

Oktober 2009

# Computeralgebra

## Rundbrief

GI\_DMV\_GAMM

- ▶ Entdeckung ganzzahliger Relationen
- ▶ Das GIMP-Projekt
- ▶ DFG-Schwerpunkt Computeralgebra
- ▶ ISSAC 2010





## Inhaltsverzeichnis

<b>Inhalt</b> . . . . .	3
<b>Impressum</b> . . . . .	4
<b>Mitteilungen der Sprecher</b> . . . . .	5
<b>Tagungen der Fachgruppe</b> . . . . .	6
<b>Themen und Anwendungen der Computeralgebra</b> . . . . .	8
<i>PSLQ: An Algorithm to Discover Integer Relations</i> (David H. Bailey, Jonathan M. Borwein) . . . . .	8
$2^{37.156.667} - 1$ ist eine Primzahl! (Hans-Michael Elvenich) . . . . .	12
<i>Komplexe Multiplikation: von numerisch bis symbolisch</i> (Andreas Enge) . . . . .	13
<b>Neues über Systeme</b> . . . . .	18
<i>Das SCIENCE EU Programm: Symbolic Computation Infrastructure for Europe</i> (Peter Horn, Dan Roozemon) . . . . .	18
<i>Neues aus Waterloo: Maple 13 und MapleSim 2+3</i> (Thomas Richard) . . . . .	22
<b>Computeralgebra in der Schule</b> . . . . .	24
<i>CAS-Einsatz aus Sicht der Schule</i> (Jan Hendrik Müller) . . . . .	24
<i>Funktionales Denken und Analysispropädeutik – Ein Beitrag zu einem qualitativen Einstieg in die Schul-         analysis durch Computereinsatz</i> (Andrea Hoffkamp) . . . . .	27
<b>Publikationen über Computeralgebra</b> . . . . .	31
<b>Besprechungen zu Büchern der Computeralgebra</b> . . . . .	32
<i>Ganzha, Mayr, Vorozhtsov: Computer Algebra in Scientific Computing (CASC 2007)</i> (Werner Seiler) . . . . .	32
<i>Joswig, Theobald: Algorithmische Geometrie</i> (Werner Seiler) . . . . .	32
<i>Kügler, Windsteiger: Algorithmische Methoden. Zahlen, Vektoren, Polynome</i> (Hartmut Führ) . . . . .	33
<i>Westermann: Mathematik für Ingenieure</i> (Marcel Ern�) . . . . .	34
<b>Berichte von Konferenzen</b> . . . . .	35
<b>Hinweise auf Konferenzen</b> . . . . .	41
<b>Mitteilungen</b> . . . . .	44
<i>DFG-Schwerpunktprogramm in der Computeralgebra eingerichtet</i> (Wolfram Decker) . . . . .	44
<b>Preisverleihungen</b> . . . . .	45
<i>Verleihung des F. L.-Bauer-Preises an Stephen Wolfram</i> (Thomas Hahn) . . . . .	45
<b>Kurze Mitteilungen</b> . . . . .	45
<b>Fachgruppenleitung Computeralgebra 2008-2011</b> . . . . .	47

## Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI, DMV und GAMM (verantwortlicher Redakteur: Dr. Markus Wessler, [markus.wessler@hm.edu](mailto:markus.wessler@hm.edu)).

Der Computeralgebra-Rundbrief erscheint halbjährlich, Redaktionsschluss 28.02. und 30.09. ISSN 0933-5994. Mitglieder der Fachgruppe Computeralgebra erhalten je ein Exemplar dieses Rundbriefs im Rahmen ihrer Mitgliedschaft. Fachgruppe Computeralgebra im Internet: <http://www.fachgruppe-computeralgebra.de>.

Konferenzankündigungen, Mitteilungen, einzurichtende Links, Manuskripte und Anzeigenwünsche bitte an den verantwortlichen Redakteur.

Die Geschäftsstellen der drei Trägergesellschaften:

**GI** (Gesellschaft für  
Informatik e.V.)  
Wissenschaftszentrum  
Ahrstr. 45  
53175 Bonn  
Telefon 0228-302-145  
Telefax 0228-302-167  
[gs@gi-ev.de](mailto:gs@gi-ev.de)  
<http://www.gi-ev.de>



**DMV** (Deutsche Mathematiker-  
Vereinigung e.V.)  
Mohrenstraße 39  
10117 Berlin  
Telefon 030-20377-306  
Telefax 030-20377-307  
[dmv@wias-berlin.de](mailto:dmv@wias-berlin.de)  
<http://www.dmv.mathematik.de>



**GAMM** (Gesellschaft für Angewandte  
Mathematik und Mechanik e.V.)  
Technische Universität Dresden  
Institut für Statik und Dynamik der  
Tragwerke  
01062 Dresden  
Telefon 0351-463-34386  
Telefax 0351-463-37061  
[GAMM@mailbox.tu-dresden.de](mailto:GAMM@mailbox.tu-dresden.de)  
<http://www.gamm-ev.de>



---

## Mitteilungen der Sprecher

---

Liebe Mitglieder der Fachgruppe Computeralgebra,

am 3. Oktober 2009 fand in Konstanz die vierte Sitzung der Fachgruppenleitung statt. Im Mittelpunkt standen diesmal die vielfältigen Tagungsaktivitäten der Fachgruppe.

Unsere diesjährige Computeralgebratagung, die vom 14. bis zum 16. Mai in Kassel stattfand, war mit 60 Teilnehmern wieder sehr gut besucht.

Der mit 500 € dotierte Nachwuchspreis für den besten Vortrag eines Nachwuchswissenschaftlers ging an Daniel Andres aus Aachen. Die Auswahl fiel diesmal schwer, denn es gab eine ganze Reihe exzellenter Nachwuchsvorträge. Daniel machte das Rennen auch deshalb, weil er noch an seiner Diplomarbeit sitzt.

Daniel Andres brachte außerdem das Kunststück fertig, auf der ISSAC-Tagung 2009 einen weiteren Nachwuchspreis zu erhalten, was unsere Wahl in eindrucksvoller Weise bestätigt. Einen längeren Bericht über die ISSAC-Tagung 2009 finden Sie auf Seite 37.



Überreichung des Nachwuchspreises in Kassel



Auf dieser internationalen Computeralgebra-Tagung, die vom 28. bis zum 31. Juli in Seoul stattfand, wurde in der üblichen Tagungstradition im Rahmen des ISSAC Business Meetings von den Teilnehmern der Tagungsort für die ISSAC 2011 bestimmt. Bewerber waren Boston und San Jose, in einer knappen Entscheidung kam diesmal San Jose zum Zug. Die ISSAC wird dann zum ersten Mal im Rahmen der großen FCRC 2011 der ACM (Federated Computing Research Conference, [www.acm.org/fcrc](http://www.acm.org/fcrc), 4.–11. Juni 2011) stattfinden.

Ferner gibt es einige Änderungen im ISSAC Steering Committee ([www.sigsam.org/issac/steering-committee.phtml](http://www.sigsam.org/issac/steering-committee.phtml)). Die Fachgruppe Computeralgebra wurde für weitere 3 Jahre als Organizational Member des Steering Committees bestätigt und wird hierbei weiterhin von Elkedagmar Heinrich vertreten. Für die SIGSAM ersetzt deren Vice Chair Elizabeth Mansfield das bisherige Mitglied Daniel Lichtblau. Schließlich wurde Franz Winkler als neues Mitglied (Members-at-Large) des Steering Committees von den Tagungsteilnehmern gewählt.

Schließlich stellte der Sprecher der Fachgruppe Wolfram Koepf als General Chair der ISSAC 2010 auf der Sitzung in Seoul die Pläne für die ISSAC-Tagung 2010 vor, die vom 25.–28. Juli 2010 in München stattfinden wird. Local Arrangements Chair ist Ernst W. Mayr von der TU München und Program Committee Chair ist Stephen M. Watt von der University of Western Ontario in Kanada. Es gibt inzwischen eine Website (<http://www.issac-conference.org/2010/>), auf welcher Sie auch den Call for Papers finden.

Die Fachgruppe als Veranstalter dieser hochrangigen internationalen Tagung würde sich natürlich sehr freuen, wenn diesmal besonders viele deutsche Wissenschaftler das wissenschaftliche Programm bereichern würden und fordert daher zu einer regen Beteiligung auf. Immerhin sind wir mit über 400 Mitgliedern die größte derartige Fachorganisation weltweit. Einsendeschluss der Tagungsbeiträge ist der 14. Januar 2010. Wer zunächst nur ein Abstract einreicht, hat bis zum 21. Januar 2010 Zeit, seine Arbeit nachzureichen. Weiteres siehe Seite 43.



Wolfram Koepf und Stephen Watt in Seoul



*Eine weitere sehr erfreuliche Nachricht gab es im Mai: Die Deutsche Forschungsgemeinschaft (DFG) hat ein (bundesweites) Schwerpunktprogramm zu Themen der Computeralgebra eingerichtet, siehe Seite 44. Koordinator des Programms ist Wolfram Koepf. Das SPP 1489 Algorithmische Methoden in Algebra, Geometrie und Zahlentheorie wird im Frühjahr 2010 beginnen, die Ausschreibung läuft momentan. Auf dem Vorbereitungstreffen des Schwerpunkts, das im August in Mainz stattfand, wurde besprochen, dass nach der ISSAC-Tagung Ende Juli 2010 in Garching bei München ein Satelliten-Workshop des Schwerpunktes stattfinden wird.*

*Die nächste Sitzung der Fachgruppenleitung findet Mitte Februar 2010 in Garching/München statt. Wir hoffen, Sie mit dem vorliegenden Heft wieder gut zu informieren.*

Wolfram Koepf

Elkedagmar Heinrich

---

## Tagungen der Fachgruppe

---



Tagungsfoto Kassel 2009

### Computeralgebra, 14. – 16.05.2009, Kassel

Bereits zum vierten Mal organisierte die Fachgruppe Computeralgebra vom 14. bis zum 16. Mai 2009 eine Tagung zum gleichnamigen Thema. Sie fand an der Universität Kassel statt und war mit 60 Teilnehmern sehr gut besucht. Neben den fünf Hauptvorträgen boten 17 weitere Vorträge auch Nachwuchswissenschaftlern und Computeralgebra-Neulingen eine gute Gelegenheit, ihre Arbeiten zu präsentieren.

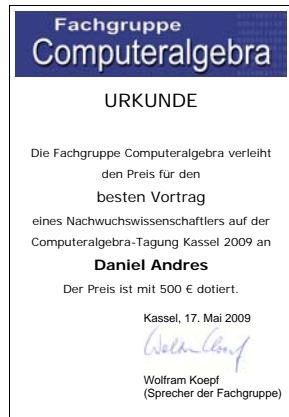
Die Hauptvorträge von Claus Diem (über Komplexitätstheorie), Viktor Levandovskyy (über D-Moduln), Thomas Markwig (über tropische Geometrie), Almar Kaid (über semistabile Vektorbündel) und Thomas Sturm (über Quantorenelimination) deckten ein breites

Spektrum an Anwendungsgebieten der Computeralgebra ab. Für den besten Vortrag eines Nachwuchswissenschaftlers gab es wiederum einen mit 500 € dotierten Nachwuchspreis. Die Auswahl fiel der Jury dabei besonders schwer, denn die Beiträge waren in diesen Jahr von so ausgeglichen hoher Qualität wie noch nie. Am Ende gewann Daniel Andres aus Aachen mit seinem Vortrag über *Algorithmen zur Berechnung von b-Funktionen*. Den Ausschlag gab dabei die Tatsache, dass Herr Andres zum Zeitpunkt der Entstehung seiner Arbeit noch Diplomand war.

Die Konferenz war ausgezeichnet organisiert. Trotz des minimalen Unkostenbeitrags von 5 € pro Teilnehmer gelang es dem Hauptorganisator Wolfram Koepf, die Veranstaltung kostenneutral durchzuführen. Dies

lag vor allem an der großen Unterstützung durch die Firmen Additive, Casio, Scientific Computers, Springer Verlag und Texas Instruments, die teilweise auch durch Ausstellungsstände und Vorträge zum Gelingen der Konferenz beitrugen.

Dank gebührt auch dem Fachbereich Mathematik der Universität Kassel, der neben den Räumlichkeiten auch die Namensschilder für die Teilnehmer und den Druck des Tagungsbands offerierte.



*Nachwuchspreisträger Daniel Andres*

Die Reihe der Computeralgebratagungen der Fachgruppe hat sich mittlerweile zu einer festen Größe in der nationalen und internationalen Computeralgebragemeinschaft entwickelt. So ist es nur zu wünschen, dass sie 2011 mit ähnlichem Erfolg und einer ebenso großen Resonanz wie bei der Konferenz in Kassel fortgeführt wird.

Martin Kreuzer (Universität Passau)



**ISSAC 2010, 25. – 28.07.2010, München**

<http://www.issac-conference.org/2010>

Im Juli 2010 wird die Fachgruppe die internationale Tagung *ISSAC 2010* in München organisieren. Ernst W. Mayr von der TU München ist Local Arrangements Chair und Wolfram Koepf ist General Chair der Tagung. Program Committee Chair ist Stephen M. Watt von der University of Western Ontario in Kanada. Die weiteren Mitglieder des Organisationskomitees

findet man auf der Tagungshomepage <http://www.issac-conference.org/2010>.

Die Fachgruppe als Veranstalter dieser hochrangigen internationalen Tagung würde sich natürlich sehr freuen, wenn diesmal besonders viele deutsche Wissenschaftler das wissenschaftliche Programm bereichern würden und fordert daher zu einer regen Beteiligung auf. Immerhin sind wir mit über 400 Mitgliedern die größte derartige Fachorganisation weltweit. Es wäre doch ein gutes Zeichen, wenn sich dies auch an der Beteiligung bei der *ISSAC 2010* festmachen würde. Einsendeschluss der Tagungsbeiträge ist der 14. Januar 2010. Wer zunächst nur ein Abstract einreicht, hat bis zum 21. Januar 2010 Zeit, seine Arbeit nachzureichen. Die Artikel werden wie bei einer sehr guten Zeitschrift begutachtet. Alle Details des *Call for Papers* finden Sie auf Seite 43. Falls Sie keinen Artikel einreichen wollen, wäre es schön, wenn wir Sie dann als Teilnehmer der Tagung begrüßen dürften.

Wir möchten noch darauf hinweisen, dass die Fachgruppenleitung beschlossen hat, auf der Tagung (zusätzlich zu den Preisen, die regelmäßig vergeben werden), an ein Fachgruppenmitglied den *Fachgruppe Award for Best Contribution ISSAC 2010* zu vergeben.

Nach der *ISSAC*-Tagung findet Ende Juli 2010 in Garching bei München ein Satelliten-Workshop des DFG-Schwerpunktprogramms 1489 *Algorithmische Methoden in Algebra, Geometrie und Zahlentheorie* statt.



*ISSAC-Tagung in München 2010*

Wolfram Koepf (Universität Kassel)



## PSLQ: An Algorithm to Discover Integer Relations

**David H. Bailey**  
Lawrence Berkeley National Laboratory, Kalifornien

**Jonathan M. Borwein**  
University of Newcastle, Callahan, New South Wales

dhbailey@lbl.gov  
jonathan.borwein@newcastle.edu.au




---

### Introduction

---

Let  $x = (x_1, x_2, \dots, x_n)$  be a vector of real or complex numbers.  $x$  is said to possess an integer relation if there exist integers  $a_i$ , not all zero, such that

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = 0.$$

By an *integer relation algorithm*, we mean a practical computational scheme that can recover the vector of integers  $a_i$ , if it exists, or can produce bounds within which no integer relation exists. As we will see in the examples below, an integer relation algorithm can be used to recognize a computed constant in terms of a formula involving known constants, or to discover an underlying relation between quantities that can be computed to high precision.

At the present time, the most widely used algorithm for integer relation detection is the “PSLQ” algorithm of mathematician-sculptor Helaman Ferguson [11, 4], although the “LLL” algorithm is also used for this purpose. One detailed comparison of these two methods found that PSLQ appears to be more numerically stable than LLL, in the sense that PSLQ reliably finds a relation, beginning nearly at the minimal precision level for the relation, whereas LLL sometimes finds a relation at one level but fails at a somewhat higher level [10]. This study also found that tuned implementations of PSLQ (which select multiple pairs of indices, and which employ two or three levels of precision [4]) are significantly more efficient than typical implementations of LLL. Additional research may further cast light on the relative merits of these two schemes. In the following, though, we will focus on PSLQ.

PSLQ operates by constructing a sequence of integer-valued matrices  $B_n$  that reduces the vector  $y = xB_n$ , until either the relation is found (as one of the columns of  $B_n$ ), or else precision is exhausted. At the same time, PSLQ generates a steadily growing bound on the

size of any possible relation. When a relation is found, the size of smallest entry of the vector  $y$  abruptly drops to roughly “epsilon” (i.e.  $10^{-p}$ , where  $p$  is the number of digits of precision). The size of this drop can be viewed as a “confidence level” that the relation is real and not merely a numerical artifact. A drop of 20 or more orders of magnitude almost always indicates a real relation (see Figure 1).

Very high precision arithmetic must be used with PSLQ or any other integer relation scheme. If one wishes to recover a relation of length  $n$  with coefficients of maximum size  $d$  digits, then the input vector  $x$  must be specified to at least  $nd$  digits, and one must employ  $nd$ -digit floating-point arithmetic. *Maple* and *Mathematica* include multiple precision arithmetic facilities and *Maple* ships with a full implementation of PSLQ. One may also use any of the several freeware multiprecision software packages, for example the ARPREC package by the first author and colleagues at LBNL [7]. In the remaining sections we describe various representative applications of PSLQ. More detail about these examples is given in [8] and the references therein.

---

### Finding Algebraic Relations Using PSLQ

---

One immediate and impressive application of PSLQ in the field of mathematical number theory is to determine whether or not a given constant  $\alpha$ , whose value can be computed to high precision, is algebraic of some degree  $n$  or less. This can be done by first computing the vector  $x = (1, \alpha, \alpha^2, \dots, \alpha^n)$  to high precision and then applying an integer relation algorithm to the resulting  $(n + 1)$ -long vector. If a relation is found for  $x$ , then this relation vector is precisely the set of integer coefficients of a polynomial satisfied by  $\alpha$  (to the precision specified).

One of the first results of this sort was the identification of the constant  $\hat{B}_3 = 3.54409035955 \dots$



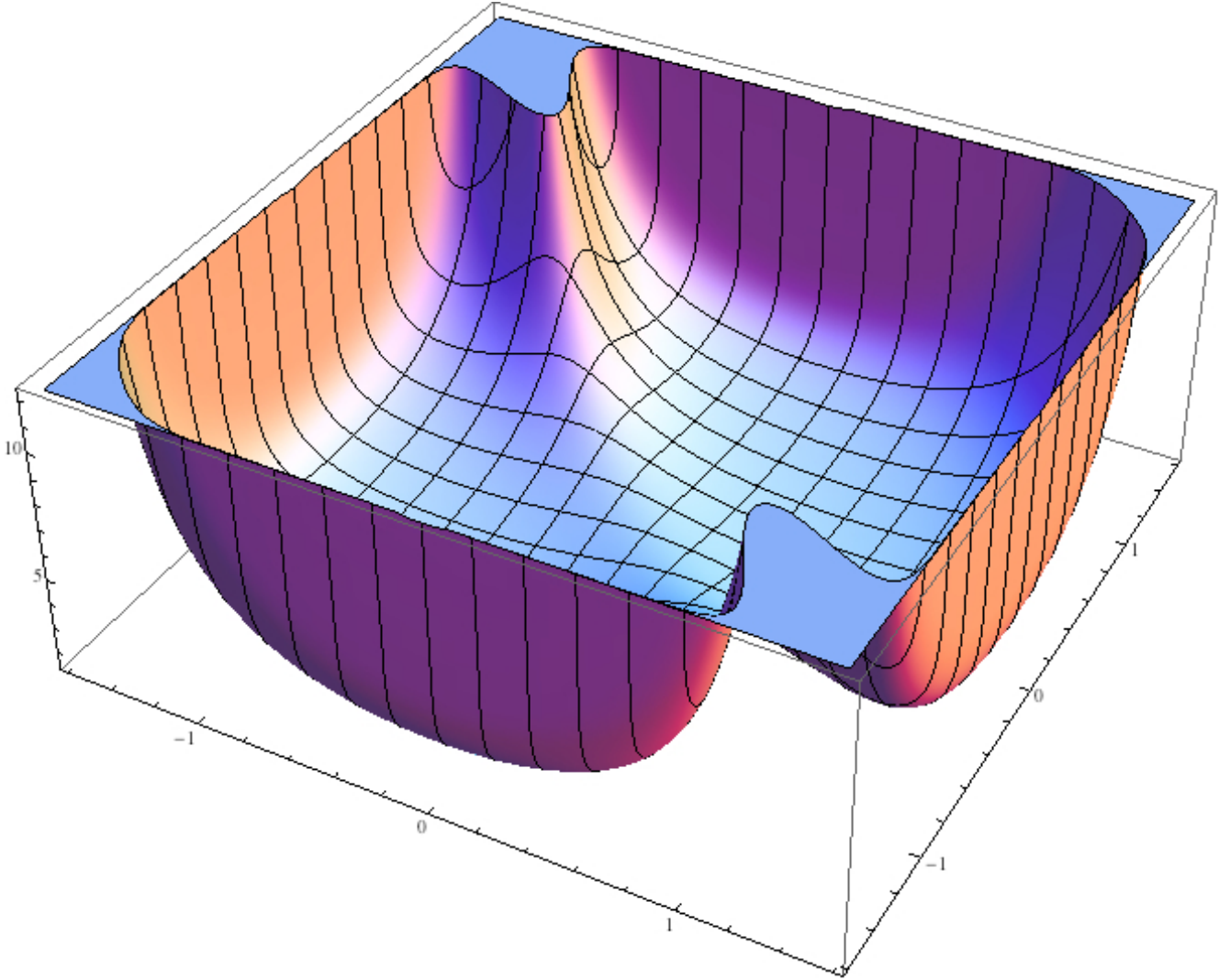


Fig. 1:  $\log_{10} \max_k |y_k|$  versus iteration number in a typical PSLQ run

$\hat{B}_3$  is the third bifurcation point of the logistic map  $x_{k+1} = rx_k(1 - x_k)$ , which exhibits period doubling shortly before the onset of chaos. To be precise,  $\hat{B}_3$  is the smallest value of the parameter  $r$  such that successive iterates  $x_k$  exhibit eight-way periodicity instead of four-way periodicity.  $\hat{B}_3$  can be computed to arbitrarily high precision by means of an iterative algorithm [6]. When PSLQ is applied to the 13-long vector  $(1, \hat{B}_3, \hat{B}_3^2, \hat{B}_3^3, \dots, \hat{B}_3^{12})$ , one obtains the result that  $\hat{B}_3$  is a root of the polynomial

$$0 = 4913 + 2108t^2 - 604t^3 - 977t^4 + 8t^5 + 44t^6 + 392t^7 - 193t^8 - 40t^9 + 48t^{10} - 12t^{11} + t^{12}.$$

Recently,  $\hat{B}_4 = 3.564407268705\dots$ , the fourth bifurcation point of the logistic map, was identified using PSLQ by the British physicist David Broadhurst [4]. Some conjectural reasoning had suggested that  $\hat{B}_4$  might satisfy a 240-degree polynomial, and some further analysis had suggested that the constant  $\alpha = -\hat{B}_4(\hat{B}_4 - 2)$  might satisfy a 120-degree polynomial. In order to test this hypothesis, Broadhurst applied a PSLQ program to the 121-long vector  $(1, \alpha, \alpha^2, \dots, \alpha^{120})$ . Indeed, a rela-

tion was found, though 10,000-digit arithmetic was required. The recovered integer coefficients descend monotonically from  $257^{30} \approx 1.986 \times 10^{72}$  to 1. This was subsequently proven using Groebner bases [6].

