Hochschule Wismar Gottlob Frege Centre

Proceedings
14. Workshop Mathematik
in ingenieurwissenschaftlichen Studiengängen

Erlangen

September 2017

Heft 01 / 2017

Wismarer Frege-Reihe Wismar Frege Series

Das **Gottlob-Frege-Zentrum** wurde am 7.11. 2000 an der Hochschule Wismar gegründet. Seine Mitglieder setzen sich für eine wissenschaftlich begründete, praxisorientierte, moderne und international ausgerichtete Ausbildung in den mathematisch-naturwissenschaftlichen Grundlagendisziplinen ein.

Weitere Informationen zum Gottlob-Frege-Zentrum finden Sie auf der Netz-Seite

http://www.hs-wismar.de/frege

ISSN 1862-1767
ISBN 978-3-942100-55-7

Printed in Germany

Inhaltsverzeichnis WFR Heft 01/2017

Didaktische und methodische Konzepte

Mathematiklehre mit Softwarenutzung

Theorie und Praxis

Anhang

Vorwort der Veranstalter

Die Workshop-Reihe „Mathematik in ingenieur-wissenschaftlichen Studiengängen (Ingmath)“ ist eine Plattform für Projekte und dient dem hochschulübergreifenden Austausch u.a. über mathematische Methoden und hochschuldidaktische Fragestellungen.
Im Jahr 2017 fand der 14. Ingmath-Workshop erstmals an der Friedrich-Alexander-Universität Erlangen-Nürnberg in Erlangen statt. Auf dem Südgelände der Universität trafen sich dazu im Department der Mathematik am 18. September Kolleginnen und Kollegen aus allen Teilen Deutschlands.
Thematisch hatte der Workshop dieses Jahr den Schwerpunkt „Digitalisierung in der Lehre“. Die Hauptvorträge „Mathematik im Web-Browser mit JSXGraph“ und „Ein Online-System für Hausaufgaben zur Ingenieurmathematik - Chancen und Herausforderungen“ lieferten einen passenden Grundstock für den kollegialen Austausch und die anregenden Diskussionen sowohl nach den Vorträgen als auch in den Pausen. Weitere Vorträge auch mit mathematischen Schwerpunkten sorgten für eine Vielfalt und eine willkommene Abwechslung.
Mit den vorliegenden Proceedings sind die interessanten und abwechslungsreichen Beiträge, Diskussionen und Ideen nachlesbar. Für die Publikation in der Wismarer Frege-Reihe sind wir daher besonders dankbar.
Im Frühjahr 2019 findet der 15. Ingmath-Workshop voraussichtlich in Rostock-Warnemünde statt.

Wir freuen uns auf ein Wiedersehen und neue Impulse und Ideen.

Wigand Rathmann
Nicolai von Schroeders

Grußwort des Herausgebers

Im Namen aller Teilnehmer danke ich den Organisatoren für eine wieder sehr gut organisierte und attraktive Tagung in der Universitätsstadt Erlangen. Am Vortag des Workshops gab es nachmittags eine interessante Führung durch das Stadtzentrum von Erlangen. Aufgrund verspäteter Anreise habe ich diese Veranstaltung leider verpassen müssen.
Am Abend trafen wir uns in der Gaststätte „Kaiser Wilhelm" zu einem gemütlichen Abendessen mit anregenden Diskussionen zu aktuellen Problemen der Mathematikausbildung im Ingenieurstudium an den Hochschulen verschiedener Bundesländer.
Ein Teil der Teilnehmer übernachtete wie ich im Hotel Grille etwas abseits des Zentrums. Mit guter Kondition ausgestattet marschierten wir dann nachts bei bester Laune bis zum Quartier. Dieser Vorgang sollte am nächsten Morgen in umgekehrter Richtung und mit etwas anderem Ziel (Department Mathematik der Universität) erfolgreich wiederholt werden. Der leichte Nieselregen konnte unsere Laune kaum trüben. Zudem wurde „Professor Google" auf dem Smartphone immer wieder zur Navigation befragt. Die intensiven Diskussionen raubten uns aber an einem entscheidenden Punkt die Aufmerksamkeit. Der dadurch erforderliche Umweg brachte uns in Zeitnot und ins Schwitzen. Mit etwa 10 Minuten Verspätung und durchgeschwitzt platzten wir mit schlechtem Gewissen in den ersten Vortrag. Dafür möchte ich mich bei den Veranstaltern und den übrigen Teilnehmern entschuldigen. Es tat gut, dass wir nicht noch zusätzlich mit Vorwürfen bedacht wurden.
Während die IngMath-Workshops zunächst im Norden Deutschlands starteten und auch von den unterschiedlichen Voraussetzungen im Osten und Westen nach der Wiedervereinigung geprägt waren, sind die Austragungsorte später eher in den Süden gewandert und haben mit Erlangen wohl diesbezüglich einen vorläufigen Wendepunkt erreicht. Da die Teilnehmer neben dem „Stammpersonal" verstärkt aus der umliegenden Region kommen, ergibt sich für uns ein guter Überblick über die Situation in Deutschland insgesamt. Wie nicht anders zu erwarten, spielen aber neben einigen regionalen Besonderheiten überall die gleichen bzw. ganz ähnliche Probleme eine Rolle. Die Ost-West-Unterschiede verblassen zunehmend und lassen zum Teil auch Nord-Süd-Unterschiede deutlicher hervortreten. Nun geht es wohl wieder zurück in den Norden (siehe Vorwort). Ich hoffe, dass auch andere Teile Deutschlands dort wieder vertreten sein werden.

Auf eine weitere gedeihliche Zusammenarbeit!
Ihr Dieter Schott

Programm

Uhrzeit	Programmpunkt	Personen
09:00	Begrüßung durch den Studiendekan der Technischen Fakultät der FAU	Prof. Dr.-Ing. habil. Kai Willner, FAU Erlangen-Nürnberg
09:15	Mathematik im Web-Browser mit JSXGraph	Prof. Dr. Alfred Wassermann, Universität Bayreuth
10:00	"Julia" in ingenieurwissenschaftlichen Studiengängen	Prof. Dr. Jürgen Vorloeper, Hochschule Ruhr West
10:30	Blended-Learning Konzept für Direktstudenten -- Versuch, die Studenten nicht aus dem Hörsaal "zu verlieren"	Prof. Dr. Petra Leitert, Hochschule Wismar
11:00	**KAFFEEPAUSE**	
11:30	Der Ableitungsbegriff der Analysis im Lichte der Geschichte versehen mit didaktischen und philosophischen Randglossen	Prof. Dr. Dieter Schott, Hochschule Wismar
12:00	Visual Complex Analysis -- die Anschaulichkeit der Geometrie nutzen!	Prof. Dr. Thomas Risse, Hochschule Bremen, City University of Applied Sciences
12:30	**MITTAGSPAUSE**	
13:30	Ein Online-System für Hausaufgaben zur Ingenieurmathematik - Chancen und Herausforderungen	Mikko Vasko, Hochschule Karlsruhe - Technik und Wirtschaft
14:15	Integration von MATLAB® in die Übungen zur Mathematik für Ingenieure	Prof. Dr. Claudia Frohn-Schauf und Prof. Dr. Joachim Fulst, Hochschule Bochum – University of Applied Sciences

14:45	**POSTER + KAFFEE**	
	Optimale Steuerung von Flachwasser-kanälen	Dr. Wigand Rathmann, FAU Erlangen-Nürnberg
	MatInEE: Konzept zur Fehleranalyse und -typisierung in einer Erstsemester-klausur zur Linearen Algebra eines in-genieurwissenschaftlichen Bachelor-studiengangs	Katharina Görtler, FAU Erlangen-Nürnberg
15:15	MathWeb - Interaktive Online Aufga-ben und Demonstrationen	Prof. Dr. Klaus Giebermann, Hochschule Ruhr West
15:45	Integrationstechniken mal anders – Er-fahrungen aus einem Flipped-Classroom-Ansatz	Nicolai von Schroeders, FAU Erlangen-Nürnberg
16:15	Ende: Eine Viertelstunde für Feedback	Dr. Wigand Rathmann, FAU Erlangen-Nürnberg

Hinweis:

Die Beiträge in den Proceedings sind nur Vortragsauszüge und tragen zum Teil einen anderen Titel. Einige Vortragende haben keinen Beitrag eingereicht.

Dieter Schott

Der Ableitungsbegriff der Analysis im Lichte der Geschichte versehen mit didaktischen und philosophischen Randglossen

Zusammenfassung: Angefangen bei der Fluxionsrechnung von Newton und dem Differentialkalkül von Leibniz ergeben sich immer wieder neue Einsichten und Verallgemeinerungsmöglichkeiten für den Ableitungsbegriff in der Analysis, die auch das Feld der Anwendungen schrittweise erweitern. Die Begriffsentwicklung wird im Vortrag didaktisch und philosophisch untersetzt, wobei u.a. die Beziehung von Ableitung und Differential und ihren Abkömmlingen eine Rolle spielt. Auch die Bezeichnungsweise erweist sich als grundlegend für die Effektivität entsprechender Theorien. Dabei kommen solche Prozesse wie Intuition – Heuristik, Verallgemeinerung – Abstraktion oder auch Kalkül – Formalisierung – Mnemotechnik vor. Natürlich werden auch Aspekte des Grenzwertbegriffes und Fragen nach dem potentiell oder aktual unendlich Kleinen berührt.

1. Einführung

Begriffe entwickeln sich oft im Zusammenhang mit Anschauungen und Vorstellungen. So sagte der Philosoph *Kant* (1724-1894): „Begriffe ohne Anschauung sind leer. Anschauung ohne Begriffe ist blind.“
Viele mathematische Begriffe haben eine lange Vorgeschichte und bedürfen weiterer Begriffe, auf die sie sich beziehen. Verändert sich in diesem Begriffsnetz eine Komponente, so hat das auch Auswirkungen auf die anderen Komponenten. Als die Infinitesimalrechnung entstand, hatte man zunächst nur vage Vorstellungen vom Grenzwertbegriff für Funktionen, der seine heutige Form im Wesentlichen durch *Cauchy* (1789-1857) erhielt. Aufbauend auf dem Grenzwertbegriff für Folgen oder über die sogenannte ε-δ-Definition direkt kann man den Grenzwert für Funktionen einer Veränderlichen (an einer vorgegebenen Stelle) definieren. Dabei geht das unendlich Kleine (Infinitesimale, Differential) als potentiell unendlich Kleines ein.
Später hat der Grenzwertbegriff in der Analysis viele Modifizierungen und Verallgemeinerungen erfahren (z.B. in Limesräumen, in Banachräumen oder in Distributionenräumen in der Funktionalanalysis). In der Nichtstandardanalysis konnte das unendlich Kleine auch als aktual unendlich Kleines gefasst werden. Auf den Begriff des Grenzwertes stützen sich dann die Begriffe der Stetigkeit und der Ableitung. Zur historischen Entwicklung der Infinitesimalrechnung verweise ich auf [5] – [7]. Die Problematik des unendlich Kleinen wird z.B. in in [1], Kapitel 8 erörtert. Genauere Ausführungen zum Ableitungsbegriff in der Analysis findet man u.a. in [3] und in der Funktionalanalysis in [4].

2. Zur Geschichte des Ableitungsbegriffes

Die Infinitesimalrechnung ist erstmalig geschlossen, auf verschiedenen Wegen und etwa zur gleichen Zeit von *Newton* (1643-1727) und *Leibniz* (1646-1716) entwickelt worden. Der Ausgangspunkt von *Newton* lag in der Dynamik und dem zentralen Problem der Momentangeschwindigkeit. Funktionen $x = x(t)$ waren Fluenden (Weg-Zeit-Gesetze) und die differenziellen Wegänderungen $dx = \dot{x}(t)dt$ entsprachen sogenannte Fluxionenmomente $\dot{x}o$ (Momente der Geschwindigkeit $\dot{x}$ als Ableitung des Weges *x* nach der Zeit *t*). Beim Rechnen war zu beachten, dass Fluxionenmomente höherer Ordnung wegen ihrer Kleinheit vernachlässigt werden konnten.
Der Ausgangspunkt von *Leibniz* lag dagegen im Tangentenproblem an Kurven $y = f(x)$ der Geometrie und in der Bestimmung von Flächen unter solchen Kurven. Er führte die Differenzialschreibweise *dx* ein, die sich zusammen mit dem Integralzeichen als außerordentlich gut geeignet zum Rechnen im Differenzialkalkül erwies, wie z.B. die Formeln

$$d\int f(x)\,dx = f(x)dx, \qquad \frac{d}{dx}\int f(x)\,dx = f(x), \qquad \frac{dy}{dx} = \frac{dy}{du}\,\frac{du}{dx}$$

zeigen. Das erkannte *Leibniz* mit großer Klarheit. So sagte er: „Es ist wichtig, darauf zu achten, dass die Bezeichnungen Entdeckungen erleichtern. In wundervoller Weise kann man so die Arbeit des Geistes reduzieren."
Denkwürdig ist auch der Prioritätsstreit um die Erstentdeckung der Infinitesimalrechnung. Die Royal Society bezichtigte *Leibniz* 1712 des Plagiats. Newtons Konzept war tatsächlich etwa 10 Jahre eher entwickelt, in großen Teilen aber geheim gehalten worden. *Leibniz* wählte aber einen anderen Zugang und veröffentlichte ihn unverzüglich. So haben beide Gelehrte Großes geleistet. Mehr dazu findet man in [6: S. 452 ff.]

3. Ableitungsvarianten bei Funktionen einer Veränderlichen

Bei Funktionen $f: D \subseteq R \to R$ mit dem Definitionsbereich *D* betrachten wir einen Häufungspunkt $x_* \in D$ von *D* und lassen *x* in *D* variieren. Existiert der Grenzwert

$$a = \lim_{x \to x_*} \frac{f(x) - f(x_*)}{x - x_*} \tag{3.1}$$

in *R*, so heißt er (erste) Ableitung (oder auch Gradient) von *f* an der Stelle x_* und wird z.B. mit $f'(x_*)$ oder mit $\frac{df}{dx}(x_*)$ bezeichnet. Die Funktion *f* heißt dann in x_* differenzierbar. Eine äquivalente Definition ergibt sich mit der Substitution $h = x - x_*$, bei der $h = \Delta x$ den Zuwachs in *x* beschreibt. Für Ableitungen ergeben sich nun Regeln, die das Differenzieren zu einem formalen Kalkül werden lassen. Daneben gibt es eine Reihe von wichtigen Anwendungen [3].

Betrachtet man die Kurve *C* mit $y = f(x)$, so stellt *a* den Anstieg der Tangente

$$y = l(x) = f(x_*) + a\,(x - x_*) \tag{3.1a}$$

an *C* in x_* dar. Über $a = \tan\alpha$ erhält man auch den Anstiegswinkel α. Bei einer Zeitfunktion $f(t)$ ist $f'(t_*)$ die Momentangeschwindigkeit zur Zeit t_*. Ist $y = f(x)$ ein beliebiger Zusammenhang zwischen den Größen *x* und *y*, so ist $f'(x_*)$ die momentane Änderungsrate von *y* bezüglich *x* für x_*.
Bringt man den erweiterten Zahlenbereich $\bar{R}$ durch Hinzunahme von $\pm\infty$ ins Spiel, so ist in (3.1) auch $x_* = \pm\infty$ und $a = \pm\infty$ zugelassen. Im ersten Fall handelt es sich um Ableitungen im Unendlichen, im zweiten Fall um unendliche Ableitungen. Man spricht dann nicht mehr von Differenzierbarkeit, aber die Tangenteninterpretation bleibt. So signalisiert eine unendliche Ableitung eine vertikale Tangente.
Ist f in x_* differenzierbar, so ist *f* dort auch stetig. Die Umkehrung gilt bekanntlich im Allg. nicht. Hat eine Funktion *f* in x_* einen „Knick" wie z.B. $f(x) = |x|$ in 0, so ist sie dort nicht differenzierbar. In diesem Falle gibt es zwei Konzepte, um Abhilfe zu schaffen. Das erste Konzept ist die Definition einseitiger (links- und rechtsseitiger) Ableitungen

$$a_\pm = f'(x_* \pm 0) = \lim_{x \to x_* \pm 0} \frac{f(x) - f(x_*)}{x - x_*}\,. \tag{3.2}$$

Dieses Konzept greift auch noch bei bestimmten Unstetigkeiten von *f* in x_* (z.B. Sprüngen), falls $f(x_*)$ erklärt ist.
Das zweite Konzept besteht in der Erfassung von Zahlen *a* mit der Eigenschaft

$$f(x) \geq l(x) = f(x_*) + a\,(x - x_*) \quad \text{für alle } x \in U(x_*), \tag{3.3}$$

wobei $U(x_*)$ eine geeignete Umgebung von x_* ist. Solche a nennt man Subgradienten. Die Geraden $y = l(x)$ „berühren" f in x_* und liegen ansonsten unterhalb der Kurve $y = f(x)$. Sie heißen auch Subtangenten (vgl. mit (3.1a)). Die Menge aller dieser Ansteige *a* von Subtangenten, ein reelles Intervall, nennt man Subdifferential $\partial f(x_*)$ von f in x_*. Dieses Konzept ist auf Funktionen zugeschnitten, die in einer Umgebung $U(x_*)$ von x_* konvex sind.
Im Falle der Funktion $f(x) = |x|$ ist $f'(0 \pm 0) = \pm 1$ und $\partial f(0) = [-1{,}1]$, wobei $f'(0 \pm 0) \in \partial f(0)$ gilt.
Im Gegensatz zu den expliziten Definitionen (3.1) und (3.2) von Ableitungen werden die Subgradienten implizit durch eine Ungleichung beschrieben.

4. Differenziale bei Funktionen einer Veränderlichen.

Leibniz behandelte Differenziale *dx* und *dy* von Funktionen $y = f(x)$ als unendlich kleine Größen, mit denen man nach bestimmten Regeln rechnen kann. Betrachtet man eine feste Stelle x_* , so kann man den „Differenzialquotienten"

$$\frac{dy}{dx} = f'(x_*) \tag{4.1}$$

auflösen und dafür in diesem Sinne $dy = f'(x_*)\,dx$ schreiben. Tatsächlich bietet sich für diese Sichtweise eine geeignete Interpretation an. Ist nämlich

$$dx = \Delta x = h = x - x_* \tag{4.1a}$$

der Zuwachs in *x*, so ist neben dem tatsächlichen Zuwachs $\Delta y = f(x) - f(x_*)$in *y* der (lineare) Zuwachs längs der Tangenten (3.1a) das oben angegebene *dy*. So ist die Tangente

$$y = l(x) = f(x_*) + dy, \quad dy = df(x_*, dx) \tag{4.1b}$$

die lineare Fortsetzung von *f* an der Stelle x_* , wobei *dy* auch von *dx* abhängt (vgl. (3.1a)). Wie wir noch sehen werden, eröffnet gerade dieser „alte" Zugang die Möglichkeit, Ableitungen auch in abstrakten Räumen einzuführen. Dabei ist aber zu beachten, dass *dx* und *dy* jetzt selbst variabel sind und keine ominösen unendlich kleinen Größen mehr repräsentieren. Ein weiterer „Gewinn" dieses Zugangs ist die Tatsache, dass für kleine $dx = \Delta x$ auch $dy \approx \Delta y$ gilt. Das kann für kleine Änderungen in *x* bei Abschätzungen für die Änderungen in *y* gut verwendet werden (Fehlerrechnung).

5. Ableitungen als Funktionen

Ersetzt man die feste Stelle $x_* \in D(f)$ durch die Variable $x \in D(f)$, so ergibt sich durch

$$Df(x) = f'(x) = \frac{df(x)}{dx} = \lim_{y \to x} \frac{f(y) - f(x)}{y - x} \tag{5.1}$$

eine neue Funktion f' , die für alle $x \in D(f') \subseteq D(f)$ definiert ist, für die der obige Grenzwert existiert. Gleichzeitig liefert der Differentialoperator *D* die Zuordnung von f zur Ableitung f'. Für die Differenziale gilt analog

$$dy = df(x, dx) = f'(x)\,dx. \tag{5.1a}$$

Nun gibt es Funktionen f mit $D(f) = D(f') = R$, Funktionen wie $f(x) = |x|$, die an einzelnen Stellen nicht differenzierbar sind und auch Funktionen, die nirgends differenzierbar sind.
Zugleich ist der Ableitungsprozess prinzipiell beliebig oft wiederholbar, so dass höhere Ableitungen entstehen:

$$f^{(n+1)}(x) = f^{(n)'}(x) \quad (n = 1,2, \dots).$$

6. Gebrochene Ableitungen

Für gebrochene Ableitungen existieren verschiedene Zugänge (siehe z.B. [2]). Einer baut auf der Verallgemeinerung der Binomialkoeffizienten durch

$$\binom{\alpha}{\beta} = \frac{\Gamma(\alpha+1)}{\Gamma(\beta+1)\cdot\Gamma(\alpha-\beta+1)} \quad (\alpha \in R, \beta \in R)$$

mit Hilfe der Γ-Funktion auf und setzt für reelle p die p-te Ableitung mit

$$f^{(p)}(x) = \lim_{h\to 0} \frac{1}{h^p} \sum_{k=0}^{n} (-1)^k \binom{p}{k} f(x-kh) \quad (0 < p \le n)$$

als Grenzwert linksseitiger Differenzenquotienten n-ter Ordnung fest. Ein anderer nutzt Eigenschaften der Fourier-Transformation **F**, wenn für transformierbare Funktionen f die p-te Ableitung als

$$f^{(p)}(x) = \mathbf{F}^{-1}\{(\mathrm{j}\,u)^p(\mathbf{F}f)(u)\}(x)$$

definiert wird. Beide Varianten liefern übrigens die Leibnizsche Vermutung

$$x^{(0,5)} = x^{0,5} = \sqrt{x}.$$

7. Ableitungen komplexer Funktionen

Komplexe Funktionen lassen sich in Real- und Imaginärteil aufspalten:

$$w = f(z) = f(x + \mathrm{j}\,y) = u(x,y) + \mathrm{j}\,v(x,y).$$

Die Definition der Ableitung $f'(z)$ kann aber analog zu (5.1) erfolgen. Dabei existiert $f'(z)$ genau dann, wenn die sogenannten Cauchy-Riemannschen Differenzialgleichungen erfüllt sind.

