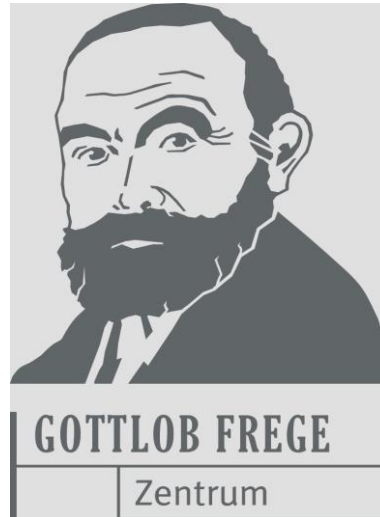


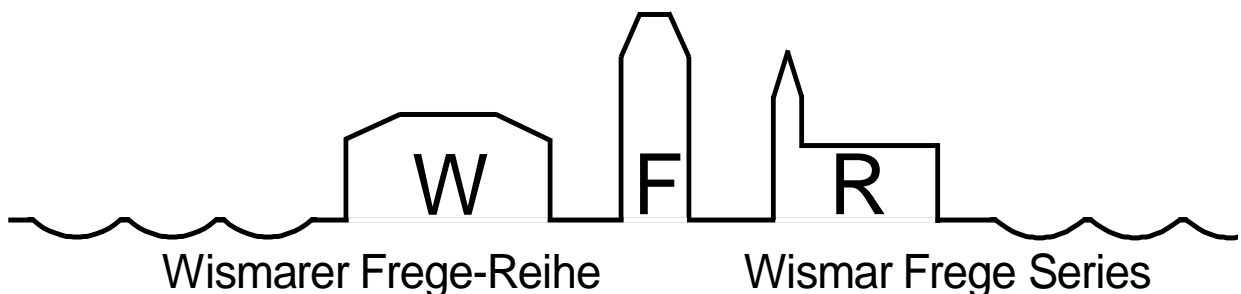
Hochschule Wismar Gottlob Frege Centre



Proceedings 1st Northern-Light Symposium on Engineering Education

Hamburg, April 2018

Heft 02 / 2018



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>

Die Wismarer Frege-Reihe ist urheberrechtlich geschützt. Eine Vervielfältigung ganz oder in Teilen, ihre Speicherung sowie jede Form der Weiterverbreitung bedürfen der vorherigen Genehmigung durch den Herausgeber.

ISSN 1862-1767

Alle Rechte vorbehalten.

© Hochschule Wismar 2018.
Printed in Germany

WFR Heft 02/2018**Inhalt / Contents**

Vorwort / Preface	2
Einladung / Invitation	4
Programm / Program	5
Abendveranstaltung / Evening Event	6

Artikel / Articles

Dieter Schott: <i>How to get in the Top Ten? An Analysis of the Bechhofer Selection Problem in Statistics.</i>	7
Thomas Risse: <i>Mathematics for Computer Science Students –beyond Linear Algebra and Calculus.</i>	23
Peter Junglas: <i>Simulationsprogramme für die Lehre – Was kommt nach Applets?</i>	32
Helena Barbas, Franz Konieczny, Alexander Lohse, Thomas Schramm: <i>oHMint. An Online Mathematics Course and Learning Platform for MINT Students.</i>	43

Anhang / Appendix

WFR - Übersicht	51
-----------------	----

Vorwort

Als geborener 49-er machte ich mir nach dem offiziellen Ausscheiden aus dem Berufsleben 2015 so meine Gedanken, welche fruchtbringenden Netzwerke auch in Zukunft noch funktionieren könnten. Als ich 2005 die IngMath-Workshops in Wismar ins Leben rief (zunächst „Mathematik für Ingenieure“, später „Mathematik in ingenieurwissenschaftlichen Studiengängen“ als Zugeständnis an die inzwischen übliche Geschlechterdifferenzierung bei Berufen), die einen Erfahrungsaustausch zur Mathematiklehre im Ingenieurstudium zum Ziel hatten, suchte ich gleichzeitig nach einem Medium, in dem die Inhalte und Ergebnisse der Workshops „aufgehoben“ (gespeichert) werden. So gibt es seit 2005 die Wismarer Frege-Reihe, in der zu Beginn die Proceedings der Workshops veröffentlicht wurden, später dann aber auch andere Beiträge aus dem Umfeld der Mathematiklehre. Die Vernetzung mit internationalen Institutionen (wie UICEE, WIETE, SEFI Mathematical Workgroup) führte neben deutschsprachigen auch zu englischsprachigen Artikeln.

Während die genannten Workshops zunächst an Orten im Norden stattfanden, später in die Mitte und in den Süden wanderten, wird es im Frühjahr 2019 eine Rückkehr an den Ursprung geben (Außenstelle Rostock-Warnemünde der Hochschule Wismar). Der Teilnehmerkreis wandelte sich mit den Veranstaltungsorten in gewisser Weise, hatte aber dabei einen festen und fixen Stamm. Mir kam nun die Idee, die alteingesessene Nordkonstante mit Peter Junglas (Diepholz), Thomas Risse (Bremen), Thomas Schramm (Hamburg) und mir (Wismar) zu einem zwanglosen Gedankenaustausch einzuladen. Diese Idee fiel bei den anderen auf fruchtbaren Boden. Zugleich erklärte sich Thomas Schramm bereit, seine Hamburger Dienststelle (HafenCity Universität) als für alle vier günstig gelegenen Tagungsort zur Verfügung zu stellen.

So erfolgte nach einer Terminabstimmung die Einladung zu einem Symposium (Tagung von Wissenschaftlern mit zwanglosen Vorträgen und Diskussionen) am 27.4. 2018, das Thomas Schramm treffend „1st Northern-Light Symposium“ taufte und damit die (zunächst) vier „Nordlichter“ zum Leben erweckte (siehe S. 4). Besonders reizte Thomas Schramm, dass das ursprünglich altgriechische Wort „Symposion“ ein Gastmahl mit Trinkgelage und Unterhaltung war. Wir fanden das auch reizend.

Peter Junglas machte uns dann darauf aufmerksam, dass es das zu unserem Symposium passende Lied „Northern Lights“ (Pulchra est amica mea) für vier

Stimmen von Ola Gjeilo (*1978) gibt, das er selbst schon im Kammerchor gesungen hat und das man im Internet auch akustisch abhören kann. Ich erinnerte mich, dass ich dieses wunderschöne Lied am 12.7. 2017 abends in der Marienkirche von Ribnitz-Damgarten gehört hatte. Es wurde damals von den Kurpfälzer Madrigalisten gesungen.

Wir trafen uns also zu viert kurz vor 16.00 Uhr im Dienstzimmer von Thomas Schramm, begannen mit Kaffee und Kuchen und hielten und diskutierten danach unsere Vorträge (siehe Programm S. 5). Sie wurden zwar in Deutsch gehalten, benutzten aber teilweise auch in Englisch formuliertes Bildmaterial.

Zum Abend wanderten wir in die urige Gaststätte „Laufauf“ in der Nähe der Universität zu einem gemütlichen Abendessen (siehe S. 6). Einem Wortspiel folgend gab es dort ein reichhaltiges Angebot an Aufläufen. Davon und vom Durstlöcher Bier machten wir mit Freuden und natürlich in Maßen Gebrauch. Während des Abends gab es anregende Diskussionen zu aktuellen Problemen der Mathematikausbildung im Ingenieurstudium an den Hochschulen und zur weiteren Perspektive des Symposiums. Wir waren uns einig, dass die Nordlicht-Premiere eine Fortsetzung verdient. Das „2nd Northern-Light Symposium“ könnte im Herbst 2019 wieder in Hamburg stattfinden. Diskutiert wurde auch eine Erweiterung des Teilnehmerkreises um einige Personen, ohne schon eine endgültige Entscheidung zu treffen. Mein Angebot, die Vorträge des Symposiums in der Wismarer Frege-Reihe zu veröffentlichen, wurde gern angenommen. Die Umsetzung liegt mit diesem Heft vor. Vorerst keine Lösung fand das Problem, wie die Wismarer Frege-Reihe nach meinem Ausscheiden als Herausgeber und Redakteur fortgeführt werden könnte.

Es ist nicht vermessen, dem „Vermesser“ und Organisator Thomas Schramm im Namen aller übrigen Teilnehmer für ein gut organisiertes und attraktives Symposium in der Metropole Hamburg zu danken.

Bis zum nächsten Northern-Light-Symposium!

Ihr Dieter Schott



Geodäsie & Geoinformatik

Prof. Dr. Thomas Schramm
Überseealle 16
20457 Hamburg

Tel.: 040 42827 5383

EINLADUNG

1st Northern-Light Symposium on Engineering Education

Liebe Kolleginnen und Kollegen,

Ich möchte Sie herzlich zu unserem ersten Northern-Light Symposium über Ausbildung im Ingenieurwesen an Hochschulen am 27. April 2018 einladen.

Beginn: 16:00 Raum 3.031 HafenCity Universität Hamburg.

Programm:

- 16:00-16:30 Prof. Dr. Dieter Schott, Hochschule Wismar: Wie kommt man unter die Top Ten? Eine statistische Analyse mit Analysis im Hauptgang und einer Prise MATLAB.
- 16:30-17:00 Prof. Dr. Thomas Risse, Hochschule Bremen: Welche Mathematik für Technische Informatiker im dritten Semester?
- 17:00-17:30 Prof. Dr. Peter Junglas, Private Hochschule für Wirtschaft und Technik, Diepholz: Simulationsprogramme für die Lehre - Was kommt nach Applets?
- 17:30-18:00 Prof. Dr. Thomas Schramm, HafenCity Universität: Math4Ing. Ein Projekt der Hamburg Open Online University (HOOU) und des OMB+-Konsortiums für einen First-Year-Online-Kurs in Ingenieurmathematik für alle Hochschulen.

Auf Wunsch kann die HCU, speziell das Geo-Lab, VR-Lab oder 3D-Lab besichtigt werden.

Mit freundlichen Grüßen
Thomas Schramm

Programm

Gegenüber der Planung (S.4) ergaben sich einige kleine Änderungen.

Uhrzeit	Programmpunkt	Personen
16:00	Begrüßung durch den Organisator	Prof. Dr. Thonas Schramm, HafenCity University Hamburg
	Kaffeepause	
	Vorträge	
16:30	Welche Mathematik für Technische Informatiker im dritten Semester?	Prof. Dr. Thomas Risse, Hochschule Bremen
17:00	Simulationsprogramme für die Lehre – Was kommt nach Applets?	Prof. Dr. Peter Junglas, Private Hochschule für Wirtschaft und Technik Diepholz
17:30	Wie kommt man unter die Top Ten? Eine statistische Analyse mit Analysis im Hauptgang und einer Prise MATLAB	Prof. Dr. Dieter Schott, Hochschule Wismar
18:00	Math4Ing. Ein Projekt der Hamburg Open Online University (HOOU) und des OMB+-Konsortiums für einen First-Year-Online-Kurs in Ingenieurmathematik für alle Hochschulen.	Prof. Dr. Thomas Schramm, HafenCity University Hamburg
18:30	Ende	
19:00	Restaurant „LAUFAUF“	
	Diskussionen	Alle vier

Hinweis:

Die Beiträge in den Proceedings sind Vortragsausarbeitungen oder Weiterentwicklungen und haben zum Teil einen anderen Titel oder weitere Autoren.

Abendveranstaltung

Ein Rundblick in das Restaurant „Laufauf“



Quelle: Internet

Kommentar: Noch sieht im Zusammenschnitt der Laufauf-Örtlichkeiten alles schön und leer aus. Dann kamen die vier Nordlichter und sorgten für Partystimmung bis das Restaurant schön voll war und die Gäste ausgelassen feierten, obwohl sie unsere Vorträge verpasst hatten.

Dieter Schott

How to get in the Top Ten? An Analysis of the Bechhofer Selection Problem in Statistics

Abstract: Following a Bechhofer selection procedure we show that the function

$$f(n) = \int_{-\infty}^{+\infty} [\Phi(z + r\sqrt{n})]^{a-t} [1 - \Phi(z)]^{t-1} \varphi(z) dz$$

is strictly monotone increasing and bounded by some constant C . The minimal n satisfying the inequality $f(n) \geq K$ where $0 < K < C$ is determined by numerical methods. Some numerical simulation results are given and discussed for $t > 1$. MATLAB files are presented for readers to carry out own experiments and to apply the results for certain selection procedures of Bechhofer type.

