Oktober 2016

Computeralgebra Rundbrief

> Ausgabe 59

www.fachgruppe-computeralgebra.de

Maple™

Mehr leisten mit Maple 2016!

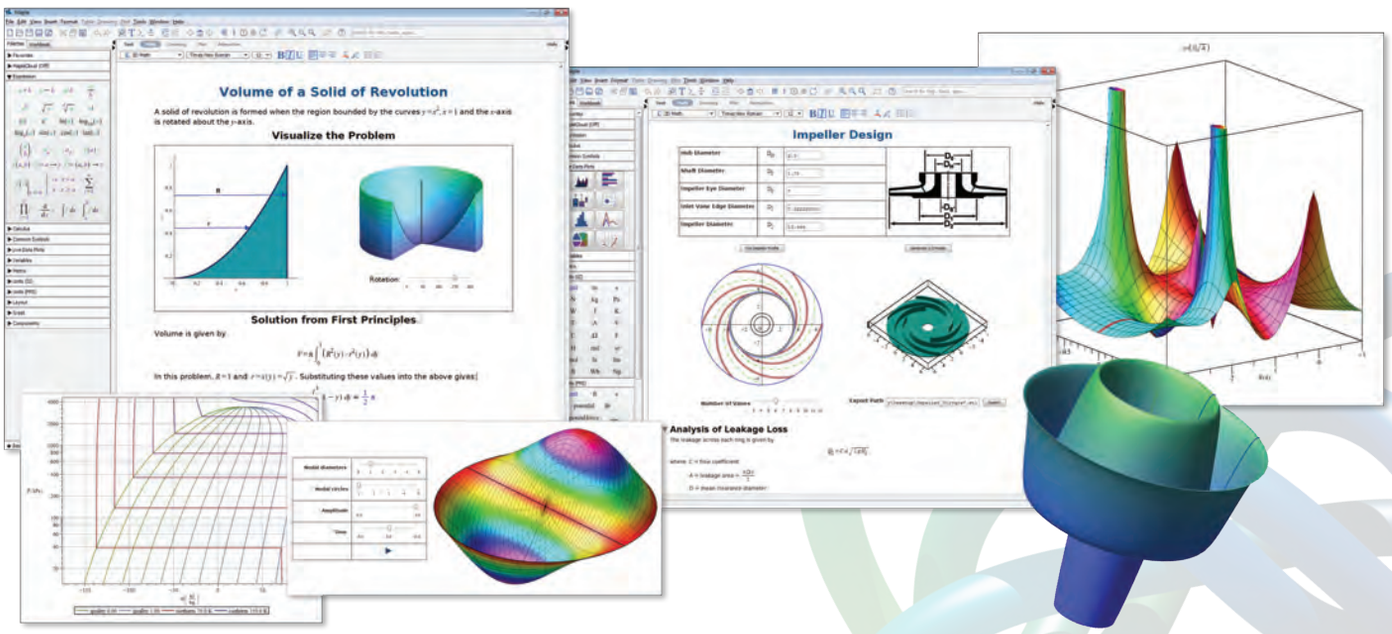
Maple 2016 lässt Sie mehr Aufgaben noch leichter lösen

Führen Sie mit Clickable Math™ neue Operationen mit einem einfachen Mausklick aus

Nutzen Sie neue Werkzeuge zur statistischen Analyse und Visualisierung, um bessere Einblicke in Ihre Daten zu erhalten

Lösen Sie noch mehr mathematische Probleme

Helfen Sie Ihren Studierenden, Konzepte mit neuen interaktiven Math Apps zu erkunden



Mehr erfahren Sie in userem Video *What's New in Maple 2016*:

www.maplesoft.com/M2016/CAR

Maplesoft

www.maplesoft.com | info@maplesoft.com • Tel: +49 241 980919 30

Computeralgebra-Rundbrief

Nr. 59 Oktober 2016

Inhaltsverzeichnis

Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und der GAMM (verantwortliche Redakteure: Prof. Dr. Michael Cuntz, Prof. Dr. Florian Heß, car@mathematik.de)

Der Computeralgebra-Rundbrief erscheint halbjährlich, Redaktionsschluss 15.02. und 15.09. 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.

GI (Gesellschaft für Informatik e.V.)
Wissenschaftszentrum
Ahrstr. 45
53175 Bonn
Telefon 0228-302-145
Telefax 0228-302-167
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
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 0351-463-33448
Telefax 0351-463-37086
GAMM@mailbox.tu-dresden.de
http://www.gamm-ev.de

Mitteilungen der Sprecher

Liebe Mitglieder der Fachgruppe Computeralgebra,

die Herbstsitzung der Fachgruppenleitung fand am 5. September in Hannover statt. Eines der Themen war die Vorbereitung der Wahl einer neuen Fachgruppenleitung, die sich in der Frühjahrssitzung 2017 konstituieren wird. Mehr hierzu weiter unten.

Eine der Aktivitäten der Fachgruppe war wieder die Verleihung einiger Preise bei der Konferenz ISSAC, siehe den hierzu den Tagungsbericht. So konnten wir einen der Preisträger dazu gewinnen, den ersten Artikel des Hefts zu schreiben, bei dem es um die Mathematik und Algorithmik von gewissen mechanischen Zeichenkonstruktionen geht. Es folgen ein Artikel über das Computeralgebra-System Normaliz und — vermutlich durch zufällige Häufung — gleich drei Artikel aus dem Bereich „Computeralgebra in der Hochschule“. Auch die Rubrik „Computeralgebra in der Schule“ ist wieder mit einem sehr anregenden Artikel vertreten. Insgesamt fällt dieses Heft damit relativ umfangreich aus.

Unter den Tagungsankündigungen befindet sich diesmal wieder unsere „eigene“ Tagung, die Computeralgebra-Tagung der Fachgruppe, die wir an dieser Stelle besonders hervorheben möchten. Sie findet im kommenden Mai in Kassel statt. Die Computeralgebra-Tagung bietet gerade Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftlern eine gute Möglichkeit, ihre Forschung vorzustellen und Kontakte zu knüpfen. Daneben gibt es Vorträge von erfahreneren Wissenschaftlerinnen und Wissenschaftlern, und genügend Zeit für Gespräche. Bitte informieren Sie in Ihrer Umgebung über die Tagung und nehmen Sie teil, mit oder ohne eigenen Vortrag.

Nun zurück zu der anstehenden Wahl der neuen Fachgruppenleitung. Es werden insgesamt neun Mitglieder der Fachgruppenleitung gewählt, daher hat jedes Mitglied der Fachgruppe bis zu neun Stimmen, wobei keine Häufung der Stimmen möglich ist. Abgesehen von den gewählten Mitgliedern besteht die Fachgruppenleitung aus je einem Vertreter der DMV, GI und GAMM, der von diesen Organisationen entsandt wird, sowie bis zu drei durch die Fachgruppenleitung berufenen Fachexperten. Die Wahlleitung wird gebildet von Wolfram Koepf und Ernst Mayr, die als Vertreter der DMV bzw. GI selbst nicht zur Wahl stehen. Bitte schicken Sie Ihren Stimmzettel im verschlossenen Wahlumschlag zusammen mit der unterschriebenen Erklärung im beigefügten Rücksendeumschlag bis zum

Dienstag, 15. November 2015 (Eingang beim Wahlleiter)

an die angegebene Adresse. Auf der nächsten Seite finden Sie die Kurzvorstellungen der Kandidatinnen und Kandidaten.

Wir wünschen Ihnen eine angenehme und anregende Lektüre dieses Hefts.

Gregor Kemper *Florian Heß*

Kandidatinnen und Kandidaten für die Fachgruppenleitung

- **Prof. Dr. Claus Fieker**, Professor für konstruktive Zahlentheorie und Computeralgebra an der TU Kaiserslautern. Arbeitsgebiete: Computeralgebra, konstruktive Zahlentheorie, Klassenkörpertheorie, Darstellungstheorie und Galois Theorie. Mitentwickler von Magma (elf Jahre in Sydney), Kant/Kash und Singular.
  http://www.mathematik.uni-kl.de/~fieker

- **Prof. Dr. Anne Frühbis-Krüger**, wissenschaftliche Mitarbeiterin und apl. Professorin am Institut für Algebraische Geometrie der Leibniz Universität Hannover, Arbeitsgebiete: Algorithmische Singularitätentheorie, Algorithmische Algebraische Geometrie, seit 1996 Mitarbeit an der Entwicklung des Computeralgebrasystems Singular.
  http://gandalf.krueger-berg.de/~anne

- **Prof. Dr. Meinolf Geck**, Lehrstuhl für Algebra an der Universität Stuttgart. Arbeitsgebiete: Darstellungstheorie, algebraische Lie-Theorie, Computeralgebra. Mitautor des Chevie-Pakets, Entwicklung des PyCox-Pakets.
  http://www.mathematik.uni-stuttgart.de/~geckmf

- **Dr. Thomas Hahn**, wissenschaftlicher Mitarbeiter am Max-Planck-Institut für Physik, München. In der Fachgruppenleitung seit 2002 als Fachexperte Physik, Autor der Computeralgebra-Softwarepakete FeynArts und FormCalc für Rechnungen im Bereich der Teilchenphysik.
  http://wwwth.mpp.mpg.de/members/hahn

- **Prof. Dr. Florian Heß**, Professor am Institut für Mathematik der Carl von Ossietzky Universität Oldenburg. Arbeitsgebiete: Algorithmische algebraische Zahlentheorie und Geometrie, speziell algebraische Funktionenkörper, Kurven und Anwendungen auf Kryptographie und Codierungstheorie. Umfangreiche Mitarbeit an den Computeralgebrasystemen Kant/Kash und Magma sowie Tätigkeiten im Bereich der Kryptographie.
  http://www.staff.uni-oldenburg.de/florian.hess

- **Prof. Dr. Max Horn**, Juniorprofessor am Mathematischen Institut der Justus-Liebig-Universität Gießen. Arbeitsgebiete: Computeralgebra (insbesondere Gruppentheorie) algebraische Lie-Theorie, Kac-Moody-Gruppen und Gebäude. Mitentwickler des Computeralgebrasystems GAP.
  http://www.quendi.de/math

- **Prof. Dr. Gregor Kemper**, Professor für algorithmische Algebra an der TU München. Arbeitsgebiete: Invariantentheorie, algorithmische kommutative Algebra, Computeralgebra. Autor von Software-Paketen für Invariantentheorie in Maple und Magma.
  http://www-m11.ma.tum.de/~kemper