## A New Formula for $\pi$

Through the centuries mathematicians have assumed that there is no shortcut to computing digits of  $\pi$  beginning at some position  $n$ . Thus, it came as no small surprise when such an algorithm was discovered in 1996 [3]. In particular, this simple scheme allows one to compute binary or hexadecimal (base-16) digits of  $\pi$  starting at an arbitrary position, without computing any of the preceding digits. For instance, the one millionth hex digit of  $\pi$  can be computed in this manner on a current-generation personal computer in only about 10 seconds run time. This scheme is based on the following new formula, which was discovered in 1996 using PSLQ:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

Since 1996, numerous formulas of this same type have been found for various constants [1, 8]. For example, a similar formula was found that permits arbitrary-

position binary digits of  $\pi^2$  to be calculated; here is a formula for  $\pi^2$  that permits arbitrary *ternary* (base-3) digits of  $\pi^2$  to be calculated:

$$\pi^2 = \frac{2}{27} \sum_{k=0}^{\infty} \frac{1}{729^k} \left( \frac{243}{(12k+1)^2} - \frac{405}{(12k+2)^2} - \frac{81}{(12k+4)^2} - \frac{27}{(12k+5)^2} - \frac{72}{(12k+6)^2} - \frac{9}{(12k+7)^2} - \frac{9}{(12k+8)^2} - \frac{5}{(12k+10)^2} + \frac{1}{(12k+11)^2} \right).$$

Sadly, it has recently been proven that there is no formula of this type for  $\pi$  itself in other than a binary base [8, Ch. 3].

proofs were found for many of these specific and general results. Three examples of PSLQ results that were subsequently proven are given in Table 1. In the table,

## Identification of Multiple $\zeta$ Constants

A large number of results has been found over the last 15 years using PSLQ in the course of research on *multiple zeta sums*, such as those shown in Table 1. After computing the numerical values of these constants, a PSLQ program was used to determine if a given constant satisfied an identity of a conjectured form. These efforts produced numerous empirical evaluations and suggested general results [2, 9]. Eventually, elegant

$$\zeta(t) = \sum_{j=1}^{\infty} j^{-t}$$

is the Riemann zeta function, and

$$\text{Li}_n(x) = \sum_{j=1}^{\infty} x^j j^{-n}$$

denotes the polylogarithm function.

$$\begin{aligned} \sum_{k=1}^{\infty} \left(1 + \frac{1}{2} + \cdots + \frac{1}{k}\right)^2 (k+1)^{-4} &= \frac{37}{22680} \pi^6 - \zeta^2(3) \\ \sum_{k=1}^{\infty} \left(1 + \frac{1}{2} + \cdots + \frac{1}{k}\right)^3 (k+1)^{-6} &= \zeta^3(3) + \frac{197}{24} \zeta(9) + \frac{1}{2} \pi^2 \zeta(7) \\ &\quad - \frac{11}{120} \pi^4 \zeta(5) - \frac{37}{7560} \pi^6 \zeta(3) \\ \sum_{k=1}^{\infty} \left(1 - \frac{1}{2} + \cdots + (-1)^{k+1} \frac{1}{k}\right)^2 (k+1)^{-3} &= 4 \text{Li}_5\left(\frac{1}{2}\right) - \frac{1}{30} \ln^5(2) - \frac{17}{32} \zeta(5) \\ &\quad - \frac{11}{720} \pi^4 \ln(2) + \frac{7}{4} \zeta(3) \ln^2(2) + \frac{1}{18} \pi^2 \ln^3(2) - \frac{1}{8} \pi^2 \zeta(3) \end{aligned}$$

Table 1: Some multiple  $\zeta$  identities found by PSLQ

## Ising integrals

One particularly fruitful application of these methods is the evaluation of definite integrals, such as those that arise in mathematical physics. For example, recently the present authors, together with Richard Crandall, investigated three classes of  $n$ -fold integrals, which arise in Ising theory and also (in some cases) in quantum field theory:

$$C_n := \frac{4}{n!} \int_0^{\infty} \cdots \int_0^{\infty} \frac{1}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2} \frac{du_1}{u_1} \cdots \frac{du_n}{u_n},$$

$$D_n := \frac{4}{n!} \int_0^{\infty} \cdots \int_0^{\infty} \frac{\prod_{i < j} \left(\frac{u_i - u_j}{u_i + u_j}\right)^2}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2} \frac{du_1}{u_1} \cdots \frac{du_n}{u_n},$$

$$E_n = 2 \int_0^1 \cdots \int_0^1 \left(\prod_{1 \leq j < k \leq n} \frac{u_k - u_j}{u_k + u_j}\right)^2 dt_2 dt_3 \cdots dt_n,$$

where in the last line  $u_k = t_1 t_2 \cdots t_k$ .

Computing high-precision values of  $n$ -fold integrals such as this is very difficult for  $n$  greater than three or four. But we found a simple substitution that reduces the  $C$  integrals to 1-dimensional integrals:

$$C_n = \frac{2^n}{n!} \int_0^{\infty} t K_0^n(p) dt,$$

where  $K(t)$  is the modified Bessel function. In this form, we were able to evaluate  $C_n$  to over 1000-digit accuracy, for  $n$  up to 1024. With these numerical values in hand, we quickly found that  $C_1 = 2$ ,  $C_2 = 1$ ,  $C_3 = L_{-3}(2) = \sum_{n \geq 0} (1/(3n+1)^2 - 1/(3n+2)^2)$ , and  $C_4 = 7\zeta(3)/12$ . We also discovered numerically that

$$\lim_{n \rightarrow \infty} C_n = 2e^{-2\gamma},$$

where  $\gamma$  is Euler's constant. Further computation established results such as:

$$\begin{aligned} D_2 &= \frac{1}{3}, \\ D_3 &= 8 + \frac{4\pi^2}{3} - 27L_{-3}(2), \\ D_4 &= \frac{4\pi^2}{9} - \frac{1}{6} - \frac{7}{2}\zeta(3) \end{aligned}$$

and

$$\begin{aligned} E_2 &= 6 - 8 \log 2, \\ E_3 &= 10 - 2\pi^2 - 8 \log 2 + 32 \log^2 2, \\ E_4 &= 22 - 82\zeta(3) - 24 \log 2 + 176 \log^2 2 \\ &\quad - 256(\log^3 2)/3 + 16\pi^2 \log 2 - 22\pi^2/3, \\ E_5 &\stackrel{?}{=} 42 - 1984 \text{Li}_4(1/2) + 189\pi^4/10 \\ &\quad - 74\zeta(3) - 1272\zeta(3) \log 2 \\ &\quad + 40\pi^2 \log^2 2 - 62\pi^2/3 + 40(\pi^2 \log 2)/3 \\ &\quad + 88 \log^4 2 + 464 \log^2 2 - 40 \log 2. \end{aligned}$$

The  $E_5$  integral was found after transforming its defining 5-fold integral representation into an extremely complicated 3-fold integral. We then computed this 3-fold integral to 250-digit precision, by using a parallel quadrature program implemented on 1024 CPUs of a parallel computer system, and then discovered the above-listed experimental identity by using PSLQ. This identity has a question mark because, unlike the others mentioned in this paper, we do not yet have a formal proof. Nonetheless it is established numerically at least 180 orders of magnitude beyond the level of numerical "chance," and so we are quite confident in the result. Such confidence is typically obtainable if the constants involved can be computed to sufficiently high precision. Sometimes as with  $C_n$  this is relatively easy. In other cases, such as  $E_5$ , it involves much more labor.

---

## Research questions

---

In spite of these and other successes, there is considerable need for even more efficient schemes for both integer relation detection and numerical integration, especially the evaluation of multi-dimensional integrals. With regards to PSLQ, there is interest in extending PSLQ to

more general number fields, such as quadratic number fields. Hopefully future research will yield better schemes that will in turn produce more results of interest in mathematics and mathematical physics.

## Literatur

- [1] David H. Bailey, *A Compendium of BBP-Type Formulas*, available at <http://crd.lbl.gov/~dhbailey/dhbpapers/bbp-formulas.pdf>.
- [2] David H. Bailey, Jonathan M. Borwein and Roland Girgensohn, *Experimental Evaluation of Euler Sums*, Experimental Mathematics, Vol. 4, No. 1, 1994, 17-30.
- [3] David H. Bailey, Peter B. Borwein and Simon Plouffe, *On The Rapid Computation of Various Polylogarithmic Constants*, Math. of Computation, Vol. 66, No. 218, 1997, 903-913.
- [4] David H. Bailey and David J. Broadhurst, *Parallel Integer Relation Detection: Techniques and Applications*, Math. of Computation, Vol. 70, No. 236, 1719-1736.
- [5] David H. Bailey, Jonathan M. Borwein, R. E. Crandall, *Integrals of the Ising class*, Journal of Physics A: Mathematical and General, Vol. 39 (2006), 12271-12302.
- [6] David H. Bailey, Jonathan M. Borwein, Vishal Kapoor, Eric Weisstein, *Ten Problems in Experimental Mathematics*, American Math. Monthly, Vol. 113, No. 6, 2006, 481-409.
- [7] David H. Bailey, Yozo Hida, Xiaoye S. Li, Brandon Thompson, *ARPREC: An Arbitrary Precision Computation Package*, 2002, available at <http://crd.lbl.gov/~dhbailey/dhbpapers/arprec.pdf>.
- [8] Jonathan M. Borwein, David H. Bailey, *Mathematics by Experiment*, AK Peters, 2003. Second edition, 2008, see also <http://www.experimentalmath.info>.
- [9] David Borwein, Jonathan M. Borwein, Roland Girgensohn, *Explicit Evaluation of Euler Sums*, Proc. Edinburgh Math. Society, Vol. 38, 1995, 77-294.
- [10] Jonathan M. Borwein, Peter Lisoněk, *Applications of Integer Relation Algorithms*, Discrete Mathematics (special issue for FPSAC 1997), Vol. 217 (2000), 65-82.
- [11] Helaman R. P. Ferguson, David H. Bailey, Stephen Arno, *Analysis of PSLQ, An Integer Relation Finding Algorithm*, Math. of Computation, Vol. 68, 1999, 351-369.

## $2^{37.156.667} - 1$ ist eine Primzahl!

**Hans-Michael Elvenich**  
(Lanxess Leverkusen)

michael@elvenich.de



Am 6. September 2008 um 21.54 Uhr MESZ entdeckte mein Computer eine der größten bisher bekannten Mersenne<sup>1</sup>-Primzahlen. Diese Zahl hat 11.185.272 Ziffern und damit mehr als die von der US-Stiftung Electronic Frontier Foundation<sup>2</sup>, kurz EFF, geforderten 10 Millionen Stellen. Diese Stiftung hatte darauf ein Preisgeld von 100.000 Dollar ausgesetzt. Allerdings hatte die Universität von Kalifornien in Los Angeles zwei Wochen vorher eine noch größere Zahl entdeckt und somit den fast zehnjährigen Wettlauf quasi mit einem „Fotofinish“ für sich entschieden.

*Mersenne-Zahlen* sind Zahlen der Form  $M_p = 2^p - 1$ , wobei der Exponent  $p$  selbst eine Primzahl sein muss, ansonsten handelt es sich um eine zusammengesetzte Zahl. Ist diese Zahl  $M_p$  prim, so spricht man von einer *Mersenne-Primzahl*.

Nachdem bisher 42 Kandidaten von meinem Rechner nach längeren Prüfungen als nicht prim identifiziert worden waren, wurde mir am 2. Februar 2008 der 43. Prüfling von GIMPS<sup>3</sup> (Great Internet Mersenne Prime Search) zugeteilt. Eigentlich hätte mein neuester Rechner, ein PC mit einer Intel® Core™ 2 Duo CPU E8300 @ 2,83 GHz diese Zahl im 24/7-Betrieb (24 Stunden am Tag und 7 Tage in der Woche) in nur 42 Tagen überprüfen können, allerdings hatte ich mich ab Januar 2008 dazu entschlossen, meinen Rechner nicht mehr wie die bisherigen vier Jahre dauernd laufen zu lassen, sondern nur noch vier bis fünf Stunden pro Tag. Die immer weiter steigenden Energiekosten hatten mich dazu veranlasst. Mein Rechner hatte einen jährlichen Verbrauch von fast 3.000 kWh. Durch diese Sparmaßnahme wurde aus dem eigentlichem Endberechnungsdatum (15. März 2008) der 6. September 2008.

Man braucht kein Mathematiker zu sein, um sich bei dieser Suche zu beteiligen. Man lädt eine Software<sup>4</sup> aus dem Internet, welche die Leerlaufzeiten des PCs ausnutzt, um zugeteilte Primzahlkandidaten zu testen. 1996 gründete der Programmierer George Woltman<sup>5</sup>

das GIMPS-Projekt. Hier werden PCs durch das Internet zusammen geschaltet, um eine größere gemeinsame Rechenkraft zu erhalten. Bis heute haben sich fast 26.000 Nutzer mit ca. 140.000 PCs registriert. Die aktiven PCs (16 %) haben heute eine durchschnittliche gemeinsame Rechenleistung von 40 TFLOP (Trillions of calculations) pro Sekunde, also 40 Billionen Gleitkommaoperationen pro Sekunde, entsprechend 40 TeraFLOPS. Zum Vergleich: Im März 2006 wurde der damals neueste „schnellste“ Computer Deutschlands in Jülich in Betrieb genommen, der JUBL (Jülicher Blue Gene/L). Mit 45,6 TeraFLOPS bot er zu diesem Zeitpunkt als sechstschnellster Computer der Welt die Rechenleistung von 15.000 „normalen“ zeitgemäßen PCs<sup>6</sup>. Als Nachfolger wurde in Jülich der JUGENE (Jülicher BlueGene/P) aufgebaut. Dieser hat seit einem Upgrade letztes Jahr 825,5 TeraFLOPS und belegt nun Platz 3.<sup>7</sup>

Diese auf Ihrem heimischen Rechner installierte Software testet zugewiesene Testkandidaten der Form  $M_p = 2^p - 1$ . Diese Form hat den Vorteil, dass man sie relativ einfach durch den Lucas-Lehmer-Test auf die Primeigenschaft testen kann. Die GIMPS-Software prüft zunächst, ob für den Mersenne-Primzahlenkandidaten ein kleiner Faktor (z. B. bis 275) gefunden werden kann. Anschließend folgt die P-1-Factoring-Stufe. Mit diesen beiden Vortests möchte man relativ kleine Primfaktoren ausschließen, bevor man den doch etwas länger dauernden Lucas-Lehmer-Test startet. Dieser Test verwendet nur Grundrechenarten. Für die Multiplikationen verwendet die Software die schnelle „Irrational Base Discrete Weighted Transform“ (IBDWT), entwickelt von Richard Crandall.<sup>8</sup> Der Lucas-Lehmer-Test liefert als Ergebnis nur eine Ja/Nein-Aussage, aber keine Faktoren. Im Zahlenbereich um  $2^{37.156.667} - 1$  lag die Chance, dass der Kandidat eine Primzahl ist, bei ca. 1 : 320.000. Innerhalb von 13 Jahren wurden durch das GIMPS-Projekt 13 neue Mersenne-Primzahlen entdeckt:

<sup>1</sup>Marin Mersenne, französischer Mathematiker uvm., 1588-1648 ([http://de.wikipedia.org/wiki/Marin\\_Mersenne](http://de.wikipedia.org/wiki/Marin_Mersenne))

<sup>2</sup>EFF Cooperative Computing Awards (<http://www.eff.org/awards/coop>)

<sup>3</sup>Great Internet Mersenne Prime Search (<http://www.mersenne.org/>)

<sup>4</sup>GIMPS Software (<http://www.mersenne.org/freesoft/>)

<sup>5</sup>George Woltman ([http://en.wikipedia.org/wiki/George\\_Woltman](http://en.wikipedia.org/wiki/George_Woltman))

<sup>6</sup>FLOPS (<http://de.wikipedia.org/wiki/FLOPS>)

<sup>7</sup>Liste der 500 schnellsten Computersysteme, Wikipedia (<http://de.wikipedia.org/wiki/TOP500>)

<sup>8</sup>Richard Crandall, US-Informatiker ([http://de.wikipedia.org/wiki/Richard\\_Crandall](http://de.wikipedia.org/wiki/Richard_Crandall))



Jahr	Zahl	Ziffern	Entdecker
1996	$2^{1.398.269} - 1$	420.921	Armengaud, GIMPS
1997	$2^{2.976.221} - 1$	895.932	Spence, GIMPS
1998	$2^{3.021.377} - 1$	909.526	Clarkson, GIMPS
1999	$2^{6.972.593} - 1$	2.098.960	Hajratwala, GIMPS
2001	$2^{13.466.917} - 1$	4.053.946	Cameron, GIMPS
2003	$2^{20.996.011} - 1$	6.320.430	Shafer, GIMPS
2004	$2^{24.036.583} - 1$	7.235.733	Findley, GIMPS
2005	$2^{25.964.951} - 1$	7.816.230	Nowak, GIMPS
2005	$2^{30.402.457} - 1$	9.152.052	Cooper, Boone, GIMPS
2006	$2^{32.582.657} - 1$	9.808.358	Cooper, Boone, GIMPS
2008	$2^{37.156.667} - 1$	11.185.272	Elvenich, GIMPS
2009	$2^{42.643.801} - 1$	12.837.064	Strindmo, GIMPS
2008	$2^{43.112.609} - 1$	12.978.189	Smith, GIMPS

Ausblick: Für die erste bekannte Primzahl mit mehr als 100 Millionen Stellen hat die EFF einen weiteren Preis in Höhe von 150.000 Dollar ausgesetzt. Meiner Meinung nach wird bei gleicher Erhöhung der Rechnerleistungen wie in den vergangenen Jahren dies erst 2018 mit Hilfe von Standard-PCs möglich sein. Ziel meiner

weiteren Aktivitäten ist es, ein Verfahren zur Generierung von Primzahlen zu finden bzw. im Umkehrschluss einen schnellen Test für spezielle Primzahlen. Primzahl-Enthusiasten können ihre interessanten Projektarbeiten unter meiner Webseite [www.primzahlen.de](http://www.primzahlen.de) veröffentlichen.

## Komplexe Multiplikation: von numerisch bis symbolisch

Andreas Enge  
(INRIA Bordeaux–Sud-Ouest)

[andreas.enge@math.u-bordeaux.fr](mailto:andreas.enge@math.u-bordeaux.fr)



Die Theorie der komplexen Multiplikation vereint in bemerkenswerter Weise Analysis (Funktionentheorie, Riemannsche Flächen) und Algebra (Zahlentheorie, Klassenkörpertheorie). In der Praxis führt das dazu, dass sich algebraische, diskrete Objekte mit analytischen, numerischen Methoden berechnen lassen.

### Anwendungen

Die Hauptanwendung der komplexen Multiplikation besteht darin, elliptische Kurven über einem endlichen Körper mit vorab bekannter Punktezahl zu konstruieren. Sei  $D < 0$  eine quadratische Diskriminante und  $q$  eine

Primzahlpotenz, so dass die Gleichung

$$4q = t^2 - v^2 D \quad (1)$$

eine Lösung in ganzen Zahlen  $t, v$  mit  $\text{ggT}(t, v) = 1$  besitzt. Dann gibt es eine elliptische Kurve über  $\mathbb{F}_q$  mit  $N = q + 1 - t$  Punkten; wir werden im folgenden sehen, wie sich eine solche Kurve in Zeit

$$O^\sim(|D|) := O(|D| \log^{O(1)} |D|)$$

bestimmen lässt.

Dies kann ausgenutzt werden, um für die Kryptographie geeignete Kurven zu berechnen. Mit den Fortschritten beim Zählen der Punkte auf zufälligen Kurven<sup>1</sup> wurde diese Anwendung zunächst obsolet, um dann im Zuge paarungsbasierter Kryptographie<sup>2</sup> eine Renaissance

<sup>1</sup>F. Vercauteren: *Counting points on curves over finite fields*, Computeralgebra-Rundbrief 43, 2008, S. 16–19

<sup>2</sup>F. Heß: *Kryptographie mit elliptischen Kurven*, Computeralgebra-Rundbrief 39, 2006, S. 14–18

zu feiern – die dort auftretenden Restriktionen lassen sich nicht „zufällig“ erfüllen.

Eine weitere wichtige Anwendung sind Primzahlbeweise und -zertifikate (ECPP) nach [1], wie sie z. B. im Computeralgebrasystem MAGMA implementiert sind.

## Elliptische Kurven mit komplexer Multiplikation

Eine *elliptische Kurve*  $E$  ist eine affine Kurve mit der Gleichung

$$Y^2 = X^3 + aX + b,$$

wobei  $a, b$  Elemente eines Körpers sind. (Im folgenden ist dies  $\mathbb{C}$  oder ein endlicher Körper  $\mathbb{F}_q$ ; für Körper der Charakteristik 2 oder 3 muss die Gleichung leicht angepasst werden.) Die Punkte auf der Kurve zusammen mit einem „unendlich fernen“ Punkt bilden eine algebraische abelsche Gruppe, in der die Summe zweier Punkte durch rationale Formeln in ihren Koordinaten und in  $a$  und  $b$  gegeben ist. Über  $\mathbb{C}$  lässt sich dieses Gruppengesetz auch analytisch darstellen: Für ein Gitter  $\mathfrak{a}$  liefert die Differentialgleichung der Weierstraßschen Funktion  $\wp_{\mathfrak{a}}$  eine Parametrisierung

$$(\wp_{\mathfrak{a}}, \wp'_{\mathfrak{a}}/2) : \mathbb{C}/\mathfrak{a} \rightarrow E$$

als Riemannsche Fläche; das Gruppengesetz entspricht der Addition in  $\mathbb{C}/\mathfrak{a}$ .

Ein *Multiplikator* oder *Endomorphismus* von  $E$  ist dann ein  $\alpha \in \mathbb{C}$  mit  $\alpha\mathfrak{a} \subseteq \mathfrak{a}$ . Neben dem trivialen Fall, in dem nur die ganzen Zahlen als Multiplikatoren auftreten, kann es vorkommen, dass der Endomorphismenring die Ordnung  $\mathcal{O}_D$  der Diskriminante  $D$  in einem imaginär-quadratischen Zahlkörper  $K = \mathbb{Q}(\sqrt{D})$  ist; man spricht dann von *komplexer Multiplikation*. Dies ist genau dann der Fall, wenn  $\mathfrak{a}$  ein eigentliches Ideal von  $\mathcal{O}_D$  ist.

Eine elliptische Kurve ist bis auf Isomorphie durch ihre  $j$ -Invariante

$$j(E) = 1728 \frac{4a^3}{4a^3 + 27b^2}$$

charakterisiert. Im Fall komplexer Multiplikation lässt sich diese auch wie folgt darstellen: Sei

$$\mathbb{H} = \{z \in \mathbb{C} : \Im z > 0\},$$

und für ein Ideal  $\mathfrak{a} = \left(A, \frac{-B + \sqrt{D}}{2}\right)$  von  $\mathcal{O}_D$  sei

$$\tau = \frac{-B + \sqrt{D}}{2A}$$

der Basisquotient in  $\mathbb{H}$ . Dann gibt es eine meromorphe Funktion  $j : \mathbb{H} \rightarrow \mathbb{C}$  mit  $j(\mathfrak{a}) := j(\tau) = j(E)$ . Die Funktion  $j$  ist *modular* für

$$\Gamma = \mathrm{Sl}_2(\mathbb{Z})/\{\pm 1\},$$

das bedeutet: Für  $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$  gilt

$$j(Mz) := j\left(\frac{az + b}{cz + d}\right) = j(z). \quad (2)$$

Dies zeigt, dass  $j(\mathfrak{a})$  nur von  $\mathfrak{a}$ , nicht jedoch von der Wahl einer speziellen Basis und somit von  $\tau$  abhängt. Wegen der Basisquotientenbildung hängt  $j(\mathfrak{a})$  genauer nur von der *Idealklasse* von  $\mathfrak{a}$  modulo Hauptidealen ab. Bezeichnet  $\mathrm{Cl}_D$  die *Klassengruppe* der Ordnung  $\mathcal{O}_D$  und  $h_D := |\mathrm{Cl}_D|$  ihre *Klassenzahl*, so gibt es also bis auf Isomorphie genau  $h_D$  verschiedene elliptische Kurven mit komplexer Multiplikation durch  $\mathcal{O}_D$  – ein erster Brückenschlag zur Zahlentheorie.

Wegen (2) ist  $j$  invariant unter der Translation

$$z \mapsto z + 1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} z$$

und lässt sich daher als Funktion der Fourier-transformierten Variablen  $q = e^{2\pi iz}$  betrachten. Dann gilt

$$j = q^{-1} + \sum_{k=0}^{\infty} c_k q^k \quad (3)$$

mit  $c_k \in \mathbb{Z}$ ; mit anderen Worten:  $j$  ist eine meromorphe Funktion auf der Kompaktifizierung der Riemannschen Fläche  $\Gamma \backslash \mathbb{H}$ . Praktisch bedeutet (3), dass sich  $j$  in jedem Argument  $\tau \in \mathbb{H}$  numerisch mit beliebiger Genauigkeit berechnen lässt.

## Elliptische Kurven über endlichen Körpern

Endomorphismen haben auch eine rein algebraische Interpretation. So entspricht die Multiplikation mit der vierten Einheitswurzel  $i$  auf der Kurve  $Y^2 = X^3 + X$  der Abbildung

$$(x, y) \mapsto (-x, iy),$$

die in der Tat nach viermaliger Anwendung zur Identität wird. Allgemeiner werden Endomorphismen algebraisch als rationale Abbildungen der elliptischen Kurve (gesehen über dem algebraischen Abschluß des Grundkörpers) auf sich selbst definiert, die gleichzeitig Gruppenhomomorphismen sind.

Über dem algebraischen Abschluss  $\overline{\mathbb{F}}_q$  eines endlichen Grundkörpers  $\mathbb{F}_q$  spielt der *Frobenius-Endomorphismus* eine besondere Rolle. Er ist durch die rationale Abbildung

$$\pi : (x, y) \mapsto (x^q, y^q)$$

gegeben, wobei der kleine Fermatsche Satz und  $a, b \in \mathbb{F}_q$  implizieren, dass das Bild eines Punktes auf  $E$  (mit Koordinaten in  $\overline{\mathbb{F}}_q$ ) wieder auf  $E$  liegt. Ebenso zeigt die  $\mathbb{F}_q$ -Rationalität der Additionsformeln, dass  $\pi$  ein Gruppenhomomorphismus ist. Dabei lässt  $\pi$  die  $\mathbb{F}_q$ -rationalen Punkte auf  $E$  invariant, so dass es sich nicht um die Multiplikation mit einer ganzen Zahl handeln

kann; in diesem Sinne haben also alle elliptischen Kurven über  $\mathbb{F}_q$  komplexe Multiplikation. Falls  $\mathbb{F}_q = \mathbb{F}_{p^m}$  mit  $p$  prim ist, besagt der Satz von Deuring genauer, dass „fast alle“ elliptischen Kurven über  $\mathbb{F}_q$  sich durch Reduktion nach  $p$  einer Kurve über  $\mathbb{C}$  mit komplexer Multiplikation ergeben, wobei der Endomorphismenring erhalten bleibt [5]. (Davon ausgenommen sind die *super-singulären* Kurven, deren Anteil etwa  $1/p$  beträgt.) Insbesondere kann  $\pi$  als ein Element der Norm  $q$  einer Ordnung  $\mathcal{O}_D$  betrachtet werden, woraus sich (1) ergibt.