8. Ableitungen von Vektorfunktionen

Die Ableitungen von Vektorfunktionen $f: D \subseteq R^n \to R^m$ ist nicht mehr in der Form (3.1) möglich, weil dann im Nenner Vektoren stehen würden. Man kann (3.1) aber äquivalent so umformen, dass eine Übertragung auf Vektorfunktionen trotzdem gelingt. Dabei kommt vor allem die Differenzialauffassung zum Tragen. Wir nennen eine lineare Abbildung **A** von R^n in R^m, wir schreiben dafür auch kurz $\mathbf{A} \in \mathrm{L}(R^n, R^m)$, die Ableitung $\mathbf{A} = f'(x_*)$ von f an der Stelle $x_* \in D(f)$, wenn folgendes gilt:

$$\lim_{x\to x_*} \frac{|f(x) - f(x_*) - \mathbf{A}\,(x - x_*)|}{|x - x_*|} = 0. \tag{8.1}$$

Bekanntlich kann **A** über die Beziehung $\mathbf{A}h = \mathrm{A} \cdot h$ mit einer $(m \times n)$-Matrix A identifiziert werden, wobei $dx = h = x - x_* \in R^n$ gilt. In der Praxis werden daher lineare Abbildung und Matrix gleichgesetzt.
Die Linearisierung von f in x_* lautet analog zum klassischen Fall

$$y = l(x) = f(x_*) + f'(x_*)\,(x - x_*) = f(x_*) + dy$$

wobei die tangentiale Funktion $l(x)$ nun einen affin-linearen Tangentialraum erzeugt. Den Zähler in (8.1) kann man übrigens sowohl als $|f(x) - l(x)|$ als auch als $|\Delta y - dy\,|$ schreiben, den Nenner als $|dx|$.
Verwendet man für $f(x)$ die Komponenten in $f = (f_1, \dots, f_m)^T$ und die Koordinaten in $x = (x_1, \dots, x_n)^T$, so gilt z.B. bei in x_* stetigen partiellen Ableitungen der f_i nach x_j für die Ableitung $f'(x_*)$ die Darstellung

$$\mathrm{A} = J(x_*) = \left(\frac{\partial f_i}{\partial x_j}\,(x_*)\right)_{i=1,\dots,m;\,j=1,\dots,n} \tag{8.2}$$

durch die Jacobi-Matrix $J(x)$ in $x = x_*$. Im Unterschied zu den partiellen Ableitungen in den Komponenten spricht man bei $f'(x_*)$ auch von der totalen Ableitung der Funktion f in x_*.
Für die Funktion $f(x) = |x| = \sqrt{x_1^2 + \cdots + x_n^2}$ von n Veränderlichen $(m = 1)$ ergibt das für alle $x \neq 0$ den Zeilenvektor

$$f'(x) = \frac{x^T}{|x|} = \left(\frac{x_i}{\sqrt{x_1^2 + \cdots + x_n^2}}\right)_{i=1,\dots,n}.$$

Schreibt man bei den Differenzialen

$$dy = df(x_*, dx) = df(x_*)(dx) = f'(x_*)\,dx = \mathrm{A} \cdot dx, \tag{8.3}$$

so verwischt der Unterschied zwischen dem Differenzial $df(x_*)$ von f und der Ableitung $f'(x_*)$ von f. Allerdings muss man dann bei Differenzialen ähnlich wie schon bei Funktionen zwischen Variablen und Abbildungen unterscheiden. Während die erste Ableitung eine *Linearform* ist, kann man höhere Ableitungen als *Multilinearformen* deuten, deren Regeln sich wieder in einem Kalkül beschreiben lassen [4].
Ersetzt man x beim Differenzieren durch $x_* + \mu \cdot e$ mit dem Skalar μ und dem Richtungsvektor e mit dem Betrag 1, so kann man *Richtungsableitungen* $\frac{\partial f}{\partial e}\,(x_*)$ einführen, die für die Einheitsvektoren in den Koordinatenrichtungen speziell die partiellen Ableitungen ergeben. Dabei gilt:

$$\frac{\partial f}{\partial e}\,(x_*) = f'(x_*)e\,.$$

9. Ableitungen von Distributionen

Betrachtet man lineare stetige Funktionale **T** über dem Raum $D = C_0^\infty(\Omega)$ der in Ω beliebig oft differenzierbaren Funktionen $\varphi(x)$ mit kompaktem Träger, so erhält man (Schwarzsche) Distributionen. Da man die speziellen Distributionen

$$\mathbf{T}_f\varphi = \int_{-\infty}^{+\infty} f(x) \cdot \varphi(x)\, dx \quad (\varphi \in D)$$

mit den Funktionen $f(x)$ identifizieren kann, spricht man bei diesen Funktionalen auch von verallgemeinerten Funktionen. Mit dem Funktional $\mathbf{T}_\delta\varphi = \varphi(0)$ ist auch der Dirac-Stoß $\delta(x)$ als spezielle Distribution enthalten. Für Distributionen **T** führt man nun durch

$$\mathbf{T}'(\varphi) = -\mathbf{T}(\varphi')$$

auf einfache Weise Ableitungen ein. Diese ergeben bei Anwendung auf gewöhnliche nicht differenzierbare Funktionen dann *verallgemeinerte Ableitungen*. In diesem Kalkül ist mit der Einheitssprungfunktion $\sigma(x)$ z.B

$$|x|' = -1 + 2\sigma(x), \qquad |x|'' = 2\,\delta(x).$$

10. Ableitungen in abstrakten Räumen

Die beschriebenen Ableitungsvarianten lassen sich unter bestimmten Voraussetzungen auch auf abstrakte Räume (z.B. Banachräume oder Hilberträume) übertragen.
Betrachtet man Funktionen (Operatoren) $F: D \subseteq X \to Y$ mit zwei Banachräumen X und Y, so wird die Frechét-Ableitung $\mathrm{A} = F'(x_*)$ oder auch starke Ableitung analog zu (8.1) eingeführt, wobei $\mathrm{A} \in L(X,Y)$ gilt und die Beträge durch die entsprechenden Normen $||.||_X$ und $||.||_Y$ ersetzt werden. Außerdem lassen sich auch Ableitungen $\frac{\partial F}{\partial e}\,(x_*)$ in Richtung von Einheitsvektoren *e* einführen.
Gilt mit einem $\mathrm{A} \in L(X,Y)$ nun die Beziehung $\mathrm{A}\,e = \frac{\partial F}{\partial e}\,(x_*)$ für alle solche Einheitsrichungen *e*, so nennt man A die Gâteaux-Ableitung oder schwache Ableitung von f in x_*. Aber im Unterschied zum klassischen Fall der Vektorfunktionen kann bei Banachräumen die schwache Ableitung existieren und die starke nicht.
Ist $Y = R$, so sind die Funktionen sogenannte Funktionale *f*. Für konvexe Funktionale *f* kann man analog zu (3.3) Subdifferenziale $\partial f(x_*)$ von *f* in x_* definieren. An die Stelle der Subgradienten *a* treten dabei lineare Funktionale ***a*** aus dem dualen Raum $X^* = L(X,R)$. Ist $X = H$ ein Hilbertraum, so ist ***a*** mit einem Element aus H identifizierbar.

Literaturverzeichnis

1. Bedürftig, T. und Murawski, R., Philosophie der Mathematik, de Gruyter, Berlin 2010.
2. Podlubny, I., Fractional Differential Equations, Mathematics in Science and Engineering Vol. 198, Academic Press 1999.
3. Schott, D., Ingenieurmathematik mit MATLAB, Hanser, Leipzig 2004.
4. Teubner-Taschenbuch der Mathematik, Teubner, Stuttgart und Leipzig 1996.
5. Wußing, H. und Arnold, W., Biographien bedeutender Mathematiker, Volk und Wissen, Berlin 1983.
6. Wußing, H., 6000 Jahre Mathematik – Eine kulturgeschichtliche Zeitreise, Band 1: Von den Anfängen bis Leibniz und Newton, Springer Verlag: Berlin 2008.
7. Wußing, H., 6000 Jahre Mathematik – Eine kulturgeschichtliche Zeitreise, Band 2: Von Euler bis zur Gegenwart, Springer Verlag: Berlin 2009.

Autor

Prof. Dr. rer. nat. habil. Dieter Schott
Fakultät für Ingenieurwissenschaften, Gottlob-Frege-Zentrum
Hochschule Wismar
Philipp-Müller-Str. 14
D-23966 Wismar
E-Mail: dieter.schott@hs-wismar.de

Thomas Risse

Anschaulichkeit der Geometrie in Beispielen

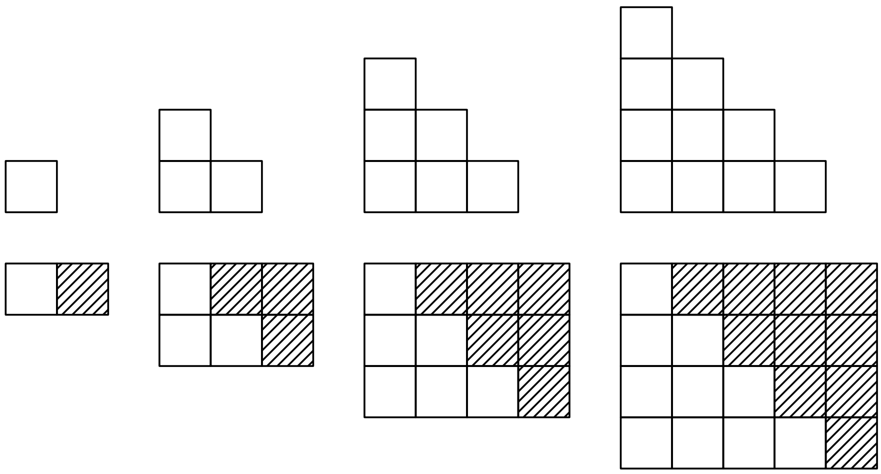
Auszug. Viele Beispiele lehren, die Anschaulichkeit der Geometrie zu schätzen. Neben klassischen Beispielen gilt dies besonders für Tristan Needham's "Visual Complex Analysis". All diese Beispiele zeigen, wie nützlich die geometrische Herangehensweise in Lehre und Forschung sein kann.

Einführung

Geometrische Überlegungen machen das, was ihnen ggfls. an mathematischer Strenge fehlt, durch ein Mehr an Anschaulichkeit und Plausibilität locker wett, wie Beispiele zeigen. Beispiele zeigen auch, daß Geometrie häufig erlaubt, überhaupt erst einmal Hypothesen aufzustellen.

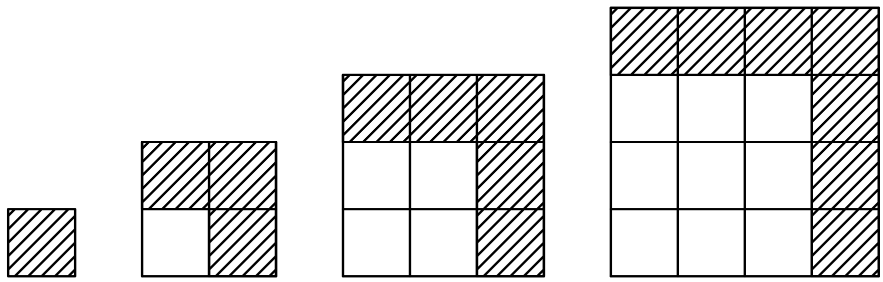
Es geht auch ohne Induktion

• $S_n = \sum_{i=1}^n i = \binom{n+1}{2}$ zeigt man geometrisch wie im Gauß-Trick durch Verdoppelung:

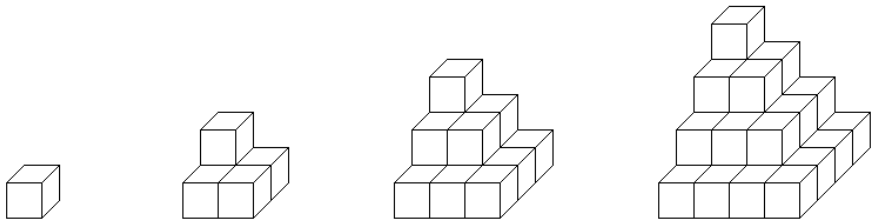


So wird $S_n = \frac{n(n+1)}{2}$ wie auch $S_{n-1} + S_n = n^2$ plausibel.

• $T_n = \sum_{i=1}^n (2i-1) = n^2$ ist geometrisch rekursiv:

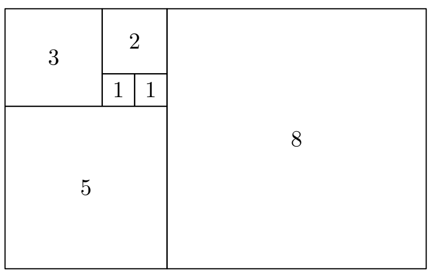


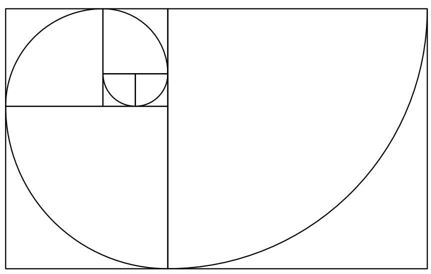
- $\sum_{i=1}^{n} i^2 = \frac{n(n+\frac{1}{2})(n+1)}{3}$ ergibt sich geometrisch als Volumen einer gestuften Pyramide:



$\sum_{i=1}^{n} i^2 = V_{umschriebenePyramide} - V_{Kehlen} - V_{Ecken} = \frac{1}{3}(n+1)^3 - 4\binom{n+1}{2} V_{unitPrism} - 4(n+1)V_{Ecke} = \frac{1}{3}(n+1)^3 - 4\binom{n+1}{2}\frac{1}{2}\frac{1}{2}\cdot 1 - 4(n+1)\frac{1}{3}\frac{1}{2}\frac{1}{2} = \frac{n(n+\frac{1}{2})(n+1)}{3}$.

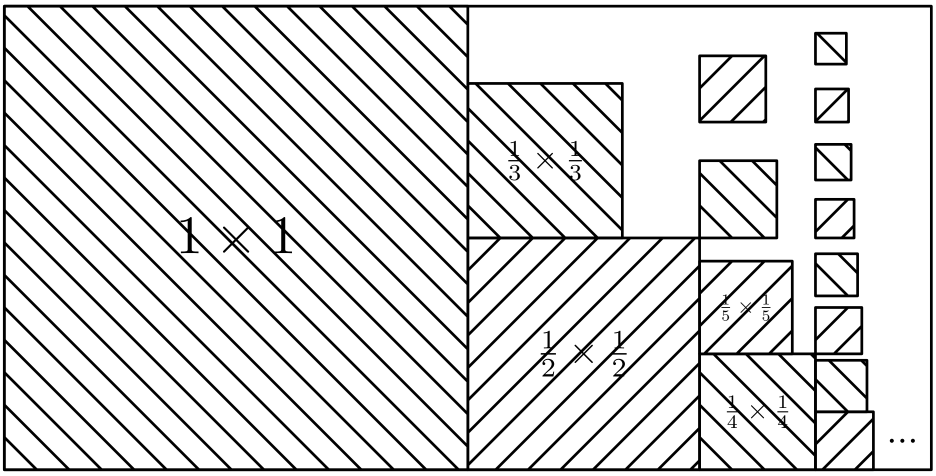
- $\sum_{i=1}^{n} f_i^2 = f_n f_{n+1}$ für die Fibonacci Zahlen $(f_i)_{i=1,2,\ldots}$ mit $f_n = f_{n-1} + f_{n-2}$ und $f_0 = 1 = f_1$





Was schon Jakob Bernoulli wußte

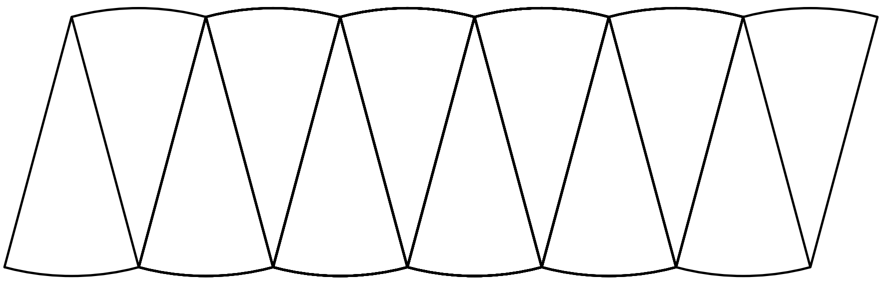
$\sum_{n=1}^{\infty} \frac{1}{n^2} = 1 + \sum_{n=2}^{\infty} \frac{1}{n^2} < 1 + \int_1^{\infty} \frac{dx}{x^2} = 2$ kann auch geometrisch abgeschätzt werden. Wegen $\sum_{n=o}^{\infty} \frac{1}{2^n} = 1 + 1 = 2$ gilt nämlich



Übrigens löst Euler 1735 mit $\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ das *Basler Problem.*

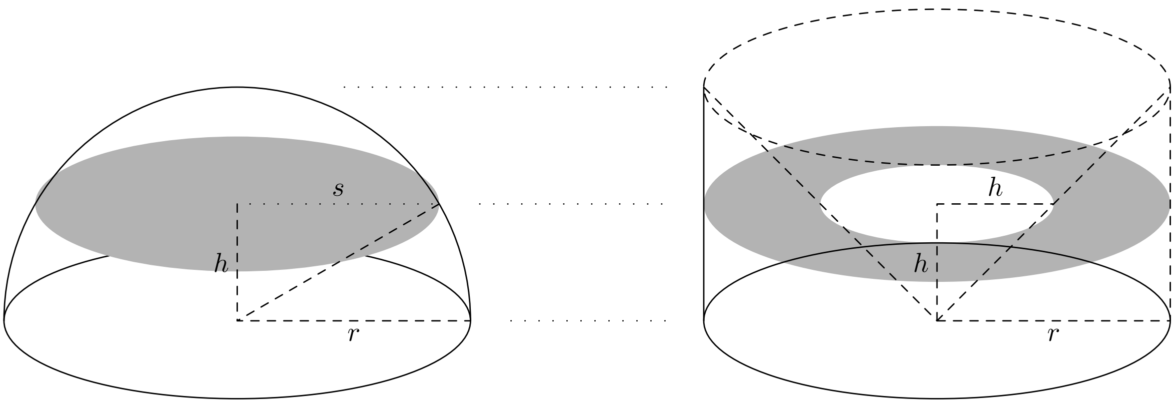
elementare Flächen und Volumina

• k sei Kreis mit Radius r. Zeige $|k| = \pi r^2$



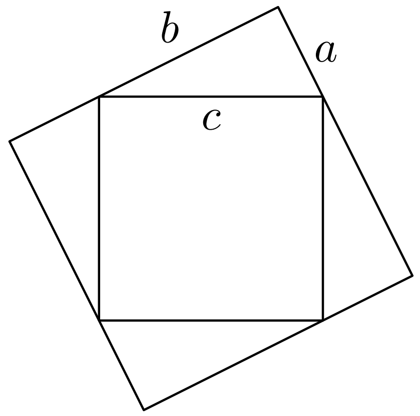
$n = 12$ aber für $n \to \infty$ $|k| = (U/2)r$ $= \frac{2\pi r}{2} r^2$

• K sei Kugel mit Radius r. Zeige $|K| = \frac{4}{3}\pi r^3 = 2(|Z| - |C|)$. Das Prinzip von Cavaglieri garantiert, daß die Rauminhalte zweier Körper übereinstimmen, wenn nur die Schnittflächen zu jedenm Niveau denselben Flächeninhalt haben, wie hier, da $\pi s^2 = \pi(r^2 - h^2)$.



