2. Was kann ein Computeralgebra-System?

Es ist erstaunlich, mit welchen Problemen ein CAS zurechtkommen und in welch kurzer Zeit es korrekte Ergebnisse liefern. Wenn auch die einzelnen Systeme in ihrer Funktionalität und die der Art ihrer Benutzung oft beträchtliche Unterschiede aufweisen, so sind sie doch von ihrem Leistungsumfang, wenn man nur die schulrelevanten Komponenten betrachtet, durchaus vergleichbar. Hier entscheidet oft die leichtere Verfügbarkeit und Berücksichtigung didaktischer Erfordernisse. DERIVE zählt zu den didaktisch ausgereiftesten Systemen, die derzeit erhältlich sind. Es hat gegenüber anderen Mathematikprogrammen zudem den Vorteil, daß es verhältnismäßig leicht zu benutzen ist und nur geringste Hardwareanforderungen stellt. Durch die Implementierung solcher Systeme auf Taschenrechnerbasis - die mit dem Erscheinen eines CA-Taschenrechners bereits konkrete Gestalt angenommen hat - ist auch eine Zukunftsvision des letzten Jahrzehnts dabei, Realität zu werden: Der Taschenrechner hat das traditionelle Rechnen vollkommen verändert, Rechenschieber sind in der Folge genauso aus unserem Blickfeld verschwunden wie endlose Tabellen für elementare mathematische Funktionen. Mit der Verfügbarkeit von CAS ist es durchaus möglich, daß dessen Verwendung im Mathematikunterricht, an Hochschulen und in der Anwendung der Mathematik und der Naturwissenschaften bald ebenso selbstverständlich wird, wie dies heute der Taschenrechner ist.

Es ist schwer, die ungeheure Palette möglicher Einsatzformen eines CAS auf einigen Seiten darstellen zu wollen. Je mehr man sich damit beschäftigt, desto vermessener erscheint ein solches Vorhaben. Es sollen deshalb hier nur einige wenige - insbesondere aus didaktischer Sichtweise - ausgewählte Bereiche betrachtet werden, für die der Einsatz von CAS im Rahmen des Mathematikunterrichts von Relevanz ist. Es mag erstaunen, daß hier CAS als grafisches Hilfsmittel nicht eigens angeführt wird. Im CAS steht Grafik und damit die Möglichkeit der Visualisierung permanent und unmittelbar zur Verfügung, so daß sie in gleicher Weise zur Darstellung der Lösungen einer Differentialgleichung wie als Voraussetzung von CA als methodisches Hilfsmittel dient. Die besondere Rolle dieser durchgehenden grafischen Unterstützung soll im Rahmen der Window-Shuttle-Technik (Kap.4.4) eingegangen werden.

2.1. Numerisches Hilfsmittel

Stellt man sich die Frage, inwiefern CAS in numerischer Hinsicht über die Möglichkeiten eines Taschenrechners hinausgehen, so treten hier - was den Mathematikunterricht betrifft -vor allem drei Aspekte ins Blickfeld: CAS erlauben es erstens, *Rechnungen* mit Symbolen auszuführen. Sind wir an numerischen Approximationen interessiert, so können wir zweitens im Unterschied zum Taschenrechner selbst die *Genauigkeit* steuern. Drittens lassen sich beinahe alle *Umformungen*, die wir mit Zahlen auszuführen gewohnt sind, auch mühelos in der Umgebung des CAS durchführen. Was ist der Preis, um den wir diese 'erweiterten Kalkülfertigkeiten' erkaufen? Wir müssen darauf achten, daß auf der einen Seite notwendige und wesentliche 'Handkalkülfertigkeiten' nicht verlorengehen - wie vielfach befürchtet wird -, daß auf der anderen Seite aber entsprechende 'Strategiekalkülfertigkeiten', die für einen sinnvollen Einsatz von CAS erforderlich sind, aufgebaut werden.

2.1.1. Exaktes Rechnen

CAS sind imstande, Brüche und Wurzeln als Brüche und Wurzeln zu behandeln und diese so zu verknüpfen, wie wir dies in der Mathematik zu tun gewohnt sind .

```
#1:  Precision := Exact                                User

#2:  Notation := Rational                              User
```

Zu #1 und #2 findet sich am Ende dieses Kapitels die programmtechnische Anmerkung 1.

#3: $$\frac{\sqrt{2} + \sqrt{(2 + \sqrt{3})}}{2 + \sqrt{3}} \cdot \frac{\sqrt{2} - \sqrt{(2 - \sqrt{3})}}{2 - \sqrt{3}} = 3 \qquad \text{User=Simp(User)}$$

```
#4:  (2 + √3)/(√2 + √(2 + √3)) + (2 - √3)/(√2 - √(2 - √3)) = √2    User=Simp(User)
```

$$\frac{2+\sqrt{3}}{\sqrt{2}+\sqrt{2+\sqrt{3}}}+\frac{2-\sqrt{3}}{\sqrt{2}-\sqrt{2-\sqrt{3}}}=\sqrt{2}$$

```
#5:  (√5 + 2)^(1/3) - (√5 - 2)^(1/3) = 1                         User=Simp(User)
```

$$(\sqrt{5}+2)^{1/3}-(\sqrt{5}-2)^{1/3}=1$$

Erst wenn wir eine Näherung haben wollen, liefert das System eine solche entweder mit der Menü-Option **approX** oder mit der Einstellung **Option Precision** *Approximate*:

```
#6:  Precision := Approximate                                    User

User=Simp(User)
```

$$\#7:\quad \frac{(2+\sqrt{3})}{\sqrt{2}+\sqrt{(2+\sqrt{3})}}\cdot\frac{(2-\sqrt{3})}{\sqrt{2}-\sqrt{(2-\sqrt{3})}}=0.333333$$

2.1.2. Rechnen mit großer Genauigkeit

Da wir die Möglichkeit haben, zwischen exaktem und näherungsweisem Rechnen hin und her zu pendeln, sind wir nun auch in der Lage, Grenzen des Taschenrechners zu simulieren und verstehend zu überschreiten.

Im Wesen irrationaler Zahlen liegt es, daß wir sie nur mit Symbolen darstellen können, die numerische Annäherung ist zwar beliebig genau möglich, bleibt aber stets eine vorläufige. Mit einem Taschenrechner ist etwa $\sqrt{2}$ nur als Näherung darstellbar, durch Quadrieren können wir damit niemals wieder 2 erhalten. Viele - neuere - Taschenrechner liefern nach Betätigung der Wurzeltaste und Quadrattaste allerdings wieder den exakten Wert. Dies liegt an den "Schutzstellen", d.h. an zusätzlichen Stellen, die bei der Berechnung mitgeführt werden. Beim Quadrieren kommt dann durch entsprechendes Runden der gewünschte ursprüngliche Wert zustande.

Im Folgenden betrachten wir ein Näherungsverfahren zur Berechnung von $\sqrt{2}$.

Beispiel 2.1: Heron-Verfahren

Berechne die Seitenlänge eines Quadrats, dessen Flächeninhalt 2 cm^2 beträgt!

Ein Zugang zur Lösung dieses Problems stammt von Heron (um 100 n. Chr.): Obwohl die Seitenlänge eines Quadrats gesucht wird, betrachten wir eine Folge von Rechtecken, die alle denselben Flächeninhalt (A = 2 cm^2) haben und sich dem gesuchten Quadrat annähern. Es wird eine beliebige Länge des Rechtecks angenommen (z.B.: l = 2 cm). Nun läßt sich die Breite aus der Flächeninhaltsformel berechnen (b = 1 cm). Um aus diesen Angaben ein Rechteck zu erhalten, welches dem Quadrat angenähert ist, bildet man aus den beiden vorhergehenden Seitenlängen das arithmetische Mittel und erhält damit eine neue Seitenlänge. Man berechnet wieder die dazugehörige andere Seite unter der Bedingung, daß der Flächeninhalt gleich bleiben soll. Das Quadrieren dieser Mittelwerte ergibt jeweils einen zu großen Flächeninhalt, die Folge dieser Mittelwerte nähert sich der gesuchten Seitenlänge des Quadrats mit gesuchtem Flächeninhalt!

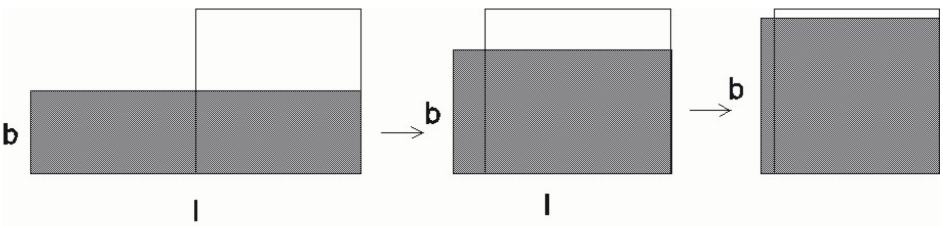

Abb. 2.1: Aus Rechtecken wird ein Quadrat

Man benötigt folgende Beziehungen, wobei *l* und *b* die Seitenlängen des Rechtecks sind. Die Variable *s* bezeichnet die Länge des neuen Rechtecks oder die Näherung zur gesuchten Seitenlänge des Quadrats.

$$l \cdot b = 2 \;=>\; b = \frac{2}{l}$$

$$l + b = 2 \cdot s \;=>\; s = \frac{1}{2} \cdot (l + b) \;=>\; s = \frac{1}{2} \cdot (l + \frac{2}{l})$$

Aus der zuerst intuitiv gewählten Länge *l* (SEITEALT) entsteht ein neues s (SEITENEU), und dieses wird wieder zur Länge eines neuen Rechtecks (SEITEALT) wodurch eine geeignetere Rechtecksseitenlänge *s* (SEITENEU) entsteht. Also allgemein mit Wortvariablen:

$$\text{SEITENEU} = \frac{1}{2} \cdot (\text{SEITEALT} + \frac{\text{FLAECHE}}{\text{SEITEALT}})$$

Diese Vorgehensweise zur näherungsweisen Berechnung von $\sqrt{2}$ läßt sich mit einem numerischen Taschenrechner durch die begrenzte Anzahl der sichtbaren Stellen nicht geeignet generieren.

Wir verwenden die Einstellungen **Option Input** *Word, Sensitive* und rechnen im *Exact*-Modus mit 20 Stellen (*Digits*). Dieser Modus ermöglicht die Darstellung der Näherungswerte als Brüche.

```
#1:  InputMode := Word                                     User
#2:  CASeMode := Sensitive                                 User
#3:  Precision := Exact                                    User
#4:  PrecisionDigits := 20                                 User
```

Die allgemeine Beziehung wird im **Author** eingegeben:

$$\#5:\quad \text{SEITENEU} = \frac{1}{2} \cdot \left[\text{SEITEALT} + \frac{\text{FLÄCHE}}{\text{SEITEALT}}\right] \qquad \text{User}$$

Mit **Manage Substitute** wird die Fläche mit 2 festgelegt:

$$\#6:\quad \text{SEITENEU} = \frac{1}{2} \cdot \left[\text{SEITEALT} + \frac{2}{\text{SEITEALT}}\right] \qquad \text{Sub(\#5)}$$

Wir substitueren für SEITEALT den frei gewählten Anfangswert 2

$$\#7:\quad \text{SEITENEU} = \frac{1}{2} \cdot \left[2 + \frac{2}{2}\right] \qquad \text{Sub(\#6)}$$

und berechnen mit **Simplify** und **approX** die neue Seitenlänge. Dadurch erhalten wir die erste Näherung als Bruch und als Dezimalzahl.

$$\#8:\quad \text{SEITENEU} = \frac{3}{2} \qquad \text{Simp(\#7)}$$

$$\#9:\quad \text{SEITENEU} = 1.5 \qquad \text{Approx(\#8)}$$

Es wird nun so getan, als wäre dies die Seite eines Quadrats, und wir testen den Flächeninhalt.

$$\#10: \quad \left[SEITENEU = \frac{3}{2}\right]^2 \qquad \text{User}$$

$$\#11: \quad SEITENEU^2 = 2.25 \qquad \text{Approx(\#10)}$$

Es zeigt sich, daß diese neue Seite, interpretiert als Quadratseite, einen zu großen Flächeninhalt liefert.

Die neue Seite wird wieder in die allgemeine Formel #6 eingesetzt, und durch Mittelwertberechnung wird die nächste Näherung generiert.

$$\#12: \quad SEITENEU = \frac{1}{2}\cdot\left[\frac{3}{2} + \frac{2}{\frac{3}{2}}\right] \qquad \text{Sub(\#6)}$$

$$\#13: \quad SEITENEU = \frac{17}{12} \qquad \text{Simp(\#12)}$$

$$\#14: \quad SEITENEU = 1.4166666666666666666 \qquad \text{Approx(\#13)}$$

$$\#15: \quad \left[SEITENEU = \frac{17}{12}\right]^2 \qquad \text{User}$$

Der Test des neuen Flächeninhalts zeigt bereits eine akzeptablere Näherung.

$$\#16: \quad SEITENEU^2 = 2.0069444444444444444 \qquad \text{Approx(\#15)}$$

Wird diese Vorgehensweise mehrmals angewendet, ergibt sich, daß die neue Seite sich immer besser eignet. Der Flächeninhalt des neuen Quadrats liegt immer näher bei 2 cm^2.

$$\#17: \quad SEITENEU = \frac{1}{2}\cdot\left[\frac{17}{12} + \frac{2}{\frac{17}{12}}\right] \qquad \text{Sub(\#6)}$$

$$\#18: \quad SEITENEU = \frac{577}{408} \qquad \text{Simp(\#17)}$$

$$\#19: \quad SEITENEU = 1.4142156862745098039 \qquad \text{Approx(\#18)}$$

$$\#20: \quad \left[SEITENEU = \frac{577}{408}\right]^2 \qquad \text{User}$$

$$\#21: \quad SEITENEU^2 = 2.0000060073048827374 \qquad \text{Approx(\#20)}$$

$$\#22: \quad SEITENEU = \frac{1}{2}\cdot\left[\frac{577}{408} + \frac{2}{\frac{577}{408}}\right] \qquad \text{Sub(\#6)}$$

$$\#23: \quad SEITENEU = \frac{665857}{470832} \qquad \text{Simp(\#22)}$$

$$\#24: \quad SEITENEU = 1.4142135623746899106 \qquad \text{Approx(\#23)}$$

$$\#25:\ \left[\mathrm{SEITENEU} = \frac{665857}{470832}\right]^2 \qquad \text{User}$$

$$\#26:\ \mathrm{SEITENEU}^2 = 2.0000000000045109504 \qquad \text{Approx(\#25)}$$

$$\#27:\ \mathrm{SEITENEU} = \frac{1}{2}\cdot\left[\frac{665857}{470832} + \frac{2}{\frac{665857}{470832}}\right] \qquad \text{Sub(\#6)}$$

$$\#28:\ \mathrm{SEITENEU} = \frac{886731088897}{627013566048} \qquad \text{Simp(\#27)}$$

$$\#29:\ \mathrm{SEITENEU} = 1.4142135623730950488 \qquad \text{Approx(\#28)}$$

$$\#30:\ \left[\mathrm{SEITENEU} = \frac{886731088897}{627013566048}\right]^2 \qquad \text{User}$$

$$\#31:\ \mathrm{SEITENEU}^2 = 2 \qquad \text{Approx(\#30)}$$

Haben wir es geschafft?

Die meisten Schüler lehnen sich erfahrungsgemäß zurück und behaupten, sie wüßten wie groß $\sqrt{2}$ sei! Das würde bedeuten, daß $\sqrt{2}$ ein Bruch wäre, der zwar eine große Zahl im Zähler und Nenner hätte, aber ein Element der Menge der rationalen Zahlen wäre. Hier stehenzubleiben würde einen völlig falschen Eindruck entstehen lassen. Mit einem CAS lassen sich jedoch Rundungsprobleme und Näherungsverfahren erlebbar machen, indem die Anzahl der Stellen erhöht wird.

```
#32: PrecisionDigits := 50                                User
```

$$\#33:\ \mathrm{SEITENEU} = \frac{1}{2}\cdot\left[\frac{665857}{470832} + \frac{2}{\frac{665857}{470832}}\right] \qquad \text{Sub(\#6)}$$

In #33 wurde zur Berechnung von SEITENEU nochmals der Wert von #23 verwendet. Es entsteht im *Exact*-Modus derselbe Bruch wie in #31,

$$\#34:\ \mathrm{SEITENEU} = \frac{886731088897}{627013566048} \qquad \text{Simp(\#33)}$$

mit **approX** jedoch eine Nährungszahl mit 49 Dezimalstellen

```
Approx(#34)
#35: SEITENEU =
          1.4142135623730950488016896235025302436149819257761
```

$$\#36:\ \left[\mathrm{SEITENEU} = \frac{886731088897}{627013566048}\right]^2 \qquad \text{User}$$

und ein Flächeninhalt, der schon sehr nahe bei 2 cm² liegt, jedoch dieses Maß noch nicht erreicht hat (#37).

```
Approx(#36)
                2
#37: SEITENEU =
          2.0000000000000000000000025435842395854372058427927
```

Diese Vorgangsweise läßt sich beliebig oft wiederholen. Es ergeben sich immer genauere neue Quadratseitenlängen. Die Berechnung des Flächeninhalts zeigt, daß jede neue Seite nur eine noch bessere Näherung ist.

Der Vergleich mit der Approximation von $\sqrt{2}$ 1 (#39) zeigt, daß bereits 23 Dezimalstellen übereinstimmen!

```
#38: √2                                                      User

Approx(#38)
#39: 1.4142135623730950488016887242096980785696718753769
```

Wir erhalten als Näherungen immer Brüche, aber $\sqrt{2}$ dürfte keine Bruch sein. Was dann? Eine neue, bisher noch nicht bekannte Zahl? Eine nicht periodische unendliche Dezimalzahl?

Ein solcher Schritt kann in dieser experimentellen Phase nicht vollzogen werden. Jedoch kann dadurch der Schüler folgende Idee mitnehmen: "Der Schritt zur irrationalen Zahl besteht darin, daß man die Möglichkeit, sich dieser Zahl zu nähern, zur Zahl erklärt."

Diese Vorgangsweise kann im Schüler in einem frühen Stadium des Lernprozesses die Grundidee eines iterativen Lösungsmodells entstehen lassen und die Begriffsbildung fördern. Das CAS stellt für Iterationen die Funktionen ITERATES und ITERATE zur Verfügung (Siehe programmtechnische Anmerkung 2, am Ende dieses Kapitels). Im Unterricht sollten jedoch vor der Anwendung dieser Funktion viele Einsetzdurchgänge durchgeführt werden, damit ein Schüler mit dieser zyklischen Maschine vertraut wird.

