• Die Analysis in den Zeiten der Computerei

Reinhard Oldenburg, Frankfurt

In diesem Beitrag werden Defizite des aktuellen Computereinsatzes im Analysisunterricht beleuchtet und ein curricularer Vorschlag unterbreitet, der eine Alternative aufzeigen soll.

1 Defizite des aktuellen Mathematikunterrichts

Es ist Allgemeingut, dass es im gegenwärtigen Analysisunterricht zu viel um das Beherrschen von Kalkülen und zu wenig um Verständnis und um Kompetenz geht. Es ist sicher auch richtig, dass zu viel Wert auf das Abarbeiten und zu wenig Wert auf das Entwickeln und Bewerten von Verfahren gelegt wird. Darüber hinaus will ich einige weitere Defizite benennen:

Defizit 1: *Es gibt fast keine Modellbildung in der Analysis!*
Dieser Befund mag überraschen, denn es gibt durchaus Aufgaben, die auf den ersten Blick wie Modellbildungsaufgaben wirken. Das Beispiel in Abb. 15.1 zeigt ein typisches Muster: Die Aufgabe besteht aus zwei Schritten, wobei im ersten Schritt modelliert wird (hier: Das Ufer wird durch ein Polynom modelliert) und erst im zweiten innermathematischen Schritt (hier: Bestimmung der Fläche unter dem Graphen der Polynomfunktion durch Integration) Analysis verwendet wird. Es wäre aber wünschenswert zu zeigen, dass in der Tat Analysis ein hervorragendes Modellbildungswerkzeug ist. Freudenthal [1973] hat in seinem Klassiker „Mathematik als pädagogische Aufgabe" eine ganze Reihe von Beispielen zusammen gestellt, in denen mit Analysis modelliert wird.

Defizit 2: *Funktionsgraph-bezogene Vorstellungen dominieren*
Ob dies überhaupt ein Defizit darstellt, kann kontrovers diskutiert werden. Es scheint mir aber sicher, dass die „ganzheitliche" Sicht, die der Graph bietet, neben Vorteilen auch Nachteile hat, weil dabei der dynamische Aspekt der Änderung zweier Größen statisch repräsentiert wird.

Defizit 3: *Konzeptentwicklung kommt zu kurz [vgl. auch Danckwerts & Vogel, 2006]*
Dieser Diagnose dürften sich die meisten Didaktiker anschließen. Ein wesentlicher Punkt zur Abhilfe liegt m. E. darin, konsequent auf Konzeptreduktion zu setzen: Wenn etwa die Konzepte des Grenzwerts und der Summation bekannt sind, kann das Riemannintegral darauf reduziert werden. Dies wird durch Computeralgebrasysteme (CAS) unterstützt.

Die Rolle von CAS ist aber ambivalent. Einige Reformhoffnungen stützen sich auf Computernutzung, allerdings gibt es auch kritische Stimmen. Exemplarisch sei als CASsandra ein Kollege zitiert: „Das mit den CAS-Rechnern ist die totale Katastrophe. Die Lehrer machen genau den gleichen Unterricht wie vorher, mit dem Erfolg, dass die Kinder gar nichts mehr lernen."

Das Zitat benennt zwei Probleme: Zum einen sind alte Ziele und neue Wege nicht automatisch kompatibel. Darüber hinaus gibt es einen erheblichen Unterschied zwischen dem, was Standardlehrer erreichen, und dem, was „Exzellenz-Lehrer" in wohlüberlegten Schulversuchen bewältigen. Angesichts dieser unübersichtlichen Lage tun die beiden folgenden Thesen u.U. bestimmten Lehrern und ihrem Unterricht Unrecht:

These 1. *Der Mathematikunterricht ignoriert Computer fast vollständig.*

These 2. *Das gilt auch bei Einsatz von GTR, DGS, TK, CAS.*

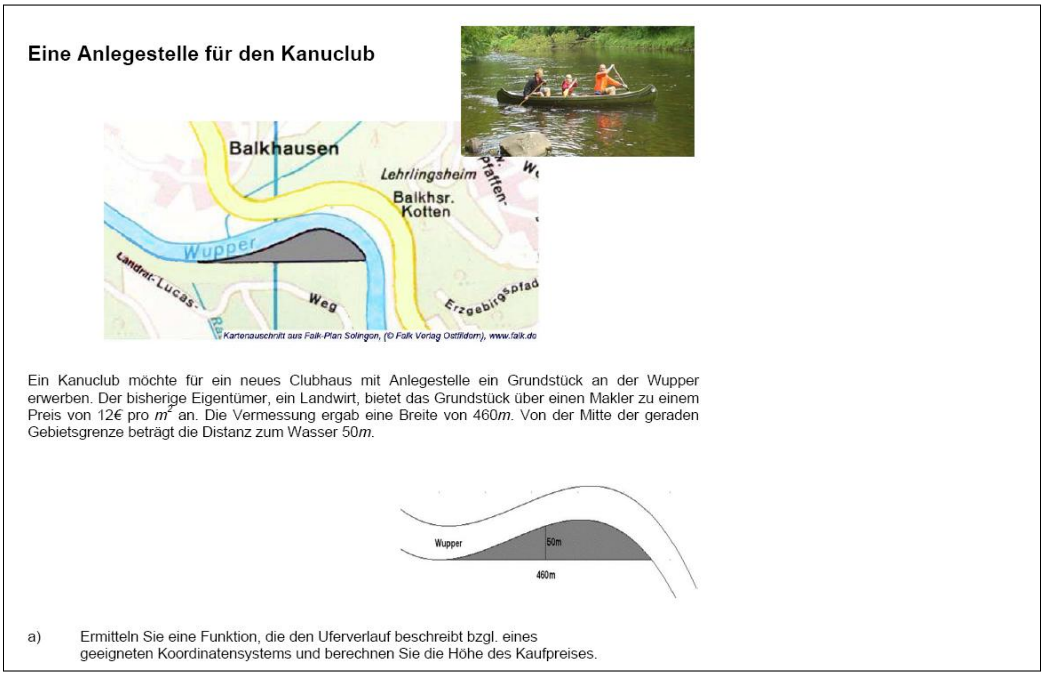
Begründung für These 2: Der Computer wird fast nur benutzt zur Visualisierung von Graphen. Allerdings sind Graphen ein statisches Medium aus der Vor-Computerzeit. Immerhin kann man durch dynamische Funktionsgraphen einen gewissen Mehrwert erzeugen, aber man versucht auch dabei, den Computer als reines Werkzeug zu benutzen. Insgesamt kann man festhalten, dass Computer bisher vor allem die Unterrichtsmethodik beeinflusst haben, nicht aber Inhalte oder Operationsmodi. Dabei könnten Computer gerade auch in den Kompetenzbereichen nützlich sein, die von der KMK nicht mit einer K-Nummer geadelt wurden, etwa „Planen" und „Beurteilen".