- **Prof. Dr. Jürgen Klüners**, Professor für Computeralgebra und Zahlentheorie an der Universität Paderborn. Arbeitsgebiete: Computeralgebra, Galois- und Zahlentheorie. Mitentwickler der Computeralgebrasysteme Kant/Kash und Magma sowie einer Datenbank für Zahlkörper. Mitglied der Koordinatorengruppe des DFG-Schwerpunktprogramms 1489.
  http://www2.math.uni-paderborn.de/people/juergen-klueners

- **Prof. Dr. Martin Kreuzer**, Universitätsprofessor, Lehrstuhl für Symbolic Computation, Fakultät für Informatik und Mathematik, Universität Passau. Arbeitsgebiete: Computeralgebra, insbesondere Gröbnerbasen und Randbasen, industrielle Anwendungen der Computeralgebra, algebraische Kryptographie, algebraische Geometrie. Leiter des Entwicklerteams des Computeralgebrapakets ApCoCoA.
  http://staff.fim.uni-passau.de/~kreuzer

- **StD Jan Hendrik Müller**, Schuldienst seit 1996, 2003–2012 Lehrauftrag für Didaktik der Mathematik an der TU Dortmund (IEEM bei Prof. Dr. Hans-Wolfgang Henn). Schwerpunkte: Computereinsatz und freie Arbeitsformen im Mathematikunterricht. Fachdidaktische Veröffentlichungen seit 2003, seit 1998 in MINT-Initiativen aktiv, Unterrichtserfahrung mit CAS (TI-92, TI-Nspire, WXMaxima, Geogebra).
  `http://www.mathebeimueller.de`

- **StR Oliver Wagener**, Schuldienst seit 2005, 2013–2016 Moderator für Mathematik im Kompetenzteam, seit 2016 abgeordnet an die Universität Duisburg-Essen Didaktik der Mathematik (AG Bärbel Barzel). Schwerpunkte: Lehrerfortbildungen, Technologie im Mathematikunterricht (insbesondere GTR und CAS), Unterrichtserfahrung mit TI-84, TI-Nspire, seit 2005 aktiv im Netzwerk T^3 Teachers Teaching with Technologie.
  `https://www.uni-due.de/didmath/oliver_wagener`

- **Prof. Dr. Eva Zerz**, Professorin für Algebra am Lehrstuhl D für Mathematik der RWTH Aachen. Arbeitsgebiete: mathematische Kontrolltheorie, algebraische Systemtheorie, Netzwerktheorie, Anwendungen von computeralgebraischen Methoden, insbesondere Gröbnerbasen, in diesen Gebieten, z. B. Singular Control Library.
  `http://www.math.rwth-aachen.de/~Eva.Zerz`

Motion Polynomials and Planar Linkages

C. Koutschan
(Johann Radon Institute for Computational and Applied Mathematics, Österreichische Akademie der Wissenschaften)

christoph.koutschan@ricam.oeaw.ac.at

Introduction

We describe an application of computer algebra to the construction of mechanisms with certain prescribed properties. In the CAS Mathematica, we have implemented the package **PlanarLinkages**; it provides commands for constructing and visualizing planar linkages that draw a prescribed algebraic curve. The construction procedure is based on so-called motion polynomials; their basic arithmetic and a factorization algorithm is also provided by the package.

In order to state the problem more precisely, let us introduce some terminology. A *linkage* is a mechanical device consisting of rigid bodies (called *links*) that are connected by *joints*. We restrict our attention to *planar linkages*, i.e., to linkages all of whose links move in parallel planes. Moreover, we consider only *rotational joints*, which means that we don't allow *prismatic joints*. In Figure 1 two examples for this type of linkages can be seen: the first one has four degrees of freedom, while the second one has only a single degree of freedom (we say it has *mobility one*). If we move a linkage of mobility one, the trace of any point located on one of the links yields a bounded curve in the plane.

The problem of constructing a planar linkage that draws a finite segment of a given algebraic curve was first addressed and solved in full generality by Kempe [2]. While his construction is very elegant in theory, it yields quite complicated linkages in practice; see [3] for an implementation. In a recent article [1] the symbolic computation group at RICAM, including the author, designed a novel algorithm for basically the same problem. The advantage of the new algorithm is that it yields much simpler linkages: the number of links and joints is only linear in the degree of the curve. Moreover, it allows for a simple collision detection, which for general linkages is a very hard problem. The drawback of our method is that it is only applicable to bounded rational curves,

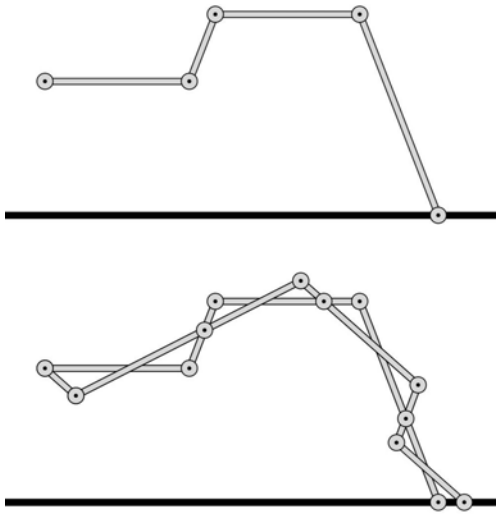


Figure 1: *An open chain linkage and its extension to a linkage of mobility one, both realizing the translational motion given by $P(t)$ in Equation* (4)*.*

i.e., to curves that are parametrizable by rational functions and that are contained in some disk of finite radius.

Theoretical Background

Before we describe our Mathematica package, we sketch the new method for constructing planar linkages and give a bit of theoretical background. The interested reader is referred to the paper [1] where all this is laid out in detail, and from which also the following simple example is taken: we consider the ellipse that is implicitly defined by the polynomial $(x+1)^2 + 4y^2 = 1$. The goal is to construct a linkage with rotational joints that draws this ellipse and that admits only one degree of freedom. More precisely, "drawing" means that there is a specific link (to which we attach the pen) that performs a motion along the ellipse while the linkage moves.

Mathematically speaking, a *motion* is a one-dimensional family of direct isometries (i.e., translations

and rotations). We denote by SE_2 the special Euclidean group, which is the set of direct isometries in the plane with composition as the group operation. For a convenient treatment in a computer algebra system, we encode direct isometries as elements of the noncommutative $\mathbb{R}$-algebra $\mathbb{K}$ of *dual complex numbers*:

$$\mathbb{K} = \mathbb{C}[\eta] \,/\, (\eta^2, i\eta + \eta i).$$

Its elements are of the form $z + \eta w$ with complex numbers $z, w \in \mathbb{C}$, and according to the defining relations, which can be seen as rewriting rules, they are multiplied as follows:

$$(z_1 + \eta w_1) \cdot (z_2 + \eta w_2) = z_1 z_2 + \eta\,(\overline{z_1} w_2 + z_2 w_1). \quad (1)$$

By defining on $\mathbb{K}$ the equivalence relation

$$k_1 \sim k_2 \;:\Longleftrightarrow\; k_1 = \alpha k_2 \text{ for some } \alpha \in \mathbb{R} \setminus \{0\}, \quad (2)$$

we can show that the multiplicative group

$$\{z + \eta w \in \mathbb{K} \mid z \neq 0\} \,/\sim$$

is isomorphic to SE_2; in Out[8] below the isomorphism is given explicitly. A univariate polynomial in $\mathbb{K}[t]$ gives rise to a one-dimensional family of direct isometries and is therefore called a *motion polynomial*. Motions that can be represented in this way are called *rational motions*. The algorithm we are going to describe takes as input a motion polynomial and outputs a planar linkage of mobility one realizing the corresponding rational motion. This task is slightly more general than drawing a rational curve, since also the orientation of the end effector is taken into account.

The ellipse $(x+1)^2 + 4y^2 = 1$ admits the rational parametrization

$$\varphi(t) = \Big(-\frac{2}{t^2+1},\, \frac{t}{t^2+1} \Big), \quad t \in \mathbb{R} \cup \{\infty\}, \quad (3)$$

from which one can read off that a translational motion along this ellipse is represented by the motion polynomial

$$P(t) = (t^2 + 1) + \eta\,(it - 2) \quad (4)$$

("translational" means that the orbit of *any* point under this motion is a translate of the ellipse). A motion polynomial $Z + \eta W \in \mathbb{K}[t]$ is called *bounded* if the complex polynomial $Z \in \mathbb{C}[t]$ does not have any real roots; the connection to the boundedness of the corresponding curve (the orbit of the origin) is established by the fact that Z appears as the denominator of the parametrization.

In order to construct a linkage that realizes the motion $P(t)$, we want to decompose it into simpler motions, namely into revolutions; these correspond exactly to motions that can be realized by a single (rotational) joint. We find [1, Lemma 4.3] that each linear motion polynomial, whose orbits are bounded, represents a revolute motion. Therefore, the desired decomposition is obtained by a factorization of P into linear polynomials.

In our example, however, one can easily check (e.g., by an ansatz with undetermined coefficients) that such a factorization does not exist. But this doesn't mean that we have to give up on drawing the ellipse! Recall that by the definition (2) of the equivalence relation $\sim$, the motion polynomial $RP \in \mathbb{K}[t]$ describes the same motion as P for any real polynomial $R \in \mathbb{R}[t]$. In this case we can take $R = t^2 + 1$ and observe that

$$R(t) \cdot P(t) = \big(t^4 + 2t^2 + 1\big) + \eta\,\big(it^3 - 2t^2 + it - 2\big)$$

indeed admits a factorization into linear polynomials:

$$\big(t + i - \eta\, i\big) \cdot \big(t - i + \tfrac{1}{2}\eta\, i\big) \cdot \big(t - i + \tfrac{3}{2}\eta\, i\big) \cdot (t + i). \quad (5)$$

In [1, Theorem 5.15] it is shown that for any bounded motion polynomial P such a real polynomial R exists, and an algorithm to compute R and the complete factorization of RP is described.

The factorization (5) allows us to construct a linkage, in the form of an open chain (upper part of Figure 1), whose links can move according to the revolutions represented by the linear factors. Since such a linkage has many degrees of freedom, we need to constrain its mobility. This is done by adding more links and joints (lower part of Figure 1), which is achieved by an iteration of the so-called flip procedure [1, Sections 6–7].