Die Funktion ITERATES wird mit Startwert 2 und 6 Iterationsschritten eingegeben, als 'Laufvariable' ersetzen wir SEITEALT durch *a*.

$$\#40:\ \mathrm{ITERATES}\left[\frac{1}{2}\cdot\left[a+\frac{2}{a}\right],\ a,\ 2,\ 6\right] \qquad \mathrm{User}$$

Die Vereinfachung liefert den Startwert und sechs Näherungen. Wir erhalten dieselben Brüche wie beim Substitutionsweg.

Simp(#40)

$$\#41:\ \left[2,\ \frac{3}{2},\ \frac{17}{12},\ \frac{577}{408},\ \frac{665857}{470832},\ \frac{886731088897}{627013566048},\ \frac{1572584048032918633353217}{1111984844349868137938112}\right]$$

Die Brüche werden näherungsweise in #42 mit 49 Dezimalstellen dargestellt. Man sollte jedoch bei der Anwendung dieser Funktion die Einstellung *Approximate* vornehmen.

```
Approx(#41)

#42: [2, 1.5,
        1.4166666666666666666666666666666666666666666666666,
        1.4142156862745098039215686274509803921568627450980,
        1.4142135623746899106262955788901349101165596221157,
        1.4142135623730950488016719273693282396963177789236O,
        1.4142135623730950488016719273693282396963177789236O]
```

Wird keine Anzahl von Iterationsschritten angegeben, so wird das Verfahren solange angewendet, bis erstmalig der neue Wert gleich dem alten Wert ist, sonst entsteht eine Endlosschleife. Damit läßt sich auch zeigen, daß es sehr wohl Flächen gibt, bei denen sich, bei geeintetem Startwert, die Seitenlänge nicht mehr ändert.

Ist etwa $\sqrt{4}$ 2 eine irrationale Zahl?

#43: PrecisionDigits := 20   User

#44: $\text{ITERATES}\left[\frac{1}{2}\cdot\left[a + \frac{4}{a}\right],\ a,\ 2,\ 6\right]$ User

#45: [2, 2, 2, 2, 2, 2, 2]   Simp(#44)

Die Auswertung von #44 ohne Angabe der Iterationsschritte

#46: $\text{ITERATES}\left[\frac{1}{2}\cdot\left[a + \frac{4}{a}\right],\ a,\ 2\right]$ User

liefert

#47: [2,2]   Simp(#46)

Im Unterricht erscheint es sinnvoll, Iterationsschritte anzugeben, da bei auftretenden Endlosschleifen der Schüler vor der Frage steht, ob er die Iteration mit **Esc** abgebrechen soll oder ob eine weitere Wartezeit noch eine Lösung liefern könnte.

2.1.3. Arithmetische Grundtätigkeiten

CAS beherrschen alle arithmetischen Grundtätigkeiten, die wir vom Taschenrechner kennen, wobei vor allem mit der Faktorisierung eine neue Dimension hinzukommt. An einigen - willkürlich herausgegriffenen - Beispielen sei dies aufgezeigt:

User=Simp(User)

#1: $\frac{10^{317} - 1}{9} =$

```
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
111111111111111111111111111
```

Diese Repunit-Zahl (= eine Zahl, die aus lauter Einsen besteht) zeigt nicht nur die Fähigkeit von CAS, sehr lange Zahlen manipulieren zu können, sondern ist auch ein Beispiel für eine eigenwillige Primzahl. (Zur PRIME-Funktion siehe Anmerkung 3, Ende dieses Kapitels.)

$$\#2: \quad \text{PRIME}\left[\frac{10^{317} - 1}{9}\right] = \text{true} \qquad \text{User=Simp(User)}$$

Eine andere Zahl, von der lange Zeit angenommen wurde, daß sie eine Primzahl wäre, ist die Zahl 2^{67} - 1. Zuerst hat dies bekanntlich Mersenne vermutet, später hat Lucas gezeigt, daß dem nicht so sein kann, und erst 1903 hat Cole nach der Investition von drei Jahren Sonntagsarbeit ("three years of sundays") die beiden Primfaktoren angegeben.

$$\#3: \quad 2^{67} - 1 = 147573952589676412927 \qquad \text{User=Simp(User)}$$

$$\#4: \quad \text{PRIME}(2^{67} - 1) = \text{false} \qquad \text{User=Simp(User)}$$

$$\#5: \quad \text{FACTOR}(2^{67} - 1) = 193707721 \cdot 761838257287 \quad \text{User=Simp(User)}$$

CAS rechnen auch mit komplexen, transzendenten und irrationalen Zahlen meist problemlos:

$$\#4: \quad \hat{e}^{\hat{\imath} \cdot \pi} = -1 \qquad \text{User=Simp(User)}$$

$$\#5: \quad \hat{e}^{2 \cdot \hat{\imath} \cdot \pi} = 1 \qquad \text{User=Simp(User)}$$

Nun noch ein Beispiel zum Rechnen mit komplexen Zahlen.

Beispiel 2.2: Wechselstromwiderstände

Ermittle den Betrag des Wechselstromwiderstands und den auftretenden Phasenwinkel einer Parallelschaltung eines ohmschen Widerstands von $R = 75\ \Omega$ und einer Spule mit einer Induktivität von $L = 0{,}5$ H. Die Parallelschaltung sei an das Lichtnetz angeschlossen.

Für die Parallelschaltung von Widerständen gilt bekanntlich: $\frac{1}{R_{ges}} = \frac{1}{R_1} + \frac{1}{R_2}$

Da bei induktiven Widerständen (Spule) die Stromstärke der Spannung nacheilt, beträgt (bei sehr kleinem ohmschen Widerstand der Spule) die Phasenverschiebung ungefähr $+90^0$ und wir können für den Widerstand einer Spule $R_L = i\ \omega\ L$ ansetzen. Damit lassen sich die gewünschten Werte leicht ermitteln. Der Wechselstromwiderstand (Impedanz) wird hier - wie bei den Elektrotechnikern üblich - mit Z bezeichnet.

$$\#1: \quad [\text{CASeMode} := \text{Sensitive}, \text{Angle} := \text{Degree}] \qquad \text{User}$$

$$\#2: \quad RL := \bullet \cdot L \cdot \hat{\imath} \qquad \text{User}$$

$$\#3: \quad \frac{1}{Z} = \frac{1}{R} + \frac{1}{RL} \qquad \text{User}$$

Nachdem wir nun die Informationen, die uns die Physik für unsere Aufgaben bereitstellt, angeschrieben haben, können wir nach *Z* auflösen und anschließend die konkreten Daten der gegebenen Schaltung einsetzen.

$$\#4: \quad Z = \frac{L^2 \cdot R \cdot \bullet^2}{L^2 \cdot \bullet^2 + R^2} + \frac{\hat{\imath} \cdot L \cdot R^2 \cdot \bullet}{L^2 \cdot \bullet^2 + R^2} \qquad \text{Solve(\#3)}$$

$$\#5: \quad Z = \frac{0.5^2 \cdot 75 \cdot 220^2}{0.5^2 \cdot 220^2 + 75^2} + \frac{\hat{\imath} \cdot 0.5 \cdot 75^2 \cdot 220}{0.5^2 \cdot 220^2 + 75^2} \qquad \text{Sub(\#4)}$$

$$\#6: \quad Z = 51.1988 + 34.9083 \cdot \hat{\imath} \qquad \text{Approx(\#5')}$$

Zur weiteren Auswertung ist es von Vorteil, die Größe *Z* mit der erhaltenen komplexen Zahl zu belegen. Dazu definieren wir:

$$\#7: \quad Z := 51.1988 + 34.9083 \cdot \hat{\imath} \qquad \text{User}$$

$$\#8: \quad |Z| \qquad \text{User}$$

$$\#9: \quad 61.9669 \qquad \text{Approx(\#8)}$$

Die Eingabe des Betragzeichens und die Funktion PHASE wird in Anmerkung 4 am Ende dieses Kapitels erklärt.

$$\#10: \quad \text{PHASE}(Z) \qquad \text{User}$$

$$\#11: \quad 34.2868 \qquad \text{Approx(\#10)}$$

Mit #9 und #11 können wir nun den Widerstand der Parallschaltung mit etwa 62Ω und Phasenverschiebung zwischen Strom und Spannung mit ca. 34^0 angeben.

Die komplexe Arithmetik, die mit dem Symbol i rechnet, führt uns zu jenem Bereich, um den es beim Einsatz von CAS in erster Linie geht: das symbolische Rechnen.

2.2. Symbolisches Hilfsmittel

Der große Fortschritt eines CAS zeigt sich darin, daß auch mit Symbolen algebraische Umformungen möglich werden. Dies ist Grundvoraussetzung für das allgemeine Lösen von Gleichungen, das Bestimmen von Ableitungen und unbestimmten Integralen oder das Lösen von Differentialgleichungen.

2.2.1. Lösen von Gleichungssystemen

Sehr häufig treten in der Schulmathematik bei Modellierungsvorgängen Bedingungen an Punkte einer Funktion auf. Die dabei entstehenden Gleichungssysteme sind zwar im Prinzip von den Schülern lösbar, der tatsächliche Zeitaufwand und die einhergehende Fehlerhäufigkeit lassen jedoch aufwendigere Gleichungssysteme nicht zu. Wenn die Betrachtung der Funktion im Mittelpunkt steht, kann das Gleichungssystem vom CAS gelöst werden.

Beispiel 2.3: Polynomfunktion - Umkehrung der Kurvendiskussion

Von einer Polynomfunktion sechsten Grades kennt man alle rellen Nullstellen, N_1 = (-2,0), N_2 = (-1.5,0), N_3 = (-0.8,0), N_4 = (1.5,0), N_5 = (2.5,0) und zwei weitere Punkte des Schaubildes P = (0, 45/2) und Q = (1, 243/16)!

Berechne die Funktionsgleichung!

Voreinstellungen:

```
#1:  InputMode := Word                                  User
#2:  CASeMode := Sensitive                              User
```

Wir definieren eine allgemeine Funktion *F* als Polynom sechsten Grades mit unbekannten Koeffizienten.

$$\#3: \quad F(x) := a\cdot x^6 + b\cdot x^5 + c\cdot x^4 + d\cdot x^3 + e\cdot x^2 + f\cdot x + g \qquad \text{User}$$

Wir benötigen zur Bestimmung der Koeffizienten *a* bis *g* sieben Gleichungen mit sieben Unbekannten.

Die Angaben der Punkte werden in Gleichungen übersetzt, wir erhalten sieben Gleichungen:

$$\#4: \quad F(-2) = 0 \qquad \text{User}$$

$$\#5: \quad F(-1.5) = 0 \qquad \text{User}$$

$$\#6: \quad F(-0.8) = 0 \qquad \text{User}$$

$$\#7: \quad F(1.5) = 0 \qquad \text{User}$$

$$\#8: \quad F(2.5) = 0 \qquad \text{User}$$

$$\#9: \quad F(1) = \frac{243}{16} \qquad \text{User}$$

$$\#10: \quad F(0) = \frac{45}{2} \qquad \text{User}$$

Durch Vereinfachung erhalten wir:

$$\#11: \quad 64\cdot a - 32\cdot b + 16\cdot c - 8\cdot d + 4\cdot e - 2\cdot f + g = 0 \qquad \text{Simp(\#4)}$$

$$\#12: \quad \frac{729\cdot a}{64} - \frac{243\cdot b}{32} + \frac{81\cdot c}{16} - \frac{27\cdot d}{8} + \frac{9\cdot e}{4} - \frac{3\cdot f}{2} + g = 0 \qquad \text{Simp(\#5)}$$

Simp(#6)

$$\#13: \quad \frac{4096\cdot a}{15625} - \frac{1024\cdot b}{3125} + \frac{256\cdot c}{625} - \frac{64\cdot d}{125} + \frac{16\cdot e}{25} - \frac{4\cdot f}{5} + g = 0$$

Simp(#7)

$$\#14: \quad \frac{729\cdot a}{64} + \frac{243\cdot b}{32} + \frac{81\cdot c}{16} + \frac{27\cdot d}{8} + \frac{9\cdot e}{4} + \frac{3\cdot f}{2} + g = 0$$

Simp(#8)

$$\#15: \quad \frac{15625\cdot a}{64} + \frac{3125\cdot b}{32} + \frac{625\cdot c}{16} + \frac{125\cdot d}{8} + \frac{25\cdot e}{4} + \frac{5\cdot f}{2} + g = 0$$

$$\#16:\ a + b + c + d + e + f + g = \frac{243}{16} \qquad \text{Simp(\#9)}$$

$$\#17:\ g = \frac{45}{2} \qquad \text{Simp(\#10)}$$

Als einzige Variable ist *g* in #17 festgelegt. Beim handschriftlichen Rechnen müßte nun ein Gleichungssystem mit sechs Gleichungen und sechs Unbekannten gelöst werden. Mit einem CAS kann dies durch die Eingabe der Bedingungen (begrenzt durch eckige Klammern) mit **soLve** erfolgen.

User

$$\#18:\ \left[F(-2)= 0, F(-1.5)= 0, F(-0.8)= 0, F(1.5)= 0, F(2.5)= 0, F(1)= \frac{243}{16}, F(0)= \frac{5}{2}\right]$$

Es entsteht der Lösungsvektor #19.

Solve(#18)

$$\#19:\ \left[a= -1, b= \frac{11}{5}, c= \frac{42}{5}, d= -\frac{289}{20}, e= -\frac{1907}{80}, f= \frac{171}{8}, g= \frac{45}{2}\right]$$

Wir setzen die berechneten Koeffizienten in die in #3 definierte Funktion *F* ein

Sub(#3)

$$\#20: F(x):=(-1)\cdot x^6 + \frac{11}{5}\cdot x^5 + \frac{42}{5}\cdot x^4 + \left[-\frac{289}{20}\right]\cdot x^3 + \left[-\frac{1907}{80}\right]\cdot x^2 + \frac{171}{8}\cdot x + \frac{5}{2}$$

und vereinfachen den Funktionsterm.

Simp(#20')

$$\#21:\ F(x):= -x^6 + \frac{11\cdot x^5}{5} + \frac{42\cdot x^4}{5} - \frac{289\cdot x^3}{20} - \frac{1907\cdot x^2}{80} + \frac{171\cdot x}{8} + \frac{45}{2}$$

Dieser Term wird faktorisiert

Fctr(#21')

$$\#22:\ F(x) := \frac{(x + 2)\cdot(2\cdot x + 3)\cdot(3 - 2\cdot x)\cdot(2\cdot x - 5)^2\cdot(5\cdot x + 4)}{80}$$

und gezeichnet (Abb. 2.2)

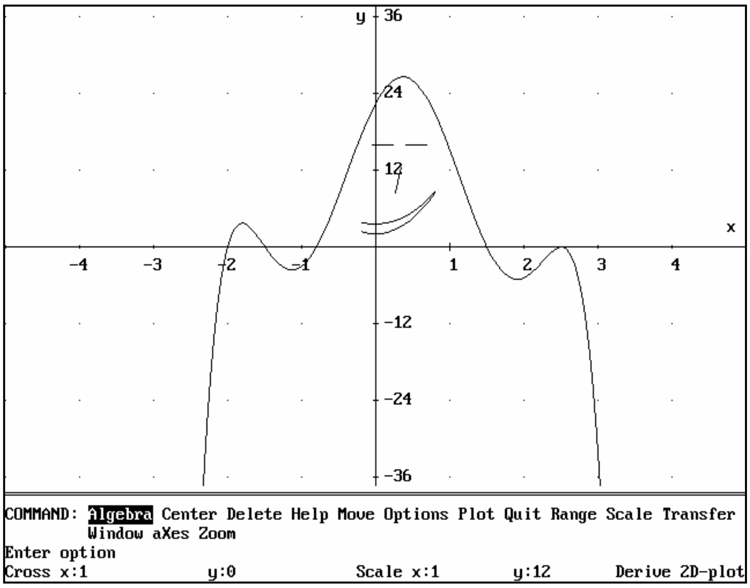

Abb. 2.2: Was sieht man beim Betrachten dieser Kurve? Vgl. mit Kap. 3.2

Es können die Nullstellen überprüft werden, Monotonieintervalle, lokale Extremstellen oder Wendepunkte rechnerisch oder im Grafikfenster mit **F3** durch Wanderung auf dem Funktionsschaubild näherungsweise gefunden werden.

2.2.2. Differenzieren und Integrieren

Ein Schwerpunkt der Schulmathematik besteht in der Herleitung und Interpretation der Differentialquotienten und in der Berechnung und im Verstehen der Bedeutung von bestimmten und unbestimmten Integralen.

In welcher Beziehung stehen Differentiation und Integration? Das folgende Beispiel soll einen Zugang zu dieser Frage öffnen.

Beispiel 2.4: Zusammenhang: Differenzieren - Integrieren

Erhält man wieder die Ausgangsfunktion, wenn eine Funktion zuerst differenziert und danach die Ableitungsfunktion integriert wird? Dieser Zusammenhang ist zu untersuchen, die Auswirkungen des

Differenzierens und Integrierens sollen hinterfragt werden!

Besteht der Schwerpunkt eines Beispiels im Bearbeiten von Termen, erweist es sich als günstig, mit der Einstellung **Option Precision** *Exact* zu arbeiten.

```
#1:  Precision := Exact                                 User
```

Eine Funktion *F* wird definiert

$$\#2:\quad F(x) := \frac{x^2 - x + 2}{x^2 - x - 2} \qquad \text{User}$$

und die rechte Seite von #2 mit **Factor** bearbeitet.

$$\#3:\quad F(x) := \frac{x^2 - x + 2}{(x + 1)\cdot(x - 2)} \qquad \text{Fctr(\#2')}$$

Diese rationale Funktion hat zwei Unstetigkeitsstellen (keine hebbaren Lücken), die Funktion ist über $\mathbb{R}\backslash\{-1,2\}$ definiert. Der Graph kann sofort gezeichnet werden.

Mit **Calculus Differentiate** wird der Term der Funktion nach x differenziert.

$$\#4:\quad \frac{d}{dx} \frac{x^2 - x + 2}{x^2 - x - 2} \qquad \text{Dif(User,x)}$$

Der Differentialquotient wird durch den vereinfachten Term #5 und den faktorisierten Term #6 beschrieben.

$$\#5:\quad \frac{4\cdot(1 - 2\cdot x)}{(x^2 - x - 2)^2} \qquad \text{Simp(\#4)}$$

$$\#6:\quad \frac{4\cdot(1 - 2\cdot x)}{(x + 1)^2\cdot(x - 2)^2} \qquad \text{Fctr(\#5)}$$

Dieser Term wird mit **Calculus Integrate** nach x integriert, das Ergebnis des unbestimmten Integrals faktorisiert und mit der gegebenen Funktion F verglichen.

$$\#7:\quad \int \frac{4\cdot(1 - 2\cdot x)}{(x^2 - x - 2)^2}\, dx \qquad \text{Int(\#5,x)}$$

$$\#8:\quad \frac{4}{x^2 - x - 2} \qquad \text{Simp(\#7)}$$

$$\#9:\quad \frac{4}{(x + 1)\cdot(x - 2)} \qquad \text{Fctr(\#8)}$$

Zur besseren Übersicht werden die beiden Funktionen in #10 und #11 definiert und einer Betrachtung unterzogen.

