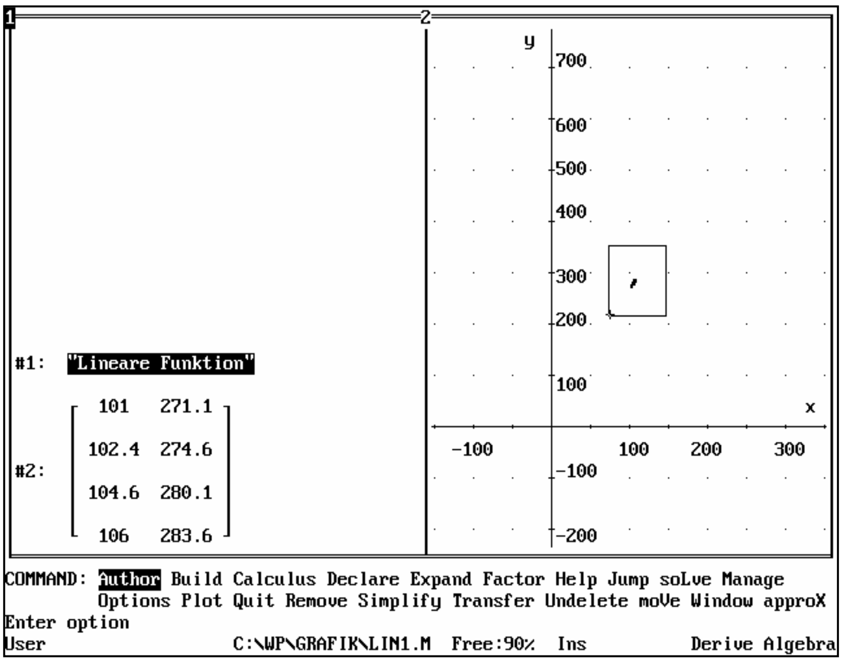

Abb. 4.13: Visualisieren von Daten

Mit dem Befehl **Zoom** oder mit dem **Range**-Befehl kann der Bildausschnitt vergrößert werden. *Connected* verbindet die gezeichneten Punkte. Man erkennt, daß der 2. Punkt der Datenmatrix nicht auf der Geraden liegt (Abb. 4.15).

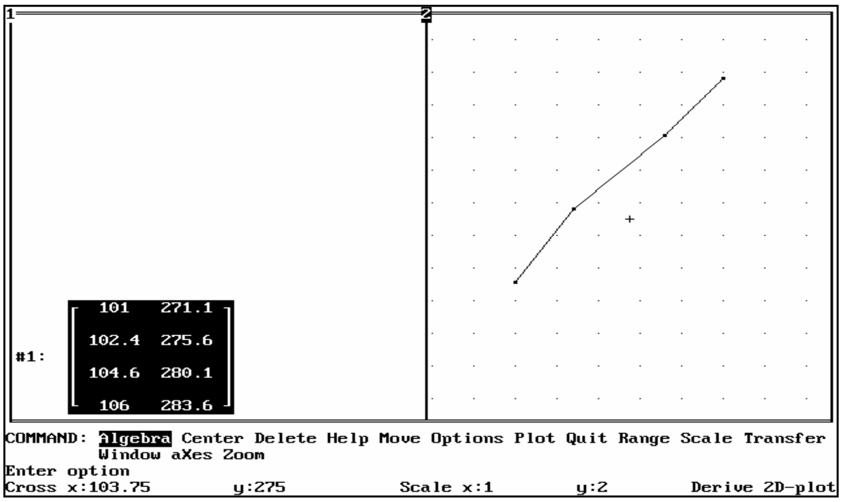

Abb. 4.14: Visualisieren von Daten

Schritt 2:

Nun kehrt man in das Algebrafenster zurück, um die Gleichung einer Geraden aufzustellen. Man wählt dafür etwa den ersten und den letzten Punkt. Das Lösen von Gleichungssystemen erfolgt durch Eingabe eines Vektors bestehend aus den zwei Gleichungen. Mit dem **soLve**-Befehl erhält man die Steigung *k* und den Abschnitt *d*.