zum Beispiel Pythagoras (570–510)

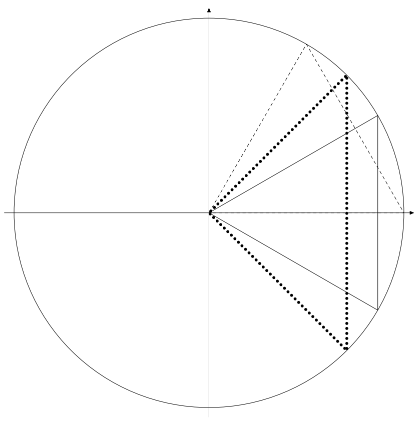
Es gibt erstaunlich viele Beweise des Satzes des Pythagoras, s. z.B. [1]. Ein elementarer Beweis verwendet nur Flächeninhalte von Rechtecken:



$$(a+b)^2 = a^2 + 2ab + b^2 = 2ab + c^2$$
$$\Downarrow$$
$$a^2 + b^2 = c^2$$

Wertepaare der trigonometrischen Funktionen

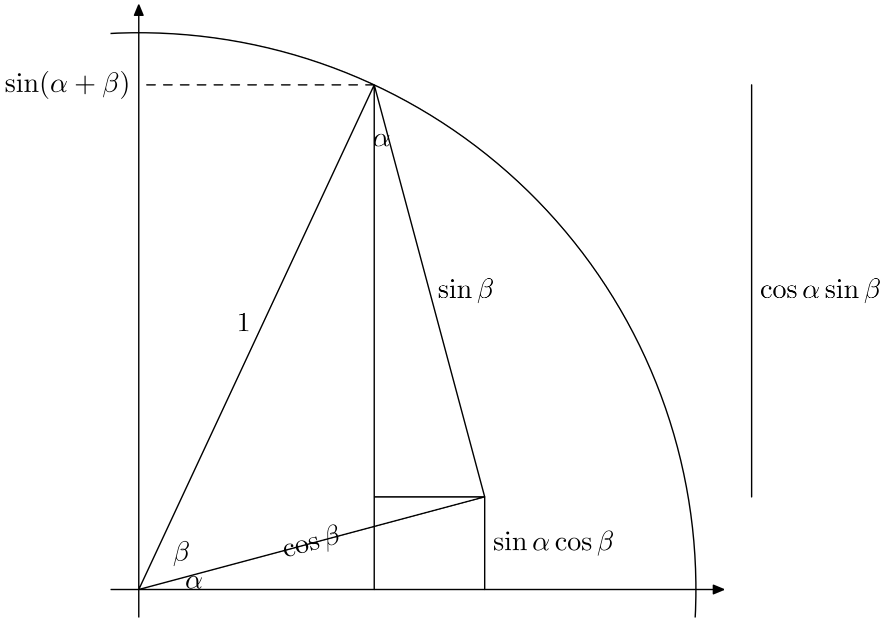
Die folgende Eselsbrücke zeigt man geometrisch natürlich im Einheitskreis: $\pi/6$: ergänze zu gleichseitigem Dreieck (durchgezogen); $\pi/4$: gleichschenklig rechtwinkliges Dreieck (punktiert); $\pi/3$: gleichseitiges Dreieck (gestrichelt)



$\sin\varphi$	φ	$\cos\varphi$	$\tan\varphi$	$\cot\varphi$
$\sqrt{0}/2$	0	$\sqrt{4}/2$	0	∞
$\sqrt{1}/2$	$\pi/6$	$\sqrt{3}/2$	$1/\sqrt{3}$	$\sqrt{3}$
$\sqrt{2}/2$	$\pi/4$	$\sqrt{2}/2$	1	1
$\sqrt{3}/2$	$\pi/3$	$\sqrt{1}/2$	$\sqrt{3}$	$1/\sqrt{3}$
$\sqrt{4}/2$	$\pi/2$	$\sqrt{0}/2$	∞	0

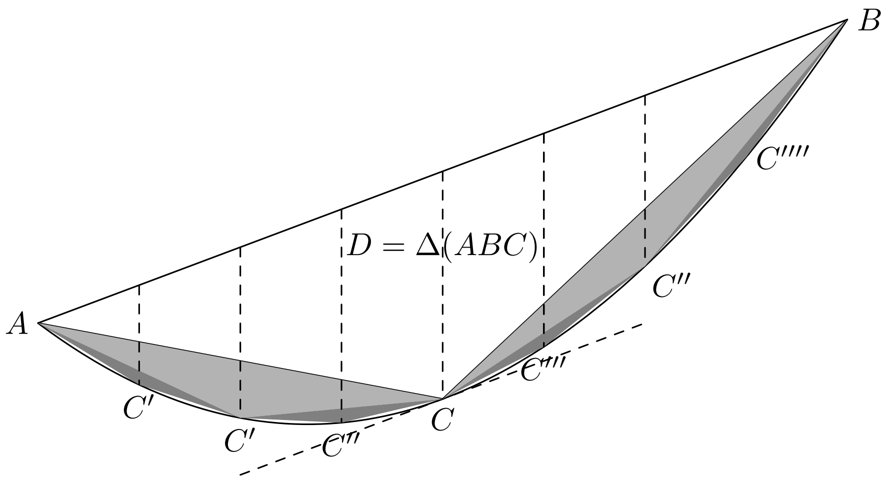
Additionstheoreme für Sinus und Cosinus

Ein richtig interpretiertes Bildchen liefert das Additionstheorem z.B. des Sinus.

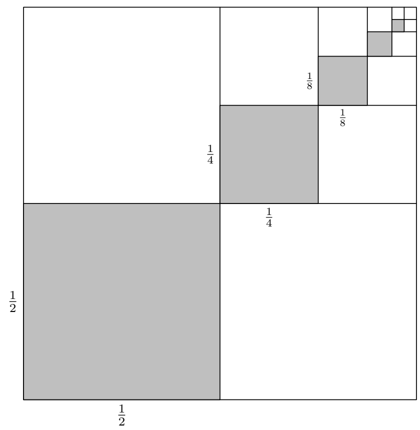


Integration à la Archimedes (287–212)

Archimedes approximiert den Flächeninhalt $|P|$ eines Parabel-Abschnittes P durch die Summe der Flächeninhalte von Dreiecken $|P| \approx |D| + \frac{1}{4}|D| + \frac{1}{4^2}|D| + \ldots + \frac{1}{4^n}|D|$.



Archimedes ermittelt auch den Grenzwert $\sum_{i=1}^{\infty} \frac{1}{4^i} = \frac{1}{3}$ geometrisch



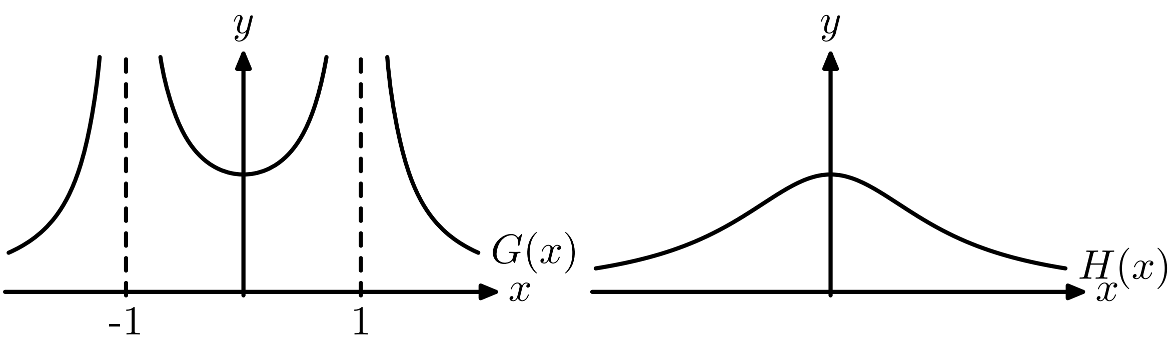
und gewinnt $|P| = \frac{4}{3}|D|$ ohne jede Infinitesimal-Rechnung.

Gängiges aus der Analysis

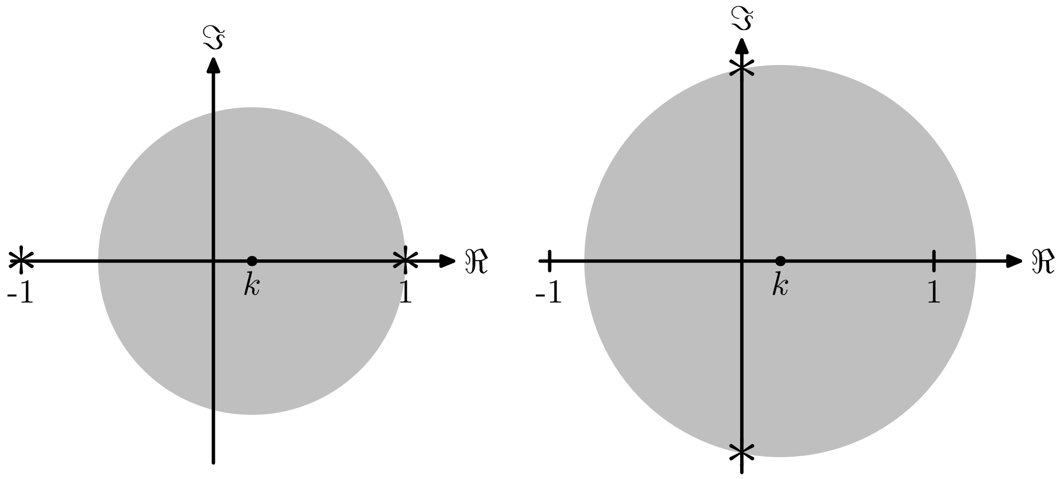
Differential/Ableitung als Tangentensteigung, Integral als Riemann'sche Summen, Lösungen von $y' = g(x, y)$ als Kurven, deren Tangenten mit dem Richtungsfeld $\vec{r}(x, y) = \big(1, g(x, y)\big)$ zusammenfallen, usw. sind allesamt gängige geometrische Veranschaulichungen er Infinitesimalrechnung.

reelle vs komplexe Analysis

Ein Beispiel [2] mag illustrieren, dass erst die komplexe Analysis Phänomene der reellen Analysis erklären kann. Sei $G(x) = \frac{1}{1-x^2} = \sum_{\nu=0}^{\infty} x^{2\nu}$ für $|x| < 1$ und $H(x) = \frac{1}{1+x^2} = G(ix) = \sum_{\nu=0}^{\infty}(-1)^{\nu} x^{2\nu}$ für $|x| < 1$, obwohl H auf ganz $\mathbb{R}$ definiert ist.



Die analytischen Fortsetzungen haben mit * markierte, komplexe Sigularitäten und die geschummerten Konvergenz-Kreise bei Entwicklung um k.



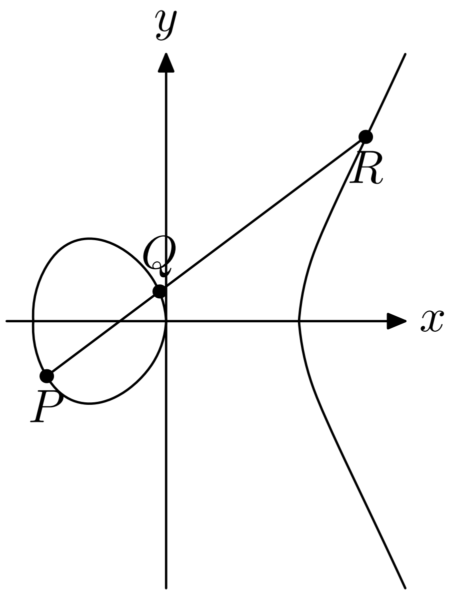
Erst $H(z)$ macht den Konvergenzradius von $H(x)$ um k plausibel.

Visual Complex Analysis

Tristan Needham verfolgt in seinem Buch [2] stringent die Idee, Funktionentheorie anhand geometrischer Anschauung zu entwickeln und möglichst viele Beweise eben geometrisch zu führen, s. Beispiele [4] und eine Anwendung auf Kurven-Integrale [3].

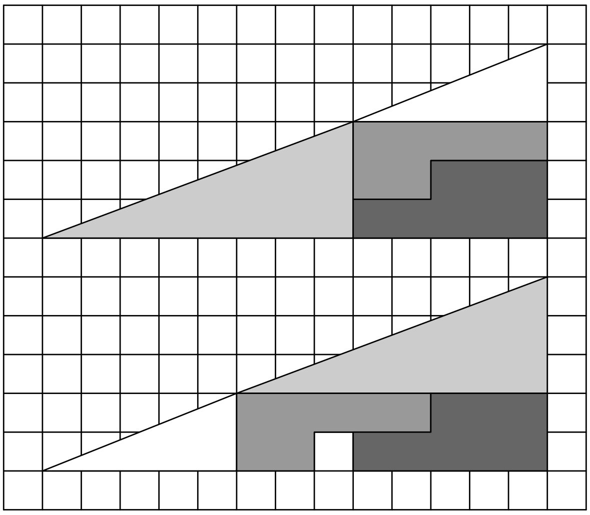
Potential und Grenzen

Z.B. die Kryptographie liefert ein Beispiel dafür, daß auch hier die Geometrie ein Verfahren wie ECC plausibel machen kann,



Man verwendet Weierstraß Kurven, $y^2 = x^3 + ax + b$ für $a, b \in \mathbb{K}$ über endlichem Körper $\mathbb{K}$ (nebenstehend gilt allerdings $\mathbb{K} = \mathbb{R}$ und $a = -1$, $b = 0$), um mit $P + Q + R = 0$ für Punkte der Kurve incl. einem unendlich fernen Punkt eine Addition zu definieren.

während andererseits ich mir keine geometrische Veranschaulichung etwa für RSA denken kann.
Auch kann man in geometrischen Beweisen genauso Fehler machen, wie die folgende 'Unmöglichkeit' zeigt.



Literaturverzeichnis

[1] **Bogomolny, A.**: *Pythagorean Theorem and its many proofs from Interactive Mathematics Miscellany and Puzzles.* www.cut-the-knot.org/pythagoras

[2] **Needham, T.**: *Visual Complex Analysis.* Claredon Press, Oxford 1997 http://people.math.sc.edu/girardi/m7034/book/VisualComplexAnalysis-Needham.pdf

[3] **Risse, Th.**: *Towards a Mathematical Library for UNUMs, an Alternative to IEEE 754 Floating Point Numbers.* J. Information and Communication Technology, http://jict.uum.edu.my, JICT Volume 17 No. 1 (Jan) 2018; www.weblearn.hs-bremen.de/risse/papers/JICT2017

[4] **Risse, Th.**: *Visual Complex Analysis – die Anschaulichkeit der Geometrie nutzen!* 14. Workshop *Mathematik in Ingenieur-wissenschaftlichen Studiengängen*, 18.9.2017 FAU, www.weblearn.hs-bremen.de/risse/papers/MathEng14

Autor

Prof. Dr. rer. nat. Thomas Risse
Fakultät E-Technik und Informatik
Hochschule Bremen, University of Applied Sciences
Flughafenallee 10, D-28199 Bremen
E-Mail: risse@hs-bremen.de

Nicolai von Schroeders

MatInEE[1]: Integrationstechniken mal anders – Erfahrungen aus einem Flipped-Classroom-Ansatz

[1] MatInEE steht für **Mat**hematik für **In**genieure, **E**rfolgreich **E**insteigen. An dem Projekt beteiligen sich sowohl die Fachwissenschaft (Dr. W. Rathmann) als auch die Fachdidaktik (N. von Schroeders).

1. Einführung

E-Learning ist ein Begriff, der seit neuester Zeit immer öfter Einzug in die hochschuldidaktische Lehre hält. In der Mathematikdidaktik wird der Begriff häufig mit dem von Spannagel (2011) verwendeten Flipped-Classroom-Ansatz (FCA) verknüpft. Dieser Ansatz soll speziell in Mathematikvorlesungen wesentliche Defizite der klassischen Hochschullehre aufheben (Handke & Schäfer, 2012, S. 93).
Für die praktische Umsetzung des FCA im Lehrbetrieb bedarf es einerseits relativ aufwendiger Vorbereitungen der Dozenten sowohl hinsichtlich der Lehrmaterialen als auch der Didaktik. Andererseits müssen die Studierenden ihr Lernverhalten überdenken und verändern, insbesondere bei der Verwendung digitaler Medien. Dabei darf nicht außer Acht gelassen werden, dass vor allem die Studierenden Kenntnisse über derartige Lern-/Lehrmethoden nicht aus der Schule mitbringen. Daher wurde im Rahmen einer Vorlesung zur Mathematik für Ingenieure mit ca. 300 Studierenden der Versuch unternommen, eine kurze Lernsequenz zum Themenblock „Elementare Integrationstechniken" mit Hilfe des FCA und E-Learning-Elementen zu behandeln. Zudem wurde im Zuge einer Masterarbeit hierzu der Aufwand und die Akzeptanz der Lehrmethode seitens der Studierenden erfasst und quantitativ ausgewertet. Die daraus resultierenden Ergebnisse und Erfahrungen werden im Folgenden dargestellt.

2. Flipped-Classroom-Ansatz

Klassische universitäre Lehrveranstaltungen laufen nach dem Prinzip ab, dass in den Vorlesungen (Präsenzphasen) die Inhalte vermittelt werden und die Studierenden im Anschluss den Lernstoff durch Nacharbeiten der Skripte und Bearbeiten von Übungsaufgaben verinnerlichen und vertiefen sollen.
Der Flipped-Classroom-Ansatz oder auch das Inverted-Classroom Modell dreht diese Reihenfolge um (Handke & Schäfer, 2012, S. 94). Die Studierenden erarbeiten sich mit zur Verfügung gestellten Lernmaterialen die Inhalte und in nachgeschalteten Präsenzveranstaltungen, werden diese Inhalte geübt, vertieft und analysiert.

Abb. 1: Klassischer Ansatz in der Hochschullehre vs. Flipped-Classroom-Ansatz

Für den FCA gibt es eine Menge an bekannten Vor- und Nachteilen (Handke & Schäfer, 2012, S. 97). Während die Lernenden sich vor allem mit hohen Anforderungen durch selbstgesteuertes Lernen konfrontiert sehen, kommt auf die Lehrenden unter anderem ein hoher Erstellungs- und Bereitstellungsaufwand für die Lernmaterialen und eine erhöhte Flexibilität in den Präsenzveranstaltungen zu. Gemeinsam ist beiden Gruppen, dass mangelnde Vorbereitung den Erfolg der Methode im Keim erstickt.

Auf der anderen Seite stehen natürlich die Vorteile. Die Festlegung des eigenen Lerntempos sowie die Möglichkeit eines direkten Austausches mit Kommilitonen liefern den Studierenden ein großes Potential für eine effektive Gestaltung und Nutzung der eigenen Lernphasen. Andererseits können die Lehrenden neben der Wiederverwendung gegebenenfalls unter adressatengerechter Anpassung der Lernmaterialien auch die Präsenzphasen lernerzentrierter und aktivierender gestalten. Dabei darf hinsichtlich des letztgenannten Arguments die Voraussetzung notwendiger didaktischer Kompetenzen nicht unberücksichtigt bleiben.

3. Planung und Realisierung des FCA

Inhaltlich wurde der FCA für die Sequenz zum Thema „Elementare Integrationstechniken“ umgesetzt. Dabei spielte zum einen eine Rolle, dass diese Sequenz aus fachlicher Sicht viele Wiederholungselemente beinhaltet und damit für die Studierenden kein vollständig neu zu erlernender Themenkomplex darstellt. Außerdem können diese Inhalte auch leicht mit sinnvollen Anwendungsszenarien verknüpft werden, was insbesondere die Motivation in einer möglichen Präsenzveranstaltung erleichtert.

In einer auf ILIAS basierenden Lernplattform wurde dazu ein Lernmodul „Integrationstechniken“ implementiert, das sowohl ein Skript in Textform als auch Beispielaufgaben mit Lösungen, Übungsaufgaben, digitale Tests, ein Diskussionsforum, eine Literaturliste und Verlinkungen auf YouTube-Videos beinhaltet. Als Zeitfenster für die FCA-Sequenz wurden zwei Wochen angesetzt (vom 29.05.17 bis 09.06.17), in denen aufgrund eines Feiertags sonst nur drei „normale“ Vorlesungen stattgefunden hätten. Am ersten Termin, dem 29.05.17, hörten die Studenten eine klassische Vorlesung zur Einführung der

Integrationstechniken. Zusätzlich – und das ist ein ganz wichtiger Aspekt, weil die Methodenkompetenz zum FCA bei den Studierenden nicht vorausgesetzt werden kann – wurde den Studierenden der FCA vorgestellt und die Erwartungen an die Studierenden für die nächsten zwei Wochen kommuniziert. Begleitet wurde diese Einführungsveranstaltung durch einen Umfragebogen, der die Studierenden nicht nur zu einer Selbsteinschätzung zu ihren Fähigkeiten hinsichtlich der Kenntnisse zu Integrationstechniken befragt, sondern auch eine erste Einschätzung des FCA abverlangt.