## Klassenkörpertheorie

Es muss noch geklärt werden, was genau mit der an sich unsinnigen Formulierung, „eine Kurve über  $\mathbb{C}$  modulo einer Primzahl  $p$  zu reduzieren“, gemeint ist. Seien  $K = \mathbb{Q}(\sqrt{D})$ ,

$$\alpha_i = \left( A_i, \frac{-B_i + \sqrt{D}}{2} \right), i = 1, \dots, h_D,$$

ein Vertretersystem von  $\text{Cl}_D$  und  $K_D = K(j(\alpha_1))$ . Der erste Hauptsatz der komplexen Multiplikation [6, §9] besagt, dass  $K_D/K$  eine Galois-Erweiterung mit Gruppe  $\text{Cl}_D$  ist. Die Reduktion findet daher nicht nach  $p$ , sondern nach einem Primideal  $\mathfrak{P}$  in  $K_D$  statt, das über  $p$  liegt und Trägheitsgrad  $m$  hat; das Ergebnis sind dann  $h_D$  Werte von  $j$ -Invarianten in  $\mathbb{F}_q = \mathbb{F}_{p^m}$ . Genauer sind die  $j(\alpha_i)$  ganz-algebraisch, und  $j(\mathcal{O}_D) \in K_D \cap \mathbb{R}$ , wodurch das *Klassenpolynom*

$$H_D(X) = \prod_{i=1}^{h_D} \left( X - j \left( \frac{-B_i + \sqrt{D}}{2A_i} \right) \right) \quad (4)$$

in  $\mathbb{Z}[X]$  liegt. Die  $j(\alpha_i) \bmod \mathfrak{P}$  sind also schlicht die Nullstellen in  $\mathbb{F}_q$  von  $H_D(X) \bmod p$ , womit die Berechnungen wieder auf symbolischer Ebene angelangt sind.

## Algorithmus

Die obigen Ausführungen lassen sich direkt in einen Algorithmus übersetzen:

- 1) Wähle  $D$  und  $q = p^m$  mit (1), so dass die Punktezahl  $N = q+1-t$  die gewünschten Eigenschaften hat.
- 2) Bestimme ein Vertretersystem von  $\text{Cl}_D$  der Form  $\alpha_i = \left( A_i, \frac{-B_i + \sqrt{D}}{2} \right), i = 1, \dots, h_D$ .
- 3) Berechne numerisch die Werte von  $j$  in den  $\alpha_i$ , z. B. mittels (3), und das Klassenpolynom  $H_D$  wie in (4).
- 4) Runde die Koeffizienten von  $H_D$  auf ganze Zahlen, reduziere modulo  $p$  und bestimme eine Nullstelle  $\bar{j} \in \mathbb{F}_q$ .

- 5) Sei  $\gamma$  ein quadratischer Nichtrest in  $\mathbb{F}_q$ . Die beiden elliptischen Kurven  $E : Y^2 = X^3 + aX + b$  und  $E' : Y^2 = X^3 + a\gamma^2 X + b\gamma^3$  mit

$$k = \frac{\bar{j}}{1728 - \bar{j}},$$

$a = 3k, b = 2k$  haben  $j$ -Invariante  $\bar{j}$ , und eine hat  $q+1-t$ , die andere  $q+1+t$  Punkte.

In 1) wäre es wünschenswert, sowohl  $q$  als auch  $N$  festzulegen; dann gilt i. A.  $|D| \approx q$ , was diesen Ansatz für größere Werte von  $q$  nicht praktikabel macht. Stattdessen hält man i. A.  $D$  fest und löst entweder die diophantische Gleichung (1) für verschiedene Werte von  $q$  oder wählt zufällige Werte  $t, v$ , bis  $q$  in (1) zu einer Primzahlpotenz wird. Alternativ kann man  $q$  fixieren und versuchen, (1) für kleine Werte von  $D$  zu lösen; oder man kann  $N$  festhalten, verliert dafür aber die Kontrolle über  $q$ .

Die Existenz zweier Kurven in 5) mit derselben  $j$ -Invariante, aber verschiedener Punktezahl scheint zunächst der Tatsache zu widersprechen, dass die  $j$ -Invariante elliptische Kurven bis auf Isomorphie charakterisiert. In der Tat gilt diese Charakterisierung nur über  $\overline{\mathbb{F}_q}$ , und die beiden Kurven sind genauer über  $\mathbb{F}_{p^2}$  isomorph. (Für  $D = -4$  und  $D = -3$  gibt es nicht nur zwei, sondern vier bzw. sechs Möglichkeiten; dies entspricht der Anzahl der Einheitswurzeln in  $\mathcal{O}_D$ .)

## Beispiel

Für  $D = -23$  und  $p = 6427752177035949684186306721878284835035747081564392976559049$  ist (1) mit  $t = -5070602400912913102387185451082$  und  $v = 1992$  erfüllt; dann hat  $N = 4 \cdot 1606938044258987421046576680470838859359164998666695040502533$  einen Primfaktor von 200 Bit, und die im folgenden berechnete Kurve kann in einem Kryptosystem mit einem Sicherheitsniveau von 100 Bit verwendet werden.

Die Klassengruppe wird durch die drei Ideale

$$\left( 1, \frac{-1 + i\sqrt{23}}{2} \right) \quad \text{und} \quad \left( 2, \frac{\mp 1 + i\sqrt{23}}{2} \right)$$

repräsentiert; die zugehörigen Werte von  $j$  sind

$$j_1 = -3493225, 6999699 \dots$$

und

$$j_{2/3} = 737, 84998496668 \dots \pm 1764, 0189386127 \dots i.$$

Man berechnet  $H_{-23}(X) = X^3 + 3491749, 99 \dots X^2 - (5151296875, 00 \dots + 0, 87 \dots \cdot 10^{-10}i) X + 12771880859374, 90 \dots = X^3 + 3491750X^2 - 5151296875X + 12771880859375$ .

Modulo  $p$  erhält man die Nullstelle  $\bar{j} = 2702246477694751372186410000772995089911829947570287185881304$  und die Kurve  $Y^2 = X^3 + 57979847989453991104160798789265455110899542658590728$

79574388X+348883774594671367325858145477016  
7989955380608272899190630631 mit  $j$ -Invariante  $\bar{j}$   
und mit  $N$  Punkten.

---

## Komplexität

---

Die Klassenpolynome  $H_D$  zeichnen sich durch sehr große Koeffizienten aus, wie das obige Beispiel anschaulich illustriert: Schon das kleinste Polynom vom Grad 3 für  $D = -23$  lässt sich nicht mehr mit dem Taschenrechner oder doppelter Genauigkeit bestimmen; denn es ist klar, dass die Anzahl der Gleitkommastellen mindestens der Anzahl  $n$  der Stellen des größten Koeffizienten entsprechen muss, um korrekt runden zu können. Man kann zeigen, dass  $n \in O(\sqrt{|D|})$  und weiterhin  $h_D \in O(\sqrt{|D|})$ , so dass die Gesamtgröße von  $H_D$  in  $O(|D|)$  liegt. Dabei sind die Konstanten und logarithmischen Faktoren in  $O$  explizit, siehe [9, 2].

Sei  $M(n) \in O(n)$  die Komplexität der Multiplikation zweier Zahlen mit  $n$  Stellen. In [9, 7] werden zwei Algorithmen beschrieben, mit denen sich die benötigten  $h_D$  Werte von  $j$  mit einer amortisierten Komplexität von  $O(M(n))$  pro Wert berechnen lassen, was zu einer Gesamtkomplexität des Algorithmus von  $O(|D|)$  führt – dies ist quasilinear in der Ausgabegröße!

Der erste Ansatz berechnet  $O(n)$  Terme von (3) und nutzt aus, dass sich mittels Algorithmen der modernen Computeralgebra ein Polynom vom Grad  $O(n)$  in  $O(n)$  Argumenten in derselben Zeit  $O(n M(n))$  auswerten lässt wie in einem einzigen Argument [10, §10.1]. Das zweite Verfahren benutzt Newton-Iterationen auf einer Funktion, die als wesentlichen Baustein das arithmetisch-geometrische Mittel (AGM) enthält – dieses lässt sich dank quadratischer Konvergenz in  $O(M(n))$  berechnen.

---

## Numerische Betrachtungen

---

Der Algorithmus mit Gleitkommaberechnungen ist numerisch sehr stabil; für große Klassenzahlen reicht es in der Praxis, die Genauigkeit um 1% höher zu wählen als die erwartete Stellenzahl der Koeffizienten. Für die Auswertung von  $j$  lässt sich dies leicht plausibel machen. Anstelle von (3) geht man für kleinere Klassenzahlen von der Dedekindschen  $\eta$ -Funktion aus und verwendet die folgenden Formeln; der resultierende Algorithmus hat eine Komplexität von  $O(|D|^{5/4})$  anstatt  $O(|D|)$ :

$$\eta(q) = q^{1/24} \sum_{k=-\infty}^{\infty} (-1)^k q^{k(3k-1)/2} \quad (5)$$

$$f_1(q) = \frac{\eta(q^{1/2})}{\eta(q)}, \quad j = (f_1^{24} + 16)^3 / f_1^{24}.$$

Das quadratische Wachstum der Exponenten in (5) führt dazu, dass der absolute Fehler in den Potenzen von  $q$  sehr schnell abnimmt; die Kumulation von Fehlern

durch die Additionen und Subtraktionen in (5) ist also praktisch vernachlässigbar, und der gesamte Rundungsfehler ist im wesentlichen durch die ersten beiden Terme  $1 - q$  bestimmt.

Ab einem Polynomgrad von etwa 100.000 ist der Algorithmus von [7] mit Komplexität in  $O(|D|)$  auch praktisch schneller. Das AGM zweier Zahlen  $a_0, b_0$ , gemeinsamer Grenzwert der beiden Folgen

$$a_{i+1} = \frac{a_i + b_i}{2}, \quad b_{i+1} = \sqrt{a_i b_i},$$

ist numerisch unkritisch, sobald die beiden Ausgangszahlen z. B. im selben Quadranten liegen; das weiterhin benutzte Newton-Verfahren stabilisiert die Auswertung zusätzlich.

---

## Software

---

Die Arithmetik von komplexen Zahlen beliebiger Genauigkeit ist in der C-Bibliothek MPC implementiert und steht auf <http://www.multiprecision.org/> unter der LGPL frei zur Verfügung. Dort findet sich auch ein unter der GPL stehendes Programm, das den oben beschriebenen Algorithmus zur komplexen Multiplikation implementiert.

---

## Ausblick

---

Anstelle der Klassenpolynome (4) verwendet man in der Praxis Polynome zu alternativen Modulfunktionen, die asymptotisch um einen konstanten Faktor kleiner sind. Typischerweise liegt dieser Faktor zwischen 12 und 72. Für ausführlichere Abhandlungen sei auf das in Kürze erscheinende Buch [11] und das erste Kapitel von [8] hingewiesen.

Es soll nicht unerwähnt bleiben, dass neben dem hier dargestellten komplexen Lift auch ein  $p$ -adischer Lift des Klassenpolynomes möglich ist [4, 3]. Auch dieser ist quasilinear in der Ausgabegröße, so dass beide Algorithmen schließlich nicht durch ihre Laufzeit, sondern durch den verfügbaren Speicherplatz begrenzt sind: Das größte mit dem Gleitkommaalgorithmus berechnete Klassenpolynom hat einen Grad von  $h_D = 100.000$  und belegt etwa 5 GB im Hauptspeicher, wobei die Rechenzeit nur etwa 3 Tage beträgt und mögliche Parallelisierungen nicht implementiert wurden [9].

Vor Kurzem wurde ein ebenfalls quasilinearer, rein symbolischer Algorithmus entwickelt, der Koeffizient nach Koeffizient mittels des chinesischen Restsatzes berechnet und ausgibt [2, 12]. Dadurch kommt er mit weniger Hauptspeicher aus, und Rekordberechnungen erreichen eine Klassenzahl von 5.000.000. Für kleinere Klassenzahlen bis etwa 1.000 bleibt der numerische Algorithmus die beste Wahl.



## Literatur

- [1] A. O. L. Atkin, F. Morain, *Elliptic Curves and Primality Proving*, Mathematics of Computation (61), 1993, 29-68.
- [2] J. Belding, R. Bröker, A. Enge, K. Lauter, *Computing Hilbert class polynomials*, Algorithmic Number Theory – ANTS-VIII, Springer, Lecture Notes in Computer Science (5011), 282-295.
- [3] R. Bröker, P. Stevenhagen, *Elliptic Curves with a Given Number of Points*, Algorithmic Number Theory – ANTS-VI, Springer, Lecture Notes in Computer Science (3076), 117-131.
- [4] J. M. Couveignes, T. Henocq, *Action of Modular Correspondences around CM Points*, Algorithmic Number Theory – ANTS-V, Springer, Lecture Notes in Computer Science (2369), 234-243.
- [5] M. Deuring, *Die Typen der Multiplikatorenringe elliptischer Funktionenkörper*, Abhandlungen aus dem mathematischen Seminar der hamburgischen Universität (14), 1941, 197-272.
- [6] M. Deuring, *Die Klassenkörper der komplexen Multiplikation*, Enzyklop. d. math. Wissenschaften (I 2 Heft 10), Teubner, 1958.
- [7] R. Dupont, *Fast evaluation of modular functions using Newton iterations and the AGM*, to appear in Mathematics of Computation, [http://www.lix.polytechnique.fr/Labo/Regis.Dupont/preprints/Dupont\\_FastEvalMod.ps.gz](http://www.lix.polytechnique.fr/Labo/Regis.Dupont/preprints/Dupont_FastEvalMod.ps.gz)
- [8] A. Enge, *Courbes algébriques et cryptologie*, Habilitation à diriger des recherches, Université Denis Diderot, Paris 7, 2007.
- [9] A. Enge, *The complexity of class polynomial computation via floating point approximations*, Mathematics of Computation (78), 2009, 1089-1107.
- [10] J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 1999.
- [11] R. Schertz, *Complex Multiplikation*, Cambridge University Press, 2010.
- [12] A. V. Sutherland, *Computing Hilbert class polynomials with the Chinese remainder theorem*, Preprint ArXiv 0903.2785v1, 2009.

### Das SCIENCE EU Programm: Symbolic Computation Infrastructure for Europe

**Peter Horn**  
Universität Kassel

**Dan Roozmond**  
Technische Universität Eindhoven

horn@math.uni-kassel.de  
d.a.roozmond@tue.nl



Das SCIENCE Projekt bringt die Entwickler der drei Computeralgebrasysteme GAP, KANT und Maple, die Universität Kassel und das RISC in Linz als Experten für Computeralgebra und die Technische Universität Eindhoven als Experten für OpenMath mit Arbeitsgruppen zu Cluster- und Grid-Computing aus St. Andrews, Edinburgh und Timisoara (Rumänien) zusammen. Das erklärte Ziel des auf fünf Jahre ausgelegten Framework 6 EU-Programmes, das noch bis 2011 läuft, ist es, einerseits die Landschaft der europäischen Computeralgebra-Entwickler zusammenzubringen, und andererseits das symbolische Rechnen auf die Anforderungen der nächsten Jahrzehnte mit einem massiven Wachstum der Anzahl der CPU-Kerne vorzubereiten.



OpenMath ist eine sehr flexible Sprache, die nur aus zwölf Sprachelementen besteht (Ganze Zahlen, Fließkommazahlen, Variablen, ...), wobei alle mathematische Semantik in sogenannten Content Dictionaries (CD) definiert ist. So werden etwa in dem CD `arith1` die elementaren arithmetischen Operationen wie `plus`, `minus` etc. definiert, in dem CD `polyd1` wird beschrieben, wie multivariate Polynome dargestellt werden, und das CD `group4` beschäftigt sich mit Cosets und Konjugationsklassen. Die OpenMath-Sprache wurde so gestaltet, dass sie für Computer und Software denkbar effizient zu benutzen ist.

Im aktuellen MathML 3 Standard ist ein OpenMath-Dialekt (Strict Content MathML) für die Codierung von mathematischer Semantik zuständig.



---

### Die Lingua Franca der Computeralgebra: OpenMath und SCSCP

---

Um die verschiedenen Computeralgebrasysteme miteinander zu verbinden, ist es notwendig, eine *gemeinsame* Sprache zu sprechen, wenn man nicht für jede Verbindung einen Spezialfall erzeugen will. Um die mathematische Semantik zu kapseln, war OpenMath [14] die natürliche Wahl, da es ein etablierter Standard ist, der bereits für ähnliche Zwecke eingesetzt wird [3, 4].



Um die Kommunikation zwischen den Systemen zu vereinheitlichen wurde das Protokoll SCSCP („Symbolic Computation Software Composability Protocol“) [5, 6] entwickelt. Dieses Protokoll dient nicht nur dazu, Funktionen eines Systems einem anderen zur Verfügung zu stellen, sondern ist auch die Grundlage für die Cluster- und Grid-Infrastrukturen, die im SCIENCE Projekt entwickelt werden.

Der große Vorteil gegenüber vorherigen Ansätzen, einzelne Systeme miteinander zu verbinden ist, dass nun ein Standard vorliegt, und jedes System, das diesen Standard implementiert, kann anderen Systemen Dienste anbieten bzw. diese nutzen.

Das Protokoll basiert auf XML, und alle Protokollnachrichten sind OpenMath-Objekte. Die eigentliche Kommunikation geschieht über TCP Sockets, standardmäßig über den von der *Internet Assigned Numbers*

Authority (IANA) für diesen Zweck registrierten Port 26133. Das Protokoll liegt in Version 1.3 vor, und es existieren Implementierungen in GAP [7], KANT [10], Maple [12] und MuPAD [13]. Außerdem wurde SCSCP in TRIP [8], Magma [2] und Macaulay2 [11] implementiert.

Um die Anbindung von weiteren Computeralgebrasystemen sowie von anderer Software an die SCSCP Infrastrukturen zu erleichtern, wurde die Java-Bibliothek `org.symcomp.scscp` [9] als Referenzimplementierung entwickelt, und steht unter der Apache2 Lizenz zur Verfügung.

## Beispiel – KANT von MuPAD aus benutzen

Um zu erklären, wie die Systeme interagieren, zeigen wir hier ein kleines Beispiel, in dem wir KANT von MuPAD aus benutzen. Zunächst konstruieren wir Swinnerton-Dyer-Polynome, indem wir für verschiedene Primzahlen  $p_1, \dots, p_n$  definieren:

$$P_n(x) = \prod (x \pm \sqrt{p_1} \pm \sqrt{p_2} \dots \pm \sqrt{p_n}).$$

Dies ist ein Polynom vom Grad  $2^n$ , das über  $\mathbb{Z}$  irreduzibel ist, aber über jedem endlichen Körper in  $2^{n-1}$  quadratische Faktoren zerfällt. Für Polynomfaktorisierungsalgorithmen sind solche Polynome besonders herausfordernd.

```
>> package("OpenMath");
>> swindyer := proc(plist) ... :
>> R:=Dom::UnivariatePolynomial(x,Dom::Rational):
>> p1 := R(swindyer([2,3,5,7,11])):
>> p2 := R(subs(swindyer([2,3,5,7,13,17])),
>>           x=3*x-2):
>> p := p1 * p2:
>> degree(p), nterms(p)
96, 49
```

Damit haben wir ein univariates Polynom  $p$  konstruiert, das das Produkt zweier Swinnerton-Dyer-Polynome ist, und 49 Terme und Grad  $96 = 2^5 + 2^6$  hat.

```
>> st := time(): F1 := factor(p): time()-st
38431
```

Die Faktorisierung in MuPAD benötigt also 38 Sekunden.

Da KANT einen der effizientesten Faktorisierungsalgorithmen hat, wollen wir nun einen KANT SCSCP Server benutzen, der sich 400km entfernt befindet:

```
>> kant := SCSCP("scscp.math.tu-berlin.de",26133):
>> st:=rttime():
>> F2:=kant::compute(hold(factor)(p)):
>> rttime()-st
1221
```

Die Faktorisierung in KANT benötigt nur 1.2 Sekunden, und in dieser Zeit ist die Konversion der Daten nach und von OpenMath und der Netzwerk-Transport enthalten.

Nun verifizieren wir noch (etwas umständlich), dass die Faktoren übereinstimmen:

```
>> FS1 := {op(Factored::factors(F1))}:
>> FS2 := {op(Factored::factors(F2))}:
>> bool(FS1=FS2)
TRUE
```

Auch wenn das Beispiel etwas konstruiert ist, wird doch der Benefit des Ansatzes klar: Man kann in „seinem“ gewohnten System arbeiten und Arbeiten, die andere Systeme besser (oder überhaupt) können, transparent delegieren.

## Ist das nicht Sage?

In der letzten Rundbrief-Ausgabe wurde das CAS *Sage* vorgestellt [1], das aus der Kombination vieler verschiedener Systeme und Bibliotheken hervorgegangen ist. Tatsächlich klingt das recht ähnlich zu dem Vorgehen im SCIENCE-Projekt, lässt sich aber klar abgrenzen.

In *Sage* werden die verschiedenen Systeme einem Python-Interface untergeordnet, so dass dieses eine System die Fähigkeiten der untergeordneten Systeme nutzen kann. Auch laufen all diese Systeme auf der gleichen Maschine. Bei SCSCP kann man wie gewohnt in dem System seiner Wahl arbeiten, und auf Funktionalitäten aus anderen System zugreifen, die auch auf anderen Maschinen laufen können. Außerdem bietet SCSCP die Basis für das Parallelisieren von Berechnungen.

Nichtsdestoweniger ist *Sage* ein interessantes Projekt, und wir würden uns über SCSCP-Unterstützung in *Sage* freuen.

## POPCORN – OpenMath in lecker

Im Rahmen der Entwicklung von SCSCP war es häufig notwendig, auch größere Fragmente OpenMath zu schreiben bzw. zu lesen. Trotz XML-Unterstützung in vielen Editoren ist das sehr unbefriedigend, da XML keine übliche mathematische Notation ist. Andererseits existieren in praktisch allen Computeralgebrasystemen *ähnliche* Notationen. Um die Eingabe und das Lesen von OpenMath zu erleichtern, haben wir POPCORN („Possibly Only Practical Convenient OpenMath Replacement Notation“) entwickelt, das wir aus typographischen Gründen einfach „Popcorn“ schreiben. Popcorn ist kein Ersatz für OpenMath, sondern einfach nur eine andere Repräsentation der OpenMath Sprache.



Ganze Zahlen, Fließkommazahlen und Strings werden genau so dargestellt, wie man es erwartet (also etwa `1.2.67` oder `''text''`), Variablen wird das Symbol `$` vorgestellt, Referenzen das `#`, und OpenMath-Symbole werden mit einem Punkt zwischen CD-Namen und Symbolnamen geschrieben (etwa `arith1.plus`).

Für zusammengesetzte Elemente gelten ähnlich einfache Regeln: Soll ein Element auf andere angewendet werden, so werden einfach die Argumente in Klammern hinter das Element geschrieben. Für Bindings werden eckige, für Attributions geschwungene Klammern benutzt.

Zusätzlich wurden für viele häufig benutzte Symbole Abkürzungen sowie Infix-Operatoren für die wichtigsten Operationen (`+`, `-`, `and`, ...) eingeführt.

---

## Popcorn – Beispiele

---

Wahrscheinlich sind Beispiele nützlicher als langer Text; hier sind jeweils die Popcorn- und die XML-Schreibweise gegenübergestellt.

Für die Addition zweier Zahlen haben wir in Popcorn: 1+2

```
<OMA>
<OMS cd="arith1" name="plus" />
<OMI>1</OMI>
<OMI>2</OMI>
</OMA>
```

Die Funktion, die  $x$  auf  $x + 1$  abbildet schreibt sich in Popcorn: lambda [\$x->1+\$x]

```
<OMBIND>
<OMS cd="fns1" name="lambda" />
<OMBVAR>
<OMV name="x" />
</OMBVAR>
<OMA>
<OMS cd="arith1" name="plus" />
<OMI>1</OMI>
<OMV name="x" />
</OMA>
</OMBIND>
```

Um der Variablen  $a$  eine Liste aus  $\frac{1}{2}$  und der komplexen Zahl  $2 + 8i$  zuzuweisen, schreiben wir in Popcorn \$a := [1/2, (2|8):x] (die komplexe Zahl bekommt noch die 'id' x)

```
<OMA>
<OMS cd="prog1" name="assign" />
<OMV name="a" />
<OMA>
<OMS cd="list1" name="list" />
<OMA>
<OMS cd="nums1" name="rational" />
<OMI>1</OMI><OMI>2</OMI>
</OMA>
<OMA id="x">
<OMS cd="nums1" name="complex_cartesian" />
<OMI>2</OMI><OMI>8</OMI>
</OMA>
</OMA>
</OMA>
```

Das Integral  $\int_0^1 \frac{1}{x^3 + \cos(x)} dx$  schreibt sich in Popcorn als defint(0 .. 1, lambda[\$x -> 1/(\$x^3 + cos(\$x))]), in XML wird das

```
<OMA>
<OMS cd="calculus1" name="defint"/>
<OMA>
<OMS cd="intervall1" name="interval"/>
<OMI>0</OMI><OMI>1</OMI>
</OMA>
<OMBIND>
<OMS cd="fns1" name="lambda"/>
<OMBVAR><OMV name="x"/></OMBVAR>
<OMA>
<OMS cd="arith1" name="divide"/>
<OMI>1</OMI>
<OMA>
<OMS cd="arith1" name="plus"/>
<OMA>
<OMS cd="arith1" name="power"/>
<OMV name="x"/>
<OMI>3</OMI>
</OMA>
<OMA>
<OMS cd="transcl1" name="cos"/>
<OMV name="x"/>
```

```
</OMA>
</OMA>
</OMA>
</OMBIND>
</OMA>
```

Hier wird ein zweiter Grund für den Namen Popcorn sichtbar: Etwas sehr Kleines wird zu etwas sehr Großem aufgebläht.