Die beiden Terme haben ein sehr unterschiedliches Aussehen, so daß die Vermutung naheliegt, daß diese beiden Funktionen außer dem Definitionsbereich wenig gemeinsam haben. Der Zähler der neu entstandenen Funktion ist durch eine Zahl festgelegt, der Zähler der Stammfunktion F ist ein Polynom 2.Grades. Besteht nun ein Zusammenhang zwischen diesen beiden Funktionen?

$$\#10:\quad F1(x) := \frac{4}{(x + 1)\cdot(x - 2)} \qquad \text{User}$$

$$\#11:\quad F(x) := \frac{x^2 - x + 2}{(x + 1)\cdot(x - 2)} \qquad \text{Fctr(\#2')}$$

Ein möglicher Versuch, die Beziehung zwischen den Termen allgemein zu beschreiben, besteht im Subtrahieren der beiden Funktionsterme:

$$\#12:\ F1(x) - F(x) = -1 \qquad \text{User=Simp(User)}$$

Wir erhalten als Differenz -1. Dieser Sachverhalt läßt sich folgendermaßen interpretieren: Die neue Funktion *F1* unterscheidet sich an jeder Stelle des Definitionsbereichs von der Funktion *F* dadurch, daß an jeder Stelle von *F1* der Funktionswert um 1 kleiner ist als die Werte der Funktion *F*. Dies bedeutet, daß die Funktion *F1* aus einer Verschiebung des Graphen von *F* entlang der y-Achse hervorgeht. Beide Funktionen haben dieselbe Ableitungsfunktion.

Werden die Terme der Funktionen *F* und *F1* expandiert wird offensichlich, daß die Terme einander nur durch das konstante Glied 1 unterscheiden.

$$\#13:\ F(x) := \frac{4}{3\cdot(x-2)} - \frac{4}{3\cdot(x+1)} + 1 \qquad \text{Expd(\#2')}$$

$$\#14:\ F1(x) := \frac{4}{3\cdot(x-2)} - \frac{4}{3\cdot(x+1)} \qquad \text{Expd(\#10')}$$

Es ist also zu erkennen, daß beim Differenzieren und nachträglichem Integrieren der konstante Summand *c* zu beachten ist. Durch die Funktion *FA* ist also eine Schar von Stammfunktionen gegeben:

$$\#15:\ FA(x) := \frac{4}{(x+1)\cdot(x-2)} + c \qquad \text{User}$$

Eine weitere Bedingung der Funktion aus #2, z.B.: *F*(0)=-1, führt durch Lösen einer Gleichung mit einer Unbekannten zur Festlegung von *c*.

$$\#16:\ FA(0) = -1 \qquad \text{User}$$

$$\#17:\ c = 1 \qquad \text{Solve(\#16)}$$

Wird *c* eingesetzt und der Funktionsterm faktorisiert, erhalten wir tatsächlich unsere ursprüngliche Funktion *F*:

$$\#27:\ FA(x) := \frac{4}{(x+1)\cdot(x-2)} + 1 \qquad \text{Sub(\#24)}$$

$$\#28:\ FA(x) := \frac{x^2 - x + 2}{(x+1)\cdot(x-2)} \qquad \text{Fctr(\#27')}$$

Diese Beziehung läßt sich grafisch darstellen (Abb. 2.3)

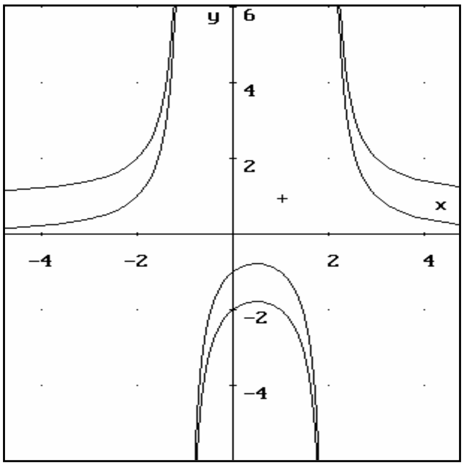

Abb. 2.3: Die Stammfunktion F und die verschobene Funktion F1

Ohne CAS muß bei derartigen Beispielen die Partialbruchzerlegung händisch durchgeführt werden. Jetzt wird diese zeitaufwendige Vorgangsweise an das CAS übertragen, und die Interpretation kann im Mittelpunkt des Interesses stehen.

Beispiel 2.5: Berechnung bestimmter Integrale

Berechne von der Funktion F aus Beispiel 2.4 die bestimmten Integrale in den Grenzen von -0.5 bis 1 und von -1 bis 0.

Mit der Befehlsfolge **Calculus Integrate** wird nach dem zu integrierenden Term, der Integrationsvariablen und nach den Integrationsgrenzen gefragt.

#1: $$\int_{-0.5}^{1} \frac{x^2 - x + 2}{x^2 - x - 2}\, dx$$ Int(User,x)

Die Auswertung ergibt bei Vereinfachung den Ausdruck #14. Das CAS liefert manchmal unerwartete Ausgaben, die der Schüler eventuell noch nicht kennengelernt hat. Die einfachste Form des Computeralgebra-Systems ist oftmals nicht identisch mit der erwarteten Darstellung.

#2: $$\frac{3}{2} - \frac{4 \cdot LN(10)}{3}$$ Simp(#1)

Mit **approX** erhalten wir einen negativen Wert für das bestimmte Integral. Dieser läßt sich durch die grafische Interpretation aus Abb. 2.4 erklären.

#3: -1.57011 Approx(User)

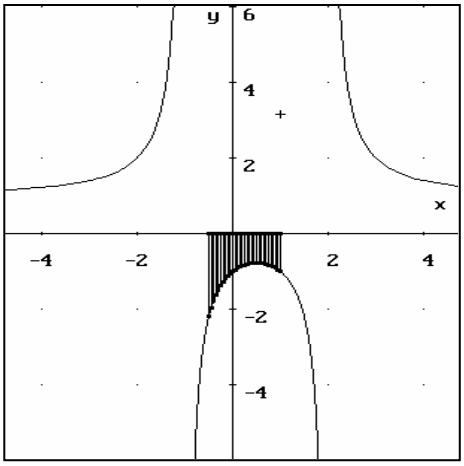

Abb. 2.4: Graph der Funktion F

Das bestimmte Integral in den Grenzen von -1 bis 0 liefert -∞. Auch das läßt sich grafisch (Abb. 2.4) einsichtig machen.

$$\#4: \int_{-1}^{0} \frac{x^2 - x + 2}{(x + 1)\cdot(x - 2)}\, dx \qquad \text{Int(User,x)}$$

$$\#5: -\infty \qquad \text{Simp(\#4)}$$

Für eine genauere Betrachtung der Ergebnisse bei der Ermittlung bestimmter Integrale vergleiche Kap. 5.2.1 (Ergebniskritik).

2.2.3. Lösen von Differentialgleichungen

Die Behandlung einfacher Differentialgleichungen steht am Ende und am Rande der Schuanalysis. Dies vor allem deswegen, weil der damit einhergehende Aufwand für die Lösung häufig zu groß ist. Gerade in einem anwendungsorientierten Mathematikunterricht treten Probleme, die auf Differentialgleichungen führen, häufiger ins Blickfeld. Bei der Verwendung von CAS stehen aber in erster Linie die Modellbildung und die Interpretation der Ergebnisse im Vordergrund.

Beispiel 2.6: Wann ist die Pizza fertig?

Walter hat seine Freundin Regina zur ersten selbstgebackenen Pizza eingeladen. Leider hat er vergessen, das Rohr vorzuwärmen. Die Pizza soll 20 Minuten bei ca. 200 °C gebacken werden. Es ist 18 Uhr, Regina kommt um 18.45. Die Temperatur des Backrohres nach dem Einschalten läßt sich beschreiben durch $O(t) = 200 - 180e^{-0.3t}$ *(in °Celsius).*

Die Temperaturzunahme ist proportional zur Differenz zwischen der Temperatur des Backrohrs und der Pizza mit einem Proportionalitätsfaktor von 0.2 .

Nach wieviel Minuten erreicht die Pizza 95% der Endtemperatur? Wenn Walter von da an 20 Minuten wartet, kann er dann noch rechtzeitig fertig werden?

Die Temperaturzunahme der Pizza, die zugleich mit dem Einschalten in den Herd kommt, ergibt sich durch:

$$\frac{dP}{dt} = 0.2\,(\,O(t) - P(t)\,)$$

Die DERIVE-Funktion DSOLVE1(u,v,x,y,x0,y0) versucht die Lösung der Differentialgleichung $u(x,y) + v(x,y)\,y' = 0$ mit der Anfangsbedingung *x0, y0* zu finden. Mit Hilfe von DSOLVE1 können wir uns nun an die Lösung obiger Fragestellung (Differentialgleichung) heranwagen.

$$\#1: \quad O(t) := 200 - 180\cdot ê^{-0.3\cdot t} \qquad \text{User}$$

Zur Übersetzung der mathematischen Schreibweise in die Notation des DSOLVE1-Moduls: die rechte Seite obiger Differentialgleichung wird zu $u(t,p)$, für $v(t,p)$ ergibt sich -1, x entspricht t und y entspricht der gesuchten Temperatur der Pizza p. Zum Zeitpunkt 0 nehmen wir eine Anfangstemperatur von 20°C an (Zimmertemperatur).

$$\#2: \quad \text{DSOLVE1}(0.2\cdot(O(t) - p),\ -1,\ t,\ p,\ 0,\ 20) \qquad \text{User}$$

$$\#3: \quad (200 - p)\cdot(ê^{t})^{1/5} + \frac{360}{(ê^{t})^{1/10}} - 540 = 0 \qquad \text{Simp(\#2)}$$

$$\#4: \quad p = -\frac{540}{(ê^{t})^{1/5}} + \frac{360}{(ê^{t})^{3/10}} + 200 \qquad \text{Solve(\#3)}$$

Nun sind wir bereits in der Lage, den Temperaturverlauf bei Backrohr und Pizza grafisch darzustellen. Die Grafik (Abb. 2.5) zeigt auch bereits, daß die Pizza rechtzeitig fertig wird.

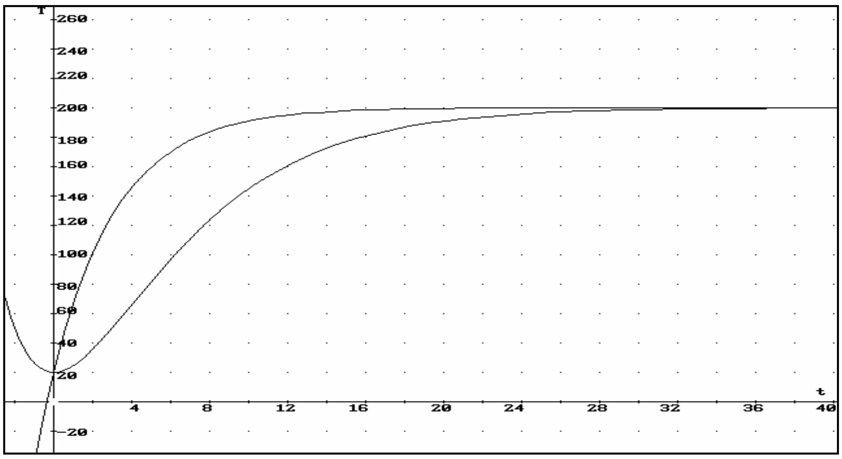

Abb. 2.5: Temperaturzunahme

Wir wollen uns aber noch rasch vergewissern, ob dem wirklich so ist:

$$\#5: \quad \text{Precision} := \text{Approximate} \qquad \text{User}$$

$$\#6: \quad -\frac{540}{(ê^{t})^{1/5}} + \frac{360}{(ê^{t})^{3/10}} + 200 = 200\cdot 0.95 \qquad \text{User}$$

Gleichung #6 wird nun näherungsweise gelöst. Dazu ist es notwendig, sinnvolle Intervallgrenzen vorzugeben. Diese können hier nur aufgrund des Problemkontextes bzw. der Grafik gefunden werden., z.B. [0;20]

```
#7:  t = 19.4436                                    Solve(#6)
```

Walter hat damit sogar noch genügend Zeit, Blumen für Regina auf den Tisch zu stellen.

2.2.4. Summen und Produkte

Im Unterschied zum Arbeiten mit dem numerischen Taschenrechner können wir uns auch Summen und Produkte symbolisch - d.h. hier allgemein - berechnen lassen.

Summen

Beispiel 2.7: Die Restschuld

Leite eine Formel für die Restschuld bei der Tilgung eines Kredits her, wenn du nur berücksichtigst, daß für den aufgenommenen Kredit Zinsen berechnet werden und die Rückzahlung abgezogen wird.

Beginnen wir damit, daß wir den Vorgang so formulieren, daß wir dabei der Alltagssprache noch nahe bleiben. Mit Wortvariablen ausgedrückt ergibt sich also für die jeweils neue Restschuld nach Ablauf einer weiteren Zinsperiode:

```
#1:  [InputMode:= Word, Precision:= Approximate,
      Notation:=Dezimal]                                User
User
#2:  schuld_neu := schuld_alt +
                              schuld_alt·zinssatz
                              ─────────────────── - rückzahlung
                                     100
Simp(#2')
                                       ┌     zinssatz ┐
#3: schuld_neu := schuld_alt·│1 + ──────── │ - rückzahlung
                                       └       100    ┘
```

Etwas stärker formalisiert bekommen wir die Berechnungsvorschrift

```
#4:  s0·q - r                                           User
```

mit s_0 als ursprüngliche Schuld, $q = 1 + \frac{Zinssatz}{100}$ 4 als Aufzinsungsfaktor und r als Höhe der Rückzahlungsrate.

Berechnen wir nun die Restschuld nach ein, zwei, drei, vier Verzinsungsperioden, indem wir schrittweise das jeweilige Ergebnis wieder in die ursprüngliche Berechungsvorschrift #5 (mit **Manage Substitute**) einsetzen:

```
#5:  s0·q - r                                          User

#6:  (s0·q - r)·q - r                                  Sub(#5)

#7:  ((s0·q - r)·q - r)·q - r                          Sub(#5)

#8:  (((s0·q - r)·q - r)·q - r)·q - r                  Sub(#5)
```

Wenn wir die einzelnen Terme der Reihe nach vereinfachen, sehen wir, worauf die Ermittlung der Restschuld hinausläuft:

```
#9:  s0·q - r                                          User
                             2
#10: (s0·q - r)·q - r = q ·s0 - q·r - r        User=Simp(User)

User=Simp(User)
                                   3      2
#11: ((s0·q - r)·q - r)·q - r = q ·s0 - q ·r - q·r - r

User=Simp(User)

#12: (((s0·q - r)·q - r)·q - r)·q - r =

                              4      3      2
                             q ·s0 - q ·r - q ·r - q ·r - r
```

Wir haben so eine Vermutung erhalten, wie die Formel für die Restschuld aussehen muß. Nun können wir die Beziehung für die Restschuld mit Hilfe von SUM formulieren:

$$\#13:\ \mathrm{RESTSCHULD}(n, s0, q, r) := q^n \cdot s0 - \sum_{i=1}^{n-1} q^i \cdot r \qquad \text{User}$$

Mit Hilfe des Befehls **Simplify** wird nun die formal angeschriebene Summe auch berechnet:

Simp(#13')

$$\#14:\ \mathrm{RESTSCHULD}(n, s0, q, r) := q^n \cdot \left[s0 - \frac{r}{q - 1}\right] + \frac{q \cdot r}{q - 1}$$

Natürlich sollte es einem Schüler keine Schwierigkeit bereiten, die Summe auch händisch zu ermitteln, es wäre ein Training zur Anwendung der geometrischen Reihe. Man muß sich nur klar darüber sein, welche Ziele man verfolgen will. Wenn wir die Beziehung für die Restschuld im CAS formulieren, haben wir die Möglichkeit, bei der Beantwortung vieler Fragen, die im Zusammenhang mit Krediten auftauchen können, auf einer Ebene zu arbeiten, die der umgangssprachlichen Form sehr nahe ist. Einige Beispiele dieser Art sollen dies verdeutlichen.

Mit der Funktion aus #14 können wir darangehen, Fragen zu beantworten, wie etwa: Wie entwickelt sich die Restschuld? Wie lange dauert es bei einer bestimmten Rückzahlungsrate *r*, bis die Schuld völlig getilgt ist? Wie groß muß die Rückzahlungsrate *r* sein, damit das ausgeliehene Kapital (Schuld) innerhalb einer bestimmten Zeit getilgt ist?

Die Schuld soll z.B. nach 10 Jahren bei 10% Verzinsung getilgt werden, wie hoch muß die Rückzahlungsrate gewählt werden?

```
#15: RESTSCHULD(10, 100000, 1.1, rückzahlungsrate) = 0   User

#16: rückzahlungsrate = 17364.0                     Solve(#15)
```

Oder: Nach wie vielen Jahren ist die Schuld getilgt, wenn jährlich 12 000,- zurückgezahlt werden?

```
#17: RESTSCHULD(jahre, 100000, 1.1, 12000) = 0           User

#18: jahre = 19.7992                                Solve(#17)
```

Ein kleiner Test unserer Beziehung zeigt weiters, daß z.B. eine Schuld von 100 000,- bei einer Verzinsung von 10% p.a. und einer Rückzahlung von 11 000.- weiter anwächst, anstatt geringer zu werden.

```
User=Simp(User)

#19: RESTSCHULD(5, 100000, 1.1, 11000) = 104894.9
```

Produkte

Durch die symbolische Berechnung von Summen und Produkten ergeben sich neue Möglichkeiten, im Mathematikunterricht zu herkömmlichen Aufgaben experimentelle Zugänge zu eröffnen. Dazu eine bekannte Aufgabe aus der Wahrscheinlichkeitsrechnung:

Beispiel 2.8: Das Geburtstagsproblem

Wie viele Personen müssen im Zimmer sein, damit man mit einer Wahrscheinlichkeit von mindestens 50% sagen kann, daß mindestens zwei von ihnen am selben Tag des Jahres Geburtstag haben?