1. The Bechhofer Selection Problem

In the books [2, p. 489f.] and [3, p. 513f.] statistical selection procedures are described to separate the t in some sense best populations from a collection of a normally distributed populations for a given risk $\beta \in]0,1[$ of wrong decision. More precisely, the populations G_i in

$$M = \{G_1, G_2, \dots, G_a\}$$

with characteristics \mathbf{x}_i distributed as $N(\mu_i, \sigma^2)$ are assumed to be ranked according to increasing expectation μ as follows:

$$(G_{(1)}, \mu_{(1)}), (G_{(2)}, \mu_{(2)}), \dots, (G_{(a)}, \mu_{(a)}).$$

Then we can create two sets

$$M_1 = \{G_{(a)}, \dots, G_{(a-t+1)}\}, M_2 = \{G_{(a-t)}, \dots, G_{(1)}\}, \quad t < a$$

with the top set M_1 of the t best populations. If $t = 10$ is chosen, then M_1 contains the top ten. The first part of the title sounds a little bit sensational but should create attention. We assume that the gap

$$d(M_1, M_2) = d(\mu_{(a-t+1)}, \mu_{(a-t)})$$

between the top populations and the remaining ones should be at least $\delta > 0$. Practically the populations are selected taking stochastically independent samples $(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{in})$ of G_i ($i = 1, \dots, a$) with constant size n and ranking them after sampling according to their mean values $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_a$. The number n has to be great enough to solve the mentioned problem, but not too great to reduce the effort of ranking. The optimal n would be the minimal one.

Problem: For given real $\sigma^2 > 0$, $\delta > 0$, $\beta \in]0, 1[$ and natural a we look for a statistics \mathbf{M}_A realizing subsets M_A of M with $t < a$ elements which satisfies

$$P(\mathbf{M}_A = M_1 \mid d(M_1, M_2) \geq \delta) \geq 1 - \beta$$

for a minimal sample size $n = n_B = n_B(a, t, \sigma, \delta, \beta)$.

Remark: The case $n = n_B = 0$ is to mean that no sampling takes place (arbitrary selection of M_1).

2. The Bechhofer Solution Conditions

Following [1] the Bechhofer problem is solved if n fulfils the inequality

$$t \int_{-\infty}^{+\infty} \left[\Phi \left(z + \frac{\delta}{\sigma} \cdot \sqrt{n} \right) \right]^{a-t} \cdot [1 - \Phi(z)]^{t-1} \cdot \varphi(z) dz \geq 1 - \beta \quad (1)$$

under the additional condition

$$\frac{1}{\binom{a}{t}} < 1 - \beta \quad (2)$$

where $\varphi(z)$ is the density function and $\Phi(z)$ is the distribution function of the standard normal distribution $N(0, 1)$.

Remark. If condition (2) is violated no real statistical experiment exists (see e.g. [3, p. 515]).

In [1], [2] and [3] tables for $\frac{\delta}{\sigma} \sqrt{n}$ can be found for some special parameters which allow determining n . E.g. we find

$$a = 10; \quad t = 1; \quad r = 1; \quad \beta = 0.05; \quad n = n_B = 12.$$

Here we look for a general mathematical solution of the problem to support an extensive simulation study [4].

We introduce the abbreviations

$$r := \frac{\delta}{\sigma} > 0, \quad 0 < K_B := \frac{1 - \beta}{t} < 1$$

and concentrate first on the Bechhofer condition (1) ignoring condition (2).

Problem A. For fixed real numbers $r > 0$, $K_B \in]0,1[$ and fixed natural numbers a, t ($a > t$) we look for minimal unknown natural numbers n fulfilling (1) where $n = n_1 = n_1(a, t, r, \beta)$.

Remark. Problem A suggests, that (1) is satisfied also for all $n \geq n_1$ and that $n_B = n_1$ holds if (2) is fulfilled.

3. Problem Analysis

The Bechhofer problem simplifies for $t = 1$. Then an effective formula for n_B exists using the β -quantiles of an $(a - 1)$ -dimensional normal distribution (see [2], [3]). Hence, we treat especially the cases $t > 1$. We use a real extension of the problem replacing the natural n by a nonnegative x . The integrand in (1) is denoted by

$$I(z, x) = [\Phi(z + r\sqrt{x})]^{a-t} [1 - \Phi(z)]^{t-1} \varphi(z), \quad z \in R, x \in R_+. \quad (3)$$

We introduce the two functions

$$f(x) = f(x; a, t, r) = \int_{-\infty}^{\infty} I(z, x) dz, \quad (4)$$

$$f_N(x) = f_N(x; a, t, r) = \int_{-N}^N I(z, x) dz \quad (5)$$

where N is an appropriate positive number. The cut function f_N comes into play if numerical integration is used. We start with some properties of these functions.

Proposition 1. The functions $f(x)$ and $f_N(x)$ are continuous, strictly monotone increasing and bounded. The difference of both functions can be made (uniformly in x) arbitrarily small for sufficiently large N .

Proof. The continuity of the two functions is obvious. The integrand $I(z, x)$ is strictly monotone increasing with respect to x (for fixed z) taking the strict monotony of \sqrt{x} and $\Phi(z)$ into account. Consequently, both $f(x)$ and its approximation $f_N(x)$ are strictly monotone increasing. Further, the integrand satisfies the estimations

$$0 < I(z, x) < [1 - \Phi(z)]^{t-1} \varphi(z) < \varphi(z)$$

since the values of Φ are contained in the interval $]0,1[$. Thus we have

$$0 < f_N(x) < f(x) < \int_{-\infty}^{\infty} \varphi(z) dz = 1.$$

Hence f and f_N are bounded (by 1). Finally we get

$$f(x) = f_N(x) + R_N(x),$$

$$\begin{aligned} 0 < R_N(x) &= \int_{-\infty}^{-N} I(z, x) dz + \int_N^{\infty} I(z, x) dz \\ &< 2 \int_N^{\infty} \varphi(z) dz = 2(1 - \Phi(N)). \end{aligned}$$

Consequently, the difference $R_N(x)$ can be made smaller than $\varepsilon > 0$ for $N > \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$. ■

Now we give a reformulation of Problem A.

Problem B. Considering (1), (4) and the strict monotony of f we have to determine a natural $n = n_1$ as small as possible such that

$$f(n) \geq K_B, \quad K_B = \frac{1 - \beta}{t} \tag{6}$$

for all $n \geq n_1$. ■

Classification of cases. Principally there are three possibilities:

- a) All $f(n)$ are smaller than K_B ,
- b) All $f(n)$ are greater than or equal to K_B ,
- c) There is a natural number $n = n^* > 0$ such that $f(n) \geq K_B$ for all $n \geq n^*$ and $f(n) < K_B$ for all $n < n^*$.

The first case a) is senseless in our application. In the second case b) we can choose $n_1 = 0$ which is also practically unrealistic. The third case c) is the important one. Here we put $n_1 = n^*$. The next section prepares the investigation of these cases.

4. Results

First we present an interesting statement for improper integration with respect to functions of the normal distribution.

Lemma 1. For nonnegative integers l and m it holds

$$\begin{aligned} \int_{-\infty}^{\infty} [\Phi(z)]^l [1 - \Phi(z)]^m \varphi(z) dz &= \sum_{i=0}^m \binom{m}{i} \cdot \frac{(-1)^i}{l + i + 1} \\ &= \frac{m!}{(l + 1) \cdot (l + 2) \cdot \dots \cdot (l + m + 1)} = \frac{l! \cdot m!}{(l + m + 1)!} \\ &= \frac{1}{(m + 1) \cdot \binom{l + m + 1}{m + 1}}. \end{aligned}$$

Proof. The first assertion can be shown by partial integration and the second one by mathematical induction. The remaining ones are simple transformations. In [5] a corresponding result is proved in detail with a more general integrand. ■

Corollary 1. For nonnegative integers m it holds

$$\int_{-\infty}^{\infty} [1 - \Phi(z)]^m \varphi(z) dz = \frac{1}{m + 1}.$$

Proof. The assertion arises from Lemma 1 by choosing there $l = 0$. ■

Now we can characterise the range of $f(x)$.

Proposition 2. The function $f(x)$ starts with $f(0) = \frac{1}{t \binom{a}{t}}$ and tends from below

to $f(\infty) = \lim_{x \rightarrow +\infty} f(x) = \frac{1}{t}$. The range of $f(x)$ is $R(f) = \left[\frac{1}{t \binom{a}{t}}, \frac{1}{t} \right]$.

Proof. We get with $l = a - t$ and $m = t - 1$ in Lemma 1

$$f(0) = \int_{-\infty}^{\infty} [\Phi(z)]^{a-t} [1 - \Phi(z)]^{t-1} \varphi(z) dz = \frac{1}{t \binom{a}{t}}$$

and with $l = 0$ and $m = t - 1$ (Corollary 1)

$$f(\infty) = \lim_{x \rightarrow +\infty} f(x) = \int_{-\infty}^{\infty} [1 - \Phi(z)]^{t-1} \varphi(z) dz = \frac{1}{t}.$$

As f is continuous and strictly monotone increasing the limit is reached from below and the range $R(f)$ is as asserted. ■

Theorem 1. Problem B has a unique solution n_1 .

Proof. Proposition 1 and Proposition 2 show that the monotone increasing $f(n)$ succeeds K_B for some natural n . Namely, we have

$$K_B = \frac{1 - \beta}{t} < \frac{1}{t} = f(\infty)$$

and therefore $f(n) \geq K_B$ for a unique minimal $n = n_1$ and all $n > n_1$. Hence, Problem B is solved uniquely. ■

Remark: Problem A has as a reformulation of Problem B also the same unique solution.

It depends on the choice of β and t whether $f(0) < K_B$ or not. By Theorem 1 case a) of the classification cannot occur while case b) can occur for β nearby 1, that is, if the selection is wrong with great probability. But in practice, a small β has to be chosen. On the other hand, for small β the limit $f(\infty)$ is nearby K_B . An analysis of f shows that $f(n)$ changes noticeable for small n and is nearly constant for large n . Observe that $f(0)$ only depends on a and t while $f(\infty)$ only depends on t (see Proposition 2). Both do not depend on the ratio r of σ and δ . If t is not small, the values of $f(0)$, K_B and $f(\infty)$ become very small.

Theorem 2. The Bechhofer conditions (1), (2) are fulfilled for a minimal $n = n_B$. All natural $n > n_B$ fulfil these conditions too.

Proof. By Proposition 2 case c) of the classification arises iff $f(0) < K_B$, that is condition (2) with $\frac{1}{\binom{a}{t}} < 1 - \beta$. Consequently we have under (1) and (2)

$$f(0) < K_B < f(\infty).$$

Hence, there is a minimal n with $f(n) \geq K_B$. This is also the minimal $n = n_B$ fulfilling the Bechhofer conditions. ■

Remark: For $f(0) \geq K_B$ we have $n_1 = 0$ while n_B does not exist. Otherwise it is $n_B = n_1$.

5. Solution Methods

Let us now turn to practical solution methods of our problem. One possibility is to apply methods with natural arguments n .

Using mathematical software as MATLAB

- the density φ and the distribution function Φ of the standardized normal distribution are pre-defined or can be easily defined.
- On this base the integrand $I(z, n)$ is declared.
- Then the cut number N is fixed
- A numerical procedure for the definite integrals $f_N(n)$ is used (trapezoidal rule, Simpson's rule or some more sophisticated method) such that certain accuracy is realised for integration.

It is not necessary to use numerical top methods since f_N has a very simple behaviour.

- If $f_N(0) \geq K_B$, then stop because condition (2) is violated.
- Starting with $f_N(0) < K_B$ the number n is increased as long until $f_N(n)$ jumps over K_B (e.g. by using a while-loop). We know that $f(n)$ is than over K_B , too.
- The program should print this last n as the one we looked for.

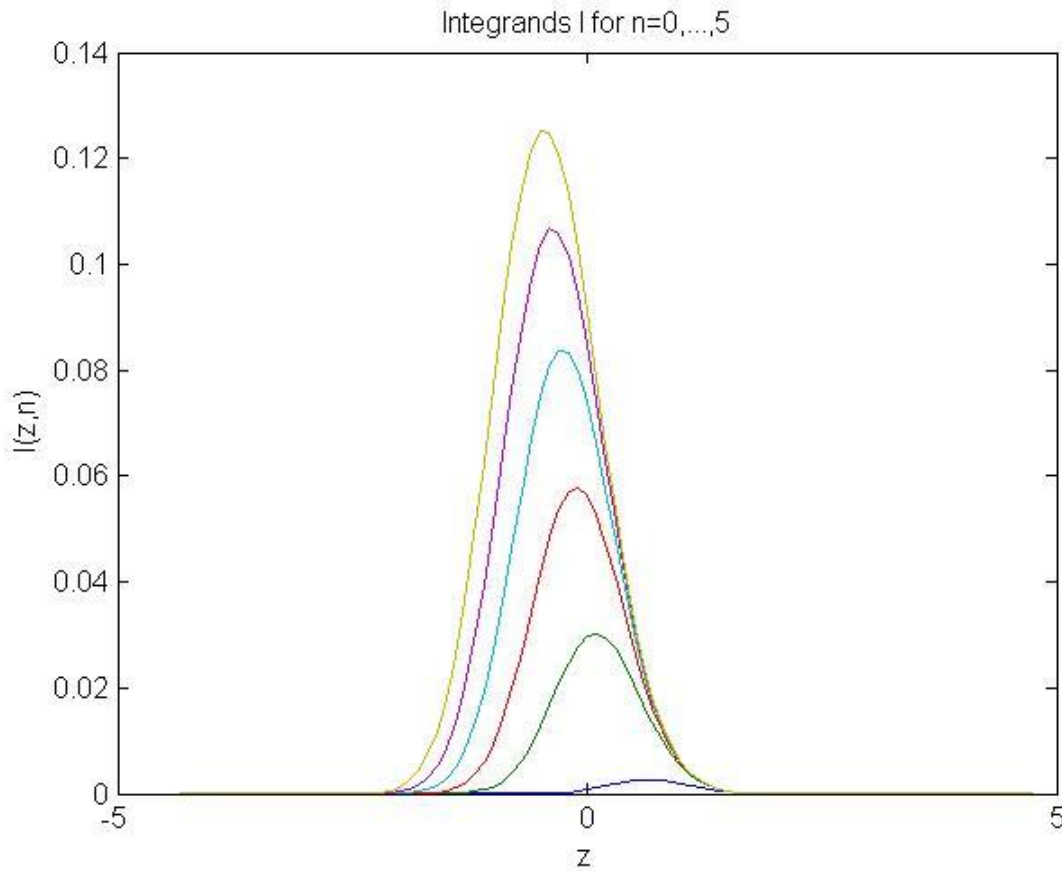
Considering the numerical errors it can happen if we are not carefully enough that n_B is chosen one unit too large. But this is meaningless for practical applications.