However, if we just want to draw the ellipse, we need not realize exactly the translational motion $P(t)$: it is enough to find a motion for which the orbit of one point is the ellipse. One can check that multiplying with a polynomial $C \in \mathbb{C}[t]$ from the left does not change the orbit of the origin. In our case, we find that the polynomial CP with $C(t) = t - i$ factors completely:

$$C(t) \cdot P(t) = \big(t - i - \tfrac{1}{2}\eta\, i\big) \cdot \big(t - i + \tfrac{1}{2}\eta\, i\big) \cdot \big(t + i + \eta\, i\big).$$

This factorization gives rise to a slightly simpler construction, which is depicted in Figure 2.

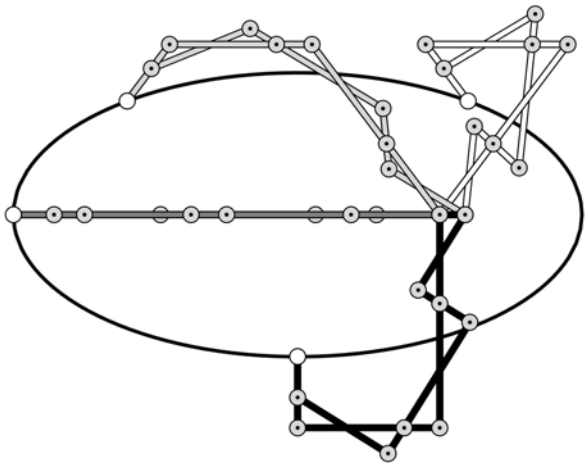


Figure 2: *A linkage drawing the ellipse* (3)*; it is shown in different positions:* $t = 2$ *(white),* $t = \frac{1}{2}$ *(light gray),* $t = 0$ *(dark gray), and* $t = -1$ *(black).*

The Mathematica Package

We now give a brief demonstration of our Mathematica package **PlanarLinkages**. The package, its source code, and a Mathematica notebook with some sample computations are freely available [4].

In[1]:= $\ll$ **PlanarLinkages.m**

> PlanarLinkages — © 2015 Christoph Koutschan
>
> This program comes with absolutely no warranty; it is free software, and you are welcome to redistribute it and/or modify it under the terms of the GNU General Public License (http://www.gnu.org/licenses/).

Motion polynomials are entered using the special symbol **eta** and Mathematica's **NonCommutativeMultiply** (written as **). As output we obtain a pretty-printed version of the motion polynomial, which internally is represented as a Mathematica expression with head **MP**.

In[2]:= $\boldsymbol{P = t + \mathrm{I} + \mathrm{eta} ** (2 - \mathrm{I})}$

Out[2]= $(\mathbb{i} + t) + \eta \cdot (2 - \mathbb{i})$

In[3]:= $\mathbf{FullForm}[\boldsymbol{P}]$

Out[3]= MP[Plus[Complex[0, 1], t], Complex[2, -1]]

Arithmetic can be done in the usual way, by taking into account the noncommutative multiplication.

In[4]:= $\boldsymbol{P + 1 + \mathrm{eta} ** \mathrm{I}}$

Out[4]= $((1 + \mathbb{i}) + t) + \eta \cdot 2$

In[5]:= $\boldsymbol{P ** (1 - \mathrm{eta}) ** P}$

Out[5]= $\left(-1 + 2\,\mathbb{i}\,t + t^2\right) + \eta \cdot \left(-1 + (4 - 2\,\mathbb{i})\,t - t^2\right)$

When executing the multiplication symbolically, we recover Equation (1):

In[6]:= $\mathbf{MP}[\boldsymbol{z_1, w_1}] ** \mathbf{MP}[\boldsymbol{z_2, w_2}]$

Out[6]= $z_1\,z_2 + \eta \cdot (\mathrm{Conjugate}[z_1]\,w_2 + w_1\,z_2)$

The command **ActR2** performs the action of an element $(x_1 + ix_2) + \eta(y_1 + iy_2) \in \mathbb{K}$, which itself represents a direct isometry in SE_2, on a point $(a, b) \in \mathbb{R}^2$, see [1, (4.2)].

In[7]:= $\mathbf{ActR2}[(\boldsymbol{x_1 + \mathrm{I}\,x_2}) + \mathrm{eta} ** (\boldsymbol{y_1 + \mathrm{I}\,y_2}),\ \{\boldsymbol{a, b}\}];$

In[8]:= $\mathbf{Simplify}[\%,\ \mathbf{Element}[\{\boldsymbol{x_1, x_2, y_1, y_2}\},\ \mathbf{Reals}]]$

Out[8]= $\left\{ \dfrac{a\,x_1^2 - a\,x_2^2 - 2\,b\,x_2\,x_1 + x_1\,y_1 - x_2\,y_2}{x_1^2 + x_2^2},\right.$
$\left. \dfrac{2\,a\,x_2\,x_1 + b\,x_1^2 - b\,x_2^2 + x_1\,y_2 + x_2\,y_1}{x_1^2 + x_2^2} \right\}$

The command **AnimateMP** visualizes the action of a motion polynomial; as a result we obtain an animation (not printed here!) showing a small triangle that moves according to the given motion. Since a linear bounded motion polynomial corresponds to a revolute motion, we can compute its fixed point. From the output, it becomes clear that the fixed point can only be given if the input polynomial is bounded, i.e., if the polynomial $t + z \in \mathbb{C}[t]$ has no real roots. In contrast, the motion polynomial $t + 1 + \eta$ corresponds to an unbounded motion, in this case a horizontal translation, and the attempt to compute its fixed point results in an error message.

In[9]:= $\mathbf{FixPoint}[\boldsymbol{t + z} + \mathrm{eta} ** (\boldsymbol{2w})]$

Out[9]= $\left\{ -\dfrac{\mathrm{Im}(w)}{\mathrm{Im}(z)}, \dfrac{\mathrm{Re}(w)}{\mathrm{Im}(z)} \right\}$

In[10]:= $\mathbf{Catch}[\mathbf{FixPoint}[\boldsymbol{t + 1} + \mathrm{eta}]]$

Out[10]= FixPoint: Input is not a normed bounded motion polynomial of degree 1.

Next, we provide a command to compute a factorization of a motion polynomial into linear factors; as a consistency check, we expand the result and obtain the original polynomial back.

In[11]:= $\mathbf{FactorMP}[(\boldsymbol{t} + \mathrm{I})\mathrm{\^{}}5 + \mathrm{eta} ** \boldsymbol{t}]$

Out[11]= $\left((\mathbb{i} + t) + \eta \cdot \frac{\mathbb{i}}{16}\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \frac{\mathbb{i}}{8}\right) \cdot$
$\left((\mathbb{i} + t) + \eta \cdot 0\right) \cdot \left((\mathbb{i} + t) + \eta \cdot \frac{\mathbb{i}}{8}\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \frac{\mathbb{i}}{16}\right)$

In[12]:= $\mathbf{Expand}[\%]$

Out[12]= $\left(t^5 + 5\,\mathbb{i}\,t^4 - 10\,t^3 - 10\,\mathbb{i}\,t^2 + 5\,t + \mathbb{i}\right) + \eta \cdot t$

If the polynomial itself cannot be factored, then the command automatically determines a minimal-degree real polynomial such that the product of the two polynomials factors completely.

In[13]:= $\boldsymbol{\mathrm{fact}} = \mathbf{FactorMP}[\boldsymbol{t}\mathrm{\^{}}2 + 1 + \mathrm{eta} ** (\mathrm{I}\,\boldsymbol{t} - 2)]$

FactorMP::R : Multiply the input with $R = 1 + t^2$

Out[13]= $\left((\mathbb{i} + t) + \eta \cdot \left(\mathrm{C}[2] - \frac{\mathbb{i}}{2}\right)\right) \cdot \left((-\mathbb{i} + t) - \eta \cdot \mathrm{C}[2]\right) \cdot$
$\left((-\mathbb{i} + t) + \eta \cdot \left(\mathrm{C}[1] + \frac{3\mathbb{i}}{2}\right)\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \mathrm{C}[1]\right)$

This factorization can now be used to construct a linkage, by calling the command **ConstructLinkage**. For this purpose we instantiate the free parameters C[1] and C[2], and give a "random" polynomial **rand** as second argument according to [1, Lemma 7.5]. While almost any choice of C[1], C[2], and **rand** yield a valid linkage, we can play with these parameters to influence the shape of the resulting linkage. For example, we can omit the second argument, in which case the program chooses a random polynomial, but the linkage then will usually look very "ugly", in the sense that some links are much longer than others. The output has to be understood as follows: each triple $\{i, j, p\}$ stands for "link i is connected to link j by a joint and their relative motion is given by the motion polynomial p", where the links are labeled with integers from 1 to 10.

In[14]:= $\boldsymbol{\mathrm{fact}} = \boldsymbol{\mathrm{fact}}\ /.\ \{\mathrm{C}[1] \to 0, \mathrm{C}[2] \to -\mathrm{I}/2\};$

In[15]:= $\boldsymbol{\mathrm{rand}} = \boldsymbol{t} + (9/5)\,\mathrm{I} + \mathrm{eta} ** 0;$

In[16]:= $\mathbf{L} = \mathbf{ConstructLinkage}[\boldsymbol{\mathrm{fact}}, \boldsymbol{\mathrm{rand}}]$

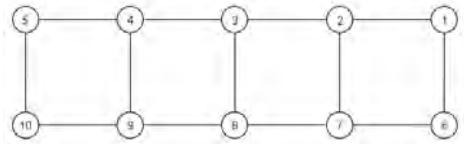
Out[16]= $\left\{\{1, 2, (\mathbb{i} + t) + \eta \cdot 0\}, \{2, 3, (t - \mathbb{i}) + \eta \cdot \frac{3\mathbb{i}}{2}\}, \right.$
$\left. \{3, 4, (t - \mathbb{i}) + \eta \cdot \frac{\mathbb{i}}{2}\}, \{4, 5, (\mathbb{i} + t) - \eta \cdot \mathbb{i}\}, \right.$

$$\{6, 7, (\mathbb{i}+t)-\eta\cdot\tfrac{45\mathbb{i}}{56}\}, \{7, 8, (t-\mathbb{i})+\eta\cdot\tfrac{3\mathbb{i}}{8}\},$$
$$\{8, 9, (t-\mathbb{i})+\eta\cdot\tfrac{41\mathbb{i}}{28}\}, \{9, 10, (\mathbb{i}+t)+\eta\cdot\tfrac{2\mathbb{i}}{7}\},$$
$$\{1, 6, \left(t+\tfrac{9\mathbb{i}}{5}\right)-\eta\cdot\tfrac{9\mathbb{i}}{28}\}, \{2, 7, \left(t+\tfrac{9\mathbb{i}}{5}\right)-\eta\cdot\tfrac{9\mathbb{i}}{8}\},$$
$$\{3, 8, \left(t+\tfrac{9\mathbb{i}}{5}\right)-\eta\cdot\tfrac{9\mathbb{i}}{4}\}, \{4, 9, \left(t+\tfrac{9\mathbb{i}}{5}\right)-\eta\cdot\tfrac{9\mathbb{i}}{7}\},$$
$$\{5, 10, \left(t+\tfrac{9\mathbb{i}}{5}\right)+\eta\cdot 0\}\}$$

Still, this mathematical description of a linkage is not very intuitive. To get an idea of how this linkage looks like and how it draws the ellipse, our package provides the command **ShowLinkage**, which offers a large variety of ways to visualize and animate linkages. For example, we can draw the *link graph*, which is a graph whose vertices correspond to the links and whose edges correspond to the joints of the linkage.

