Hans-Stefan Siller, Universität Salzburg

Zwei Fächer, eine Idee – Funktionales Modellieren in Mathematik und Informatik

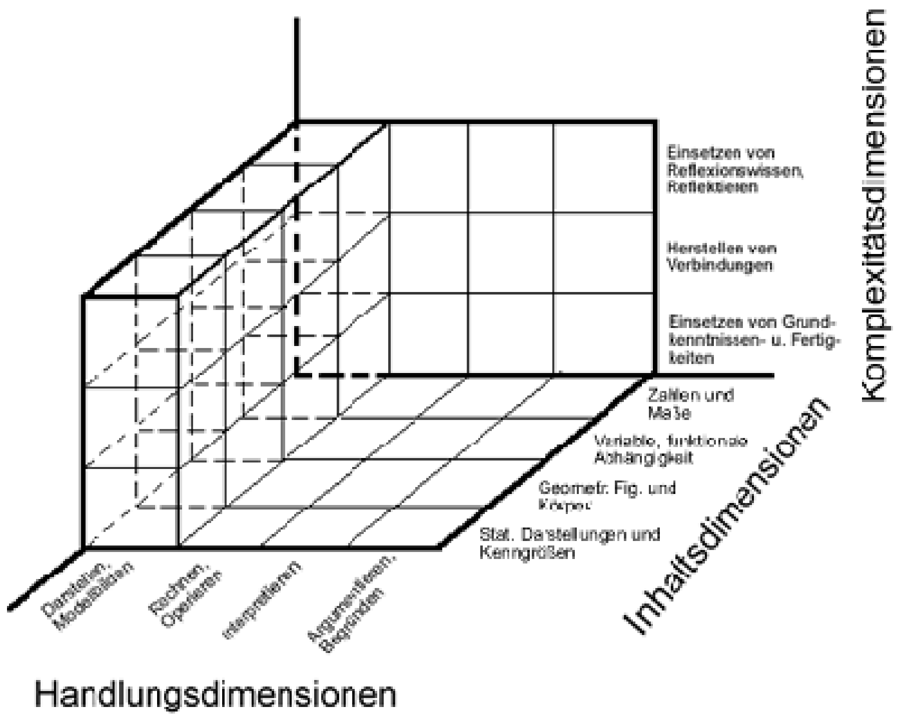
Abstract:

Durch die Entwicklung von Bildungsstandards bzw. durch die Betonung fächerübergreifenden / -verbindenden Arbeitens im Mathematikunterricht hat Modellieren an zentraler Bedeutung gewonnen. Modellieren / Modellbilden ist ein Thema, das auch in der Informatik eine große Bedeutung hat. Versucht man in beiden Fächern den Modellierungsaspekt zu betonen, stößt man auf die Idee der funktionalen Modellierung. Sie erlaubt es zum einen, mathematische Inhalte im Informatikunterricht zu diskutieren. Andererseits stellen informatische Sichtweisen zur funktionalen Modellierung eine Bereicherung des Mathematikunterrichts dar.
In diesem Vortrag sollen die Möglichkeiten einer fächerverbindenden funktionalen Sichtweise vorgestellt und diskutiert werden. Außerdem werden unterrichtsrelevante Beispiele zu lehrplankonformen Themen in Unter- und Oberstufe den Teilnehmer/innen präsentiert.

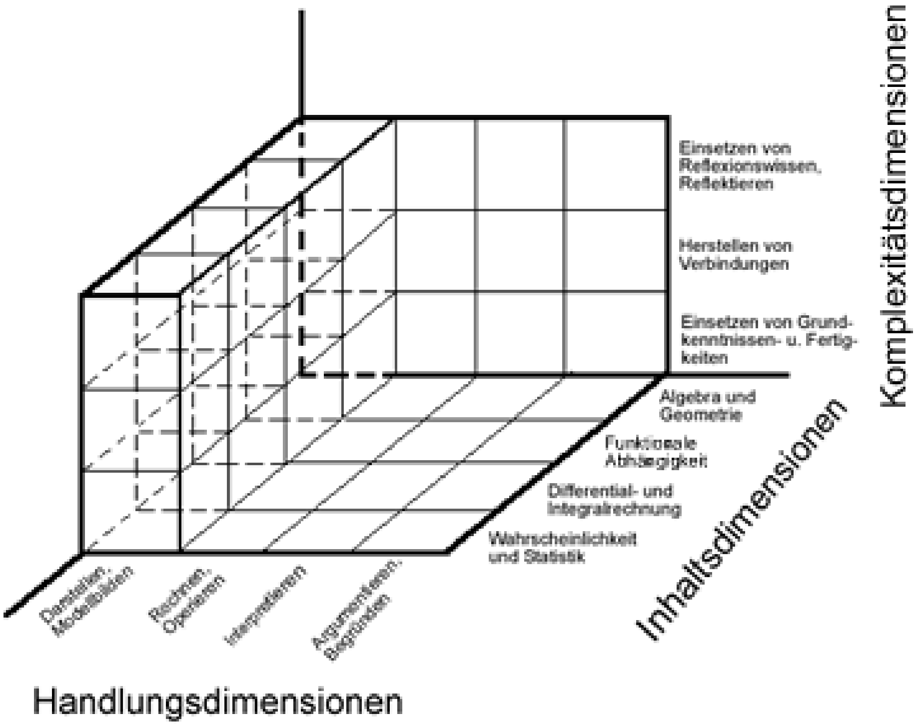
1. Modellieren aus der Sicht der Bildungsstandards

Seit der Entwicklung der Bildungsstandards für das Fach Mathematik in Österreich ist die zentrale Idee der Modellbildung wieder in das Bewusstsein der Mathematiklehrer/innen in Österreich getreten. Das Modellbilden nimmt als eine der Handlungskompetenzen im Kompetenzmodell [12], [15] eine zentrale Rolle ein. Ich habe versucht dies in den Grafiken darzustellen:

- Kompetenzmodell für die Unterstufe



- Kompetenzmodell für die Oberstufe



In den schriftlichen Erklärungen zum Kompetenzmodell kann man zum Punkt Modellbilden lesen: „Modellbilden erfordert über das Darstellen hinaus, in einem gegeben Sachverhalt die relevanten mathematischen Beziehungen zu erkennen (um diese dann in mathematischer Form darzustellen), allenfalls Annahmen zu treffen, Vereinfachungen bzw. Idealisierungen vorzunehmen u. Ä.
Charakteristische Tätigkeiten sind z. B.:

- Alltagssprachliche Formulierungen in die Sprache/Darstellung der Mathematik übersetzen.
- Problemrelevante mathematische Zusammenhänge identifizieren und mathematisch darstellen.
- Ein für die Problemstellung geeignetes mathematisches Modell verwenden oder entwickeln.
- Verschiedene mathematische Modelle für ein Problem entwickeln und ihre Problemadäquatheit abwägen.
- Geeignete Darstellungsformen oder Technologien auswählen.
- Einen gegebenen mathematischen Sachverhalt in eine andere Darstellungsform übertragen (in eine tabellarische, grafische, symbolische, rekursive oder bei Technologienutzung in eine werkzeugspezifische Darstellungsform); zwischen Darstellungen oder Darstellungsformen wechseln.
- Komplexe Probleme modularisieren.“

Der Begriff Modellbilden, wie im österreichischen Kompetenzmodell verwendet, ist ein, wie man an den charakteristischen Tätigkeiten erkennen kann, an und für sich sehr weit gefasster Begriff. Trotzdem werden unter dem Begriff des Modellbildens im Kompetenzmodell im

Wesentlichen „nur“ die Punkte: Vereinfachen, Erstellen eines (tragfähigen) Realmodells und Mathematisieren verstanden. Aufgrund der Einteilung der Handlungsdimensionsachse im Kompetenzmodell ist dies auch nicht weiter verwunderlich, da die weiteren Handlungsdimensionen im Kompetenzmodell „Operieren/Rechnen“, „Interpretieren“ und „Begründen/Argumentieren“ sind. Unter dem Aspekt der Überprüfung kann man diese Einteilung als sinnvoll erachten. Betrachtet man aber den Begriff aus „streng“ mathematik-didaktischer Sicht (vgl. Kapitel 2) ist das Darstellen bzw. Erstellen einer mathematischen Beschreibung der Problemstellung zu wenig.