Another possibility is to replace n by real $x \geq 0$. This approach is used in the simulation study [4] on the base of the statistical software R. Then $f_N(n)$ becomes a continuous, strongly monotone function $f_N(x)$ which takes the value K_B exactly once, say at $x = x^*$. Then the procedure is as follows.

- Check the condition $f_N(0) < K_B$. Stop if it does not hold..
- Calculate the zero x^* of $f_N(x) - K_B = 0$ by a numerical method (as bisection, Newton's method or other ones).
- Determine $n_B = \lceil x_* \rceil$, where $\lceil \cdot \rceil$ rounds off to the next integer.

6. Simulation

If we want to look behind the solved problem it is interesting to study the shape of the integrands (3) and the functions (4). The figures in this section are produced by MATLAB. Figure 1 shows an example for (3).

Figure 1. $I(z, x)$ for $a = 10$, $t = 3$, $r = 1$ and $x = n = 0, 1, \dots, 5$ 

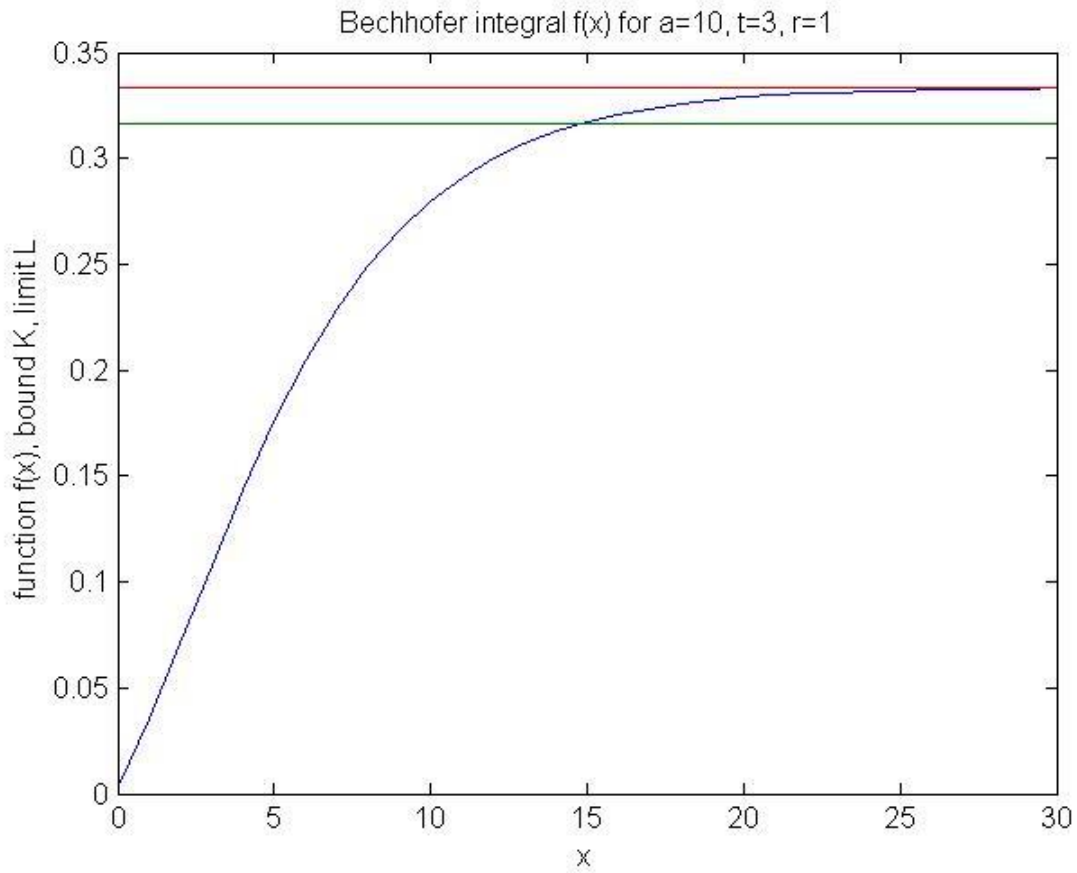
These $I(z, n)$ are unimodal for fixed n . The global maxima $I_{max} = I(z_{max}, n)$ increase and their positions z_{max} walk to the left (see Table 1).

Table 1. Global maxima of the integrands in Figure 1

n	0	1	2	3	4	5
z_{max}	0.6	0.1	-0.1	-0.3	-0.4	-0.5
I_{max}	0.0027	0.0303	0.0578	0.0837	0.1066	0.1253

Figure 2 shows a special function $f(x)$ with the Bechhofer constant K_B and the asymptote of $f(x)$.

Figure 2. Function $f(x)$ with parameters $a = 10, t = 3, r = 1$.



We specify the parameters for simulation. We consider

$$a = 30, 50, 100, 200;$$

$$t = 2, 3;$$

$$\beta = 0.05; r = \frac{1}{2}, 1.$$

By Theorem 2 we get for certain $a > 2$ and $t = 2$

$$f(0) = \frac{1}{a \cdot (a - 1)}, \quad K_B = \frac{19}{40}, \quad f(\infty) = \frac{1}{2} = \frac{20}{40}$$

and for certain $a > 3$ and $t = 3$

$$f(0) = \frac{2}{a \cdot (a - 1) \cdot (a - 2)}, \quad K_B = \frac{19}{60}, \quad f(\infty) = \frac{1}{3} = \frac{20}{60}.$$

The initial values are for $t = 2$

a	30	50	100	200
$f(0)$	0.001	0.004	0.0001	0.000025

and for $t = 3$

a	30	50	100	200
$f(0)$	$8.21 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$	$2.1 \cdot 10^{-6}$	$3 \cdot 10^{-7}$

Hence, for our data $f(0) < K_B$, that is (2), holds. Table 2 and Table 3 show our simulation results for the given data produced by the MATLAB files given in the supplement.

Table 2. Values of n_B in the simulation for $t = 2$.

r	a			
	30	50	100	200
$r = \frac{1}{2}$	71	78	87	95
$r = 1$	18	20	22	24

Table 3. Values of n_B in the simulation for $t = 3$.

r	a			
	30	50	100	200
$r = \frac{1}{2}$	77	84	93	102
$r = 1$	20	21	24	26

The simulation shows that n_B increases for decreasing ratio r , increasing t and increasing $a - t$ assuming the other parameters are fixed. This can be found also by theoretical investigations. Considering the expression $\frac{\delta}{\sigma}\sqrt{x}$ in (1) and the relation

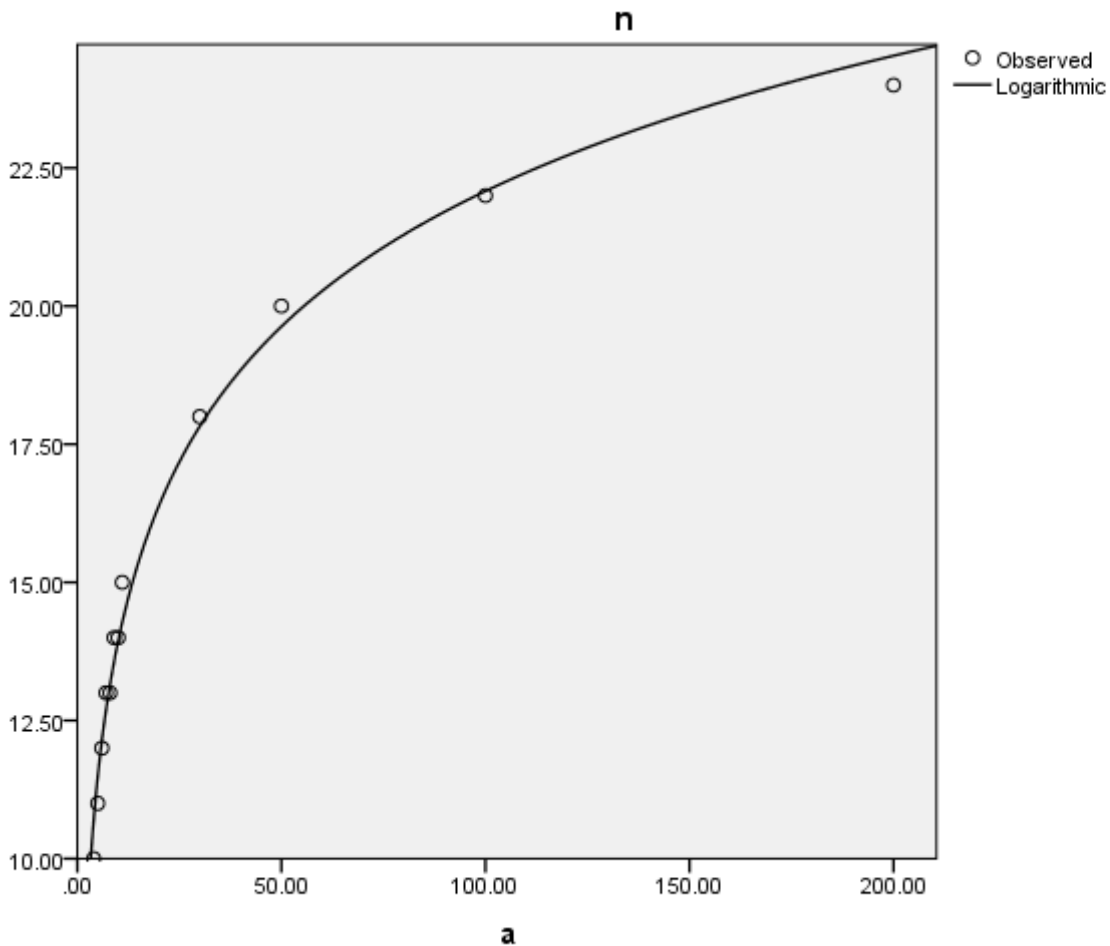
$$\frac{r}{c}\sqrt{c^2n} = r\sqrt{n}, \quad r = \frac{\delta}{\sigma} > 0, \quad c > 0$$

for $x = n$ it becomes clear that dividing r by c means increasing n_B by c^2 . For $c = 2$ the number n_B increases by the factor 4. This result is confirmed by our simulation in Table 1 and Table 2 if the rounding, especially the rounding off to the next integer, is taken into account. If the zeros x^* are calculated instead of n_B (see section 5), this rounding off effect is eliminated.

Investigating the relation between a and n the integral expression in (1) suggests to look for a logarithmic function since a is in the exponent and n in the base.

Figure 3 shows indeed that for $r = 1$ and $t = 2$ the calculated values (see Table 1) can be approximated very well by a logarithmic term. Additionally, the values for n up to $a = 12$ are taken from the book [2] and book [3], respectively, to get more data for fitting. This logarithmic approximation can also be confirmed by asymptotic considerations for other parameter values.

Figure 3. Relation between a and n for $r = 1$ and $t = 2$



7. Supplement: MATLAB FILES

Now a MATLAB file is given for users which want to reproduce our simulation results or which want to make own simulations with different parameter values. The code does not use special toolboxes, is elementary and commented. Hence, it can simply be transformed into other programming languages.

In the following MATLAB main file the condition $f = f(0) < K = K_B$ is not checked at the beginning. The number n starts with $n = 0$. If the condition is violated, the loop is left for $n = 1$ instead of $n = 0$ which is also a statistically not relevant case. Considering simple programming and statistical relevance this minor handicap is accepted.