5. Ist Ihnen das Flipped Classroom-Konzept bekannt? Ja / Nein
6. Haben Sie schon einmal mit dem Flipped Classroom-Konzept gearbeitet? Ja / Nein
7. Wie viele Stunden pro Woche planen Sie, sich mit der Lernumgebung zu beschäftigen? ____________ Stunden pro Woche
8. Welche Vorteile sehen Sie im Flipped Classroom-Konzept?

   ____

   ____

   ____

   ____
9. Wo sehen Sie Schwierigkeiten im Flipped Classroom-Konzept?

   ____

Abb. 2: Auszug aus dem Fragebogen zum FCA (Soukenik, 2017)

Für die Empirie wurde zusätzlich erfasst, ob die Übungsaufgaben und Tests in der Lernumgebung bearbeitet wurden. In die Klausur am Ende des Semesters wurde eine Aufgabe aufgenommen, die sich, um eine Vergleichbarkeit zu gewährleisten, inhaltlich ganz stark an einer Aufgabe aus dem vorherigen Jahr orientiert.

Aufgabe 2 (9 Punkte)

a) (2 Punkte) Es sei $f : \mathbb{R} \to \mathbb{R}$ eine auf $\mathbb{R}$ differenzierbare Funktion. Zeigen Sie folgende Regel:

$$\int f'(z)f(z)\mathrm{d}z = \frac{1}{2}(f(z))^2 + c,\ c \in \mathbb{R}.$$

b) (7 Punkte) Bestimmen Sie die Menge aller Stammfunktionen

$$\int \frac{(\cosh^2(\tau) + \cosh(\tau) + 2)\sinh(\tau)}{\cosh^3(\tau) - \cosh^2(\tau) + \cosh(\tau) - 1}\mathrm{d}\tau.$$

Abb. 3: Klausuraufgabe (Rathmann, 2017)

4. Erfahrungen und Ergebnisse

Wie bereits bei den Nachteilen des FCA angesprochen, darf der

Erstellungsaufwand bei so einem Projekt nicht unterschätzt werden. Dabei hängt der eigentliche Aufwand natürlich stark von den schon vorhandenen Materialien ab und inwieweit auch didaktisch-methodische Überlegungen einbezogen werden. Aufgerundet ergaben sich annähernd folgende Zeitumfänge (in Arbeitstagen gerechnet):

- 1 Tag für die Generierung der Skripte und Texte für die Lernumgebung,
- 5 Tage für die Zusammenstellung der Übungsaufgaben und Tests für die Lernplattform,
- 5 Tage für die technische Umsetzung in der Lernplattform,
- 2 Tage für die Konzeption der Aufgaben in den Präsenzveranstaltungen und
- 1 Tag für die Reflexion der Lern-/Lehrphase.

In der Einführungsveranstaltung zum FCA nahmen insgesamt 177 Studierende teil, von denen auch 176 Umfragebögen in die Auswertung einfließen konnten. Dabei ergaben sich sowohl bei der Nennung von Vorteilen als auch Nachteilen des FCA u.a. folgende interessante Aussagen:

- Immerhin 32% der befragten Studierenden gaben an, dass sie das eigenständige Vorbereiten/Erarbeiten **positiv** bewerten.
- 11% empfanden als **positiv**, sich direkt unter Zwang gesetzt zu fühlen und nicht erst vor der Klausur die Lernphasen zu starten.
- Gute 35% schätzten aber auch ihre eigene motivationale Grundeinstellung als zu gering ein und sahen dies als Manko der Methode an.
- Weitere 10% erwarteten einen erhöhten Zeitaufwand und bemängelten den erzeugten Zeitdruck.
- Nur 3% der Befragten vermissten jedoch die Erklärungen seitens einer Fachkraft.

Für eine Veranstaltung mit 5 ECTS für die Vorlesung (Übung separat noch mal 2,5 ECTS) ist allerdings etwas erstaunlich, wie sich die Rückmeldungen zum

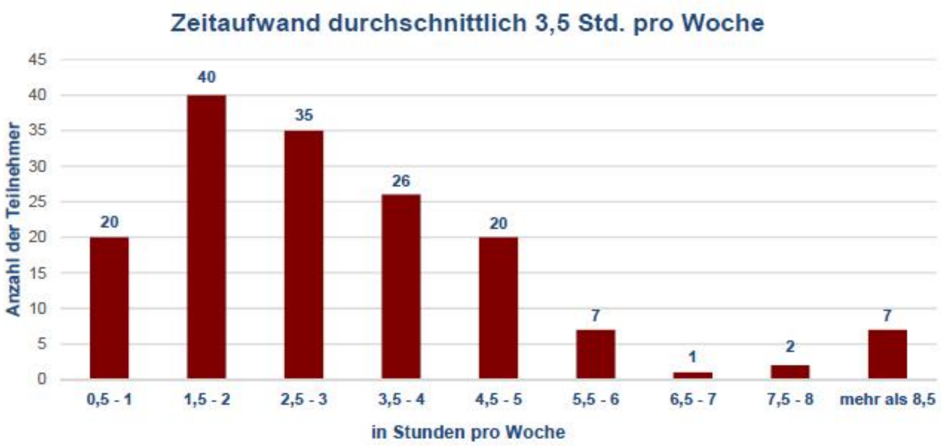


Abb. 4: Einkalkulierter Zeitaufwand durch Selbsteinschätzung (Soukenik, 2017)

einkalkulierten Zeitaufwand verteilen. Annähernd 54% der befragten Studierenden gaben nur maximal drei Stunden an, was deutlich unter den großzügig kalkulierten vier Stunden für ein Selbststudium liegen, während ein paar wenige, vermutlich nach Selbsteinschätzung ihrer Kenntnisse über Integrationstechniken, deutlich über 7,5 Stunden einplanten.

Als kleiner Erfolg kann verbucht werden, dass immerhin 45% aller 300 Studierenden die Übungsaufgaben „Rechenregeln zur Integration", 31% die Aufgaben zur Partialbruchzerlegung und noch 20% jeweils die Aufgaben zur partiellen Integration und Substitution bearbeitet haben. Hierbei muss berücksichtigt werden, dass ca. 300 Studierende für die Veranstaltung angemeldet waren, aber nur 177 davon die Einführungsveranstaltung zum FCA besucht haben.

Ein Vergleich der Mittelwerte (im SS 2016 3,7 Punkte, im SS 2017 4,6 Punkte) der Klausurergebnisse für die Aufgabe aus Abbildung 3 liefert mit Hilfe eines t-Tests (Bortz & Schuster, 2010, S. 120) einen signifikanten Unterschied (Varianz-homogenität und Normalverteilung gegeben, $\rho=0{,}0059$, Cohen's $d=0{,}324$ kleiner Effekt). Wohlwissend, dass dieses Ergebnis sicherlich empirischen Testkriterien nicht standhalten kann, kann doch zumindest die Hypothese aufgestellt werden, dass sich durch den Einsatz des FCA die Kenntnisse zu den Integrationstechniken hier nicht deutlich verschlechtert haben. Dabei muss erwähnt werden, dass die Messung eines Erfolgs der Methode nicht im Fokus des Versuchs stand.

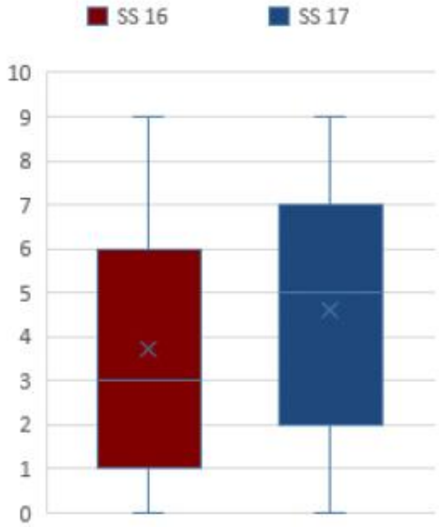


Abb. 5: Boxplots der Punkte aus den Klausuraufgaben aus dem SS16 und SS17

5. Fazit

Es hat sich gezeigt, dass der Aufwand für die Erstellung der digitalen Lernmaterialien relativ hoch sein kann. Wenn in dem Zusammenhang Lernvideos erstellt werden müssen, bietet es sich an, den Hinweisen von Handke (2015) zu folgen. Dies benötigt allerdings deutlich mehr Vorlaufzeit.

Zu unterschätzen ist auch nicht der deutlich höhere fachliche und didaktische Flexibilitätsgrad, der in den Präsenzveranstaltungen von Nöten ist. Die intensivere Interaktion mit den Studierenden schafft eine Atmosphäre, die nicht nur deren

Lernfortschritt zu Gute kommt, sondern auch die Kommunikation zwischen den Studierenden und Dozenten deutlich verbessert. Daraus kann sich dann vor allem auch für die Dozenten ein methodisch-didaktischer Lernzuwachs ergeben.
Die Einführung in die Methode und die Darlegung der Erwartungen sowohl an die Studierenden als auch die Dozenten schafft auf beiden Seiten Transparenz und Vertrauen. Dass hierbei ein fachlicher Schwerpunkt gesetzt worden ist, der nicht – oder zumindest bei den meisten Studierenden – gänzlich unbekannt war, ermöglicht, den Fokus auf die eigentliche Methode zu legen.
Wie bereits in Abschnitt 2 angesprochen, sind die Schwierigkeiten so einer Methodik hinlänglich bekannt. Auch Studierende, die mit so einem neuen Lehransatz konfrontiert werden, können jedoch schnell Vor- und Nachteile identifizieren. Auch ist klar, dass leistungsstärkeren Studierenden von solchen Lehrformen keine Nachteile erwachsen, hingegen schwächere Studierende bedingt durch ihre geringere Frustrationstoleranz beim Lernen als auch ihre defizitären Lerntechniken hier schnell an ihre Grenzen stoßen. Umso wichtiger ist es, dass solche Methoden in „kleinen“ Portionen in den Lehrbetrieb integriert werden und die Studierenden dabei intensiv betreut werden. Vorstellbar wäre hier ein Begleitangebot, in dem Ansprechpartner sowohl für die fachlichen als auch die methodischen Fragen jederzeit „analog“ zur Verfügung stehen.

Literaturverzeichnis

1. Bortz, J., Schuster, C., Statistik für Human- und Sozialwissenschaftler, SpringerMedizin, Springer-Verlag: Heidelberg 2010.
2. Handke, J, Handbuch Hochschullehre Digital, Tectum Verlag: Marburg 2015.
3. Handke, J, Schäfer, A.-M., E-Learning, E-Teaching und E-Assessment in der Hochschullehre, Oldenbourg Wissenschaftsverlag: München 2012.
4. Soukenik, D., Flipped Classroom-Konzept in der Vorlesung Mathematik für Ingenieure D2, Masterarbeit am Lehrstuhl für Didaktik der Mathematik, FAU Erlangen-Nürnberg 2017.
5. Spannagel, C., Die umgedrehte Mathematikvorlesung, https://cspannagel.wordpress.com/2011/08/07/; Zugriff am 24.09.2017.
6. Rathmann, W., Klausur Mathematik für Ingenieure D2, SS 17, FAU Erlangen-Nürnberg 2017.

Autor

Dipl.-Math., Dipl.-Wi.-Math. Nicolai von Schroeders
Philosophische Fakultät, Department Fachdidaktiken
Lehrstuhl für Didaktik der Mathematik
Regensburger Straße 160
90478 Nürnberg
E-Mail: nicolai.von.schroeders@fau.de

Petra Leitert

Blended-Learning-Konzept für die Mathematikausbildung der Direktstudenten

1. Einführung

Die Probleme der Studierenden im 1. Studienjahr (unterschiedliches Ausgangsniveau, ungenügende Fähigkeiten, selbständig zu studieren, mangelnde Selbstdisziplin usw.) sind allzu gut bekannt. Besonders große Probleme haben dabei die Studierenden im Fach Mathematik. Um den Studierenden zusätzliche Lernmöglichkeiten bereitzustellen, entwickle ich seit einigen Jahren eine Lernplattform, dessen Schwerpunkt bisher die beiden Grundlagenkurse „Lineare Systeme“ (Lineare Algebra und lineare Optimierung) und „Analysis“ sind. Die Kurse sollen die Präsenzausbildung unterstützen und ergänzen, den Studierenden viele Übungsmöglichkeiten anbieten und sie an das regelmäßige Selbststudium heranführen. Leider neigen Studierende bei E-Learning-Angeboten häufig schnell dazu, nicht mehr an den Vorlesungen und Übungen teilzunehmen. Sie glauben, dass die Beschäftigung mit dem Lernsystem (oft erst kurz vor der Prüfung) ausreicht. Welche Überlegungen beim Aufbau dieser Kurse berücksichtigt werden bzw. wurden, um die Studierenden nicht aus dem Hörsaal „zu verlieren“, stelle ich im folgenden Abschnitt vor.

2. Blended-Learning-Konzept der Grundlagenkurse

Die beiden Grundlagenkurse sind für die Direktstudenten der Betriebswirtschaft und der Wirtschaftsinformatik des 1. Studienjahres mit dem Lernsystem Ilias umgesetzt. Dabei habe ich beim Aufbau der Kurse und der Auswahl der Lernmöglichkeiten eine Reihe pädagogischer und didaktischer Überlegungen, Methoden und Regeln beachtet. Auf der Grundlage dieser Erfahrung entstehen nun auch die Grundlagenkurse für die Studierenden der Ingenieurwissenschaften.

2.1 Aufbau der Kurse

Die Kurse sind einheitlich strukturiert. Grundlage sind die jeweiligen Themenbereiche, die dem inhaltlichen Vorgehen in den Vorlesungen und Seminaren entsprechen.

Neben den Themenbereichen enthalten die Kurse die Bestandteile:

- Kommunikations- und Informationsbereich,
- Übersichtsbereich für das gesamte Semester,
- Bereich zur Prüfungsvorbereitung und
- Rätselbereich.

2.2 Lernelemente der Bereiche

Die Bereiche der einzelnen **Themengebiete** enthalten folgende Elemente:

- Vorlesungsskript nur für das Thema
- Aufgabensammlung für das Thema (Grundlage der Seminare)
- mehrere Videos
    - Eigenproduktionen zu einem eingeschränkten Thema aus der Vorlesung – beispielsweise Musterlösungen oder Verfahren
    - Links zu ausgewählte Videos aus dem Internet
- zusätzliche Übungsblätter (mit unterschiedlichen Niveaustufen)
- Lösungen für alle Aufgaben (z.T. mit animierten Lösungen, um parallel die Aufgaben zu erarbeiten)
- Glossar mit typischen Verständnisfragen (einschließlich Antworten)
- Tests (auf der Basis der verfügbaren Fragetypen)

Mit Semesterbeginn stehen den Studierenden im **Übersichtsbereich** bereits das vollständige Vorlesungsskript sowie die gesamte Aufgabensammlung des Kurses, eine weitere Aufgabensammlung des „Taschenbuches der Wirtschafts-mathematik", eine Übersicht der typischen ökonomischen Anwendungsbeispiele (die auch in den Vorlesungen vorgestellt werden) sowie jeweils ein Test zu den im Kurs benötigten Abiturkenntnisse (einschließlich der Lösungen) zur Verfügung.

Der **Kommunikations- und Informationsbereich** enthält neben einem Forum, Informationen, Termine und Nachrichten zum Kurs (einschließlich der Prüfung) sowie interessante Zeitungs- und Online-Artikel zum Studium und über die Mathematik.

Uneingeschränkt steht der Bereich zur **Prüfungsvorbereitung** zur Verfügung. In diesem Bereich finden die Studierenden:

- Tipps zum Lernen im Fach Mathematik und speziell für die Prüfungen
- mehrere Probetests und eine abschließende Probeklausur
- alle alten Prüfungsklausuren sowie
- eine Vielzahl Videos über die Bedienung von Taschenrechnern zur Durchführung unterschiedlicher Operationen (z.B. Determinanten-Berechnung usw.)

Im **Rätselbereich** finden die Studierenden jeweils 4 Wochen vor Weihnachten bzw. Ostern einen Kalender mit kleineren humorvollen Rätselaufgaben. Die Rätsel früherer Jahre können sich die Studierende jederzeit (mit Lösungen) ansehen.

3. Pädagogische Überlegungen

Beim Aufbau der Kurse und der Wahl der einzelnen Bestandteile haben verschiedene pädagogische und didaktische Überlegungen eine Rolle gespielt.

Kurse

Die Bestandteile der E-Learning-Kurse für das 1. Studienjahre sind immer abgestimmt mit der Präsenzausbildung und bilden die Grundlage aller

Lernangebote (Vorlesung, Seminare, Tutorien, Propädeutikum). Sie ermöglichen die Vorbereitung, Nachbereitung und auch Vertiefung der verschiedenen Präsenz-Lehrangebote. Aus der Vielzahl der unterschiedlichsten Lernelemente können sich die Studierenden die Angebote, die sie ansprechen, auswählen und erhalten somit Ergänzungen zu den Vorlesungs- und Seminareinhalten und viele zusätzliche Übungsmöglichkeiten.
Die Kurse unterstützen die Motivation der Studierenden, geben ihnen Informationen über ihren aktuellen Lernstand der Lerninhalte und können gut im Selbststudium und zur Prüfungsvorbereitung genutzt werden.
Alle Bestandteile des Kurses werden im aktuellen Semester immer an die momentane Situation in der Lehre angepasst. Die einzelnen Themenbereiche werden erst parallel zum Ausbildungsstand der Vorlesungen und Seminare veröffentlicht. Diese schrittweise Freischaltung der Kursinhalte hilft den Studierenden, die Übersicht während des Semesters zu behalten. Zu Beginn des Studiums sind insbesondere die jüngeren Studierenden (vor allem, wenn sie keine Erfahrungen mit E-Learning haben) mit einem vollständigen Kurs durch die Vielfalt der Elemente überfordert.
Der Schwerpunkt der Vorlesungen sind ausführlich erklärte und vorgerechnete Beispiele, die die mathematischen Grundlagen aus dem Skript für die Studierenden verständlich machen sollen.
Jedoch enthalten die Kurse diese Vorlesungsbeispiele nicht. Auch Vorlesungsmitschnitte sind nicht Bestandteile der Kurse. Die Studierenden sollen die Erfahrung machen, dass es sich lohnt, in die Vorlesung zu gehen.

Im Semester, in dem die Vorlesungen nicht angeboten werden, bleibt der jeweilige Kurs vollständig freigeschaltet, um den Studierenden ein systematisches Nacharbeiten im Selbststudiums für die Vorbereitung der Nachprüfung zu ermöglichen. Treten Fragen zu den Tests, den Übungen der Aufgabensammlungen oder den alten Klausuren auf, können die Studierenden Konsultationen vereinbaren. Bedingung ist jedoch dabei, dass sie mit durchgerechneten Aufgaben zu den Konsultationen erscheinen.

Skript und Aufgabensammlung
Das Skript und auch die eigentliche Aufgabensammlung (die sich im Übersichtsbereich befinden) sind darüber hinaus in themenbezogene Teildokumente zerlegt. Diese finden die Studierenden dann in den einzelnen Themenbereichen. Mit diesen „kleineren“ Dokumenten können viele Studierende besser arbeiten als mit den umfangreichen Semesterunterlagen. Das Skript ist auch so aufgebaut, dass die Studierenden die Vorlesungsbeispiele gut im Dokument ergänzen können. Umfangreiche Vorlesungsbeispiele werden auf Wunsch der Studierende anschließend zusätzlich im Lernsystem veröffentlicht. Durch dieses Vorgehen nutzen die Studierenden verstärkt das Lernsystem und werden so auch immer wieder auf die anderen Lernmöglichkeiten aufmerksam.

In den Vorlesungen und z.T. in den Skripten erhalten die Studierende darüber hinaus auch Hinweise, was sie zusätzlich im Lernsystem nutzen können.
Die Lösungen der Aufgabensammlung werden erst freigeschaltet, wenn das Themengebiet im Seminar abgeschlossen wurde. Die Aufgabensammlung ist so umfangreich, dass nie alle Aufgaben im Seminar besprochen werden können. So stehen für die Tutorien, die persönliche Wiederholung und die Vorbereitung der Prüfungen neben den Seminaraufgaben noch weitere Aufgaben mit Lösungen zur Verfügung.
Dabei werden die einzelnen Rechenschritte von einer Reihe Aufgaben mit einem umfangreichen Lösungsweg zusätzlich auch animiert angeboten. Dadurch können sich die Studierenden schrittweise die Lösung ansehen. Anliegen ist es, dass die Studierende parallel zu ihrer selbstständigen Aufgabenlösung anschließend jeden Schritt sofort vergleichen können, um gar nicht erst in eine „falsche Richtung zu laufen". Da dieses Angebot bisher sehr gut von den Studierenden angenommen wurde, wollen wir es im nächsten Semester weiter ausbauen. Bei guten Studierenden tauchet jedoch immer wieder die Frage nach weiteren Lösungswegen auf. Sie werden bisher nur in den Vorlesungen und Seminaren besprochen.