2. Modellieren im Mathematikunterricht

Über den Begriff Modellieren bzw. Modellbilden wird unter Fachdidaktiker/innen, Fachwissenschaftler/innen, aber auch unter Lehrer/innen seit einigen Jahren heftig diskutiert. So hat es auch immer wieder neue Ansätze für den Modellbildungsbegriff aus Sicht der Fachdidaktik gegeben, welche eine Vielzahl von Beispielen, die man mit Schüler/innen behandeln könnte, entstanden ist. Aus der Diskussion über diesen Begriff hat sich herauskristallisiert, dass der Prozess des Modellbildens (= Modellieren) auf vier Grundpfeilern beruht:

- Vorliegen einer realen Ausgangssituation
- Mathematisches Modell
- Aussagen, Folgerungen
- Aussagen über die Ausgangssituation

Diese vier Grundpfeiler beschreiben den Modellbildungsprozess in seinen einfachsten Strukturen. Der Unterschied zum Modellbildungsbegriff im österreichischen Kompetenzmodell ist jedoch auch schon hier augenscheinlich. Der meiner Meinung nach detaillierteste Modellbildungsprozess, welcher neben anderen publizierten Modellbildungsprozessen, z.B. Weigand/Weller, Schweiger, Reichel/Humenberger oder Tietze/Klika/Wolpers, in der Literatur zu finden ist, stammt von Blum/Leiß [2]. In Abbildung 1 ist dieser Kreislauf[1] dargestellt. Hier wird von einem vorliegenden Problem ausgegangen und entsprechend der Nummerierung werden folgende Schritte durchlaufen:

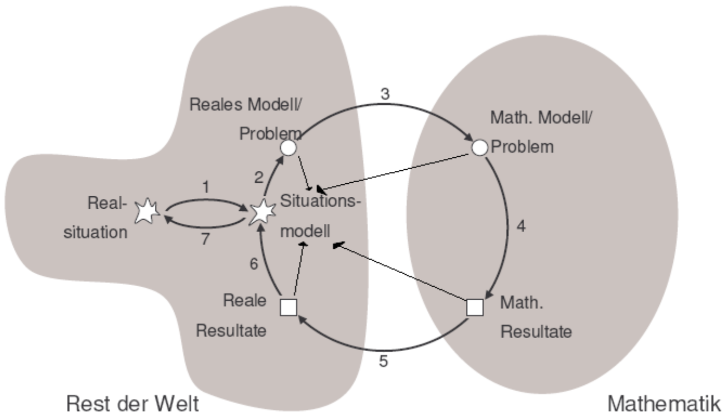


Abbildung 1

1. Konstruieren/Verstehen
   Aufgrund der vorliegenden Realsituation wird aufbauend auf das jeweilige Verständnis für das Realproblem ein Situationsmodell konstruiert.
2. Vereinfachen/Strukturieren
   Das konstruierte Situationsmodell wird durch Betrachtung und Diskussion der Parameter vereinfacht und in ein reales Modell, das für eine weiterführende Diskussion interessant erscheint, weiterentwickelt.

[1] bearbeitet Siller H.-St.

3. Mathematisieren
   Im diesem Schritt gelangt man durch eine mathematisch abstrakte Betrachtung der Parameter zu Gleichungen, Funktionen bzw. funktionalen Abhängigkeiten, die das vorliegende Realmodell als mathematisches Modell erscheinen lassen.
4. Mathematisch arbeiten
   Aufgrund der mathematischen Beschreibung kann man für das Mathematische Modell durch bekannte Strategien Lösungen berechnen.
5. Interpretieren
   Diese Lösungen müssen in Bezug auf die Anwendung interpretiert werden. Kann die Interpretation entsprechend erfolgen, kann man den nächsten Schritt durchführen.
6. Validieren
   Ist die Interpretation der Lösung gelungen, so muss diese noch auf Gültigkeit mit den vorhandenen „realen Lösungen“ überprüft werden.
7. Darlegen
   Sind die Schritte 1-6 gelungen, so kann die Lösung für das vorhandene Modell verwendet werden. Außerdem hat man die Gewissheit, dass das erstellte Modell den vorliegenden Sachverhalt gut genug beschrieben hat.

Es ist wichtig zu wissen, dass man zu jedem Zeitpunkt des Modellbildungsprozesses zum eigentlichen Ausgangspunkt, dem Situationsmodell, zurückkehren kann, um dieses entsprechend zu verändern.

Der Modellierungskreislauf ist in dieser Ausführung sehr komplex und kann, meiner Meinung nach, nur mit motivierten Schüler/innen in einem Wahlpflichtfach oder auch Leistungskurs durchgeführt werden. Um Modellbildung mit Schülern im Regelunterricht durchzuführen, ist es notwendig, gewisse Abstriche in den Anforderungen, d.h. Offenheit des Beispiels, durchzuführen, ohne jedoch den Kern der Aufgabe zu zerstören. Auf keinen Fall sollte man „eingekleidete“ Aufgaben als Modellierungsaufgaben „verkaufen“, da die Motivation und v.a. die Nachhaltigkeit bei Schüler/innen durch diesen Aufgabentyp nicht gesteigert wird. Ein typisches Beispiel einer solchen „eingekleideten“ Aufgabe ist das Folgende, in dem nur elementare Volums- und Oberflächenformeln in einem realen Kontext angewendet werden[2]:

„Ein Kessel besteht aus einer Halbkugel mit aufgesetztem Zylindermantel. Wie sind seine Maße zu wählen, damit der Kessel mit Deckel bei gegebener Oberfläche O ein möglichst großes Volumen hat?“ [7]

3. Modellieren in der Informatik

In der Informatik existieren zum Modellbildungsbegriff derzeit unterschiedliche Auffassungen, die sich jedoch grundsätzlich nicht widersprechen, da diese aufgrund der unterschiedlichen Programmierparadigmen (= Modellierparadigmen) entstanden sind [9]. G. Futschek definiert Modellbilden folgendermaßen: „... Zur Lösung eines Anwenderproblems entwirft der Informatiker zunächst ein Modell der Anwendung, (...) Das erste Modell wird in eine Reihe neuer Modelle umgeformt, die immer genauer und formaler werden, bis ein ablauffähiges Modell in einer bestimmten Programmiersprache erreicht ist...“ [6]

Futscheks Sichtweise ist, wie man anhand der Definition erkennen kann, typisch imperativisch-prozedural orientiert. Dies ist aufgrund der zeitlichen Einordnung dieser

[2] Im Gegensatz zum Schwimmbojenbeispiel [20]

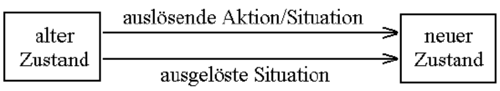
Definition offensichtlich, da damals fast ausschließlich imperativisch-prozedural gearbeitet wurde.

Weitere Programmierparadigmen, die augenblicklich von der Informatik vertreten werden, sind neben der imperativisch-prozeduralen Programmierung/Modellierung, die

- Funktionale Programmierung/Modellierung
  Zur Lösung eines Problems teilt der Informatiker ein Modell in kleine Teilprobleme die sich mit Hilfe von Funktionen leicht behandeln lassen. Diese Teilmodule kommunizieren miteinander und können mit Hilfe von weiteren Funktionen angesprochen werden. Ein Modell wird also mit Hilfe von Funktionen (Funktionsdefinitionen) und geschachtelten Funktionsaufrufen beschrieben und als Programm dargestellt.

- Objektorientierte Programmierung/Modellierung
  Zur Lösung eines realen Problems wird die vorliegende Umwelt mit Hilfe von Klassen beschrieben. Das vorliegende Modell ist also nichts anderes als ein Bauplan. Durch die konkreten Wert-Belegungen der Eigenschaften, d.h. durch Instanzierung, erhält man Objekte, mit denen das Modell für einen entsprechenden Sachverhalt beschrieben werden kann.