In[17]:= **ShowLinkage[L, Return → "graph"]**

Out[17]=



While the link graph gives information about the topological structure of the linkage, it completely hides its geometry. To see how a physical model of the linkage would look like, one can use the same command with different values of the **Return** option. This way one can get a two-dimensional picture (similar to the one in Figure 2) or the corresponding animation. By specifying **Return** → **"picture3D"** one obtains a three-dimensional drawing of the linkage as in Figure 3. Such a 3D graphics can be animated as well; we refer to our website [4] for some sample movies. Figure 3 also shows that the depicted linkage cannot move without self-collisions. Thanks to the special structure of the linkages constructed by our algorithm — the link graph will always be ladder-shaped as in Out[17] — these collisions can be detected by solving a relatively simple system of polynomial equations. The following computation shows that when we choose a certain spatial arrangement of the links, then there are only few collisions and all happen at the same position $t = \infty$. This is the position when the pen passes through the origin (the right-most point of the ellipse) and when all joints are located on the horizontal axis (compare Figure 2).

In[18]:= **ord = {2, 1, 6, 7, 3, 8, 4, 5, 10, 9};**

In[19]:= **ShowLinkage[L, Links → ord, Return → "collisions"]**

Out[19]= $\{\{\{2, 6, 7\}, \frac{2}{7}, \infty\}, \{\{3, 8, 4\}, \frac{54}{61}, \infty\},$
$\{\{3, 8, 4\}, \frac{6}{7}, \infty\}, \{\{4, 5, 9\}, \frac{5}{7}, \infty\}\}$

We conclude with an example that is motivated by a popular formulation of Kempe's theorem, stating that *"There is a linkage that signs your name"*, which is attributed to William Thurston. However, as remarked by O'Rourke [5], it is not very plausible that a concrete "signing linkage" has ever been constructed by Kempe's procedure due to its complexity, in terms of links and joints.

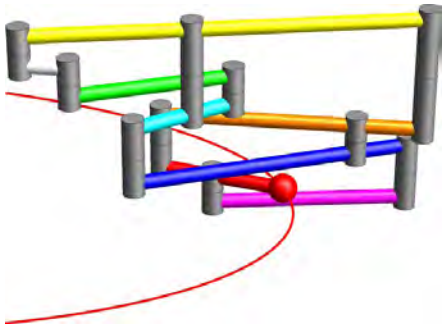


Figure 3: *Three-dimensional view of a linkage drawing the ellipse given by the parametrization* (3).

As an example to support his claim, O'Rourke points out that already constructing a linkage drawing the first letter "J" of John Hancock's famous signature on the United States Declaration of Independence would be a challenging task. The solution to this problem was found by our program and is depicted in Figure 4.

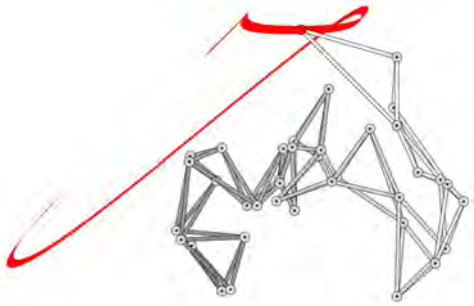


Figure 4: *A rational curve approximating the "J" in John Hancock's signature and a linkage drawing it using a quill pen whose shape is a line segment in direction* $(5, 6)$.

References

[1] Matteo Gallet, Christoph Koutschan, Zijia Li, Georg Regensburger, Josef Schicho, and Nelly Villamizar. Planar linkages following a prescribed motion. *Mathematics of Computation*, 2016. To appear (preprint on arXiv:1502.05623), DOI: 10.1090/mcom/3120.

[2] Alfred B. Kempe. On a general method of describing plane curves of the n^{th} degree by linkwork. *Proceedings of the London Mathematical Society*, s1-7(1):213–216, 1876.

[3] Alexander Kobel. Automated generation of Kempe linkages for algebraic curves in a dynamic geometry system. Bachelor's thesis, University of Saarbrücken, 2008.

[4] Christoph Koutschan. Mathematica package PlanarLinkages and electronic supplementary material for the paper "Planar linkages following a prescribed motion", 2015. Available at `http://www.koutschan.de/data/link/`.

[5] Joseph O'Rourke. How to fold it. Cambridge University Press, Cambridge, 2011.

Normaliz: A package for linear diophantine systems, polyhedra and lattices

W. Bruns
(Universität Osnabrück, Institut für Mathematik, Germany)

wbruns@uos.de

The very first version of Normaliz [5] was meant to compute normalizations of affine monoids (or semigroups), hence the name. Over the years it has been extended to a powerful package for discrete convex geometry. We explain its main computation goals by a simple example, sketch the mathematical background and discuss the basic steps in the Normaliz primal algorithm. Some remarks on the technical aspects and the history conclude this overview.

The main computation goals

Suppose we are interested in the following type of 3×3 matrices

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

and the problem is to find nonnegative integer values for $x_1, \dots, x_9$ such that the 3 numbers in all rows, all columns, and both diagonals sum to the same constant $\mathcal{M}$. Sometimes such matrices are called *magic squares* and $\mathcal{M}$ is the *magic constant*. This leads to a system of 7 diophantine linear equations:

$$\begin{aligned} x_1 + x_2 + x_3 &= x_4 + x_5 + x_6; \\ \dots &= \dots \\ x_1 + x_2 + x_3 &= x_3 + x_5 + x_7. \end{aligned} \tag{6}$$

Our goal is to understand the set of solutions in nonnegative integers.

Since the equations are linear homogeneous, and the positive orthant is closed under addition, the sum of two solutions is again a solution (the magic constants add up). It follows that the set M of all solutions is a monoid (or semigroup). In a systematic framework we can describe it as follows:

$$M = C \cap L$$

where C is a pointed cone in $\mathbb{R}^d$ and L is a sublattice of $\mathbb{Z}^d$. In our case $d = 9$, C is the positive orthant, and L is the lattice of integer solutions to the system (6). There are other ways to describe M, but this choice of C and L is exactly Normaliz' approach to the problem. Two natural questions suggest themselves:

> (*Generation*) Is the monoid of magic squares finitely generated, and if so, what is a system of generators?
>
> (*Enumeration*) Given $\mathcal{M}$, how many magic squares of magic constant $\mathcal{M}$ are there?

The answers to *Generation* and *Enumeration* are the main computation goals of Normaliz.

Some background

The introductory example is a rather special case of the objects for which Normaliz solves *Generation* and *Enumeration*: since version 3.0 it can be applied to arbitrary intersections of rational polyhedra and affine lattices. In other words, Normaliz is a solver for systems of affine-linear diophantine inequalities, equations and congruences.

Nevertheless, for the sake of simplicity, in this overview we stick to the special case $M = C \cap L$, and specialize it even further, following Normaliz' course of computation. In our example we can first introduce coordinates in the solution space of the system (6), and then intersect the positive orthant with it. After this reduction step, we can assume that we want to compute the integer points in a d-dimensional pointed cone $C \subset \mathbb{R}^d$. This is the core case to which Normaliz reduces the given input data by preliminary transformations. Let us first solve *Generation* for cones:

Theorem 1 (Minkowski-Weyl) Let $C \subset \mathbb{R}^d$. Then the following are equivalent:

1. There exist (integer) vectors $x_1, \dots, x_n \in \mathbb{R}^d$ such that

$$C = \{x \in \mathbb{R}^d : x = \alpha_1 x_1 + \cdots + \alpha_n x_n, \\ \alpha_1 \dots, \alpha_n \geq 0\}.$$

2. There exist linear forms $\lambda_1, \dots, \lambda_s$ (with integer coefficients) on $\mathbb{R}^d$ such that

$$C = \{x \in \mathbb{R}^d : \lambda_i(x) \geq 0,\ i = 1, \dots, s\}.$$

If the equivalent conditions of the theorem are satisfied, C is called a (*rational*) *cone*. (Sometimes the attribute *polyhedral* is added.) The cone C is *pointed* if it does not contain a linear subspace of positive dimension. In this case the elements of a minimal set $x_1, \dots, x_n$ of generators are unique up to permutation and multiplication by positive scalars. We call them *extreme rays*. The scalar is 1 in the integer case if we require that the coefficients are coprime.

The dimension of a cone is the dimension of the vector subspace it generates. If $\dim C = d$, then the linear forms λ_i are unique up to permutation and positive scalar multiples if the system is minimal, and again the scalar must be 1 if we require that their coordinates are coprime integers. The linear forms $\lambda_1, \dots \lambda_s$ then define the (relevant) *support hyperplanes* $H_i = \{x : \lambda_i(x) = 0\}$, and Theorem 1(2) represents C as an intersection of the (positive) *halfspaces* $H^+_{\lambda_i} = \{x : \lambda_i(x) \geq 0\}$.

The conversion from generators to support hyperplanes is usually called *convex hull computation* and the converse is called *vertex enumeration*. Both directions are completely equivalent since they amount to the dualization of a cone.