Die so ermittelte lineare Funktion wird nun im Grafikfenster eingezeichnet (Abb.4.16).

Schritt 3:

Danach wird der Funktionswert des zweiten Punkts korrigiert und überprüft, ob auch der dritte Punkt auf der Geraden liegt. Jetzt wird die Datenmatrix richtiggestellt und zur Probe nochmals gezeichnet (Abb.4.17).

Dieses Beispiel soll aufzeigen, daß die Chance, in verschiedenen Fenstern mehrere Darstellungsarten gleichzeitig zu nutzen, neue didaktische Möglichkeiten bietet.

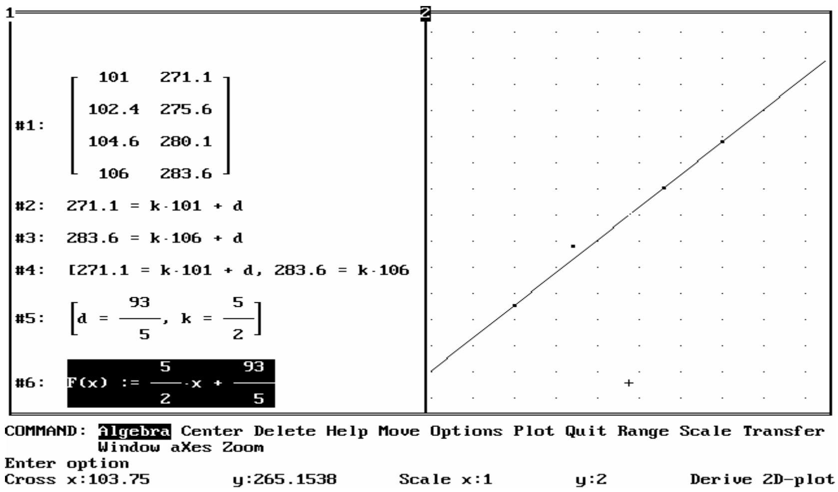

Abb. 4.15: Linearisieren

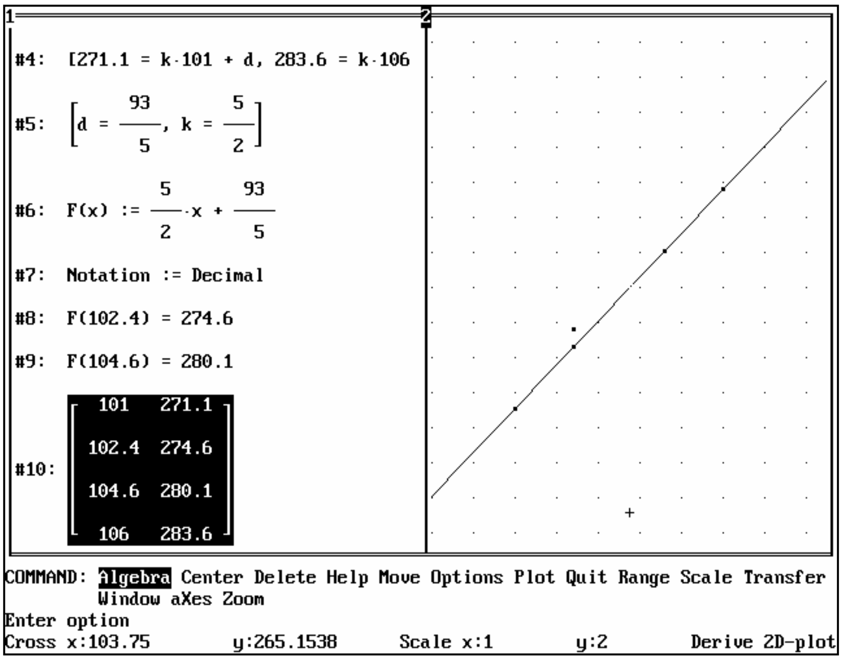

Abb. 4.16: Linearisieren

4.4.1. Die Idee der Window-Shuttle-Technik

Viele Anregungen zur Formulierung dieses Prinzips sowie zur Entwicklung von Unterrichtsmaterialien und zur Durchführung von Experimenten mit Schülern verdanken wir einer Arbeit von W. Dörfler mit dem Titel "Computer als kognitives Werkzeug und kognitives Medium" [Dörfler, 1991]. Besonders Experimente in Klassen, wo das CAS in jeder Arbeitssituation zur Verfügung steht - also nicht nur im Unterricht, sondern auch zu Hause und in der Prüfungssituation - haben die These bestätigt, daß Kognition als funktionales System gesehen werden muß, das Mensch und Werkzeug, aber auch den sonstigen materiellen und sozialen Kontext umfaßt. Das Werkzeug ist demnach nicht nur "Verstärker" menschlicher Leistungen, es kann Kognition qualitativ verändern und neue Fähigkeiten generieren.

Sowohl im Abschnitt "Kognitive Modelle und Gegenständlichkeit des Denkens" als auch im Kapitel "Computer als Medium für Prototypen" finden sich Thesen, die die Window-Shuttle-Technik begründen:

"Denkprozesse erfolgen oft vorteilhaft anhand gegenständlicher Vorstellungen, Repräsentationen, Modellierungen der jeweiligen Problemsituation"

"Allgemeinbegriffe bzw. Kategorien werden mittels prototypischer Repräsentanten oder Vertreter kognitiv verfügbar gemacht für kognitive Funktionen wie Gedächtnis, Problemlösen, Herstellen von Beziehungen zwischen Klassen. Es wird also i.a. mental nicht abstrakt mit definierten Eigenschaften oder Bündeln von Eigenschaften operiert, sondern gegenständlich mittels der mentalen Vorstellung charakteristischer Exemplare oder sogar direkt an materiellen prototypischen Repräsentanten."

"Erst die Bereitstellung verschiedener Prototypen eines bestimmten Allgemeinbegriffs, das Herstellen von Beziehungen zwischen den einzelnen Prototypen lenkt die Aufmerksamkeit des Lernenden über die Spezifität des einzelnen Prototyps hinaus auf den übergeordneten Begriff."