- Zustandsorientierte Programmierung/Modellierung



Modellbilden in der Informatik ist ein sehr weitgefasster Begriff, v.a. weil man im Prozess des Modellierens in der Informatik auch andere problemrelevante Themen und Fächer miteinbeziehen muss. Der Begriff Modellbilden könnte in der Informatik sogar als Charakterisierung des Faches Informatik herangezogen werden, so wie dies Alfred Aho und Jeffrey Ullman [21] tun: „Die Informatik ist eine Wissenschaft der Abstraktion und ihrer Mechanisierung. Es geht um die Konstruktion des passendsten Modells zur Repräsentation eines Problems und um die Erfindung der geeignetsten Rechenverfahren, die das Modell zur Lösung des Problems nutzen. Ein Modell ist umso besser, je klarer es ist und je besser sich damit rechnen lässt.“ Dies spiegelt sich auch im österreichischen Lehrplan für die 9. Schulstufe [24] zum Informatikunterricht wieder: „Durch Modellbildung, Formalisierung und Abstraktion leistet die Informatik einen wesentlichen Beitrag zur Auseinandersetzung mit Natur und Technik und führt zu einer verbesserten Entscheidungs-und Handlungskompetenz. […] Die so gewonnenen Erkenntnisse sind für die Modellierung eines Informatiksystems zu visualisieren und gegebenenfalls mit informatischen Verfahren zu abstrahieren. Die oftmals zyklische Vorgangsweise des Sammelns, Auswählens, Strukturierens, Abstrahierens, Auswertens und Interpretierens von Daten ist beim Problemlösen wegen ihrer zentralen Rolle immer wieder anzuwenden.“ Aber auch im Lehrplan für den weiterführenden Informatikunterricht (Schulstufe 10.–12.) liest man [25]: „Ein Schwerpunkt des Informatikunterrichts hat in der formalen Modellierung von Sachverhalten zu liegen, welche aus Analyse, Beschreibung in verschiedenen Darstellungsformen, Implementation, Überprüfung und Interpretation besteht.“

Der Modellierungsbegriff in der Informatik ist ebenso wie in der Mathematik, trotz unterschiedlicher Schwerpunktsetzungen (ähnlich dem Funktionsbegriff [5]), ein zentraler Begriff der helfen soll anwendungsorientierte Inhalte im Unterricht aufzugreifen und zu behandeln.

4. Modellieren – eine fundamentale Idee für fächerübergreifenden Unterricht in Mathematik und Informatik

Aufgrund der Ausführungen in den beiden vorigen Abschnitten kann man erkennen, dass Modellbilden eine zentrale Leitidee sowohl des Mathematikunterrichts als auch des Informatikunterrichts ist. Schlägt man in verschiedenen Katalogen zu fundamentalen Ideen nach, findet sich der Begriff Modellbilden/Modellieren ebenfalls. So schreibt z.B. Andreas Schwill, der die drei tragenden Säulen der Informatik in der Algorithmisierung, strukturierten Zerlegung und Sprache sieht [14]: „Sie (Anm.: die drei Säulen) stützen zugleich eine Sichtweise der Informatik, die bei Versuchen, ihr Wesen zu klären, häufig betont, jedoch selten weiter reflektiert wird, aber gleichwohl interessante Hinweise auf Unterschiede zur Mathematik liefert: die Methoden der informatischen Modellierung. Gelegentlich wird die Informatik in diesem Zusammenhang auch als die Wissenschaft von der Modellbildung bezeichnet."

Aber auch im Ideenkatalog von Humenberger/Reichel [10] findet man den Begriff wieder: „Der Ausgangspunkt beim Anwenden von Mathematik ist meistens eine „problemhaltige Situation" (kurz: „Problem"), und zwar nicht aus dem Reich der Mathematik, sondern eine aus dem „Rest der Welt". […] Wichtig ist allerdings, daß mit Mathematik dabei überhaupt etwas anzufangen ist, sie könnte ja auch völlig fehl am Platz sein, und zum jeweiligen Problemkreis gar nicht gehören. […] Meist ist es notwendig, den Bereich, aus dem das Problem stammt, schon relativ genau zu kennen, oder zumindest sich mit ihm vertraut zu machen, Fragen zu formulieren, Informationen zu beschaffen etc., […]."

Dies ist auch nicht weiter verwunderlich, wenn man die Behelfsdefinition von F. Schweiger betrachtet [13]:

Eine fundamentale Idee …

1. lässt Probleme auf unterschiedlichen Niveaus zu,
2. leitet in besonderer Weise zum Sprechen über Mathematik oder Informatik an,
3. erlaubt es, dass Lehrplaninhalte an ihr aufgehängt werden,
4. ist in der historischen Entwicklung aufzeigbar.

Wie man anhand der Ausführungen verschiedener Didaktiker erkennen kann, ist Modellbilden/Modellieren in den Unterrichtsfächern Mathematik bzw. Informatik als fundamentale Idee weitgehend akzeptiert. Meines Erachtens ist es jedoch auch notwendig, den übergreifenden und verbindenden Aspekt im nachstehenden Sinn zu definieren:

- Eine fundamentale Idee leitet in besonderer Weise zum Sprechen über fächerbindende bzw. fächerübergreifende Aspekte an und ist in der gemeinsamen Entwicklung der beiden Fächer aufzeigbar.

Ähnliche Aspekte kann man auch in dem folgenden Heymannschen Zitat erkennen [8]:

„Der allgemeinbildende Mathematikunterricht sollte Schüler dazu befähigen, Mathematik auch dort zu 'sehen', wo sie bei flüchtiger Betrachtung unsichtbar bleibt. Das Anwenden von Mathematik oder das 'mathematische Modellieren' bekommt damit eine besondere Bedeutung. Zwar läßt sich der Mathematikunterricht nicht durchweg von Anwendungsproblemen her und auf Anwendungsprobleme hin gestalten. Aber im herkömmlichen Mathematikunterricht kommen solche Probleme meist zu kurz: Sie dienen oft nur als 'motivierende' Einstiegsbeispiele oder werden in Gestalt realitätsfremder 'eingekleideter Aufgaben' an systematische Kurse angehängt." (Heymann, S. 48)

Gründe, warum man fächerübergreifenden Unterricht aus mehreren Gründen durchführen sollte, sind:

- Die Schüler/innen erfahren Analogien bei der Modellbildung unter verschiedenen fächerdifferenzierten Perspektiven.
- Die Gemeinsamkeiten und Fremdheiten innerhalb der Modelle werden deutlich.
- Wichtige Kompetenzen wie Beobachten, Beschreiben, Erläutern, Vergleichen oder Interpretieren werden gleichzeitig im fächerübergreifenden Unterricht und beim Modellbilden vereint.

Fächerübergreifendes Arbeiten ist, wie man an diesen Punkten exemplarisch erkennen kann, aus mehreren Gründen wichtig. Dies kann man auch bei A. Beckmann [1] nachlesen. Somit ist es offensichtlich, warum fächerübergreifendes Arbeiten in Mathematik und Informatik gefördert und intensiviert werden soll. Weitere Argumentationsgrundlagen könnten außerdem sein:

- Verbindung von Informationen aus verschiedenen Modellen und deren Interpretation für die reale Welt (vernetztes Denken). Dies ist auch bei F. Vester nachzulesen [22].
- der Schüler erfährt die Modellmethode als ein Mittel des fächerübergreifenden Lösens wissenschaftlicher Problemstellungen

Ein Thema, welches in den bisherigen Arbeiten zum Fächerübergreifenden Unterricht übersehen wurde, ist das Funktionale Modellieren. Dies möchte ich nun näher beleuchten.