Das Reizvolle an dieser Aufgabe ist, daß die Lösung genauso einfach wie überraschend ist. Wären z.B. fünf Personen in dem Raum, so beträgt die Wahrscheinlichkeit, daß alle an verschiedenen Tagen des Jahres ihren Geburtstag feiern:

$$\text{P(Geburtstag an verschiedenen Tagen)} = 1 \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \frac{362}{365} \cdot \frac{361}{365} =$$

$$(1 - \frac{1}{365}) \cdot (1 - \frac{2}{365}) \cdot (1 - \frac{3}{365}) \cdot (1 - \frac{4}{365}) = \prod_{i=1}^{4} (1 - \frac{i}{365})$$

Mit Hilfe der Gegenwahrscheinlichkeit ergibt sich damit folgender Ansatz:

$$1 - \prod_{i=1}^{n-1} (1 - \frac{i}{365}) \geq 0,5$$

Wir sehen (Abb. 2.6): Liegt bei fünf anwesenden Personen die Wahrscheinlichkeit bereits bei 2,7% (#3), so ergibt sich bei 23 Personen bereits eine Wahrscheinlichkeit von über 50% (#5). Da die Klassenschülerzahl oft in diesem Bereich liegt, kann bei dieser Aufgabe leicht die Probe aufs Exempel gemacht werden.

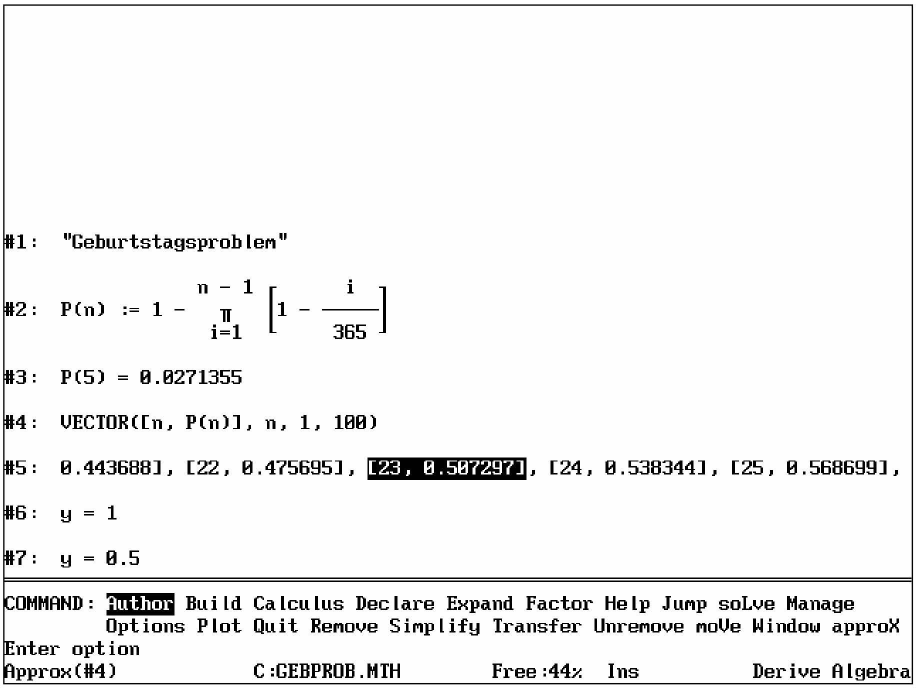

Abb.2.6: Wann ist die Wahrscheinlichkeit 50% ?

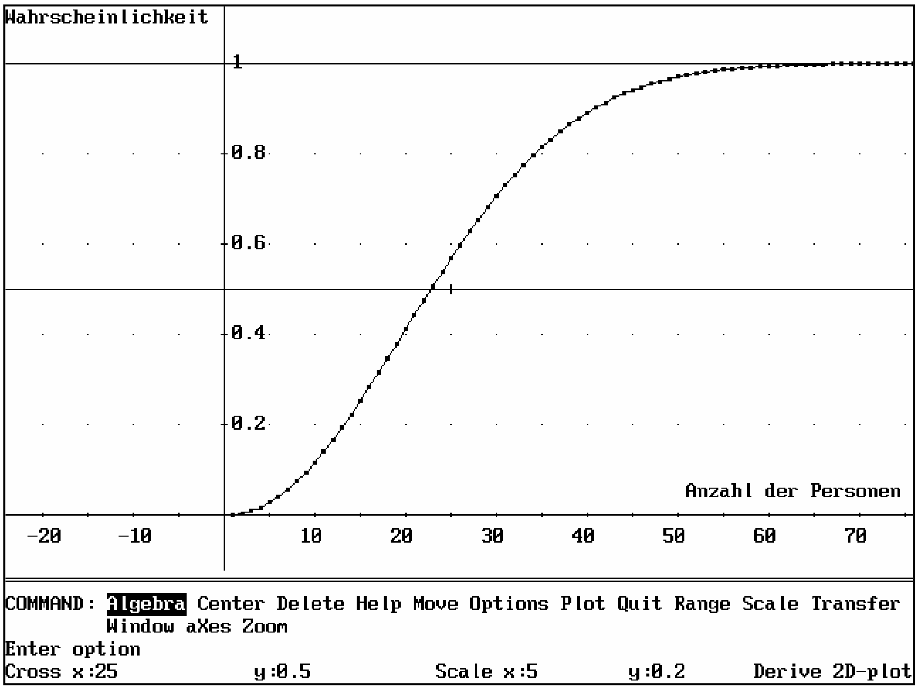

Abb. 2.6: Geburtstagsproblem

Wie die Wahrscheinlichkeit, daß zwei Personen am gleichen Tag Geburtstag haben, mit der Anzahl der anwesenden Personen ansteigt, zeigt Abb. 2.7, die entsteht, wenn wir die Wertepaare aus #5 plotten.

Beziehung #2 liefert uns im Zusammenwirken mit der Rechenleistung des Systems Informationen, die weit über die ursprüngliche Problemstellung hinausgehen. Wir sehen nun, wie die Wahrscheinlichkeit mit der Personenanzahl zusammenhängt - dies nicht nur an einer Stelle.

2.3. Das Computeralgebra-System als algorithmisches Hilfsmittel

Roland Fischer spricht in seinem Buch 'Mensch und Mathematik' von der "Chance der materiellen Auslagerung bestimmter Fähigkeiten des Menschen mit Hilfe des Computers" [Fischer, 1985, S.259]. Durch Computeralgebra-Systeme ergibt sich vor allem die Möglichkeit der Auslagerung algorithmischer Fertigkeiten, aber auch die Neufassung des Begriffs Algorithmus.

2.3.1. Ausführen implementierter Algorithmen

Der erste Schritt ist das Nutzen der Algorithmen, die vom CAS zur Verfügung gestellt werden. Das können einerseits solche sein, die exakte Lösungen liefern, und andererseits Algorithmen, die zu Näherungslösungen führen. Das folgende Beispiel soll das näherungsweise Lösen mit einem CAS zeigen.

Beispiel 2.9: Das Plancksche Strahlungsgesetz [vgl. Dorninger, 1988**]**

Das Emissionsvermögen eines schwarzen Körpers ist gegeben durch die Funktion E(λ) in Zeile #2. Dabei bedeuten λ die Wellenlänge der Strahlung, t die absolute Temperatur des Körpers, c die Lichtgeschwindigkeit und k die Boltzmannsche Konstante (alle Größen sind positiv). Man bestimme das Maximum von E(λ) für λ > 0!

Die erste Ableitung wird vom CAS berechnet. (Für λ werden -mittels Declare Variable- nur positive reelle Zahlen zugelassen.)

#1: $\lambda :\in \text{Real } (0, \infty)$ User

#2: $E(\lambda) := \dfrac{c^2 \cdot h}{\lambda^5} \cdot \dfrac{1}{\hat{e}^{c \cdot h/(k \cdot \lambda \cdot t)} - 1}$ User

#3: $\dfrac{d}{dl} E(\lambda)$ Dif(#2',l)

#4: $\dfrac{c^2 \cdot h \cdot (\hat{e}^{c \cdot h/(k \cdot \lambda \cdot t)} \cdot (c \cdot h - 5 \cdot k \cdot \lambda \cdot t) + 5 \cdot k \cdot \lambda \cdot t)}{k \cdot \lambda^7 \cdot t \cdot (\hat{e}^{c \cdot h/(k \cdot \lambda \cdot t)} - 1)^2}$ Simp(#3)

Man stellt fest, daß der Ausdruck sehr komplex ist. In solchen Fällen versucht man häufig, durch Substitution zu einfacheren, überschaubaren Termstrukturen zu kommen. Man ersetzt den Exponenten durch die Variable *x*.

#5: $x = \dfrac{c \cdot h}{k \cdot \lambda \cdot t}$ User

Die Gleichung muß nun nach λ aufgelöst werden, um λ durch einen Term, der von *x* abhängt, zu ersetzen.

$$\#6:\quad \lambda = \frac{c\cdot h}{k\cdot t\cdot x} \qquad \text{Solve(\#5)}$$

$$\#7:\quad \lambda := \frac{c\cdot h}{k\cdot t\cdot x} \qquad \text{User}$$

$$\#8:\quad \frac{k^6\cdot t^6\cdot x^6\cdot(\hat{e}^x\cdot(x-5)+5)}{c^4\cdot h^5\cdot(\hat{e}^{2\cdot x}-2\cdot\hat{e}^x+1)} \qquad \text{Simp(\#4)}$$

Im *Approximate*-Modus kann man näherungsweise die Nullstellen mit dem **soLve**-Befehl ermitteln. Wenn man den Bruchterm genauer betrachtet, erkennt man, daß es genügt, die Nullstellen der Funktion in Zeile #10 zu suchen. DERIVE fragt nach dem Intervall, in dem gesucht werden soll. Akzeptiert man das vom CAS vorgeschlagene Intervall [-10,+10], liefert DERIVE den Wert $x = 0$ (Zeile #11). Dieser Wert ist aber unbrauchbar.

```
#9:  Precision := Approximate                           User
  x
#10: F(x) := ê ·(x - 5) + 5                              User

#11: x = 0                                       Solve(#10)
```

Ohne Visualisierung wäre es nicht leicht, ein brauchbares Intervall zu finden (Abb. 2.8).

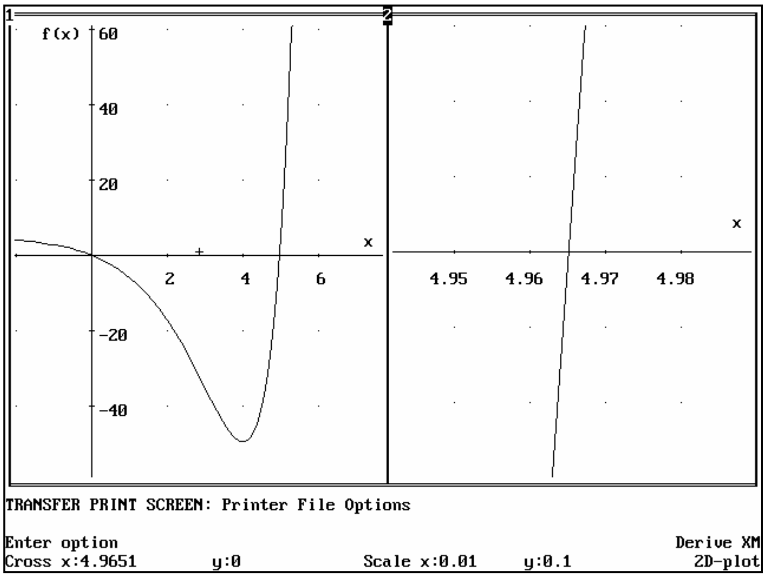

Abb. 2.7:Nullstellen durch Visualisieren

Nun gibt es zwei Möglichkeiten:

(1) Man ermittelt die Nullstelle gleich aus der Zeichnung. Um zu einer besseren Genauigkeit zu kommen, vergrößert man den Ausschnitt um die gesuchte Nullstelle durch **Zoom** oder mit Hilfe des **Range**-Befehls so lange, bis die Nullstelle mit ausreichender Genauigkeit im Grafikfenster abgelesen werden kann (siehe Fenster 2). In Abb. 2.8 zeigt *Cross* im Fenster links unten die Position des Cursors an. Man erhält $x \approx 4.9651$.

(2) Man nimmt aus dem Grafikfenster das geeignete Intervall, um die Nullstelle von #10 mit **soLve** im *Approximate*-Modus zu ermitteln. Das kann entweder mit dem **soLve**-Befehl erfolgen, dann fragt das CAS nach dem Intervall, oder man verwendet wie in Zeile #12 die SOLVE-Funktion und gibt das Intervall sofort mit ein.

```
#12: SOLVE(F(x), x, 4, 6)                                    User

#13: [x = 4.96511]                                      Simp(#12)
```

Wie man sieht, ist die Genauigkeit der aus dem Graphen ermittelten Nullstelle nicht viel schlechter. Sollte die Genauigkeit nicht ausreichen, kann sie, wie in Zeile #14 zu sehen ist, erhöht werden.

```
#14: PrecisionDigits := 20                                   User

#15: [x = 4.9651142317442804596]                        Simp(#12)
```

Schließlich wäre jetzt auch noch λ durch Einsetzen zu ermitteln.

An weiteren Beispielen werden wir deutlich machen, daß natürlich nicht alle Algorithmen 'Black Boxes' bleiben sollen. Der von uns vertretene didaktischen Ansatz wird in Kap. 3 und Kap. 4 näher erläutert.

2.3.2. Implementieren von Algorithmen durch den Benutzer

Nicht immer sind jene Algorithmen im CAS verfügbar, die im Unterricht gerade betrachtet werden. Dafür kann es zwei Gründe geben:

Der gewünschte Algorithmus ist im System nicht vorhanden (dies ist bei der enormen Leistungsfähigkeit und wachsenden Vielseitigkeit der CAS natürlich immer seltener der Fall), oder der gewünschte Algorithmus ist nicht direkt zugänglich. Dies liegt vor allem daran, daß der Kern der CAS nicht für den Benutzer einsehbar ist. (Beim Vorgänger von DERIVE - dem Programmpaket µ-MATH - war dies noch der Fall. Hier konnten die einzelnen LISP-Programme noch eingesehen werden. Dies war für all jene Benutzer interessant, die im LISP-Dialekt µ-LISP sattelfest waren). Fehlen solche spezifischen Programmiersprachenkenntnisse, so ist auch ein offenes System nicht unbedingt ein Vorteil. Für die tägliche Benutzung ist sicher ein gut dokumentierter, möglichst vielseitiger Satz von Programmierbefehlen hilfreicher.

In DERIVE - das ebenfalls in µ-LISP [Rich/Stoutemyer, 1994] entwickelt wurde und wird - steht ein solcher Satz leistungsfähiger Kommandos zur Verfügung. Algorithmen müssen in DERIVE funktional dargestellt werden, d.h., Programme werden dadurch geschrieben, daß Funktionen ineinandergeschachtelt werden. Dies erfordert sicher für jemanden, der es gewohnt ist, prozedural zu denken und zu programmieren, anfangs eine Umstellung. Die funktionale Programmierweise ist aber sehr effizient und letztendlich liegt darin auch das Geheimnis der Brauchbarkeit des Gesamtsystems DERIVE bei gleichzeitig minimalem Platten- und Hauptspeicherbedarf.

Die gängigen CAS stellen auch durchwegs Umgebungen zum Implementieren von Algorithmen, wie sie z.B. im Rahmen des Mathematikunterrichts auftreten, dar.

Das näherungsweise Lösen einer Gleichung wie z.B. $e^x . (x - 5) + 5 = 0$ läßt sich vom CAS elegant und mühelos durchführen (Bsp. 2.9), das Unbefriedigende daran ist aber, daß der Lösungsvorgang als Black Box abläuft und der Benutzer keinerlei Hinweis darauf erhält, wie das System zu der gewünschten Lösung gekommen ist. Im Unterricht kommt daher sehr bald der Wunsch einer Aufklärung auf.

Es wird zwar nicht möglich sein, lückenlos aufzuzeigen, was im CAS beim Lösen der Gleichung genau abläuft, aber diese Situation läßt sich im Unterricht konstruktiv zur Implementation eigener Algorithmen nützen.

Bemerkung: Was macht DERIVE nun tatsächlich bei der näherungsweisen Suche nach Nullstellen? "DERIVE uses the bisection method, terminating when a midpoint abscissa rounds to one of the endpoints at the current precision level. Thus, the accuracy is the maximum that you could expect, considering roundoff error in evaluating the expression.

The method often succeeds even when the expression values at the initial endpoints are both positive or both negative: In those respective cases, bisection converges toward a local minimum or maximum until either a midpoint value has the opposite sign or a local extremum is reached, where the expression value is compared with 0." [Rich/Stoutemyer, The International DERIVE Journal, 1/1994, p.14]

Beispiel 2.10: Algorithmus zur näherungsweisen Berechnung von Nullstellen

Entwickle einen Algorithmus zur näherungswise Bestimmung der Nullstelle einer Funktion durch binäres Suchen. Basis für diesen Algorithmus sei folgender Satz:

Ist f auf [a,b] stetig und gilt f(a) < 0 ⋀ f(b) >0 oder f(a) >0 ⋀ f(b) < 0, so besitzt f mindestens eine Nullstelle in]a,b[.