"Um das "Prototypische" als invariante Eigenschaft erkennen zu können, muß der Prototyp öfter Veränderungen unterworfen werden."

Man denke zum Beispiel nur an den Funktionsbegriff. Nicht durch eine, noch so "saubere", abstrakte Definition wird der Schüler einen Zugang zu diesem Begriff finden, sondern durch das Angebot an geeigneten Prototypen

- enaktive (d.h.:handlungsorientierte), ikonische oder symbolische - , die die Aufmerksamkeit des Lernenden auf die wesentlichen Eigenschaften des Begriffs lenken. Eine besonders wichtige Tätigkeit in diesem Prozeß ist das Herstellen von Beziehungen zwischen den einzelnen Prototypen. Dadurch kann der Lernende zur Einsicht gelangen, daß der einzelne Prototyp nur eine von mehreren Erscheinungsformen des Begriffs Funktion ist, und erst danach ist eine verbale oder formal-allgemeine Definition des Begriffs "Funktion" sinnvoll.

Der Ablauf des Lernprozesses nach der Window-Shuttle-Technik und die Funktion des CAS bei diesem Prozeß:

- Der Lernende aktiviert in verschiedenen Fenstern des CAS verschiedene für das jeweilige Problem oder den jeweiligen Begriff adäquate Prototypen. Zum Beispiel einen symbolischen im Algebrafenster und einen grafischen im Grafikfenster. Durch den Einsatz des Computers werden noch dazu Darstellungsmöglichkeiten erschlossen, die vorher nicht zugänglich waren, wie etwa rekursive Modelle.
- Nun erfolgt die Bearbeitung der einzelnen Prototypen, wobei die Vorzüge des CAS, wie Interaktivität, leichte Manipulierbarkeit, Wiederholbarkeit, genutzt werden können.
- Die Mehrfenstertechnik erlaubt das simultane Arbeiten mit den verschiedenen Prototypen. So kann etwa in ständiger Wechselwirkung zwischen Algebra- und Grafikfenster die Auswirkung algebraischer Operationen auf den Graphen untersucht werden, oder es können sich aus der Untersuchung des Graphen Ideen für Aktivitäten im Algebrafenster ergeben. Die Konsequenzen der Veränderung einzelner Parameter im Algebrafenster (etwa bei rekursiven Darstellungen) können direkt im anderen Fenster (etwa an Tabellen oder am Graphen) untersucht werden.
- Arbeiten mit der Window-Shuttle-Technik bedeutet also, daß sich ein Begriff, eine Problemlösung durch mehrmaliges Hin- und Herpendeln ("Shutteln") zwischen verschiedenen Darstellungsformen, das heißt zwischen verschiedenen Fenstern des CAS, entwickelt.