Main file

```
% Bechhofer_n.m
% Bechhofer approach
% Determination of sample number n
clc; clear all; disp(' ');

% global variables used also in the function file
global n r a t

% inputs
a =input('number of populations : a = ');
t =input('number of preferred populations : t = ');
r =input('ratio factor : r = ');
pc=input('probability of correct selection: pc = ');
N =input('cut number of integration: N = ');
K = pc/t; % bound
```

```

% initialisation
n=0;
f=0;

% while loop
while f < K
n = n+1;
% integral f(n) calling function
f = quad(@bechhofer_I,-N,N);
end % loop

%outputs
disp(' '); % empty line
disp(['number of samples n= ',num2str(n)]);
disp(['integral f(',num2str(n),') = ',num2str(f)]);
disp(['lower bound K = ',num2str(K)]);

```

Function file (called in the main file)

```

% Bechhofer integrand
% function file bechhofer_I.m
function I = bechhofer_I(z);
global n r a t % global variables
% normal density function
phi = 1/sqrt(2*pi)* exp(-z.^2/2);
% normal distribution function (erf: error function)
Phi = 0.5 * (1 + erf(z./sqrt(2)));
v = z+ r*sqrt(n); % shifted argument
% distribution with shifted argument
VPhi = 0.5 * (1 + erf(v./sqrt(2)));
% Bechhofer integrand
I = VPhi.^(a-t).*(1-Phi).^(t-1).*phi;

```

References

- [1]. R.E. Bechhofer, A Single Sample Multiple Decision Procedure for Ranking Means of Normal Populations with Known Variances. *Ann. Math. Statist.* 25, 16 – 39 (1954).
- [2]. D. Rasch und D. Schott: Mathematische Statistik für Mathematiker, Natur- und Ingenieurwissenschaftler, Wiley-VCH 2016.
- [3]. D. Rasch and D. Schott: Mathematical Statistics , Wiley 2018.
- [4]. D. Rasch, Y. Takuya, D. Schott and J. Pilz: Statistical Selection Procedures for Expectations – a Review and Recent Simulation Results. Paper for the 10th *International Workshop on Simulation and Statistics*, Salzburg 2019.
- [5]. D. Schott: Solution of Inequalities for Monotone Functions Generated by Improper Integrals. To be submitted.

Author

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

Mathematics for Computer Science Students – beyond Linear Algebra and Calculus?

Abstract. Which mathematical topics do need students who study computer science at an university of applied sciences? Which topics might motivate them? From which topics will they benefit most? We try and suggest specific answers.

Introduction

Most students attitude [17] towards mathematics can be well characterized by the line 'Math is an asshole!'. This aversion and maybe even hatred may spring from experiences of failure and incapability as well as from too rare feelings of success, from devaluation of own achievements, from ignoring the impact of mathematics on modern life [1] and from prejudice about people who like and excel in math.

We want to examine whether certain mathematical topics could hopefully spark interest in mathematics. Then instructors face the challenge to motivate students for mathematics.

The following considerations are generalizations of suggestions we made for a redesign of the mathematics courses for students studying in the engineering degree course 'Technical Informatics' at Hochschule Bremen, City University of Applied Sciences. Right now, the Technical Informatics students share the mandatory three semester math courses with those studying automation and mechatronics. Up to now, they all learn in the first two semesters linear algebra, analytical geometry and analysis and in the third semester multidimensional calculus, stochastics and ordinary differential equations or Fourier analysis. In the future, Technical Informatics students should get in the third semester their own redesigned math course.

In a sense, what the example in [20] does for mechanical engineering we here try for technical informatics – only restricted to the third and last semester of a mathematical education which is oriented on engineering applications.

Very Soft Criteria

Very idealistically, a suitable module starts with an example well known to the students which illustrates the topic and emphasises its importance. Then the underlying math is developed. The students are eager to apply what they just learned to relevant examples. So they acquire *action-based knowledge* [20]. They also learn that in general there is much more to explore.

Besides these wishes it might be favourable if the modules in some way use and extend what the students have learnt in the first two semesters. At the same time however, it might give them a chance to establish a new relation to math when starting with a topic say in discrete mathematics from scratch. Obviously, guidance and staging determine whether students perceive some topic as new, fresh and seemingly unrelated to old stuff connected with bad experiences.

In our opinion, to learn mathematics today must be supported by using computer tools. It is preferable to integrate the use of a CAS like open source SAGE [15] which handles both classical numerics as well as more or less any discrete mathematics.

We would like to exclude some actual topics as for example artificial intelligence, big data, neuronal networks and the like. We consider e.g. the latter as not suitable because of say the lack of theoretical foundation. It is inherent that the result of a neuronal network depends fundamentally on its design, i.e. number of layers, activation functions etc, on its training and on its initialization. This generates a time consuming trial and error optimization of any number of parameters determining the net. Still, this process will generate neither more insight into the problem nor any more confidence in the results once new test data are presented. So in our opinion e.g. neuronal networks are more suitable as an option than as some suspiciously mandatory course students have to take.

Procedure

Not very imaginative we first conducted a literature research to see which mathematical topics other colleagues selected as suitable or even necessary for computer science students. Detailed results are presented in [18]. Here it may suffice to state that most of the two dozen German titles presented

the classical curriculum. Some colleagues additionally include differential equations or numerics. Outside the classical curriculum there are – in the order of decreasing frequency – interesting topics like algebraic structures, set theory, logic, graph theory, combinatorics, cryptology, mappings and functions, number theory, methods of proof, error correcting codes, order theory as part of set theory, Fourier analysis, (linear) optimization and computer graphics.

Not very surprisingly we ourselves already taught those topics as options, namely cryptology, error correcting codes, generative computer graphics, digital image processing etc – topics which all can be classified as *applied* computer science. Why not give an introduction to some of these topics in the math course of the third semester and at the same time generate interest for them as a later option? Then in the third semester the *Core Zero* and *Core* levels [20] of mathematics would have been established and at the same time the path to *Electives* and *Specialist Modules* [20] would be paved.

A Kit of Suggested Modules

Here, we suggest to choose from a couple of modules when designing a Math III course for computer science students. In alphabetic order

Algorithms, A We would distinguish numerical [8] and non-numerical algorithms [2], start with the classical sorting and searching algorithms and illustrate best, average and worst case time and space requirements [9]. Here we venturously propose to then ask the students to participate in the *Project Euler* <http://projecteuler.net>. They then are engaged, can join the competition, and they learn about the efficiency of their self-designed algorithms: the harder the problem the more important it is to find an efficient algorithm. In the Euler project, it should take less than a minute of CPU time on a modestly powered computer to solve any of the 600+ problems. A forum specific to each question may be viewed after the student has correctly answered the given question. The project Euler lets the participants use any computer program they know and like. However, because of time requirements and ease of comparing different solutions we would favour to ask all students to learn to use the open source CAS SAGE [15].

Error Correcting Codes, ECC We would recap fields, vector spaces, matrices, introduce arithmetic in \mathbb{GF}_2 , \mathbb{GF}_p and $\mathbb{GF}(p^m)$ and define generator- and parity-matrices. Hamming codes serve as first simple examples of linear $[n, n-m, 3]$ codes over \mathbb{GF}_q [19]. Cyclic codes with their generation and decoding may present another class of important codes. Reed-Solomon or BCH codes might serve as an appetizer for an option.

HDDs, CDs, any radio transmission etc obviously prove the importance of error correcting codes. Linear algebra obviously is indispensable even to talk about ECC. From old codes students can by extension, puncture, or concatenation construct new codes and check their error correction and error detection capabilities.

Freeform Curves and Surfaces, FF We would illustrate the importance of freeform surfaces say in the automobile industry. We then would recap curves and surfaces in parameter representation, introduce Bézier- and spline curves and how to use them to specify surfaces. B-Splines could pave the way to NURBS etc [7]. Matrix transformations scale, shift and rotate such curves and surfaces.

FF combine linear algebra and analysis. Visualization clarifies the impact of modifications. Interpolation and approximation or even differential geometry [21] are possible extensions. The students benefit from conspicuity and work constructively, e.g. construct a helix through two or more given points in space.

(Descriptive) Geometry, DG We would start with examples like CAD, GPS, or VR etc to show the relevance. We would then recap the canonical geometric objects in \mathbb{R}^2 or \mathbb{R}^3 and their representation, then introduce curves, surfaces and bodies in general by Constructive Solid Geometry, CSG or in Boundary Representation, B-Rep [7] and handle projections and perspective. The classical book [5] offers much more related material including exercises with solutions.

As in the case of FF, students benefit from clearness, they work constructively combining analysis and analytical geometry, solve problems in [5] or construct and explore geometric objects in space by visualizing them from different view points using projection or perspective.

Networks as Graphs, NG Euler laid the foundations of graph theory with the famous *Königsberg bridge problem* which serves as a good starting point. We would define graphs, paths and cycles. Graphical

representation of graphs introduces planarity or non-planarity. Dijkstra's algorithm for finding a shortest path between two vertices of a weighted graph illustrates routing problems. Trees describe e.g. how algebraic expressions are evaluated by a computer or index sorted data structures like file systems or data bases. Another relevant algorithm colours graphs. Compilers do the same when they bind variables to registers.

The students gain insight into routing problems in networks (data, railways, streets etc) or networks-on-chip. E.g. the traveling salesman problem exhibits the classical example of a \mathcal{NP} -complete problem.

Cryptology, C We might take the historical approach in order to illustrate the interplay between attack and defense starting with Caesar and Vigenère, then maybe introduce arithmetic in finite fields \mathbb{GF}_p and $\mathbb{GF}(p^m)$ in order to present symmetric methods DES, Triple-DES and AES, proceeding to asymmetric methods DHM and RSA. Students can try brute force attacks or experience the different cryptographic algorithms interactively [12]. They might find new methods like elliptic curve cryptography [13] or post quantum cryptography [4] [16] intriguing enough to consider a corresponding option later.

Optimization, O We would recap maximizing or minimizing functions of one variable, generalize to functions of several variables and introduce the least squares methods (regression) without and with side conditions. The simplex might at last lead to problems which are solved by integer linear programming (planing of networks and tours).

Students obviously pick up on calculus, generalize it, get to know important applications in science, metrology etc. may apply these methods to e.g. own statistical data and benefit from the geometrical representation of the simplex method [2] where they again benefit from their linear algebra armamentarium they acquired in the first two semesters.

Stochastics, S We would have to introduce basic probability theory in order to present as an initial example Google's page ranking [14] by modelling it as a Markov chain and calculating its limiting distribution. Then we could delve more into Markov chains in classifying their states or introduce probability distributions to model e.g.

RAMS systems by queueing theory [6]. However, we would need basics of probability distributions to model even simple M/M/1 queueing systems. E.g. [3] in detail describe various scheduling strategies in existing operating systems whereas e.g. [10] present the underlying general theory (however, with example computer systems predating 1984). Between these extremes teaching staff would have to focus on only a few, simple scheduling strategies for the benefit of a deeper mathematical treatment.

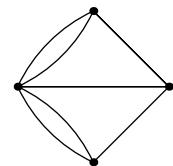
We now consider our suggested eight modules in the light of the eight competencies specified in [20]. Instead of relating each module to each and every competency we here restrict ourselves to consider each module in the light of exactly one competency:

Thinking mathematically – in A Complexity abstracts the execution of algorithms from real computing machines. Abstraction, this very mathematical procedure leads to an unbiased measure of the inherent difficulty of an algorithm in terms of storage and time (or gates or communication overhead etc) as a function of the size of the problem.

Posing and solving mathematical problems – in ECC To a linear code given by a generator matrix one wants to determine e.g. its error correcting capabilities (via the biggest number d such that any set of $d - 1$ columns of the parity matrix are linear independent) and one wants to devise algorithms to decode some word (determination of the nearest or most probable codeword).

Reasoning mathematically – in FF The more data points are to be interpolated the higher is the degree of the interpolating polynomial and the more extremes, i.e. unwanted oscillations occur. This fact is overcome by *piecewise* polynomial interpolation which is the basic idea of splines and b-splines etc.

Modelling mathematically – in DG The graph below models the above mentioned *Königsberg bridge problem* by identifying the four connected components of the Königsberg area by vertices and the seven bridges by connecting edges.



Then Euler in 1736 proved that an Eulerian circuit does not exist.

Representing mathematical entities – in NG The representation of a graph by adjacency list (sparse), adjacency matrix (dense) or inci-

dence matrix considerably determines the cost of operations on the graph.

Handling mathematical symbols and formalism – in C For sure, cryptology like any other mathematical discipline uses its own symbols and formalisms. For example, the names and symbols of bit operations in DES or AES are inherited either from (Boolean) logic (\wedge, \vee, \neg) or from logic circuits (gates $\square, \curvearrowright, \triangleright$) whereas the symbols of RSA are naturally primarily inherited from number theory.

Communicating in, with, and about mathematics – in O The teaching staff has the task to help the students to develop their abilities to communicate about mathematics by encouragement, insistence and practice. Students will of their own volition develop these skills in *A* when talking about and comparing different solutions of Euler problems.

Making use of aids and tools – in S Monte Carlo simulations allow to solve problems which are analytically intractable. In general, they can only be carried out by computerized tools.

Summary

Mathematics is the base of many areas of applied computer science. It does so in different ways: providing a language to describe problems as well as providing methods to find and assess solutions. Hence it does not suffice to present just one of the above suggested modules. One has to make a wise selection. It has to be expected that on one hand we as teaching staff will tend to give an option instead of an appetizer. However, as coaches and guides we have to restrain ourselves and focus on the very mathematical core of some chosen module. On the other hand, students need enough basic math in order to be able to solve problems in the very field on their own. (Again, a CAS like SAGE helpfully is suitable for *all* topics.) It is our task to find an adequate balance.