---

## WUPSI

---

Um ein Werkzeug zum Testen und Debuggen der verschiedenen OpenMath und SCSCP Dienste zu haben, wurde WUPSI („Wonderful Universal Popcorn SCSCP Interface“) entwickelt. Es handelt sich dabei um eine Java Kommandozeilenanwendung, die sich an SCSCP Server verbinden kann und dann eine auf Popcorn basierende Eingabemöglichkeit für Kommandos bietet. Außerdem ist eine elementare Hilfe zu OpenMath in WUPSI eingebaut.



Damit ist WUPSI das „Schweizer Taschenmesser“ für OpenMath und SCSCP.

Zur Illustration dient am Besten wieder ein kleines Beispiel:

```
WUPSI 1.x -- Wonderful Universal Popcorn SCSCP
Interface
(c) 2009 D. Roozmond & P. Horn
```

```
WUPSI[n/a]0> connect some.server:26139 as gap
# connected to 'some.server' on port '26139' using
# symbolic name 'gap'
# Service Info: service Name 'GAP', service
# version '4.dev'
```

```
WUPSI[gap]0> 126+2323*232
539062
```

```
WUPSI[gap]1> local \a := \$_out0
# Stored this in local variable '\a':
539062
```

```
WUPSI[gap]2> connect 127.0.0.1:26134 as mupad
# connected to '127.0.0.1' on port '26134' using
# symbolic name 'mupad'
```



```
# Service Info: service Name 'MuPAD', service
version '0.6.0-mupad-5.2.0'

WUPSI[mupad]2> output format latex
# switched output format to LATEX.

WUPSI[mupad]2> sum(1 .. infinity, lambda[$x -> 1/
  $x^2])
{\pi}^2 \cdot \frac{1}{6}

WUPSI[mupad]3> output format popcorn
# switched output format to POPCORN.

WUPSI[mupad]3> local $p := 2^127-1
# Stored this in local variable '$p':
170141183460469231731687303715884105727

WUPSI[magma]4> use gap
# switched to system with symbolic name 'gap',
service Name 'GAP', service version '4.dev'.

WUPSI[gap]4> $p-2^101*$a
168774498924748772136428072069291311103

WUPSI[gap]4> describe arith1.plus
# -- Description for 'arith1.plus' --
The symbol representing an n-ary commutative
function plus.
# -- END description for 'arith1.plus' --
```

WUPSI bietet über diese Fähigkeiten hinaus noch viele andere Möglichkeiten, mit SCSCP Servern und Clients zu interagieren, etwa um Berechnungen automatisch auf verschiedene Systeme zu parallelisieren. Nicht zuletzt ist es ein hervorragendes Beispiel, wie die Java-Bibliotheken `org.symcomp.openmath` und `org.symcomp.scscp` benutzt werden können.

## Zusammenfassung

Wir haben versucht, einen Überblick über die Aktivitäten im europäischen Projekt „SCIENCE“ zu geben, insbesondere zum auf OpenMath basierenden SCSCP Protokoll, das derzeit von GAP, KANT, Maple, Macaulay2, Magma, MuPAD und TRIP unterstützt wird.

Wir hoffen und erwarten, dass in den nächsten Jahren viele weitere Systeme den Standard implementieren werden, so dass die Computeralgebra-Welt hierdurch weiter zusammenwachsen kann und auch die im Projekt entwickelten Infrastrukturen für Cluster und Grids für diese Systeme benutzbar werden.

## Lizenzen und Verfügbarkeit

Die `org.symcomp.openmath` und `org.symcomp.scscp` Bibliotheken, das MuPAD SCSCP Package [13] und WUPSI sind unter der Apache 2 Lizenz frei verfügbar. Bei GAP, KANT und Maple ist oder wird die SCSCP Unterstützung Teil der Distribution.

## Literatur

[1] M. Albrecht, H. Schilly, *Sage*, Rundbrief Computeralgebra 44, 2009.

- [2] W. Bosma, J. J. Cannon (Eds), *Handbook of Magma Function*. Edition 2.15, School of Mathematics and Statistics, University of Sydney, 2008. <http://magma.maths.usyd.edu.au/>
- [3] O. Caprotti, A. M. Cohen, *Connecting proof checkers and computer algebra using OpenMath*. In: The 12th International Conference on Theorem Proving in Higher Order Logic, Nice, France, September 1999.
- [4] O. Caprotti, A. M. Cohen, M. Riem, *Java Phrasebooks for Computer Algebra and Automated Deduction*. SIGSAM Bulletin, 2000. Special Issue on OpenMath.
- [5] S. Freundt, P. Horn, A. Kononov, S. Linton, D. Roozmond, *Symbolic Computation Software Composability*. In: Intelligent Computer Mathematics, AISC/Calculemus/MKM 2008 Proceedings, Lecture Notes in Computer Science 5144, 2008, Springer, 285-295.
- [6] S. Freundt, P. Horn, A. Kononov, S. Linton, D. Roozmond, *Symbolic Computation Software Composability Protocol (SCSCP) specification*. Version 1.3, 2009. <http://www.symbolic-computation.org/scscp/>
- [7] The GAP Group, *GAP – Groups, Algorithms, and Programming*. Version 4.4.12, 2008. <http://www.gap-system.org>
- [8] M. Gastineau, *SCSCP C Library – A C/C++ library for Symbolic Computation Software Composability Protocol*. IMCCE, 2009. <http://www.imcce.fr/Equipes/ASD/trip/scscp/>
- [9] *Java libraries org.symcomp.openmath, and org.symcomp.scscp*. <http://java.symcomp.org/>
- [10] *KANT/KASH*: <http://www.math.tu-berlin.de/~kant/kash.html>
- [11] *Macaulay2: A software system for research in algebraic geometry*: <http://www.math.uiuc.edu/Macaulay2/>
- [12] *Maple 13*: <http://www.maplesoft.com/>
- [13] *MuPAD OpenMath Package*: <http://mupad.symcomp.org/>
- [14] *OpenMath*: <http://www.openmath.org/>

## Neues aus Waterloo: Maple 13 und MapleSim 2+3

Thomas Richard  
(Scientific Computers GmbH)

T.Richard@scientific.de



Ende April 2009 sind Maple 13 und MapleSim 2 auf den Markt gebracht worden. Zunächst zu einigen wichtigen Neuerungen in Maple:

---

### Grafik und GUI

Ein „Maple Portal“ genanntes Worksheet, das nach der Installation auf dem Desktop abgelegt wird, dient neuen Anwendern als erster Einstiegspunkt in die umfangreiche Dokumentation. Auf oberster Ebene ist es eine Linksammlung wie bei einer Webseite; davon ausgehend verzweigt es in speziellere Themengebiete oder aber zu Einleitungen, die auf Zielgruppen wie Ingenieure, Studenten oder Mathematik-Lehrkräfte zugeschnitten sind.

Auf vielfachen Kundenwunsch bietet Maple jetzt auf allen Plattformen direkten Export von Worksheets nach PDF (*Portable Document Format*). Die Qualitätsstufe lässt sich im Options-Dialog einstellen: entweder durchsuchbarer Text und kleinere Dateigröße – oder Verwendung von Bitmap-„shapes“ für höchste Auflösung.

Mit dem Eintrag „Copy as MathML“ im Kontextmenü lassen sich Ausdrücke einfacher als bisher in andere Applikationen übertragen, sofern diese ebenfalls MathML verstehen.

Der **Worksheet Migration Assistant** erlaubt die Konvertierung von mws-Dateien (wie sie typischerweise in Classic Worksheet erstellt wurden) in das neuere mw-Format von Standard Worksheet. Dabei wird optional 1D-Eingabe (Maple-Notation) in 2D-Eingabe umgewandelt. Praktisch ist dieser Konverter insbesondere bei größeren Sammlungen von mws-Dateien, deren individuelle Anpassung (Laden, Ändern, Speichern) viel zu umständlich wäre.

Ein weiterer Assistent ist der neue **Equation Manipulator**, mit dem man Gleichungen interaktiv umformen kann: Terme auf einer Seite gruppieren, elementare Funktionen und simplify-Befehle auf die Gleichung anwenden, quadratische Ergänzung und etliches mehr.

Fast alle Verbesserungen bei 2D-Plots, die seit Maple 11 eingeführt wurden, sind nun auf den 3D-Fall ausgedehnt worden, beispielsweise Formelsatz inklusive griechischer Zeichen, physikalische Einheiten, mehr Freiheiten bei Achsen-Einteilung und -Beschriftung. Moderne XML-basierte Exportformate (COLLADA,

Extensible 3D und X3D Geometry) sind hinzugekommen, einige veraltete Formate hingegen entfernt worden. Bei der interaktiven Rotation von 3D-Plots hat sich das Koordinatensystem geändert: man stellt jetzt drei statt zwei Winkeln ein – hier muss man sich ein wenig umgewöhnen, hat aber mehr Informationen im Blick. Mit den sog. Viewpoint-Animationen sind Kamerafahrten um oder durch Objekte möglich. Weitere Änderungen „unter der Oberfläche“ finden sich im Abschnitt „**Technisches**“.

Der CAD-Link unterstützt nun auch NX 6.0 von Siemens PLM (früher UGS).

---

### Symbolik und Numerik

Wie gewohnt werden in der neuen Version weitere Klassen von Differentialgleichungen symbolisch gelöst und spezielle Befehle in den Paketen PDEtools und DEtools eingeführt. So lassen sich beispielsweise rein polynomiale Lösungen von partiellen Differentialgleichungen ermitteln.

Erweiterungen an den numerischen Lösern für ODEs und DAEs sind aufgrund von Erfordernissen und Erfahrungen bei der Simulation physikalischer Modelle mittels MapleSim eingeflossen.

Das verbesserte **int**-Kommando findet jetzt Stammfunktionen zu weiteren Integranden, die z. B. Ci, Si, erf oder die Fresnel-Integrale enthalten. Optional kann man eine oder mehrere Methoden direkt vorgeben und sich den Lösungsvorgang per **infolevel[IntegrationTools]:=3** bis zu einer gewissen Detailtiefe anschauen. Außerdem gibt es eine vereinfachte Syntax für mehrdimensionale und numerische Integration.

Die ohnehin schon umfangreichen Pakete zur Graphentheorie und zur Differentialgeometrie wurden erweitert – ersteres etwa durch weitere spezielle Graphen und neue Dateiformate, letzteres durch ein Unterpaket zu Tensoren.

Die Routinen des in Maple 12 eingeführten Pakets **DynamicSystems** sind nun auch per Kontextmenü zugänglich, sodass für viele regelungstechnische Aufgaben gar keine Befehle mehr eingegeben werden müssen. Ebenfalls für dieses Themengebiet relevant sind neue

Befehle in **LinearAlgebra** zur numerischen Lösung bestimmter Matrixgleichungen: **SylvesterSolve** und (als Spezialfall hiervon) **LyapunovSolve**.

Das neue **Student**-Unterpaket **NumericalAnalysis** enthält Rechen- und Grafikroutinen sowie interaktive Tutoren für die Numerik-Ausbildung; die behandelten Themengebiete reichen von Taylorentwicklung über lineare Gleichungssysteme bis zur Lösung von Anfangswertaufgaben für gewöhnliche Differentialgleichungen.

Ein syntaktische Ergänzung der Programmiersprache Maple: mit dem angehängten Tilde-Zeichen kann praktisch jeder binäre oder unäre Operator und Funktionsaufruf elementweise auf Datenstrukturen angewendet werden. Dies ist lesbarer und oft auch effizienter als explizites Mapping.

Neben dem Einstiegshandbuch *User Manual* (dem früheren *Learning Guide*) sind nun auch die beiden Programmierhandbücher *Introductory* und *Advanced Programming Guide* in die Hilfe von Maple aufgenommen worden. Darüber hinaus stehen sie weiter als PDF-Downloads im Maplesoft Documentation Center sowie (außer bei der Student Edition) in gedruckter Form zur Verfügung. In der eingebauten Hilfe unter „Applications and Examples“ sind vier weitere ausführliche Demonstrationsbeispiele hinzugekommen.

Das Programmieren von Multithread-Anwendungen ist inhärent kompliziert, wenn man sich selbst um die Synchronisation der Threads kümmern muss, daher wurde jetzt ein Unterpaket für das wesentlich komfortablere „Task“-Modell ergänzt. Hiermit überlässt man – verkürzt gesagt – Maple den Low-Level-Teil der Aufgabenverteilung, sodass man sich auf die mathematischen und algorithmischen Aspekte konzentrieren kann.

---

## Technisches

---

Eine sofort wahrnehmbare Verbesserung ist die Nutzung von hardware-beschleunigtem OpenGL für 3D-Plots. Diese beanspruchen nun weniger Speicher, und die Wartezeiten mit der Meldung „Initializing plot...“ sind vorbei. Allerdings setzt dieses Feature einen relativ aktuellen Treiber für die Grafikkarte voraus. Detaillierte Hinweise dazu finden sich am Ende der Install.html von Maple 13 sowie auf den FAQ-Seiten bei Maplesoft. In der eingebauten Hilfe ist das Thema unter **?plot3d,gldriver** dokumentiert. Sollte dennoch ein Problem offen bleiben, bitten wir um Rückmeldung an [maple.support@scientific.de](mailto:maple.support@scientific.de) mit Angabe der Konfiguration.

Bei Linux werden einige neuere Distributionen unterstützt, ältere sind dafür entfallen. Das Update auf Maple 13.02 bringt erstmals offizielle Unterstützung für Windows 7 als Plattform. Mac OS X 10.6 steht zwar nicht explizit in der Liste, bereitet aber ersten Erfahrungen zufolge keine Probleme.

Seit Version 13 wird Maple auf einer DVD ausgeliefert, welche die früheren CD-Sets ablöst. Es handelt sich um eine Hybrid-DVD, d. h. sie enthält die Installer für alle Plattformen; sie sind jedoch nur unter dem jeweils

passenden Betriebssystem sichtbar. Auf Wunsch erhält der Kunde zusätzlich individuelle Download-Links.

Auch die meisten Toolboxes (Maple Toolbox for MATLAB, BlockImporter for Simulink, Global Optimization, Grid Computing, Financial Modeling) sowie MapleNet sind nun auf einer DVD mit einem gemeinsamen Installer zusammengefasst, was u. A. die IT-Administratoren entlasten dürfte.

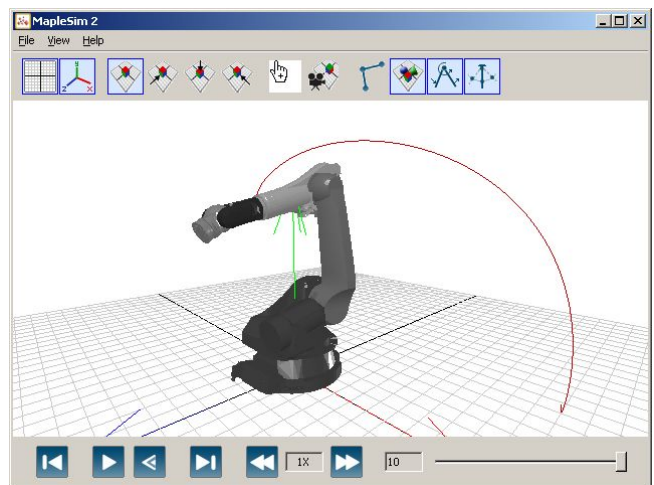
Außerdem ist ein separater **Maple Player** in Vorbereitung, welcher Worksheets in einem bestimmten Dateiformat darstellen und ausführen (jedoch nicht erstellen) kann – und dazu ohne ein installiertes Maple auskommt. Auf diese Weise kann zum Beispiel ein Autor seine Lehrinhalte online oder begleitend zu einem Buch einer wesentlich größeren Leserschaft anbieten.

---

## MapleSim 2 und 3

---

Gleichzeitig mit Maple 13 ist MapleSim 2 erschienen, die aktuelle Version der auf Maple basierenden Modellierungs- und Simulationsumgebung. Sie fügt eine leistungsfähige 3D-Visualisierung hinzu, mit der man insbesondere Mehrkörpersysteme realistisch veranschaulichen kann (die Abbildung zeigt einen Industrieroboter, dessen Nachschwing-Verhalten am Ende der Bahnkurve sich mit rein numerischer Software nur schwer erfassen lässt).



Visualisierung eines Industrieroboters mit MapleSim 2

Die Geometrie kann aus STL-Dateien von CAD-Systemen übernommen werden. Weitere Neuerungen beinhalten einen *Results Manager* zur Verwaltung von Modell-Varianten und Simulationsergebnissen, zusätzliche Analyse-Worksheets (Monte-Carlo-Simulation, Zufallsgeneratoren, Sensitivitätsanalyse), verbesserte Performance und neue Komponenten, z. B. digitale Elektronikbausteine.

Für Ende Oktober, also kurz nach Erscheinen dieser Rundbrief-Ausgabe, ist Version 3 angekündigt, die grundlegende Hydraulikkomponenten als neue Domain enthält und eine Projektverwaltung, komfortablere Hilfe sowie neue Diagnosemöglichkeiten bietet. Danach wird eine umfangreiche Sammlung von Add-Ons erscheinen, d. h. anwendungsspezifische Domain-Libraries (etwa

zum Thema Pneumatik) und Toolboxen, darunter Konnektoren zur LabVIEW-Welt von National Instruments.

Auf den Webseiten des Herstellers unter [www.maplesoft.com](http://www.maplesoft.com) ist seit ein paar Monaten die komplette Dokumentation aller Produkte erreichbar, die

bisher nur innerhalb der jeweils eingebauten Hilfe zugänglich war. Bei uns unter [www.scientific.de](http://www.scientific.de) erscheinen demnächst deutschsprachige Anwendungsbeispiele zu MapleSim. Anregungen dazu sind jederzeit willkommen.

---

## Computeralgebra in der Schule

---

### CAS-Einsatz aus Sicht der Schule

**Jan Hendrik Müller**  
(Rivius-Gymnasium Attendorn)

[jan.mueller@math.uni-dortmund.de](mailto:jan.mueller@math.uni-dortmund.de)



Möchte man an seiner Schule ein CAS einführen, so stellen sich vielfältige Fragen: Soll es ein CAS als Softwarelösung oder integriert im Taschenrechner sein? Wer soll es bezahlen und was darf es kosten? Wie gehen wir nach der Anschaffung damit um? Diese Fragen wurden an unserer Schule intensiv diskutiert. Über diesen Prozess wird in diesem Artikel berichtet.

---

### Welches CAS ist sinnvoll für uns?

Der Einsatz eines CAS in der Schule stellt vielerlei Ansprüche: oft vielmehr organisatorischer als fachlicher Art. Unsere Fachkonferenz diskutierte zunächst die Frage, ob wir ein CAS als Software kaufen (und auf den Rechnern im Computerraum installieren) oder Taschenrechner, in denen ein CAS integriert ist. Die Softwarelösung erwies sich als preiswerter als die Anschaffung von Taschenrechnern. Unter <http://maxima.sourceforge.net/> kann man das CAS wxMaxima sogar kostenlos herunterladen. Preiswerte Softwarelösungen sind jedoch oftmals nicht netzwerktauglich. Ein CAS-Taschenrechner würde diese Probleme nicht aufwerfen.

Die Bedienbarkeit von CAS-Software am Computer durch die gewohnte Steuerung mit Maus und Tastatur ist ebenfalls deutlich komfortabler als bei vielen CAS-Taschenrechnern: Die Tasten sind bei den Geräten oft sehr klein und mehrfach belegt. Zudem kann CAS-Software von allen Schülern einer Klasse ausschließlich nur im Computerraum genutzt werden. Eine individuelle Nutzung von CAS-Software im Computerraum setzt wiederum das Vorhandensein der Schüleranzahl

entsprechend vieler Rechner voraus (die zudem auch alle funktionsbereit sein müssen). Die Nutzung des Taschenrechners kann hingegen in jedem Klassenraum erfolgen. CAS-Taschenrechner booten – wenn überhaupt – wesentlich schneller als Computer und müssen langfristig auch nicht so aufwändig gepflegt und gewartet werden.

An unserer Schule hat sich die Fachschaft in Abwägung dieser und weiterer Argumente für die Anschaffung von Taschenrechnern entschieden, was die Frage nach dem Modell aufwarf. Um dies zu klären, kontaktierten wir Fachhändler und bestellten verschiedene mit CAS ausgestattete Geräte in ausreichender Zahl (jeweils 20) zur Ansicht. Diese wurden Klassen zur Erprobung ausgehändigt und anschließend gemeinsam erörtert, welches Gerät am geeignetsten erschien.

---

### Wie sind wir bei der Anschaffung vorgegangen?

---

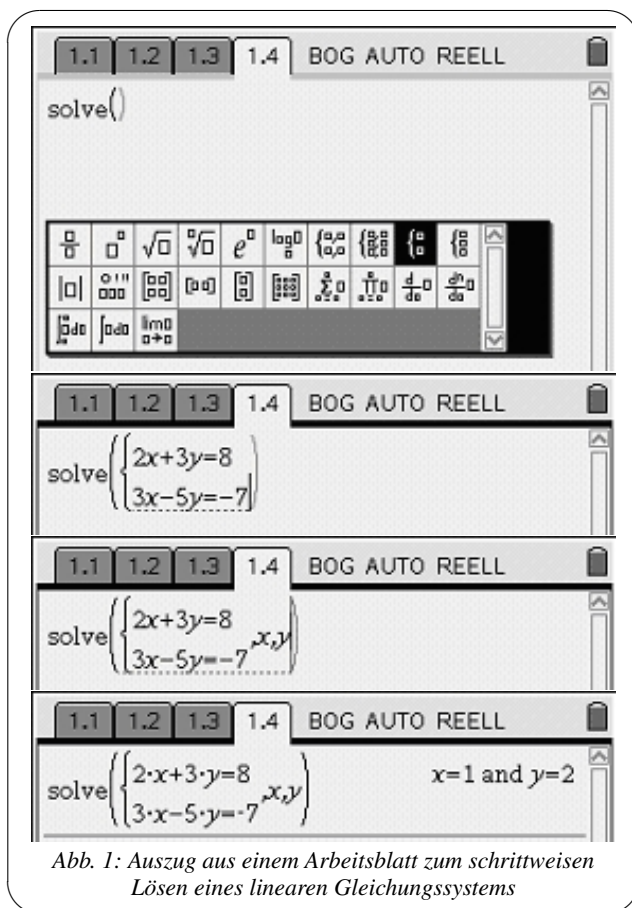
Die Fachkonferenz hatte zunächst den Wunsch, dass sich jeder Schüler ab Klasse 7 ein solches Gerät kauft. Dieser Wunsch stieß schnell auf die folgenden Gegenargumente:

- Der Anschaffungspreis von ca. 150 € war etwa der Hälfte aller Eltern deutlich zu hoch.
- Das CAS nimmt den Kindern das „Denken“ ab.
- Wird das CAS auch von jeder Kollegin und jedem Kollegen gleich intensiv genutzt?



- Kann man ein CAS auch in anderen Fächern nutzen? Oder müssen Eltern demnächst auch für den Englischunterricht einen Sprachcomputer kaufen?

Das Argument „Denkaktivität“ ließe sich sicher entkräften, aber alle anderen Argumente waren nachvollziehbar. Unsere Lösung war schlussendlich folgende: Durch den Einsatz zeitgemäßer Technologien erhalten die Lernenden eine umfangreichere Ausbildung, von der auch die Industrie unserer Stadt profitiert. Somit sollte man versuchen, industrielle Drittmittel einzuwerben. Schließlich wurde auch ein großindustrieller Betrieb gefunden, der bereit war, die Schule mit etwa 5000 € zu unterstützen. Es wurde ein Klassensatz CAS-Taschenrechner (35 Stück – dies entspricht in etwa dem 20. Teil unserer Schüleranzahl) einschließlich einer ausreichenden Anzahl an Akkus und zwei Ladegeräten angeschafft. Diese Anzahl erweist sich als ausreichend für unsere Schule, da die Taschenrechner nicht permanent genutzt werden und nicht streng gleichschrittig unterrichtet wird.



## Wie nutzen wir das CAS (mittlerweile)?

Rückblickend lassen sich zwei wesentliche Strategien im Umgang mit dem CAS nennen, die in Unterrichtsstunden in der Regel nicht gleichermaßen gut realisiert werden können: die Einführung in die Bedienung und die Nutzungs- und Anwendungsmöglichkeiten.