Computer sollten genutzt werden als
- ▷ Problemlösewerkzeug
- ▷ Problemaufwerfer
- ▷ Kognitives Werkzeug [Jonassen et al., 1998]

2 Empirik

Die Kombination von Computern und Mathematik transformiert fast alle wissenschaftlichen Disziplinen und noch weitere Bereiche unseres Lebens. Um herauszufinden, ob Schüler zumindest eine grobe Vorstellung vom Nutzen der Computer für die Mathematik haben, wurden von Markus Vogel und mir Studierende des Lehramts zu Beginn der Veranstaltung „Computer im Mathematikunterricht" mittels eines Fragebogens befragt.

86% dieser Studierenden haben im eigenen Schulunterricht Computereinsatz im Mathematikunterricht erlebt, vor allem Tabellenkalkulation (81%) und Computeralgebra (27%). Fast alle waren überzeugt, dass der Computer im MU eingesetzt werden sollte. Aber bei der Frage nach „Bei-


Eine Anlegestelle für den Kanuclub

Ein Kanuclub möchte für ein neues Clubhaus mit Anlegestelle ein Grundstück an der Wupper erwerben. Der bisherige Eigentümer, ein Landwirt, bietet das Grundstück über einen Makler zu einem Preis von 12€ pro m^2 an. Die Vermessung ergab eine Breite von 460m. Von der Mitte der geraden Gebietsgrenze beträgt die Distanz zum Wasser 50m.

a) Ermitteln Sie eine Funktion, die den Uferverlauf beschreibt bzgl. eines geeigneten Koordinatensystems und berechnen Sie die Höhe des Kaufpreises.

Abbildung 15.1: Die Kanu-Aufgabe aus Sinus-NRW

spielen, wo Computer nützlich sind für die Mathematik allgemein und ihre Anwendungen.” gaben 16% keine Antwort, 19% nannten schnelles Rechnen (ohne weitere Erläuterung, auf mündliche Nachfrage wurde gesagt, das sei nötig, damit man an der Supermarktkasse nicht lange auf die Rechnung warten müsse – ein groteske Fehleinschätzung der Geschwindigkeit von Computern), 11% nannten komplizierte Formeln zu berechnen, ohne aber Beispiele für Gebiete nennen zu können, in denen diese Formeln auftreten, und schließlich sagten sehr viele, Computer seien nützlich zur Veranschaulichung. Bei der Frage „Haben Sie schon einmal mit Gewinn Mathe mit dem Computer gemacht? Geben Sie ggf. Beispiele.” zeigte sich in den Antworten ein ähnliches Bild: 65% antworteten mit „Nein“, 8%: nannten „DGS in Geometrievorlesung”, 8% nannten den „Numerikschein”.

3 Das Curriculum

Dieser Beitrag ist motiviert vom Wunsch, die beschriebene Situation zu ändern. Dazu soll ein curricularer Vorschlag unterbreitet werden, der durch folgende Charakteristika ausgezeichnet ist:

- ▷ Die Leitidee „Änderung” steht am Anfang und wird durchgängig verwendet [vgl. Körner, 2005].
- ▷ Vom Rechnen wird zu den Konzepten fortgeschritten – Probleme lösen; authentische Fragen beantworten; Modellbildungen, in denen die Analysis eine Rolle spielt
- ▷ Grundvorstellungen betonen [Malle]
- ▷ Diff’rechnung und Integration „integrieren” (auch als eine Antwort auf den Dauerbrenner „I-Rechnung vor D-Rechnung?”, siehe Blum & Toerner [1983])
- ▷ Neben Graphen weitere Repräsentationsformen für Funktionen verwenden
- ▷ Computereinsatz – aber nicht immer und überall

Wenn Änderung als Leitidee das gesamte Curriculum durchziehen soll, dann hat das gewichtige Auswirkungen: Funktionales Denken erscheint (nur) als ein wichtiger Spezialfall. Denken in Änderungen ist für Schüler keine leichte Übung. Es zeigt sich, dass etwa die Aufgabe „Angenommen, es gilt immer $a = 2b+3$. Was passiert mit b, wenn a um 2 größer wird?” nur von 31% gymnasialer Elftklässler gelöst wird ($n > 200$). Schon in der Sekundarstufe I können Änderung und Akkumulation Computer-orientiert behandelt werden, etwa mit der Turtle-Grafik bei der Unterscheidung von `forward` und `moveTo` (relative vs. absolute Positionierung). Abb. 15.2 zeigt ein kleines Scratchprogramm, das die waagrechte Position einer Figur durch die aktuelle Lautstärke bestimmt. Wenn stattdessen die Lautstärke die Schrittweite des nächsten Schrittes bestimmt, erhält man eine Figur, die in ihrer Position die Lautstärke akkumuliert.

Viele Dinge bleiben in seinem solchen Curriculum unverändert, u.a.

- ▷ Ableitung, Ableitungsregeln
- ▷ Integral, Hauptsatz
- ▷ Bogenlänge

Es gibt auch neue Bestandteile:

- ▷ Differentialgleichungen

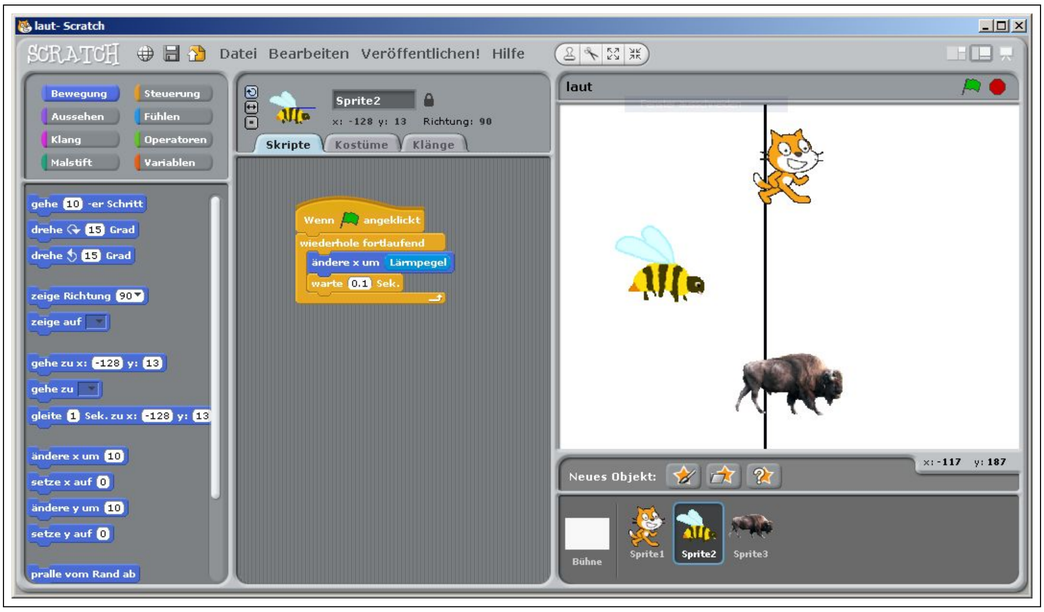


Abbildung 15.2: Änderung und Akkumulation von Lautstärke mit Scratch

▷ Numerische Optimierung
▷ Multivariate Analysis

Zum Ausgleich kann auch auf einige traditionelle Inhalte verzichtet werden:

▷ Stammfunktionskalkül (Produktregel, Substitutionsregel ...)
▷ Rotationskörper
▷ Taylorreihen

All das wäre zu begründen, aber dazu bräuchte es einer längeren Arbeit als der vorliegenden.

Damit ein Curriculum wie das hier vorgestellte erfolgreich durchlaufen werden kann, sollten die Schüler über folgende Idealvoraussetzungen aus der Sekundarstufe I verfügen:

▷ Approximationsidee bei Reellen Zahlen
▷ Einfache Algorithmen
  ◦ Heron
  ◦ Archimedes'sche Kreisapproximation
▷ Funktionen in mehreren Variablen
  ◦ Bei PC-Einsatz ist dies ohnehin sinnvoll, da z. B. TK, CAS solche verwenden.
  ◦ Funktionen als Bausteine im Sinne von Lehmann
▷ Vertrautheit mit einer Programmiersprache oder sogar mit CAS

3.1 Folgen und Funktionen

In der Sekundarstufe II kann auch ein innovativer Lehrgang ganz klassisch beginnen, nämlich mit Folgen. Diese sind als Selbstzweck interessant, nicht als Mittel zur Exaktifizierung des Grenzwerts. Eigenständig interessant sind z. B. Wachstumsprozesse, auch beschränktes Wachstum und evtl. logistisches Wachstum [siehe Körner, 2005; Danckwerts & Vogel, 2006]. Dabei ist der zentrale Begriff der der Änderung:

$$\Delta N = f(t + \Delta t) - f(t)$$

Auch numerische Verfahren können als Quelle von Folgen fungieren. Schließlich treten Folgen bei zeitlichen Prozessen auf (Weigand), etwa als Messwerte von zeitlichen Prozessen (Abkühlung einer Kaffeetasse, etc.).

Bei der Behandlung solcher Prozesse sollte durchweg viel numerisch gearbeitet werden, u.a. um Vorstellungen von Größenordnungen zu entwickeln, was letztlich wichtig ist um zentrale Ideen der Analysis zu verstehen. Bei diesen numerischen Erkundungen kann man schon eine Reihe von Konzepten kennen lernen:

▷ Konzepte: Intervallschachtelung, monotone Folge, beschränkte Folge
▷ Grenzwert einer Folge (etwa am Beispiel eines beschränkten Wachstums)
  ◦ Kreisfläche nach Archimedes damit betrachten

In Untersuchung mit CAS kann der limit-Befehl als Anreger und Unterstützer von Grenzwertüberlegungen dienen, die sich an inhaltlichen Wachstumsüberlegungen festmachen. Ein einfaches Beispiel für solche Überlegungen kodiert im CAS Maxima ist `limit(1/(1+1/n),n,inf)`. An dieser Stelle reicht eine informelle Grenzwertdefinition aus: Für ausreichend späte Werte der Folge werden die Abstände zum Grenzwert beliebig klein. Damit werden viele Wachstumsprozesse – im Rahmen des Modells – langfristig prognostizierbar (Symbolik macht Asymptotik beherrschbar). Als Initialproblem dafür kann die Abkühlung einer Tasse als Folge von Temperaturwerten gemessen und als Folge oder Funktion modelliert werden. Das Beispiel zeigt gleichzeitig, dass ähnliche Fragestellungen auch bei Funktionen relevant sind. Konzepte und Techniken zu Folgen wie Änderungsverhalten und Monotonie können dann am Graphen gedeutet werden und die Grenzwertüberlegungen mit CAS unterstützt werden, etwa `limit((3*x^2+5)/(4*x^2-x), x, inf)`. Als methodisches Ziel wird hiermit eine Vertrautheit mit dem Limit-Befehl für spätere Anwendungen angestrebt.

Funktionen sollten in verschiedenen Darstellungsformen (Graph, Dynagraph) untersucht werden und in jeder Form sollten die grundlegenden Änderungseigenschaften interpretiert werden.

Eine weitere Form von Funktionen, bei der der Aspekt der Änderung schön thematisiert werden kann, ist der Bildstrom, den eine WebCam liefert. Er kann als Funktion $B(t)$ des Bildes von der Zeit aufgefasst werden. In Abb. 15.3 ist ein Beispiel zu sehen, wo dieser Bildstrom live ins Negativ konvertiert wird und um 2 Sekunden verzögert wird. Änderungen im Bildstrom findet man durch Differenzen wie $B(t) - B(t-0.2)$, was man zur Verstärkung noch vergrößern sollte, z. B. indem man durch 0.2 teilt. Das Programm zeigt dann bei statischen Urbildern nur ein schwarzes Bild, sobald aber Bewegung ins Urbild kommt, werden die Konturen deutlich erkennbar: Der Differenzenquotient detektiert Änderungen.