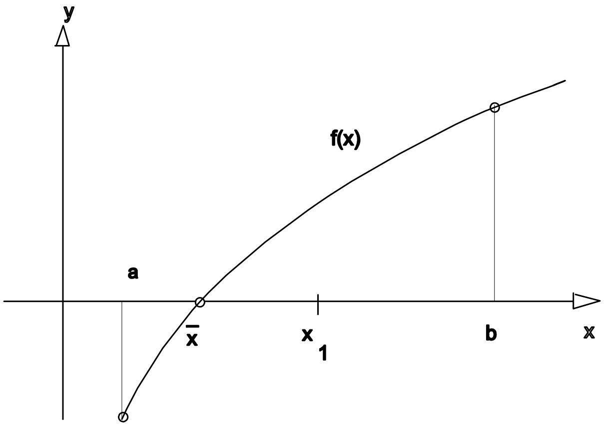

Abb. 2.8:Bisektionsverfahren

Versuchen wir nun daraus einen Algorithmus zu bilden:

Algorithmus in Worten	**Algorithmus formalisiert**
a) Bilde den Mittelwert der Intervallgrenzen.	a) `Intervall  := [a,b]` `Mittel(Intervall):= (a+b)/2`

b) Wähle jenes Teilintervall, bei dem die Funktionswerte der Grenzen verschiedene Vorzeichen haben.	b) `Wenn (f(a)*f((a+b)/2)< 0,`
	`dann Intervall_neu:=[a,(a+b)/2],`
	`sonst Intervall_neu:=[(a+b)2,b])`

c) Iteriere b)	c) `Wende das Ergebnis von b)` `wieder auf b) an`

Algorithmus im CAS

```
#1:  "Binäres Suchen"                                   User
                  x
#2:  F(x) := ê ·(x - 5) + 5                             User
```

a) Berechnung der Intervallmitte: Dazu wird zuerst allgemein ein Intervall I und damit dann eine CA-Funktion für die Berechnung des arithmetischen Mittels der Intervallgrenzen definiert.

```
#3:   i := [a, b]                                                 User
                      i  + i
                       1    2
#4:   MITTEL(i) := ─────────                                      User
                         2
```

Ein kurzer Test dieser CA-Funktion:

```
#5:   MITTEL([4, 5]) = 4.5                          User=Simp(User)
```

b) Ermittlung des nächsten Intervalls: Wir übertragen dazu die oben entwickelte Wenn-Dann-Sonst - Beziehung direkt in die Sprache des CAS:

```
User
      NEXT_INT(i) := IF(F(i )·F(MITTEL(i)) < 0 ,
#6:                        1
        [ i , MITTEL(i)] , [ MITTEL(i), i  ] )
           1                             2

#7:   NEXT_INT([4, 5]) = [4.5, 5]                   User=Simp(User)
```

c) Der letzte Schritt besteht nun in der Iteration des Vorgangs:

```
#8:  ITERATES(NEXT_INT(i), i, [4, 5], 17)                User

      ┌    4        5     ┐
      │   4.5       5     │
      │  4.75       5     │
      │  4.875      5     │
      │ 4.9375      5     │
      │ 4.9375   4.96875  │
      │ 4.95312  4.96875  │
      │ 4.96093  4.96875  │
#9:   │ 4.96484  4.96875  │                       Approx(#8)
      │ 4.96484  4.96679  │
      │ 4.96484  4.96582  │
      │ 4.96484  4.96533  │
      │ 4.96508  4.96533  │
      │ 4.96508  4.96520  │
      │ 4.96508  4.96514  │
      │ 4.96508  4.96511  │
      │ 4.96510  4.96511  │
      └ 4.96511  4.96511  ┘
```

Nach 17 Iterationsschritten haben wir somit die 2. Nullstelle der Funktion *f*(*x*) auf 5 Nachkommastellen genau bestimmt.

Abschließend können wir noch alle drei Schritte, aus denen unser Algorithmus besteht, in eine einzige CA-Funktion packen und damit diesen Algorithmus zu einem neuen CA-Kommando machen, das uns künftig zur Verfügung steht. (Ein Hinweis: Wird an Stelle von **ITERATES** der Befehl **ITERATE** verwendet, so erhält man lediglich das Ergebnis des letzten Iterationsschritts.)

```
#10: BIN_SUCH(startintervall, schrittzahl):=

     ITERATE(NEXT_INT(i),i,startintervall,schrittzahl)   User

#11: BIN_SUCH([4,5],17) = [4.96511,4.96511]   User=Simp(User)
```

Weitere Anregungen zum algorithmischen Vorgehen finden sich in Kapitel 4.3 - Modulprinzip.

2.4. Methodisches Hilfsmittel

Das didaktisch interessanteste Einsatzgebiet von CAS liegt sicher in ihrem Einsatz als methodische Hilfsmittel. "Über diese Rolle als Rechen- und Zeichenknecht hinaus können Computer [...] als methodische Hilfsmittel neuer Art (neben den bewährten) eingesetzt werden, welche zur Förderung gerade anspruchsvollerer Ziele beitragen und die Aneignung mathematischer Ziele für Schüler erleichtern sollen", schreibt W.Blum [Postel/Kirsch/Blum,1991, S.75]. Und als Beispiele führt er u.a. an: "Vertrautwerden mit Funktionen und deren Eigenschaften durch zielgerichtetes und systematisches Verändern von Parametern im Term der behandelten Funktionen" oder auch die "vielfältigen Möglichkeiten für Simulationen, z.B. bei exponentiellen Prozessen oder bei ökologischen Systemen, wodurch sowohl die betreffenden Realsituationen besser verstehen als auch allgemeine Einsichten in Modellbildungsprozesse gewinnen können" [a.a.O,S.76].

2.4.1. Hilfe bei der Modellbildung

Modellbildung ist ein zentrales Thema jedes anwendungsorientierten Mathematikunterrichts (siehe Kap. 3.4). Dem Schüler sollen neben einer geeigneten Arbeitsumgebung Grundmodelle zur Verfügung stehen.

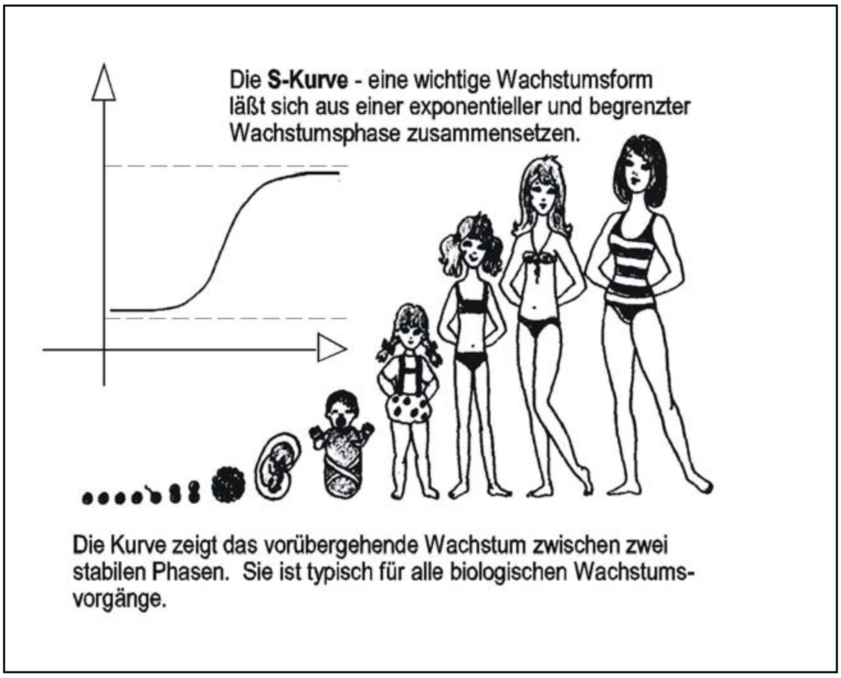

Abb. 2.9: Logistisches Wachstum

Visualisierung

CAS stellen eine ideale Umgebung zur Untersuchung grundlegender Wachstumstypen dar, die als Bausteine zur Modellbildung verstanden werden können. Dabei soll speziell auf die Möglichkeit der Visualisierung und Untersuchungen mit Parametervariationen hingewiesen werden. Als wesentliche Grundmodelle sind das lineare (additive) Wachstum, das freie Wachstum (exponentielles Wachstum/Zerfall), das beschränkte oder begrenzte Wachstum, das logistische und das hyperbolische (explosives oder überexponentielles) Wachstum zu nennen.

Untersuchung von Wachstumsprozessen - exponentielles Wachstum

Beispiel 2.11: Freies Wachstum - Variation der Basis

Untersuche die Auswirkungen der Veränderung der Basis a auf die Lage des Graphen der Exponentialfunktion $y = c\, a^x$! Wähle jeweils c = 1 und a als eine positive relle Zahl! Welche besonderen Eigenschaften können aus dieser Untersuchung hergeleitet werden?

Zuerst wird die Gleichung der zu untersuchenden Funktionsklasse allgemein angegeben.

```
                x
#1:  y = c·a                                              User
```

Wir setzen den konstanten Multiplikationsfaktor *c* gleich 1.

```
              x
#2:  y = a                                                User
```

Aus dieser Funktionsgleichung in #2 werden durch Einsetzen von Zahlen für die Basis *a* mit **Manage Substitute** einzelne Funktionen erzeugt. Wir verwenden als jeweilige Basis die Zahlen 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, und 2 .

Wir wählen an dieser Stelle bewußt einen 'langsamen' Weg mit **Manage Substitute**, weil der Schüler dadurch auf die einzelnen Funktionen leicht zugreifen kann (Diese Funktionen ließen sich auch mit einem VECTOR-Befehl erzeugen).

```
                x
#3:  y = 0.2                                            Sub(#2)
                x
#4:  y = 0.4                                            Sub(#2)
                x
#5:  y = 0.6                                            Sub(#2)
                x
#6:  y = 0.8                                            Sub(#2)
              x
#7:  y = 1                                              Sub(#2)
                x
#8:  y = 1.2                                            Sub(#2)
                x
#9:  y = 1.4                                            Sub(#2)
                x
#10: y = 1.6                                            Sub(#2)
                x
#11: y = 1.8                                            Sub(#2)
              x
#12: y = 2                                              Sub(#2)
```

Die einzelnen Funktionen werden mit dem Cursor unterlegt und mit **Plot Overlay Plot** gezeichnet.

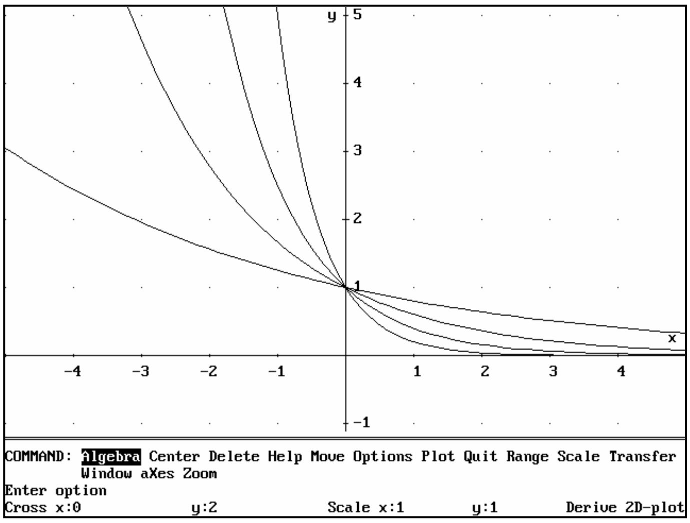

Abb. 2.10: Basis: $0 < a < 1$

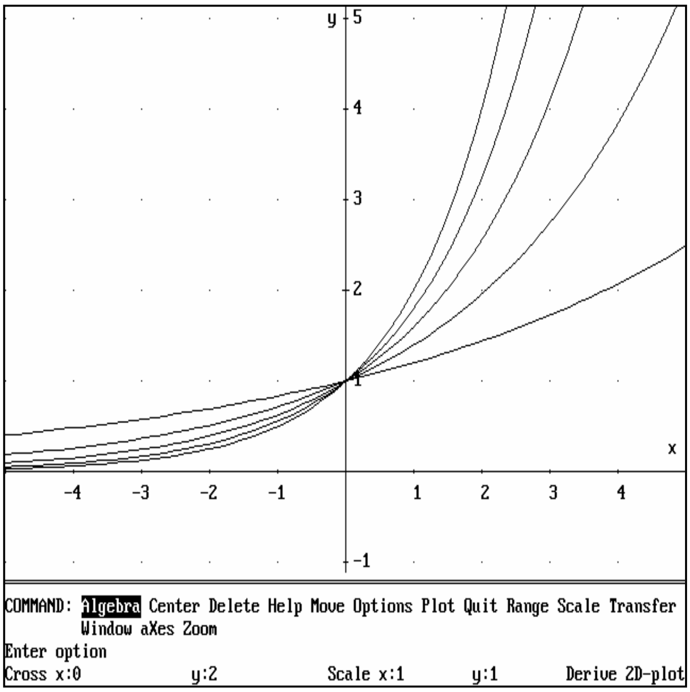

Abb. 2.11: Basis: $a > 1$

Es zeigt sich, daß mit $a = 1$ eine Gerade parallel zur x-Achse mit der Gleichung $y = 1$ festgelegt ist. Die Funktionen aus #3 bis #6, also wenn $0 < a < 1$, sind streng monoton fallend, die Grundmenge kann alle rellen Zahlen beinhalten und alle Funktionswerte sind positiv (Abb. 2.11). Die Funktionen dieses Typs sind Beschreibungen von Zerfallsprozessen ("Sterbeprozesse").

Bei gleichbleibender Grund- und Zielmenge zeigen die Funktionen aus #8 bis # 12 mit $a > 1$ streng monoton steigende Graphen (Abb. 2.12). Diese Funktionen stellen unbegrenzte Wachstumsvorgänge ("Geburtsprozesse") dar.

Es ist zu erkennen, daß alle Graphen der Funktionen durch den Punkt (0,1) gehen, da $c = 1$ gewählt wurde. Darüber hinaus läßt sich durch die Lage der Graphen die Vermutung aufstellen, daß sich zu jeder fallenden Exponentialfunktion ($0 < a < 1$) eine dazugehörende steigende ($a > 1$) finden läßt, so daß die beiden Funktionen durch Spiegelung an der y-Achse auseinander hervorgehen. Die Funktionen können 'nachgeplottet' (sie werden in einer neuen Farbe gezeichnet) und gespiegelte Funktionen herausgefunden werden, z.B. #6 und #8.

Beispiel 2.12: Freies Wachstum - Variation des Anfangswerts

Untersuche die Auswirkungen der Veränderung des Faktors c auf die Lage des Graphen der Exponentialfunktion $y = c\,a^x$! Wähle für a zuerst 1.2 und danach 0.8!

Welche besonderen Eigenschaften können aus dieser Untersuchung hergeleitet werden?

```
#1:  y = c·a^x                                  User
```

Die Funktionsgleichungen können durch Einsetzen (**Manage Substitute**) wie in dem vorherigen Beispiel erzeugt werden. Eleganter und schneller läßt sich eine Schar von Funktionen mit Hilfe des VECTOR-Befehls #2 (für *c* werden Werte zwischen -10 und 10 mit Schrittweite 1 eingesetzt) erzeugen.

```
#2:  VECTOR([y = c·1.2^x], c, -10, 10)          User
```

Die Vereinfachung dieses Ausdrucks in #2 liefert 21 Funktionen, die gemeinsam gezeichnet werden können:

```
#3:  [[y = - 10·ê^(0.182321·x)], ...]           Approx(#2)
```

Die Abbildung 2.13 zeigt die Graphen aus #3.

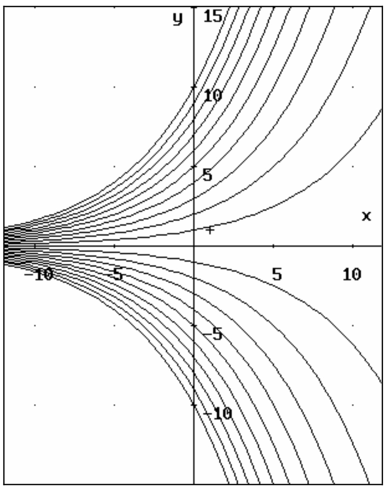

Abb. 2.12: Variieren von c bei a = 1.2

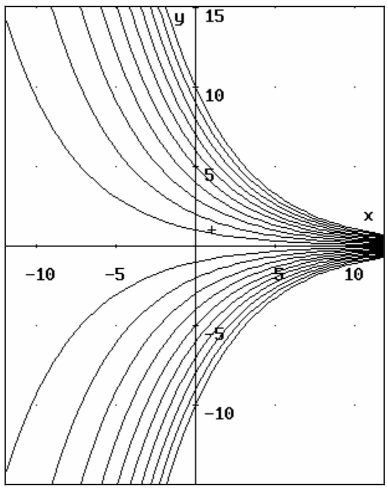

Abb. 2.13: Variieren von c bei a=0.8

Die Übersetzung der Basis 1.2 in eine e-Potenz läßt sich leicht nachvollziehen.

```
#4:  1.2 = ê^λ              User
#5:  λ = - LN[5/6]          Solve(#4)
#6:  λ = 0.182321           Approx(#5)
```

Ersetzt man den Parameter *c* durch dieselben Werte und wählt die Basis 0.8, ensteht der VECTOR-Befehl #7.

```
#7:   VECTOR([y = c·0.8^x], c, -10, 10)                    User
```

Die Funktionen werden mit **approX** #7 erzeugt und gezeichnet (Abb. 2.14).

Der Faktor *c* gibt die y-Koordinate für $x = 0$ an, also einen 'Startwert' y_0. Des weiteren zeigen sich bei negativem *c* die an der x-Achse gespiegelten Funktionen zu den Wachstums- oder Zerfallsprozessen mit positivem *c*.

Wir können also die Funktionen $y = y_0\, a^t$ durch die Gleichung $y = y_0\, e^{\lambda t}$ darstellen, wobei $a = e^{\lambda}$ und $\lambda = \ln(a)$ ist. Ist $a > 1$, sprechen wir von unbegrenztem exponentiellen (freien) Wachstum.

Wachstum mit Beschränkung

Beispiel 2.13: Untersuchung exponentieller Annäherung an einen Gleichgewichtswert

Simuliere die Auswirkungen der Veränderungen eines Ausgangswerts auf den Graphen eines begrenzten exponentiellen Wachstums an einen vorgegebenen Sättigungswert G = 100. [vgl. Timischl, 1988, S.65]

Wir arbeiten mit folgenden Einstellungen:

```
#1:   [InputMode := Word, CASeMode := Sensitive]          User
```

Die allgemeine Beziehung eines beschränkten exponentiellen Wachstums wird in #2 angegeben. Es wird *x* als die vergangene Zeit und *y* als jene Variable angenommen, die den Zustand eines Systems beschreibt (z.B. Temperatur eines Körpers, Höhe eines Baums oder die Konzentration eines gelösten Stoffs in einer Zelle).

Dazu machen wir mittels Exponentialfunktion die Differenz zum Sättigungswert *G* immer kleiner. Die Variable y_0 wird für den Ausgangswert zum Zeitpunkt 0 verwendet.

```
#2: y = G + (y0 - G)·ê^(- λ·x)                          User
```

Mit **Manage Substitute** wird der Sättigungswert mit 100 und der Anfangswert mit 5 festgelegt. Wir wählen für λ eine positive reelle Zahl, bei dieser Untersuchung den Wert 1.

```
#3:   y = 100 + (5 - 100)·ê^(- 1·x)                     Sub(#2)
```

Die Vereinfachung des Terms liefert folgende Funktion:

```
#4:   y = 100 - 95·ê^(-x)                               Simp(#3')
```

Weitere Anfangswerte werden eingegeben ($y_0 = 50$, $y_0 = 150$ und $y_0 = 200$) und die Terme vereinfacht.

Es werden also Startwerte y_0 einerseits kleiner, andererseits größer als der Sättigungswert (die Grenze) gewählt.

```
#5:   y = 100 + (50 - 100)·ê^(- 1·x)                    Sub(#2)
#6:   y = 100 - 50·ê^(-x)                               Simp(#5')
#7:   y = 100 + (150 - 100)·ê^(- 1·x)                   Sub(#2)
#8:   y = 50·ê^(-x) + 100                               Simp(#7')
#9:   y = 100 + (200 - 100)·ê^(- 1·x)                   Sub(#2)
#10:  y = 100·ê^(-x) + 100                              Simp(#9')
```

Darüber hinaus wird die Gerade, der sich diese Funktionen asymptotisch nähern (der angenäherte Gleichgewichtszustand) angegeben.

```
#11: y = 100                                                User
```

Die vier Funktionen, die ein begrenztes exponentielles Wachstum beschreiben, werden mit der Geraden gezeichnet (Abb. 2.15).