### 1. Zur Einführung in die Bedienung des CAS:

Ein CAS lässt sich meist nicht intuitiv bedienen. Es zeigt sich im Unterricht aller Erfahrung nach jedoch, dass es gravierende Unterschiede darin gibt, wie schnell verschiedene (oftmals aber eher wenige) Schüler den Umgang mit neuen Technologien lernen. Diese „Leistungsschere“ gilt es also methodisch zu kompensieren. Deshalb finden rein „technische“ Einweisungen bei uns eher in Übungsphasen von Unterrichtsreihen statt. Das CAS kann hier zur selbständigen Überprüfung von Rechenergebnissen genutzt werden: Wurde der herkömmliche Taschenrechner genutzt, um Ergebnisse von Rechenaufgaben zu überprüfen, so erweitert das CAS diese Möglichkeit auf den Kontext algebraischer Aufgabenstellungen.

Diese Art der Nutzung erfordert eine Sensibilisierung für einen sinnvollen Einsatz des CAS: Übungsaufgaben, z. B. zu lösende lineare Gleichungen, sollten handschriftlich vom Schüler überprüft werden können. Die handschriftliche Überprüfung ist jedoch dann wenig sinnvoll, wenn z. B. bei Gleichungen mit „krummen“ Koeffizienten (wie etwa  $2,71x + 3,14 = 3,14x - 2,71$ ) eine Probe ebenso mit Rechenfehlern behaftet sein kann wie die Rechnung, die zur Lösung führt. Irgendwo im „Komplexitätskontinuum“ zwischen  $2x + 3 = 3x - 2$  und  $ex + \pi = \pi x - e$  muss also vom Lernenden abgewägt werden, ob der Einsatz eines CAS angemessen erscheint oder nicht. Damit sich die Lernenden die Möglichkeit erhalten, sich selbständig in den Umgang mit dem Gerät einzuarbeiten, können vom Lehrer erstellte Informationsblätter genutzt werden (Abb. 1).

### 2. Zu Nutzungs- und Anwendungsmöglichkeiten des CAS:

Das CAS wird an unserer Schule in Klasse 7 im Kontext der Termumformungen eingeführt. Dies ist die Schnittstelle zwischen Arithmetik zur Algebra, bei der Lernende erstmals mit der Kalkülvorstellung zum Variablenkonzept konfrontiert sind. Ein CAS kann hier Hilfe leisten:

#### (a) Das CAS liefert Fragestellungen:

Ein CAS kann z. B. als Lieferant für fragwürdige Situationen genutzt werden. Schreibkonventionen, Rechenregeln und Rechengesetze im Kontext der Terme können hiermit erschlossen werden. Abb. 2 zeigt eine Möglichkeit, das CAS am Anfang einer Unterrichtsstunde zu nutzen. Welche Gründe gibt es dafür, dass die Eingaben vom CAS verändert wurden? Vor- und Nachteile der „mathematischen Stenografie“ sind abzuwägen, weiterführende Aufgaben zu berechnen und selbst erfundene Terme zu komprimieren und zu expandieren. Es gibt eine Fülle fachdidaktischer Aufsätze mit guten Ideen, die den Einsatz eines CAS auf diese Weise beschreiben (vgl. z. B. den Schulversuch CALIMERO). Die Nutzung des CAS steht hierbei am Anfang des Lernprozesses, da es Fragen aufwerfen soll. Der Einsatz wird vom Lehrer initiiert, da er die Aufgabenstellungen entsprechend seiner Unterrichtsziele

und Lehrplanvorgaben auswählen muss. Zudem ist das CAS hierbei zunächst der Mittelpunkt des Lerngeschehens, denn es wirft selbst Fragen auf und „beantwortet“ Lösungsvorschläge der Schüler.

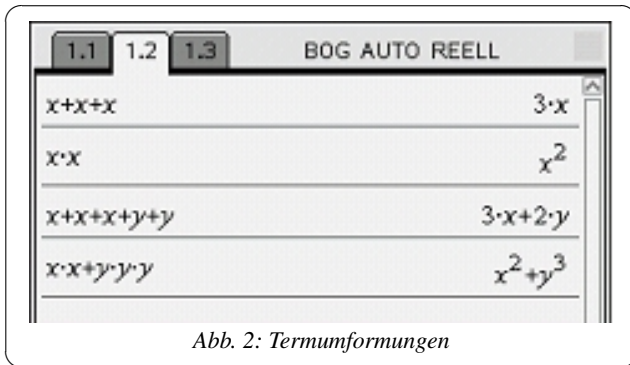


Abb. 2: Termumformungen

Demzufolge ist es erforderlich, dass alle Schüler (zumindest partnerweise) über ein CAS verfügen, um angemessen mit- und weiterarbeiten zu können. Der Unterricht ist divergent ausgerichtet, da die Schüler im Umgang mit dem CAS neue Fragen und Probleme selber finden, stellen und beantworten sollen (Idee der Aufgabenvariation).

#### (b) Das CAS liefert Lösungen:

Ein CAS kann ebenso im Kontext bereits gestellter Probleme genutzt werden. Routinen, die zeitaufwändig oder fehleranfällig sind (etwa das Lösen von Gleichungssystemen oder das Finden von Stammfunktionen), können auch durch ein CAS gelöst werden, um sich einen Überblick zu verschaffen, einen Lösungsansatz zu überprüfen, mit Parametern zu experimentieren oder ...

In Klasse 11 werden grundlegende Polynomeigenschaften wie Grad oder Achsen- und Punktsymmetrie im Kontext von Interpolationsproblemen behandelt. Ist ein Polynom zu vorgegebenen Interpolationspunkten zu finden, so muss zunächst der Grad des Polynoms festgelegt werden, um rechnen zu können. Die Wahl des Polynomgrades führt auf unterschiedlich komplexe Gleichungssysteme (schon bei Polynomgrad 3 kann dies die Lösung eines vierdimensionalen inhomogenen Gleichungssystems sein). Bei ernstgemeinten anwendungsorientierten Aufgaben liegen dann Daten vor, die eine handschriftliche Berechnung des Gleichungssystems unnötig zeitaufwändig und fehleranfällig machen würde. Ein CAS hingegen liefert schnell Lösungen – der Fokus kann auf sinnvolle Fragestellungen wie z. B. die Angemessenheit der Modellierung gerichtet werden. Auch für solche Kontexte gibt es in der Literatur eine Fülle erfolgreich erprobter Unterrichtsbeispiele. Hierbei steht die Nutzung des CAS oft jedoch nicht im Vordergrund der Problembearbeitung, sondern erwächst a posteriori aus dem Problemlöseprozess auf Wunsch des Lernenden. Ein a posteriori-Einsatz des CAS ist jedoch problematisch, wenn sich das CAS auf den Rechnern im Computerraum befindet, denn das Vorhandensein der

Rechner suggeriert die Notwendigkeit der Nutzung (aus welchen Gründen sollte der Unterricht auch sonst im Computerraum stattfinden?). Das CAS stellt hierbei nur eine von vielen Möglichkeiten zur Problemlösung dar: Das CAS steht zur Verfügung, muss aber nicht notwendigerweise hierfür genutzt werden. Der Unterricht ist demzufolge also eher konvergent auf die Lösung eines Problems ausgerichtet und soll in angemessener Weise als weitere Heuristik in das Repertoire der Schüler übergehen.

## Wünschenswertes für die Zukunft

An der Schule werden außer Mathematik noch viele weitere Fächer unterrichtet, in denen der Einsatz computergestützter Medien lohnenswert ist. Sprachprogramme, digitale Bildverarbeitung oder GIS sind nur wenige Beispiele, die in den Sprachwissenschaften, dem Kunst- oder Geographieunterricht gewinnbringend genutzt werden können. Um diese Möglichkeiten in den Unterricht einbringen zu können, wird mittel- bis langfristig eine „Systemlösung“ benötigt. Dies könnten Laptops für Schüler sein, für die entsprechende Software z. B. vom Bildungsträger gekauft werden kann. Umfassende Informationen zu Projekten aus verschiedenen Bundesländern hierzu findet man z. B. unter dem Link <http://www.medienberatung.nrw.de/FachThema/Schule/Laptopklassen/linkliste.htm>.

Die „Laptoplösung“ stößt aber auch auf vielschichtige Kritik. Eine Alternative hierzu wäre die Nutzung von Steckmodulen, die viele Jugendliche für Spielekonsolen nutzen. Der Rechner muss kein Betriebssystem laden – eine Wartung der Geräte entfällt. Wenn man diesbezüglich im Internet sucht, findet man bereits Ansätze, die dieser Idee folgen.

Aus schulischer Sicht ist die Nutzung computergestützter Medien mittelfristig unabdingbar, die Realisierung mit Blick auf die zuvor genannten Argumente jedoch immer noch schwierig.

## Literatur

- [1] Regina Bruder, Wilhelm Weiskirch (Hrsg.), *CaLiMERO – Computeralgebra im Mathematikunterricht*. Band 1-6, Münster (2007-2009)
- [2] Fachgruppe Computeralgebra der DMV, GAMM und GI, *Computeralgebra in Lehre, Ausbildung und Weiterbildung*. Tagungsband V, Ellwangen (2006)
- [3] Frank Förster, Hans-Wolfgang Henn, Jörg Meyer (Hrsg.), *Istron Band 6: Computeranwendungen* (2000)
- [4] MSWWF NRW, *Mathematikunterricht mit Computeralgebrasystemen (CAS) – Handreichungen für die gymnasiale Oberstufe*. Schriftenreihe Schule in NRW, Frechen (2001)

# Funktionales Denken und Analysispropädeutik – Ein Beitrag zu einem qualitativen Einstieg in die Schulanalysis durch Computereinsatz

Andrea Hoffkamp  
Technische Universität Berlin

hoffkamp@math.tu-berlin.de



## Zusammenfassung

„Funktionales Denken beginnt bei intuitiven Vorstellungen über funktionale Zusammenhänge wie „Wenn man die eine Größe ändert, dann ändert sich die andere“ oder „Je mehr..., desto mehr“, und es ist voll entwickelt bei Denkweisen der Analysis“ (Vollrath [7]). Die Realität ist aber ein kalkülorientierter Analysisunterricht mit wenig inhaltlichen Vorstellungen. Deswegen plädieren viele Didaktiker für einen qualitativen Zugang zur Differential- und Integralrechnung. Basierend auf der DGS Cinderella wurden interaktive Lernumgebungen entworfen, die die dynamische Komponente funktionalen Denkens durch eine zweistufige dynamische Visualisierung akzentuieren. Sie zielen darauf ab, inhaltliche Vorstellungen von Änderungsverhalten funktionaler Abhängigkeiten zu entwickeln, bevor das Kalkül im Unterricht entwickelt wird. Im Artikel wird exemplarisch an einer Lernumgebung gezeigt, wie dies aussehen kann.

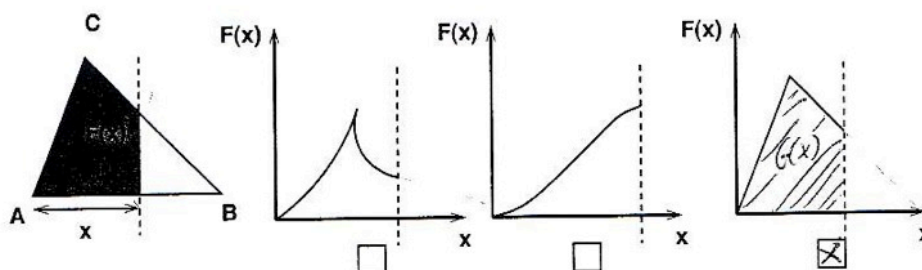
## Funktionales Denken und Analysis

In der Meraner Reform (1905) wurde die „Erziehung zum funktionalen Denken“ als Sonderaufgabe gefordert. Gemeint war ein gebietsübergreifendes Denken in Variationen und funktionalen Abhängigkeiten mit Blick auf Bewegung und Veränderlichkeit. Die Differential- und Integralrechnung sollte nicht aufgesetzter Zusatzstoff, sondern Höhepunkt in einem organisch aufgebauten Mathematikunterricht sein. In diesem Sinne kann die Erziehung zum funktionalen Denken als Propädeutik zur Differential- und Integralrechnung gesehen werden [3]. Die Komplexität

funktionalen Denkens ist einerseits bedingt durch die verschiedenen Darstellungsmöglichkeiten funktionaler Abhängigkeiten (sprachliche Beschreibung, Graph, Formel, Tabelle usw.) und andererseits durch verschiedene Aspekte: Zuordnungsaspekt (punktweise Sicht), Aspekt der Änderung (dynamische Sicht) und Objektaspekt (Sicht auf Funktion als Ganzes) [7].

Insbesondere der Aspekt der Änderung und der Objektaspekt bereiten Schülerinnen und Schülern Schwierigkeiten. Das äußert sich beispielsweise darin, dass Funktionsgraphen als photographische Bilder von Realsituationen interpretiert werden (**Graph-als-Bild-Fehler**).

Die gestrichelte Linie wird vom Punkte A um die Entfernung  $x$  nach rechts gezogen. Der Wert  $F(x)$  gibt die Größe der grau unterlegten Fläche an. Welcher Graph passt? Begründe Deine Wahl!



Begründen Sie Ihre Wahl:

Die Fläche  $G(x)$  ist so, wie die Fläche  $F(x)$

Typischer Graph-als-Bild-Fehler



Betrachtet man den Schulunterricht in der Sekundarstufe I (bis Klasse 10), so ergibt sich folgendes Bild: Der Zuordnungsaspekt funktionaler Abhängigkeiten ist überbetont, was sich z. B. darin äußert, dass der Weg über eine Wertetabelle die vorherrschende Technik ist, wenn es um die Darstellung von Funktionen geht. Will man aber lokale und globale Eigenschaften, wie Symmetrie, Monotonie, Wendestellen von Funktionen beschreiben oder sich einen Überblick über charakteristische Momente des Änderungsverhaltens verschaffen, so sind Tabellen denkbar ungeeignet. Die Realität des Analysisunterrichts der Sekundarstufe II ist geprägt von Kalkülorientierung und Verharren in Berechnungen losgelöst von inhaltlichen Vorstellungen. Viele Didaktiker plädieren deswegen immer wieder für eine stärkere Gewichtung der qualitativen Anfänge der Analysis (z. B. [2], [5]).

## Lernumgebung „Dreiecksfläche“

Basierend auf der DGS Cinderella und unter Ausnutzung der integrierten funktionalen Programmiersprache *CindyScript* wurde die hier vorgestellte interaktive Lernumgebung „Dreiecksfläche“ entwickelt. Sie ist zusammen mit Lehrmaterial unter [1] verfügbar. Zur Nutzung der Lernumgebungen im Unterricht genügt ein Standardinternetbrowser. Spezielles Wissen zur Funktionsweise der Software ist nicht nötig (*geringer technischer Overhead*). Die Grundidee ist eine interaktiv-experimentelle Computernutzung mit dem Ziel, die

dynamische Komponente funktionalen Denkens hervorzuheben und inhaltliche Vorstellungen im Hinblick auf Propädeutik zur Differential- und Integralrechnung zu entwickeln. In der Lernumgebung sollen die Schülerinnen und Schüler den funktionalen Zusammenhang zwischen Abstand  $AD$  und Flächeninhalt des dunkelblauen Flächenanteils des Dreiecks untersuchen. Insbesondere soll das Änderungsverhalten charakterisiert werden und die Wendestelle als charakteristisches Moment, in dem sich die Qualität des Wachstums ändert, wahrgenommen werden. Der mathematische Hintergrund basiert auf dem Hauptsatz der Differential- und Integralrechnung. Der Graph zeigt den Bestand an Flächeninhalt in Abhängigkeit vom Abstand  $AD$ . Das Dreieck – als stückweise lineare Funktion interpretiert – zeigt den Änderungsgraphen, also die Ableitungsfunktion. Mit dem Kalkül ist die Wendestelle nicht zu finden, da die Ableitung nicht differenzierbar ist.

Folgende Gestaltungsleitlinien liegen hierbei zugrunde:

**Verknüpfung Situation–Graph:** Anknüpfend an inhaltliche Vorstellungen ist der Ausgangspunkt ein funktionaler Zusammenhang innerhalb einer Situation und deren dynamische Verknüpfung mit der Darstellungsform Graph. Die graphische Darstellung wurde gewählt, weil sie sich besonders auf die dynamische Komponente funktionalen Denkens bezieht und die gesamte Information sowie lokale und globale Eigenschaften der Funktion „auf einen Blick“ enthält.

Dreiecksfläche

Mit der Abbildung kannst Du ausprobieren, wie sich der Flächeninhalt der blauen Fläche verändert, wenn man den Abstand von A zu D verändert. Klicke dazu mit der Maus auf den Punkt D und halte die Taste gedrückt während Du D bewegst. Der Graph zeigt Dir die Größe des Flächeninhalts in Abhängigkeit von der Lage von D.

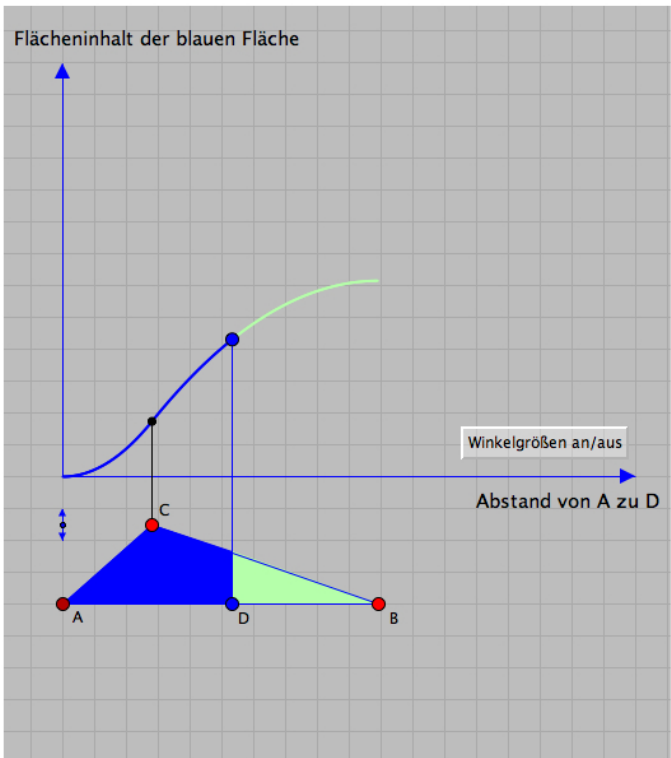
**Aufgaben:**

- Beschreibe die Form des Graphen möglichst genau!
- Warum hat der Graph diese Gestalt?
- Was geschieht oder ändert sich am schwarzen Punkt über C?

Zur Beantwortung der Fragen kannst Du auch die Form des Dreiecks verändern, indem Du die Punkte B und C nach rechts oder links schiebst. C kannst Du mit Hilfe des blauen Schiebereglers nach oben und unten bewegen.

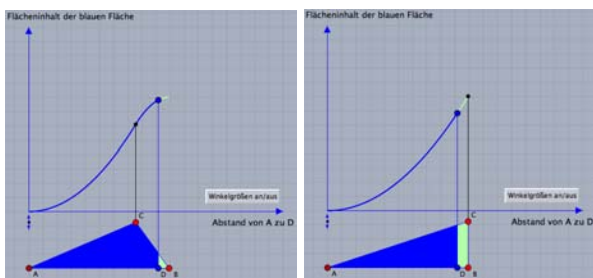
Beantworte dazu folgende Fragen:

- Wie sieht der Graph aus, wenn C direkt über A oder B liegt?
- Wie hängt die Form des Graphen von den Winkeln des Dreiecks ab? (Z.B. wenn der Winkel  $\alpha$  größer ist als  $\beta$  oder beide Winkel gleich groß.)
- Was für Graphen kannst Du erzeugen?



© 2008, Andreas Fest und Andrea Hoffkamp, Technische Universität Berlin  
 Created with Cinderella

**Zwei Variationsstufen:** Bewegung des Punktes  $D$  bedeutet **Variation innerhalb der Situation** und somit simultane Visualisierung des Änderungsaspektes in Situation und Graph. Dahinter steht die Idee der **Supplantation** [6]: Soll ein Lernender einen dynamischen Sachverhalt verstehen, so muss er ein „lauffähiges“ mentales Modell konstruieren, um mentale Simulation zu erreichen. Supplantation meint die visuelle Unterstützung mentaler Simulationsprozesse. Monotonie lässt sich nun darin ausdrücken, dass „immer mehr blau dazu kommt“. Eine Beschreibung der Charakteristik der Wendestelle als Stelle, bis zu der die Zunahme ansteigt und danach die Zunahme fällt, ist für Schülerinnen und Schüler schwieriger. Eine zweite Variationsstufe – im weiteren Verlauf **Metavariation** genannt – erlaubt nun durch Bewegung der Punkte  $B$  und  $C$  das Verändern der Situation und damit der Funktion als Ganzes. Metavariation bedeutet Variation innerhalb der Funktion, die der Situation den Flächeninhaltsgraphen zuordnet – also Variation innerhalb des Integraloperators. Der Flächeninhaltsgraph ist ein Bild unter der Metafunktion Integraloperator.



Somit bezieht sich Metavariation insbesondere auf den Objektaspekt funktionaler Zusammenhänge. Tatsächlich hängen Objektsicht und Änderungssicht von Funktionen eng zusammen und lassen sich nur theoretisch trennen. Will man globale (Objekt-) Eigenschaften wie Monotonie beschreiben, so benutzt man die „Sprache des Änderungsaspektes“: Ist  $x \leq y$  so auch  $f(x) \leq f(y)$  für alle  $x, y$ . Metavariation ermöglicht aber auch die Loslösung von konkreten Werten und legt den

Schwerpunkt auf qualitative Betrachtungen. Charakterisierende Eigenschaften der Flächeninhaltsfunktion werden hervorgehoben: So ist Monotonie invariant unter Metavariation, die Wendestelle als charakteristischer Moment im Änderungsverhalten ist „beinahe invariant“. Das führt zu Erklärungszwängen und zu Schüleräußerungen wie: „Nach dem schwarzen Punkt über  $C$  nimmt der zu addierende Flächeninhalt ab.“ Aber auch Begriffe wie „konvex“ und „konkav“ (falls  $C$  über  $B$  bzw.  $A$  liegt) werden genannt und können reflektiert werden.

Weiteres Material zum Thema und Literaturhinweise zu Publikationen der Autorin über den Einsatz im Unterricht und lerntheoretische Fundierung finden sich unter <http://www.math.tu-berlin.de/~hoffkamp>.

## Literatur

- [1] A. Fest, A. Hoffkamp (2008): *Interaktive Lernumgebung „Dreiecksfläche“ und Lehrmaterial*. <http://www.math.tu-berlin.de/~hoffkamp/Material/Dreieck>, <http://www.math.tu-berlin.de/~hoffkamp/Material/dreieckmaterial.html>
- [2] S. Hahn, S. Prediger (2008): *Bestand und Änderung – Ein Beitrag zur Didaktischen Rekonstruktion der Analysis*. *Journal für Mathematikdidaktik* 29 (3/4), S. 163–198.
- [3] K. Krüger (2000): *Kinematisch-funktionales Denken als Ziel des höheren Mathematikunterrichts – das Scheitern der Meraner Reform*. *Mathematische Semesterberichte* 47, S. 221–241.
- [4] J. Richter-Geibert, U. Kortenkamp (2006): *Math in Motion – The Interactive Geometry Software Cinderella*. Version 2.0. <http://cinderella.de>
- [5] H. Stellmacher (1986): *Die nichtquantitative Beschreibung von Funktionen durch Graphen beim Einführungsunterricht*. In: von Harten, Jahnke et al. (Hrsg.): *Funktionsbegriff und funktionales Denken* IDM-Reihe, Band 11, S. 21–34. Aulis Verlag, Köln.
- [6] M. Vogel (2006): *Mathematisieren funktionaler Zusammenhänge mit multimedialbasierter Supplantation*. Franzbecker, Hildesheim.
- [7] H.-J. Vollrath (1989): *Funktionales Denken*. *Journal für Mathematikdidaktik* 10(1), S. 3–37.

**mathemas ordinate**  [www.ordinate.de](http://www.ordinate.de)

☎ 0431 23745-00/ ☒ -01, [info@ordinate.de](mailto:info@ordinate.de) → Software for mathematical people !

 **Mathematica, ExtendSim,**

**MathType, KaleidaGraph, Fortran, NSBasic, @Risk**

**und a.m.**

mathemas ordinate, Dipl. Math. Carsten Herrmann, M. Sc.  
Königsbergerstr. 97, 24161 Altenholz

Mehr als 20 Jahre Erfahrung mit Software-Distribution !

$$\infty + \mu < \heartsuit$$

$$\int_{x_1}^{x_2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx$$





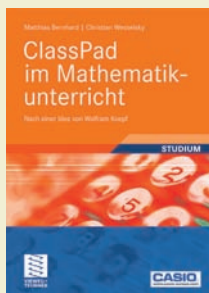
## Publikationen über Computeralgebra

- Bernhard, M., Wesselsky, C., *ClassPad im Mathematikunterricht. Nach einer Idee von Wolfram Koepf*, Vieweg+Teubner, Wiesbaden, 2009, 185 Seiten, ISBN 978-3-8348-0840-0, € 19,90.
- Joswig, M., Theobald, T., *Algorithmische Geometrie*, Vieweg+Teubner, Wiesbaden, 2008, 265 Seiten, ISBN 978-3-8348-0281-1, € 31,50. (Eine Besprechung finden Sie auf Seite 32.)
- Kügler, P., Windsteiger, W., *Algorithmische Methoden. Zahlen, Vektoren, Polynome*, Birkhäuser Verlag, Basel, Boston, 2009, 160 Seiten, ISBN 978-3-7643-8434-0, € 18,90. (Eine Besprechung finden Sie auf Seite 33.)
- Sala, M., Mora, T., Perret, L., Sakata, S., Traverso, C. (Eds.), *Gröbner Bases, Coding, and Cryptography*, Springer Verlag, Berlin, Heidelberg, New York, 2009, 430 Seiten, ISBN 978-3-540-93805-7, \$ 129,00.
- Sturmfels, B., *Algorithms in Invariant Theory*, 2nd edition, Springer Verlag, Berlin, Heidelberg, New York, 2008, 197 Seiten, ISBN 978-3-2117-7416-8, € 42,75.
- Westermann, T., *Mathematik für Ingenieure*, Springer Verlag, Berlin, Heidelberg, New York, 2008, 656 Seiten, ISBN 978-3-540-77730-4, € 39,95. (Eine Besprechung finden Sie auf Seite 34.)