3.2 Flächen

Flächeninhalte sind durch die vielfältigen Erfahrungen aus der Sekundarstufe I ein sehr greifbarer Inhalt. In der Sekundarstufe II macht man sich auf den Weg zur Integralrechnung, und wie das Sinus-Beispiel aus dem ersten Abschnitt zeigt, werden gelegentlich (Pseudo-)Anwendungen der Integralrechnung zur Flächenbestimmungen verwendet. Im Gegensatz dazu sollte ein Lehrgang, der Computer als Werkzeuge ernst nimmt, die Flächenberechnung von Polygonen an den Anfang stellen. Eine Möglichkeit ist ein Miniprojekt zur Berechnung von Flächeninhalten aus Landkarten oder aus Google-Earth-Daten. Die Problemformulierung führt über Abstraktion zur Frage der Fläche eines Polygons. Über eine schrittweise Entwicklung (Dreieck; Stern-Zerlegung; …) mit etwas Programmierung gelangt man zu einer Lösung (dies ist die Endform!) wie in den Abb. 4 und 5 dargestellt. Flächeninhalte unter Funktionsgraphen spielen dabei nur die Rolle eines Sonderfalls. Mehr „Integralrechnung" ist an dieser Stelle noch gar nicht nötig und auch nicht sinnvoll. Beachtung verdient aber, wie mathematische Konzepte (Summation und Grenzwert) im CAS umgesetzt werden können.

3.3 Optimierung

Das übliche Curriculum behandelt Optimierungsprobleme als Anwendung der Differentialrechnung, man kann die Reihenfolge aber auch umkehren und Extremwertprobleme vor Ableitungen thematisieren! Es gibt viele sinnvolle Fragestellungen, von der „alten Schachtel" bis zur Milchtüte [Boer] in denen sich eine Sachsituation auf die Frage verdichten lässt, das Minimum oder Maximum einer Funktion zu bestimmen. Allerdings verschenkt man etwas, wenn man die Funktion gleich als Graph zeichnen lässt. Besser scheint mir, mit Zahlenwerten zu starten. Als konkretes Beispiel sei die Frage gestellt, bei welchem Durchmesser ein Zylinder des Volumens 850 minimale Oberfläche hat. Man startet mit einem konkreten Zahlenwert für den Durchmesser (z. B. den einer tatsächlich vorhandenen Dose; dies liefert den Startwert) und berechnet die zugehörige Oberfläche. In einem Koordinatensystem würde das einen einzigen Punkt liefern. Der Wunsch nach Optimierung lässt sich jetzt so konzeptualisieren: Kann man den Durchmesser so ändern, dass die Oberfläche abnimmt? Ein probeweises Vergrößern des Durchmessers um einen kleinen Schritt liefert einen zweiten Punkt und anhand der Änderung der Oberfläche lässt sich sofort sagen, ob man in diese Richtung gehen sollte oder lieber in die andere. Damit ist ein iteratives Verfahren zur Berechnung vorbereitet. Ziel für die Schülerinnen und Schüler sollte eine bequeme und schnelle maschinelle Suche sein. Das leistet im analogen Problem einer Maximierung z. B. der folgende Algorithmus:

```
def f(x): return 1.0/(x*x+3*x+10)
x0=0
delta=0.00001
while True:
    if f(x0-delta)>f(x0): x0+= -delta
    else:
        if f(x0+delta)>f(x0): x0+= delta
        else: break
print "Maximum␣bei:", x0
```

Numerische Algorithmen sind durch die begrenzte Sicht des Computers auf die Funktion gekennzeichnet, der eben den Graphen nicht ganzheitlich wahrnehmen kann. Die Umsetzung in ein Programm ist – bei Kenntnis einiger Konzepte einer Programmiersprache, wie sie in etwa 15 Unterrichtsstunden erworben werden können – sehr einfach.

Damit erschließt man sich sehr viele Anwendungen, insbesondere können praktisch alle schulüblichen Optimierungsprobleme behandelt werden. Denkbare Varianten zum Algorithmus gibt es viele, so lässt sich z. B. mit Einschachtelungsalgorithmen (siehe Istron-Band 9) die Genauigkeit noch wesentlich erhöhen.

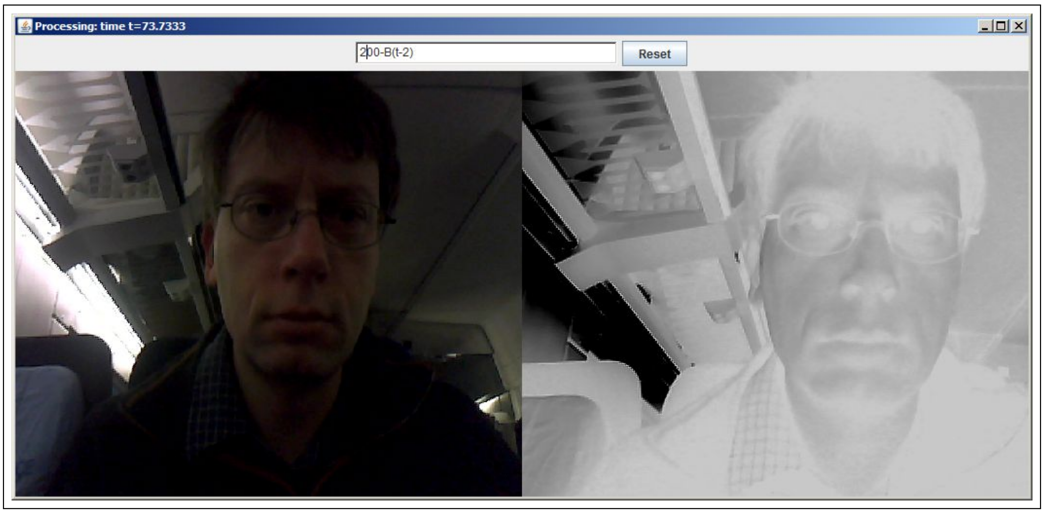


Abbildung 15.3: Ein Web-Cam-Funktions-Programm

Der Algorithmus kann auch zum Gegenstand einer Problematisierung gemacht werden: Wenn eine Funktion mehrere Maxima/Minima besitzt, welches wird gefunden? Wodurch wird die Genauigkeit bestimmt?