Weitere Übungsaufgaben

Neben der Semester-Aufgabensammlung können die Studierenden noch eine Reihe weitere Aufgabensammlungen nutzen. So gibt es zu jedem Thema auch ein **Übungsblatt** mit 4 bis 6 Aufgaben, die in die drei Schwierigkeitsstufen „leicht", „mittel" und „schwer" unterteilt sind. Diese Einteilung unterstützen die Motivation der Studierenden, weil sie durch die leichteren Aufgaben schnell Erfolgserlebnisse haben und angeregt werden, sich auch an die schwierigeren Aufgaben zu wagen. Die Übungsblätter stehen den Studierenden vor allem für das Selbststudium zur Verfügung, werden aber auf Wunsch auch in den Tutorien und im Propädeutikum besprochen. Mit Abschluss des Themas in den Seminaren sind die Lösungen ebenfalls einsehbar.
Die **Probetests und Probeklausuren** enthalten zu allen wichtigen Inhalten noch einmal Aufgaben und dienen der Orientierung für die Studierenden, in wie weit sie das Themengebiet verstanden haben. Zu den Probeklausuren erhalten die Studierenden in der Prüfungsvorbereitung die Lösungen über das Lernsystem. Während des Semesters sollen sie zuerst die Aufgaben selber lösen und können dann in den Seminaren, Tutorien oder im Propädeutikum nachfragen.
Mit den veröffentlichten **alten Klausuren** verfügen die Studierenden über viele weitere Übungsmöglichkeiten. Sie können so testen, ob sie in der Lage sind, die Prüfungsanforderungen zu erfüllen. Das bedeutet, dass die Studierenden nicht nur die Aufgaben lösen können. Sie müssen auch in einer vorgegebenen Zeit fertig werden. Der ungewohnte Zeitdruck in den Klausuren hat besonders im den ersten Semestern einen großen Einfluss auf die Prüfungsergebnisse. Die Lösungen der Klausuren bekommen die Studierenden nicht im Lernsystem bereitgestellt. Sie haben jedoch auch hier wieder die Möglichkeit, in einer

Konsultation, in den Tutorien oder im Propädeutikum ihre Lösungen vorzulegen und mit den Tutoren bzw. den Lehrkräften die Probleme und Fehler zu besprechen.
Zu jedem Thema gibt es einen **Test** auf der Basis der aktuellen Fragetypen von Ilias. Die Tests sind so konfiguriert, dass die Studierenden sofort ihre Ergebnisse prüfen und sich die richtigen Lösungen anzeigen lassen können. Sie können jede Aufgabe beliebig oft wiederholen. Da bei den beiden Aufgabentypen „Formel" und „Stack" Variablen benutzt werden, erhalten die Studierenden bei jeder Aufgabenwiederholung andere Ausgangswerte, so dass der Übungseffekt erhöht wird.
Das **Glossar** (als FAQ-Fragen bezeichnet), enthält typische Fragestellungen zu jedem Thema, deren Beantwortung wichtige Zusammenhänge verdeutlichen. Zu jeder Frage können sich die Studierenden die richtigen Antwort anzeigen lassen, um zu kontrollieren, ob ihre richtig sind.
Die **Rätsel** dienen vor allem der Motivation und der Anerkennung der guten Studierenden. Bei den Rätseln werden nur mathematische Kenntnisse des aktuellen Vorlesungsstoffes genutzt und durch humorvolle Knobelaufgaben präsentiert.
Die Erklärung des Lösungsweges, die Auswertung der eingereichten Lösungen sowie die Prämierung mit kleinen Preisen erfolgt dann in der Folgewoche in den Vorlesungen. Die Veröffentlichung vergangener Rätsel (mit Lösungen) soll den Studierenden zeigen, dass die Aufgaben nicht nur von den sehr guten Studierenden gelöst werden können.
Zusätzlich können die Studierenden die **Aufgabensammlung** (mit Lösungen) zum **Taschenbuch „Wirtschaftsmathematik"** von Eichholz/Vilkner nutzen [1]. Die beiden Autoren sind ehemalige Mathematik-Professoren der Hochschule (Fakultät Wirtschaftswissenschaften). Deshalb passen die Aufgaben sehr gut zu unseren Ausbildungsinhalten.

Videos

Zu jedem Thema gibt es in den Themenbereichen verschiedene Videos. In Eigenproduktion sind mit Studierende Serien von Videos zu einem Themengebiet (z.B. Ableitungsregeln) entwickelt worden. Die Erklärungstexte haben die Studierenden selbst geschrieben (wir haben sie nur auf fachliche Korrektheit geprüft) und vorgetragen. Sie sprechen die Studierenden besonders an und werden von ihnen sehr gern genutzt. Zusätzlich erzeuge ich kleine Erklärungsvideos zu einem sehr eingegrenzten Problem, bei dem Sachverhalte aus der Vorlesung noch einmal in einem anderen Zusammenhang dargestellt werden.
Da die Studierenden auch viel nach geeigneten Videos im Internet suchen (und manchmal sehr viel Zeit dabei verbrauchen), aber nicht die Qualität dieser Videos einschätzen können, gibt es immer wieder Diskussionen zu falschen oder unverstandenen Lösungsmöglichkeiten. Deshalb bieten wir den Studierenden

Links zu (frei verfügbaren) Videos an, die gut zu unseren Ausbildungsinhalten passen und auch fachlich korrekt sind. an
Für diese Kurse wurden bewusst keine Videos erstellt, die Mitschnitte der Vorlesungen darstellen. Eine Reihe von Kollegen haben die Erfahrung gemacht, dass Vorlesungsmitschnitte die Studierenden sehr schnell dazu verführen, nicht mehr in die Vorlesung zu kommen. Die Studierenden unterschätzen jedoch den Arbeitsaufwand und die Disziplin, die für dieses Selbststudium erforderlich sind.

4. Effekte der Kurseinbindung in die Ausbildung

Die Zugriffszahlen beweisen, dass viele Studierende die Kurse aktiv über das gesamte Semester, natürlich aber besonders stark in der Prüfungszeit nutzen. Dadurch lernen viele Studierende im 1. Studienjahr die Nutzung der E-Learning-Elemente zum Selbststudium, was sich in den höheren Semestern auswirkt.
Die Aufteilung der Zensuren hat sich insbesondere in den letzten beiden Jahren spürbar verändert. Die Anteile der sehr guten und guten Note hat sich deutlich erhöht, der „3en“ ist etwa konstant geblieben und der „4en“ hat sich verringert.
Leider ist jedoch der Anteil der Studierenden, die durchfallen oder gar nicht zur Prüfung antreten nur leicht rückgängig. Aus den Erfahrungen wissen wir, dass es vor allem die Studierende sind, die wir mit den zusätzlichen Angeboten bisher nicht erreichen konnten. D.h. die Studierenden, die die Lernplattform aktiv als Ergänzung zu den normalen Lehrveranstaltungen nutzen, verbessern oder stabilisieren ihre Leistungen. Die Studierenden, die wir mit den normalen Lehrangeboten nicht erreichen, erreichen wir auch nicht mit den zusätzlichen Angeboten.
Ein besonderer Effekt tritt bei den Prüfungswiederholern auf. Die Studierenden, die bereits ein oder zweimal in den Mathematikprüfungen durchgefallen sind und nun doch anfangen, systematisch mit dem Lernprogramm zu arbeiten, erzielen z.T. gute bis sehr gute Ergebnisse.

Literaturverzeichnis

1. Eichholz, W; Vilkner, E. Taschenbuch der Wirtschaftsmathematik“, Fachbuchverlag: Leipzig 2013.

Autorin:

Prof. Dr. Petra Leitert
Fakultät für Wirtschaftswissenschaften, Gottlob-Frege-Zentrum
Hochschule Wismar
Philipp-Müller-Str. 14
D-23966 Wismar
E-Mail: petra.leitert@hs-wismar.de

Mikko Vasko

Ein Online-System für Hausaufgaben zur Ingenieurmathematik - Chancen und Herausforderungen

Zusammenfassung: Das Online-System STACK steht als Open-Source-Plug-In für die Lernplattformen Moodle und ILIAS zur Verfügung und wird an der Hochschule Karlsruhe seit 2014 für Mathematik-Übungen verwendet. STACK wird von Lehrenden sowie von Studierenden sehr gut angenommen – im Wintersemester 2017/18 bearbeiteten ca. 70% der Studienanfängerinnen und -anfänger der Hochschule Karlsruhe STACK-Übungsaufgaben in ILIAS. In diesem Artikel wird das System vorgestellt und über Erfahrungen mit dem System berichtet.

1. Einführung

„Abgabe der Übungsblätter sollte Pflicht sein“, fordert ein Student der Elektrotechnik in einer an der Hochschule Karlsruhe durchgeführten Befragung zu den Studienbedingungen. Auch andere Studierende geben ähnliche Antworten zu der Frage, wie der Erfolg der Studierenden in den Prüfungen verbessert werden könnte. Diese Befragungen wurden zwischen Sommersemester 2013 und Wintersemester 2014/15 im Rahmen des Projekts SKATING durchgeführt. Dieses durch den Qualitätspakt Lehre für den Zeitraum von 2011 bis 2020 geförderte Projekt wird in Kooperation zwischen der Hochschule Karlsruhe – Technik und Wirtschaft und der Geschäftsstelle der Studienkommission für Hochschuldidaktik an Hochschulen für Angewandte Wissenschaften in Baden-Württemberg (GHD) durchgeführt. Ziel dieses Projekts ist es, gemeinsam mit den Lehrenden und den Studierenden die Studienbedingungen in den ingenieurwissenschaftlichen Studiengängen weiter zu verbessern (Berendes et al. 2017).

Mathematik stellt für viele angehende Ingenieurstudierende an den Hochschulen für Angewandte Wissenschaften eine große Herausforderung dar. Als Ursachen bekannt sind die sehr unterschiedlichen Mathematik-Vorkenntnisse der Studienanfängerinnen und Studienanfänger, aber auch die Probleme mit der Prokrastination und der Zeitplanung sowie mangelndes regelmäßiges semesterbegleitendes Lernen und Üben (Abel und Weber 2014; Schulmeister und Metzger 2011).

Pflicht-Übungen oder Übungen, die in die Prüfungsnote einfließen, sind didaktische Mittel, die an Universitäten häufig gegen Prokrastination eingesetzt werden. An den Hochschulen für Angewandte Wissenschaften sind solche Übungen eher ungewöhnlich, was auf das hohe Lehrdeputat der Professorinnen und Professoren sowie auf den kaum vorhandenen akademischen Mittelbau zurückzuführen ist. Es besteht häufig keine Kapazität, um abgegebene Pflicht-Übungen zu korrigieren.

Online-Systeme für Hausaufgaben können ein Lösungsansatz für diese Problematik sein. An der Hochschule Karlsruhe wird dieser Ansatz im Projekt SKATING seit 2012 systematisch verfolgt und umgesetzt. In diesem Beitrag berichten wir von unseren Erfahrungen mit dem Online-System STACK für Mathematik-Hausaufgaben und diskutieren damit einhergehende Chancen und Herausforderungen.

2. Einführung eines Online-Übungssystems an der Hochschule Karlsruhe

Einer der wichtigen Erfolgsfaktoren bei der Einführung eines Online-Systems für die Lehre ist die frühzeitige Einbindung der Lehrenden (Bremer et al. 2010). Dementsprechend wurde bereits im Jahr 2012 durch das Projekt SKATING ein Treffen mit Mathematik-Professorinnen und -Professoren der Hochschule Karlsruhe organisiert, um Erfahrungen und Ideen zum Thema Online-Übungsaufgaben auszutauschen. Bei diesem Treffen haben sich zwei zentrale Anforderungen für ein Online-Übungssystem für Mathematik herauskristallisiert. Erstens, die Studierenden sollten als Antworten nicht nur Zahlen, sondern auch andere mathematische Ausdrücke eingeben können. Zweitens sollte für die Überprüfung der studentischen Antworten ein Computer-Algebra-System (CAS) eingesetzt werden. In enger Kooperation mit dem Informationszentrum der Hochschule Karlsruhe wurde die Umsetzung dieser Anforderungen im Projekt SKATING verfolgt: Es wurde systematisch nach Systemen mit den erwünschten Funktionalitäten gesucht.

Im Rahmen eines Lehrprojekts wurde im Jahr 2013 gemeinsam mit drei Mathematik-Lehrenden eine Lizenz für das Online-Übungssystem Maple T.A. für zwei Jahre erworben. Maple T.A. ist ein kommerzielles Produkt der Firma MapleSoft und wurde speziell für Mathematik-Online-Übungen entwickelt (Maple T.A. 2017). Die Lehrenden waren mit den Funktionalitäten von Maple T.A. sehr zufrieden, allerdings erwies sich die Verbindung des Systems mit der zentralen ILIAS-Lernplattform der Hochschule als problematisch – die Integrierbarkeit in ILIAS hatte sich als weitere Anforderung von Seiten der Studierenden ergeben. Auch die Ausweitung der Maple T.A.-Lizenz auf die gesamte Hochschule wäre auf Dauer nicht finanziell tragbar gewesen.

Aus diesen Gründen wurde die Idee, ein Open-Source-Mathematik-Übungssystem für die Lernplattform ILIAS zu entwickeln, in einer ILIAS-Anwendergruppe vorgestellt. Die Idee stieß sofort auf das Interesse seitens anderer Hochschulen. Die Thematik wurde in der Special Interest Group

Mathe+ILIAS (SIG Mathe+ILIAS), die sich mit dem Einsatz von ILIAS in der Mathematik beschäftigt, vorangetrieben. Es wurden mehrere Systeme eruiert und Möglichkeiten erforscht, ein geeignetes Übungssystem für Mathematik mit der Lernplattform ILIAS zu verbinden.

Am Anfang des Jahres 2014 wurde in der SIG Mathe+ILIAS der Entschluss gefasst, das Open-Source-System STACK mit ILIAS zu verbinden und dafür ein Plug-In zu entwickeln. Das Entwicklungsprojekt wurde in Kooperation von sieben Hochschulen (Albert-Ludwigs-Universität Freiburg, DHBW Karlsruhe, DHBW Mannheim, FH Münster, Friedrich-Alexander-Universität Erlangen-Nürnberg, Hochschule Bremen, Hochschule Karlsruhe – Technik und Wirtschaft) durchgeführt. Technisch umgesetzt wurde das Plug-In an der Friedrich-Alexander-Universität in Erlangen (STACK Question Type 2017).

Die Hochschule Karlsruhe war von Anfang an aktiv an der Testung des Plug-Ins beteiligt; die erste Pilot-Phase mit Studierenden des Studiengangs Infrastructure Engineering der Hochschule Karlsruhe fand bereits im Wintersemester 2014/15 statt. Es folgten zwei weitere Pilot-Semester mit weiteren Lehrenden, bevor im Sommersemester 2016 das STACK-Plug-In auf der zentralen ILIAS-Lernplattform der Hochschule Karlsruhe allen Lehrenden zur Verfügung gestellt wurde. Das Projekt SKATING unterstützte die Verbreitung des Systems an der Hochschule Karlsruhe maßgeblich – durch die Erstellung von Aufgaben und das Betreuen von Online-Übungen, durch Schulungen sowie durch die Akquise und die Einarbeitung von wissenschaftlichen Hilfskräften zur Erstellung von STACK-Aufgaben.

Die im Rahmen des Projekts SKATING entwickelten STACK-Aufgaben wurden zunächst allen Lehrenden der Hochschule Karlsruhe in einem zentralen Fragenpool zur Verfügung gestellt. Später wurden die Aufgaben als Open Educational Resource über das SIG Mathe+ILIAS auf www.ilias.de allen Hochschulen frei zur Verfügung gestellt. Dadurch sind zahlreiche Kooperationen mit anderen Hochschulen entstanden. Zur weiteren Verbreitung des Plug-Ins in Baden-Württemberg wurde im Jahr 2016 ein Workshop zum STACK-Plug-In in das landesweite Weiterbildungsprogramm der GHD aufgenommen.

3. Das Online-Übungssystem STACK

STACK ist ein Akronym aus den Wörtern „System for Teaching and Assessment with Computer-Algebra Kernel". Das System wurde von Chris Sangwin an der University of Birmingham in Großbritannien entwickelt. Die erste Version des Systems entstand im Jahr 2005. Ursprünglich handelte es sich um ein eigenständiges System, aber ab 2013 mit der Version 3.0 wurde STACK als Plug-In mit der Open-Source-Lernplattform Moodle verknüpft. STACK steht ebenfalls als Open-Source-Lösung frei zur Verfügung. (STACK Development History 2017; Sangwin 2013, 102-106) Die STACK-Plug-Ins für Moodle und für ILIAS unterscheiden sich nur an der Benutzeroberfläche – die Funktionalitäten beider Plug-Ins sind identisch. Es besteht daher die Möglichkeit, STACK-Aufgaben problemlos von der einen Lernplattform zu anderen zu exportieren bzw. zu importieren (STACK Question Type 2017).

Das ursprüngliche Ziel für die Entwicklung von STACK war, ein stabiles System für Mathematik-Online-Übungen und -Tests zu entwickeln. Die wichtigste Eigenschaft des Systems ist, dass Studierende mathematische Ausdrücke als Antwort angeben können und die Antworten der Studierenden mithilfe eines Computer-Algebra-Systems auf Korrektheit geprüft werden. STACK benutzt für die Aufgabengenerierung und für das Prüfen der studentischen Antworten das Open-Source-CAS Maxima. Dadurch ergeben sich viele Vorteile gegenüber Übungssystemen ohne CAS-Anbindung (Sangwin 2013, 102-113).

Studierende können nicht nur Zahlen als Antwort angeben, sondern auch mathematische Ausdrücke, wie z. B. Funktionsterme, Gleichungen, Vektoren, Mengen oder Matrizen. Sie geben die Antworten als Freitext ein, ähnlich wie in einen Taschenrechner, können die Antwort-Syntax validieren und bekommen Hinweise zu Syntax-Fehlern sowie eine Vorschau der Antwort als mathematische Formel (Abbildung 1).

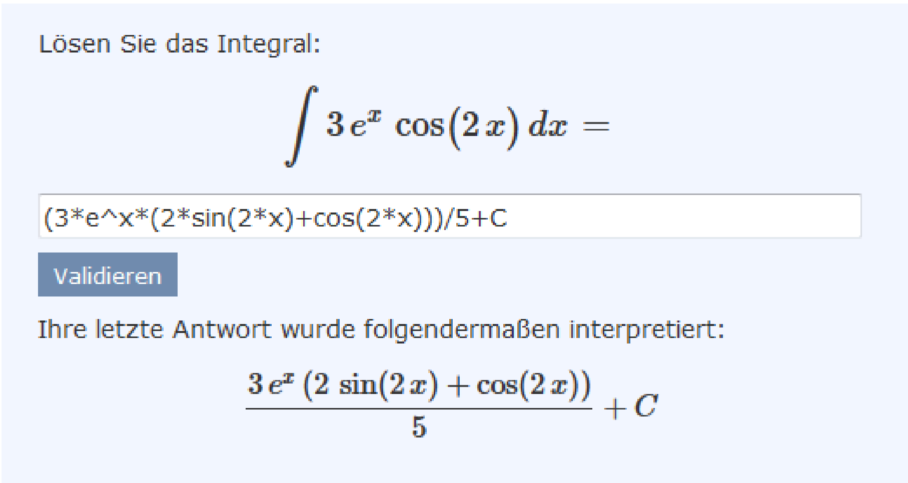


Abbildung 1: STACK-Aufgabe mit studentischer Antwort und Vorschau

Mit STACK können die Antworten der Studierenden automatisch auf Korrektheit geprüft werden, auch in den Fällen, wenn die Aufgabe mehrere oder sogar unendlich viele richtige Lösungen hat (Abbildung 2). Dies wird durch das CAS ermöglicht, womit die mathematischen Eigenschaften der Antwort geprüft werden. Zudem kann den Studierenden ein differenziertes antwortspezifisches Feedback gegeben werden, indem auch auf typische Fehler und Fehlkonzepte eingegangen werden kann (Abbildung 3).