Bibliography

- [1] **Aigner, M.:** *Mathematics Everywhere*. American Mathematical Society 2010

- [2] **Alt, H.:** *Höhere Algorithmik*. Freie Universität Berlin 2011
<http://www.pnjb.de/uni/ws1011/hoehere-algorithmik.pdf>
- [3] **Arpaci-Dusseau, R.; Arpaci-Dusseau, A.:** *Operating Systems – Three Easy Pieces*. Arpaci-Dusseau Books 2015, <http://pages.cs.wisc.edu/~remzi/OSTEP>
- [4] **Bernstein, D.J.; Buchmann, J.; Dahmen, E. (Eds.):** *Post-Quantum Cryptography*. Springer 2009
- [5] **Coxeter, H.S.M.:** *Introduction to geometry*. Wiley 1961 <http://plouffe.fr/simon/math/Wiley-Coxeter-IntroductionToGeometry.pdf>
- [6] **Feller, W.:** *An Introduction to Probability Theory and its Applications*. Vol I and Vol II, Wiley 1968
- [7] **James D. Foley, J.D.; van Dam, A.; Feiner, S.K.; Phillips, R.L.:** *Introduction to Computer Graphics*. Addison-Wesley 1997
- [8] **Heath, M.T.:** *Scientific Computing – An Introductory Survey*. McGraw-Hill 2002
- [9] **Knuth, D.E.:** *The Art of Computer Programming (TAOCP)*. Volumes 1–3 and right now 4A, Addison-Wesley, 1997-2011 http://broiler.astroetry.net/~kilian/The_Art_of_Computer_Programming-Vol1.pdf fundamental http://plouffe.fr/simon/math/The_Art_of_Computer_Programming_vol_2.pdf seminumerical
- [10] **Lazowska, E.D.; Zahorjan, J.; Graham, G.S.; Sevcik, K.C.:** *Quantitative System Performance – Computer System Analysis Using Queueing Network Models*. Prentice-Hall, 1984, <https://homes.cs.washington.edu/~lazowska/qsp/>
- [11] **Risse, Th.:** *List of Publications*. <http://www.weblearn.hs-bremen.de/risse/papers/papers.htm>
- [12] **Risse, Th.:** *Mathematical Puzzles – interactively investigate (arithmetic in) finite fields, cryptology, compression etc.*
<http://www.weblearn.hs-bremen.de/risse/MAI/docs/ePuzzles.pdf>
- [13] **Risse, Th.:** *Teaching Elliptic Curves Cryptography – Reflecting Experiences*. 24th Int. Conf. *Science in Practice*, Hochschule Bremen, May 14th – 15th, 2007, No.84 in [11]
- [14] **Risse, Th.:** *Markov Chains and Googles Page Ranking*. London South Bank University, February 27th, 2008, No.91 in [11]
- [15] **Risse, Th.:** *ThR: SAGE – the Computer Algebra System to End up all Computer Algebra Systems?* 15th SEFI MWG Seminar, Wismar, June 20th – 23rd, 2010, No.110 in [11]
- [16] **Risse, Th.:** *How SAGE helps to implement Goppa Codes and the McEliece Public Key Crypto System*. Ubiquitous Computing and Communication Journal, UbiCC, ISSN 1992-8424, Special Issue on 5th International Conference on Information Technology (ICIT'11) 2011, No.116 in [11]

- [17] **Risse, Th.:** *Attituden von Informatik-Studienanfängern – am Beispiel der Mathematik.* FBTI-workshop Studierfähigkeit der Erstsemester, Fachbereichstag Informatik, FH Fulda, 25.4.2012, No.121 in [11]
- [18] **Risse, Th.:** *Mathematik III für Technische Informatiker?* 1st Northern-Lights Symposium on Engineering Education, HCU, 27.4.2018, No.152 in [11]
- [19] **Roth, R.:** *Introduction to Coding Theory.* Cambridge University Press 2006.
- [20] **SEFI MWG:** *A Framework for Mathematics Curricula in Engineering Education.* 2013 <http://sefi.htw-aalen.de/Curriculum/Competencybasedcurriculuminclads.pdf>
- [21] **Shifrin, T.:** *Differential Geometry – A First Course in Curves and Surfaces.* University of Georgia 2013
<http://alpha.math.uga.edu/~shifrin/ShifrinDiffGeo.pdf>

Autor

Prof. Dr. rer. nat. Thomas Risse
Faculty EEE & CS
Hochschule Bremen, City University of Applied Sciences
Flughafenallee 10, D-28199 Bremen
E-Mail: risse@hs-bremen.de

Peter Junglas

Simulationsprogramme für die Lehre – Was kommt nach Applets?

Auszug. Dass Java-Applets von den Browsern nicht mehr unterstützt werden, hat viele Entwickler, die Simulationsprogramme für die Lehre erstellen, gezwungen, ihre Programme nach JavaScript zu portieren. Für das PhysBeans-Projekt stellt dies allerdings eine besondere Herausforderung dar. Daher werden hier alternative Möglichkeiten untersucht, die Ergebnisse des Projekts weiterhin verfügbar zu machen.

Physikalische Simulationen mit Java-Applets

In der Physikausbildung ist der Einsatz von Simulationsprogrammen seit langem etabliert. Stellvertretend für die riesige Anzahl frei verfügbarer Programme sei hier auf umfangreiche Projekte wie Physlet [1] und das verwandte EJS (“Easy Java Simulations”) [2], das PhysBeans-Projekt [3] des Autors sowie auf die erstaunlichen Sammlungen [4, 5] verwiesen.

Zur Implementierung wurden dabei aus guten Gründen Java-Applets verwendet: Java stellt alle benötigten Konstruktionen und eine standardisierte Bibliothek für numerische und graphische Funktionen zur Verfügung, die Programme sind unabhängig vom Rechnertyp oder Betriebssystem, und die Applets lassen sich bequem in Webseiten integrieren, die Handlungsanweisungen oder Hintergrundinformationen enthalten.

Im Zuge der Umstellung auf HTML 5 und JavaScript haben aber die Browser-Hersteller seit 2015 die Unterstützung von Plugins im allgemeinen – und damit insbesondere von Java-Applets – eingestellt. Mit der Version Java 9 wurden dann die appletbezogenen Klassen in der Sprachdefinition als veraltet bezeichnet und sollen in zukünftigen Versionen wegfallen. Ein weiteres Problem entstand durch die zunehmende Verbreitung mobiler Endgeräte mit Android- oder iOS-Betriebssystem, auf denen nur Java-Grundfunktionen unterstützt werden, so dass Applets dort nicht laufen.

Als Ersatz für Java-Applets mit der Möglichkeit, Simulationen wie bisher direkt in Webseiten einbinden zu können, kommt zunächst nur JavaScript in Frage. Die Portierung einer großen Codebasis von Java nach

JavaScript ist allerdings sehr aufwändig, da die Sprachkonstrukte und Bibliotheken, vor allem zur Implementierung einer graphischen Benutzerschnittstelle, in beiden Sprachen sehr unterschiedlich sind. Trotzdem haben die oben aufgeführten Projekte – mit Ausnahme von PhysBeans – inzwischen weitgehend auf JavaScript umgestellt: Das EJS-Projekt heißt nun “Easy Java/JavaScript Simulations” und erzeugt seit Version 5 Java- und JavaScript-Programme [6], die dritte Auflage von [1] (inzwischen online vertrieben [7]) verwendet EJS 5 und enthält JavaScript-Versionen der bisherigen Programme, und auch die Applets der beiden Sammlungen [4, 5] wurden manuell bzw. im Rahmen des SwingJS-Projekts [8] konvertiert.

Die Konvertierung von PhysBeans scheiterte dagegen bisher nicht nur an den fehlenden Ressourcen, sondern vor allem an der etwas anderen Zielsetzung des Projekts: Es stellt einen Baukasten zur Verfügung, der es ermöglicht, ohne Programmierkenntnisse Applets für die eigenen Bedürfnisse zu konstruieren oder die vorhandenen Beispiele anzupassen. Es verwendet dazu das JavaBeans-Framework, um Komponenten zu erstellen, die in einer Entwicklungsumgebung graphisch manipuliert werden können. Eine analoge Infrastruktur existiert in JavaScript aber nicht. Im Folgenden sollen daher alternative Möglichkeiten untersucht werden, mit deren Hilfe die Ideen und Ergebnisse – und soweit wie möglich auch der Code – des PhysBeans-Projekts für moderne Browser und Endgeräte verfügbar gemacht werden können.

Das PhysBeans-Projekt

Das PhysBeans-Projekt stellt ein Toolkit zur Verfügung, das die Erstellung von virtuellen Experimenten für physikalische (oder auch mathematische) Simulationen in der Form von Java-Applets ermöglicht. Es wurde 1998 begonnen, um Simulationen für die CD von [9] zu entwickeln, und wuchs im Lauf von 15 Jahren zu einem vielseitigen Werkzeug heran, mit dem auch didaktische Fragestellungen (etwa in [10]) oder numerische Probleme der Simulation (z. B. in [11]) untersucht wurden.

Inzwischen werden auf der Homepage des Projekts [12] über 160 Applets frei zur Verfügung gestellt, vor allem aus den Bereichen Mechanik, lineare Schwingungen (s. Abb. 1), nichtlineare Schwingungen und chaotische Systeme, Wellen, Optik (s. Abb. 2), Elektrodynamik (s. Abb. 3), Thermodynamik und Quantenmechanik (s. Abb. 4).

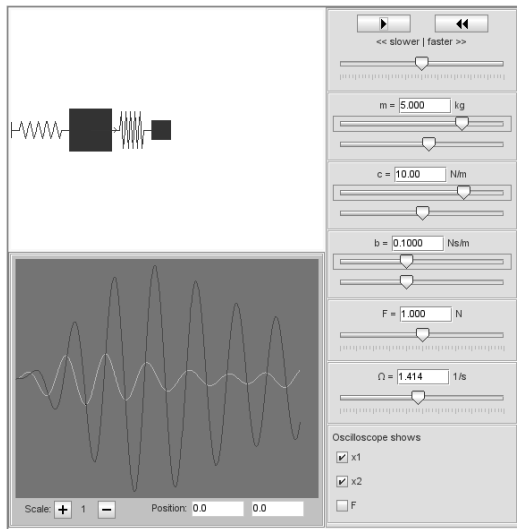


Abb. 1: Schwingerkette

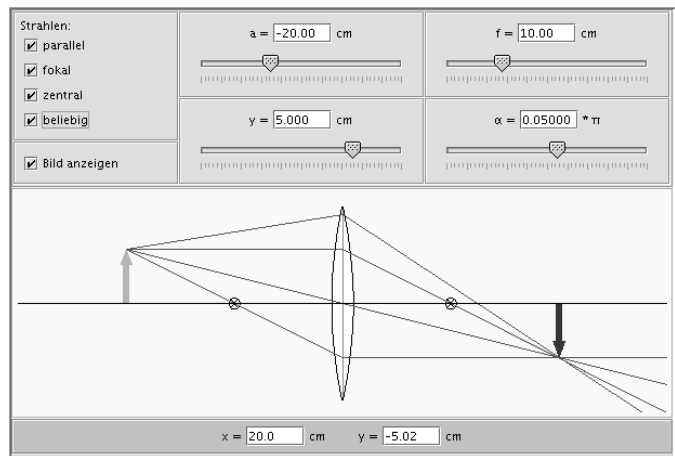


Abb. 2: Dünne Sammellinse

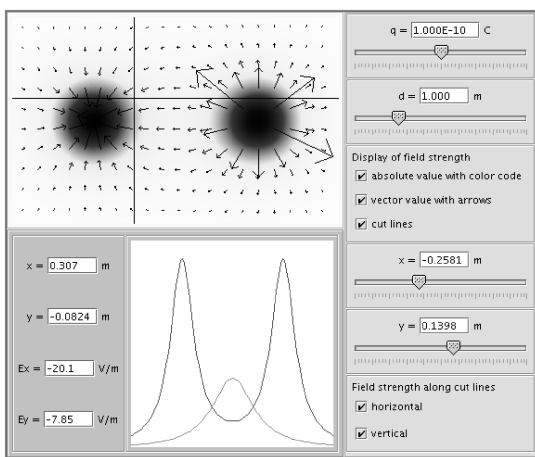


Abb. 3: Elektrischer Dipol

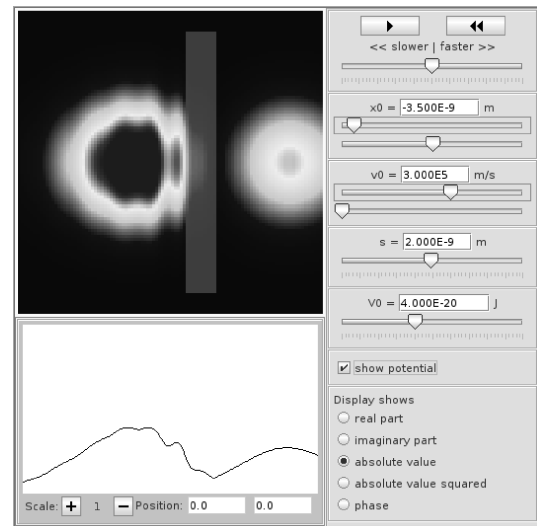


Abb. 4: Tunneleffekt

Die Basis des Toolkits ist die physbeans-Bibliothek, die eine große Zahl an Komponenten in der Form von JavaBeans [13] zur Verfügung stellt. Damit können in einer Entwicklungsumgebung auf graphische Weise eigene Simulationen erstellt werden, ohne dass vertiefte Java-Kenntnisse nötig sind. Ausgangspunkt der Entwicklung ist stattdessen ein Flussdiagramm, das alle Komponenten und die zwischen ihnen ausgetauschten Nachrichten darstellt (s. Abb. 5). Das genaue Vorgehen und viele Beispiele werden beschrieben in [3].