Da alle schulüblichen Probleme (und viele mehr) lösbar sind, stellt sich die Frage, wozu die Theorie überhaupt noch notwendig ist. Antwort: Theorie ist nötig, um theoretisch fundiertes Wissen zu gewinnen. Aber Schüler kennen jetzt die Theorie noch nicht! Können sich theoretische Konzepte aus der numerischen Anwendungssituation entwickeln lassen? Dazu eignen sich Prinzipfragen, zum Beispiel: Hat $f(x) = 1.0/(x \cdot x + 3 \cdot x + 10)$ nur ein Maximum? Eine Antwort kann über Ungleichungen gegeben werden, aber das hat ein hohes algebraisches Anforderungsniveau. Man kann aber auch den Algorithmus neu betrachten: $(x_0; y_0)$ ist das einzige Maximum, wenn rechts davon der Algorithmus immer nach links und links davon immer nach rechts läuft, und das ist für $f(x+\Delta) > f(x)$ bzw. $f(x+\Delta) < f(x)$ der Fall.

Das Wachstumsverhalten entscheidet sich also an $f(x+\Delta) - f(x) > 0$ oder < 0. Monoton steigend bzw. fallend können so algebraisch definiert werden, wobei aber sofort die Frage nach der geeigneten Größe für Δ aufkommt. Bei zu großem Wert könnte man beim „Bergsteigen” einen Schritt über ein Minimum hinweg machen. Also ergibt sich die Zielformulierung „delta möglichst klein zu wählen”.

Damit ist die Bühne für den Differenzenquotienten vorbereitet. Es bietet sich an, jetzt auf die Experimente zum Ableitungsbegriff [Oldenburg, 2007] zu wechseln, um die zentralen Grundvorstellungen möglichst früh und koordiniert zu entwickeln. Ergebnisse davon sind: Differenzenquotient; Grundvorstellungen zu Tangente, Änderungsrate, linearer Approximierbarkeit; „Differentialquotient bestimmt Änderungen”.

Insbesondere die Grundvorstellung zur „Linearen Approximation” $f(x+\Delta x) = f(x) + f'(x) \cdot \Delta x + \text{Fehler}$, die durch das Experiment mit der Kugelbahn und durch das Funktionsmikroskop entwickelt werden kann, ist für numerische Anwendungen wichtig. Man fasst sie z. B. in Änderungssprache als: Die Ableitung ist eine Näherung des Verstärkungsfaktor der Änderungen:

$$\Delta f(x) \approx f'(x) \cdot \Delta x$$

Diese Sichtweise ist nützlich bei der Berechnung von Getriebemaschinen und kann anschaulich bei Dynagraph-Anwendungen erfahren werden (mit Felix1D ist auch gleich die Umkehrfunktion erfassbar). Dabei wird auch die Grundvorstellung „Ableitung detektiert Änderung” entwickelt (das WebCam-Programm kann hier aufgegriffen werden).

Das Kugelbahn-Experiment [siehe Oldenburg, 2007] illustriert die lineare Prognose, die man zu folgender allgemeiner Definition ausbauen kann: Zu einer Funktion $y = f(x)$ sind an einem Punkt $(x_0; y_0)$ die Differentiale dx, dy die bestmögliche lineare Prognose für die Änderungen von x und y um diesen Punkt. Demnach sind $\Delta x = dx$ willkürlich (nicht infinitesimal) und die Ableitung ist der Proportionalitätsfaktor der linearisierten Änderungen. Das Funktionsmikroskop zeigt die lokale Linearität: $\Delta y \approx dy$ nahe bei $(x_0; y_0)$.

Man beachte, dass die Formulierung Anleihen beim Wahrscheinlichkeitsbegriff nimmt. Eine weitere berücksichtigte Erkenntnis ist, dass Änderungsraten schwieriger als Änderung sind (0/0-Problem). Historisch gesehen gibt es viele Lehrbücher, die ganz unbefangen mit Differentialen arbeiten, z.B. indem sie „Leibniz”-Differentiale (Definition $dy = f'(x)dx$) einführen. Infinitesima-

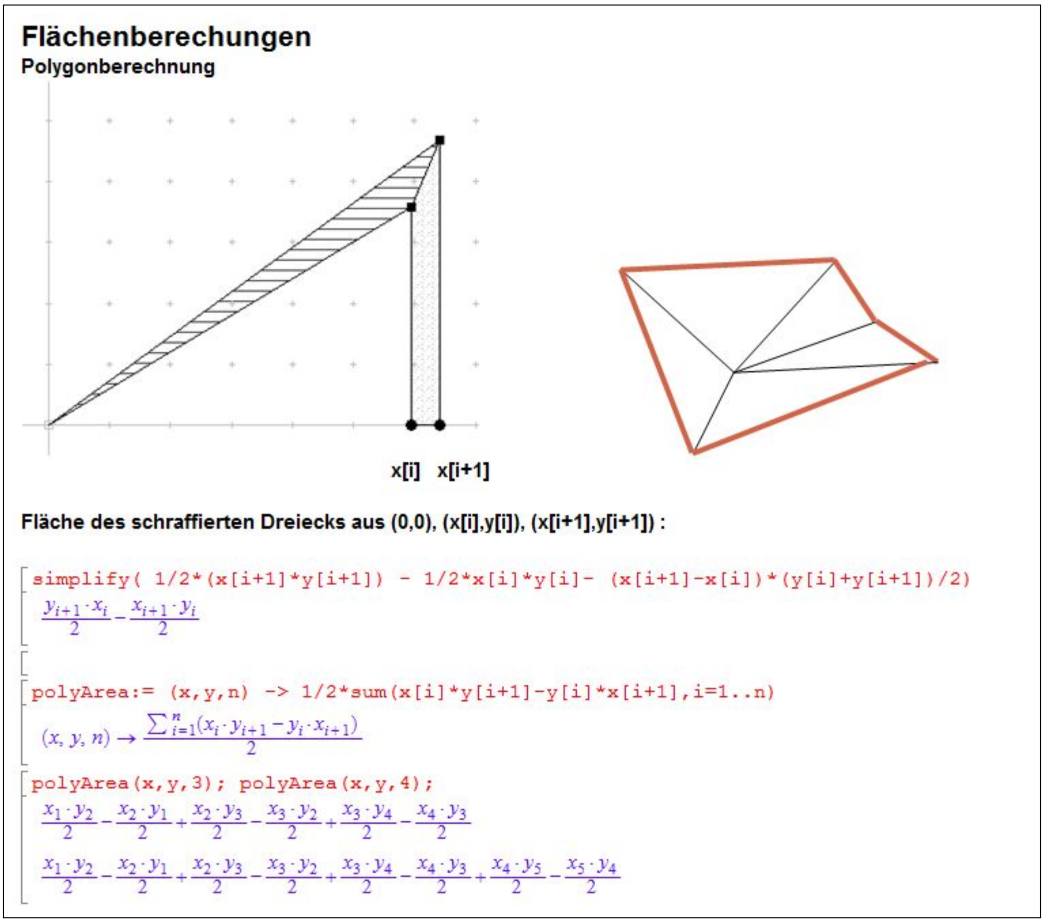


Abbildung 15.4: Polygonberechnung in MuPAD I

le im Sinne der Nichtstandardanalysis sind nicht nötig, aber eine interessante Option [Wunderling, 2007; Artigue, 2002]. Differentiale sind nützliche Beschreibungswerkzeuge in Theorie (z. B. totales Differential bei algebraischen Kurven) und Modellbildung (Klassiker bei Freudenthal; Roboterbewegung).