That *Generation* makes sense also for lattice points is guaranteed by

Theorem 2 (Gordan's lemma) Let $C \subset \mathbb{R}^d$ be a rational pointed cone. Then the monid $M = C \cap \mathbb{Z}^d$ is finitely generated. More precisely, it has a unique minimal system E of generators.

A set $\{z_1, \dots, z_m\} \subset M$ generates the monoid M if every element of M is a linear combination of $z_1, \dots, z_m$ with nonnegative integer coefficients. The unique minimal generating set $\mathrm{Hilb}(C)$ (or $\mathrm{Hilb}(M)$) in Theorem 2 is called the *Hilbert basis*

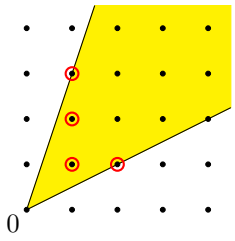


Figure 1: *A Hilbert basis*

of C (or M). We do not see a good historical reason for this nomenclature – the term "Gordan basis" would be much more appropriate. In the restricted setting that we have reached, we can now reformulate *Generation* as follows:

Compute the Hilbert basis of a rational cone C.

The Hilbert basis is the set of *irreducible* elements in M, i.e., elements x that can not be written in the form $x = y + z$ with $y, z \in M$, $y, z \neq 0$. This proves uniqueness, and is important for the reduction of a generating set to the Hilbert basis.

Let us turn to *Enumeration*. For it we need a *grading*, a $\mathbb{Z}$-linear form $\deg : \mathbb{Z}^d \to \mathbb{Z}$. For the next theorem we restrict the generality even further.

Theorem 3 (Hilbert; Ehrhart) Let $d \geq 1$. Suppose that the extreme rays of the d-dimensional cone C have degree 1 and set $M = C \cap \mathbb{Z}^d$. Then the lattice point enumerator

$$H(M, k) = \#\{x \in M : \deg x = 1\}$$

is given by a polynomial q_M with rational coefficients for all $k \geq 0$. Equivalently, the generating function $\sum_{k=0}^{\infty} H(M, k) t^k$ defines a rational function of type

$$H_M(t) = \frac{1 + h_1 t + \cdots + h_u t^u}{(1-t)^d}, \\ h_1, \dots, h_u \in \mathbb{Z},\ u < d.$$

The polynomial q_M is called the *Hilbert polynomial* and $H_M(t)$ the *Hilbert series* of M – for very good reason since the theorem can be derived from the theory of graded algebras, a key observation of Stanley. In the combinatorial context it was independently proved by Ehrhart, and therefore one speaks of the *Ehrhart polynomial* and *Ehrhart series* as well. The h_i are nonnegative; this follows from Hochster's theorem by which the monoid algebra $K[M]$ is Cohen-Macaulay for any field K, but it can also be shown combinatorially.

Without the hypothesis on the degree of the extreme rays, the theorem must be reformulated: the polynomial is only a quasipolynomial in general, i.e., a "polynomial" with coefficients that are periodic in k, and the denominator takes a more complicated form.

The leading coefficient of the Hilbert/Ehrhart polynomial q_M has the form $e(M)/(d-1)!$ with a positive integer $e(M)$ that is called the *multiplicity* of M. It is the lattice normalized volume of the polytope spanned by the extreme rays. Moreover, $e(M) = 1 + h_1 + \cdots + h_u$. Clearly

Compute the Hilbert series of M

is the right formulation of *Enumeration* now.

The magic squares continued

The input to Normaliz for the computation of the magic squares is encoded as follows:

```
amb_space 9
equations 7
1 1 1 -1 -1 -1  0  0  0
...
1 1 0  0 -1  0 -1  0  0
grading
1 1 1 0 0 0 0 0 0
```

The first equation reads

$$x_1 + x_2 + x_3 - x_4 - x_5 - x_6 = 0,$$

as desired, and the other equations are encoded analogously. The magic constant is the grading. If only equations are given, but no inequalities, Normaliz assumes that the nonnegative solutions should be computed. We look at some data in the output file:

```
5 Hilbert basis elements
5 Hilbert basis elements of degree 1
4 extreme rays
4 support hyperplanes
embedding dimension = 9
rank = 3
...
grading:
1 1 1 0 0 0 0 0 0
with denominator = 3
```

The input grading is the magic constant. However, as the denominator 3 shows, the magic constant is always divisible by 3, and therefore the effective degree is $\mathcal{M}/3$. This degree is used for the multiplicity and the Hilbert series.

```
multiplicity = 4
Hilbert series:
1 2 1
denominator with 3 factors:
1: 3
degree of Hilbert Series ... = -1
Hilbert polynomial:
1 2 2
with common denominator = 1
```

The Hilbert series and the Hilbert polynomial are

$$H_M(t) = \frac{1 + 2t + t^2}{(1-t)^3} \quad \text{and} \quad q_M(k) = 1 + 2k + 2k^2,$$

and after substituting $\mathcal{M}/3$ for k we obtain the number of magic squares of magic constant $\mathcal{M}$, provided 3 divides $\mathcal{M}$.

```
5 Hilbert basis elements of degree 1:
0 2 1 2 1 0 1 0 2
1 0 2 2 1 0 0 2 1
1 1 1 1 1 1 1 1 1
1 2 0 0 1 2 2 0 1
2 0 1 0 1 2 1 2 0
```

The 5 elements of the Hilbert basis represent magic squares. We show the first, third and fifth:

0	2	1
2	1	0
1	0	2

1	1	1
1	1	1
1	1	1

2	0	1
0	1	2
1	2	0

All other solutions are linear combinations of these squares with nonnegative integer coefficients.

Actually we were lucky: if we increase the format of the squares to 4×4 or higher, the extreme rays are no longer of degree 1. Normaliz computes the Hilbert basis and Hilbert series quickly for 5×5 squares, and the Hilbert basis (of $522,347$ vectors) for 6×6 squares in reasonable time, but the Hilbert series for 6×6 is out of reach.

The primal algorithm

For the computation of Hilbert bases Normaliz provides two algorithms at the user's disposal. Here we restrict ourselves to the *primal algorithm* that also computes the Hilbert series if this is desired. For the *dual algorithm* we refer the reader to Bruns and Ichim [3].

The primal algorithm is based on triangulations. We assume that the cone C is pointed and defined by a generating set. (If it is defined by inequalities, Normaliz first computes the extreme rays.) After some preliminary transformations we can further assume that C has dimension d and that $\mathbb{Z}^d$ is the lattice to be used. Then the primal algorithm proceeds in the following steps:

1. Fourier-Motzkin elimination computing the support hyperplanes of C;
2. pyramid decomposition and computation of the lexicographic triangulation Δ;
3. evaluation of the simplicial cones in the triangulation:
   (a) enumeration of the set of lattice points E_σ in the fundamental domain of a simplicial subcone σ,
   (b) reduction of E_σ to the Hilbert basis $\mathrm{Hilb}(\sigma)$,
   (c) Stanley decomposition for the Hilbert series of $\sigma \cap L$;
4. Collection of the local data:
   (a) reduction of $\bigcup_{\sigma\in\Delta} \mathrm{Hilb}(\sigma)$ to $\mathrm{Hilb}(C \cap L)$,
   (b) accumulation of the Hilbert series of the monoids $\sigma \cap L$.

This is the true chronological order only for small examples. Typically the steps are interleaved in a complicated way. We now explain some steps in more detail.

Convex hulls and triangulation

Fourier-Motzkin elimination allows us to compute the support hyperplanes incrementally by extending the cone $C' = \mathbb{R}_+x_1 + \cdots + \mathbb{R}_+x_{n-1}$ to $C = \mathbb{R}_+x_1 + \cdots + \mathbb{R}_+x_n$. Suppose that

$$C' = H^+_{\lambda_1} \cap \cdots \cap H^+_{\lambda_r}$$

for linear forms $\lambda_1, \ldots, \lambda_r$. We may assume that

$$\lambda_i(x_n) \begin{cases} = 0 & i = 1, \ldots, p, \\ > 0 & i = p+1, \ldots, q, \\ < 0 & i = q+1, \ldots, r. \end{cases}$$

Set

$$\mu_{ij} = \lambda_i(x_n)\lambda_j - \lambda_j(x_n)\lambda_i, \quad \begin{matrix} i = p+1, \ldots, q, \\ j = q+1, \ldots, r. \end{matrix}$$

Theorem 4 The cone $C = \mathbb{R}_+x_1 + \cdots + \mathbb{R}_+x_n$ is the intersection of the halfspaces defined by the linear forms λ_i, $i = 1, \ldots, q$, and μ_{ij}, $i = p+1, \ldots, q$, $j = q+1, \ldots, r$.

The theorem leads to an extremely simple algorithm. However, one must discard the superfluous ones among the μ_{ij}, and this needs some care; see Bruns and Ichim [3] for the strategies of Normaliz.

The extension of a triangulation of C' to a triangulation of C is easy as well. A triangulation is a face-to-face decomposition of C into simplicial cones, i.e., cones whose extreme rays are linearly independent.

Theorem 5 Let Δ' be a triangulation of C'. Then we obtain a triangulation Δ of C by adding to Δ' all simplicial cones $(\sigma \cap H) + \mathbb{R}_+x_n$ where σ runs through Δ and H is a support hyperplane of C' such that x_n belongs to the negative halfspace defined by H.

The triangulations computed by Theorem 5 are called *lexicographic* or *pushing*.

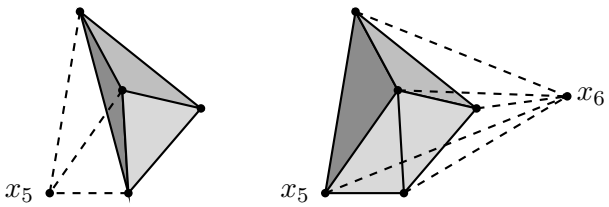


Figure 2: *A lexicographic triangulation of a cone in dimension* 4

Applied algorithmically, both theorems require a double loop, and especially the rapid growth of triangulations in high dimensions forbids a direct application of Theorem 5. We replace it by a hybrid approach based on *pyramid decomposition*: instead of computing all the intersections $\sigma \cap H$, $\sigma \in \Delta'$, we triangulate $(H \cap C') + \mathbb{R}_+x_n$ directly. Pyramid decomposition can be applied both recursively and in parallel for several H.