Der Java-Quellcode der Bibliothek umfasst etwa 85000 Zeilen. Dazu kommen knapp 50000 Zeilen für die damit erstellten Applets, die i. W. mit

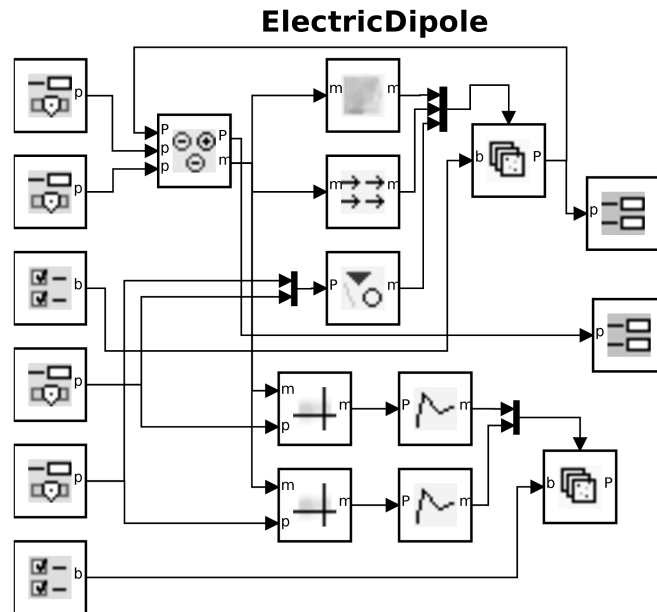


Abb. 5: Flussdiagramms des Applets “Elektrischer Dipol” (Abb. 3)

Hilfe der Entwicklungsumgebung aus dem graphischen Modell generiert wurden. Alle Quellen, das fertige jar-File sowie die Applets sind unter den üblichen Bedingungen auf der PhysBeans-Homepage [12] frei verfügbar.

Maßnahmen mit geringen Quellcode-Änderungen

Ist man bereit, auf die direkte Einbindung des Applets in eine Webseite zu verzichten, gibt es eine einfache Möglichkeit, mit geringen Änderungen am Quellcode die PhysBeans-Programme weiter zur Verfügung zu stellen: die Java Web-Start-Technologie [14], eine Methode, um Java-Anwendungen über das Internet zu laden und zu starten. Aus der Sicht des Anwenders ändert sich nicht viel: Statt des eingebetteten Applets enthält die Webseite einen Link, nach dessen Anklicken die Java-Anwendung über das Netz geladen und in einem eigenen, vom Browser unabhängigen Fenster gestartet wird.

Damit dies funktioniert, müssen die Applets in Java-Applikationen umgeschrieben werden. Dazu bekommen sie eine `main`-Funktion, die die Aufgaben übernimmt, die sonst der Browser ausführt: Erzeugen eines Fensters (`JFrame`), Initialisieren (`init()`) und Starten des Applets (`start()`). Diese Funktionalität kann in eine globale Elternklasse ausgelagert werden, so dass die Änderungen an den einzelnen Applets marginal sind. Ein weiterer Unterschied besteht in der Art, wie Applets und Applikationen auf

Ressourcen, z. B. Bilddateien, zugreifen können, aber auch das lässt sich leicht anpassen. Auf diese Weise lassen sich Programme erzeugen, die sowohl als Applets als auch als Applikation gestartet werden können. Da die PhysBeans-Applets von Anfang so konzipiert worden sind, waren im konkreten Fall überhaupt keine Änderungen am Javacode nötig.

Da das Laden eines Programms aus dem Netz grundsätzlich ein hohes Risiko birgt, erfordert Java Web Start zusätzlich, dass man ein Zertifikat erstellt (oder kauft) und damit die Jar-Dateien signiert. Schließlich muss noch eine Startdatei (JNLP-Datei) für jedes zu startende Applet erstellt werden, auf das der Link der Webseite verweist. Dies alles ist wenig aufwändig und lässt sich leicht automatisieren.

Das klingt nach einer einfachen und bequemen Lösung: Abgesehen von der etwas schlechteren Verzahnung zwischen Text und Simulation laufen alle Programme bei minimalem Aufwand genau wie vorher. Aber das Verfahren hat auch Nachteile: Zunächst gibt es häufig Probleme mit der Sicherheitskontrolle, vor allem bei Verwendung eines selbst erstellten Zertifikats. Damit dies akzeptiert wird, muss man die Sicherheitsstufe auf dem Client-Rechner heruntersetzen, was je nach Betriebssystem einfach oder fast unmöglich ist. Vor allem aber benötigt der Anwender eine lokale Installation des Java-Laufzeitpakets. Dies lässt sich im Rahmen etwa einer Vorlesung vielleicht auf die Studenten abwälzen, ist aber dem allgemeinen Anwender in Zeiten komplett gekapselter Apps kaum noch vermittelbar. Aus diesem Grund hat Oracle den Java Web-Start-Mechanismus als “veraltet” eingestuft und verkündet, die Unterstützung in kommenden Versionen einzustellen [15].

Maßnahmen mit großen Quellcode-Änderungen

Das Problem, für verschiedene Plattformen oder unterschiedliche Software-Umgebungen zu entwickeln, haben natürlich auch andere. Insbesondere bei der Spiele-Programmierung sind die Anforderungen sehr ähnlich, die Komplexität dagegen häufig wesentlich größer als bei physikalischen Simulationen. Daher wurden verschiedene Werkzeuge erstellt, die Spiele-Entwicklern eine plattform-unabhängige Programmierung mit komplexer Grafik und hoher Performance ermöglichen.

libGDX [16] ist ein solches Framework, das komplett auf Open-Source-Komponenten beruht. Es hat eine sehr aktive Nutzerschaft und

ist gut dokumentiert, u. a. durch mehrere Lehrbücher (z. B. [17]). Es ermöglicht eine Entwicklung komplett in Java, wobei graphische und andere “hardware-nahe” Funktionen durch Aufrufe der libGDX-Bibliothek ersetzt werden. Mit Hilfe diverser Tools werden die Programme dann automatisch auf andere Plattformen portiert, so dass gleichzeitig eine Java-Applikation, eine HTML5/JavaScript-Anwendung und Apps für Android und iOS erzeugt werden.

Für eine Portierung von PhysBeans müssen alle graphischen Komponenten für die libGDX-Plattform neu entwickelt werden, während sämtliche physikalischen, mathematischen und sonstigen Komponenten unverändert übernommen werden können. Da die Graphik-Funktionen in libGDX sehr systemnah sind, ist der Portierungsaufwand relativ hoch, dafür lässt sich das typische Aussehen aber auch recht detailliert nachbilden. Die schematische Darstellung physikalischer Objekte (“View-Beans”) ist dabei noch vergleichsweise einfach, dagegen stellt die Nachbildung der Ein-/Ausgabe-Bausteine (“Input-/Output-Beans”) eine größere Herausforderung dar. Dass dies grundsätzlich möglich ist, zeigt die erfolgreiche Portierung des Applets “Dünne Sammellinse” (Abb. 2), für das auf der PhysBeans-Homepage eine libGDX-basierte JavaScript-Version (Abb. 6) sowie eine Android-App zur Verfügung gestellt werden.

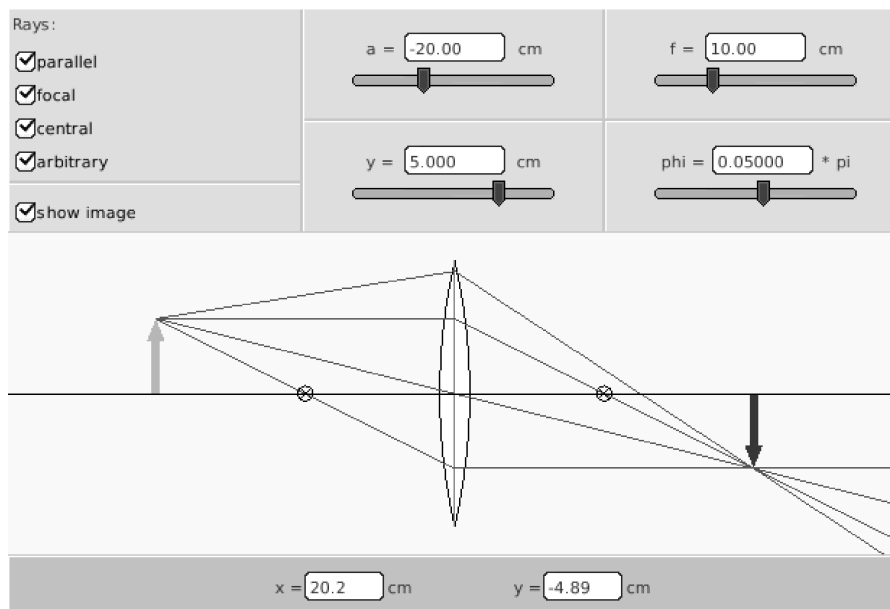


Abb. 6: JavaScript-Version des Applets “Dünne Sammellinse” (Abb. 2)

Eine vollständige Portierung von PhysBeans auf libGDX erfordert einige Anstrengung. Allerdings ist die Zahl verschiedener Input-/Output-Beans nicht groß, sie werden häufig wiederverwendet. Für die vielen View-

Beans lässt sich vermutlich durch geschicktes Einführen einer Zwischenschicht und geeignete objekt-orientierte Konstruktionen der Aufwand auf lange Sicht deutlich verringern. Dafür erhält man direkt vom Browser innerhalb einer Webseite ausführbare JavaScript-Versionen – und sogar Android- oder iOS-Apps, falls gewünscht – bei gleichzeitigem Erhalt des größten Teils der vorhandenen Codebasis. Vor allem aber bleibt für Nutzer von PhysBeans die Möglichkeit bestehen, physikalische Simulationsprogramme weitgehend ohne eigenen Java-Code zu erstellen.

Komplette Neuentwicklung in JavaScript

Wenn es nur um die Portierung der im Rahmen des PhysBeans-Projekts erstellten Applets geht – ohne die Möglichkeit, weitere Applets mit graphischen Methoden zu erstellen –, kommt eine Komplet-Portierung nach JavaScript in Frage. Die Möglichkeiten von HTML5, insbesondere das Canvas-Element und die Web-Worker-API [18], sowie hilfreiche Bibliotheken wie Numeric Javascript [19] stellen zumindest die grundsätzlich benötigten Werkzeuge dafür zur Verfügung.

Außerdem ist die Performance von JavaScript-Anwendungen durch Verfahren wie Just-in-time-Kompilierung stark angestiegen. Ob das allerdings reicht für die hohen Anforderungen der PhysBeans-Applets zur Quantenmechanik (s. Abb. 4), ist fraglich: Trotz ausgefeilter numerischer Verfahren zur Lösung der zweidimensionalen Schrödingergleichung incl. transparenter Randbedingungen mussten Handoptimierungen für Matrixberechnungen und Speicherzugriffe implementiert werden, um halbwegs flüssige Animationen in den Java-Applets zu erzielen [11]. Für alle anderen bisher betrachteten Anwendungsgebiete sollte aber die Geschwindigkeit der Programme keine Rolle mehr spielen.

Der hohe Aufwand – bei reduziertem Nutzen gegenüber dem ursprünglichen Projekt – stand bisher einer kompletten Portierung der Applets entgegen. Man kann hoffen, dass automatische Java-nach-JavaScript-Compiler wie SwingJS zukünftig in der Lage sein werden, auch komplexe Swing-Anwendungen automatisch zu übertragen.

Komplette Neuentwicklung in Matlab

Löst man sich immer weiter von den ursprünglichen Anforderungen und stellt die Frage, wie man physikalische Simulationen für die Lehre heute entwickeln würde, käme auch ein Werkzeug wie Matlab in Frage. Es stellt eine große Vielfalt komplexer numerischer Funktionen und ausgefeilter graphischer Möglichkeiten zur Verfügung bis hin zur Werkzeug-unterstützten Entwicklung graphischer Benutzeroberflächen. Es ist zwar proprietär, aber weit verbreitet, insbesondere auch in der Lehre. Alternativ könnte man auf Open-Source-Projekte wie Scilab [20] oder Octave [21] zurückgreifen, die beide Matlab ähnlich sind und einen erstaunlichen Funktionsumfang bieten.