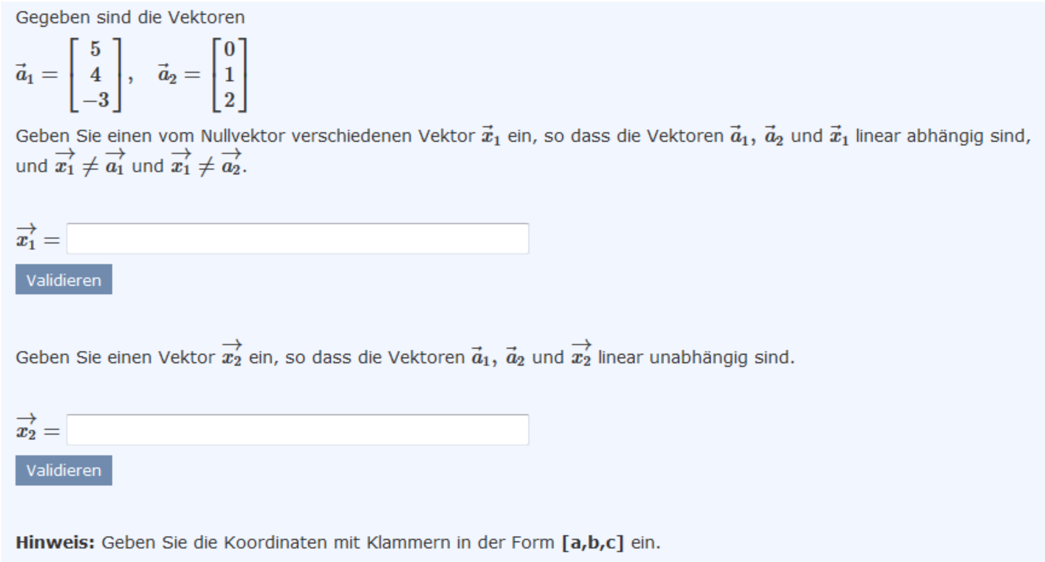


Abbildung 2: STACK-Aufgabe mit unendlich vielen möglichen richtigen Lösungen

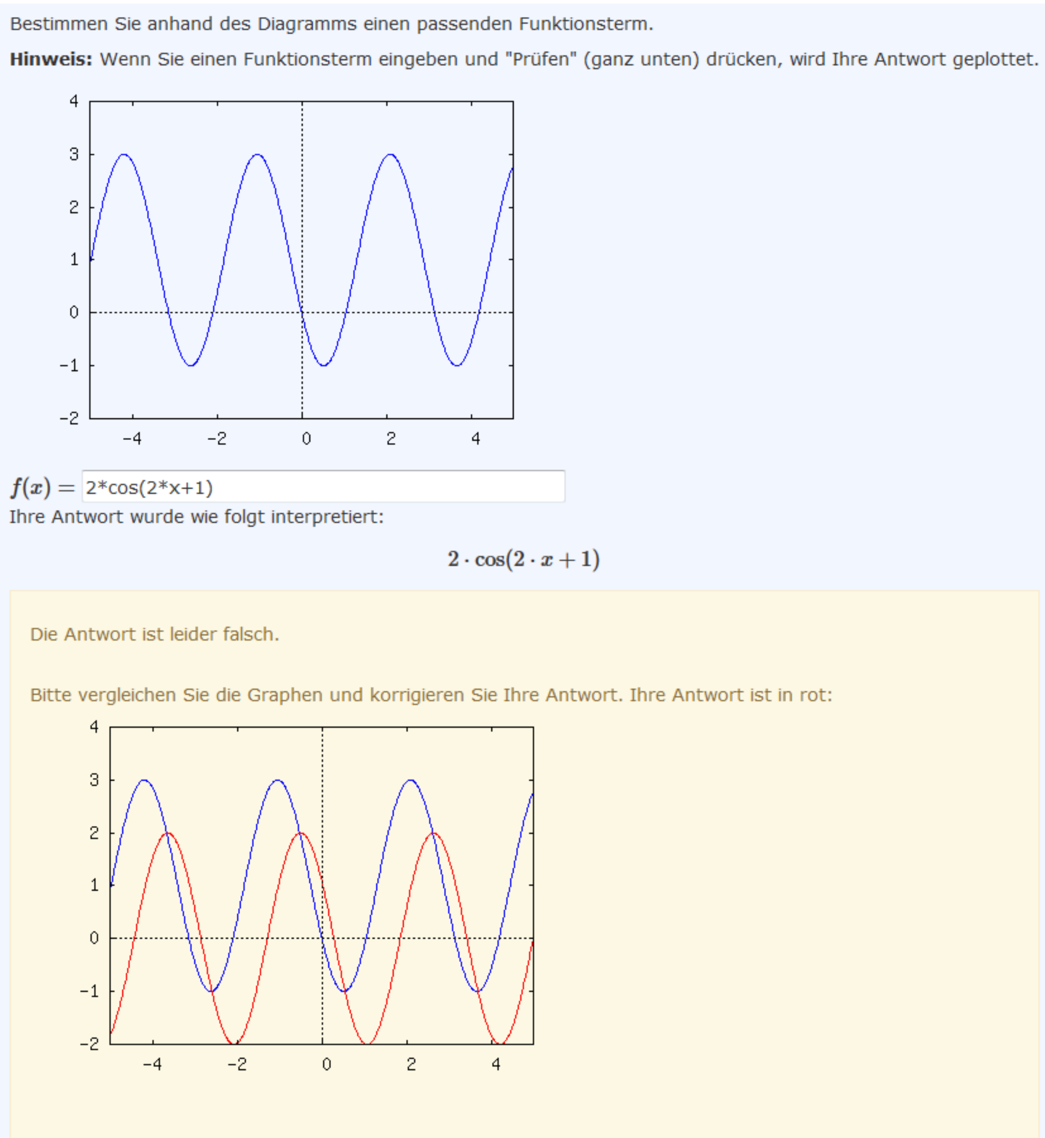


Abbildung 3: STACK-Aufgabe mit antwortspezifischem Feedback

Außerdem können die Aufgaben parametrisiert werden, so dass die Studierenden jeweils ähnliche, jedoch nicht identische Aufgaben erhalten. Dies ist ein deutlicher Vorteil gegenüber Papier-Übungsblättern. Dadurch können die Studierenden sich in Lerngruppen über die Aufgaben und mögliche Lösungswege austauschen, ohne dass sie die richtigen Lösungen von den Mitstudierenden abschreiben können (Sangwin 2013, 5-6).

Das Erstellen der STACK-Aufgaben erfolgt durch eine Eingabemaske in Moodle oder in ILIAS. Die Erstellung kann zunächst etwas kompliziert erscheinen, aber durch eine kurze Impulsschulung können die Anfangshürden schnell überwunden werden. Autoren von STACK-Aufgaben sollten allerdings über mathematische Kenntnisse verfügen sowie mit einem CAS umgehen können.

Da die Erstellung von STACK-Aufgaben mit komplexer Antwortüberprüfung und didaktisch sinnvollem differenzierten Feedback zeitaufwändig sein kann, ist der Austausch von Aufgaben innerhalb von Hochschulen sowie über Hochschulgrenzen hinweg zu befürworten. Ein großer Vorteil von STACK ist hierbei, dass Aufgaben zwischen Moodle- und ILIAS-Installationen problemlos ausgetauscht werden können.

4. Akzeptanz bei den Studierenden und den Lehrenden

An der Hochschule Karlsruhe können Lehrende im Rahmen der Studien- und Prüfungsordnung semesterbegleitende Übungen oder Tests anbieten, mit denen die Studierenden ihre Klausurnote bis zu 10% verbessern können. Diese Regelung ist ein guter Ausgangspunkt, um Online-Übungen curricular zu verankern und die Akzeptanz und Motivation bei den Studierenden zu erhöhen. Tatsächlich zeigen Online-Übungen, die so verankert sind, konstant hohe Teilnehmerzahlen (Tabelle 1).

Tabelle 1: Teilnehmerzahlen in Online-Übungen in Höhere Mathematik 1 in der Fakultät für Elektro- und Informationstechnik an der Hochschule Karlsruhe

Semester	*N*	*% Übungen erfolgreich abgeschlossen*
SoSe 2015	67	70%
WiSe 2015/16	44	61%
SoSe 2106	21	71%
WiSe 2016/2017	35	81%
SoSe 2017	78	70%
Summe	245	70%

Online-Übungen können mit unterschiedlicher didaktischer Konzeption durchgeführt werden. An der Hochschule Karlsruhe wird folgendes Konzept häufig angewendet: Die Studierenden bekommen 1-3 Wochen Zeit, um die Online-Übungsaufgaben zu bearbeiten. Dabei variiert die Anzahl der Aufgaben meist zwischen 5 und 20 Aufgaben. Nach Eingabe einer Antwort bekommen die Studierenden vom System eine sofortige Rückmeldung dazu, ob die jeweilige Antwort richtig oder falsch ist. Sie dürfen eine falsche Antwort anpassen und prüfen lassen, bis die Antwort richtig ist. In manchen Aufgaben werden auch ein differenzierteres Feedback oder Lösungshinweise gegeben. Die Studierenden müssen in der vorgegebenen Zeitspanne 80% bis 100% der Aufgaben richtig gelöst haben, um eine Übung erfolgreich abzuschließen. Wenn die erwartete Anzahl der Übungen erfolgreich abgeschlossen wird, wird die Klausurnote verbessert.

Die Meinungen der Studierenden zu den Online-Übungen wurden mit Fragenbogenverfahren über mehrere Semester hinaus erfasst und die Antworten sind sehr positiv: Die Studierenden finden diese Art von Übung lernförderlich und es macht ihnen sogar Spaß. Die Studierenden der ersten Semester wurden auch gefragt, ob sie auch im zweiten Semester Online-Übungen bearbeiten möchten; 96% (N=91) der Studierenden haben dies bejaht.

Auch bei den Lehrenden stößt das System STACK auf eine positive Resonanz. Dies zeigt beispielsweise die Anzahl der Nutzer. Während zwischen 2013 und 2015 für das System Maple T.A. drei Lehrende gewonnen werden konnten, wird STACK an der Hochschule Karlsruhe bereits 2017 von mehr als 15 Lehrenden in zahlreichen Lehrveranstaltungen angewendet. Mittlerweile beschäftigen sich ca. 70% der Studienanfängerinnen und Studienanfänger an der Hochschule Karlsruhe innerhalb der ersten drei Semester ihres Studiums mit STACK-Aufgaben.

Auch deutschlandweit steigt die Anzahl der STACK-Nutzerinnen und -Nutzer. Nach Recherche des Autors wird STACK an mehr als 17 Hochschulen in Deutschland angewendet (Stand: 11/2017). Weltweit kommen viele Hochschulen hinzu, besonders verbreitet ist STACK auch in Großbritannien und in Finnland (Sangwin 2015).

An der Hochschule Karlsruhe werden STACK-Aufgaben nicht nur in Mathematik-Veranstaltungen angeboten. Auch Elektrotechnik-, Physik- und Technische Mechanik-Aufgaben lassen sich gut mit STACK umsetzen (Abbildung 4).

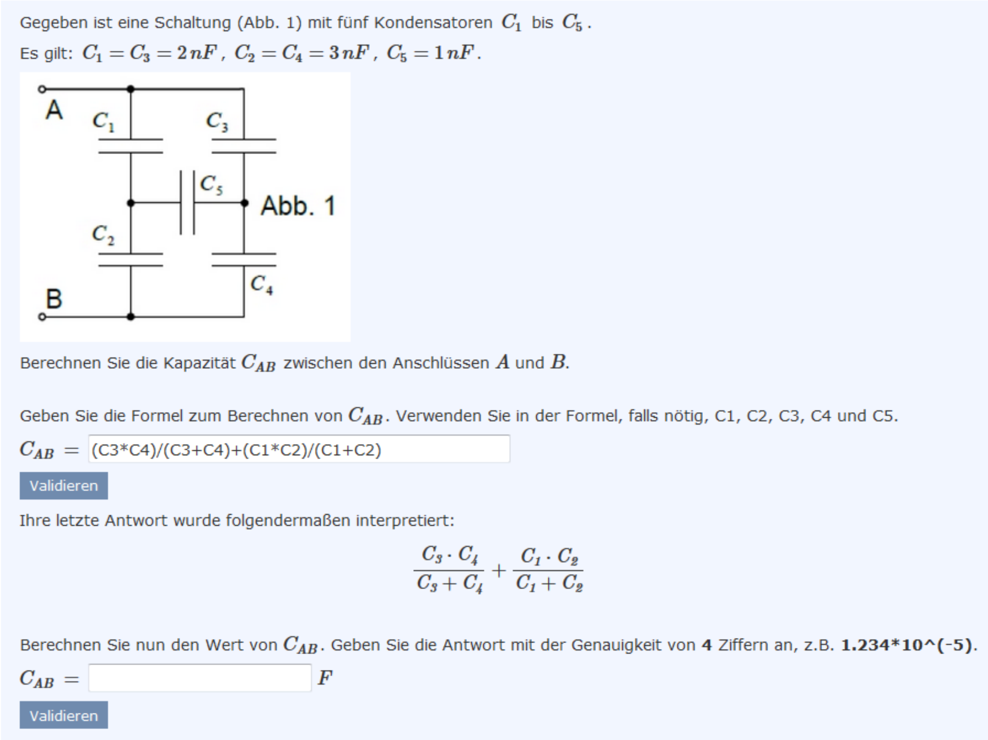


Abbildung 4: Eine STACK-Aufgabe zur Elektrotechnik

5. Fazit

Online-Hausaufgaben können Studierenden positive Lernerfahrungen in der Mathematik geben. Sofortiges Feedback und die Möglichkeit, Aufgaben mehrmals zu beantworten, sind Funktionalitäten, die Studierende sehr schätzen.

Eine Herausforderung bei der Einführung von Online-Übungen liegt für Lehrende in der sinnvollen curricularen Verankerung. Notwendig sind geeignete Anreize für die Bearbeitung der Übungsaufgaben. Es muss für die Studierenden ersichtlich sein, dass sich der Aufwand lohnt. Die Bonierung kann ein Anreiz sein, empfehlenswert ist aber auch die durchdachte inhaltliche und zeitliche Abstimmung von Übungen mit den Abschlussklausuren. Eine weitere Herausforderung liegt in der Erstellung von Aufgaben, die nicht das direkte Anwenden von Formeln, sondern ein tiefergehendes mathematisches Verständnis fördern. Zur Erstellung solcher Aufgaben sind Kooperationen mit anderen Lehrenden und Hochschulen sehr hilfreich.

Das Open-Source-Übungssystem STACK existiert als Plug-In für die Lernplattformen Moodle und ILIAS und bietet eine Vielfalt an Funktionalitäten, die es erlauben, Online-Aufgaben nicht nur für Mathematik, sondern auch für technische Fächer zu erstellen und bereitzustellen. Ein großer Vorteil des Systems ist, dass Aufgaben problemlos durch Export/Import zwischen den zwei Lernplattformen ausgetauscht werden können. Daher ist STACK ein geeignetes System, um den hochschulübergreifenden Austausch sowie Kooperationsprojekte zwischen den Hochschulen zu fördern.

Literaturverzeichnis

1. Abel H. und Weber B., 28 Jahre Esslinger Modell – Studienanfänger und Mathematik. In: Mathematische Vor- und Brückenkurse, Konzepte und Studien zur Hochschuldidaktik und Lehrerbildung in Mathematik, edited by Isabell Bausch et al., Springer Fachmedien: Wiesbaden 2014, 9 – 19.
2. Berendes, J., Engelbrecht, D., Metzger, G., Vasko, M., Voss, H.-P. und Zellner, M., Projekt SKATING, Studienreformprozess Karlsruhe zur Transformation des Ingenieurstudiums, Schlussbericht der ersten Förderphase, September 2011 - August 2016. Rektorat der Hochschule Karlsruhe - Technik und Wirtschaft: Karlsruhe 2017. https://www.hs-karlsruhe.de/fileadmin/hska/SCSL/SKATING/SKATING-Schlussbericht.pdf
3. Bremer, C., Krömker, D. und Voss, S., Wirtschaftlichkeits- und Wirksamkeitsanalysen sowie Vorgehensmodelle zur Einführung und Umsetzung von E-Learning an Hochschulen. In: E-Learning in Hochschule und Weiterbildung, Hrsg. R. Holten und D. Nittel. W. Bertelmann Verlag: Bielefeld 2010, 61-80.
4. Heck, A., Assessment with Maple T.A.: Creation of Test Items.
5. Maple T.A., https://de.maplesoft.com/products/mapleta/, gelesen am 24.11.2017
6. Sangwin, C.J., Computer Aided Assessment of Mathematics, Oxford University Press: Oxford 2013.
7. Sangwin, C.J., Who uses STACK? A survey of users of the STACK CAA system, May 2015, Loughborough University: Loughborough 2015. https://dspace.lboro.ac.uk/dspace-jspui/bitstream/2134/18540/1/2015-STACK-Report.pdf
8. Schulmeister, R. und Metzger C., Die Workload im Bachelor: Ein empirisches Forschungsprojekt. In: Die Workload im Bachelor: Zeitbudget und Studierverhalten. Eine empirische Studie, Hrsg. R. Schulmeister and C. Metzger. Waxmann: Münster 2011, 13 – 128.
9. STACK Development History, https://github.com/maths/moodle-qtype_stack/blob/master/doc/en/Developer/Development_history.md, gelesen am 24.11.2017.
10. STACK Question Type, https://www.ilias.de/docu/goto_docu_cat_4119.html, gelesen am 24.11.2017.

Autor

Mikko Vasko
Projekt SKATING
Hochschule Karlsruhe – Technik und Wirtschaft
Moltkestr. 30
D-76133 Karlsruhe
E-Mail: mikko.vasko@hs-karlsruhe.de

Jürgen Vorloeper

`Julia` in ingenieurwissenschaftlichen Studiengängen

Auszug. In vielen ingenieurwissenschaftlichen Studiengängen gehört neben der Grundlagenmathematik auch die Ausbildung in mathematischen Simulationsmethoden und Programmierung mit zum Curriculum. In jüngerer Zeit hat die frei zugängliche und moderne Programmiersprache `Julia` im Bereich des wissenschaftlichen Rechnens große Beachtung erfahren. Sie verbindet Einfachheit in der Handhabung mit der Performanz klassischer Sprachen. In diesem Artikel wird der Frage nachgegangen, welche Chancen und Herausforderungen der Einsatz von `Julia` in ingenieurwissenschaftlichen Studiengängen bietet. Neben einigen technischen Aspekten der Programmiersprache werden unterrichtspraktische Beispiele vorgestellt.

Was ist `Julia`?

`Julia` ist eine hochperformante, dynamische, funktionsorientierte Programmiersprache, die seit 2012 am MIT entwickelt wurde, siehe [2, 3, 5]. Das Programm ist Opensource mit MIT/GPL Lizenz. `Julia` ist derzeit eine stark wachsende Programmiersprache, die sich durch moderne Sprachelemente auszeichnet. Die Syntax ist ähnlich zu der von Matlab. Die Sprache verwendet einen Just–In–Time–Compiler basierend auf LLVM [7].

Das folgende Code–Beispiel illustriert einige Gemeinsamkeiten und Unterschiede in der Syntax von `Julia` und Matlab.

```
# Solution of Poisson Equation -\Delta u=1, u(0)=u(1)=0

A = [
 2 -1  0   0;
-1  2 -1   0;
 0 -1  2  -1;
 0  0 -1  2.0
]

b = 1/5 * ones(4)

y = A\b

import PyPlot
PyPlot.plot(collect(0.0:0.2:1.0),[0.0; y; 0.0])
```

Das Aufstellen von Matrizen und Vektoren, das Lösen eines linearen Gleichungssystems, Kommentare und die grafische Ausgabe funktionieren in beiden Sprachen nahezu gleich. Gleichwohl lassen sich feine Unterschiede festhalten. In Zeile 7 bewirkt die `2.0`, dass `A` vom Typ `Array{Float64,2}` ist, an Stelle von `Array{Int64,2}`, was beim Eintrag 2 der Fall gewesen wäre. In `Julia` gibt es einen Unterschied zwischen Vektoren und Range–Objekten, den es in Matlab in dieser Form nicht gibt. In Zeile 15 überführt der `collect` Befehl ein Range–Objekt in einen Vektor. Im Vergleich zu Matlab zeichnet sich `Julia` durch eine ausgeprägte Typisierung aus. Der `import` Befehl in Zeile 14 bindet das Paket `PyPlot` ein. Zahlreiche `Julia` Pakete werden auf GitHub gehostet. Der integrierte Paketmanager ist eng mit GitHub verbunden.

Elementar kann man `Julia` über eine Kommandozeile oder über ein Befehlsfenster bedienen. Als Entwicklungsumgebung lässt sich der Editor Atom [1] einsetzen, der mit dem Paket `uber-juno` für `Julia` konfiguriert wird. `Atom` ist seit 2014 verfügbar und ist damit ein junger Editor. Alternativ kann man `Julia` mit dem webbasierten Jupyter Notebook [6] verwenden.

Eine hohe Hürde stellt in `Julia` die Benutzung eines Debuggers dar. Während es für andere Sprachen eine umfangreiche und benutzerfreundliche Unterstützung in verschiedenen Entwicklungsumgebungen gibt, erfordert die Verwendung eines Debuggers in `Julia` Expertenwissen.

Die Installation von `Julia` kann lokal ohne Administrator–Rechte erfolgen. Alternativ ist eine Installation auf einem USB–Stick möglich. Festzustellen ist, dass die automatische Installation einiger weniger `Julia` Pakete noch nicht ganz zuverlässig verläuft und es daher gelegentlich zu Situationen kommen kann, die ein manuelles Eingreifen erfordern. Als Zielsysteme werden Windows, Linux und macOS unterstützt.