3.4 Differentialgleichungen

Differentialgleichungen werden im heutigen Analysisunterricht nur selten behandelt. Trotzdem spielen sie eine Rolle, wenn auch implizit. Unter dem Gesichtspunkt „Rekonstruktion des Bestandes aus der Änderungsrate" werden etwa Aufgaben gestellt wie die Folgende:

Aufgabe. *En Becken ist anfangs leer, der Zufluss ist $z(t) = 10 + 10 \cdot t$ (in Liter pro Sekunde) Gefragt ist der Bestand (Füllvolumen) V zur Zeit $t = 20$?*

Diese Informationen können ohne eine Differentialgleichung (DGL) nicht angemessen formalisiert werden: Die Schüler und Schülerinnen sollen also unvermittelt ein Integral hinschreiben. Die direkte Übertragung der Information der Aufgabenstellung in mathematische Formalismus liefert dagegen eine Differentialgleichung:

$$V(0) = 0$$

$$V'(t) = 10 + 10 \cdot t$$

Gesucht: $V(20)$

Die traditionelle Lösung mit dem Hauptsatz führt dann auf das Integral.

Eine technologielastige, numerische Lösung dagegen entsteht sehr einfach aus der Differentialgleichung, die man mit Differentialen zunächst als $dV = (10 + 10 \cdot t) \cdot dt$ schreibt und dann in lokaler lineare Näherung die Differentiale durch Änderungen ersetzt. In der Programmiersprache Python sieht die Lösung dann so aus:

```
V=0
t=0
dt=0.5
def z(t): return 10+10*t

while t<20:
    V+= z(t)*dt
    t+= dt
    print "t=",t," V(t)=",V
```

Die Ausführung liefert:

```
t= 0.5   V(t)= 5.0
```

```
polyArea([1,1,0,1],[0,5,0,0],3);  polyArea([1,5,6,2,1],[1,1,4,4,1],4);
```

$\frac{5}{2}$

12

Frage: Ist das translationsinvariant?

```
kreisArea:= (n) -> 1/2*sum(cos(i/n*2*PI)*sin((i+1)/n*2*PI)-
                   sin(i/n*2*PI)*cos((i+1)/n*2*PI),i=1..n)
```

$$n \to \frac{\sum_{i=1}^{n}(\cos(\frac{i}{n}\cdot 2\cdot\pi)\cdot\sin(\frac{i+1}{n}\cdot 2\cdot\pi) - \sin(\frac{i}{n}\cdot 2\cdot\pi)\cdot\cos(\frac{i+1}{n}\cdot 2\cdot\pi))}{2}$$

```
limit(kreisArea(n),n=infinity)
```

π

Der Flächeninhalt unter einem Funktionsgraphen ist ebenfalls ein Sonderfall: x[0]=a, y[0]=0, dann x[1]=a, y[1]=f(x[1]) ...
Das kann ausgeführt werden. Einfacher ist aber:

```
funArea:= (f,a,b,n) -> sum((f(a+(i+1)*(b-a)/n)+f(a+i*(b-a)/n))/2*(b-a)/n,i=1..n-1)
```

$$(f, a, b, n) \to \sum_{i=1}^{n-1} \frac{\frac{f\left(a+\frac{(i+1)\cdot(b-a)}{n}\right)+f\left(a+\frac{i\cdot(b-a)}{n}\right)}{2}\cdot(b-a)}{n}$$

```
funArea(x->x^2,0,b,n)
```

$$\frac{b\cdot(2\cdot b^2\cdot n^3+b^2\cdot n-3\cdot b^2)}{6\cdot n^3}$$

Wichtig: Nummerisches ausrechnen (auch mit konventioneller Programmiersprache möglich!)

```
float(funArea(x->x^2,0,10,1000))
```

333.3334995

```
limit(funArea(x->x^2,0,b,n),n=infinity)
```

$\frac{b^3}{3}$

Abbildung 15.5: Polygonberechnung in MuPAD II

```
t= 1.0  V(t)= 12.5
t= 1.5  V(t)= 22.5
t= 2.0  V(t)= 35.0
...
t= 20.0  V(t)= 2150.0
```

Das gleiche geht natürlich auch in anderen Sprachen und insbesondere in CAS. Dort hat man eine weitere, bisher viel zu selten genutzte Option: Durch gezieltes Löschen numerische Information erhält man eine (halb)-symbolische Lösung:

Die folgenden Maxima-Befehle liefern das gleiche Ergebnis wie obiges Python Programm:

```
V:0; t:0; dt:0.5;
z(t):=10+10*t;
while t<20 do (
  V:V+z(t)*dt,
  t:t+dt,
  print(t,V)
);
```

Wenn man nun die Funktion `z` löscht (mit `kill(z);`) erhält man sukzessive Summen von Produkten der Art `z(0)*0.5+z(0.5)*0.5+z(1)*0.5+...` Wenn man sogar noch `dt` löscht (dann muss man die Schleife auch etwas modifizieren), ergibt sich sogar `dt*z(4*dt)+dt*z(3*dt)+dt*z(2*dt)+dt*z(dt)+z(0)*dt`.

Es zeigt sich also, dass die Lösung der DGL auf eine Struktur führt, die man von der Berechnung von Flächeninhalten unter Funktionsgraphen kennt. Die Lösung der DGL erfolgt also durch Integration. Das ist bereits die Essenz des Hauptsatzes, dessen Aussage intensiv – auch numerisch – erkundet werden sollte.