Wie stark sich die Entwicklung damit gegenüber einer Java-Programmierung vereinfacht, zeigt die Beispiel-Simulation “Schwingerkette (s. Abb. 7). Ihr gesamter Code umfasst lediglich 263 Zeilen, dazu kommt eine Binärdatei, die die Informationen zum Aufbau der Benutzeroberfläche enthält und vom GUI-Builder automatisch erstellt worden ist. Das Programm stützt sich dabei wesentlich auf Matlabs Solver zur Lösung von Differentialgleichungen sowie die komplexen Plot-Routinen.

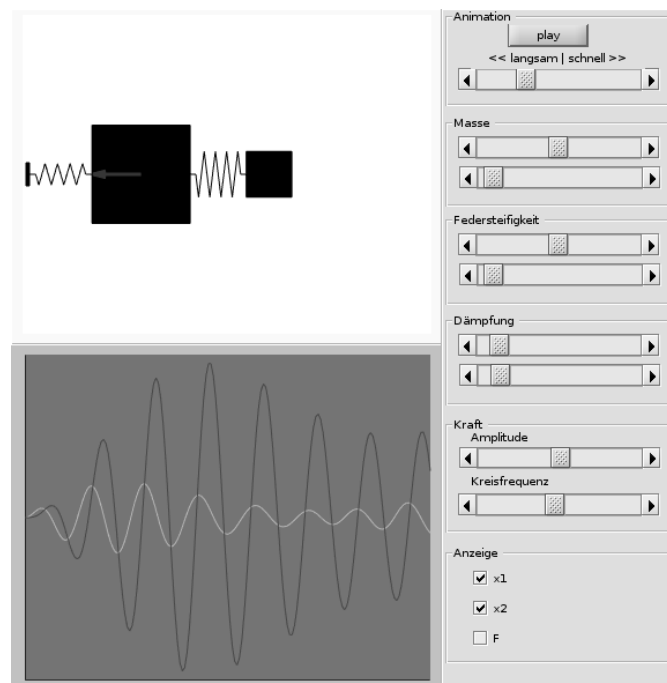


Abb. 7: Matlab-Version des Applets “Schwingerkette (Abb. 1)

Tatsächlich setzt der Autor schon lange Matlab als Prototyping-Werkzeug ein, bevor er die mühsamere Implementierung in Java beginnt.

Damit wurden nicht nur die komplexen numerischen Algorithmen der Wellen- und Quantenmechanik-Applets entwickelt, sondern auch Beispiel-ergebnisse berechnet, mit denen die Applet-Resultate getestet wurden. Aufgrund von Matlabs optimierten Numerik-Funktionen war dabei die Performance sogar für die hohen Anforderungen der Quantenmechanik-Beispiele ohne weitere Optimierungen ausreichend.

Verglichen mit dem PhysBeans-Projekt bleibt dabei bisher eine wesentliche Anforderung offen: die direkte Einbindung in Webseiten, möglichst ohne eine Matlab-Installation beim Client. Für die Verbreitung fertiger Matlab-Applikationen (“Deployment”) stellt Mathworks zwar diverse Werkzeuge zur Verfügung [22], die aber alle die hier gestellten Anforderungen verfehlen: Mit dem “Matlab Coder” kann ein C/C++-Programm erstellt werden, das aber keine graphischen Funktionen unterstützt. Mit dem “MATLAB Compiler SDK” werden Bibliotheken u. a. für Java und Python erzeugt, aber nicht für JavaScript. Und der “MATLAB Web App Server” erlaubt zwar die Verbreitung von Applikationen über das Internet, benötigt dazu aber einen dezidierten Server in einer streng geschützten Umgebung. Scilab bietet eine Cloud für Scilab-basierte Web-Anwendungen, aber ebenfalls keine clientseitigen Lösungen, Octave beinhaltet keinerlei Deployment-Unterstützung.

Fazit

Damit die Simulationsprogramme des PhysBeans-Projekts trotz fehlender Browser-Unterstützung weiterhin funktionieren, werden sie auf der Projekt-Homepage über den Java-Web-Start-Mechanismus zur Verfügung gestellt. Wie oben beschrieben, ist dies aber leider nur eine temporäre Lösung.

Für größere Portierungsaktionen sind leider die Ressourcen des Autors nicht ausreichend – freiwillige Mitarbeit wäre hoch willkommen. Den geringsten Aufwand macht sicher die Portierung auf die libGDX-Plattform. Allerdings ist bei OpenSource-Projekten immer fraglich, wie lange die bisherigen Entwickler noch weiter machen bzw. ob sich neue finden. Auch die Stabilität der API ist nicht gewährleistet, so ließ sich das vor vier Jahren erstellte Beispiel bei einem Test vor zwei Jahren nicht mehr übersetzen.

Eine Neuentwicklung in JavaScript kommt momentan überhaupt nicht in Frage, es bleibt die Hoffnung auf bessere automatische Tools.

Vom Umfang her wäre eine Matlab-Version vielleicht sogar möglich, scheitert aber bislang an den fehlenden Deployment-Werkzeugen für Client-Anwendungen.

Seit kurzem ist allerdings ein weiterer Ansatz in Sicht, der die Portierung überflüssig machen könnte: Der WebAssembly-Standard [23] definiert ein Binärformat, das die Ausführung von Code nahezu mit der Geschwindigkeit kompilierter Programme in einem Browser ermöglicht. Er wird von den wichtigsten Browsern inzwischen unterstützt. Dazu kommen Compiler, die C/C++-Programme in das WebAssembly-Format übertragen. Geplante Weiterentwicklungen sollen die direkte Kompilation von Java ermöglichen. Damit könnte der vorhandene PhysBeans-Code auch in zukünftigen Webbrowsern verfügbar gemacht und das Projekt weitergeführt werden, statt die Energie auf Portierungen zu verschwenden.

Literaturverzeichnis

- [1] **Christian, W.; Belloni, M.:** *Physlet Physics*. Prentice Hall Upper Saddle River, USA, 1. Aufl. (2003).
- [2] **Christian, W.; Esquembre, F.:** *Modeling Physics with Easy Java Simulations*. *Physics Teacher*, **45**, 475-480 (2007).
- [3] **Junglas, P.:** *cliXX PhysBeans*. Verlag Harri Deutsch Frankfurt (2008).
- [4] **Falstad, P.:** *Math and physics applets*.
<http://www.falstad.com/mathphysics.html/> .
- [5] **Fendt, W.:** *Java-Applets zur Physik*. <http://www.walter-fendt.de/ph14d/> .
- [6] **Esquembre, F.:** *Easy Java Simulations Wiki*.
<http://www.um.es/fem/EjsWiki/Main/HomePage> .
- [7] **Christian, W.; Esquembre, F.:** *Physlet® Physics 3Ed*.
<http://www.compadre.org/books/Physlets-3E> .
- [8] **Hanson, R.:** *Why bother converting Java applets to JavaScript?*
<https://chemapps.stolaf.edu/swingjs/site/swingjs/examples/index.html> .
- [9] **Stöcker, H. (Hrsg.):** *Taschenbuch der Physik mit CD-ROM*. Verlag Europa-Lehrmittel Haan-Gruiten, 7. Aufl. (2014).
- [10] **Junglas, P.:** *Einsatz von Applets in der Physik-Ausbildung - Fallstudie Nichtlineare Systeme und Chaos*. *Global J. of Engng. Educ.*, **7**, 337-347 (2003).
- [11] **Junglas, P.:** *Transparent boundary conditions for simulation programs*. *Wismarer Frege-Reihe*, **3**, 60-65 (2010).

- [12] **Junglas, P.:** *Homepage des PhysBeans-Projekts.*
<http://www.peter-junglas.de/fh/physbeans/index.html> .
- [13] **Zwintzsch, O.:** *Software-Komponenten im Überblick: Einführung, Klassifizierung & Vergleich von JavaBeans, EJB, COM+, .Net, CORBA, UML 2.* Springer Nature Campus Berlin (2004).
- [14] **Oracle Corporation.:** *Overview of Java Web Start Technology.*
<https://docs.oracle.com/javase/10/deploy/overview.htm> .
- [15] **Oracle Corporation.:** *Java Client Road Map Update 2018-03-05.*
<http://www.oracle.com/technetwork/java/javase/javaclientroadmapupdate2018mar-4414431.pdf> .
- [16] **Zechner, M.:** *libGDX Cross-platform Game Development.*
<https://libgdx.badlogicgames.com/> .
- [17] **Nair, S. B.; Oehlke, A.:** *Learning LibGDX Game Development.* Packt Publishing Birmingham, 2. Aufl. (2015).
- [18] **Lawson, B.; Sharp, R.:** *Introducing HTML5.* Pearson Education San Francisco, 2. Aufl. (2011).
- [19] **Loisel S.:** *Numeric Javascript.*
<http://numericjs.com/> .
- [20] **scilab enterprises:** *Scilab – Open source software for numerical computation.*
<http://www.scilab.org/> .
- [21] **Eaton, J. W.:** *GNU Octave Scientific Programming Language.*
<https://www.gnu.org/software/octave/> .
- [22] **The MathWorks, Inc.:** *Mathworks – Desktop and Web Deployment.*
<https://de.mathworks.com/solutions/desktop-web-deployment.html> .
- [23] **Haas, A.; Rossberg, A.; et al.:** *Bringing the web up to speed with WebAssembly.* In: Proc. 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, New York, USA, 185-200. (2017).

Autor

Prof. Dr. rer. nat. Peter Junglas
 Private Hochschule für Wirtschaft und Technik Vechta/Diepholz
 Schlesierstraße 13a
 D-49356 Diepholz
 E-Mail: peter@peter-junglas.de

**Helena Barbas, Franz Konieczny,
Alexander Lohse, Thomas Schramm**

oHMint: An Online Mathematics Course and Learning Platform for MINT Students¹

Abstract. The oHMint project is an initiative with the goal of providing an online course and learning platform for STEM students of higher mathematics. Its flexible design allows it to be used for self-study as well as blended learning scenarios including flipped classrooms. In a pilot project in 2017/18 the chapter *Differential Calculus* is being produced as a prototype of an oHMint unit. This pilot is organized and funded through the Hamburg Open Online University, with technical support and implementation by integral-learning GmbH. It is intended as a catalyst for the future development of the full oHMint course spanning four semesters. A unique characteristic of oHMint is its broad supporting base among a variety of German institutions in the form of the OMB+ consortium.

Introduction

Engineering mathematics is the main tool to describe the world and to simulate its processes. A profound mathematical education is therefore essential for future engineers and scientists. Digitalization is becoming increasingly important in teaching and studying at university level. More and more online learning tools are being developed to address the current need for educational resources that go well beyond the classical textbook by providing high temporal and spatial flexibility for users while at the same time offering an interactive learning environment. In this spirit, the oHMint project was initiated through the OMB+ consortium, a group of 14 German institutions of higher education. The acronym stands for **online Higher Mathematics (for m)int**. It is the aim of oHMint to provide university students of STEM degree programmes with a modern way of transitioning smoothly from a high school level of mathematics to a bachelor degree level. The OMB+ consortium has already developed the online mathematics bridging course OMB+, which is widely used and recommended as a preparation for those interested in a STEM university degree programme. This experience is a valuable asset for oHMint.

Currently, in a pilot project with a lifespan of 14 months one base unit of oHMint is being created. This serves as an opportunity to test new didactical approaches for mathematics online courses as well as innovative types of exercises addressing the specific needs of the current student generation.

In this article we recall some information from [1, 2], where the general scope of the oHMint project was described (Sections 2, 3), and focus on using oHMint in a flipped classroom learning scenario (Section 4).

¹ This contribution is a permitted copy of a paper presented at ASIM 2018 Symposium on Simulation Engineering in Hamburg.

1. Background of oHMint

1.1 OMB+ mathematics bridging course

The OMB+ consortium has developed and maintains the well-accepted “Online Mathematics Bridging Course Plus” OMB+ [3] under the auspices of the TU9, a group of nine important German technical universities. The course is based on the recommended cosh catalogue [4], which sets a standard of mathematical requirements for freshmen in STEM studies, and should also give an orientation for school teachers in mathematics.

The OMB+ is text oriented, but includes a large amount of questions, interactive elements, videos and examples with standard solutions, which can be uncovered systematically. As of today, around 50 German institutions use and recommend the OMB+ which is available in German and English and currently being translated into Chinese. Supplementary chapters covering e.g. stochastics, complex numbers and formal logic have been added and further educational videos are in the making. The oHMint project uses the same enriched text oriented style and shares similar features. For more information on OMB+ see [3] and [5].

1.2 The Hamburg Open Online University

The acronym HOOU stands for Hamburg Open Online University (www.hoou.de), a joint initiative of six institutions of higher education in Hamburg: Universität Hamburg (UHH), Hamburg University of Applied Sciences (HAW), Hamburg University of Technology (TUHH), HafenCity University Hamburg (HCU), Hochschule für bildende Künste (HFBK) and the Hochschule für Musik und Theater (HFMT). The “Hamburger Senat” (Senate of Hamburg) founded it in 2015 as the educational branch of “Strategie Digitale Stadt” (Strategy Digital City), an initiative to bundle processes of digitalization within the city.