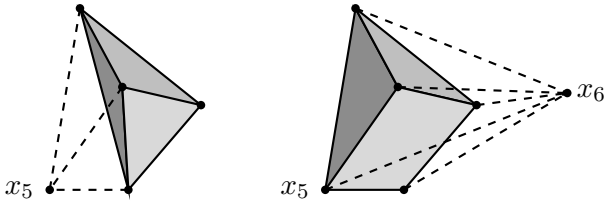


Figure 3: *Pyramid decomposition of Figure 2*

Also the computation of support hyperplanes profits from it. We refer the reader to Bruns, Ichim and Söger [6] for a detailed description of the strategy applied by Normaliz.

Reduction, simplicial cones and Hilbert bases

It is characteristic for Hilbert basis algorithms that they first compute a superset of the Hilbert basis and then reduce it by discarding reducible elements:

Theorem 6 Let $C \subset \mathbb{R}^d$ be a pointed rational cone, and let G be a system of generators of the monoid $M = C \cap \mathbb{Z}^d$. Then

$$\mathrm{Hilb}(C) = \{x \in G : x - y \notin C \text{ for all } y \in G, x \neq y\}.$$

Fortunately the condition $x - y \notin C$ can be tested quickly if the linear forms λ_i defining the support hyperplanes of C are known (and there are not too many of them): $x - y \notin C \iff \lambda_i(x) < \lambda_i(y)$ for at least one i. Nevertheless the reduction algorithm wants to be well-organized; see [3].

Suppose Σ is a triangulation of C. Then the union of the Hilbert bases $\mathrm{Hilb}(\sigma)$, $\sigma \in \Sigma$, is evidently a system of generators for M, and it must "only" be reduced. So it remains to explain how $\mathrm{Hilb}(C)$ is computed if C is a simplicial cone.

Let $v_1, \ldots, v_d \in \mathbb{Z}^d$ be linearly independent, generating the simplicial cone C. Set

$$\mathrm{par}(C) = \{x : x = \alpha_1v_1 + \cdots + \alpha_dv_d, \\ 0 \leq \alpha_i < 1,\ i = 1, \ldots, d\}.$$

The set $\mathrm{par}(C)$ is a semiopen parallelotope. It is a fundamental domain for the action of $U = \mathbb{Z}v_1 + \cdots + \mathbb{Z}v_d$ on $\mathbb{R}^d$ by parallel translation.

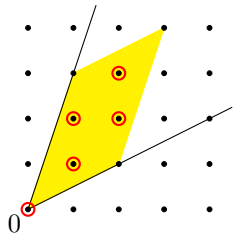


Figure 4: *The parallelotope* $\mathrm{par}(C)$ *and the set* E'

Theorem 7

1. The set $E' = \mathrm{par}(C) \cap \mathbb{Z}^d$ represents the residue classes of $\mathbb{Z}^d$ modulo the subgroup U.
2. $\#E' = |\det(v_1, \ldots, v_d)|$.
3. $E = \{v_1, \ldots, v_d\} \cup E'$ generates the monoid $C \cap \mathbb{Z}^d$.

The theorem shows that one can compute the elements of E' by enumerating the residue classes of $\mathbb{Z}^d$ and division with remainder. We have taken great care to make this process as efficient as possible.

The sets E' are reduced "locally" for all simplicial cones in the triangulation, and then their union is reduced "globally". This concludes the computation of the Hilbert basis by the primal algorithm.

Normaliz applies a strategy of *partial triangulation* which is based on pyramid decomposition. It tries to avoid the triangulation of those pyramids that can only yield previously known Hilbert basis elements. See Bruns et al. [2]. Often partial triangulation has an overwhelming effect.

With notation as above, let L be the monoid generated by $v_1, \dots, v_d$. Then $\bigcup_{x \in E'} x + L$ is a disjoint (!) decomposition of the monoid $M = C \cap \mathbb{Z}^d$, and this allows one to write down the Hilbert series of M immediately:

$$H_M(t) = \frac{\sum_{x \in E'} : t^{\deg x}}{(1-t^{g_1}) \cdots (1-t^{g_d})}, \qquad \begin{matrix} g_i = \deg v_i, \\ i = 1, \dots, d. \end{matrix}$$

The next step is not so easy, since a general cone is not the disjoint union of the simplicial cones in a triangulation. Fortunately there exist a disjoint decomposition into semiopen simplicial cones, and their Hilbert series are almost as easy to compute as those of closed simplicial cones. For the details we refer the reader to [6].

Use of Normaliz

Normaliz is implemented as a front end, called `normaliz`, and the kernel `libnormaliz`. The latter is a C++ class library and serves as an API. It uses GMP for infinite precision arithmetic and a small part of the Boost library. Parallelization is based on OpenMP. We provide binaries for Linux, MacOS and MS Windows.

The communication with the frontend `normaliz` is via an input file. Normaliz then returns the computation results in one or several output files. There is a multitude of input types for generators, equations, inequalities, congruences and gradings.

The computation goals and several other options can be set on the command line or in the input file. In particular, the computation goals can be restricted, for example to the support hyperplanes, to the lattice points in a polytope or the multiplicity. Normaliz tries to do its computation in 64 bit arithmetic, and if this precision fails, it automatically switches to infinite precision.

Normaliz has interfaces to CoCoA, GAP, Macaulay 2, polymake and Singular. Especially the GAP interface is very extensive, and at present GAP is the best environment for interactive access to Normaliz. The GUI jNormaliz (by Vicinius Almendra and Bogdan Ichim) is also very helpful.

Of course Normaliz has found many applications in the areas for which it has been made, commutative algebra, toric geometry, polytope theory and integer optimization. But since it is a solver for linear diophantine systems, it can be applied everywhere where such systems must be solved. A prime example is Burton's topological package Regina. The very efficient triangulation algorithm is applied in the package SecDec by Borowka et al. for the computation of multiscale integrals.

Normaliz has an offspring NmzIntegrate that computes weighted Ehrhart series and integrals of polynomials over rational polytopes [8].

Performance data of Normaliz can be found in [6].

History

The first version of Normaliz was developed by the author and his PhD student Robert Koch as a C program in 1998–2001 (see [7]). As the implementation did not allow any extensions, Normaliz was transferred to C++ by Bogdan Ichim in 2007–2008. On this basis the present code has been written by Christof Söger and the author since 2009. In 2014 Richard Sieg joined the team and made some contributions. The team is also supported by Tim Römer. The present published version is 3.1.1.

The first interface to Normaliz was the Singular library written by the author in 2002. The Macaulay 2 package was developed by Gesa Kämpf, and we owe the GAP interface to Sebastian Gutsche, Max Horn and Christof Söger. The CoCoA interface is due to John Abbott, Anna Bigatti and Christof Söger.

The list of references below has been restricted to primary references to Normaliz and its algorithms. We trust that the reader will be able to locate all mentioned software packages. The sources [1] and [6] contain extensive lists of references.

Acknowledgement. In 2013–2016 the development of Normaliz was supported by the DFG SPP 1489 "Algorithmische und experimentelle Methoden in Algebra, Geometrie und Zahlentheorie".

References

[1] W. Bruns and J. Gubeladze, *Polytopes, rings and K-theory*, Springer, 2009.

[2] W. Bruns, R. Hemmecke, B. Ichim, M. Köppe, and C. Söger, *Challenging computations of Hilbert bases of cones associated with algebraic statistics*. Exp. Math. **20** (2011), 25–33.

[3] W. Bruns and B. Ichim, *Normaliz: Algorithms for affine monoids and rational cones.* J. Algebra **324** (2010), 1098–1113.

[4] W. Bruns, B. Ichim, T. Römer, R. Sieg and C. Söger: Normaliz. Algorithms for rational cones and affine monoids. Available at `http://normaliz.uos.de`.

[5] W. Bruns, B. Ichim, T. Römer, R. Sieg and C. Söger: Normaliz. Algorithms for rational cones and affine monoids. Available at `http://normaliz.uos.de`.

[6] W. Bruns, B. Ichim and C. Söger. *The power of pyramid decompositions in Normaliz*. J. Symb. Comp. **74** (2016), 513–536.

[7] W. Bruns and R. Koch, *Computing the integral closure of an affine semigroup.* Univ. Iagel. Acta Math. **39** (2001), 59–70.

[8] W. Bruns and C. Söger, *Generalized Ehrhart series and integration in Normaliz*. J. Symb. Comp. **68** (2015), 75–86.

The SYMBOLICDATA Project – Maturing the Computer Algebra Social Network Perspective

H.-G. Gräbe
(Universität Leipzig)

graebe@informatik.uni-leipzig.de

Introduction

In [1, 2, 3, 4] we described the roots and goals of the SYMBOLICDATA Project and also basic Linked Data and RDF principles. In this note we report about advances of the project during the last year. Our main efforts were directed to strengthen and consolidate the Computer Algebra Social Network (CASN) part.

In March 2016 we released SYMBOLICDATA version 3.1 including

- new data from different CA subcommunities,
- a reorganized git repo structure,
- close integration of the CASN part into the main project and
- a set of examples based on the *EasyRDF* PHP library and the Bootstrap web framework to show how to set up web presentations of the data in a very simple way.

The focus of the SYMBOLICDATA project moved from a project mainly centered around activities of the German CA Fachgruppe to a broader international scope. This opening started with a presentation of the project in the Work in Progress section of the 2014 CICM conference. It was furthermore discussed with the SIGSAM chair at the ACA 2015 conference in Kalamata, where *Hans-Gert Gräbe* gave a presentation in combination with a hands on session on SYMBOLICDATA[1]. The contacts were deepened by *Albert Heinle* and *Victor Levandovskyy* with a paper [5] about the SDEval framework in the *ACM Communications in Computer Algebra*.

In 2016 we had the opportunity to present and discuss various aspects of SYMBOLICDATA and more general perspectives of a digital research infrastructure for the CA community at the session *Information Services for Mathematics: Software, Services, Models, and Data* at ICMS-2016 in Berlin (organized by Wolfram Sperber, FIZ Karlsruhe, and Michael Kohlhase, Jacobs University Bremen) and at the session *Information Services for Mathematical Software, Models, and Research Data* at ACA-2016 in Kassel (organized by Hans-Gert Gräbe, University of Leipzig, Albert Heinle, University of Waterloo, and Wolfram Sperber, FIZ Karlsruhe). A more detailed report on these discussions will appear in a forthcoming publication.

Enlarging the SYMBOLICDATA Database

With SYMBOLICDATA 3.1 we consolidated the integration of several data sources into our main data and metadata collection that were available so far only in a draft version.