Die Beobachtung der Schüler im Rahmen unseres CAS-Projekts lassen gestützt auf die in der Arbeit von W. Dörfler formulierten Thesen über den Computer als kognitives Medium den Schluß zu, daß die Arbeitsweise mit der Window-Shuttle-Technik, die erst durch CAS möglich gemacht wurde, wesentlich zur qualitativen Veränderung von Kognition beitragen kann.

Beispiel 4.18: Asymptotische Polynomfunktionen an rationale Funktionen

Ziele:Ausgehend von einem naiven Asymptotenbegriff ("nähert sich beliebig") soll diese Aufgabensequenz zu einem exakteren Begriff führen. Der Schüler soll erkennen, daß asymptotische Funktionen von rationalen Funktionen nicht nur lineare Funktionen, sondern auch Polynomfunktionen höheren Grades sein können.

Voraussetzungen:Ein naiver Asymptotenbegriff; Grenzwert von Zahlenfolgen; Polynomdivision.

1. Schritt:

Die Funktion $f(x)=\frac{x^2}{x-1}$ 26 wird gezeichnet und mit dem **ZOOM**-Befehl verkleinert, um das Verhalten im Unendlichen untersuchen zu können.

1. Vermutung: Die Asymptote ist eine Gerade. Nun soll der Schüler experimentell einige Geradengleichungen aufstellen und im Grafikfenster testen (siehe Abb.4.18).

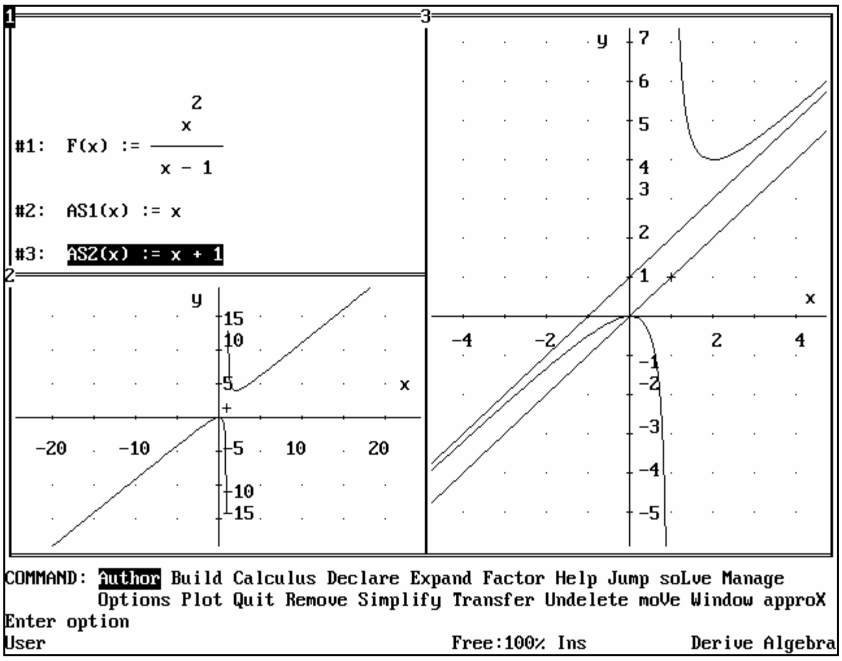

Abb. 4.17: Asymptotische Funktionen

2. Schritt:

Die Idee "nähert sich beliebig" oder "der Abstand zwischen der Funktion und der asymptotischen Funktion strebt gegen Null" wird nun im Algebrafenster durch die Definition der Distanzfunktion DIST(x) "mathematisiert".