The HOOU focuses on learners and collaboration for problem solving and develops study material at academic level. Its target groups are not only the custom university students but also interested non-academic persons. Thus, all study material is open as Open Educational Resource and is distributed under a CC license. The production of the first chapter of oHMint is funded by the HOOU and running from November 2017 through December 2018.

2. Modularisation and scope

2.1 Course structure

The OMB+ consortium designed the preliminary structure of the curriculum for the full oHMint course. It spans four semesters and aims to cover all the mathematics typically taught in STEM studies at German institutions of higher education. The curriculum was carefully structured, keeping in mind that different

(types of) universities have different needs for their respective degree programmes. The course will be available in English and German. With oHMint we aim to create a course that consists of units which can be combined in various ways to cover these needs. The fact that a broad spectrum of institutions is represented in the OMB+ consortium justifies our hope that the final course will be widely accepted within the relevant community. This clearly distinguishes oHMint from existing higher mathematics online courses in Germany.

There are three types of units in oHMint: base units cover standard content and are expected to be included in every higher mathematics course, regardless of the institution it is taught at. Supplementary units go deeper into the (theoretical) background of a base unit, and include more abstract views on base unit concepts as well as more involved parts. Finally, optional units exist for subjects that are not necessarily always part of a higher mathematics course, depending in particular on the study programme and/or institution. The idea is that lecturers will be able to choose and combine the units that are relevant for their classes to provide students with everything they need (but not more).

Parallel to the development of the content of oHMint, a data base is being created that keeps track of all the interconnections between mathematical concepts and results in the units and which will help to ensure the desired modularity of the course.

The units themselves are text-based and consist of lectures, exercises, trainings and final tests. They are supplemented by videos where important concepts or ideas are also shown. Each chapter starts with a motivation for the new mathematical concepts taught such that students develop a sense for the importance of the content they are supposed to learn. Moreover, many interactive elements are being implemented in oHMint, some are discussed in more detail below.

2.2 Content of the first-semester course

The learning material from which a lecturer can choose for a first-semester oHMint course consists of the following units:

- Three base units: numbers and functions, differential calculus, integral calculus.
- Four supplementary units: differential calculus, integral calculus, sequences, series.
- Eight optional units: continuity, determination of zeros, L'Hospital's rule, partial fraction decomposition, approximate integration, sequences, series, Fourier analysis.

The scope of the content of each unit has been roughly assessed in terms of the European Credit Transfer and Accumulation System (ECTS) which is commonly used at European institutions of higher education. ECTS points measure the volume of learning based on workload and desired learning outcome. In Germany, one ECTS point is equivalent to 25-30 hours of studying.

The oHMint working group has made an assignment between the content of the chapters of the first-semester course and ECTS points. The base units are assigned

1 ECTS point each, the supplementary units and optional units are assigned 0.5 ECTS points. The workload for the corresponding units is estimated to be within the given range.

These values are used as an orientation for the lecturers who use the material for their courses. The actual assignment of credit points to the course is made by the lecturer or their institution.

3. Didactical concepts of oHMint

3.1 Didactical methods

The oHMint materials are designed to be used for a wide range of blended learning environments up to fully web-based self-learning courses. We selected our didactical methods to specifically suit these environments.

Our guiding principles are a spiral curriculum (after J. Bruner, see [6]), the development of intuitive understanding, awareness, and knowledge of mathematical thinking (see e.g. [7], [8], [9]), and in general an application oriented approach. We follow the ideas of cognitive load theory [10] which focuses on the limited capacity of working memory of students. New ideas and concepts are introduced incrementally, always keeping the cognitive load small and focused. The formally exact representation is the result of the teaching process and not as often a starting point, making the constructive process of mathematical knowledge visible to the students and fostering the awareness between intuitive thinking and correct formulation of mathematics.

Our database of mathematical interconnections (see 2.1) supports the development of the spiral curriculum as examples and exercises are shared between the units and viewed from different perspectives.

The revised version of Bloom's taxonomy [8] (see also [11]) is used for a careful evaluation of the learning goals for each unit and exercises and quizzes are designed to reflect this information. This allows the student to assess their learning achievements and the growth of their knowledge continuously. Automatically corrected problems on different levels of content provide them with a differentiated feedback to support the learning process.

Different types of exercises support the comprehension of ideas, the stabilization of acquired knowledge or the development of routine skills. Application-oriented exercises from science and engineering impart an understanding of the importance of the methods to the relevant application domains.

3.2 Serious games and gamification

Serious games are games that are designed in our context for educational purposes. Gamification can then be defined as the use of typical game design elements and scenarios in non-game contexts, see [12]. We develop a variety of such components for oHMint which create a new incentive for active participation. It is the aim of the current pilot project to test several approaches

and choose the most successful ones for implementation in the entire course which will subsequently be developed. Some of our components are:

Badges: We are developing a system of badges that participants receive for completing certain tasks. The positive effect of badges has been documented, see [13]. This will however be an optional element since not everyone likes to be “distracted” by such matters.

Exercises in game form (competitive and non-competitive): For skills that require routine we are developing a gaming approach. The player is given a task such as “what is the derivative of the function $f(x)$ ”. If answered correctly, the player receives points and proceeds. After finishing the game, players can enter their score into a high score list. We are also investigating the option of enabling players to compete against each other in a duel or small groups and the possibility of a “Who wants to be a millionaire?”-styled game. Developing and realizing these ideas and more general the whole course we benefit from close interaction with the e-learning group at Universität Hamburg as well as with student helpers also from the HafenCity University who provide valuable feedback from the target group of oHMint.

3.3 The advantage of the oHMint concept

Because of the modularisation (see 2.1) lecturers can design their own course by selecting units depending on their programme-level.

Innovative types of exercises are implemented: audio files train the transition between written and spoken mathematics; in “reversed” exercises the students shall find flaws in a given line of arguments; in “drawing” exercises they are asked to sketch the graph of a function with certain properties. Moreover, there is a large amount of quick checks, i.e. short questions or problems interspersed in the course where students can immediately check if they understood some new content or not.

In the available group instructing mode professors can not only compile a course suitable for their needs but also follow the learning progress of their students in a gradebook overview. Also, a flipped classroom version will be available (see 4). And last, a call centre run by integral-learning GmbH will be integrated to oHMint as it is already in the OMB+. Thus, participants of oHMint can contact a call centre for mathematical questions from 10 am till 8 pm via internal chat, telephone, skype (telephone and chat) or internal forum as a free service.

For more details on our didactical methods see [1].

4. Flipped Classroom

At many universities a typical mathematics lecture for engineers suffers from low intrinsic motivation of the students, their low competence in school mathematics, (which we hope to enhance through OMB+) and the procrastinating study style of many students.

Flipped classroom is an interactive teaching model that addresses this problem. The information and content instruction takes place outside of class time and class time is used for practice oriented activities. The concept has been successfully introduced in university mathematics and evaluated in recent years [14, 15, 16]. It has been shown that the average performance of students in exams is at least equal to the one of students instructed with traditional methods. Moreover, through flipped classroom students tend to have a better understanding of underlying concepts. The gap between better and weaker students tends to get smaller while the well performing students perform equally well as or better than with traditional classroom lectures.

The students get preparatory material focusing on the basic concepts with a set of key questions guiding their studies and a quiz of control questions that have to be answered before class time. The quizzes are automatically corrected and give students helpful feedback and encourage them to prepare. In-class time is used for interactivity with more challenging problems that are followed up with homework.

In winter term 2018/19 the first prototype module for differential calculus will be used in a flipped classroom context for a freshmen class of Mathematics in a Geodesy and Geoinformation course at the HCU. The class time will be split. The first half will be an interactive session to handle feedback questions from students and presentations from students. The second half will be used for group work on special exercises to solidify and deepen the gained new mathematical competencies.

Implementing the first base unit of oHMint into a real teaching scenario at this early stage will result in valuable experience and feedback for the further development of the oHMint course.

5. Conclusions

As explained above, the production of the first oHMint base unit *Differential Calculus* is funded by the HOUU. It is meant as a sample for the remaining units of the course, testing new approaches in digital teaching and innovative types of exercises. The oHMint working group, currently a subgroup of the OMB+ consortium, intends to create attention for the project and raise further funding through this sample unit. Other working groups are encouraged to join the oHMint project and/or produce an oHMint unit themselves.

The main assets of oHMint are its flexibility in terms of teaching style, ranging from self-study to flipped classroom settings. Each unit will be translated into

English once it is completed. A preliminary version of *Differential Calculus* will be put to the test in the winter term 2018/19 at HCU Hamburg as a flipped classroom version, feedback from HCU students and lecturers will be incorporated into the future development of oHMint.

Overall, we believe that oHMint poses an important step in the digitalization of STEM education in Germany. The broad supporting base guiding its development ensures that it is tailor-made for the needs of lecturers and students alike, and its modular structure makes it suitable for all institutions.

Acknowledgements. We gratefully acknowledge the financial support of the HOUU for the current project, the guidance and input of the OMB+ consortium (especially Volker Bach) in the development of oHMint and support beyond the technical lead by integral-learning GmbH.

References

- [1] Barbas H., Gasser I., Konieczny F., Lohse A., Seiler R.: oHMint: Online Higher Mathematics for MINT Students - An Online Mathematics Course and Learning Platform, Preprint, to appear in the Proceedings of the 46th SEFI Annual Conference, 2018, Copenhagen, Denmark.
- [2] Barbas H, Gasser I, Konieczny F, Lohse A, Seiler R.; oHMint: Höhere Mathematik für MINT-Studierende in Digitalisierung und Hochschulentwicklung, Getto B, Hintze P, Kerres M (eds.). Münster Waxmann 2018
- [3] OMB+ <https://www.ombplus.de>
- [4] Braun I., Schröder J. et al.: cosh cooperation Schule:Hochschule Mindestanforderungskatalog Mathematik (Version 2.0), 2014.
https://lehrerfortbildung-bw.de/u_matnatech/mathematik/bs/bk/cosh/katalog/index.html
- [5] Barbas H., Schramm T.: The Hamburg Online Math Test MINTFIT for Prospective Students of STEM Degree Programmes, MSOR Connections, 2018, 16(3): 43-51.
- [6] Bruner J.: The Process of Education, Harvard University Press. Cambridge, Mass., USA, 1960.
- [7] Pintrich PR.: The Role of Metacognitive Knowledge in Learning, Teaching and Assessing, Theory into Practice, 2002, 41(4): Revising Bloom's Taxonomy, 219-225.
- [8] Krathwohl DR.: A Revision of Bloom's Taxonomy: An Overview, 2002.
<https://www.depauw.edu/files/resources/krathwohl.pdf>
- [9] Zech F.: Grundkurs Mathematikdidaktik, Beltz Verlag, Weinheim und Basel, 1996.
- [10] de Jong T.: Cognitive load theory, educational research, and instructional design: some food for thought, Instructional Science, 2010, 38: 105-134.
- [11] Anderson, LW., Krathwohl, D. (eds.): A Taxonomy for Learning, Teaching, and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives. New York Longman, 2001.
- [12] Deterding S., Dixon D., Khaled R., Nacke LE.: Gamification: Toward a definition, CHI 2011, 2011, Vancouver, Canada.

- [13] Denny P.: The Effect of Virtual Achievements on Student Engagement, CHI 2013: Changing Perspectives, 2013, Paris, France, pp. 763-772.
- [14] Epstein J.: The Calculus Concept Inventory -- Measurement of the Effect of Teaching Methodology in Mathematics, Notices of the AMS 2013, 60(8), <https://www.ams.org/journals/notices/201308/rnoti-p1018.pdf>
- [15] Maciejewski, W.: Flipping the calculus classroom: an evaluative study, Teaching Mathematics and its Applications: An International Journal of the IMA, Volume 35, Issue 4, 1 December 2016, Pages 187–201. <https://doi.org/10.1093/teamat/hrv019>
- [16] Cronhjort, M., Filipsson, L., Weurlander, M.: Improved engagement and learning in flipped-classroom calculus, Teaching Mathematics and its Applications: An International Journal of the IMA, , hrx007. <https://doi.org/10.1093/teamat/hrx007>

Authors

Helena Barbas

Universität Hamburg, Fachbereich Mathematik
Bundesstraße 55, D-20146 Hamburg

Franz Konieczny

integral-learning GmbH
Clausewitzstraße 2, D-10629 Berlin

Alexander Lohse

Universität Hamburg, Fachbereich Mathematik
Bundesstraße 55, D-20146 Hamburg

Thomas Schramm

HafenCity Universität Hamburg
Überseeallee 16, D-20457 Hamburg
E-Mail: thomas.schramm@hcu-hamburg.de

Anhang

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.
Heft 01/2018	Sergiy Klymchuk: Puzzle-Based Learning in Engineering Mathematics: Students' Attitudes.
Heft 02/2018	Proceedings 1st Northern-Light Symposium on Engineering Education, Hamburg, April 2018.

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