Wegen der Wichtigkeit noch einmal explizit: Die CAS-Befehle delete/kill sollten nicht nur als Hilfsmittel zum Handling der Programme gesehen werden, sondern als kognitive Tools!

Im Sinne eines Spiralcurriculum können zu diesem Stand des Curriculums Wachstumsprozesse neu formuliert werden.

In Kooperation mit der Physik können die Bewegungsgleichungen der Mechanik $v(t) = x'(t)$, $F/m = v'(t)$ iterativ gelöst werden. Das geht sogar vektoriell und führt mit ein paar Zeilen Code zu einer auch optisch ansprechenden Simulation:

```
from visual import *

blowup=10 # sowie mal groesser als
    masstaeblich
erde = sphere(pos=(0,0,0), radius=6300000*
    blowup,
  color=color.blue)
erde.v=vector(0,0,0)
erde.m=5.9e24 # Masse in kg
```

```
# Einige Informationen aus dem Lexikon...
mond = sphere(pos=(370000000,0,0), radius
     =1700000*blowup,
  color=color.red)
mond.v = vector
     (0,0,370000000*6.28/(28*24*3600))
mond.m= 7.3e22
dt = 1800                     # Zeitschritt 30
     Minuten

def kraft(A,B): # Kraft, die B auf A üausbt
    gamma=6.6e-11
    r=mag(B.pos-A.pos) # äLnge des Vektors =
         Abstand
    return (B.pos-A.pos)*A.m*B.m*gamma/(r*r*r)

while True:
    rate(48) # 1s in Simulation entspricht 48*
         dt=24h
    mond.v+= kraft(mond,erde)/mond.m *dt
    erde.v+= kraft(erde,mond)/erde.m *dt
    mond.pos+= mond.v *dt
    erde.pos+= erde.v *dt
```

Mit drei weiteren Code-Zeilen lässt sich die Position des Mondes mit der Maus verschieben.

3.5 Kalkül

Die Idee der Konzeptreduktion ist nicht neu, aber immer noch relevant: Wenn die Schüler die Konzepte der Summation und des Grenzwertes verstanden haben (z. B. in Form der Funktionen limit und sum eines CAS), dann können sie versuchen ihre intuitiven Konzepte z. B. zum Integral oder zur Ableitung auf diese Basiskonzepte zu reduzieren. Bei der Behandlung der Ableitung ist es durch CAS-Nutzung möglich, verschiedene Ableitungsbegriffe zu untersuchen. In Oldenburg [2007] wurde berichtet, dass Schüler anhand der dort beschriebenen Experimente ganz unterschiedliche Differenzenquotienten gebildet haben, so gab es rechts-, links- und beidseitige Varianten.

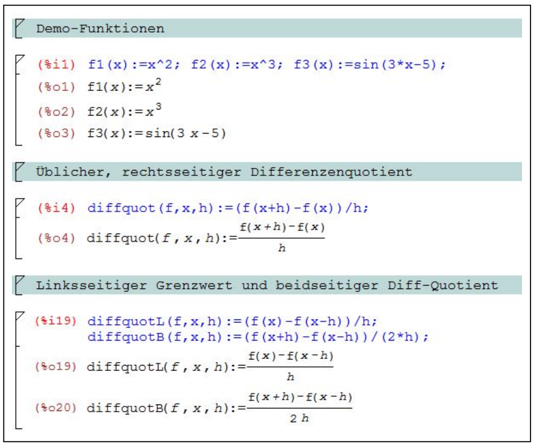


Abbildung 15.6: Ableitungen aus verschiedenen Differenzenquotienten I

Um die Herleitung von Kalkülregeln zu üben, ist es ein großer Vorteil, verschiedene Fassungen zu haben, weil damit (leichtere) Analogieaufgaben gestellt werden können. In diesem Sinne bietet es sich auch an, die q-Ableitung [Oldenburg, 2005] einzubeziehen. Die Arbeit damit sollte aus einer Mischung von händischen und CAS-basierten Phasen bestehen, um jeweils klar zu machen, ob tiefe Fähigkeiten der Blackbox verwendet werden oder nicht. Die folgenden Abbildungen zeigen einige Impressionen von Rechnungen im CAS Maxima.

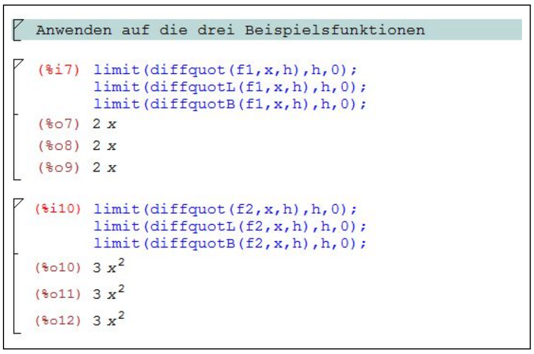


Abbildung 15.7: Ableitungen aus verschiedenen Differenzenquotienten II

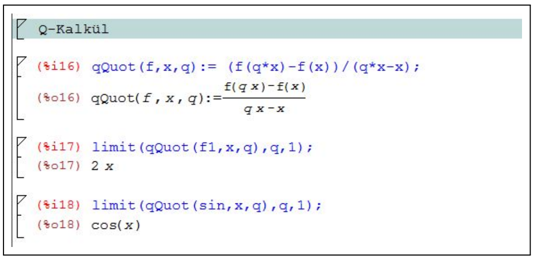


Abbildung 15.8: q-Ableitung

3.6 Integral

Eine Zusammenschau der bisher schon behandelten Fragestellungen Flächeninhaltsberechnungen und Rekonstruktion von Beständen aus Änderungen ergeben als Gemeinsamkeit, dass Grenzwerte von Summen bestimmt werden. Damit ist auch gleich ein Verfahren zur approximativen Berechnung von Integralen verfügbar und Eigenschaften wie die Linearität und die Additivität über Integrationsintervallen lassen sich direkt ablesen. Mittels Konzeptreduktion lassen sich Integrale dann im CAS auch definieren und der Hauptsatz kann mehr oder weniger klassisch behandelt werden. Damit ist das Integralkonzept erarbeitet, reduziert auf andere Konzepte und mit dem Lösen von DGL vernetzt. Eine weitergehende Behandlung z. B. des Stammfunktionenkalküls erscheint dagegen überflüssig, weil sie wenig neue Sachverhalte erschließt und vor allem Rechentechniken bringt.