Weitere Bücher können auf der Seite <http://www.fachgruppe-computeralgebra.de/Buecher> oder direkt bei Eva Zerz ([eva.zerz@math.rwth-aachen.de](mailto:eva.zerz@math.rwth-aachen.de)) zur Besprechung angefordert werden.

WWW.VIEWEGTEUBNER.DE

## Unterrichtshilfe CAS Taschenrechner



Matthias Bernhard | Christian Wesselsky

**ClassPad im Mathematikunterricht**  
Nach einer Idee von Wolfram Koepf

2009. Mit CD (Arbeitsmaterialien mit Lösungen). X, 185 S. Br. EUR 19,90  
ISBN 978-3-8348-0840-0

In diesem Buch werden mathematische Problemstellungen aus der Geometrie, Algebra und Analysis mithilfe des ClassPad CAS Taschenrechners von Casio bearbeitet. Es orientiert sich am Schulstoff der Sekundarstufe II und richtet sich primär an Mathematiklehrer, die ClassPad als sinnvolle Erweiterung ihres Mathematikunterrichts nutzen wollen.

Aufgeteilt in kurze Übungen und veranschaulicht durch zahlreiche Abbildungen und Display-Screenshots des ClassPad werden ausgewählte, komplexe mathematische Probleme schrittweise erarbeitet. Anschließend gibt es zu jedem Abschnitt einen Aufgabenteil mit Lösungen.

Alle Rechenschritte finden sich auf einer beiliegenden CD, die zudem auch einige neu programmierte Funktionalitäten in ClassPad enthält.

### Autoren

Matthias Bernhard, Department Mathematik, Universität Erlangen-Nürnberg  
Christian Wesselsky, Department Mathematik, Universität Erlangen-Nürnberg  
Prof. Dr. Wolfram Koepf, Mathematik / Computational Mathematics, Universität Kassel

Einfach bestellen:  
[buch@viewegteubner.de](mailto:buch@viewegteubner.de) Telefax +49(0)611. 7878-420

TECHNIK BEWEGT.



### V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (Eds.) **Computer Algebra in Scientific Computing (CASC 2007)**

Lecture Notes in Computer Science 4770, Springer Verlag, 2007, 460 Seiten, ISBN 978-3-540-75186-1, € 64,20

Dieses Buch enthält die Proceedings des *10. International Workshop Computer Algebra in Scientific Computing (CASC'2007)*, der im September 2007 in Bonn stattfand. Es ist Vladimir P. Gerdt gewidmet, einem der Initiatoren der CASC-Konferenzen, dessen 60. Geburtstag mit der Tagung gefeiert wurde. Die CASC-Konferenzen dienen von Anfang an insbesondere auch der Ost-West-Kommunikation, und sie haben daher auch immer einen besonders hohen Anteil an Teilnehmern aus Staaten der früheren Sowjetunion.

Thematisch decken die Beiträge des Bandes einen weiten Bereich ab. Wie bei den meisten CASC-Tagungen ist ein klarer Schwerpunkt bei Differential- und Differenzgleichungen bzw. Dynamischen Systemen mit Anwendungen in der Physik zu erkennen. Neben der Be-

rechnung symbolischer Lösungen geht es hier auch oft um die Herleitung und Analyse numerischer Verfahren. Aber auch klassische Computeralgebrathemen wie Polynomarithmetik, Faktorisierung, Resultanten, Gröbner-Basen und Quantorenelimination kommen in dem Band nicht zu kurz. Von den beiden eingeladenen Hauptvorträgen liegen leider nur Abstrakte vor.

Das gesamte Programm der Tagung ist auf der Webseite <http://www14.in.tum.de/konferenzen/CASC2007/program.html> zu finden, die Online-Version der Proceedings (LNCS 4770) steht auf <http://www.springerlink.com/content/978-3-540-75186-1/> zur Verfügung.

Werner M. Seiler (Kassel)

### M. Joswig, T. Theobald **Algorithmische Geometrie**

Vieweg Verlag, 2008, 265 Seiten, ISBN 978-3-8348-0281-1, € 31,50

Algorithmische Fragen in der Geometrie erhielten in den letzten Jahrzehnten vor allem durch Anwendungen in der Computergraphik oder bei Optimierungsproblemen eine große Aufmerksamkeit. Das vorliegende Lehrbuch von Joswig und Theobald möchte auf einem auch für Bachelor-Studiengänge geeigneten Niveau in dieses Gebiet einführen.

Das Buch gliedert sich in drei Teile. Der erste behandelt die lineare algorithmische Geometrie. Er beinhaltet zunächst eine kurze Einführung in die projektive Geometrie (die an vielen Stellen im Buch benutzt wird) und studiert dann ausführlicher Polytope und Polyeder. An algorithmischen Fragestellungen werden die lineare Optimierung mit dem Simplex-Algorithmus sowie die Berechnung von konvexen Hüllen, Voronoi-Diagrammen und Delone-Zerlegungen diskutiert. Der zweite Teil beschäftigt sich mit der nichtlinearen algorithmischen Geometrie, was hier vor allem das Lösen polynomialer Gleichungssysteme mit Gröbnerbasen bedeutet. Dazu werden zunächst die benötigten Grundbegriffe aus kommutativer Algebra und algebraischer Geometrie eingeführt. Dann werden Gröbnerbasen und deren Berechnung mit dem Buchberger-Algorithmus diskutiert. Das Lösen erfolgt schließlich mit Hilfe von

Eliminationsidealen (und vereinzelt auch Resultantenmethoden). Im dritten Teil werden weiterführende Anwendungen angeschnitten. Hier wird als erstes das Problem der Kurvenrekonstruktion studiert. Dann werden einige Fragen zu Geradenkonfigurationen mit Hilfe von Plücker-Koordinaten behandelt. Den Abschluss bilden einige Anwendungen der nichtlinearen Geometrie, wie das direkte kinematische Problem bei Stewart-Plattformen oder das Global Positioning System (GPS).

Laut Klappentext ist das Buch für Studierende ab dem 4. Semester gedacht. Zumindest der erste Teil kann aber sicherlich auch schon im 3. Semester benutzt werden. Der zweite Teil setzt jedoch eine Einführungsvorlesung in Algebra voraus, wie sie typischerweise erst im 3. Semester gehört wird. Die Autoren geben sich große Mühe, alle Beweise auf einem dieser Studienphase angemessenen Niveau zu führen sowie alle Konstruktionen durch nachvollziehbare Anwendungen zu motivieren.

Wie es sich für ein ordentliches Lehrbuch gehört, werden alle Kapitel durch Übungsaufgaben ergänzt sowie kurzen „Anmerkungen“, in denen u. A. auf weiterführende Literatur oder verfügbare Software verwiesen wird. An vielen Stellen werden konkrete

Berechnungen in Systemen wie POLYMAKE, SINGULAR oder MAPLE vorgeführt.

Kritisch lässt sich nur wenig anmerken. Wünschenswert wäre ein weiterer Anhang, in dem die verwendeten Konzepte aus der Topologie kurz eingeführt werden, da gerade bei Bachelor-Studierenden nicht unbedingt vorausgesetzt werden kann, dass sie eine Topologie-Vorlesung besucht haben. Da die Autoren verschiedentlich auf Komplexitätsfragen eingehen, verwundert es, dass nicht einmal in den Anmerkungen zum Buchberger-Algorithmus etwas über die *praktische* Komplexität des Algorithmus zu finden ist.

An einigen wenigen Stellen haben sich auch kleine Fehler eingeschlichen, die sich aber leicht korrigieren lassen. Aufgrund eines Fehlers des Verlags ist übrigens der größte Teil der Auflage ohne Inhaltsverzeichnis erschienen; die Autoren stellen auf der Webseite <http://www.mathematik.tu-darmstadt.de/~joswig/algo/index.html> einen Ersatz in Form einer PDF-Datei zur Verfügung.

Insgesamt handelt es sich aber um ein sehr gelungenes und gut lesbares Lehrbuch, das sich in Bachelor-Studiengängen vielfältig einsetzen lässt.

Werner M. Seiler (Kassel)

## **P. Kügler, W. Windsteiger** **Algorithmische Methoden. Zahlen, Vektoren, Polynome**

Birkhäuser Verlag, 2009, 160 Seiten, ISBN 978-3-7643-8434-0, € 18,90

Das vorliegende Buch ist als einsemestrige Einführung in die algorithmische Mathematik konzipiert, gerichtet an Studierende vorwiegend des dritten Semesters. Kenntnisse über die Grundstrukturen von Analysis und linearer Algebra (speziell Zahlen, Vektoren, Polynome) werden also vorausgesetzt – eine kurze Übersicht des benötigten Stoffs findet sich gleichwohl zu Beginn des jeweiligen Kapitels – und weite Teile des Buches sind mit der Implementation dieser Strukturen und der zugehörigen Rechenoperationen auf dem Computer befasst.

Das Buch umfasst vier Kapitel. Das erste Kapitel führt die Grundbegriffe des Buches ein, auf eher informelle Weise und anhand von Beispielen, die dann in den folgenden Kapiteln wiederholt aufgegriffen werden. Bei diesen Begriffen handelt es sich einerseits um Grundkonzepte der Informatik (Schleifen, Rekursion), und andererseits um Kriterien für die Bewertung von Algorithmen, zum Beispiel (relative und absolute) Rundungsfehler, die Kondition eines Problems, Korrektheit, Stabilität (vorwärts und rückwärts), Komplexität. Die im weiteren Verlauf des Buches entwickelten Algorithmen werden anhand dieser Qualitätsmerkmale analysiert und Weiterentwicklungen daraus motiviert. Wie an den aufgezählten Kriterien zu erkennen ist, liegt der Schwerpunkt der Analyse auf numerischen Aspekten.

Die algorithmische Umsetzung wird in einem recht intuitiven Pseudocode dokumentiert. Sämtliche im Buch beschriebenen Algorithmen wurden sowohl in MATLAB als auch in *Mathematica* implementiert (als Vertreter vorwiegend numerischer beziehungsweise symbolischer Mathematik-Pakete); die Implementationen sind frei über das Internet verfügbar.

Kapitel 2 behandelt die algorithmische Behandlung von Zahlbereichen, von natürlichen über rationale zu den reellen Zahlen. Es werden jeweils das Pro-

blem der Darstellung dieser Zahlen im Rechner und die mittlerweile gebräuchlichen (IEEE-)Standardlösungen dafür diskutiert, gefolgt von der Implementation der Grundrechenarten. Dabei werden neben den naheliegenden auch effizientere Lösungen präsentiert (etwa: der Karatsuba-Algorithmus zur Multiplikation ganzer Zahlen oder Henrici-Algorithmen für die Addition und Multiplikation in  $\mathbb{Q}$ ).

Kapitel 3 ist mit der Übertragung der Grundrechenarten auf Vektoren – genauer: Elemente von  $\mathbb{R}^d$  – befasst. Die Implementation des Gram-Schmidt-Orthonormalisierungsverfahrens mitsamt einer Analyse des Stabilitätsverhaltens schließt das Kapitel ab. Das letzte Kapitel behandelt Polynome, und dort speziell Algorithmen zur Polynomdivision mit Rest, den euklidischen Algorithmus, Polynomauswertung und -interpolation. Am Ende jedes Kapitels finden sich Übungsaufgaben zur Vertiefung des Stoffes.

Die Autoren sind sichtlich um eine lebendige und problemorientierte Darstellung ihres Stoffes bemüht. Der Großteil der im ersten Kapitel entwickelten Kriterien beschäftigt sich mit dem Grundproblem der numerischen Mathematik, nämlich der approximativen Behandlung (potenziell) unendlicher mathematischer Objekte mit endlichen Speicherplatz- und Zeitressourcen. Die Darstellung in diesem Kapitel springt zeitweise munter zwischen algorithmischen Konzepten (Schleifen, Rekursion, etc.), numerischen Begriffen (Fehlerabschätzungen, Maschinengenauigkeit) und mathematischen Sätzen, was zumindest beim Rezensenten eine gewisse Desorientierung zur Folge hatte.

Das Bemühen, die nötigen Begriffe anhand möglichst konkreter Beispiele einzuführen, ist hier erkennbar und an einigen Stellen auch durchaus erfolgreich. Zumindest im ersten Kapitel aber scheint mir dies durch einen Mangel an Struktur und Kohärenz (zu)

teuer erkaufte. Eine weitere in meinen Augen problematische strukturelle Entscheidung der Autoren betrifft die Darstellung der reellen Zahlen im Computer, die erst im zweiten Kapitel ausführlich behandelt wird. Viele der in Kapitel 1 entwickelten numerischen Begriffe lassen sich aber meines Erachtens wesentlich präziser beschreiben

und motivieren, wenn man explizit auf diese Darstellung verweisen kann; zudem wäre der Leser dann bereits hinreichend für die zugrundeliegende Problematik sensibilisiert.

Hartmut Führ (Aachen)

## **T. Westermann**

### **Mathematik für Ingenieure**

Springer Verlag, 2008, 656 Seiten, ISBN 978-3-540-77730-4, € 39,95

Im 3. Jahrtausend ist die Einbindung von Computer-Software in die Lösung mathematischer Aufgaben nicht mehr wegzudenken. Eine Kombination von Lehrbuchtext, Programmierhilfen und visuellen Animationen mit Maple ist daher sicherlich eine sehr wünschenswerte Hilfe für Studierende der Ingenieurwissenschaften. Hier schließt die Springer-Publikation „Mathematik für Ingenieure“ von Thomas Westermann eine Bedarfslücke früherer Jahre. Die auf dem Cover zu lesende Betonung, dies sei ein anwendungsorientiertes Lehrbuch, sollte allerdings keine überzogenen Hoffnungen beim anwendenden Ingenieur wecken: Aufgaben aus dem „echten“ Anwendungsbereich der Ingenieurwissenschaften sind in der deutlichen Minderzahl gegenüber den rein mathematischen Übungsbeispielen.

Der Anwendungsbezug liegt offensichtlich in den mit Maple erstellten Worksheets. Er entbindet jedoch nicht von mathematischer Korrektheit und, soweit auf dem für Ingenieure zumutbaren Level möglich, auch nicht von ausreichender Präzision in der Formulierung notwendiger Definitionen und Sätze. In dem vorliegenden Buch werden leider wichtige Begriffe (z. B. lineare Abbildungen) gar nicht erklärt oder missverständlich bis falsch beschrieben („Eine Basis ist die kleinste Menge von Vektoren, welche den Vektorraum erzeugt“!). Zahlreiche Mängel in den Notationen und Definitionen sowie vermeidbare Ungenauigkeiten bei Argumen-

tationen hinterlassen ebenso wenig wie Druckfehler und grammatikalische Unzulänglichkeiten einen positiven Eindruck. Schade auch, dass Sätze fast ausnahmslos ohne jede Begründung lapidar in den Raum gestellt werden. Nicht nur Grundlagenforscher, sondern auch moderne Dozenten der Ingenieurwissenschaften verlangen eine gewisse Fundierung der benutzten mathematischen Methoden. Ein Vorbild exakter, zugleich spannender und wirklich anwendungsorientierter Darstellung ist hier immer noch die im gleichen Verlag vor mehr als zwei Jahrzehnten erschienene „Höhere Mathematik“ von Meyberg und Vachenaier.

Im Gegensatz zu den Schwächen im reinen Lehrtext verwirklicht die Publikation von Westermann in vorbildlicher Weise das Konzept der Verbindung von Theorie und Praxis in Form der Maple-Worksheets. Der Autor hat mit Gründlichkeit die den jeweiligen Kapiteln entsprechenden Arbeitsblätter konzipiert und diese in einer auch für Studienanfänger leicht nachvollziehbaren Form dargestellt. Besonders hilfreich sind die im Rahmen kleiner Projekte gestalteten Aufgabenpakete.

Fazit: Eine große Hilfe für alle, die Ingenieuraufgaben mit Hilfe von Computer-Software behandeln wollen. Die Darstellung der mathematischen Inhalte ist anderen Autoren besser gelungen.

Marcel Erné (Hannover)



### 1. CAPP – Computer Algebra and Particle Physics 2009

Berlin, 29.03. – 03.04.2009

<https://indico.desy.de/conferenceDisplay.py?confId=1573>

Die CAPP wird seit 2005 alle zwei Jahre am DESY Zeuthen abgehalten. Der Hauptschwerpunkt liegt zum einen darin, den Teilnehmern Grundlagen von Algebraprogrammen wie MATHEMATICA oder FORM zu erklären, und zum anderen darin, Packages und Programme, welche in der Teilchenphysik von hoher Relevanz sind, vorzustellen.

Essentiell war, dass jedem der ca. 30 Teilnehmenden ein eigener Rechner zu Verfügung stand. Viele Vorträge waren darauf ausgelegt, dass man kleine Beispielprogramme schreiben bzw. die vorgestellten Tools direkt ausprobieren konnte. So wurden z. B. zu dem Vortrag zur Einführung in FORM lauter Beispielprogramme zur Verfügung gestellt, an denen man die Funktionsweise der einzelnen, neu erlernten Befehle direkt ausprobieren konnte.

Die Tragweite von Algebraprogrammen in der Physik wurde durch mehrere Vorträge deutlich, in denen auf die explizite Anwendung und Benutzung von Packages wie z. B. FORMCALC oder MATAD, welche in MATHEMATICA bzw. in FORM geschrieben sind, eingegangen wurde.

Auf die für Phänomenologen wichtigen Monte-Carlo-Programme zur Eventsimulation am LHC wurde in separaten Sessions eingegangen. Dabei wurden sowohl Hinweise zur effektiven Programmierung gegeben, als auch das Monte-Carlo-Programm HELAC vorgestellt.

Jan Germer (MPI München)

### 2. Tagung der Fachgruppe Computeralgebra

Kassel, 14. – 16.05.2009

<http://www.fachgruppe-computeralgebra.de>



Auf Seite 6 finden Sie einen ausführlichen Bericht zu dieser Tagung.

### 3. CoCoA 2009 – International School on Computer Algebra

Barcelona, Spanien, 08. – 12.06.2009

<http://cocoa.dima.unige.it/conference/cocoa2009/>

In Barcelona fand vom 08. bis zum 12.6. die mittlerweile sechste Ausgabe der überaus erfolgreichen CoCoA-Schulen statt. Diese richten sich an internationale Diplomanden und Doktoranden, die für ihre Arbeit Computeralgebra benötigen

oder erlernen wollen. Die diesjährige Ausgabe hatte 22 Teilnehmer, die an den folgenden beiden Kursen teilnahmen:

Marilina Rossi (Genua/Italien): *On Castelnuovo regularity and related problems*

Anthony V. Geramita (Kingston/Kanada): *Secant varieties*

Nach zwei Vorträgen am Vormittag gab es jeweils zugehörige Computerübungen am Nachmittag, die mit dem Computeralgebrasystem CoCoA zu bearbeiten waren. Sie wurden von Anna Bigatti, Eduardo de Cabezón und Enrico Carlini sachkundig geleitet. Die Kursmaterialien und die Übungsaufgaben stehen auf den Webseiten der Schule zum Download bereit.

Die entspannte Atmosphäre und die ansteckende Begeisterung der beiden Kursleiter trugen zu einer sehr erfolgreichen Schule bei, die von den Teilnehmern ausnahmslos positiv kommentiert wurde. Die Schule wurde von der Firma Shell International Exploration and Production (Rijswijk/NLD) finanziell unterstützt. Für 2011 ist eine Fortsetzung der Serie in Passau geplant.

Martin Kreuzer (Universität Passau)

### 4. Conference on Computational Commutative Algebra

Barcelona, Spanien, 12. – 13.06.2009

<http://cocoa.dima.unige.it/conference/robbiano65/>



Lorenzo Robbiano

Im Anschluss an die CoCoA-Schule und vor der MEGA-Konferenz fand an der Universität Barcelona eine kurze Tagung zu Ehren des 65. Geburtstags von Lorenzo Robbiano statt. Die große Aktivität und hohe Bekanntheit Robbianos spiegelte sich in einer beachtlichen Liste von 33 Teilnehmern wider.

Den ersten Hauptvortrag hielt kein geringerer als Bruno Buchberger (Linz). Sein Titel „How to replace senior mathematicians by machines“ reflektiert einen weiteren Aspekt von Robbianos Schaffen: die Präsentation auch schwierigster mathematischer Inhalte auf humorvolle und verständliche Art. Weitere Vorträge wie „CoCoATeamTime“ (durch das CoCoA Team), „Bisectors Cha-Cha-Cha“ (Tomas Recio), „How to write THE BOOK“ (der Berichtersteller) oder „How to do Gotzmann and Hilbert function? Coefficiently!“ (Anthony Geramita) basierten auf dem selben Prinzip. Ein Galadinner mit einer Tanzeinlage des Geehrten rundete die Veranstaltung ab.

Martin Kreuzer (Universität Passau)

## 5. Summer School on Computer Algebra and Syzygies

Sophus-Lie-Konferenzzentrum, Nordfjordeid, Norwegen, 15. – 19.06.2009

<http://www.math.uio.no/div/nordfjordeid/nordfjord.html>

Die jährlichen Sommerschulen am Sophus Lie Center im idyllischen Nordfjordeid (Norwegen) haben bereits eine lange Tradition, die bis 1996 zurück reicht. Die aktuelle Ausgabe vom 15.-19.06.2009 behandelte Themen aus der Computeralgebra. Es fanden drei Kurse statt:

1. Mats Boij (Stockholm/Schweden): *Graded Betti numbers*
2. Takayuki Hibi (Osaka/Japan): *Combinatorial techniques in commutative algebra*
3. Martin Kreuzer (Passau): *Computations with Gröbner bases in applications*



Tagungsfoto aus Nordfjordeid

Jeder Kurs bestand aus sieben einstündigen Vorträgen und jeweils zugehörigen Aufgabenblättern und Übungsstunden. Die Programmieraufgaben waren mit dem Computeralgebrasystem ApCoCoA zu bearbeiten. Am Ende gab es auch eine *Open Problem Session* mit Hinweisen auf mögliche Themen für weitere Forschungen. Mit 30 Teilnehmern aus der ganzen Welt, wobei der Schwerpunkt naturgemäß auf den nordischen Ländern lag, war die Schule sehr gut besucht. Die Organisatoren Gunnar Fløystad (Bergen), Kristian Ranestad (Oslo), Trygve Johnsen (Tromsø) und Sverre Smaløe (Trondheim) leisteten hervorragende Arbeit und boten allen Teilnehmern eine angenehme und produktive

Atmosphäre. Ein Bus- und Wanderausflug zum Briksdale-Gletscher ergänzte das wissenschaftliche Programm. Es bleibt nur zu hoffen, dass auch in Zukunft wieder einmal ein Thema aus der Computeralgebra gewählt wird, da man die Schulen am Nordfjord geeigneten Doktoranden nur empfehlen kann.

Martin Kreuzer (Universität Passau)

## 6. MEGA 2009 – Effective Methods in Algebraic Geometry

Barcelona, Spanien, 15. – 19.06.2009

<http://www.imub.ub.es/mega09/>

Die Tagung MEGA 2009 (Effektive Methoden in der Algebraischen Geometrie) war bereits die 10. Ausgabe dieser erfolgreichen zweijährlichen Tagung. Sie fand vom 15. bis 19. Juni in Barcelona statt. Die Organisatoren um L. M. Pardo (Santander) haben keine Mühen gescheut, und die Tagung verlief auf hohem Niveau. Es gab keine parallelen Sektionen, d. h. die Vorträge wurden alle in einer gemeinsamen Sitzung gehalten und fanden im zentral gelegenen (und u. A. mit klassizistischen Gemälden dekorierten) Hauptgebäude der mathematischen Fakultät der Universität Barcelona statt.

Es gab vier Arten von Vorträgen: einstündige Hauptvorträge (von M. Casanellas, J. Draisma, M. Husty, J. Landsberg, G. Lecerf, M. Shub, W. Stein und R. Vakil), eingereichte Vorträge à 35 Minuten, Forschungsberichte à 25 Minuten und dazu noch fünf Software-Präsentationen. Die Themen der Vorträge umfassten algebraische Geometrie, Computeralgebra, Differentialgleichungen, Gruppentheorie und deren Anwendungen. Es wurde beschlossen, dass die MEGA 2011 in Stockholm stattfinden soll.

Viktor Levandovskyy (RWTH Aachen)

## 7. CALC 2009 — Helmholtz International School – Workshop on Calculations for Modern and Future Colliders

Dubna, Russland, 10. – 20.07.2009

<http://theor.jinr.ru/~calc2009>

Thema dieser zehntägigen Schule/Workshop waren Rechenmethoden für die Präzisions-Physik an Teilchenbeschleunigern. Dabei gab es einerseits allgemeinere Vorlesungen, die sich mit den physikalischen Grundlagen und der Phänomenologie der Hochenergiephysik befassten und sich in erster Linie an Studenten richteten. Andererseits wurden verschiedene Methoden und Programmpakete vorgestellt sowie Statusberichte und Ergebnisse einiger Gruppen präsentiert. Teilweise waren diese dem Workshop-Teil der Veranstaltung zuzurechnenden Vorträge jedoch sehr speziell und eher an die Experten im Auditorium adressiert.