5. Funktionale Modellierung – ein Thema für fächerübergreifenden Unterricht in Mathematik und Informatik

Durchläuft man den Prozess des Modellbildens, so ist es bei der Erstellung des mathematischen Modells notwendig, sich Gedanken über Abhängigkeiten der in das Modell einfließenden Parameter zu machen. Durch diese Beschreibung der Modellparameter gelangt man automatisch zu einer funktionalen Beschreibung des zugrundeliegenden Sachverhalts, der wiederum durch Funktionen ausgedrückt werden kann. Im Grund genommen heißt Modellbilden also „nichts anderes“ als „Arbeiten mit Funktionen/funktionalen Abhängigkeiten“. Der Begriff Funktionale Modellierung soll jedoch noch einmal die Bedeutung der funktionalen Abhängigkeit zentral in den Mittelpunkt rücken. Vor allem bei der Implementierung der mathematischen Problemlösung mit Hilfe von Technologie. Funktionale Programmierung ist ein in der Informatik sehr wichtiges und fundiertes Programmierparadigma mit dessen Hilfe man komplexe Probleme in Teilprobleme modularisiert, diese Teilprobleme mit Funktionen beschreibt und durch Verkettung dieser Funktionen zu einer Lösung des Gesamtproblems gelangt. Wesentliche Grundkompetenzen der mathematischen Bildung, Arbeiten mit Funktionen, Eigenschaften von Funktionen und Verketten von Funktionen, können durch diese Art der Implementierung im Unterricht behandelt werden. Schüler/innen aber auch Lehrer/innen in der Vorbereitung setzen sich mit Hilfe dieser Behandlung eines Problems mit einer völlig neuartigen Denkweise auseinander. Durch Gliederung in Black-Box- und White-Box-Phase [3] ist es möglich den Funktionsbegriff besser zu verstehen.

Durch eine niveaustufenorientierte Einteilung der Aufgaben ist es möglich, Schüler/innen behutsam zum „Programmieren“ mit Funktionen hinzuführen. Wichtig dabei ist, dass diese Art der Modellierung bereits in der Sekundarstufe I begonnen wird, dass Schüler/innen mit der manchmal etwas schwierigen Denkweise aufwachsen und bei „realen“ Problemen in der

Sekundarstufe II bereits mit diesem Denkschema und der Umsetzung vertraut sind. Werkzeuge mit denen man Funktionale Modellierung sehr gut umsetzen kann, sind:

- Tabellenkalkulation
  Die Eingabe einer Formel verlangt das Verständnis des Funktionsbegriffs. Eine Formel in Excel wird immer mittels funktionaler Abhängigkeiten (unbewusst) „programmiert", z.B. „=REST(GANZZAHL(ZUFALLSZAHL()*6);3)".
- Hand-Held-Rechner
  Der Vorteil eines Hand-Held-Rechners liegt in seiner Portabilität und der mathematischen Schreibweise bzw. Eingabe einer Funktion. Der Nachteil liegt in der begrenzten Berechenbarkeit von umfassenden Problemen und des grafischen Outputs. Fächerübergreifender Unterricht kann mit einem solchen Gerät bereits stark gefördert werden, auch um Schüler/innen zu zeigen, dass Informatik nicht ausschließlich über einen PC abgewickelt werden muss (s.o. Stichwort: Portabilität [17]).
- Computer-Algebra-System (CAS)
  Verwendet man CAS, die stark funktional arbeiten (z.B. Mathematica, Derive) so ist eine funktionale Umsetzung vorliegender realer Probleme mühelos möglich. Die wesentliche Voraussetzung für das Gelingen der funktionalen Modellierung liegt im Beherrschen der Syntax des jeweiligen CAS. Die Einbindung von CAS in den Informatikunterricht wäre wünschenswert. Schüler/innen könnte man zeigen, dass das Programmieren (eine für mich wesentliche Grundkompetenz) nicht auf Allzweck-Programmiersprachen, z.B. C++, JAVA, beschränkt ist.
- Funktionale Programmiersprachen
  Die „reinste" Form der Umsetzung von realen Problemen kann mit speziell für die funktionale Programmierung erstellten Programmiersprachen durchgeführt werden. Programme dafür gibt es inzwischen unzählige. Die gebräuchlichsten sind: LISP, Haskell (Freeware mit Hugs98) und DrScheme. Damit man mit solchen Sprachen arbeiten kann, ist es aber notwendig, sich intensiv mit ihrer Syntax auseinanderzusetzen, da diese sehr „karg" gehalten sind und im Vergleich mit den zuvor genannten Allzweck-Programmiersprachen schwierig zu erlernen sind.

Will man das fächerübergreifende Arbeiten in Mathematik und Informatik durch Funktionale Modellierung forcieren, ist es notwendig, sich einerseits des mathematischen Modellbildungsprozesses (Kap. 2) bewusst zu sein, andererseits den Modellbildungsbegriff der Informatik (Kap. 3) aufzugreifen. Die Begriffsdefinitionen in beiden Fächern widersprechen sich nicht, sondern ergänzen sich zu einem Gesamtbild, dass man gemäß dem folgenden Ablaufplan (vorteilhaft mit fließenden Übergängen) entwickeln könnte:

- Problembeschreibung und Mathematisierung (Ebene der Mathematik),
- Erstellung eine grafischen Repräsentation (Ebene der Mathematik/Informatik),
- Umsetzung mit Technologie (Ausführung, Test, Modifikation) (Ebene der Informatik)

beschreiben könnte. Einen wesentlichen Bereich stellt die grafische Repräsentation in diesem Ablaufplan dar. Dabei ist es, auch im Sinne informatischer Modellierung, notwendig sich Gedanken über eine geeignete grafische Aufbereitung des Modells zu machen. Die grafische Repräsentation von Funktionen beschäftigt Mathematiker schon lange [16]. Meiner Meinung nach ist es wichtig den Input-Output-Aspekt, d.h. wie verändern sich die Output-Werte bei entsprechender Änderung der Input-Werte, deutlich hervorzuheben. Dies hat bereits Vollrath herausgestrichen [23].

In der Informatik ist es üblich Implementierungen ebenfalls im Vorfeld grafisch darzustellen, um so die Erarbeitung des Codes zu erleichtern. Die üblichen Darstellungen sind Flussdiagramme (DIN 66001) oder Nassi-Shneiderman-Diagramme (DIN 66261) [11]. Mit Datenflussdiagrammen lässt sich die Darstellung von Funktionen realisieren [17]. Probleme ergeben sich jedoch bei der Behandlung von rekursiv definierten Funktionen, da Beschreibungen dieses Typs auch mit Datenflussdiagrammen nur schwer möglich sind. Ich habe mich für die Darstellung mit PROGRAPH-Diagrammen entschieden. Der Vorteil dieser Darstellung liegt in der leichten Erlernbarkeit der Symbole [16] (beschränkte auf einige wenige) und die verwendete „Bild in Bild"-Struktur als Metapher für Rekursionen. Der Aufbau eines solchen Diagramm-Typs gehorcht zwei Gesetzmäßigkeiten:

- Prinzip der einmaligen Zuweisung (d.h. ein zu Beginn determinierter Eingabewert bleibt „durch den Rest des funktionalen Systems" unverändert).
- jegliche Funktion liefert genau einen Wert zurück.

Bei genauer Thematisierung dieses Diagramm-Typs ist es auch leicht möglich mit Schüler/innen Funktionen in mehreren Variablen zu besprechen. Gerade bei der Analyse von Formeln könnte dies eine große Hilfe darstellen.

Der oben theoretisch angesprochene Ablaufplan und die theoretisch beschriebenen PROGRAPH-Diagramme sollen nun mittels Beispielen näher erläutert und verständlich gemacht werden.

6. Mögliche Beispiele

a. Ein mögliches Beispiel für die Unterstufe

Funktionales Modellieren kann bereits in der Unterstufe begonnen werden. Die Umsetzung mit Technologie (Hand-Held-Rechner bzw. Computer) soll auch für Schüler/innen einen zusätzlichen Anreiz geben, tiefer in dieses Thema einzutauchen. Möglicherweise kann man mit entsprechenden Aufgabenstellungen auch die Motivation der Schüler/innen hin zur Mathematik fördern.