Häufige Paket–Updates in `Julia` weisen auf eine aktive Entwicklungsgemeinde hin. Im Zuge der Weiterentwicklung werden noch gelegentlich Änderungen an zentralen Schnittstellen vorgenommen.

`Julia` in der Lehre

Die folgende Ausführungen beruhen auf Erfahrungen, die der Autor in den Modulen „Wissenschaftliche Simulation“ in technischen Masterstudiengängen an der Hochschule Ruhr West gesammelt hat. Die Studierenden in diesen Veranstaltungen brachten erste Programmiererfahrungen in verschiedenen Sprachen (häufig C, Java oder Matlab) mit.

Lernergebnisse und Studiengangsziele

Die Ausgestaltung einer Lehrveranstaltung erfolgt mit einem Abgleich von anvisierten Lernergebnissen und Studiengangszielen. Hierzu gehören in diesem Veranstaltungsrahmen, dass die Studierenden…

- numerische Methoden alleine und im Team auf konkrete Probleme aus Technik und Naturwissenschaften anwenden können,
- numerische Methoden auswählen, in modernen Softwaresystemen realisieren und die Ergebnisse bewerten können,
- ihre Arbeitsergebnisse fachgerecht kommunizieren, sowohl mündlich wie schriftlich,
- Verbesserungspotentiale in technischen Systemen identifizieren und daraus Maßnahmen ableiten können.

Die genannten Lernergebnisse sind grundsätzlich mit einer Vielzahl von Programmierwerkzeugen zu erreichen. Vielerorts kommen eine oder mehrere etablierte Programmiersprachen zum Einsatz, etwa Fortran, C/C++, Python. Zu nennen sind auch Matlab, Octave, Scilab sowie Mathematica und Maple. Der Einsatzkontext `Julia` ist am ehesten mit denen von Matlab, Octave oder Scilab zu vergleichen.

Im oben genannten Veranstaltungsrahmen ist die Programmiersprache kein eigenes Lernziel – anders als in dezidierten Programmierkursen beispielsweise. Gleichwohl unterstützt die Auswahl einer Programmierumgebung idealerweise das Erreichen der Lernergebnisse. Aus diesem Betrachtungswinkel heraus bringt `Julia` Chancen und Herausforderungen für Lernende und Lehrende mit.

Einsatz von `Julia` im Semesterverlauf

Im Folgenden wird ein möglicher Aufbau einer einsemestrigen Veranstaltung skizziert.

- Als Auftakt wird `Julia`/`Atom` zu Semesterbeginn lokal installiert. Dies fördert die Autonomie und Selbstverantwortung der Studierenden und es entsteht von allein Kommunikation und Teambildung bei technischen Problemen.
- Bei der Behandlung des Themengebietes Lineare Optimierung werden die Grundfunktionen (Vektoren, Matrizen, Funktionen,…) eingeführt. Zur Lösung der Optimierungsprobleme können übersichtliche `Julia` Pakete verwendet werden, etwa `GLPK` oder auch eine vorbereitete selbst entwickelte Schnittstelle hierzu. Es wird dabei die Verwendung des Paketmanagers eingeübt sowie die Verwendung von externen Modulen.
- Die programmiertechnische Realisierung der Lösung von (nicht-) linearen Gleichungssystemen und gewöhnlichen Differentialgleichungen können sich die Studierenden mit stufenweiser Anleitung anhand der Dokumentation erarbeiten. Zugleich wird die Verwendung umfangreicherer `Julia` Pakete eingeübt.
- Die Behandlung partieller Differentialgleichungen kann nur in einem einführenden Rahmen erfolgen. Die selbstständig geschriebenen Programmcodes werden deutlich umfangreicher, so dass nach und nach Wert auf ein modulares Design gelegt werden sollte. Die Studierenden arbeiten zunehmend selbstständig, kommunizieren fachliche (Teil-) Ergebnisse untereinander sowie vor der Lerngruppe.
- Je nach Schwerpunktsetzung der Veranstaltung bieten sich zum Abschluss Vertiefungen einzelner Themen an, etwa die Verwendung von Versionsverwaltungssystemen, paralleles Rechnen oder Integration und Interaktion von `Julia` mit anderen Softwaresystemen.

Blick über den Tellerrand

Als junge Progammiersprache bietet `Julia` zahlreiche Möglichkeiten für ein selektives Ergänzen oder Vertiefen entsprechend den Interessen und Vorkenntnissen der Studierenden. Damit wird ein binnendifferenziertes Unterrichten unterstützt. Im Folgenden werden einige ergänzende Themengebiete vorgestellt:

- Das Arbeiten mit GitHub gibt einen Einblick in ein System für Software–Versionsverwaltung und ermöglicht kollaboratives Arbeiten.

- Die Softwaredokumentation in GitHub erfolgt in der Regel mit Markdown [8]. Markdown ist eine einfache Auszeichnungssprache, in der die Textformatierung durch einfache Elemente erfolgt und die zugleich eine leichte Lesbarkeit der Textdateien unterstützt.

- Für einen ersten Blick in die Interna von `Julia` und einigen Paketen eignen sich die BLAS/Lapack–Routinen oder die hinter PyPlot stehende Bibliothek matplotlib [9]. Letztere liefert einen Anknüpfungspunkt zur Programmiersprache Python.

- Die Bordmittel von `Julia` ermöglichen paralleles bzw. verteiltes Rechnen. Dies ist ein weites Feld, für das man mit einfachen Progammbeispielen einen ersten Eindruck vermitteln kann.

- Ein weiteres ergänzendes Thema ist die Interaktion von `Julia` mit anderen Softwaresystemen. Neben Beispielen zur Anwendung der C–Schnittstelle können auch grundlegende Konzepte des Betriebssystems zur Interprozesskommunikation behandelt werden. Zahlreiche `Julia` Pakete können als Anschauungsmaterial dienen.

Eigene Erfahrungen

Im folgenden soll kurz auf eigene unterrichtspraktische Erfahrungen des Autors eingegangen werden.

Während viele Programmiersprachen den Studierenden zumindest vom Namen her bekannt sind, teilweise auch schon erste praktische Erfahren vorliegen, ist `Julia` für die Studierenden etwas ganz Neues und muss entsprechend motiviert werden. Die Performanz der Sprache ist in der Lehre nachrangig. Vielmehr lassen sich die Studierenden dadurch motivieren, das etwas Neues entdeckt wird. In den Kursen des Autors waren auch Studierende, die nicht Informatik studieren, der Sprache gegenüber äußerst aufgeschlossen. Hilfreich ist die Anregung zur Reflexion über die Einsatzszenarien von `Julia` sowie ein Vergleich zu einem etablierten System wie z. B. Matlab.

Neben der englischsprachigen Originaldokumentation gibt es im Internet zahlreiche (Video-) Tutorials und Diskussionsgruppen. Allerdings gibt es nur sehr wenige deutschsprachige Lehrbücher, die `Julia` behandeln. Ein Grund hierfür ist, dass `Julia` eine junge Sprache ist und derzeit noch zahlreichen Änderungen und Anpassungen unterworfen ist. Dies bedingt eine phasenweise intensive Lehrbegleitung und eine gute Auswahl der Begleitmaterialien.

An `Julia` angrenzende Technologien und der Zugang zu Interna der Sprache unterstützen binnendifferenziertes Unterrichten und kollaboratives Arbeiten.

Im Lehralltag ist mit `Julia` die Anwendung von etablierten Konzepten möglich und sinnvoll. Zugleich eröffnet es ein Experimentierfeld zum Einsatz von neuen und jungen Technologien.

Literaturverzeichnis

[1] `https://atom.io/`, abgerufen am 24. 11. 2017.

[2] **Bezanson, J., Karpinski, S., Shah, V.B., Edelman, A.**: *Julia: A Fast Dynamic Language for Technical Computing*, preprint, arXiv:1209.5145 [cs.PL], 2012.

[3] **Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.,** : *Julia: A Fresh Approach to Numerical Computing.* (2017) SIAM Review, 59: 65–98.

[4] **Bornemann, F.**: *Numerische lineare Algebra*, Springer Spektrum, 2016.

[5] `http://www.julialang.org`, abgerufen am 13. 12. 2017.

[6] `http://jupyter.org/`, abgerufen am am 13. 12. 2017.

[7] `http://www.llvm.org/`, abgerufen am am 13. 12. 2017.

[8] `https://daringfireball.net/projects/markdown/`, abgerufen am am 13. 12. 2017.

[9] `https://matplotlib.org/`, abgerufen am am 13. 12. 2017.

Autor

Prof. Dr. rer. nat. Jürgen Vorloeper
Institut Naturwissenschaften
Hochschule Ruhr West
Duisburger Straße 100
D–45479 Mülheim an der Ruhr
E–Mail: juergen.vorloeper@hs-ruhrwest.de

Claudia Frohn-Schauf und Joachim Fulst

Integration von MATLAB$^{\text{TM}}$ in die Übungen zur Mathematik für Ingenieure - ein Erfahrungsbericht

Zusammenfassung: MATLAB zur parallelen Lösung der handschriftlichen Übungsaufgaben in der Mathematik einzusetzen, ist ein Ansatz, den Umgang mit einem praxisrelevanten Computeralgebra-System früh einzuführen und durch einen anderen Zugang das Verständnis für Mathematik zu verbessern. Die im Fachbereich Mechatronik und Maschinenbau der Hochschule Bochum umgesetzte Form und die gemachten Erfahrungen werden in diesem Beitrag vorgestellt.

Idee und Ziele

Die Mathematik in ingenieurwissenschaftlichen Studiengängen wird von den Studierenden häufig als praxisfern und in ihrer Form geradezu überflüssig wahrgenommen. Um diesem Eindruck entgegenzuwirken, wird im Fachbereich Mechatronik und Maschinenbau der Hochschule Bochum seit dem Sommersemester 2011 das Computeralgebra-System MATLAB in die Übungen zur Veranstaltung Mathematik 2 integriert. Wesentliche Zielsetzungen sind dabei,

- den Umgang mit praxisrelevanter Software frühzeitig einzuüben,
- durch einen anderen Zugang zur Mathematik Verständnis und Interesse zu fördern,
- durch die Möglichkeiten einer Software-Umgebung mit einem Computeralgebra-System mehr Begeisterung für Mathematik und Informatik zu wecken und
- den Wunsch, selbst mehr zu lernen und zu können, hervorzurufen.

Zielgruppe und Rahmenbedingungen

In den Bachelorstudiengängen Maschinenbau, Mechatronik, Wirtschaftsingenieurwesen mit Vertiefungsrichtung Maschinenbau im 2. Semester und in den dualen Studiengängen Mechatronik und Maschinenbau im 3. Fachsemester steht die Veranstaltung Mathematik 2 mit einem Praktikum und einer Übung im Studienverlaufsplan. Da die Studierenden im ersten Semester bereits Matrizen kennengelernt haben, bietet es sich an, im zweiten Semester mit dem Einsatz der Software MATLAB zu beginnen. Die Hochschule Bochum hat eine aktuelle Campus-Lizenz (R2016a/R2017a), in deren Rahmen insbesondere die Symbolic Toolbox zur Verfügung steht, so dass die Studierenden auch von zu Hause aus Zugriff auf die Software haben. Da in Kleingruppen von etwa 15 Studierenden gearbeitet wird, sind ausreichende Kapazitäten an Rechnerräumnen erforderlich.

Im Rahmen der Campus-Lizenz können die Studierenden auf alle Online-Kurse und Unterlagen der Firma MathWorks zugreifen. Es hat sich jedoch gezeigt, dass es sinnvoll ist, eine eigene Einführung in MATLAB zu erstellen, die ganz konkret diejenigen Befehle einführt und auf die Themen vorbereitet, die im folgenden Semester benötigt werden. Diese Unterlage („Einführung und Aufgaben zum Blockpraktikum“) ist die Basis für einen 8-stündigen Blockkurs (2x4 UE) vor Vorlesungsbeginn und dient den Studierenden als Nachschlagewerk im laufenden Semester.

Abgestimmt auf die Vorlesungen Mathematik I und Mathematik II existieren bereits Aufgabensammlungen (mit Ergebnissen), die ebenfalls als Basis für das MATLAB-Praktikum dienen. Denn im Praktikum sollen die gleichen Aufgaben behandelt und berechnet werden wie in den Übungen zu Mathematik II. Dadurch soll das Verständis für die Aufgaben und ihre Lösung verbessert werden; Vor- und Nachteile des Einsatzes von Software sollen deutlich werden. Zudem wird die Interpretation mit MATLAB berechneter Lösungen eingeübt.

Während des Semesters findet das Praktikum 14-tägig mit zwei Unterrichtseinheiten (1 SWS) statt. Zu Erlangung des Testats ist Anwesenheit erforderlich (ein Fehltermin und Nachholtermine sind möglich). In jeder Praktikumseinheit muss die erfolgreiche Bearbeitung einer Pflichtaufgabe nachgewiesen werden, zu der die Studierenden auch Fragen beantworten können müssen, die sich auf den mathematischen Hintergrund beziehen. Einmal im Semester muss eine sogenannte Testataufgabe maximal in einer Zweier-Gruppe bearbeitet werden. Dazu gibt es eine Anleitung und Musterbeispiele. Zu dieser Testataufgabe, die per E-Mail innerhalb von zwei Wochen eingereicht werden muss, gehört neben dem Programmcode auch eine ausführliche Dokumentation des Vorgehens in MATLAB mit einer Beschreibung der verwendeten Befehle, eine Erörterung des mathematischen Hintergrundes und Zusammenhangs sowie gegebenenfalls die erforderliche Interpretation der Ergebnisse (z.B. bei nicht eindeutig lösbaren linearen Gleichungssystemen).

Kursinhalte und Beispielaufgaben

Folgende Themenfelder werden im Praktikum behandelt, wobei die Berechnungen auch unter Verwendung der Symbolic Toolbox stattfinden:

- Nullstellen und Linearfaktorenzerlegung von Polynomen, Grenzwerte, Ableitungen, Integrale, Matrizen, Determinanten (Wiederholungen aus Mathematik I)
- Lösung linearer Gleichungssysteme (mit Parametern)
- Komplexe Zahlen und Zeiger
- Kugeln und Kreise
- Flächenberechnungen mit Integralen
- Funktionen von zwei Variablen, implizite Funktionen
- Funktionen in Polarkoordinaten und Parameterdarstellung, Vektorfunktionen
- Differentialgleichungen erster und zweiter Ordnung sowie Richtungsfelder

Bei nahezu allen Aufgaben wird besonderer Wert auf eine Visualisierung gelegt.

Alle Aufgabenblätter sind einheitlich in die Teile „Kurzdarstellung bekannter und neuer MATLAB-Befehle“, „Aufgabenstellungen“, in denen Bezug zu den „gewöhnlichen“ Übungsaufgaben der Mathematik II Veranstaltung genommen wird, sowie wöchentlich zu erledigende „Pflichtaufgabe“ gegliedert. Eine Auswahl an Testataufgaben wird auf einem gesonderten Blatt gestellt. Die Anleitung und eine Musterdokumentation mit m-files stehen im zugehörigen Moodle-Kurs zur Verfügung.

Beispiel aus einer Aufgabenserie zu Analysis II:

Aufgabe 104

a) *Man berechne y' und y'' zu der Funktion*

$$\begin{aligned} x &= \sin(t) - t\cos(t) \\ y &= \cos(t) + t\sin(t) \end{aligned}$$

b) *Gegeben sei die Funktion in Parameterdarstellung*

$$\left.\begin{aligned} x &= \cos^2(t) \\ y &= \sin^3(t) \end{aligned}\right\} \textit{für } -\frac{\pi}{2} \leq t \leq \frac{\pi}{2}$$

Man berechne eine Wertetabelle der Funktion für $-\frac{\pi}{2} \leq t \leq \frac{\pi}{2}$ *in Schritten von* $\frac{\pi}{6}$ *und zeichne eine Skizze der Funktion* ($1LE \mathrel{\widehat{=}} 5cm$).

Man berechne y' *und die Gleichung der Tangente im Punkt* $P_0(\frac{1}{4}, \frac{3\sqrt{3}}{8})$. *Hat die Kurve waagerechte Tangenten? Wenn ja, wo?*

Plotten von Funktionen in Parameterdarstellung

Einen Kreis mit Radius R, der in Parameterdarstellung gegeben ist ($x(t) = R\cos(t), y(t) = R\sin(t)$), kann man mit den folgenden Befehlen plotten:

```
t=0:pi/100:2*pi;         % Die Schrittweite wird als Anteil von pi
                         % gewaehlt, damit man keine Luecke in
                         % der Kreislinie erhaelt; oder:
t=linspace(0,2*pi);
R=4;                     % Kreis mit Radius 4.
x=R*cos(t); y=R*sin(t);
plot(x,y);
axis off                 % unterbindet die Anzeige der Achsen.
```

Alternativ plottet `ezplot(x,y)` auch Funktionen in Parameterdarstellung, wenn `x` und `y` als *anonymous functions,* symbolische Ausdrücke oder symbolische Funktionen definiert sind.

Abb. 1: Kurzdarstellung bekannter MATLAB-Befehle aus dem ersten Teil eines MATLAB-Aufgabenblattes

Aufgabe 25 (Programmierung einer MATLAB function):

Schreiben Sie eine MATLAB function, die zu einer Funktion in Parameterdarstellung die Ableitungsfunktion y' (als symbolische Funktion) sowie die Steigung m und den y-Achsenabschnitt b der Tangente an einer vorgegebenen Stelle t_0 zurückgibt.
Die MATLAB function soll gleichzeitig sowohl die Funktion als auch die Tangente in **ein** Diagramm zeichnen. Testen Sie Ihre function mit den Kurven aus den Aufgaben 104 b) und c) zur Analysis.

Abb. 2: Aufgabenstellung aus dem zweiten Teil desselben MATLAB-Aufgabenblattes

Durchführung und Erfahrungen

Das integrierte MATLAB-Praktikum wird nunmehr seit mehr als sechs Jahren als Pflichtfach in verschiedenen Varianten (mit/ohne Pflichtaufgaben, mit/ohne Testataufgaben) durchgeführt. Gefordert wird stets eine Anwesenheitspflicht mit „aktiver Teilnahme".

Ohne eine Kontrolle über die Erledigung von Pflichtaufgaben oder Testataufgaben entstand der Eindruck, dass einige Studierende das Testat ohne die gewünschten MATLAB-Kenntnisse erhalten. Der größte Lernerfolg ist erwartungsgemäß dann zu erzielen, wenn sowohl Pflicht- als auch Testaufgaben (größtenteils mit Korrekturhilfe) gelöst werden müssen. Bei der vergleichsweise kleinen Gruppe der dual Studierenden führte dieses Vorgehen zu guten Ergebnissen. Allerdings ist der Personalaufwand hierfür erheblich und überschreitet im Normalfall die zur Verfügung stehenden Lehrkapazitäten. Eine Steigerung der intrinsischen Motivation, sich gegebenenfalls auch zu Hause mit Mathematik „auseinanderzusetzen", konnte nur bei wenigen Studierenden beobachtet werden. Erschwerend kommt hinzu, dass laut geltenden Studienverlaufsplänen das Erlernen einer Programmiersprache in Informatik in den grundständigen Studiengängen erst im gleichen Semester vorgesehen ist. Eine Verbesserung der Programmierfähigkeiten zum Ende des Semesters hin konnte jedoch nicht festgestellt werden.

Offene Fragen

- Wie kann man die intrinsische Motivation der Studierenden steigern?
- Welches Format (Pflicht-, Testataufgaben, . . . ?) kann den Lernerfolg auch ohne zusätzliches Lehrpersonal sicherstellen?
- Ist es eventuell sinnvoll, MATLAB als erste Programmiersprache (statt Java oder C) einzuführen?

Autoren
Prof. Dr. rer. nat. Claudia Frohn-Schauf
Prof. Dr. rer. nat. Joachim Fulst
Fachbereich Mechatronik und Maschinenbau
Hochschule Bochum
Lennershofstraße 140
D-44801 Bochum

email: claudia.frohn-schauf@hs-bochum.de, joachim.fulst@hs-bochum.de

Johannes Hild, Wigand Rathmann

Optimiale Steuerung von Flachwasserkanälen

Auszug. Die Optimalsteuerung von Prozessanlagen spielt in vielen technischen Bereichen eine äußerst wichtige Rolle. Hierbei wird grundsätzlich unterschieden zwischen der **closed-loop-Steuerung**, die eine Messung des Zustands der Prozessanlage in Echtzeit für die Steuerentscheidungen verwertet, und der **open-loop-Steuerung**, deren Steuerentscheidungen auf Vorhersagen basieren, die aus einer (möglichst genauen) rechnergestützten Simulation der Prozessanlage resultieren. Die hier vorgestellte Flachwasserkanalanlage wurde konzipiert, um die Funktionsweise von Steuerentscheidungen erlebbar machen.