$$\#1:\quad F(x) := \frac{x^2}{x-1}$$

```
#2:  AS1(x) := x

#3:  AS2(x) := x + 1

#4:  DIST(x) := F(x) - AS2(x)
```

Zuerst wird durch Einsetzen mit Zahlen experimentiert.

```
#5:  Precision := Approximate

#6:  DIST(5) = 0.25

#7:  DIST(-5) = -0.1666666666666666666

#8:  DIST(10000) = 0
```

Dabei kann es notwendig sein, die Rechengenauigkeit zu erhöhen, um falsche Vermutungen auszuschließen.

```
#9:  PrecisionDigits := 20
```

$$\#10:\quad DIST(10000) = 1.0001000100010001000 \cdot 10^{-4}$$

Danach ermittelt man den Limes von DIST(x) für $x \to \pm\infty$. Dabei wird das CAS als Black Box eingesetzt (siehe White Box/Black Box-Prinzip).

```
      lim  DIST(x) = 0
#11:  x->∞

       lim  DIST(x) = 0
#12:  x->-∞
```

Vereinfacht man mit **Simplify** die Distanzfunktion, so erhält man:

```
                     1
#13:  DIST(x) =  ─────────
                   x - 1
```

3.Schritt:

Nach dieser experimentellen Phase soll nun nach einer Methode gesucht werden, die asymptotische Funktion rechnerisch zu ermitteln.

Aus Zeile #3 ergibt sich: F(x) = AS(x) + DIST(x), wobei AS(x) die asymptotische Polynomfunktion ist und DIST(x) die Distanzfunktion, die für $x \to \pm\infty$ gegen 0 strebt.

Andererseits ist von der Polynomdivision bekannt:

Seien p und q Polynome m-ten und n-ten Grades, so erhält man bei der Division $p(x) \div q(x)$ ein Quotientenpolynom $a(x)$ und ein Restpolynom $r(x)$, wobei der Grad von r kleiner n ist.

Somit ist eine rationale Funktion $F(x)$ folgendermaßen darstellbar:

$f(x) = \frac{p(x)}{q(x)} = a(x) + \frac{r(x)}{q(x)}$. Von rationalen Funktionen $\frac{r(x)}{q(x)}$ mit Grad(r) < Grad(q) wissen wir aber, daß sie für $x \to \pm\infty$ gegen 0 streben.

Diese Überlegungen führen also zum Satz von der asymptotischen Annäherung einer rationalen Funktion durch eine Polynomfunktion:

Es sei f: y = p(x)/q(x) eine rationale Funktion und a(x) das Quotientenpolynom der Polynomdivision p(x):q(x). Dann ist a eine asymptotische Funktion an f. [Reichel, 1991]

Damit haben wir aber auch eine Methode gefunden, die gerade mit CAS sehr einfach ermöglicht, die asymptotische Polynomfunktion zu finden. Bei Verwendung von DERIVE erhält man mit dem Befehl **Expand** sofort das Ergebnis der Polynomdivision.

```
          1
#14:  ───────── + x + 1                               Expd(#1)
        x - 1

#15:  AS(x) := x + 1                                      User

                     1
#16:  REST(x) := ─────────                                User
                   x - 1
```

4. Schritt:

Nachdem man mit **F3** die asymptotische Funktion (#15) und das Restglied (#16) isoliert hat, wechselt man gemäß dem Shuttle Prinzip wieder in das Grafikfenster und untersucht die Graphen der Asymptote und des Restgliedes in Relation zur rationalen Funktion.