Tagungsfoto CALC 2009

Ein Schwerpunkt im Bereich der Computeralgebra waren Methoden zur Berechnung von Multi-Loop- und Multi-Leg-Prozessen – eine Aufgabe, welche ohne den Einsatz von CAS nahezu unmöglich wäre. Hierbei wurde das gesamte Spektrum gängiger Methoden abgedeckt, u. A. Unitaritätsmethoden, On-shell-Rekursion, Integration-by-parts-Relationen und Mellin-Barnes-Darstellung von Integralen. Oft gab es dazu auch ein dazugehöriges (öffentliches) Programmpaket. Weiter wurden einige Pakete vorgestellt, die es dem Anwender erlauben, vollautomatisiert Prozesse der Hochenergiephysik in führender (CalcHEP) und nächstführender Ordnung (FeynArts/FormCalc, SANC) zu berechnen.

Das Rahmenprogramm mit einer Exkursion ins etwa 100 km entfernte Moskau und einer Bootsfahrt auf der Wolga mit anschließendem Barbecue ließ Raum für einige interessante und lehrreiche Diskussionen und rundete so die Veranstaltung im sonnigen Dubna ab.

Max Huber (MPI München)

## 8. First International GeoGebra Conference

Hagenberg, Österreich, 14. – 15.07.2009

<http://www.geogebra.org/conferences/2009.htm>

Die erste internationale Tagung der GeoGebra Community fand diesen Juli im Rahmen des RISC Summer im Schloss Hagenberg bei Linz statt. 120 Teilnehmer aus 35 Ländern diskutierten ihre Erfahrungen, Ideen und Pläne rund um die Open-Source-Software GeoGebra. Darunter waren neben Didaktikern von Universitäten auch Lehrer und interessierte Schüler zu finden. Nachdem sich die meisten Teilnehmer vorher nur virtuell via E-Mail und aus dem Online-Benutzerforum kannten, war diese Tagung eine gute Gelegenheit, um persönliche Kontakte zu knüpfen und zukünftige Zusammenarbeit anzubahnen.

Die fünf Arbeitsgruppen mit jeweils 20 bis 30 Teilnehmern befassten sich zwei Tage lang in mehreren kontinuierlichen Sessions mit den Themen *Software Development & Online Systems*, *Teaching Experiences in Primary and Secondary Schools*, *Creation of Instructional Materials*, *GeoGebra at Universities and in Teacher Education* sowie *GeoGebra Institutes & Research*. In den Working Groups wurde versucht, möglichst allen Teilnehmern die Möglichkeit zur aktiven Mitwirkung zu geben. Aus diesem Grund wurde ein Format mit fünfminütigen Impulsvorträgen gewählt, um genügend Zeit für Diskussionen zu erlauben.

In den Hauptvorträgen wurde einerseits ein Überblick über Pläne zur Weiterentwicklung der Software selbst (z. B. GeoGebraCAS und GeoGebra3D) sowie des internationalen Netzwerks von mittlerweile 15 GeoGebra Instituten (<http://www.geogebra.org/igi>) gegeben. Andererseits kamen Tomas Recio (Universität Cantabria) und Damjan Kobal (Universität Ljubljana) zu Wort, um Ihre Erfahrungen und Anregungen aus mathematischer bzw. didaktischer Sicht zu erläuterten.

Einen ausführlichen Konferenzbericht, die Präsentationsdateien der Hauptvorträge und Working Groups sowie Bilder von der Konferenz finden Sie auf der oben angegebenen Konferenzwebseite bzw. dem Konferenzwiki <http://ggbcconference2009.pbworks.com>.

Markus Hohenwarter (Florida Atlantic University)

## 9. ICTMA 14 – 14th International Conference on the Teaching of Mathematical Modelling and Applications

Hamburg, 27. – 31.07.2009

<http://www.ictma14.de/>

An der Tagung haben etwa 150 Kolleginnen und Kollegen aus 30 Ländern rund um den Globus teilgenommen. Aus Sicht unserer Fachgruppe war die Sektion „Technogoly“ besonders interessant. Dort wurden aus den Perspektiven verschiedenster Länder u. A. unterschiedliche Aspekte des Einsatzes von CAS vorgestellt und diskutiert. Ein Höhepunkt war der Hauptvortrag von Helmut Neunzert (Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, Kaiserslautern): *Mathematical Modelling and a New Role for Mathematics as a Key Technology*. Die Tagung und ihr Rahmenprogramm waren exzellent von Gabriele Kaiser und ihrem Team vorbereitet und organisiert worden. Jeden Abend gab es einen „offiziellen“ Anlass zum Treffen mit alten und Gewinnen von neuen Freunden. Wie üblich werden Proceedings erscheinen (bei Springer).

Hans-Wolfgang Henn (TU Dortmund)

## 10. ISSAC 2009

Seoul, Südkorea, 28. – 31.07.2009

<http://issac2009.kias.re.kr/>



Tagungsfoto ISSAC 2009

Die ISSAC 2009 (International Symposium on Symbolic and Algebraic Computation) fand am KIAS (Korea Institute of Advanced Studies) in Seoul, Südkorea, statt und wurde von Hyungju (Alan) Park als Local Arrangements Chair geleitet.



Korea Institute of Advanced Studies



Die Tagung wurde von 115 Teilnehmern besucht, wovon diesmal nur recht wenige aus Deutschland kamen. Die mit Abstand größte Teilnehmergruppe kam mit 28 Personen aus Frankreich.

Die Tagung hatte wieder ein sehr interessantes und breitgefächertes Programm, das – bis auf drei eingeladene Plenarvorträge von Markus Püschel (*Automatic Synthesis of High Performance Mathematical Programs*), Tetsuo Ida (*Symbolic and Algebraic Methods in Computational Origami*) und Marc Giusti (*A Gröbner Free Alternative to Solving and a Geometric Analog of Cook's Thesis*) – in jeweils zwei Parallel-Sektionen durchgeführt wurde.

Auf dem ISSAC Business Meeting, das am Abend des 29. Juli stattfand, wurde in der üblichen Tagungstradition von den Teilnehmern der Tagungsort für die ISSAC 2011 bestimmt. Bewerber waren Boston und San Jose, in einer knappen Entscheidung kam diesmal San Jose zum Zug. Die ISSAC wird dann erstmals im Rahmen der großen FCRC 2011 der ACM (Federated Computing Research Conference, [www.acm.org/fcrc](http://www.acm.org/fcrc), 4.–11. Juni 2011) stattfinden.



General Co-Chairs Alan Park und Jeremy Johnson

Auf derselben Sitzung stellte der Sprecher der Fachgruppe Wolfram Koepf als General Chair der ISSAC 2010 die Pläne für die ISSAC-Tagung 2010 vor, die vom 25.–28. Juli 2010 in München stattfinden wird. Local Arrangements Chair der ISSAC 2010 ist Ernst W. Mayr von der TU München und Program Committee Chair ist Stephen M. Watt von der University of Western Ontario in Kanada. Es gibt inzwischen eine Website (<http://www.issac-conference.org/2010>), auf welcher Sie auch den *Call for Papers* finden, s. auch S. 43.



Dohan Kim sowie Tanzeinlage auf dem Bankett

Eine weitere übliche Tagungstradition ist die Vergabe einiger Preise anlässlich des Banketts, das am Abend des 30. Juli stattfand. Der *Distinguished Paper Award* ging diesmal an Chris Brown für seine Arbeit *Fast simplifications for Tarski formulas*, und es wurden vier Preise für *Best Student Papers* vergeben. Die vier Gewinner waren Jorge Martin Morales und Wolf Daniel Andres für ihren Artikel *Principal intersection and Bernstein-Sato polynomial of affine variety*

(gemeinsam mit Viktor Levandovskyy) sowie Luca De Feo für *Fast arithmetics in Artin-Schreier towers over finite fields* (gemeinsam mit Eric Schost) und Yong Jae Cha für *Liouvillian solutions of irreducible linear difference equations* (gemeinsam mit Mark van Hoeij). Der *Best Poster Award* ging an Marc Moreno Maza und der Preis für die *Best Software Presentation* an Viktor Levandovskyy.

Das Gastgeberland Südkorea war sehr spendabel und sponsorte das Bankett großzügig. Der Präsident der Korean Mathematical Society Prof. Dr. Dohan Kim konnte mit großer Freude verkünden, dass das IMU Executive Committee die Stadt Seoul als einzigen Kandidaten für die Austragung des ICM 2014 (International Congress of Mathematicians) nominiert hat.



Das Editorial Board des JSC

Alle zwei Jahre findet auf der ISSAC ein Treffen des Editorial Boards des JSC (Journal of Symbolic Computation) <http://www.sciencedirect.com/science/journal/07477171> statt. Der Editor-in-Chief Hoon Hong berichtete, dass es Bestrebungen des Verlags Elsevier gibt, diese – wie auch andere Zeitschriften – nur noch digital anzubieten. Wann diese Änderungen umgesetzt werden, weiß momentan niemand. Es wurde diskutiert, ob man die Zeitschrift dann lieber als Online-Zeitschrift weiterführen möchte. Hoon Hong machte klar, dass der Name der Zeitschrift Eigentum des Verlags ist. Daher könnte man höchstens eine neue Zeitschrift gründen, die dann natürlich auch bzgl. Ratings wie dem Citation Index bei Null beginnt.

Beim *Journal of Algorithms* trat aus ähnlichen Gründen der Editor-in-Chief Don Knuth und das gesamte Editorial Board zurück. Die Antwort von Elsevier ließ nicht lange auf sich warten. Es wurde umgehend ein neues Editorial Board eingesetzt und den Lesern wurde mitgeteilt:

The Managing Editors and the Publisher announce that the Editorial Board of the Journal of Algorithms has resigned per January 1 of this year because of an unresolved dispute concerning the commercial aspects of scientific publishing. Papers which have been submitted prior to this date will be refereed in the usual way and published in the course of this year and next year. Papers submitted after this date will be forwarded to the new Editorial Board, which will be appointed shortly. It is expected that this transition will not result in any additional publication delay.

(s. <http://www-cs-faculty.stanford.edu/~knuth/joalet.pdf>) und <http://www.cs.colorado.edu/~hal/s.pdf>. Beim JSC bleibt zunächst einmal alles beim Alten, und wir müssen weiter beobachten, wie sich die Lage entwickelt.

Wolfram Koepf (Universität Kassel)

11. **CASC 2009 – The 11. International Workshop on Computer Algebra in Scientific Computing**  
Kobe, Japan, 13. – 17.09.2009

<http://www14.in.tum.de/konferenzen/CASC2009/program.html>

Vom 13. bis zum 17. September 2009 fand der 11. *International Workshop on Computer Algebra in Scientific Computing* in Kobe in Japan statt, in hervorragender Weise vor Ort organisiert von Prof. Nagasaka von der Universität Kobe, Prof. Kitamoto von der Universität Yamaguchi, und von Dr. Yamaguchi von Cybernet Systems.

Wie in vergangenen Jahren deckte die Tagung, die einer Zusammenarbeit zwischen Staaten der ehemaligen Sowjetunion (GUS) und Deutschland entsprang und eigentlich abwechselnd in GUS-Ländern bzw. in Deutschland stattfindet, dieses Mal aber ausnahmsweise auf Grund des Wunsches einer besonders aktiven Gruppe von Teilnehmern an den bisherigen Tagungen aus Japan eben dorthin verlegt wurde, einen weiten Themenbereich ab.



Kobe, Sannomiya District

Wie schon bei früheren CASC-Tagungen lag ein klarer Schwerpunkt bei Differential- und Differenzengleichungen bzw. Dynamischen Systemen mit Anwendungen in der Physik. Dabei spielen *hybride* Systeme eine besondere Rolle, in denen symbolische Computeralgebra-Methoden als essentieller Teil oder aber auch zur Herleitung und Analyse von numerischen Verfahren benutzt werden. Hier ist insbesondere die Anwendung symbolischer Methoden zur Herleitung neuer Differenzenschemata zur numerischen Integration partieller Differentialgleichungen zu nennen, z. B. für zweidimensionale Navier-Stokes-Gleichungen. Aber auch klassische Fragestellungen der Computeralgebra kamen zu Wort, mit Themen wie: Berechnung von Gröbnerbasen, Quantorenelimination, Zahlentheorie, geschlossene Lösung von Differenzengleichungen.

In den beiden eingeladenen Vorträgen behandelte zunächst X.-S. Gao das Problem der Triangulierung impliziter algebraischer Flächen mit Singularitäten, wie es insbesondere in der Computergraphik, der geometrischen Modellierung und in FE-Methoden immer wieder auftritt. Der zweite eingeladene Vortrag, von J.-Ch. Faugère, behandelte die Fragestellung der erfolgreichen (und effizienten) Anwendung von Computeralgebra-Methoden auf Fragen der Sicherheit bestimmter kryptographischer Systeme.

Das gesamte Programm der Tagung ist auf der Webseite <http://www14.in.tum.de/konferenzen/CASC2009/program.html> zu finden, die Online-Version der Proceedings (LNCS 5743) steht auf <http://www.springerlink.com/content/978-3-642-04102-0/> zur Verfügung.

Ernst W. Mayr (München)

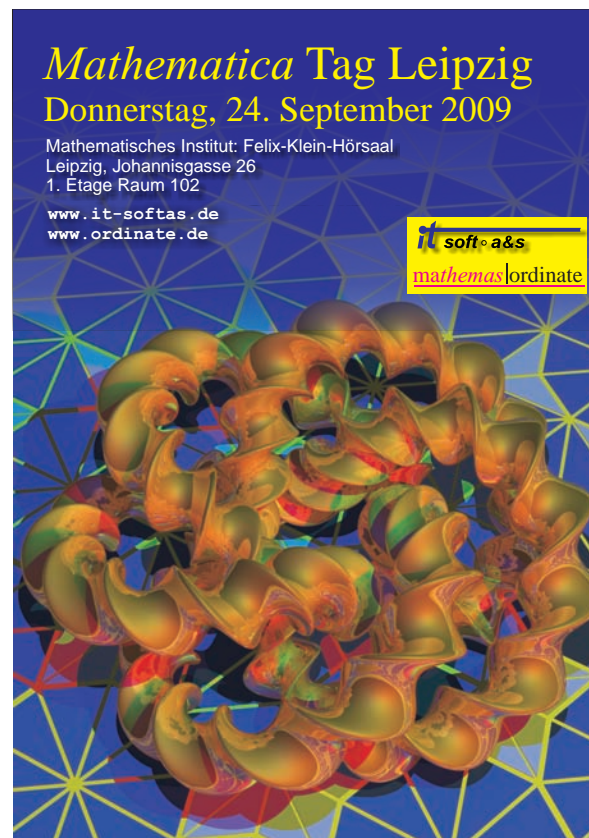
12. **Zweiter Mathematica-Tag Leipzig**

Leipzig, 24.09.2009

<http://www.ordinate.de/mathematicaTag.htm>

Am 24.09.2009 fand im Felix-Klein-Hörsaal der Fakultät für Mathematik und Informatik der Universität Leipzig der nunmehr zweite Leipziger Mathematica-Tag statt. In einführenden Worten erinnerte Hans-Jörg Möhring (it-soft a&s) an Dr. Jens Kuska, der Anfang Juli unerwartet verstorben war.

Im ersten Vortrag erläuterte Carsten Herrmann von *mathemas ordinate* die Prinzipien und die Neuigkeiten in Version 6 und 7 (u. A. dynamische Interaktivität, integriertes paralleles Rechnen und Bildverarbeitung). Dann gab es zwei Vorträge von Anwendern, die sehr schön die Zusammenarbeit von Mathematica mit einem andern Softwarepaketen zur FEM demonstrierten: „Vom CT-Datensatz zum Finite-Elemente-Modell einer knöchernen Struktur“ von Juliane Mai (UFZ Leipzig-Halle) und „Anwendung von Mathematica zur Analyse von Bilddaten aus Gefügeuntersuchungen“ von Stephan Weinold (HTWK Leipzig).



Dr. Bernd Fiedler sprach dann über „Große Gruppenringe, diskrete Fouriertransformationen und große Integermatrizen – Erfahrungen bei der Anwendung von Mathematica“. Dabei wurden die Möglichkeiten und Vorteile eines Computeralgebrapaketes herausgearbeitet. Den Abschluss bildete schließlich ein Crashkurs „Wenn Mathematica die Grafikkarte zu Hilfe nimmt: Die Compute Unified Device Architecture von NVIDIA“. Es wurde deutlich, wie man mit Mathematica sehr schnelle Berechnungen machen kann. Am 20.11.2009 findet der 11. Berliner Mathematica-Tag (<http://www.ordinate.de/mathematicaTag.htm>) statt.

Carsten Herrmann (Altenholz)



### 13. MCAT – Zwölfter Mitteldeutscher Computeralgebra-Tag

Anhalt, 09.10.2009

<http://www.informatik.uni-leipzig.de/~graebe/MCAT/mcat12.html>

Am 9. Oktober 2009 fand der Zwölfte Mitteldeutsche Computeralgebratag (MCAT) an der Fachhochschule Anhalt, Standort Köthen statt. Die lokale Organisation lag in den Händen von Werner Loch (Fachbereich Informatik).

Im Mittelpunkt des Programms stand dieses Mal das System *Matlab* der Firma Mathworks, das mit der Übernahme von MuPAD und der Integration desselben als Symbolic Toolbox im letzten Jahr nach einer offensichtlich nicht sehr fruchtbaren mehrjährigen Liaison mit Maple einen neuen Anlauf nimmt, sich auch im symbolischen Bereich zu verstärken. Spannend an diesen Entwicklungen ist vor allem, dass das Zusammenwachsen numerischer und symbolischer Methoden zu einer neu ausgerichteten Disziplin „Computermathematik“ – wie von Johannes Grabmeier schon vor über 10 Jahren vorausgesagt – hier von der Seite eines im Numerikbereich etablierten Systems aus erfolgt. Leider konnten wir niemanden von Mathworks gewinnen, uns über diese Entwicklungen zu berichten.

Matlab ist ein im Ingenieurbereich weit verbreitetes System für numerische Rechnungen, das damit in der Lehre besonders an Fachhochschulen eine wichtige Rolle spielt. Dies – besonders Kollegen aus Fachhochschulen anzusprechen

– war eine weitere Zielgröße des diesjährigen Computeralgebratags, die leider mit Blick auf die Resonanz nicht erreicht wurde. Dabei war das Programm durchaus die Reise wert, wie mir ein Teilnehmer im Nachhinein noch einmal bestätigte.

Das Vormittagsprogramm bestritt Jörg-M. Sautter (drei Jahre Senior Application Engineer bei MathWorks, heute Professor an der Fachhochschule Aschaffenburg), der in einem ersten Vortrag einen detaillierten Überblick über die Möglichkeiten von Matlab in Forschung und Lehre gab und in einem zweiten Beitrag über das Lösen von Optimierungsproblemen mit Matlab vortrug.

Die Nachmittagssitzung fand gemeinsam mit dem Workshop „Contextual Analysis of High-Resolution ECG’s for the Prediction of Sudden Cardiac Arrest“ statt, wo Leif Sörnmo (Univ. Lund) in seinem Vortrag „The Lund Matlab ECG Toolbox: History and Future“ eine Anwendung von Matlab im Bereich der biomedizinischen Signalverarbeitung vorstellte. Neben technischen Fragen ging Sörnmo in seinem Beitrag auch auf die Möglichkeiten der Verzahnung von Forschung und Ausbildung im studentischen und Graduiierungsbereich ein, die ein komplexes Projekt wie die Entwicklung einer Toolbox mit dezidierten Coding- und Dokumentationsstandards bietet.

Für weitere Informationen zu den Mitteldeutschen Computeralgebratagen verweise ich auf <http://www.informatik.uni-leipzig.de/~graebe/MCAT>.

Hans-Gert Gräbe (Universität Leipzig)

### 1. CADE 2009 – Workshop on Computer Algebra and Differential Equations

Pamplona, 28. – 31.10.2009

<http://www.unirioja.es/cu/aipasc/cade/Welcome.htm>

CADE 2009 is the continuation of a series of international workshops on computer algebra and its applications that started in Dubna (Russia) in 1979, 1982, 1985, 1990 and 2001 and followed in Turku (Finland) in 2007. This series of conferences covered all areas of computer algebra, in particular those related to the applications of symbolic and algebraic computation to ordinary and partial differential equations.

The goal of this workshop is to bring together researchers developing and applying computer algebra techniques in the field of differential equations. In this way the meeting will be a cross-fertilisation between the different topics of computer algebra and differential equations, and will increase communication between the scientists working in the theoretical aspects of computer algebra and the researches who use them, thus encouraging cooperation among the participants.

The programme schedule will consist of three working days plus an extra day for an excursion to some historical sights in Navarra. The first three days regular contributions will be held and no parallel sessions will be scheduled. The number of expected participants is between 30 and 50.

### 2. Elfter Berliner Mathematica-Tag

Berlin, 20.11.2009

<http://www.ordinate.de/mathematicaTag.htm>

Der elfte Berliner Mathematica-Tag wird am 20.11.2009 stattfinden – ein interdisziplinäres Kolloquium zur Anwendung von Mathematica in den Naturwissenschaften. Gastgeber ist wie in den Jahren zuvor das WIAS Berlin.

Geplant sind u. A. Vorträge über *webMathematica*, ein „Mathematica-Training“ sowie ein Vortrag von Holger Perlt (Universität Leipzig) zum Thema *Operations Research: Ein Mathematica-Paket zur Lösung von Optimierungsaufgaben mit Randbedingungen*.

Weitere Informationen sind demnächst auf der oben genannten Webseite verfügbar. Einen Bericht zum Mathematica-Tag in Leipzig finden Sie auf S. 39 in diesem Rundbrief.

### 3. Joint Conference of ASCM 2009 and MACIS 2009

Fukuoka, Japan, 14. – 17.12.2009

<http://gcoe.math.kyushu-u.ac.jp/ascm-macis2009/>

Two international conferences, the 9th Asian Symposium on Computer Mathematics (ASCM 2009) and the 3rd International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2009) will be held jointly at Fukuoka in December 14th -17th, 2009, supported by the GCOE program „Math-for-industry“ of the Graduate School of Mathematics of Kyushu University and Mathematical Research Center for Industrial Technology of Kyushu University.

ASCM is a series of conferences which serve as a forum for participants to present original research, learn of research progress and developments, and exchange ideas and views on doing mathematics using computers. The previous ASCM meetings were held in Beijing, China (1995), Kobe, Japan (1996), Lanzhou, China (1998), Chiang Mai, Thailand (2000), Matsuyama, Japan (2001), Beijing, China (2003), Seoul, Korea (2005), Singapore, Singapore (2007). Further information on previous ASCM symposia may be found at <http://www.mmrc.iss.ac.cn/ascm>.

MACIS is a new series of conferences where foundational research on theoretical and practical problems of mathematics for computing and information processing may be presented and discussed. MACIS also addresses experimental and case studies, scientific and engineering computation, design and implementation of algorithms and software systems, and applications of mathematical methods and tools to outstanding and emerging problems in applied computer and information sciences.

The programs of ASCM and those of MACIS will be organized independently by each program committee except invited talks. Sessions of ASCM and those of MACIS will be held in parallel and invited talks will be given in plenary sessions.



#### 4. ACAT 2010 – 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Jaipur, Indien, 22. – 27.02.2010

<http://acat2010.cern.ch/>

The ACAT workshop series, created back in 1990 as AI-HENP (Artificial Intelligence in High Energy and Nuclear Research) has been covering the tremendous evolution of computing in its most advanced topics, trying to setup bridges between computer science, experimental and theoretical physics.

The gap between the need for adapting applications to exploit the new hardware possibilities and the push toward virtualisation of resources is widening, creating more challenges as technical and intellectual progress continues.

ACAT 2010 proposes to explore and confront the different boundaries of the evolution of computing, and its possible consequences on our scientific activity.

The proceedings of the conference will be published in „Proceedings of Science“, the open access online journal organized by SISSA, the International School for Advanced Studies based in Trieste. The proceedings will be peer-reviewed. The refereeing process will be overseen by the editorial board.

#### 5. Gemeinsame Jahrestagung der DMV und der GDM 2010

München, 08. – 12.03.2010

<http://www.math2010.de/>

Zum zweiten Mal in ihrer Geschichte gestalten die Deutsche Mathematiker-Vereinigung (DMV) und die Gesellschaft für Didaktik der Mathematik (GDM) eine gemeinsame Jahrestagung. Im Jahr 2010 ist München nach 1998 zum zweiten Mal Tagungsort der GDM und seit 1893 nun bereits das fünfte Mal Tagungsort der DMV. Wir freuen uns, Sie im Namen des Mathematischen Instituts der Ludwig-Maximilians-Universität und des Zentrums Mathematik der Technischen Universität zu dieser Tagung einzuladen. Ort der Tagung wird das zentral gelegene Hauptgebäude der Ludwig-Maximilians-Universität sein.

Mathematik und Mathematikdidaktik sind zwei eng verbundene Disziplinen. So ist einerseits die Fachdidaktik als Wissenschaft vom mathematischen Denken und dessen Vermittlung auf immer neue Impulse aus aktuellen Zweigen der Mathematik angewiesen, andererseits ist der Bezug zur Fachdidaktik für die Fachwissenschaft von Bedeutung, und das besonders in Bezug auf die Ausbildung von Studierenden und Nachwuchswissenschaftlerinnen und -wissenschaftlern. Eine gemeinsame Jahrestagung stellt eine herausragende Möglichkeit zum gegenseitigen Austausch und zur Intensivierung von Kontakten dar.