Es ist zu erkennen, daß bei einem Ausgangswert $y_0 < G$ eine streng monoton wachsende Funktion mit Grenzwert 100 entsteht. Bei $y_0 > G$ erfolgt die Annäherung an G durch streng monoton fallende Funktionen.

Wenn $y_0 = G$ ist, kommt es zu einem Gleichgewichtszustand, also zu einer Geraden $y = 100$. Weitere Untersuchungen mit Veränderung des Sättigungswerts, der Startwerte und anderen $\lambda > 0$ können vorgenommen werden.

Für die weiteren Überlegungen verwenden wir die definierte Funktion B.

$$B(x) := G + (y0 - G)\cdot \hat{e}^{-\lambda\cdot x}$$

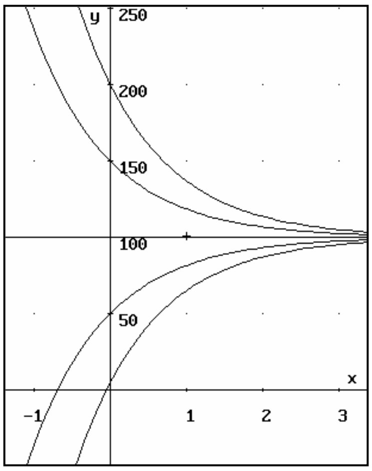

Abb. 2.14: Sättigungswert 100 und Ausgangswert variabel

Logistisches Wachstum

Beispiel 2.14: Die innere Struktur des logistischen Wachstums [vgl. Timischl 1988, S.67]

Stellt man das Wachstum eines Organismus oder einer Population in Abhängigkeit von der Zeit grafisch dar, so erhält man eine Wachstumskurve. Diese Wachstumskurven besitzen oft einen S-förmigen Verlauf.

a) Untersuche die logistische Funktion durch Ableitungen und gib Wendestellen an!

b) Läßt sich diese Funktion durch andere Wachstumsmodelle (unbegrenztes exponentielles Wachstun, begrenztes Wachstum, lineares Wachstum) näherungsweise beschreiben?

Zeigt das logistische Wachstum Eigenschaften, die bereits bei anderen Wachstumsprozessen untersucht wurden?

Voreinstellungen:

```
#1:  [InputMode := Word, CASeMode := Sensitive]          User
```

Wir wählen für diese Untersuchungen den Sättigungswert G mit 100 und definieren den Anfangswert der Wachstumsfunktion mit 5. Der Koeffizient λ der Basis e wird mit 1 festgelegt (vgl. Beispiel 2.17). Wir setzen also $k.G = \lambda = 1$, daraus folgt $k = 0.01$).

```
#2:  G := 100                                                User
#3:  y0 := 5                                                 User
```

Die allgemeine Beziehung wird als logistische Funktion *L* definiert. Im Unterricht könnte eine Herleitung des logistischen Wachstums über Differentialgleichungen erfolgen (siehe wieder Beisp. 2.17). An dieser Stelle arbeiten wir bereits mit dem fertigen Modell und versuchen dieses durch bereits bekannte Modelle zu verstehen.

#4: $$L(x) := \frac{G}{1 + \left[\frac{G}{y0} - 1\right]\cdot ê^{-1\cdot x}}$$ User

Der Klammerausdruck im Nenner ist ein Maß für die Entfernung vom Endzustand. Mit der Zeit nimmt der 'Wachstumsdrang' ab und nähert sich einem Gleichgewichtszustand. Vereinfachen wird diesen Klammerausdruck, so erhalten wir

#5: $$L(x) := \frac{G}{1 + 19\cdot ê^{-x}}$$ Simp(#4')

sowie bei Vereinfachung des ganzen Terms:

#6: $$L(x) := \frac{100\cdot ê^{x}}{ê^{x} + 19}$$ Simp(#5')

Die Formeln aus #5 und #6 erhält man durch einfachen Umformungen, die jederzeit auch händisch nachvollzogen werden können. In den einzelnen Lehr- und Sachbüchern werden weitere Formeln angegeben, die die logistische Funktion beschreiben. Alle lassen sich auf eine bereits angeführte Darstellung zurückführen.

Diese Funktion *L* wird bezüglich Monotonieverhalten und Krümmungsverhalten untersucht, die auftretende Wendestelle wird grafisch nährungsweise mit dem Cursor gesucht (Abb. 2.16) und dann berechnet.

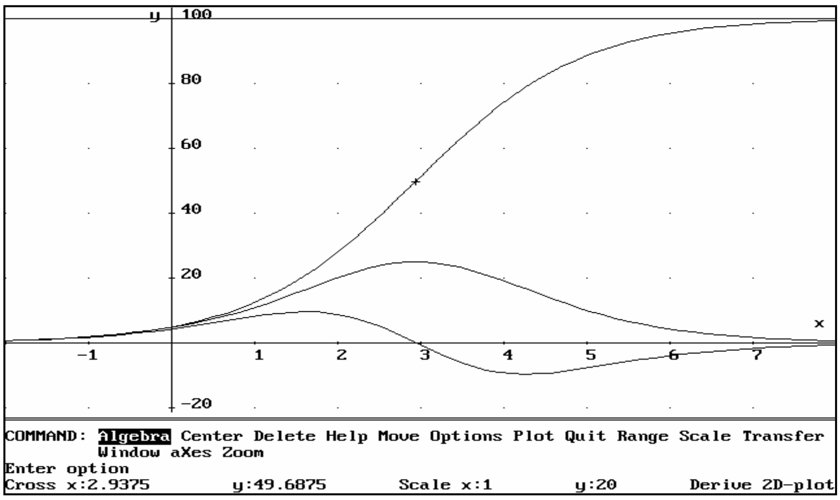

Abb. 2.15: Funktion und die Ableitungen

Die Ableitungen lassen sich direkt mit der Funktion **DIF** berechnen und mit Anfügen des Gleichheitszeichens auswerten.

$$\#7: \quad \left[\frac{d}{dx}\right]^{1} L(x) = \frac{1900 \cdot \hat{e}^{x}}{(\hat{e}^{x} + 19)^{2}} \qquad \text{User=Simp(User)}$$

$$\#8: \quad \left[\frac{d}{dx}\right]^{2} L(x) = \frac{1900 \cdot \hat{e}^{x} \cdot (19 - \hat{e}^{x})}{(\hat{e}^{x} + 19)^{3}} \qquad \text{User=Simp(User)}$$

Aus der Grafik wird das Intervall, in dem sich die Nullstelle der zweiten Ableitung befindet, mit [2,4] festgelegt und im *Approximate*-Mode **(Option Precision)** die Nullstelle berechnet.

```
#9:  Precision := Approximate                                  User
```

$$\#10: \quad \left[\frac{d}{dx}\right]^{2} L(x) = 0 \qquad \text{User}$$

$$\#11: \quad \frac{1900 \cdot \hat{e}^{x} \cdot (19 - \hat{e}^{x})}{(\hat{e}^{x} + 19)^{3}} = 0 \qquad \text{Simp(\#10')}$$

```
#12: x = 2.94445                                        Solve(#10)
```

Die Lösung dieser Gleichung zeigt, daß im Zähler nur der Ausdruck $(19 - e^x)$ Null zu setzen ist (alle anderen Teile sind positiv), so daß man die Lösung $x = ln(19)$ erhält.

Dieser Sachverhalt kann im *Exact*-Modus nachgewiesen werden.

```
#13: Precision := Exact                                        User

#14: x = LN(19)                                         Solve(#10)
```

Die weiteren Lösungen lauten: ∞, $-\infty$ und $\infty + \hat{\imath}\,\pi$.

Der Funktionswert an der Wendestelle ist die Hälfte des Sättigungswerts.

```
#15: L(LN(19)) = 50                                User=Simp(User)
```

Nach Abschluß dieser Kurvendiskussion wenden wir uns Teil b) der Aufgabenstellung zu. Wir definieren eine unbegrenzte Wachstumsfunktion E mit dem Anfangswert 5 und dem Wachstumskoeffizienten $\lambda = 1$.

$$\#16: \quad E(x) := y0 \cdot \hat{e}^{x} \qquad \text{User}$$

$$\#17: \quad E(x) := 5 \cdot \hat{e}^{x} \qquad \text{Simp(\#16')}$$

Des weiteren definieren wir eine beschränkte Wachstumsfunktion B mit den vorgegebenen Werten für den Anfangswert und den Sättigungswert mit $\lambda = 1$.

-x

```
#18: B(x) := G + (y0 - G)·ê                                  User

                            -x
#19: B(x) := 100 - 95·ê                               Simp(#18')
```

Wenn man die logistische Funktion aus #6 und die beiden Funktionen *E* und *B* in eine Grafik einzeichnet, erkennt man, daß zu Beginn die Funktion *E* eine sehr gute Näherung des Anfangsverhaltens der logistischen Funktion darstellt und die Funktion *B* ziemlich gut das Näherungsverhalten an den Sättigungswert angibt, wenn man die Funktion *B* um das Argument des Wendepunkts nach rechts verschiebt (Abb. 2.17).

```
                            - (x - LN(19))
#20: B(x) := 100 - 95·ê                                      User
```

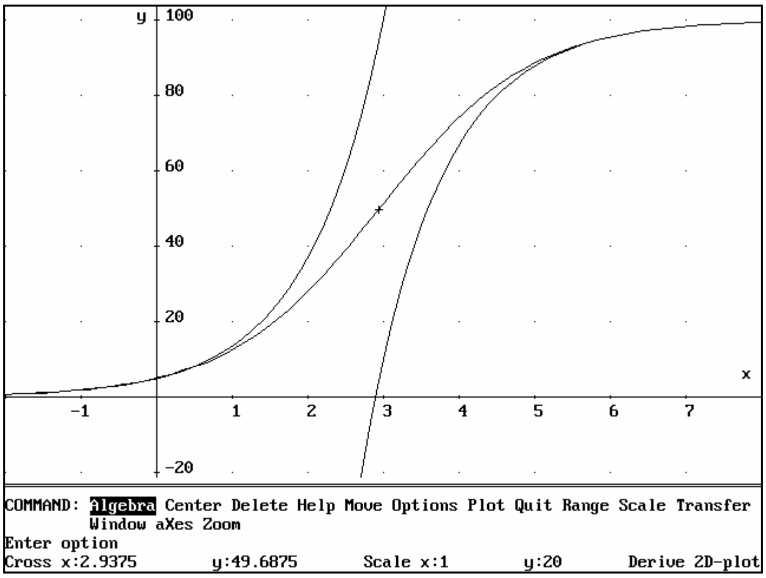

Abb. 2.16: Erster Versuch einer Approximation

Wir schließen daraus, daß sich zu Beginn die logistische Funktion *L* 'exponentiell' verhält, dann eine 'lineares' Verhalten zeigt und danach 'beschränkt exponentiell' verläuft.

Das kann auch folgendermaßen beschrieben werden: zuerst ein 'embryonales', rasant anwachsendes Stadium, dann eine 'lineare' Entwicklungsphase und später eine Endphase mit verringertem Wachstum bis zum 'Stillstand'.

Wir können nun versuchen, diesen Vorgang durch eine zusammengesetzte Funktion darzustellen. Dazu müssen wir geeignete Stellen verwenden, so daß aus der Grafik ersichtlich ist, daß bis dorthin die Funktionen als gute Näherung verwendet werden können. Wir wählen für die unbegrenzte Wachstumsfunktion *E* den Bereich $0 \le x \le 1$ und für die begrenzte Wachstumsfunktion *B* Argumente ab $2\ \ln(19) - 1 \le x$. An diesen Stellen ist die Differenz der Funktionswerte der jeweiligen Näherungsfunktion und der logistischen Funktion gering. Wir haben also Intervalle für eine geeignete Approximation gefunden. Als 'Klebstoff' verwenden wir die IF-Funktion und legen den linearen Anteil vorübergehend mit dem Funktionswert der Wendestelle fest.

#21: $\mathrm{LKLEB}(x):=\mathrm{IF}(x\le 1,E(x),\mathrm{IF}(2\cdot\mathrm{LN}(19)-1\le x,B(x),50))$ User

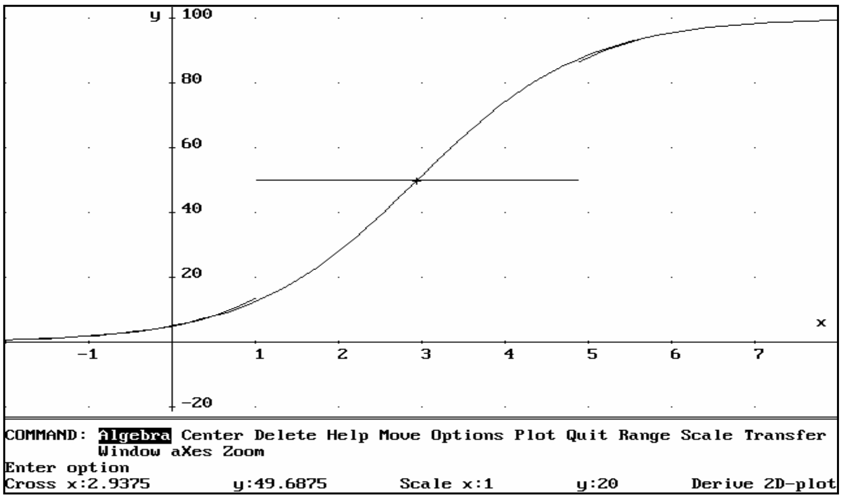

Abb. 2.17: Zweiter Versuch der Approximation

Diese stückweise zusammengesetzte Funktion wird in Abb. 2.18 über die logistische Funktion gezeichnet.

Dann 'basteln' wir den linearen Teil als 'Fortsetzung' durch das Erstellen einer Geraden, die die beiden Grenzstellen mit den Funktionen *E* und *B* verbindet. Wir lösen ein Gleichungssystem mit 2 Gleichungen und 2 Unbekannten.

#22: $E(1) = 5\cdot ê$ User=Simp(User)

#23: $B(2\cdot\mathrm{LN}(19) - 1) = 100 - 5\cdot ê$ User=Simp(User)

#24: $y = k\cdot x + d$ User

#25: $5\cdot ê = k\cdot 1 + d$ Sub(#24)

#26: $100 - 5\cdot ê = k\cdot(2\cdot\mathrm{LN}(19) - 1) + d$ Sub(#24)

#27: $[5\cdot ê = k\cdot 1 + d,\ 100 - 5\cdot ê = k\cdot(2\cdot\mathrm{LN}(19) - 1) + d]$ User

Die Lösung dieses Gleichungssystems liefert die Steigung *k* der Geraden und die y-Koordinate des Schnittpunkts mit der y-Achse *d*.

#28: $\left[d = \dfrac{5\cdot(ê\cdot\mathrm{LN}(19) - 10)}{\mathrm{LN}(19) - 1},\ k = \dfrac{5\cdot(10 - ê)}{\mathrm{LN}(19) - 1}\right]$ Solve(#27)

Wir definieren *k* und *d* und die lineare Funktion *G:*

#29: $k := \dfrac{5\cdot(10 - ê)}{\mathrm{LN}(19) - 1}$ User

#30: $d := \dfrac{5\cdot(ê\cdot\mathrm{LN}(19) - 10)}{\mathrm{LN}(19) - 1}$ User

#31: $G(x) := k\cdot x + d$ User

```
                    5·(ê·LN(19) + x·(ê - 10) - 10)
#32: G(x) := ────────────────────────────────────────        Simp(User')
                           1 - LN(19)
```

Diese sehr eigenwillige Darstellung der für unser Problem benötigten Verbindungsgeraden wird als 'Klebeband' verarbeitet. Die Funktion *L* wird damit gut approximiert.

```
#33: LKLEB(x):=IF(x≤1,E(x),IF(2·LN(19)-1≤x,B(x),G(x)))    User
```

Abschließend werden die einzelnen Phasen mit Zuordnungslinien gekennzeichnet, so daß der Verlauf und die stückweise definierte Funktion besser ersichtlich werden. Wir verwenden eine Funktion ZOL für das Zeichnen von Zuordnungslinien.

```
                 ┌ x  0 ┐
#34: ZOL(x, f) := │ x  f │                                     User
                 └ 0  f ┘
```

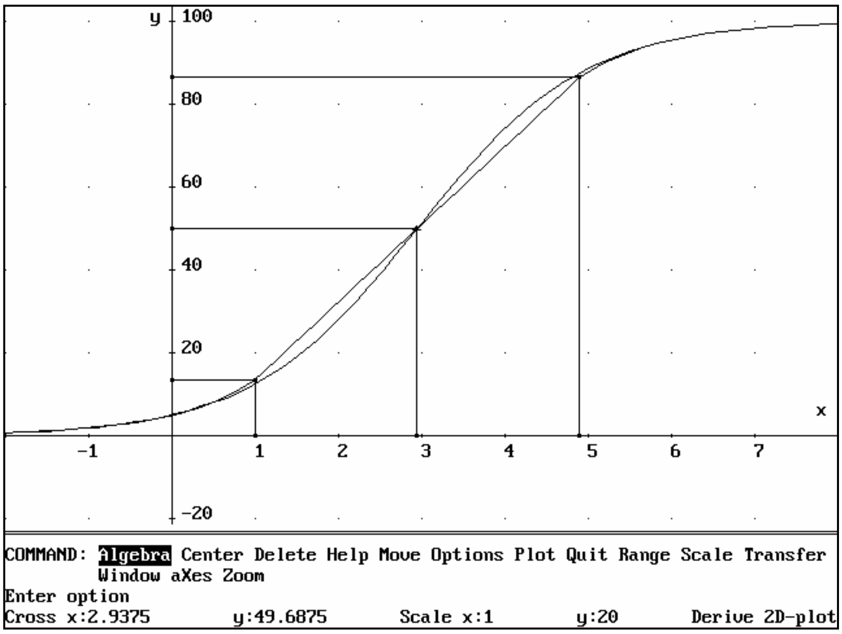

Abb. 2.18: Dritter Versuch: Nur geklebt und hält doch!

Die Auswertung dieser Funktion ZOL

```
(ZOL(1, E(1)), ZOL(LN(19), L(LN(19))),
 ZOL(2·LN(19)-1, B(2·LN(19)-1))
```

liefert die eingezeichneten Abszissen und Ordinaten an den Stellen, an denen sich das schon beschriebene typische Verhalten ändert (Abb. 2.19).