3.7 Multivariate Analysis

Die durch Ausdünnung der Integralrechnung gewonnene Zeit kann in eine Behandlung von multivariaten Fragestellungen behandelt werden. Funktionen in mehreren Variablen und die Frage nach ihren Extremstellen lassen sich leicht aus Anwendungen gewinnen (Physik, Fermatpunkt, etc.). Die numerische Behandlung ist nicht viel schwieriger als im eindimensionalen Fall. Es erschließen sich u. a. viele Anwendungen im Bereich der Modellierung von Daten [Engel, 2009; Oldenburg, 2009]. Auf der AKMUI-Tagung 2009 hat Joachim Engel gezeigt, wie eine Potenzfunktion benutzt werden kann, um Messwerte eines Mikrowellenexperimentes zu fitten (siehe Engel [2012], auf S. 149 in diesem Band). Engel benutzte dazu mit dem Programm R eine mächtige Blackbox. In Erweiterung des obigen Optimierungsbeispiels kann man die Dinge aber auch algorithmisch elementar angehen:

```
daten=[[2,19],
       [3,15],
       [4,13],
       [5,12],
       [8,10]]

def S(a,b): # quadratische Fehlersumme des
     Modells a*x^b
    global daten
    S=0
    for [x,y] in daten: S+=(y-a*x**b)**2
    return S

def optimiereInXRichtung(ab,delta):
    [a,b]=ab
    if S(a+delta,b)>S(a,b): delta=-delta
    while S(a+delta,b)<S(a,b): a+=delta
    return [a,b]

def optimiereInYRichtung(ab,delta):
    [a,b]=ab
    if S(a,b+delta)>S(a,b): delta=-delta
    while S(a,b+delta)<S(a,b): b+=delta
    return [a,b]

def optimiere(ab,delta):
    while True:
        print ab
        ab1=optimiereInXRichtung(ab,delta)
        ab2=optimiereInYRichtung(ab1,delta)
        if abs(ab[0]-ab2[0])+(ab[1]-ab2[1])
             <0.001*delta:
            return ab2
        ab=ab2

print "Minimum bei:",optimiere([10,1],0.0001)
```

Es braucht kaum erwähnt zu werden, dass dieser Algorithmus aus technischer Sicht sehr schlecht ist: Es geht um Größenordnungen schneller (und auch genauer), aber aus didaktischer Sicht ist das kein gravierendes Argument.

Weitere Anwendungen der multivariaten Analysis:

- ▷ Deutung von partiellen Ableitungen mit GoogleEarth
- ▷ Helligkeitsgradienten in (digitalen) Bildern
- ▷ Implizite Kurven, Normalen daran

Diese Beispiele sollen nur illustrieren, dass sich hier ein reiches Betätigungsfeld erschließt, das es in der Vergangenheit möglicherweise nur deswegen nicht zum Schulstoff gebracht hat, weil die Visualisierungs- und Berechnungsmöglichkeiten fehlten, die heute Dank einfacher Programmiersprachen, CAS und dynamischer Raumgeometrie (insbesondere Archimedes Geo3D eignet sich hier) allgemein verfügbar sind.

4 Schlussbetrachtungen

Kritische Einwände könnten sein: Muss man wirklich programmieren? Ist das noch zeitgemäß? Sehr viel geht auch mit TK, aber

- ▷ Programme bringen Konzepte oft klarer und vor allem knapper zum Ausdruck
- ▷ Interviews mit Ingenieuren: Obwohl heute viele naturwissenschaftliche Anwendungen ohne Programmieren nutzbar sind (z. B. Finite Elemente Methoden mit Designer), bleibt Programmieren Alltagsgeschäft
- ▷ Allgemeinbildende Schule soll nicht Nutzerwissen vermitteln, sondern Einsichten (z. B. auch in Genauigkeitsgrenzen)

Die Ideen der Logo-Didaktik können heute viel weiter gefasst werden und sich nicht nur auf das Erstellen netter Zeichnungen beziehen, sondern auch auf Fragen der Analysis und der weiteren Oberstufenmathematik. Vor diesem Hintergrund ist die Kritik der Logo-Philosophie [Bender, 1987] neu zu bewerten.

Literatur

Artigue, Michèle (2002): Analysis. In: Tall, David (Hg.): Advanced Mathematical Thinking, New York: Kluwer.

Bender, Peter (1987): Kritik der Logo-Philosophie. Journal für Mathematik-Didaktik, 8, 3–103.

Blum, Werner & Günter Toerner (1983): Didaktik der Analysis. Moderne Mathematik in elementarer Darstellung, Göttingen: Vandenhoeck und Ruprecht.

Danckwerts, Rainer & Dankwart Vogel (2006): Analysis verständlich unterrichten. München: Elsevier.

Engel, Joachim (2009): Anwendungsorientierte Mathematik: Von Daten zur Funktion. Eine Einführung in die mathematische Modellbildung für Lehramtsstudierende. Heidelberg, Berlin: Springer Verlag.

Engel, Joachim (2012): Von Daten zur Funktion: Skizzen eines anwendungsorientierten Analysisunterricht. In: Kortenkamp, Ulrich & Anselm Lambert (Hg.): Medien Vernetzen / Zur Zukunft des Analysisunterrichts vor dem Hintergrund der Verfügbarkeit Neuer Medien (und Werkzeuge), Hildesheim: Franzbecker, 147–152.

Freudenthal, Hans (1973): Mathematik als pädagogische Aufgabe, Band 2. Stuttgart: Klett.

Jonassen, David, Chad Carr & Hsiu-Ping Yueh (1998): Computers as mindtools for engaging learners in critical thinking. TechTrends, 43, 24–32, 10.1007/BF02818172, URL http://dx.doi.org/10.1007/BF02818172.

Körner, Henning (2005): Mit Wachstum durch die Analysis. Der Mathematikunterricht, 51(4), 4–18.

Oldenburg, Reinhard (2005): The Q-Way of Doing Analysis. International Journal for Technology in Mathematics Education, 12(4), 155–160.

Oldenburg, Reinhard (2007): Experimentell zum Ableitungsbegriff. mathematik lehren, (141), 52–56.

Oldenburg, Reinhard (2009): Ein Bild zerfließt. mathematik lehren, (157), 56–59.

Wunderling, Helmut (Hg.) (2007): Analysis als Infinitesimalrechnung. Berlin: Duden Paetec.