Im Rahmen dieser Tagung wird es möglich sein, sowohl innerhalb der beiden Fachcommunities jeweils neue Ergebnisse zu diskutieren als auch im wechselseitigen Gespräch die Kooperation zu stärken. Wie bisher sind fachspezifische Einzelvorträge, Symposien und Hauptvorträgen vorgesehen, zu denen auch Interessierte aus der jeweils anderen Fachcommunity herzlich eingeladen sind. Darüber hinaus wird es Schnittstellenvorträge geben, die für beide Seiten interessante Impulse bieten sollen. Ein Rahmenprogramm bietet weitere Möglichkeiten zum informellen Austausch, auch über die Grenzen der beiden Gesellschaften hinweg.

Zu den Hauptvortragenden gehört auch Jonathan M. Borwein, auf dessen Artikel in diesem Rundbrief (S. 8) wir an dieser Stelle verweisen.

#### 6. 81. Jahrestagung der GAMM

Karlsruhe, 22. – 26.03.2010

<http://www.gamm2010.uni-karlsruhe.de/>

The GAMM e.V. cordially invites you to its 81st Annual Scientific Conference in Karlsruhe from March 22 to March 26, 2010.

On behalf of the DGLR and the GAMM we also invite you to the 53rd Ludwig Prandtl Memorial Lecture, which opens the conference program on Monday, March 22.

We invite all GAMM members to the regular General Assembly of GAMM e.V. during the conference at the Universität Karlsruhe (TH) on Wednesday, March 24.

Plenary Lectures include: Günter Brenn (TU Graz): *Viscoelastic Polymer Solutions in Elongational Flow*, Samuel Forest (Mines ParisTech): *Mechanics of Generalized Continua and Heterogeneous Materials*, Gilles Francfort (Université Paris 13): *Recent Developments in Brittle Fracture: The Variational Viewpoint and What It Implies*, Dierk Raabe (MPI Düsseldorf): *Multiscale Modelling of Crystal Mechanics Using Ab Initio and Continuum Methods*, Ingo Rehberg (Universität Bayreuth): *Snooping in the Sand*, Reinhold Schneider (TU Berlin): *Computation of the Electronic Structure*, Andrew Teel (UC Santa Barbara): *Hybrid Dynamical Systems: Robust Stability and Control*, and Christiane Tretter (Universität Bern): *Spectral Theory of Block Operator Matrices and Applications*.

#### 7. GCR 2010 – Geometric Constraints and Reasoning

Sierre, Schweiz, 22. – 26.03.2010

<http://www.lsi.upc.edu/~robert/gcr2010/gcr2010.html>

Geometric Constraints and Reasoning (GCR) is a technical track of the International Symposium on Applied Computing. For the past twenty years, the ACM Symposium on Applied Computing (SAC) has been a primary forum for applied computer scientists, computer engineers and application developers to gather, interact, and present their work.

SAC is sponsored by the ACM Special Interest Group on Applied Computing (SIGAPP), its proceedings are published by ACM in both printed form and CD-ROM; they are also available on the web through ACM's Digital Library.

As a special track of SAC, GCR is devoted to geometric reasoning taken in a broad sense. Initially, this track focused on geometric constraint solving but it appears that geometric computing and reasoning is closely related to this topic. Our aim is then to widen the audience and to make GCR a place where the communities of geometric constraint solving, computer aided deduction in geometry and related disciplines can meet and have fruitful exchanges.

SAC 2010 is also an opportunity to attend tracks related to GCR, about combinatorial optimization, constraint programming (non geometrical constraints), graph algorithms, numerical methods or interval analysis, etc.

#### 8. SCC 2010 – Second International Conference on Symbolic Computation and Cryptography

London, Großbritannien, 23. – 25.06.2010

<http://scc2010.rhul.ac.uk/>

Diese Tagung ist (nach Beijing 2008) die zweite in einer Reihe von internationalen Konferenzen, die sich mit dem aufstrebenden Gebiet der Anwendung von Methoden aus der

Computeralgebra in der Kryptographie befassen. Hauptthemen sind insbesondere das Design und die Kryptoanalyse von Kryptosystemen mit Hilfe der Computeralgebra sowie die Implementation von Algorithmen der Computeralgebra, die Anwendungen in der Kryptographie besitzen.



*Royal Holloway, University of London: Tagungsort der SCC 2010*

Als eingeladene Hauptvortragende haben bisher Antoine Joux (Université de Versailles Saint-Quentin-en-Yvelines) und Vladimir Gerdt (Joint Institute for Nuclear Research, Moskau) zugesagt.

*Extended abstracts* von Konferenzbeiträgen sind bis zum 14.3.2010 einzureichen. Die Tagung wird von Martin Albrecht, Carlos Cid und Jean-Charles Faugère organisiert.

## 9. ISSAC 2010 – International Symposium on Symbolic and Algebraic Computation

Technische Universität München,  
25. – 28.07.2010

<http://www.issac-conference.org/2010>

The International Symposium on Symbolic and Algebraic Computation is the premier conference for research in symbolic computation and computer algebra. ISSAC 2010 is the 35th meeting in the series.

The conference traditionally presents a range of invited speakers, tutorials, poster sessions and vendor exhibits with a centre-piece of contributed research papers. ISSAC 2010 will be hosted by the Technische Universität München.

ISSAC 2010 invites the submission of original research contributions to be considered for publication and presentation at the conference. All areas of computer algebra and symbolic mathematical computation are of interest.

**Algorithmic aspects:** Exact and symbolic linear, polynomial and differential algebra. Symbolic-numeric, homotopy, perturbation and series methods. Computational geometry, group theory and number theory. Summation, recurrence

equations, integration, solution of ODE and PDE. Symbolic methods in other areas of pure and applied mathematics. Theoretical and practical aspects, including general algorithms, techniques for important special cases, complexity analyses of algebraic algorithms and algebraic complexity.

**Software aspects:** Design of packages and systems. Data representation. Software analysis. Considerations for modern hardware, e.g., current memory and storage technologies, high performance systems and mobile devices. User interface issues, including collaborative computing and new methods for input and manipulation. Interfaces and use with systems for, e.g., document processing, digital libraries, courseware, simulation and optimization, automated theorem proving, computer aided design and automatic differentiation.

**Application aspects:** Applications that stretch the current limits of computer algebra, use computer algebra in new ways or in situations with broad impact.

Organizing Committee: Wolfram Koepf (General Chair), Stephen M. Watt (Program Committee Chair), Ernst W. Mayr (Local Arrangements Chair), Sergei Abramov (Tutorials Chair), Ilias Kotsireas (Poster Committee Chair), Michael Monagan (Software Exhibits Chair), Thomas Hahn (Treasurer), Peter Horn (Publicity Chair).



*Ernst W. Mayr und Stephen M. Watt*

### Important Dates:

Abstract submission:	Thursday 14 January 2010
Full paper deadline:	Thursday 21 January 2010
Reviews available:	Thursday 25 March 2010
Author response period:	29-31 March 2010
Acceptance notification:	Thursday 8 April 2010
Camera ready copy due:	Thursday 6 May 2010

Abstracts should be submitted by the abstract submission date. Authors may choose to submit the full paper with the abstract, or later, up to the full paper deadline. Submission is via EasyChair, at the web site

<http://www.easychair.org/conferences/?conf=issac2010>.

Papers will be reviewed by the Program Committee and external referees. Authors will have an opportunity to respond to reviews before the acceptance decisions are made. At least one author of each accepted paper must register for the conference and present the paper.



### DFG-Schwerpunktprogramm in der Computeralgebra eingerichtet – SPP1489: Algorithmische Methoden in Algebra, Geometrie und Zahlentheorie

**Wolfram Decker**  
Technische Universität Kaiserslautern

decker@mathematik.uni-kl.de



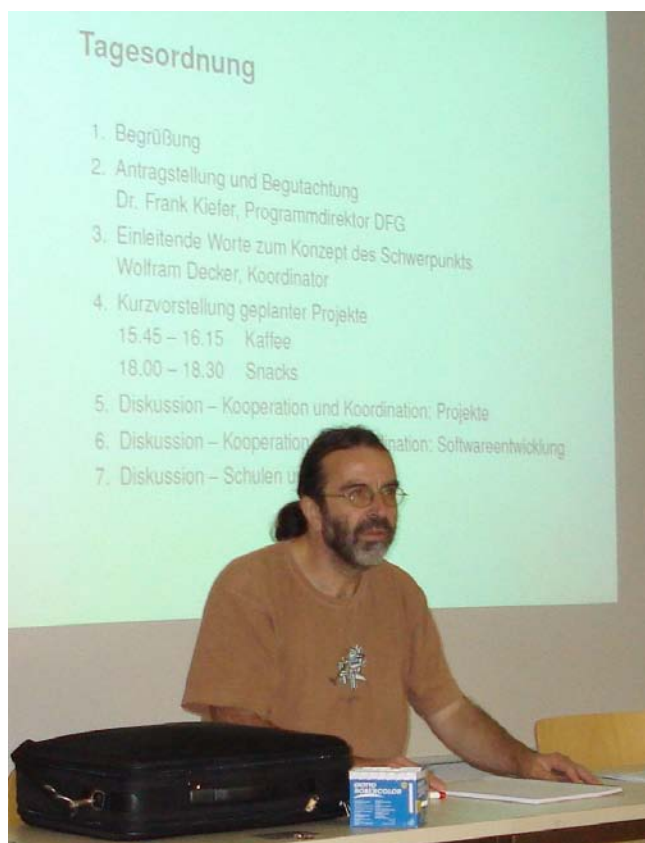
Der Senat der Deutschen Forschungsgemeinschaft hat die Einrichtung des oben genannten Schwerpunktprogramms mit einer geplanten Laufzeit von sechs Jahren beschlossen. Ziel des voraussichtlich ab April 2010 geförderten Programms ist es, die algorithmischen und experimentellen Methoden in den angegebenen Gebieten substantiell voranzutreiben, sie – wo erforderlich – zu verknüpfen und sie – kombiniert mit theoretischen Ansätzen – an zentralen Fragestellungen aus Theorie und Praxis zu erproben. Des Weiteren soll die Weiterentwicklung und Vernetzung von in Deutschland (mit-)entwickelten freien Computeralgebrasystemen projektbezogen auf verschiedenen Ebenen unterstützt werden. Alle erstellten Programmpakete und Datenbanken sollen öffentlich und frei zur Verfügung gestellt werden.



*Treffen in Mainz im August 2009*

Das Schwerpunktprogramm wird die übergreifende Zusammenarbeit der verschiedenen Teildisziplinen und Forschergruppen organisieren und unterstützen. Aufgrund seiner Ausrichtung werden insbesondere Anträge begrüßt, die Methoden von zwei oder mehr Themenschwerpunkten kombinieren, um sie in den Kerngebieten Zahlentheorie, Gruppentheorie und algebraische Geometrie einzusetzen. Von Interesse sind dabei auch methodische Querverbindungen zu inner- und außer-mathematischen Anwendungsbereichen (zum Beispiel

System- und Kontrolltheorie, Codierungstheorie, Kryptographie, CAD, algebraische Kombinatorik, algebraische Statistik) sowie die Kombination numerischer und symbolischer Methoden.



*Prof. Dr. Wolfram Decker, Koordinator des Programms*

Die konsequente Nachwuchsschulung und der effektive Wissenstransfer sind von fundamentaler Bedeutung für den Schwerpunkt. Neben dem regen Austausch von Mitarbeitern im Nachwuchsbereich und dem Webserver des Schwerpunkts sind vier verschiedene Typen von jährlich stattfindenden Schulen, Workshops und Konferenzen geplant, die die Bedürfnisse von Forschern in verschiedenen Karrierestadien berücksichtigen. Die Aktivitäten des Schwerpunkts sollen in Abstimmung mit der Fachgruppe Computeralgebra organisiert werden.



Zur Vorbereitung des Programms fand in der Zeit vom 25. bis 26. August 2009 an der Universität Mainz ein Treffen potenzieller Antragstellerinnen und Antragsteller statt. Die rege Teilnahme an diesem Treffen ist ein Indikator für das große Interesse an dem Programm. Im Rahmen des Treffens wurden erste Projektideen vorgestellt und mögliche Kooperationen besprochen. Weitere Themen waren die Koordination der Softwareentwicklung sowie die Konzeption von Schulen und Workshops.

Informationen zum Schwerpunktprogramm erteilt der Koordinator des Programms: Prof. Dr. Wolfram De-

cker, TU Kaiserslautern, [decker@mathematik.uni-kl.de](mailto:decker@mathematik.uni-kl.de). Informationen zur Antragstellung bei der DFG erteilt der verantwortliche Programmdirektor: Dr. Frank Kiefer, [frank.kiefer@dfg.de](mailto:frank.kiefer@dfg.de).

Anträge für zunächst drei Jahre sind in englischer Sprache einzureichen und müssen bis spätestens 1. November 2009 unter Angabe des Stichworts **SPP 1489** bei der Deutschen Forschungsgemeinschaft, Fachreferat Mathematik, Kennedyallee 40, 53175 Bonn, eingegangen sein.

---

## Preisverleihungen

---

### Verleihung des F. L.-Bauer-Preises an Stephen Wolfram

Thomas Hahn

Am 15. Juni 2009 fand in der Bayerischen Akademie der Wissenschaften in München eine Festveranstaltung zum 85. Geburtstag von Friedrich L. Bauer statt. Nach gebührender Ehrung von Bauer als „Vater der deutschen Informatik“ wurde der seinen Namen tragende Preis an Stephen Wolfram verliehen.

Der mit 25.000 € dotierte F.L.-Bauer-Preis wird zweijährig an Personen vergeben, die sich besondere Verdienste um das wissenschaftliche Rechnen erworben haben. Bisherige Preisträger waren Z. Manna, R. Milner, A. Troelstra, G. Stewart, H. Cohen und C.A.R. Hoare.

Die exzellente Laudatio „A New Kind of Scientist“ wurde von Bruno Buchberger gehalten. Er zeichnete den Lebensweg des Preisträgers nach und wies auf dessen Verdienste hin: die Entwicklung von Mathematica, sein Buch „A New Kind of Science“ und die dahinterstehenden wissenschaftlichen Konzepte und schließlich die neue Suchmaschine Wolfram Alpha. Er hob die wissenschaftliche, aber auch die unternehmerische Kompetenz Wolframs hervor. Etwa, dass die Kommerzialisierung von Mathematica eine brillante Idee war, denn sie vergrößerte nicht nur den User-Kreis ungemein, sie gab Usern auch die Möglichkeit, sich bei Problemen zu beschweren und Wünsche zu äußern. Kurz, die Entwicklung, die Mathematica letzten Endes genommen hat,

wäre als an einer Uni entwickelte „Nischensoftware“ kaum denkbar gewesen.

Die übergeordnete Struktur seines Lebenswerks wurde sodann vom Preisträger in seinem Festvortrag über „Computable Knowledge“ weiter erläutert. Nach einem kurzen Abriss der Geschichte des Rechnens ging er darauf ein, wie er Mathematica entwarf, zunächst für sein eigenes Verständnis, als ein System, mit dem sich (anders als in bis dahin existierenden Hochsprachen) beliebige Strukturen einfach implementieren ließen. Aus seiner Arbeit zu zellulären Automaten entstand dann die Einsicht, dass bereits sehr simple „Programme“ enorm komplexen Output liefern, was schließlich im Konzept des „Computable Universe“ mündete, das ist das „Universum“ aller Programme. Das traditionell induktive Herangehen der Wissenschaft könne dieses Computable Universe nicht auch nur zu einem Teil auszuloten. Deshalb, so plädiert er, müsse man über kurz oder lang vom verständnisgetriebenen Entwickeln von Algorithmen zu einem „Absuchen“ des Computable Universe übergehen. Wolfram Alpha darf als erste Implementierung dieses „Absuchens“ verstanden werden, und Wolfram sagt, er sei selbst überrascht gewesen, wie viel Computable Knowledge mit derzeitiger Hardware möglich sei.

---

## Kurze Mitteilungen

---

**Prof. Dr. Wolfram Decker** hat zum 1. Oktober 2009 im Rahmen der Mathematikinitiative des Landes Rheinland-Pfalz, der Technischen Universität Kaiserslautern und des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik (ITWM) den Ruf auf eine W3-Professur für Algebraische Geometrie und Computeralgebra angenommen. Mittelfristig wird er die Leitung der Entwicklergruppe des Computeralgebrasystems SINGULAR übernehmen.

## Aufnahmeantrag für Mitgliedschaft in der Fachgruppe Computeralgebra

(Im folgenden jeweils Zutreffendes bitte im entsprechenden Feld ☐ ankreuzen bzw. \_\_\_\_\_ ausfüllen.)

Titel/Name: _____		Vorname: _____	
<b>Privatadresse</b>			
Straße/Postfach: _____			
PLZ/Ort: _____		Telefon: _____	
E-mail: _____		Telefax: _____	
<b>Dienstanschrift</b>			
Firma/Institution: _____			
Straße/Postfach: _____			
PLZ/Ort: _____		Telefon: _____	
E-mail: _____		Telefax: _____	
Gewünschte Postanschrift: <input type="checkbox"/> Privatadresse <input type="checkbox"/> Dienstanschrift			

1. Hiermit beantrage ich zum 1. Januar 200\_\_\_\_ die Aufnahme als Mitglied in die Fachgruppe

### Computeralgebra (CA) (bei der GI: 0.2.1).

2. Der Jahresbeitrag beträgt €7,50 bzw. €9,00. Ich ordne mich folgender Beitragsklasse zu:

- ☐ **€7,50** für Mitglieder einer der drei Trägergesellschaften
- |                               |                        |
|-------------------------------|------------------------|
| <input type="checkbox"/> GI   | Mitgliedsnummer: _____ |
| <input type="checkbox"/> DMV  | Mitgliedsnummer: _____ |
| <input type="checkbox"/> GAMM | Mitgliedsnummer: _____ |

Der Beitrag zur Fachgruppe Computeralgebra wird mit der Beitragsrechnung der Trägergesellschaft in Rechnung gestellt. (Bei Mitgliedschaft bei mehreren Trägergesellschaften wird dies von derjenigen durchgeführt, zu der Sie diesen Antrag schicken.) ☐ Ich habe dafür bereits eine Einzugsvollmacht erteilt. Diese wird hiermit für den Beitrag für die Fachgruppe Computeralgebra erweitert.

- ☐ **€7,50.** Ich bin aber noch nicht Mitglied einer der drei Trägergesellschaften. Deshalb beantrage ich gleichzeitig die Mitgliedschaft in der

☐ GI ☐ DMV ☐ GAMM.

und bitte um Übersendung der entsprechenden Unterlagen.

- ☐ **€9,00** für Nichtmitglieder der drei Trägergesellschaften. ☐ Gleichzeitig bitte ich um Zusendung von Informationen über die Mitgliedschaft in folgenden Gesellschaften:

☐ GI ☐ DMV ☐ GAMM.

3. Die in dieses Formular eingetragenen Angaben werden elektronisch gespeichert. Ich bin damit einverstanden, dass meine Postanschrift durch die Trägergesellschaften oder durch Dritte nach Weitergabe durch eine Trägergesellschaft wie folgt genutzt werden kann (ist nichts angekreuzt, so wird c. angenommen).

- |  |  |
|--|--|
| <input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/> | a. Zusendungen aller Art mit Bezug zur Informatik, Mathematik bzw. Mechanik.<br>b. Zusendungen durch wiss. Institutionen mit Bezug zur Informatik, Mathematik bzw. Mechanik.<br>c. Nur Zusendungen interner Art von GI, DMV bzw. GAMM. |
|--|--|

Ort, Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

Bitte senden Sie dieses Formular an:

Sprecher der Fachgruppe Computeralgebra  
Prof. Dr. Wolfram Koepf  
Fachbereich Mathematik/Informatik  
Universität Kassel  
Heinrich-Plett-Str. 40  
34132 Kassel  
0561-804-4207, -4646 (Fax)  
koepf@mathematik.uni-kassel.de

---

# Fachgruppenleitung Computeralgebra 2008-2011

---

**Sprecher,  
Vertreter der DMV:**

Prof. Dr. Wolfram Koepf  
Fachbereich Mathematik  
Universität Kassel  
Heinrich-Plett-Str. 40  
34132 Kassel  
0561-804-4207, -4646 (Fax)  
koepf@mathematik.uni-kassel.de  
<http://www.mathematik.uni-kassel.de/~koepf>

**Fachreferent Internet:**

Dr. Hans-Gert Gräbe, apl. Prof.  
Institut für Informatik  
Universität Leipzig  
Postfach 10 09 20  
04009 Leipzig  
0341-97-32248  
graebe@informatik.uni-leipzig.de  
<http://www.informatik.uni-leipzig.de/~graebe>

**Fachexperte Physik:**

Dr. Thomas Hahn  
Max-Planck-Institut für Physik  
Föhringer Ring 6  
80805 München  
089-32354-300, -304 (Fax)  
hahn@feynarts.de  
<http://www.th.mppmu.mpg.de/members/hahn>

**Fachreferent Themen und Anwendungen:**

Prof. Dr. Florian Heß  
Institut für Mathematik  
Technische Universität Berlin  
Straße des 17. Juni Nr. 136  
10623 Berlin  
030-314-25062, -29953 (Fax)  
hess@math.tu-berlin.de  
<http://www.math.tu-berlin.de/~hess>

**Fachreferent CA-Systeme und -Bibliotheken:**

Prof. Dr. Gregor Kemper  
Zentrum Mathematik – M11  
Technische Universität München  
Boltzmannstr. 3  
85748 Garching  
089-289-17454, -17457 (Fax)  
kemper@ma.tum.de  
<http://www-m11.ma.tum.de/~kemper>

**Fachreferent CA an der Hochschule:**

Prof. Dr. Gunter Malle  
Fachbereich Mathematik  
Technische Universität Kaiserslautern  
Gottlieb-Daimler-Straße  
67663 Kaiserslautern  
0631-205-2264, -3989 (Fax)  
malle@mathematik.uni-kl.de  
<http://www.mathematik.uni-kl.de/~malle>

**Fachreferent Schule:**

StD Dr. Jörg Meyer  
Schäfertrift 16  
31789 Hameln  
05151-54236  
J.M.Meyer@t-online.de

**Redakteur Rundbrief:**

Dr. Markus Wessler  
Fakultät für Betriebswirtschaft  
Fachhochschule München  
Am Stadtpark 20  
81243 München  
089-1265-2773, -2714 (Fax)  
markus.wessler@hm.edu

**Stellvertretende Sprecherin,  
Fachreferentin Fachhochschulen:**

Prof. Dr. Elkedagmar Heinrich  
Fachbereich Informatik  
Hochschule für Technik,  
Wirtschaft und Gestaltung Konstanz  
Brauneggerstr. 55  
78462 Konstanz  
07531-206-343, -559 (Fax)  
heinrich@htwg-konstanz.de  
[http://www.in.fh-konstanz.de/inhalte/de/KONTAKT/persseiten\\_nbc/heinrich.html](http://www.in.fh-konstanz.de/inhalte/de/KONTAKT/persseiten_nbc/heinrich.html)

**Fachreferent Computational Engineering,  
Vertreter der GAMM:**

Prof. Dr. Klaus Hackl  
Lehrstuhl für Allgemeine Mechanik  
Ruhr-Universität Bochum  
Universitätsstr. 150  
44780 Bochum  
0234-32-26025, -14154 (Fax)  
klaus.hackl@rub.de  
<http://www.am.bi.ruhr-uni-bochum.de>

**Fachreferent Lehre und Didaktik:**

Prof. Dr. Hans-Wolfgang Henn  
Fakultät für Mathematik  
Technische Universität Dortmund  
44221 Dortmund  
0231-755-2939, -2948 (Fax)  
henn@math.tu-dortmund.de  
<http://www.wolfgang-henn.de>

**Fachexperte Industrie:**

Prof. Dr. Michael Hofmeister  
Siemens AG  
Corporate Technology  
Discrete Optimization  
Otto-Hahn-Ring 6  
81739 München  
089-636-49476, -42284 (Fax)  
michael.hofmeister@siemens.com  
<http://www.siemens.com>

**Fachreferent Jahr der Mathematik:**

Prof. Dr. Martin Kreuzer  
Fakultät für Informatik und Mathematik  
Universität Passau  
Innstr. 33  
94030 Passau  
0851-509-3120, -3122 (Fax)  
martin.kreuzer@uni-passau.de  
<http://www.fim.uni-passau.de/~kreuzer>

**Fachreferent ISSAC 2010,  
Vertreter der GI:**

Prof. Dr. Ernst W. Mayr  
Lehrstuhl für Effiziente Algorithmen  
Fakultät für Informatik  
Technische Universität München  
Boltzmannstraße 3  
85748 Garching  
089-289-17706, -17707 (Fax)  
mayr@in.tum.de  
<http://www.in.tum.de/~mayr/>

**Fachreferentin Publikationen und Besprechungen:**

Prof. Dr. Eva Zerz  
Lehrstuhl D für Mathematik  
RWTH Aachen  
Templergraben 64  
52062 Aachen  
0241-80-94544, -92108 (Fax)  
eva.zerz@math.rwth-aachen.de  
<http://www.math.rwth-aachen.de/~Eva.Zerz/>