With that integration the conceptual design of the SYMBOLICDATA database changed from a data store to a metadata store – the new data collections provide (only) metadata information about the core research data that is hosted and maintained in a separate remote data store by a CA subcommunity. The CA subcommunity provides expertise to maintain the research data in a semantically correct way and provides metadata in "raw form". The SYMBOLICDATA team collects that metadata, transforms it into RDF and prepares it for search and filter processing.

Note that such a design change was enabled by the consistent transformation of the metadata collected so far along RDF principles during preparation of SYMBOLICDATA version 3. RDF requires a strong distinction between data and metadata whereas the data structure design of CA research data usually handles metadata as "data extension" and stores it together with the primary data in a common file.

Our new conceptual approach along that lines supports the formation of an *interlinked distributed rese-*

[1] See our publications and presentations overview at http://wiki.symbolicdata.org/Publications for more details.

arch data infrastructure within the CA community and between its several subcommunities. We realized that in many cases CA subcommunities (in particular subcommunities developing specialized research CA software systems) have well established research data infrastructures with no need to be duplicated, but interlinking these infrastructures is yet a challenge.

In the following subsections we describe the advances at the "data frontier" in more detail.

Fingerprints for Polytopes and Groups

In [2] we announced draft versions of RDF based resource descriptions (i.e., metadata or *fingerprints*, see [3]) of Fano and Birkhoff polytopes collected by Andreas Paffenholz as part of the *polymake* data store and of transitive groups from the *Database for Number Fields* collected and stored in a similar way by Jürgen Klüners and Gunter Malle.

The draft versions were extracted from the primary data sources and transformed into RDF based *fingerprints* by *Andreas Nareike* in 2013. The metadata was provided as part of the primary data in different formats within the respective remote collections. The derived metadata is now integrated into the SYMBOLICDATA main database. Each such metadata record contains a link to the corresponding data record within the remotely maintained research data store provided by the respective CA subcommunity.

As main advantage such separated metadata can be queried in a common, uniform and well established way using the SPARQL query language. This is another W3C web standard with many tools and concepts, which were mainly developed for a performant management of big data given in the RDF semantic web format and for integration with other applications.

SPARQL plays a similar role for querying the worldwide distributed and interlinked semantic web data store as SQL plays for querying local databases. Using that technology one can navigate within such data, restructure it for efficient search or identify a given example within the database. We refer to our wiki[2] for more background and some example queries.

Transforming Test Sets into the Normaliz Format

In a similar, we transformed and enlarged the data on integer programming – the SYMBOLICDATA test sets collection.

The old SYMBOLICDATA test sets collection was compiled by *Raymond Hemmecke* several years ago along the former SYMBOLICDATA rules – develop a data model, an XML-binding for data storage to represent this special data type, and an RDF ontology for metadata (fingerprints and maintenance information).

The former exclusive usage of XML-bindings for data representations was inspired by the success and wide usage of XML as a unified way to represent data in other application areas at that time, in particular influenced by the upcoming MathML standard. Meanwhile XML is much less prominent for exchange practices of structured data, and within the redesign of SYMBOLICDATA we decided to accept and use data also in other formats.

Such a decision also was inspired by the observation that specialized CA software comes with well a defined input data format, and within subcommunities using a common software the data are stored and exchanged in just that format. Hence for such a subcommunity (as, e.g., the polytope subcommunity around *polymake* or the integer programming subcommunity around *Normaliz*) there is no need to develop another standard for data exchange – such a standard would hardly be accepted. Note that the situation is different within the polynomial systems solving subcommunity since there exist several major software systems with different input formats, as *Singular*, *Macaulay2*, *Magma*, *GB* or *CoCoA* – to name only the most important ones.

For the new test sets collection we use and store data in the *Normaliz* exchange format and thus prepared the data in a similar way as for polytopes and transitive groups. This work was done by *Tim Römer* who transformed also the "legacy" test sets into the new format. All content written in the old format was cleaned up from the repos and the web pages were adjusted.

Towards a Computer Algebra Social Network

Motivation

All parties want to have a powerful digital research infrastructure, but they are rarely willing or able to invest in it. It is a complex social challenge to organize active goal-oriented cooperation in such an area outside the scientific reputation process. We learned over the years not only to concentrate on the collection of scientific *data* but also on structured and semantically enriched information about the scientific and social *processes* to produce this data.

Several years ago the SYMBOLICDATA Project extended its scope to analyze and support the exchange of such information in a structured way. Our vision is a distributed and tool based network of semantic aware nodes corresponding to the (small and big) nodes of the real CA research network. Such a *Computer Algebra Social Network* (CASN) should be a semantically enriched digital infrastructure for a social network of scientific research and scientific researchers within the Computer Algebra community and its several subcommunities similar to other social networks as, e.g., Facebook.

Note that the starting point for such a CASN is at least in two ways different from the Facebook starting point. First, there is no Marc Zuckerberg nor such an amount of money to push the project. Second, there is already a digital "CA memory" – a huge number of very loosely related web pages about conferences, meetings, working groups, projects, private and public repositories, private and public mailing lists etc. The CASN design has to take such a diversity into account and deve-

[2] See, e.g., `http://wiki.symbolicdata.org/MoreQueries`.

lop a decentralized solution based on modern semantic technologies. This solution must increase the awareness of the different parts of that already existing "CA network" and supports the *exploration* of that network to get useful deep results in an easy manner.

CASN Nodes

For a proper CASN design it is essential to exploit the potential of concepts, tools and standards of the fast growing distributed Linked Open Data (LOD) Cloud[3]. Pascal Hitzler emphasizes the importance of such a coordinated conceptual approach to set up an interoperably interlinked digital universe, since "with the omnipresence and availability of data from different times, locations, perspectives, topics, cultures, resolutions, qualities, and so forth, *exploration* becomes an additional (4th) paradigm of science" [6].

As a first step towards a digital network within the CA community capable to *explore* social and scientific relations

- we operate the RDF based SYMBOLICDATA main data store together with its SPARQL endpoint [8] to query centrally maintained data and
- propose to convert other nodes of the "CA memory" into CASN nodes that provide part of their data in structured RDF format.

RDF principles neither require such nodes to be uniformly structured nor running on big web resources. LOD sources are self-explanatory by design and its structure can be explored with appropriate RDF tools by interested third parties at run time to prepare to fetch the information in a structured way. Hence efforts to present and explore data within such a CASN network can be shared in a wide scope between data providers and data consumers.

In a first version such a node can be even only a directory with valuable RDF files publicly accessible in the web as provided by the CASN sample node[4] of the SYMBOLICDATA Project. As proof of concept we provide detailed information about five CA conferences in the subdirectory `Conferences` using the (meanwhile outdated) *Semantic Web Conference Ontology*[5].

The CASN node of the German CA Fachgruppe[6] is designed along a more advanced concept. During the revision of concepts towards SYMBOLICDATA 3.1 we consequently redesigned this data to form a proper CASN node with publicly accessible but locally maintained RDF sources of (almost) all structured information displayed on the web site of the German CA Fachgruppe. This information is explored by a special plugin and rendered in its Wordpress based web presentation[7]. Hence one can explore both the "pure" information in standard RDF notation to embed it into third party web workflows as *interlinked data* and in the "old fashion" as *hyperlinked text*. Note that the technical realization is unpretentious – the RDF data is stored as plain files in RDF/XML format in the CASN node and the plugin uses the *EasyRDF* PHP library and the Wordpress Shortcode mechanism for rendering. No advanced technique as RDF store or SPARQL endpoint has to be set up. The code is mirrored as best practice example in our *maintenance* Git repo.

CA Conferences

As another service within the CASN we maintain a list of *Upcoming Conferences*. The data about conferences is extracted from several sources, transformed into RDF format and delivered by our main SPARQL endpoint [8]. This information is used by the German CA Fachgruppe on one hand to present an online list of upcoming conferences and on the other hand to generate the conference announcement section of the printed version of their CA Rundbrief. The RDF database contains more advanced information about conferences as, e.g., submission deadlines or program committees.

We run this service in a draft version for several years already and compiled from it a list of (at the moment 166) *Past Conferences*. In summer 2016 this data was enhanced with additional data about past conferences supplied by the SIGSAM web team and extended by a *Conference Series* concept from the SIGSAM collection. The SIGSAM collection provides structured information about such conference series (description and publication rules) in an (almost) unstructured way that was transformed to structured RDF using predicates `sd:description` and `sd:publicationRules`. Not to duplicate information without reason we use the standard predicate `rdfs:seeAlso` to link with the corresponding part in the SIGSAM conference series web page for additional information.

The SYMBOLICDATA People Database

The concept of the Unique Resource Identifier (URI) as part of the RDF standard provides a generic way to disambiguate people and artifacts. More precisely, each such URI, considered as *digital identity*, is the entry point from the real world to the digital universe, and any statement within the digital universe can be followed and traced back using digital technology only up to such (combinations of) URIs. URIs are bound to real world entities by more complex socio-political and technical "agreements". To shape politically such "agreements" is the real core of digital privacy.

The use of URIs provides an easy way to assign digital facts to special digital identities and thus solve the *disambiguation problem* – a great problem in the text oriented "hyperlinked universe" that required powerful text mining so far. One of the great challenges to acade-

[3] `http://lod-cloud.net`
[4] `http://symbolicdata.org/rdf/`
[5] `http://data.semanticweb.org/ns/swc/swc_2009-05-09.html`
[6] `http://www.fachgruppe-computeralgebra.de/rdf/`
[7] `http://www.fachgruppe-computeralgebra.de/`

mic content providers within the transformation of their digital universes is *author disambiguation*. Such disambiguation is required to, e.g., assign URIs of publications to the correct author URIs. Most of the academic content providers come up with own solutions for their own universe, i.e., for the provider's internal data collection that counts as its main "capital". Interoperability between providers remains a great challenge since it requires to interlink data sources that are very private from a business point of view. While this Gordian knot is hard to cut from a provider's position, a comparatively small scientific community could solve that interoperability challenge by a common effort – develop its own People Database, i.e., its own URI system for people and provide dictionaries to the part of the URI systems of the different providers relevant to their academic scope.

This is the goal of the SYMBOLICDATA People Database for the CA community. As one of the benefits of such a disambiguation one can track reputation and merits more precisely querying the whole SYMBOLICDATA database or even interlinking it with other RDF based sources within the Linked Open Data Cloud. Moreover, people within the CASN can systematically provide and update information about their own scientific activities.