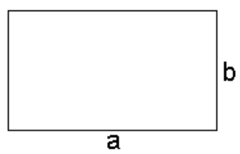
Aufgabenstellungen in der Unterstufe gibt es zur Genüge [18], [19]. Eine weitere mögliche Aufgabenstellung könnte lauten:

„Berechne den Flächeninhalt eines Rechtecks bei gleichbleibendem Umfang. Modelliere die Änderung des Flächeninhalts abhängig von den Seitenlängen!"

Für Schüler/innen der Sekundarstufe I stellt diese Aufgabenstellung sicher eine große Herausforderung dar. Sind Sie jedoch entsprechend offen gestellte Aufgaben gewohnt, bin ich mir sicher, dass sie mit Begeisterung zu arbeiten beginnen. Der im vorigen Kapitel erwähnte Ablaufplan sollte bereits auf dieser Stufe eingehalten werden.

1. Problembeschreibung und Mathematisierung:

Zunächst sollte man das Problem veranschaulichen, indem man ein Rechteck zeichnet und die konstanten, variablen und gesuchten Größen festlegt:



- Konstante Größe: u = const.
- Variable Größen: a, b
- Gesuchte Größe: A (a, b) = ?

Die Darstellung A(a, b) der gesuchten Größe macht deutlich, dass es sich hier um eine

Funktion mit 2 Variablen handelt, die man unschwer wie folgt auf eine Funktion mit einer Variablen, d.h. A(a) = ? bzw. A(b) = ? reduzieren kann:

$$A(a,b) = a \cdot b$$

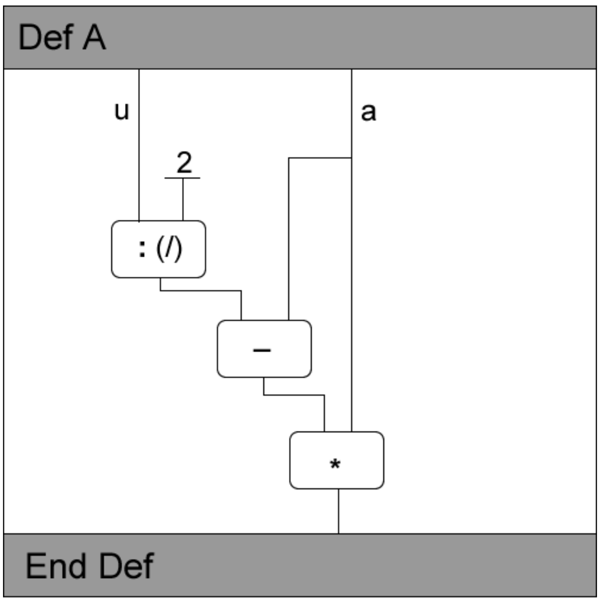
$$u = 2 \cdot (a + b) \Leftrightarrow b = \frac{u}{2} - a$$

$$\Rightarrow A(a) = a \cdot \left(\frac{u}{2} - a\right)$$

Somit ist ein mathematisches Modell (A(b) analog) gefunden, mit dessen Hilfe man das PROGRAPH-Diagramm zeichnen kann.

2. Erstellung einer grafischen Repräsentation:

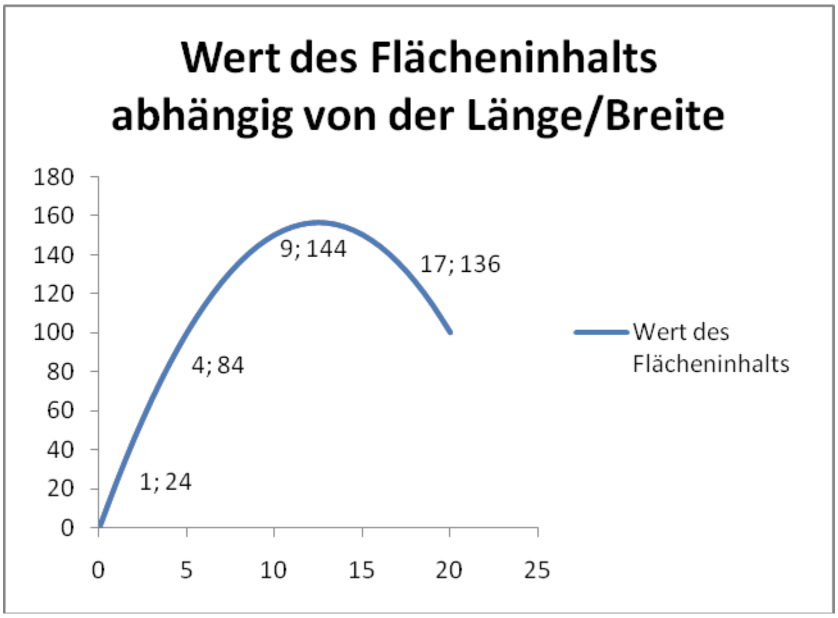
Eine Erklärung der hier verwendeten Symbole für das jeweilige Diagramm ist aus Platzgründen nicht möglich. Um die Symbole jedoch im Detail zu verstehen, kann eine Erklärung dieser in den Folien des Vortrages „Informatik mit einem Hand-Held – Funktionales Modellieren“ von der Web-Site der 11. Internationalen Schulmathematiker-Tagung abgerufen werden.
(http://www.univie.ac.at/mathematik_didaktik/2008_schulmathematik_tagung/vortraege/download/Funktionales_Modellieren.pdf)



3. **Umsetzung mit Technologie:**

Die Umsetzung dieses Beispiels in der Unterstufe gelingt, wenn man noch keinen Hand-Held-Rechner verwendet, am besten mit einem Tabellenkalkulationsprogramm. Ich habe in diesem Fall Excel verwendet:

a	A(a)
0	0
1	24
2	46
3	66
4	84
5	100
6	114
7	126
8	136
9	144
10	150
11	154
12	156
13	156
14	154
15	150
16	144
17	136
18	126
19	114
20	100



b. Ein mögliches Beispiel für die Oberstufe

In der Sekundarstufe II ist Funktionales Modellieren in fast allen Bereichen des Lehrplans möglich. Der/Die Lehrer/in ist dazu aufgefordert seiner/ihrer Fantasie freien Lauf zu lassen und entsprechende funktionale Zusammenhänge zu erkennen. Viele Beispiele, auch für fächerübergreifenden Unterricht Mathematik/Informatik finden sich in „Basics in Functional Modelling“ [4]. Wie ich selbst mit meinen Student/innen erleben durfte, bereitet das Auffinden funktionaler Zusammenhänge an Figuren des alltäglichen Lebens viel Freude, fördert die Kreativität und reizt zu Neuem. Eine mögliche Aufgabenstellung für die 11. Schulstufe könnte folgendermaßen aussehen:

„Das in der Figur dargestellte Weinglas soll als Rotationskörper modelliert werden. Wie könnte man die erzeugende Kurve (das „Profil“) möglichst geschickt (was von der Lage des Weinglases zum Koordinatensystem abhängt) durch möglichst einfache und doch gut passende Funktionen (stückweise) modellieren? Welche Verbesserungen für dieses Modell könnte man durchführen?“

Natürlich ist es auch in der Sekundarstufe II notwendig, den Ablaufplan einzuhalten.

1. **Problembeschreibung und Mathematisierung:**

Durch entsprechendes Drehen des Bildes oder Kippen des Koordinatensystems ist es möglich das Profil des Weinglases mit Funktionen zu beschreiben. Für die leichtere Handhabung mit einem CAS ist es vorteilhaft das Bild zu kippen, also eine Drehung um 90° (nach rechts) auszuführen (siehe Abbildung 2).