Simulation

Das CAS ist - wie wir in Beispiel 2.14 gesehen haben - auch ein Werkzeug zur Untersuchung von Wachstumsvorgängen und vernetzten Systemen. Gelingt es, die Änderung eines Zustands von einem Zeitschritt zum nächsten zu beschreiben, so lassen sich diese Zustandsänderungen iterieren, und wir können aus der lokalen Beschreibung eine globale Sicht gewinnen. Wir erhalten Einblick in das Langzeitverhalten eines Systems und können durch die Simulation einen Blick auf ein mögliches Entwicklungsszenario werfen.

Beispiel 2.17: Ein gefährdetes Gleichgewicht? [Nach Sziruscek, 1990, S.181]

Ein großer Teich bietet rund 400 Fischen Platz. Wie lange wird es dauern, bis der Teich bis zur Grenze seiner Aufnahmefähigkeit mit Fischen besetzt ist, wenn zur Zeit 20 Fische vorhanden sind? Annahmen: Das Wachstum der Fische sei proportional zum bereits vorhandenen Fischbestand und proportional zum noch vorhandenen Freiraum. Der monatliche Wachstumsfaktor (Proportionalitätsfaktor) sei k_1=0,001.

Es soll eine Simulation des Wachstums des Fischbestands erstellt werden. Wie würde sich ein geringfügig stärkeres Wachstum auswirken?

Die Entwicklung der Fischpopulation läßt sich mit Hilfe einer Differenzengleichung beschreiben:

$$\frac{y(t+\Delta t) - y(t)}{\Delta t} = k_{\Delta t} \cdot y(t) \cdot (G - y(t))$$

bzw.

$$y(t+\Delta t) = y(t) + k_{\Delta t} \cdot y(t) \cdot (G - y(t)) \cdot \Delta t$$

$y(t)$ sei hier der Fischbestand zur Zeit t, G die Wachstumsgrenze. Setzt man den Zeitraum eines Monats als eine Zeiteinheit an, also Δt=1, so ergibt sich für den monatlichen Zuwachs:

$$y(t+1) = y(t) + 0{,}001 \cdot y(t) \cdot (400 - y(t))$$

Allgemein läßt sich ein beliebiger diskreter Wachstumsvorgang mit einer Änderungsrate $r(t,y)$ somit folgendermaßen beschreiben:

$$y(t+\Delta t) = y(t) + r(\,t\,, y(t)\,) \cdot \Delta t$$

Iterieren wir nun diese Gleichung, so ergibt sich die Langzeitentwicklung der betrachteten Fischpopulation. Dazu definieren wir im ersten Schritt die Zuwachsrate des Fischbestands (die vom Bestand y und der Zeit t abhängt) als eigenständige CA-Funktion:

```
#1:  R(t, y) := 0.001·y·(400 - y)                         User
```

Nun können wir eine zweite universell einsetzbare CA-Funktion festlegen, die mit konkreten Werten belegt, eine Tabelle für den diskreten Wachstumsvorgang liefert und die sich zudem unmittelbar plotten läßt. Iteriert wird hier eine Liste, die aus zwei Komponenten besteht: Die erste ($t+\Delta t$) beschreibt, wie man von einem Zeitpunkt zum nächsten gelangt, die zweite beschreibt, wie der Bestand y (hier an Fischen) in der Zeitdauer Δt zunimmt. Durch das Kommando ITERATES werden die derart berechneten Werte stets wieder als Ausgangswerte für t bzw. y verwendet (*sz* gibt die Schrittzahl an).

```
User
   #2:  WACHSTUM(t0, y0, δt, sz) :=

        ITERATES([t + δt, y + R(t,y)·δt], [t, y], [t0, y0], sz)
```

Mit den Anfangswerten $t_0 = 0$ für die Zeit und $y_0 = 20$ für den Bestand ergibt sich bei einem Zeitschritt $\Delta t = 1$ für zehn Iterationsschritte die Tabelle (#4).

```
#3:  WACHSTUM(0, 20, 1, 10)                                User

      ┌  0     20    ┐
      │  1    27.6   │
      │  2  37.8782  │
      │  3  51.5947  │
      │  4  69.5706  │
#4:   │  5  92.5588  │                             Approx(#2)
      │  6  121.015  │
      │  7  154.776  │
      │  8  192.731  │
      │  9  232.678  │
      └ 10  271.610  ┘
```

Für die ersten 50 Monate ergibt sich damit folgende Simulation (siehe Abb. 2.20):

```
#5:   WACHSTUM(0, 20, 1, 50)                                User
```

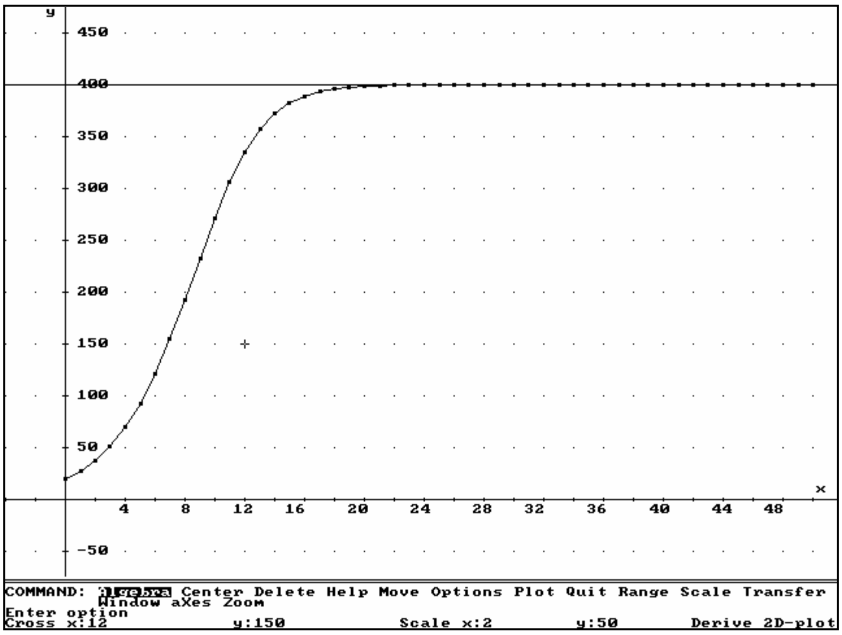

Abb.2.20: Diskretes logistisches Wachstum

Wenn wir das Wachstum etwas verstärken, indem wir herumexperimentieren und den Proportionalitätsfaktor des monatlichen Wachstums erhöhen, können wir das für das diskrete logistische Wachstum charakteristische Abdriften ins Chaos beobachten. Wir definieren dazu die Änderung des Fischbestands schrittweise neu und vereinfachen dazwischen stets Ausdruck #5, um eine plotbare Tabelle zu erhalten:

```
#6:   R(t, y) := 0.00375·y·(400 - y)                        User

#7:   R(t, y) := 0.005·y·(400 - y)                          User

#8:   R(t, y) := 0.00625·y·(400 - y)                        User

#9:   R(t, y) := 0.0075·y·(400 - y)                         User
```

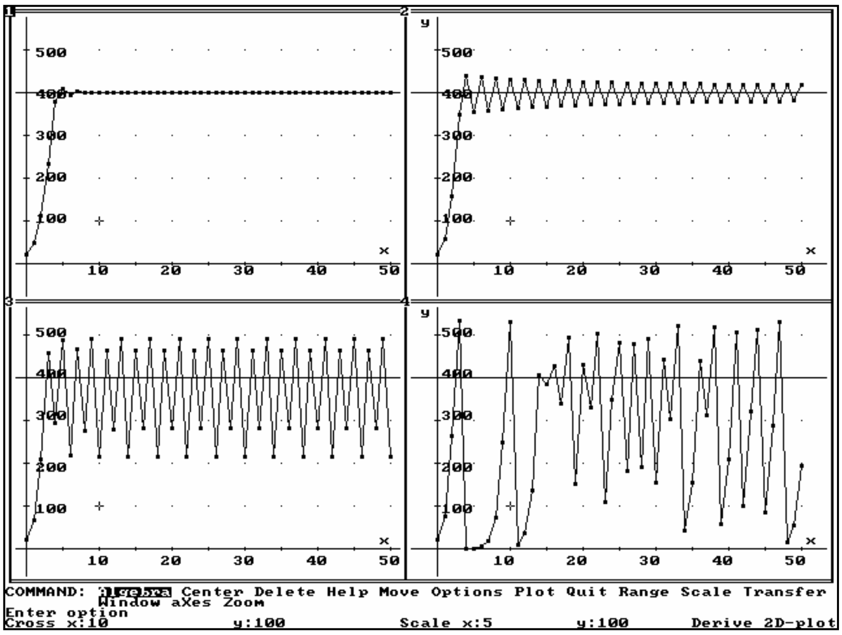

Abb. 2.19:Diskretes logistisches Wachstum und Abgleiten ins Chaos

Tritt dieser Effekt auch bei der Beschreibung dieser Wachstumssituation mit Hilfe des kontinuierlichen Modells auf? Verwenden wir zur Klärung dieser Frage die oben gewonnene Funktion für das logististische Wachstum:

$$y(t) = \frac{y_0 \cdot G}{y_0 + (G - y_0) \cdot e^{-k \cdot G \cdot t}}$$

Wenn wir nun der Reihe nach diskretes und kontinuierliches Wachstum gegenüberstellen, so können wir beobachten, daß beim Vergrößern der Wachstumsrate bei kontinuierlichem Wachstum das "Abdriften" ins Chaos nicht eintritt (Abb. 2.22).

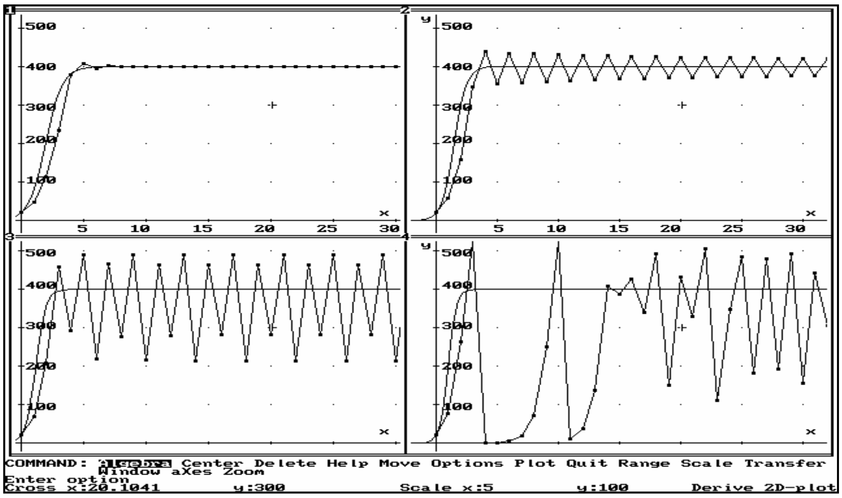

Abb. 2.20:Diskretes und kontinuierliches logistisches Wachstum

Zusammenfassend sehen wir, daß die diskrete Modellierung zu völlig anderen Ergebnissen führen kann als die kontinuierliche. Beide Arten der Modellierung sind nicht einfach zwei alternative Zugänge, sondern es gibt offenbar einen qualitativen Unterschied zwischen diskret und kontinuierlich.

2.4.2. Das CAS als Hilfsmittel beim Begriffsbildungsprozeß

Wesentlich erscheint nun der Prozeß, der zum Begriff führt, den Begriff verändert und exaktifiziert, zu sein, nicht aber das fertige Produkt 'Begriff', von dem R. Fischer sagt: "Bei fast keinem grundlegenden mathematischen Begriff gelingt es, ihn durch eine exakte Definition ganz in den Griff zu bekommen." [Fischer, 1985, S. 145]

In diesem Buch werden mehrere Beispiele angeboten, die zeigen, welche Rolle das CAS beim Begriffsbildungsprozeß spielen kann, wie etwa beim Begriff der irrationalen Zahl (Kap. 2.1.2). Eine Begriffsdeutung kann durch den Computer sehr gut vorbereitet werden.

Das folgende Beispiel soll eine Möglichkeit zeigen, mit dem CAS die Begriffe 'Linearisierung'und 'Differenzierbarkeit' zu entwickeln:

Beispiel 2.18: Differenzierbarkeit und Linearisierung

Die fundamentale Idee der Linearisierung, der Begriff der Differnzierbarkeit.

Versucht man diese zentralen Begriffe der Analysis gleich symbolisch und abstrakt an die Schüler heranzubringen, wird man viele nicht erreichen. Für einen ersten Zugang ist die Visualisierung unbedingt notwendig.

Zeichnet man den Graphen einer Polynomfunktion *V*(*x*), geht im *TRACE*-Modus mit dem Cursor an irgendeine (möglichst gekrümmte) Stelle und vergrößert immer weiter, so erscheint die Funktion im Grafikfenster bei ausreichender Vergrößerung in der Umgebung des Kurvenpunkts als Gerade (siehe Abb. 2.22). Ziel dieser experimentellen Phase ist die Idee, daß eine (differenzierbare) Funktion in einer kleinen Umgebung eines Punkts durch eine Gerade ersetzt werden kann. In einer exaktifizierenden Phase gilt es dann, diese Gerade auch zu finden.

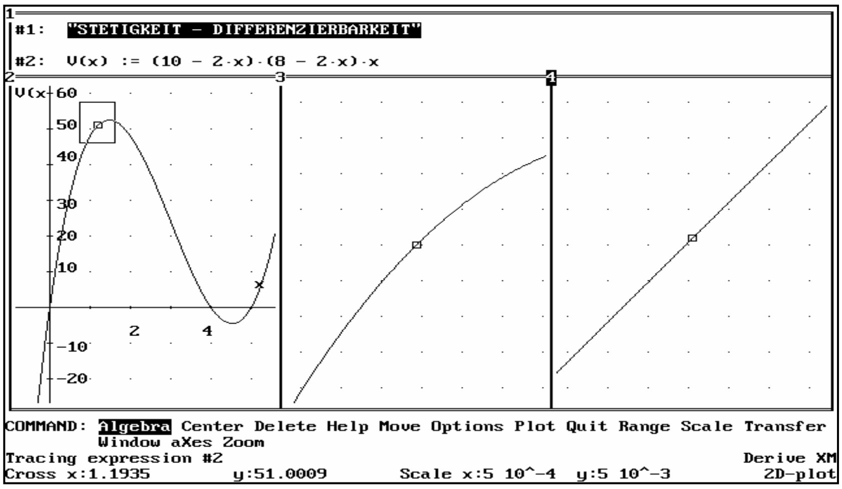

Abb. 2.21: Linearisierung

Die Skalierungen in den einzelnen Fenstern:

Fenster (2): x:=1;y:=10. Fenster (3): x:=0.05;y:=0.5. Fenster (4): x:=0.0005;y:=0.005.

Daß diese Idee der Linearisierung nicht bei jeder beliebigen Funktion an jeder Stelle funktioniert, kann man an Hand einer Betragsfunktion untersuchen (siehe Abb. 2.23). Am besten nimmt man gleich $|V(x)|$ und wandert mit dem Cursor an eine 'heikle' Nullstelle. Bei noch so starker Vergrößerung wird hier keine Linearisierung zu beobachten sein. Die Betragsfunktion ist eben an dieser Stelle nicht differenzierbar.

Skalierungen in den Grafikfenstern der folgenden Abbildung:

Fenster (2):x:=1;y:=10. Fenster (3):x:=0.2;y:=2. Fenster (4):x:=0.00005;y:=0.0005

Die Vergrößerung erfolgte nach der Cursorwanderung im *TRACE*-Modus mit Hilfe des **ZOOM**-Befehls.

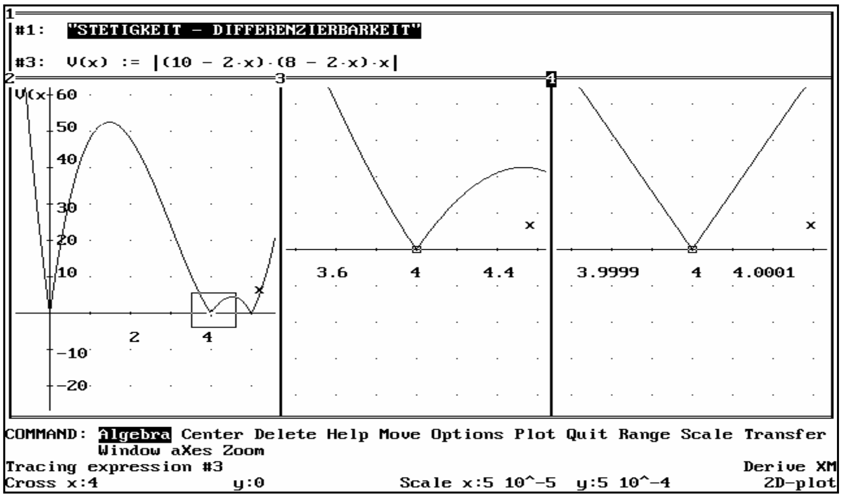

Abb. 2.22: Linearisierung nicht möglich!

Natürlich sollte vor allem in höher dotierten Schulformen damit der Begriffsbildungsprozeß nicht abgeschlossen sein. Aber mit diesem Vorverständnis müßte die Exaktifizierung durch den Lehrer leichter möglich sein.

2.5. Sprachliches Hilfsmittel

2.5.1. Hilfe beim Übersetzen von Umgangssprache in die formale Sprache der Mathematik

In einem CAS-unterstützten Unterricht gewinnen sicher Übungen zum Mathematisieren von Situationen und Übungen zum Modellbilden an Bedeutung, Mathematik wird wichtig als Sprache und Darstellungsmittel. Bisher war es üblich, Beziehungen möglichst rasch und vollständig zu formalisieren. Durch den Einsatz von CAS können auch Zwischenschritte auf diesem Weg von der Umgangssprache zum Formalismus didaktisch interessant werden.

Beispiel 2.19: Prozentrechnung. Nach [Mauve/Moos, 1994, S.20]

Umgangssprache ->	Wortformel ->	formale Darstellung
"Anteil =	Anteil =	
p Prozent vom Grundwert	p . Prozent_von . Grundwert	$A = \frac{p}{100} \cdot G$ 5

Hier wird sozusagen eine Zwischenstufe eingezogen, mit der mit DERIVE auch schon gearbeitet werden kann. Die Wortformel wird durch das CAS lebendig:

	`Prozent_von:= 1/100`	
" 15% von 250 ? "	`15 . Prozent_von . 250`	A= 250 . 0,15
		Simplify bzw. approX liefert `37.5`

Solche Wortformeln können eine wesentliche Hilfe beim Übersetzungsprozeß von Sprache in Formalismus darstellen. Genauso wichtig ist natürlich, stets darauf hinzuweisen, warum es günstig ist, die Mühen dieses Übersetzungsvorgangs auf sich zu nehmen (das Problem wird durch den Vorrat an mathematischen Techniken und Methoden besser bearbeitbar bzw. lösbar, ökonomischer Aspekt der Mathematik). Natürlich sollte dieses Ziel - eine möglichst effiziente Berechnung (rechte Spalte im Beispiel) - nicht aus dem Auge verloren werden. Insofern kann die Wortformel aber auch nur eine didaktisches Hilfsmittel, ein Übergangszustand sein.