Die Aufgabenstellung

Kommunale Abwasserkanalsysteme sind in Deutschland flächendeckend eingerichtet, um Haushalts- und Industrieabwässer für die umweltschonende Aufbereitung in Klärwerke zu entlasten. In sogenannten Mischwassersystemen wird außerdem auch Regenwasser abgeleitet, vgl. Abbildung 1. Die Nutzung von Mischwassersystemen hat den Vorteil, dass durch regelmäßige Regenereignisse Schmutzablagerungen an den Abwasserkanälen reduziert werden. Bei Starkregen kann sich jedoch der Nachteil ergeben, dass belastetes Wasser in regionale Gewässer entlastet werden muss. Gemäß europaweiten Richtlininen zur Verbesserung der Gewässerqualität ist die Entlastung von Schmutzwasser in die Umwelt zu vermeiden (siehe auch [5]).

Mit Hilfe von mathematisch fundierten Steuerentscheidungen sollen Abwässer auch bei starken Regenereignissen durch den Einsatz von steuerbaren Elementen (Wehre, Pumpen) so geführt werden, dass unter Ausnutzung der maximalen Auslastung der Speichermedien (Kanäle, Sammelbecken) eine Entlastung von Schmutzwasser in die Umweld vermieden oder zumindest minimiert wird. Dies wird bereits im Rahmen von closed-loop-Steuerstrategien umgesetzt, wobei die Steuerentscheidungen auf Grund der Überwachung der Wasserströme mit Durchfluss- und Höhenstandsmessern erfolgt.

Die in diesem Artikel vorgestellten Methoden zielen darauf ab, die bereits existierenden closed-loop-Steuerstrategien mit open-loop-Konzepten

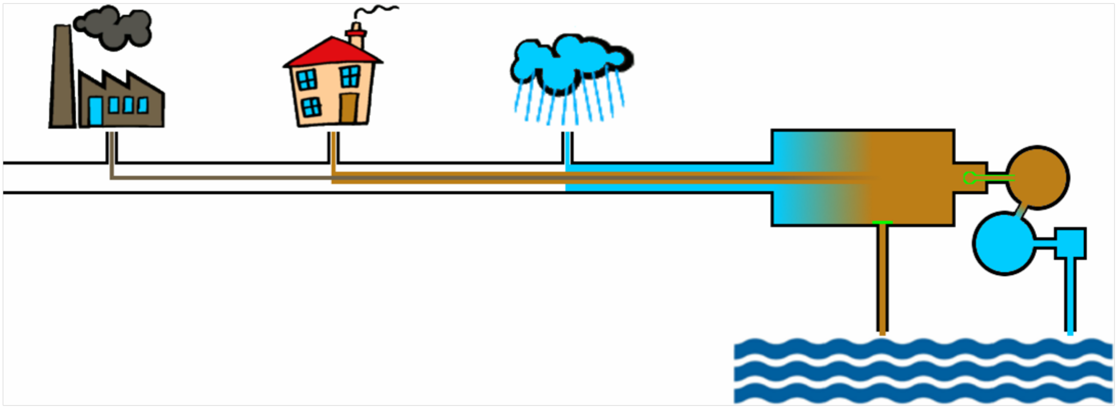


Abb. 1: Schematische Darstellung von Mischwassersystemen

zu erweiteren. Diese Konzepte sehen zunächst vor, das zu steuernde Kanalnetzwerks als dynamisches System mit Anfangs- und Randbedingungen zu modellieren und dieses Modell für Simulationen zu nutzen. Mit Hilfe von gradientenbasierten Optimierungsmethoden sollen optimale Steuerentscheidungen für das Modell generiert werden, die wiederum auf das real existerende Kanalnetzwerk angewendet werden sollen. Um den Einsatz der Modelle in Echtzeit zu ermöglichen, sollen die Anfangs- und Randbedingungen auf aktuellen Messungen basieren und Simulationen nur über einen kurzen Zeithorizont erfolgen. Die Entwicklung und Untersuchung dieses modellprädiktive Steuerungsverfahren wird in [3] und in [2] im Detail behandelt.

Modellierung mit Flachwassergleichungen

Die in den Kanalsystemen zu simulierenden Wasserströme werden mit Hilfe der Flachwassergleichungen modelliert. Das Wasser verfügt also zu jeder Zeit t in einem (kurzen) Zeitintervall $T = [t_0, t_1]$ und an jedem Ort x eines eindimensionalen Kanals $X = [x_a, x_b]$ einen zweidimensionalen Zustand, wobei die erste Komponente des Zustands die Wasserquerschnittsfläche $A \in \mathbb{R}^+$ und die zweite Komponente die Flussrate $Q \in \mathbb{R}$ beschreibt. Wichtige Funktionen des Wasserzustands sind der hydrostatische Druck $\eta : \mathbb{R}^+ \to \mathbb{R}^+$ und die Reibungsfunktion $S : \mathbb{R}^+ \times \mathbb{R} \to \mathbb{R}$.

Die Flachwassergleichungen lauten dann allgemein:

$$\begin{aligned} \frac{\partial}{\partial t}A(t,x) + \frac{\partial}{\partial x}Q(t,x) &= 0 && (1)\\ \frac{\partial}{\partial t}Q(t,x) + \frac{\partial}{\partial x}\left(\frac{Q(t,x)^2}{A(t,x)} + \eta(A(t,x))\right) &= S(A(t,x),Q(t,x)). && (2) \end{aligned}$$

Die Flachwassergleichungen bilden ein hyperbolisches System von Erhaltungsgleichungen, deren Lösbarkeit nicht im Allgemeinen gegeben ist. Eine weitere Herausforderung bei der Anwendung für Kanalwassersysteme besteht darin, dass ein solches Netzwerk aus einer Vielzahl von Kanälen mit verschieden Längen, Durchmessern und Reibungsfunktionen besteht. Die Flachwassergleichungen müssen für eine erfolgreiche Simulation auf jedem einzelnen Kanal gelöst werden, wobei algebraische Kopplungsbedingungen an Abzweigungen und Einmündungen angenommen werden müssen.

Das Steuerungsproblem

Ziel der Steuerung ist die Entlastung von Abwasser in die Umwelt zur verhindern bzw. so gering wie möglich zu halten. Dieses Ziel wird mittels eines Kostenfunktionals J formuliert. Die Steuerung der steuerbaren Elemente ist im Vektor U hinterlegt, im Vektor $V = (A, Q)$ sind die Wasserzustände gesammelt. Mit einer Funktion $G : (U, Y) \to \mathbb{R}$ wird die Entlastung des Abwassers in die Umwelt beschrieben. Typischerweise handelt es sich hierbei um die Flussrate durch einen Entlastungskanal. Das Kostenfunktional kann wiederum als Doppelintegral über Ort und Zeit verstanden werden. Das Optimierungsproblem lautet daher:

$$\min_{U(t)} J\left(U(t), Y(t,x)\right) = \min_{U(t)} \int_{t\in T}\int_{x\in K} G\left(U(t), Y(t,x)\right) \mathrm{d}x\mathrm{d}t \qquad (3)$$

Details zur numerischen Lösung dieses Optimierungsproblems unter Zuhilfenahme einer Finite-Volumen-Diskretisierung und eines Adjungiertenkalküls findet sich in [2].

Die Anlage

Die oben dargestellten Aspekte wollen wir mit einer Anlage erlebbar machen. Dazu haben wir uns für zwei Kanäle und den in Abbildung 2 dargestellten Aufbau entschieden. Die Anlage wurde als ein mobiles System konzipiert, so dass damit auch in Schulen Projekttage durchgeführt werden können.

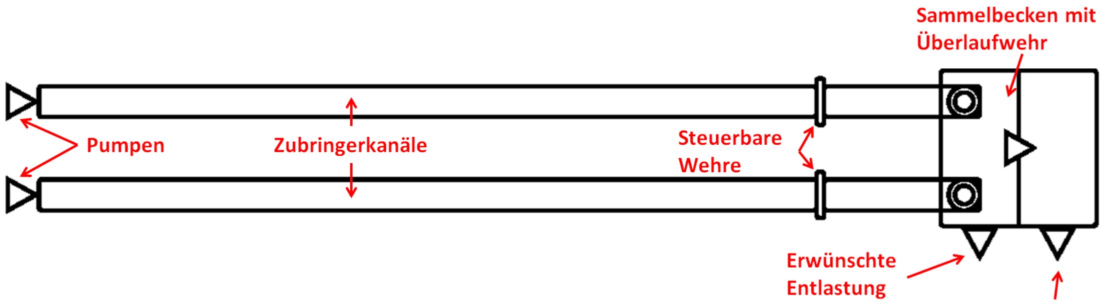


Abb. 2: Prinzipieller Aufbau

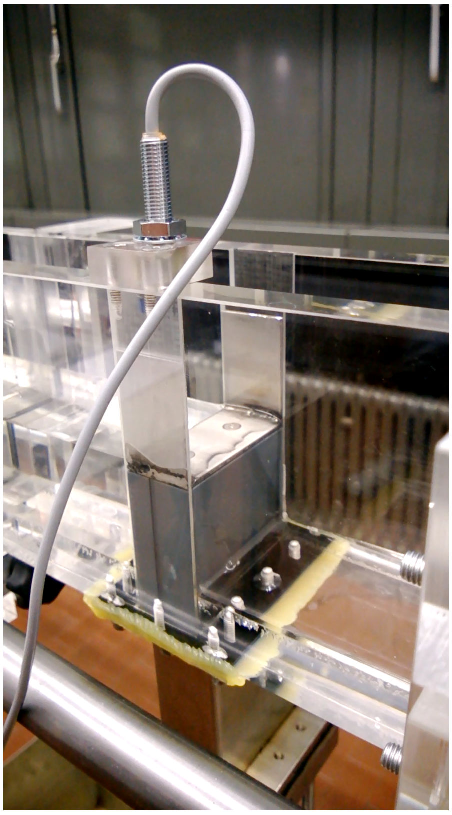


Abb. 3: Wehr der Anlage

Die Kanäle wurden in Acrylglas ausgeführt, damit der Wasserfluss gut zu beobachten ist. Wichtig war auch, dass die Wehre von unten nach oben gefahren werden können, so dass das Wasser über die voll angehobenen Wehr abfließen kann. Der Ausfluss der Kanäle führt in eine Kammer (Kammer 1) des geteiltes Sammelbecken. Jede Kammer hat am Boden ein Loch, so dass ein kontinuierlicher Abfluss gewährleistet ist. Das Ziel der Steuerung ist ein Überlaufen der Kammer 1 in Kammer 2 zu verhindern.
Der erste Entwurf dieser Anlage wurde von Studierenden des Chemie- und Bioingenieurwesens im Rahmen eines Praktikums geplant.

Die Entwicklung und die Realisierung der Anlage wurde federführend durch den Lehrstuhl für Prozessmaschinen und Anlagentechnik (iPAT) durchgeführt. Diese Zusammenarbeit zwischen den Ingenieuren und uns Mathematikern wurde dabei als sehr bereichernd empfunden.

Der Zufluss in die Kanäle wird über Pumpen geregelt, deren Drehzahl sich einstellen lässt. Diese Pumpen sind nicht mit den Pumpen in realen Systemen zum Umpumpen von Wasser zu verwechseln. Als Steuerungen verwenden wir die zwei Wehre, die in den Kanälen eingebaut sind. Die Bedienung der Anlage erfolgt über eine graphische Oberfläche, die auf einem Hutschienen-PC unter Windows 7 Embedded läuft. Parallel dazu kann die Anlage über eine analoge Schnittstelle von außen angesprochen werden. So lässt sich eine Fernbedienung mit Hilfe eines Mikrocontrolers (z.B. eines Arduinos) oder PCs implementieren. Gegenwärtig ist die Anlage technisch fertig. Der nächste Schritt ist nun, die beschriebenen open-loop-Steuerungskonzepte direkt auf die Wehre anzuwenden.

Studentische Arbeiten

Die erste Konzeption für die Realisierung der Anlage wurde von Studierenden erstellt. Dabei waren zu beachten, dass die Anlage modular aufgebaut werden soll und dass die Wehre im Boden verschwinden müssen. Die Umsetzung und den Aufbau der Anlage wurde von einer studentischen Hilfskraft durchgeführt. Als Ansprechpartner stand ein Mitarbeiter am iPAT zur Verfügung.

Eine Anbindung der analogen Schnittstelle an eine graphische Oberfläche auf einem PC durch studentische Hilfskräfte mit Fachkenntnissen in Informatik und Elektrotechnik soll im nächsten Schritt realisiert werden, um modellbasierte Steuerungen auf die Anlage aufzuspielen. Da sich die Flachwassergleichungen als schwer zu lösendes System herausgestellt haben, soll im Rahmen einer Arbeit von Studierenden der Mathematik ein vereinfachtes mathematisches Modell für diese Anlage entwickelt werden. Das Modell soll auf Basis von gewöhnlichen Differentialgleichungen mit Totzeiten entstehen und dann die Grundlage für die Berechnung einer optimalen Steuerung bilden.

Danksagung

Bedanken dürfen wir uns für die Unterstützung bei der Finanzierung und beim Aufbau der Anlage bei: Prof. Dr. Günter Leugering, Prof. Dr. Eberhard Schlücker, Dr. Benjamin Pohrer, Markus Bosser, Werner Sippl und Sebastian Zametzer.

Literaturverzeichnis

[1] **Björn Geißler**: *Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization.* PhD thesis, Dissertation, Department Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2011.

[2] **Johannes Hild**: *Real-Time Control of Hydrodynamic Process Models on Finite Volume Net- works* PhD thesis, Dissertation, Department Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2011.

[3] **Alexander Martin and Günter Leugering** (eds.): *Mathematical Optimization of Water Networks.* Birkhäuser Basel, 2011.

[4] **Andrea Zelmer**: *Designing Coupled Energy Carrier Networks by Mixed-Integer Programming Methods.* PhD thesis, Dissertation, Fachbereich Mathematik, Technische Universität Darmstadt, 2010.

[5] **European Parliament and Council**: *Directive 2000/60/ec of the European parliament and of the council of 23 October 2000 establishing a framework for community action in the field of water policy.* Official Journal L 327, 22/12/2000, 2000.

Autoren

Dr. rer. nat. Johannes Hild
Dr. rer. nat. Wigand Rathmann
Friederich-Alexander-Universität Erlangen Nürnberg
Department Mathematik
Cauerstr. 11
91058 Erlangen
E-Mail: johannes.hild@fau.de
wigand.rathmann@fau.de

Teilnehmerliste

Name	Organisation	E-Mail
Dr. Susanne Bellmer	Ostfalia Hochschule für angewandte Wissenschaften, Wolfenbüttel	s.bellmer@ostfalia.de
Dipl.-Stat. Gabriela Bender	Ostfalia Hochschule für angewandte Wissenschaften, Wolfenbüttel	g.bender@ostfalia.de
Prof. Dr. Katharina Best	Hochschule Hamm-Lippstadt	katharina.best@hshl.de
Dipl.-Math. Anja Bird	FAU Erlangen-Nürnberg	anja.bird@fau.de
Prof. Dr Manuela Boin	Hochschule Ulm	boin@hs-ulm.de
Prof. Dr. Claudia Frohn-Schauf	Hochschule Bochum	claudia.frohn-schauf @hs-bochum.de
Prof. Dr. Joachim Fulst	Hochschule Bochum	joachim.fulst@hs-bochum.de
Prof. Dr. Klaus Giebermann	Hochschule Ruhr West	klaus.giebermann @hs-ruhrwest.de
Prof. Dr. Laurenz Göllmann	Fachhochschule Münster	goellmann@fh-muenster.de
Prof. Dr. Christian Henig	Hochschule Osnabrück	c.henig@hs-osnabrueck.de
Katharina Görtler	FAU Erlangen-Nürnberg	katharina.goertler@gmail.com
Prof. Dr. Tim Kröger	TH Nürnberg	tim.kroeger@th-nuernberg.de
Prof. Dr. Petra Leitert	Hochschule Wismar, Gottlob-Frege-Zentrum	petra.leitert@hs-wismar.de
Prof. Dr. Anne Leucht	TU Braunschweig	a.leucht@tu-bs.de
Dr. Wigand Rathmann	FAU Erlangen-Nürnberg	wigand.rathmann@fau.de
Prof. Dr. Beate Rhein	TH Köln	beate.rhein@th-koeln.de
Prof. Dr. Thomas Risse	Hochschule Bremen, City University of Applied Sciences	risse@hs-bremen.de
Prof. Dr. Kathrin Thiele	Ostfalia Hochschule für angewandte Wissenschaften, Wolfenbüttel	k.thiele@ostfalia.de
Dipl. Math. Nimet Sarikaya	Fachhochschule Dortmund	nimet@mailbox.org
Prof. Dr. Petra Scheffler	Hochschule Stralsund	petra.scheffler @hochschule-stralsund.de

Prof. Dr.-Ing. Thomas Schiemann	Hochschule für angewandte Wissenschaften Hamburg	thomas.schiemann@haw-hamburg.de
Prof. Dr. Markus Schmidt-Gröttrup	Hochschule Osnabrück	M.Schmidt-Groettrup@hs-osnabrueck.de
Prof. Dr. Angela Schmitz	TH Köln	angela.schmitz@th-koeln.de
Dip.-Math. Dipl.-Wi.-Math. Nicolai von Schroeders	FAU Erlangen-Nürnberg	nicolai.von.schroeders@fau.de
Prof. Dr. Dieter Schott	Hochschule Wismar, Gottlob-Frege-Zentrum	dieter.schott@hs-wismar.de
Mikko Vasko, M.A.	Hochschule Karlsruhe	mikko.vasko@hs-karlsruhe.de
Prof. Dr. Jürgen Vorloeper	Hochschule Ruhr West	juergen.vorloeper@hs-ruhrwest.de
Prof. Dr. Mario Walther	Ernst-Abbe-Hochschule Jena	mario.walther@eah-jena.de
Prof. Dr. Alfred Wassermann	Universität Bayreuth	alfred.wassermann@uni-bayreuth.de

WFR - Wismarer Frege-Reihe / Wismar Frege Series

Beiträge zur Mathematikausbildung von Ingenieuren

Heft 01/2005	Proceedings 4. Workshop Mathematik für Ingenieure, Bremen, Oktober 2005.
Heft 05/2006	Proceedings 5. Workshop Mathematik für Ingenieure, Wismar, Teile 1 – 3, September 2006.
Heft 01/2007	Proceedings Minisymposium Moderne Mathematikausbildung für Ingenieure, Humboldt-Universität Berlin, Teile 1 – 2, März 2007.
Heft 02/2007	Mathematik für Ingenieure – Thesen zum Jahr der Mathematik 2008, Dezember 2007. Mathematics for Engineers – Theses to the Year of Mathematics 2008, December 2007.
Heft 03/2008	Proceedings 6. Workshop Mathematik für Ingenieure, Soest, Teile 1 – 2, September 2008.
Heft 04/2008	Proceedings Minisymposium Moderne Mathematikausbildung für Ingenieure, Friedrich-Alexander-Universität Erlangen-Nürnberg, Teile 1 – 2, September 2008.
Heft 03/2009	Peter Junglas: Interaktive Simulationsprogramme zur Demonstration von klassischen und quantentheoretischen Wellenphänomenen, Juni 2009.
Heft 04/2009	Proceedings 7. Workshop Mathematik für Ingenieure, Wolfenbüttel, Juni 2009.
Heft 02/2010	Information – Programme and Abstracts, 15th SEFI MWG Seminar & 8th Workshop GFC, Wismar, June 2010.
Heft 03/2010	Proceedings 8. Workshop Mathematik für Ingenieure, Wismar, Juni 2010.
Heft 05/2010	Larissa Fradkin: Teaching Algebra and Calculus to Engineering Entrants, December 2010.
Heft 02/2011	Proceedings 9. Workshop Mathematik für ingenieurwissenschaftliche Studiengänge, Wilhelmshaven, September 2011.
Heft 03/2013	Proceedings 11. Workshop Mathematik für ingenieurwissenschaftliche Studiengänge, Teile 1 – 2, Bochum, September 2013.
Heft 02/2015	Proceedings 12. Workshop Mathematik für Ingenieure, Teile 1 – 2, Hamburg, Februar 2015.

Heft 04/2016	Proceedings 13. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen, Lingen, September 2016.
Heft 01/2017	Proceedings 14. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen, Erlangen, September 2017.

Hinweis:
Die Proceedings zur Workshopreihe beginnen erst mit dem 4. Workshop.
Die Proceedings zum 10. Workshop Mathematik erschienen in einem Extraband an der Hochschule Ruhr/West in Mülheim.

Herausgeber und Redakteur

Prof. Dr. rer. nat. habil. Dieter Schott
Gottlob-Frege-Zentrum
Fakultät für Ingenieurwissenschaften
Hochschule Wismar
Philipp-Müller-Str. 14
D - 23966 Wismar
Telefon: ++49 / (0)3841 / 753 7333
Fax: ++49 / (0)3841 / 753 7130
E-Mail: dieter.schott@hs-wismar.de

Vertrieb:
Direkt über den Herausgeber oder das Gottlob-Frege-Zentrum

ISSN 1862-1767
ISBN 978-3-942100-55-7