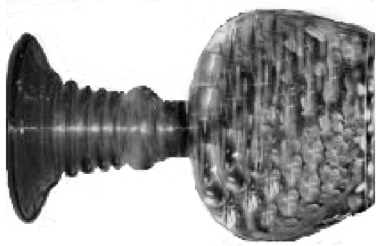


Abbildung 2

Im vorliegenden Bild kann man die einzelnen Teile des Glases durch Polynomfunktionen beschreiben:

- Fuß des Glases: Polynomfunktion 3. Grades $f(x) = ax^3 + bx^2 + cx + d$
- Stiel des Glases: Polynomfunktion 2. Grades $g(x) = ex^2 + fx + g$
- Knopf am Stiel: Polynomfunktion 3. Grades: $h(x) = hx^3 + ix^2 + jx + k$
- Verbindungsteil Stiel/Füllbehälter: Lineare Funktion i(x) = lx + m
- Füllbehälter: Polynomfunktion 4. Grades $j(x) = nx^4 + px^3 + qx^2 + rx + s$

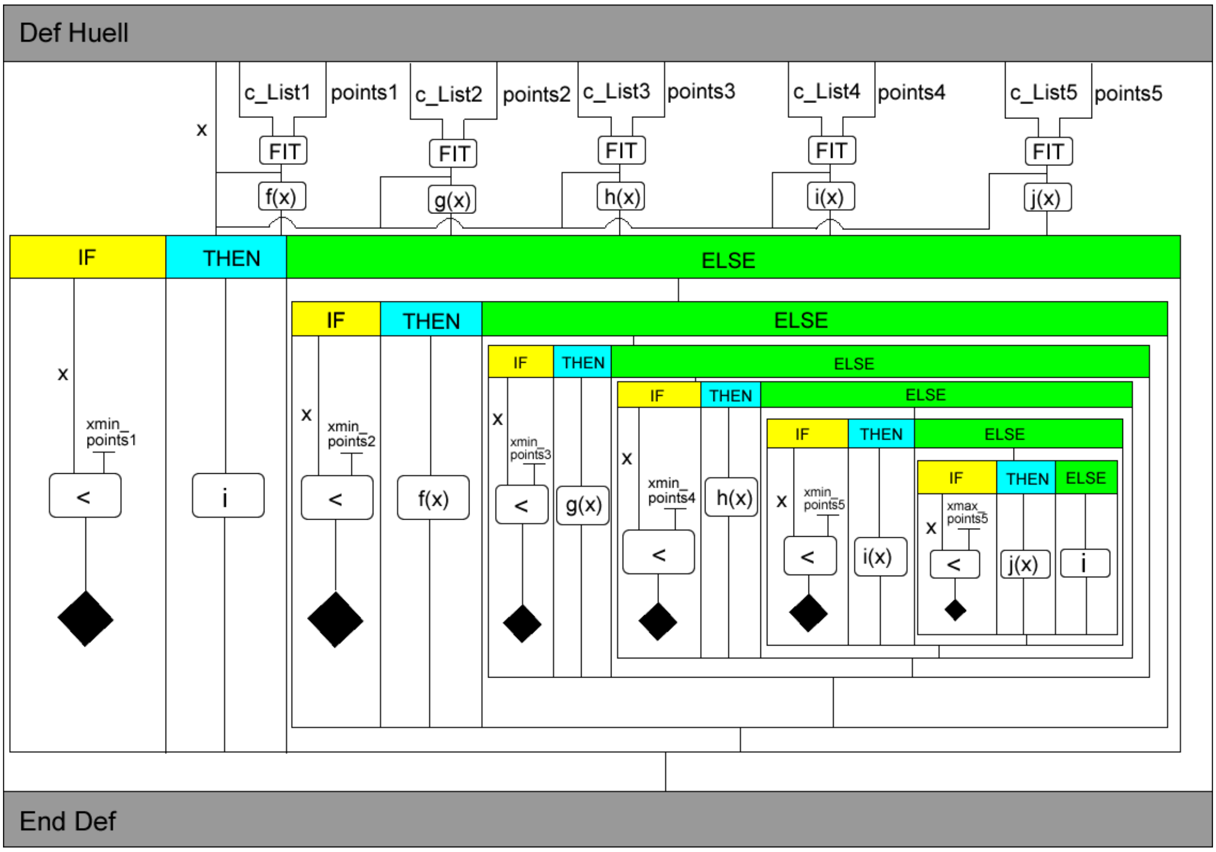
Durch Zusammensetzen der einzelnen Funktionen ist es möglich die Hüllkurve des Glass zu beschreiben: huell(x)=
$$\begin{cases} nicht\ definiert\ für\ x < x_1 \\ f(x)\ für\ x_1 \le x < x_2 \\ g(x)\ für\ x_2 \le x < x_3 \\ h(x)\ für\ x_3 \le x < x_4 \\ i(x)\ für\ x_4 \le x < x_5 \\ j(x)\ für\ x_5 \le x \le x_6 \\ nicht\ definiert\ für\ x > x_6 \end{cases}$$

Diese Hüllkurve kann man durch Multiplikation mit einem „Drehvektor“ um die x-Achse rotieren lassen und durch eine Darstellung im räumlichen Koordinatensystem zeichnen.

Da solche Interpolationspolynome sehr oft mit „Ausreißern“ zu kämpfen haben und auch der Übergang in den jeweiligen Datenpunkten x_1, … x_5 mathematisch nicht sauber vollzogen wird, kann man überlegen, dass die oben angeführte Hüllkurve durch Modellieren mit kubischen Splines exakt möglich ist. Der Vorteil kubischer Splines ist darin zu sehen, dass in den „Übergangsdatenpunkten“ die Steigung und die Krümmung (ähnlich dem Anlegen einer Schnur) gleich sind. Damit auch die Randpunkte, mathematisch ordentlich definiert sind, setzt man hier die Krümmung 0.

2. **Erstellung einer grafischen Repräsentation:**

Die grafische Darstellung für diesen Sachverhalt gestaltet sich etwas komplizierter, da man sehr viele Bedingungen – was anhand der Definition der Funktion huell(x) ersichtlich wird – berücksichtigen muss. Betrachtet man den vorliegenden Sachverhalt genauer, so ist die in der Grafik vorliegende Struktur einsichtig.



Anmerkungen zum Diagramm:

- c_List# := dieser Eintrag beinhaltet jeweils die Koeffizienten der zugehörigen Funktion wie vorhin angeführt. Für f(x) wäre dies z.B. die Koeffizienten-Liste c_List1 = {a, b, c, d}.
- xmin_points# := bezeichnet das jeweilige Minimum der der x-Werte der nummerierten Datenmatrizen. Für points1 wäre dies z.B. xmin_points1 = -1,139 analog dazu ist der Befehl xmax_points# gewählt, der den maximalen x-Wert der Datenmatrix wiedergibt.

3. Umsetzung mit Technologie (Ausführung, Test, Modifikation):

- Umsetzung mit Interpolationspolynomen (Fitten einer Kurve)

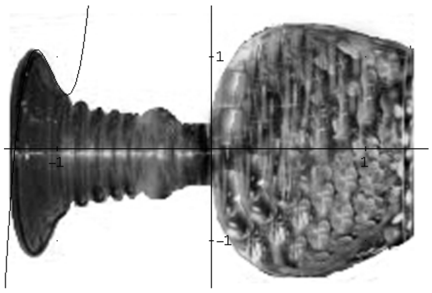
Um das Glas mit Polynomen zu modellieren, wende ich zunächst den integrierten FIT-Befehl (von DERIVE) auf selbst definierte Datenmatrizen an. Diese Punkte kann man entweder in der üblichen Form eintippen oder man tippt im 2D-Graph-Fenster [hcross, vcross]= in der Eingabezeile ein und drückt Enter. Somit erhält man im Algebra-Fenster Punktepaare, die man mit F3 in eine „größere" Datenmatrix übernehmen und diese dann entsprechend definieren kann. Im Folgenden werden die Schritte entsprechend 1. und 2. von oben exemplarisch dargestellt – Definition der Datenmatrix, Definition der Funktion, Output der Funktion und grafische Darstellung mit und ohne Hintergrundbild – Codierung der abschnittsweise definierten Funktion und Erzeugung des Rotationskörpers. Die Vereinfachung am Stiel (quadratische Funktion anstatt periodischer Funktion) könnte mit Schüler/innen weiter diskutiert werden. Für eine weiterführende Diskussion wäre auch die beim

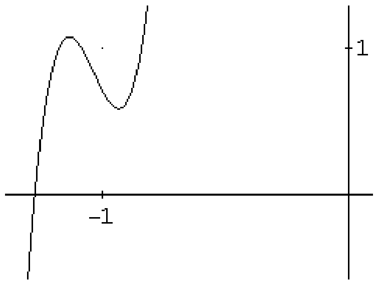
FITTEN dahintersteckende Methode der kleinsten Quadrate oder ein anderes Interpolationsverfahren, z.B. die Newtonschen Interpolationspolynome, besser bekannt als „Dividierten Differenzen“, welche man Schüler/innen als Vorstufe zum FITTEN von Funktionen vermitteln könnte, wünschenswert.