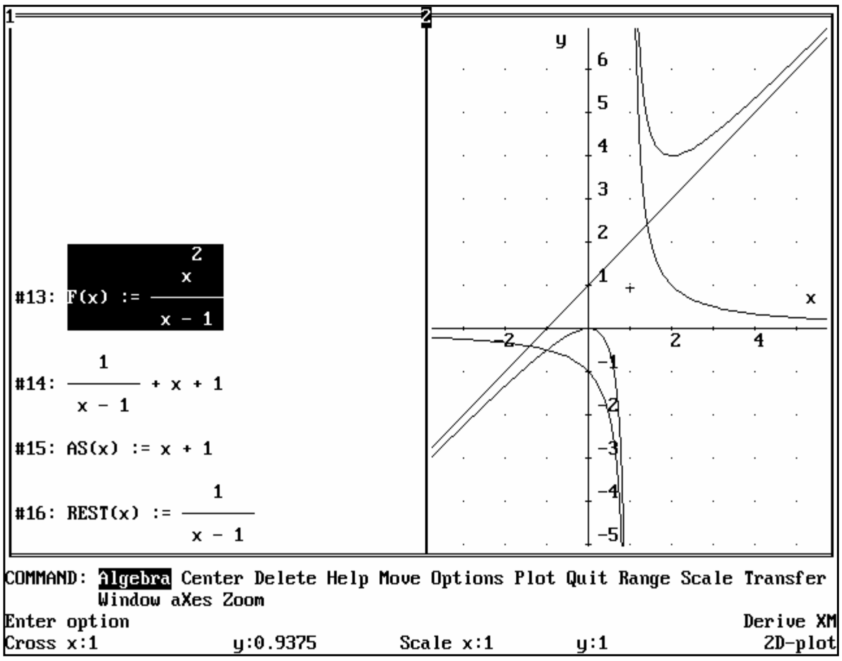

Abb. 4.18: Asymptotische Funktionen

Es sollten auch einige rationale Funktionen mit asymptotischen Funktionen höheren Grades untersucht werden, wie zum Beispiel in Abb.4.20:

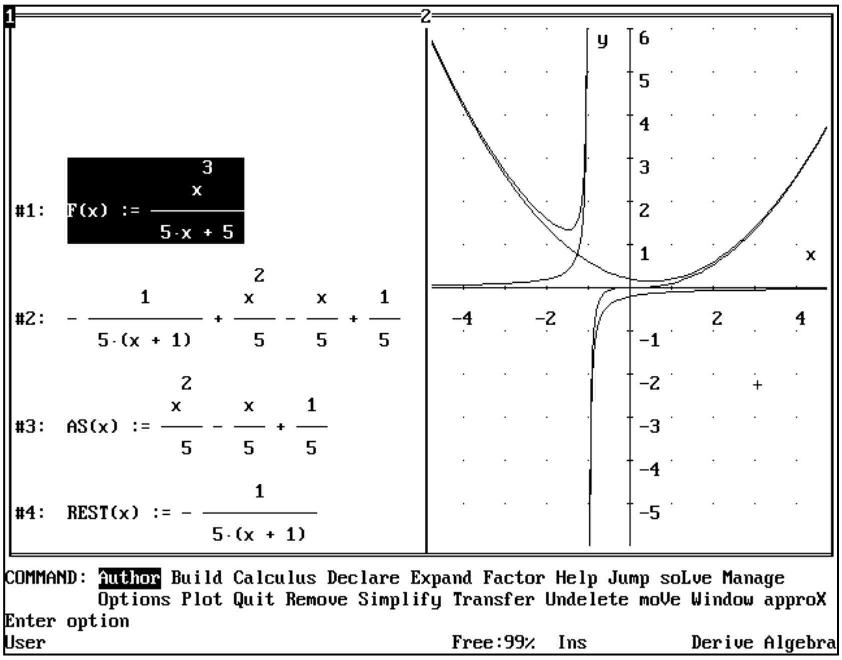

Abb. 4.19: Asymptotische Funktionen

Ein Beispiel, bei dem mehrere Algebrafenster verwendet werden, um Strategien für das Lösen von Gleichungen zu vergleichen, findet man in der 'Gleichungsbox' in Kap.4.1 (White Box/ Black Box-Prinzip). Der Schüer vergleicht verschiedene Prototypen von Äquivalenzumformungen mit dem Ziel, durch Abwägen der Vor- und Nachteile eine ihm passende Strategie zu finden.