Beispiel 2.20: Prozentrechnen 'leicht gemacht'. Nach einer Idee von [Josef Böhm, 1995]

Ein Lehrer zahlt monatlich 21% seines Gehalts für seine Wohnung. Nun erhöhte sich sein Monatsgehalt um 1900 Schilling, so daß die Miete nur noch 19% des (neuen) Gehalts ausmacht. Berechne, wie hoch das ursprüngliche Gehalt war! Wie hoch ist die Miete?

```
                         1
#1:  prozent_von :=  ─────                          User
                        100
User
#2:  21·prozent_von·gehalt = 19·prozent_von·(gehalt + 1900)

#3:  gehalt = 18050                            Solve(#2)

#4:  miete = 18·prozent_von·18050                   User

#5:  miete = 3249                               Simp(#4)
```

Solche Wortformeln ('Wortmodelle') können neben Zinsaufgaben für die verschiedensten Typen von Textaufgaben (= 'verkürzte Modellbildungsprozesse') erstellt werden. Wesentlich ist, daß der Schüler erstens mit der betrachteten Situation etwas anfangen kann. Dies ist oft bei Bewegungs-, Mischungs- oder Leistungsaufgaben deswegen nicht der Fall, weil dem Schüler z.B. der Leistungsbegriff selbst nicht klar ist. Zweitens soll der Schüler mit der betrachteten Situation umgehen können. Hier sind die verwendeten Wortvariablen eventuell hilfreich, weil man dem umgangssprachlichen Text noch näher ist, aber (mit DERIVE) auch schon richtig damit rechnen kann.

Beim Übersetzungsvorgang darf man aber auch nicht Beispiele wie "auf 1 Professor kommen 6 Studenten" (Rosnick-Clement-Phänomen, nach [Fischer/Malle, 1985]) außer acht lassen, wo der Weg von der Sprache zur Formel leicht auf das falsche Gleis (nämlich $1P = 6S$!) führt. Hier können verschiedene andere Formulierungen in der Umgangssprache (z.B. etwa "das sechsfache der Anzahl der Professoren ist gleich der Anzahl der Studenten") bzw. Plausiblitätsüberlegungen weiterhelfen.

2.5.2. Bereitstellung neuer Sprachelemente

Die Sprache der Mathematik besteht aus vielen verschiedenen Konstrukten: Zahlen, Operatoren, Termen, Funktionen, Gleichungen und Ungleichungen, Vektoren, Matrizen, logischen Ausdrücken und Algorithmen, um einige bekannte zu nennen. All diese Sprachelemente helfen uns, Probleme zu beschreiben, bei denen Mathematik irgendwie nützlich sein kann. Ist es einmal gelungen, ein Problem zu mathematisieren, so steht ein ungeheurer Vorrat an Fertigkeiten, Techniken, Methoden und Kalkülen bereit, der uns in der Problembearbeitung weiterhelfen und uns einer Lösung näherbringen kann. CAS können in diesem Zusammenhang als kompakte und interaktive Sammlungen solchen Vorratswissens betrachtet werden. Und sie stellen auch eine neue Art von Objekten zur mathematischen Beschreibung von Sachverhalten bereit, die in der Literatur noch keinen einheitlichen Namen gefunden haben, oft werden sie nur als CA-Funktionen bezeichnet. Wir wollen sie hier Objekte nennen.

Beispiel 2.21: Ein Objekt als Baustein der Sprache Mathematik

Mittels CAS sollen verschiedenste Aspekte des Objekts/Begriffs "Kreis" erschlossen werden.

Wir definieren das Objekt KREIS, indem wir ihm die Kreisgleichung zuweisen:

```
                                2            2    2
#1:  KREIS(x, y, m, n, r) := (x - m)  + (y - n)  = r      User
```

Objekte entstehen im CAS durch/über den Zuweisungsoperator (:=). Sie haben die Eigenschaft, sich in verschiedene Darstellungsebenen 'projizieren' zu lassen. Die wichtigsten dieser Ebenen sind die grafische (ikonische), die numerische, die symbolische und die logische Ebene.

```
#2:  "Projektionen des Objekts:"                          User
```

a) Projektion in die grafische Ebene (**Plot Plot**):

```
#3:  KREIS(x, y, 0, 2, 2)                                 User
```

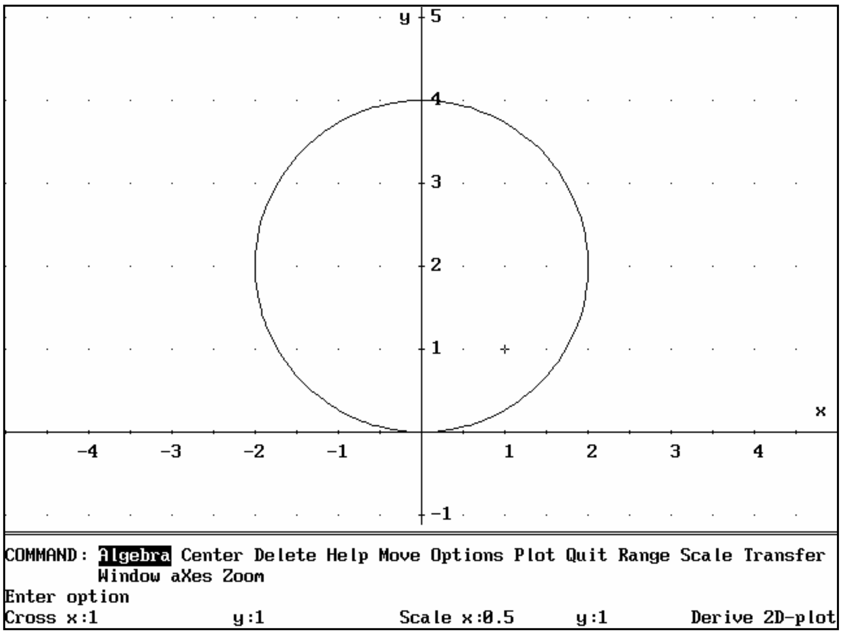

Abb. 2.23:Objekt Kreis projiziert in die Graphikebene

b) Projektion in die symbolisch-algebraische Ebene:

```
#3:  KREIS(x, y, 0, 2, 2)                               User

       2          2
#4:  x   + (y - 2)   = 4                            Simp(#3)
```

Die folgende Projektion in die symbolisch-algebraische Ebene liefert die Mittelpunkte aller Kreise, die den Punkt P(0/4) als Element besitzen:

```
#5:  KREIS(0, 4, m, n, 2)                               User

       2          2
#6:  m   + (n - 4)   = 4                          Approx(#5)
```

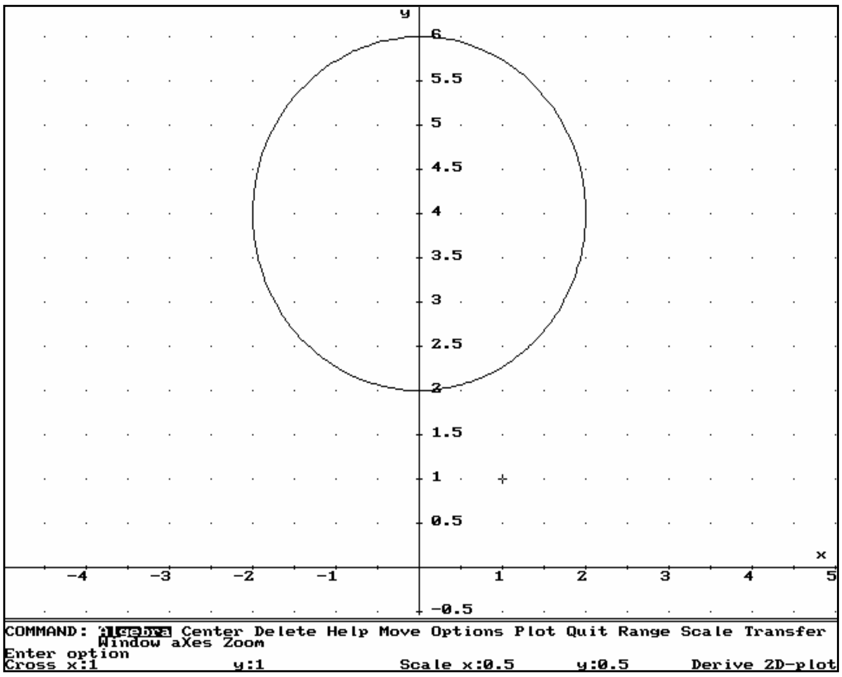

Abb. 2.24: Mittelpunkte aller Kreise

Eine weitere Projektion in die symbolisch-algebraische Ebene:

$$\#7:\quad \mathrm{KREIS}(x,\ y,\ a,\ 2\cdot b - c,\ (u - v)^2) \qquad \text{User}$$

Simp(#7)

$$x^2 - 2\cdot a\cdot x + y^2 + 2\cdot y\cdot(c - 2\cdot b) + a^2 + (2\cdot b - c)^2 = (u - v)^4$$

Wir können unser Kreis-Objekt auch differenzieren:

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

$$\#9:\quad \frac{d}{dx}\ \mathrm{KREIS}(x,\ Y(x),\ 0,\ 2,\ 2) \qquad \text{User}$$

$$\#10:\quad \left[\frac{d}{dx}\ Y(x)\right]\cdot(2\cdot Y(x) - 4) + 2\cdot x = 0 \qquad \text{Approx(\#9)}$$

$$\#11:\quad \left[\frac{d}{dx}\ Y(x)\right]\cdot(2\cdot Y(x) - 4) = - 2\cdot x \qquad \text{Simp(User)}$$

$$\#12:\quad \frac{d}{dx}\ Y(x) = \frac{x}{2 - Y(x)} \qquad \text{Simp(User)}$$

c) Projektion in die numerische Ebene:

$$\#13:\quad \mathrm{SOLVE}(\mathrm{KREIS}(x,\ y,\ 0,\ 2,\ 2),\ y) \qquad \text{User}$$

$$\#14:\quad \left[y = \sqrt{(4 - 1^2)} + 2,\ y = 2 - \sqrt{(4 - 1^2)}\right] \qquad \text{Sub(\#13)}$$

$$\#15:\quad [y = 3.73205,\ y = 0.267949] \qquad \text{Approx(\#14)}$$

d) Projektion in die logische Ebene:

Liegt der Punkt P(2/1) am Kreis?

$$\#16:\quad \mathrm{KREIS}(2,\ 1,\ 0,\ 2,\ 2) \qquad \text{User}$$

$$\#17:\quad 5 = 4 \qquad \text{Simp(\#16)}$$

Wir erhalten eine falsche Aussage, da P(2/1) nicht die Kreisgleichung erfüllt, bei Q (0 /4) ist dies anders:

$$\#18:\quad \mathrm{KREIS}(0,\ 4,\ 0,\ 2,\ 2) \qquad \text{User}$$

$$\#19:\quad 4 = 4 \qquad \text{Simp(\#17)}$$

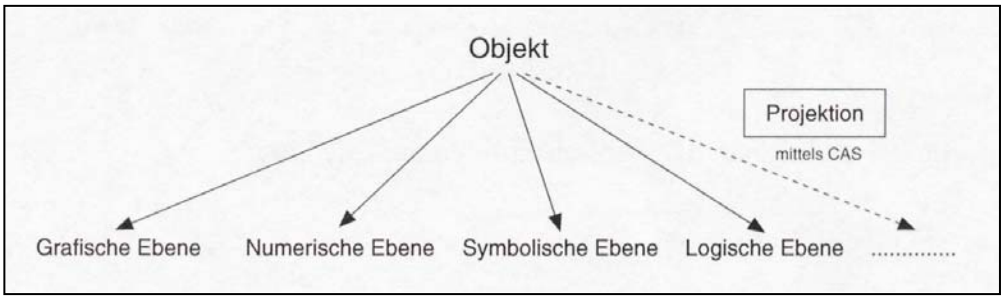

Abb. 2.25:Projektionen eines mathematischen Objekts

Das wesentliche an solchen neuen mathematischen Sprachobjekten, die durch das CAS zur Verfügung gestellt werden, besser: mit Hilfe des CAS durch den Benutzer generiert werden, ist die Möglichkeit mit Begriffen (wie eben dem Begriff 'Kreis') auf eine neue Art zu operieren. All diese vorgestellten Projektionen sind nur verschiedene Ausprägungen ein und desselben Objekts, die miteinander korrespondieren. Damit werden dem Schüler auch vielfältige Aspekte eines Begriffs leichter zugänglich. Was hier noch wichtig ist: Man wird auf neue Ideen kommen, es eröffnen sich Ansätze einer neuen Betrachtungsweise mathematischer Objekte.

Programmtechnische Anmerkungen:

Anmerkung 1:

Sehr oft treten am Beginn eines Arbeitsfiles Zeilen wie

```
#1:  Precision:=Exact
#2:  Notation:=Rational
```

auf. Dies sind Programmvoreinstellungen, die mit dem Hauptmenüpunkt **Option** gesetzt werden. Solche Voreinstellungen kann man platzsparend in eine Liste zusammenfassen:

```
#1:  [Precision := Exact, Notation := Rational]
```

Weitere Hinweise zu diesen Voreinstellungen finden sich in Kap. 5.2.3, "Unterrichtsvorbereitungen und Arbeitsunterlagen".

Anmerkung 2:

ITERATES(u,x,x0,n) wird mit vier Parametern aufgerufen:

1.Parameter: zu iterierender Term   2.Parameter: 'Laufvariable'

3.Parameter: Startwert   4.Parameter: Anzahl der Iterationsschritte

Beispiel: F(x):= Leerfunktion

`ITERATES(F(x),x,a,3)` liefert eine Liste mit vier Werten

`[a,F(a),F(F(a)),F(F(F(a)))]`

ITERATE(u,x,x0,n) liefert bei gleicher Syntax nur den letzten Wert

Anmerkung 3:

PRIME(n) ist eine logische Funktion, die überprüft, ob n eine Primzahl ist, NEXT_PRIME(n) liefert die nächste Primzahl, die größer als n ist.

Anmerkung 4:

ABS(z) liefert den Absolutbetrag der komplexen Zahl (= Abstand zwischen z und dem Ursprung in der komplexen Zahlenebene).

Die Eingabe von ABS(*a*+î .*b*) wird in der Form $|a + î \cdot b|$ dargestellt und zu

$\sqrt{(a^2 + b^2)}$ ausgewertet.

PHASE(z) liefert den Phasenwinkel des Arguments *z*, wobei (bei Einstellung im Gradmaß) gilt:

$-90° <$ PHASE(z) $\leq 90°$

Noch ein Hinweis: Die imaginäre Einheit i wird über **ALT**+i eingegeben.

Anmerkung 5:

DSOLVE1 ist Teil der Hilfsdatei ODE1.MTH (Ordinary Differential Equations) und kann als Utility-File geladen werden. Hilfsdateien sind Sammlungen von Funktionsdefinitionen (im informatischen bzw. Ca-Sinn) und Variablenzuweisungen. Zur Lösung von Differentialgleichungen stehen in DERIVE standardgemäß die Zusatzdateien ODE1.MTH (Differentialgleichungen 1.Ordnung), ODE2.MTH (Differentialgleichungen 2.Ordnung) und ODE_APPR.MTH (Näherungsmethoden für Differentialgleichungen) zur Verfügung.

Anmerkung 6:

SUM(Term(i),i,m,n) liefert die Summe über alle Summanden der Gestalt `Term(i)`, wobei i von der unteren Grenze *m* bis zur oberen *n* läuft. Zum Beispiel (Die Grenzen müssen aber nicht endlich sein):

$$\sum_{k=1}^{100} k = 5050$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = 1$$

Es kann auch über eine Liste von Werten summiert werden:

$$\Sigma(i^2, i, [2, 3, 5, 7, 11, 13, 17]) = 666$$

Anmerkung 7:

VECTOR(u,k,a,b) wird mit vier Parametern aufgerufen. Dieser Befehl entspricht einer Zählschleife, die von a bis b läuft und ihr Ergebnis in Form einer Liste liefert.

1.Parameter: Ausdruck (meist von k abhängig) 2.Parameter: 'Laufindex'

3.Parameter: untere Grenze 4.Parameter: obere Grenze

Beispiel: VECTOR(k^2,k,1,3) = [1,4,9]

Anstelle der Grenzen für den 'Laufindex' kann auch eine Liste verwendet werden.

Beispiel: VECTOR(k^2,k,[1,4,9]) = [1,16,81]

Anmerkung 8:

SOLVE(u,x) wird mit zwei Parametern aufgerufen und löst Gleichungen exakt.

1.Parameter: Gleichung/Term der mit 0 verglichen wird 2.Parameter: Lösungsvariable

Beispiel: SOLVE(x^2-4,x) = [x=-2,x=2]

SOLVE(u,x,a,b) löst Gleichungen im einem vorgegebenen Intervall [*a,b*] im Näherungsmodus und liefert die erste gefundene Lösung.

Beispiel: SOLVE(x^2-4,x,1,3) = [x=2]

Anmerkung 9:
Bei der Definition von MITTEL(i) wird der Befehl SUB verwendet, der es gestattet, im CAS einen Index zu verwenden. SUB ist eigentlich ein Operator, mit dem man über den Index ein Element aus einem Vektor (Liste) oder einer Matrix herausgreifen kann.

Die Eingabe von ["ene","mene","mu"] SUB 2 führt zu

$$[\text{"ene","mene","mu"}]_2 = \text{"mene"}$$

Anmerkung 10:
DIF(u,x,n) wird mit drei Parametern aufgerufen:

1. Parameter: Funktionsterm

2. Parameter: Ableitungsvariable

3. Parameter: Ordnung der Ableitung

Beispiel: DIF(F(x),x,2) liefert dann beim Vereinfachen den Term der 2.Ableitung

Anmerkung 11:
IF(u,r,s,t) hat vier Parameter.

1. Parameter: Bedingung(en) - logischer Ausdruck

2. Parameter: Anweisung 1 - Then-Zweig

3. Parameter: Anweisung 2 - Else-Zweig

4. Parameter: Anweisung 3 - Unknown-Zweig

Der Else- und Unknown-Zweig können entfallen. Jede Anweisung kann wieder ein IF enthalten. Damit lassen sich verschachtelte Verzweigungen aufbauen.

IF((0<x<3,x^2,2x) liefert in [0,3] die Parabelwerte, ansonsten Werte der Geraden 2 x.

Dieser Ausdruck läßt sich sofort graphisch darstellen.