#1: $$\text{points1} := \begin{bmatrix} -1.139 & 1.074 \\ -1.069 & 0.957 \\ -1.032 & 0.829 \\ -1.006 & 0.734 \\ -0.981 & 0.659 \end{bmatrix}$$

#2: $$f(x) := \text{FIT}\left(\left[x,\ a\cdot x^3 + b\cdot x^2 + c\cdot x + d\right],\ \text{points1}\right)$$

#3: $$f(x) := \frac{650995395831091900000\cdot x^3}{5323275762309994881} + \frac{672568937858683084600\cdot x^2}{1774425254103331627} + \frac{206510757454314384878 81\cdot x}{53232757623099948810} + \frac{117033922900165015007 2927}{88721262705166581350 00}$$

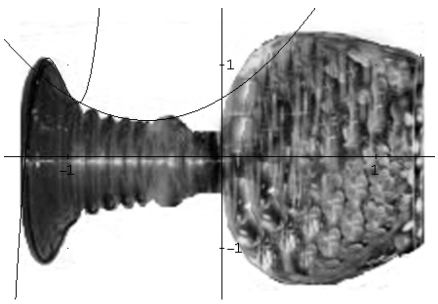




#4: $$\text{points2} := \begin{bmatrix} -0.981 & 0.659 \\ -0.943 & 0.595 \\ -0.841 & 0.521 \\ -0.728 & 0.457 \\ -0.627 & 0.436 \\ -0.449 & 0.383 \end{bmatrix}$$

#5: $$g(x) := \text{FIT}\left(\left[x,\ a\cdot x^2 + b\cdot x + c\right],\ \text{points2}\right)$$

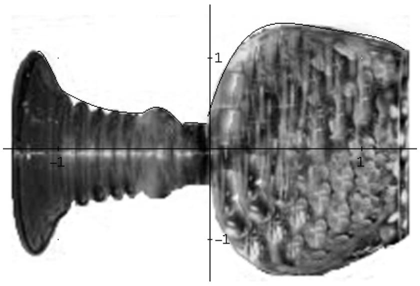
#6: $$g(x) := \frac{134385847132050\cdot x^2}{156674901177557} + \frac{2343635399159337\cdot x}{3133498023551140} + \frac{431995210004851587}{783374505887785000}$$

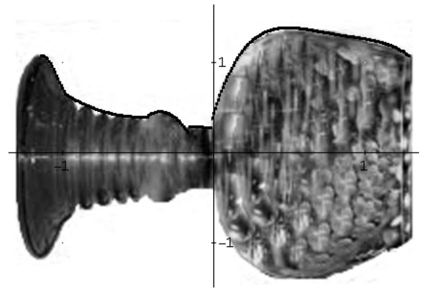


usw.

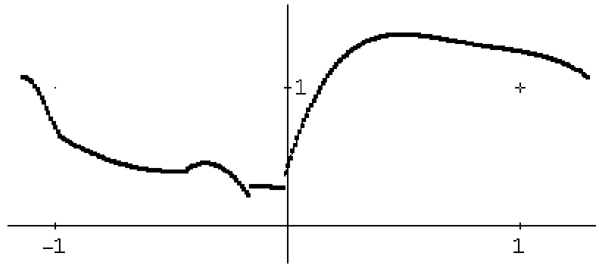
```
Huell(x) :=
  If x < -1.139
     ?
     If x ≤ -0.981
        f(x)
        If x ≤ -0.449
#16:       g(x)
           If x ≤ -0.169
              h(x)
              If x ≤ -0.018
                 i(x)
                 If x ≤ 1.291
                    j(x)
```

```
Huell(x) :=
  If x < -1.139
     ¿
     If x ≤ -0.981
        f(x)
        If x ≤ -0.449
#16:       g(x)
           If x ≤ -0.169
              h(x)
              If x ≤ -0.018
                 i(x)
                 If x ≤ 1.291
                    j(x)
```





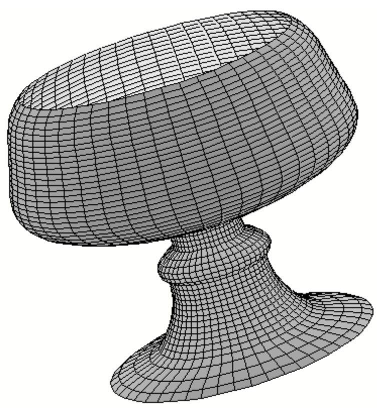
Bzw. ohne Hintergrundbild



Ohne die Hintergrundgrafik kann man schön erkennen, dass die Kurve nicht geschlossen dargestellt wird. Für die Rotation um die x-Achse stört dies jedoch nicht weiter, da nur die Datenpunkte dazwischen berücksichtigt werden. Das Bild „fließt“ also – im Gegensatz zu Grafikprogrammen – nicht aus. Der Rotationskörper wird durch entsprechende Großkreise in der y-z-Ebene beschrieben:

```
#17:  [t, Huell(t)·COS(s), Huell(t)·SIN(s)]
```

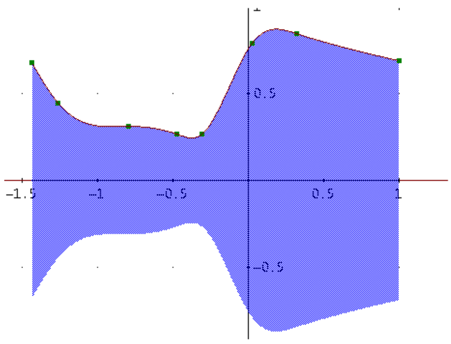
Damit erhalten wir das nachmodellierte Weinglas, wie im nachstehenden Bild erkennbar:



- Umsetzung mit Splines

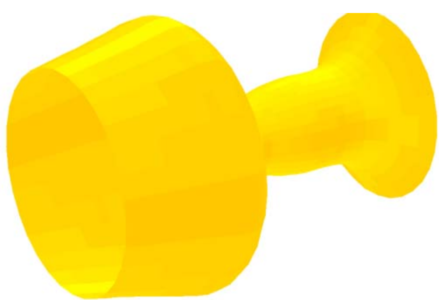
Eine weitere (bessere) Möglichkeit um den Übergang in den Datenpunkten besser zu beschreiben wäre die Methode der kubischen Splines. Hier füge ich aus Platzgründen nur die Definition der Bedingungen und das Endergebnis ein. Anzumerken ist, dass ich hier ein etwas anderes Glas modelliert habe, als das obige:

```
#1:   CaseMode := Sensitive

#2:   InputMode := Word

           ⎡ -1.435   0.677 ⎤
           ⎢  -1.26   0.444 ⎥
           ⎢ -0.794   0.311 ⎥
           ⎢ -0.474   0.266 ⎥
#3:   A := ⎢ -0.307   0.266 ⎥
           ⎢  0.025   0.788 ⎥
           ⎢   0.32   0.844 ⎥
           ⎣      1   0.688 ⎦

#4:   Stuetz_Bed := [P1(-1.435) = 0.677, P1(-1.26) = 0.444, P2(-1.26) = 0.444, P2(-0.794) = 0.311, P3(-0.794) = 0.311, P3(-0.474) = 0.266, P4(-0.474)
        0.266, P4(-0.307) = 0.266, P5(-0.307) = 0.266, P5(0.025) = 0.788, P6(0.025) = 0.788, P6(0.32) = 0.844, P7(0.32) = 0.844, P7(1) = 0.688]

#5:   Steig_Bed := [P1'(-1.26) = P2'(-1.26), P2'(-0.794) = P3'(-0.794), P3'(-0.474) = P4'(-0.474), P4'(-0.307) = P5'(-0.307), P5'(0.025) = P6'(0.025),
        P6'(0.32) = P7'(0.32)]

#6:   Kruemm_Bed := [P1''(-1.26) = P2''(-1.26), P2''(-0.794) = P3''(-0.794), P3''(-0.474) = P4''(-0.474), P4''(-0.307) = P5''(-0.307), P5''(0.025) =
        P6''(0.025), P6''(0.32) = P7''(0.32), P1''(-1.435) = 0, P7''(1) = 0]

                                     2         3
#7:   P1(x) := a1 + b1·x + c1·x  + d1·x

                                     2         3
#8:   P2(x) := a2 + b2·x + c2·x  + d2·x

                                     2         3
#9:   P3(x) := a3 + b3·x + c3·x  + d3·x

                                     -         -
#10:  P4(x) := a4 + b4·x + c4·x  + d4·x

                                     2         3
#11:  P5(x) := a5 + b5·x + c5·x  + d5·x

                                     2         3
#12:  P6(x) := a6 + b6·x + c6·x  + d6·x

                                     2         3
#13:  P7(x) := a7 + b7·x + c7·x  + d7·x

#14:  APPEND(Stuetz_Bed, Steig_Bed, Kruemm_Bed)

#15:  SOLVE(APPEND(Stuetz_Bed, Steig_Bed, Kruemm_Bed), [a1, a2, a3, a4, a5, a6, a7, b1, b2, b3, b4, b5, b6, b7, c1, c2, c3, c4, c5, c6, c7, d1, d2, d
        d4, d5, d6, d7])

#52:  f(x) := P1(x)·χ(-1.435, x, -1.26) + P2(x)·χ(-1.26, x, -0.794) + P3(x)·χ(-0.794, x, -0.474) + P4(x)·χ(-0.474, x, -0.307) + P5(x)·χ(-0.307, x, 0.0
        P6(x)·χ(0.025, x, 0.32) + P7(x)·χ(0.32, x, 1)

#53:  |y| ≤ |f(x)|

#54:  [A, f(x)]
```




```
#55:  [t, f(t)·COS(S), f(t)·SIN(S)]
```

In anderen CAS ist es möglich die hier (in DERIVE) definierten kubischen Splines als implementierte Funktionen aufzurufen. Ein Beispiel dafür wäre MAPLE. Wie man jedoch am grafischen Output von DERIVE sieht ist das Ergebnis durchaus ansprechend!



Literatur

[1] Beckmann, A.: Fächerübergreifender Mathematikunterricht, Teil 1: Ein Modell, Ziele und fachspezifische Diskussion, Verlag Franzbecker, Hildesheim, 2003

[2] Blum, W.; Leiß, D.: Modellieren im Unterricht mit der "Tanken"-Aufgabe, in: Mathematik lehren, H. 128, S. 18-21, 2005

[3] Buchberger, B.: Teaching Math by Math software: The White Box /Black Box Principle, Paper of the RISC-Institute of the Johannes Kepler Universität Linz, Linz, 1992

[4] Fuchs, K.J.; Siller, H.-St.; Vásárhelyi, E.: Informatics with CASIO CP 300+, Part II: Basics in Functional Programming, published by CASIO Europe GmbH, Budapest, 2008

[5] Fuchs, K.J.: Die Funktion – Basiselement der Informatik, Beiträge zum Mathematikunterricht, Verlag Franzbecker, Hildesheim, 2008

[6] Futschek, G.: Informatik als Wissenschaft. In: Reiter, A.; Rieder, A.: Didaktik der Informatik, Jugend und Volk, Wien, 1990

[7] Geretschläger, R.; Griesel, H.; Postel, H: Elemente der Mathematik 7, Dorner Verlag, Wien, 2006

[8] Heymann, Werner: Mathematikunterricht und sein (möglicher) Beitrag zur Allgemeinbildung. – In: Pädagogik (1997) H. 1, S. 46–49

[9] Hubwieser, P.: Didaktik der Informatik – Grundlagen, Konzepte, Beispiele, Springer Verlag, Berlin, 2007

[10] Humenberger, J.; Reichel, H.-C.: Fundamentale Ideen der Angewandten Mathematik und ihre Umsetzung im Unterricht, B.I. Wissenschaftsverlag, Mannheim, 1995

[11] Nassi, I.; Shneiderman, B.: Flowchart technique for structured programming, ACM SIGPLAN Notices, vol. 8, S. 12–26, 1973

[12] Peschek, W.; Heugl, H.: Standards für die mathematischen Fähigkeiten österreichischer Schülerinnen und Schüler am Ende der 8. Schulstufe, Version 4/07, Hrsg.: Institut für Didaktik der Mathematik – Österreichisches Kompetenzzentrum für Mathematikdidaktik – Fakultät für interdisziplinäre Forschung und Fortbildung Alpen-Adria-Universität Klagenfurt, Klagenfurt, 2007

[13] Schweiger, F.: Fundamentale Ideen - Eine geistesgeschichtliche Studie zur Mathematikdidaktik. In: JMD, Jg. 13, H. 2/3, S. 199-214, 1992

[14] Schwill, A.: Fundamentale Ideen in Mathematik und Informatik, In: Hischer, Horst & Weiß, Michael (Hrsg.): Fundamentale Ideen – Erörterungen zur Zielorientierung eines künftigen Mathematikunterrichts unter Berücksichtigung der Informatik. Bericht über die 12. Arbeitstagung des Arbeitskreises "Mathematikunterricht und Informatik" in der Gesellschaft für Didaktik der Mathematik e. V. vom 23. bis 26. September 1994 in Wolfenbüttel. Hildesheim: Franzbecker, 1995.

[15] Siller, H.-St.: Bildungsstandards im Fach Mathematik – Das mathematische Kompetenzmodell – eine (kompakte) Handreichung für Lehrer/innen (www.phsalzburg.at/ahs)

[16] Siller, H.-St.: Über die Bedeutung der grafischen Repräsentation beim Funktionalen Modellieren, Beiträge zum Mathematikunterricht, Verlag Franzbecker, Hildesheim, 2008

[17] Siller, H.-St.; Fuchs, K.J.; Vásárhelyi, E.: Funktionales Modellieren mit einem Hand-Held, Beiträge zum Mathematikunterricht, Verlag Franzbecker, Hildesheim, 2008

[18] Siller, H.-St.: Functional Modeling – a creative way in modeling, paper accepted for the 11th ICME, DG 9, Mexico, 2008

[19] Siller, H.-St.: PROGRAPH diagrams – a new old system for teaching Functional Modeling, paper accepted for the 11th ICME, TSG 22, Mexico, 2008

[20] Siller, H.-St.: Computerunterstützter Analysis-Unterricht – Auf Mathematica basierende Lerneinheiten zur Differential- und Integralrechnung mit M@thDesktop, VDM, 2008

[21] Aho, A.V.; Ullman, J.D.: Foundations of Computer Science (Principles of Computer Science Series), W.H. Freeman & Company, 1995.

[22] Vester, F.: Unsere Welt - ein vernetztes System, dtv, 2002

[23] Vollrath, H.J.: Funktionales Denken. In: Journal für Mathematikdidaktik, S. 3 – 37, 1989

[24] Österr. AHS-Lehrplan Informatik für die 9. Schulstufe, BMUKK, 2004 (www.bmukk.gv.at)

[25] Österr. AHS-Lehrplan Informatik für die 10.–12. Schulstufe, BMUKK, 2004 (www.bmukk.gv.at)

Anschrift des Autors:

Univ.Ass. Mag. Dr. Hans-Stefan Siller
IFFB – Abteilung für Didaktik der Mathematik und Informatik
Universität Salzburg
Hellbrunnerstraße 34
A – 5020 Salzburg

Mail: hans-stefan.siller@sbg.ac.at