Currently the SYMBOLICDATA People Database contains more than 1200 entries, i.e., digital identities of scientists that are active in the area of Computer Algebra. These URIs were mainly extracted from program committee lists of different conferences or (in a restricted scope) from lists of authors of accepted papers.

As standard information we provide personal information as instance of `foaf:Person` with (a subset of) keys `foaf:name`, `foaf:homepage` and `sd:affiliation` (a literal). Due to privacy reasons we do not provide `foaf:mbox` (email) values. This list is steadily enlarged and used as URI reference for reports about different activities (invited speakers, conference organizers, program committees etc.) in other parts of our CASN database.

As proof of concept we aligned our URIs in a common task with the "Zentralblatt" with their author disambiguation system and produced more than 300 `sd:hasZBMathAuthorID` matches. This work was done in 2014 on an early version of the SYMBOLICDATA People Database and can be queried from our RDF store, too. In a near future we plan to update that alignment with "Zentralblatt". The concept can easily be extended to other content providers (in particular to the ACM people database or the MathSciNet author disambiguation system) that are interested in such a cooperation.

The CA Dissertations Project

The CA Rundbrief of the German CA Fachgruppe maintains a section with reports about dissertations in Computer Algebra finished in working groups within the Fachgruppe. We made the metadata available also in RDF within the CASN node of the Fachgruppe and display it at their web site. Within the discussions with SIGSAM in summer 2016 we realized that there is a large data pool of similar information collected by SIGSAM for years that could be integrated into a common database of dissertations in Computer Algebra. For the moment we moved the existing RDF data about dissertations to the SYMBOLICDATA main data store and aligned the presentation in the web site of the German CA Fachgruppe accordingly.

The CA Systems Project

In summer 2016 we also intensively discussed perspectives of the swMATH project [7]. In particular we considered ways to popularize it to a larger audience (within the CA community) and discussed to what extend RDF principles and LOD alignment could support such a popularization. We agreed that it would be helpful to represent a core part of the swMATH metadata in RDF, provide URIs with a consistent naming scheme, and publish this data as Linked Open Dataset to achieve better visibility within the semantic web community. Such a metadata extraction also makes the alignment with other overviews on CA systems as, e.g., the one maintained by SIGSAM, much easier.

A first prototypical draft version of such an RDF based overview on *CA systems* extracted from the swMATH database was compiled during our discussions in summer 2016 and is available in our RDF store. We also set up a prototypical view on that data within the SYMBOLICDATA info pages[8].

Additionally, we discussed whether the swMATH data model has to be redesigned better to reflect subtleties as the relation between CA systems and CA packages or different versions of the same system. All these questions require much deeper analysis. Since RDF can be used in a consistent way to express modeling aspects a Linked Open Dataset as just described could support also such a discussion.

Advances in the SYMBOLICDATA Infrastructure

In October 2015 we converted our main git repo[9] to an organizational account. With SYMBOLICDATA version 3.1 we reorganized our git repo structure and set up several new repos with different maintenance rules.

- *data* – the data repo with a single master branch mainly to backup recent versions of data, no versioning,
- *code* – code directory with master and develop branches, under versioning,
- *maintenance* – code chunks from different tasks and demos how to work with RDF based data, no versioning,

[8] `http://wiki.symbolicdata.org/info`
[9] `https://github.com/symbolicdata`

- *publications* – as a backup store of the LaTeX sources of SymbolicData publications, only master branch, no versioning,

- *web* – as an extended backup store of the SymbolicData web site that provides useful code to learn how RDF based data can be presented in the web.

The main development is coordinated by the SymbolicData *Core Team* (Hans-Gert Gräbe, Ralf Hemmecke, Albert Heinle) with direct access to the organizational account.

The repos *maintenance* and *web* are intended to show best practice code for using the RDF based data of the SymbolicData project. In particular, the *maintenance* repo contains a mirror of the Wordpress plugin code used by the German CA Fachgruppe and the transformation code developed by Andreas Nareike in 2013 for polytopes and groups databases. To use the code you may fork the repo, but there is almost no reason to pull code back. If you have a valuable contribution please contact the Core Team to discuss how that contribution can be added to the project.

The repo *data* is mainly for backup purposes. If you plan to add valuable data to the project please contact the Core Team to discuss how that contribution can be added. We provide help to put the data in an appropriate Linked Open Data format.

The repo *publications* is used mainly for reference and not intended for public additions. We provide LaTeX sources of our papers and slides and also information about the review processes of our work since reviews provide many valuable suggestions for the further development of our project. The repo *code* contains several coding subprojects concerning SymbolicData tools for various purposes.

Literatur

[1] Hans-Gert Gräbe, Simon Johanning, Andreas Nareike. The SymbolicData Project – from Data Store to Computer Algebra Social Network. *Computeralgebra-Rundbrief*, 55:22–26, 2014.

[2] Hans-Gert Gräbe, Simon Johanning, Andreas Nareike. The SymbolicData Project – Towards a Computer Algebra Social Network. *Workshop and Work in Progress Papers at CICM 2014*. CEUR-WS.org, vol. 1186, 2014.

[3] Hans-Gert Gräbe. Semantic-aware Fingerprints of Symbolic Research Data. In Gert-Martin Greuel, Thorsten Koch, Peter Paule, Andrew Sommese (Eds.). *Mathematical Software – ICMS 2016*. Volume 9725 of Lecture Notes in Computer Science, page 411–418, 2016.

[4] Hans-Gert Gräbe. The SymbolicData Project – a Community Driven Project for the CA Community. Talk given at the ACA 2016 Session "Information services for mathematical software, models, and research data." `http://symbolicdata.org/Papers/aca-16.pdf`. [2016-09-11]

[5] Albert Heinle, Viktor Levandovskyy. The SDEval Benchmarking Toolkit. *ACM Communications in Computer Algebra*, 49.1:1–10, 2015.

[6] Pascal Hitzler, Krzysztof Janowicz. Linked Data, Big Data, and the 4th Paradigm. *Semantic Web*, 4.3:233–235, 2013.

[7] swMATH – an Information Service for Mathematical Software. `http://swmath.org/about_contact`. [2016-09-16]

[8] The SymbolicData SPARQL Endpoint. `http://symbolicdata.org:8890/sparql`. [2016-09-11]

CAS-unterstütztes Assessment von Mathematik

M. Kallweit
(Ruhr-Universität Bochum)

michael.kallweit@rub.de

Digitales Assessment

Die steigende Verbreitung von eLearning-Angeboten ist nicht nur den neuen Lehr- und Lernmöglichkeiten geschuldet. Oftmals sind es knappe Ressourcen und der Wunsch die herkömmliche Lehre zu entlasten, die den Einsatz digitaler Tools vorantreiben. Neben didaktisch und medial aufbereiteten Materialien (welche ohne neuen Aufwand immer wieder verwendet werden können) sind es Routine-Übungsaufgaben, die man in elektronische Systeme auslagern möchte.

Mathematikfertigkeiten, die klassischerweise über Papieraufgaben abgefragt und von einer Lehrperson korrigiert werden, digital zu überprüfen, stellt hohe Anforderungen an das zugrundeliegende technische System.

Wirft man einen Blick in die digitale Landschaft vorhandener Lernmaterialien und Aufgaben, so werden am häufigsten geschlossene Antwortformate verwendet. Bei diesen werden dem Nutzer Antwortmöglichkeiten präsentiert und es muss die richtige auswählt werden. Neben Multiple-Choice zählen auch Zuordnungs- und Sortieraufgaben zu diesem Typus. Gefordert wird hierbei vom Nutzer eine (alleinige) Identifikations- und Selektionsleistung.

Andere wichtige Fähigkeiten, wie das Konstruieren von mathematischen Objekten (z.B. ein gesuchter algebraischer Term), verlangen nach weitergehenden Formaten, den offenen Aufgaben. Die Antwort muss dabei selbst erdacht und in ein System eingegeben werden. Nahezu alle eLearning-Plattformen bieten die Möglichkeit eine freie Eingabe mittels Stringvergleich oder mit regulären Ausdrücken auszuwerten. Für die Eingabe von Zahlen lassen sich numerische Toleranzbedingungen einstellen. Eine Behelfsmöglichkeit für die Abfrage von Termen lässt sich z.B. im System MUMIE [4] finden. Dort wird die Übereinstimmung eines eingegebenen Funktionsterms mit der Musterlösung durch eine numerische Auswertung der Terme an vielen zufälligen Stellen getroffen.

Einen Überblick über vorhandene computerunterstützte Assessmentsysteme findet man in [2].

Unterstützung durch CAS

Flexibilität bei der Behandlung symbolischer Ausdrücke gewinnt man durch den Einsatz von Computeralgebrasystemen. Die verbreitetsten Learning-Management-Systeme (LMS) sind Client-Server-Anwendungen, wobei als Client zumeist der Internetbrowser fungiert. Ein CAS kann hier auf der Seite des Clients (z.B. auf Javascript-Basis) oder serverseitig eingebracht werden. Robuster und aus prüfungstechnischer Sicht sicherer ist der Einsatz eines CAS im Backend, weshalb die meisten Systeme diesen Weg wählen.

STACK

Die Software STACK ist ein quelloffenes Assessmentsystem für mathematische Fragestellungen, welches im Hintergrund das CAS Maxima [6] verwendet. STACK lässt sich in verschiedene Lernplattformen einbinden; zur Zeit stehen Plugins für Moodle [7] und Ilias [3] zur Verfügung. Die Möglichkeiten von STACK werden in [1] beschrieben und im Nachfolgenden genauer vorgestellt.

Möglichkeiten zur Antwortauswertung

Die einfachste Art der Auswertung ist der Vergleich mit einer hinterlegten Musterlösung. Da sich jedoch mathematische Objekte mit unendlich vielen verschiedenen Textausdrücken beschreiben lassen, ist hierbei der Rückgriff auf die Methoden des CAS zur Vereinfachung und Normalisierung von Ausdrücken unabdingbar. Abbildung 1 beschreibt einen einfachen Algorithmus.

```
IF simplify(input - solution) = 0
   THEN mark := 1
   ELSE mark := 0
```

Abbildung 1: *Auswertung durch Vergleich mit Lösung*

Je nach didaktischem Konzept der Aufgabe sind weitere Verfahren zur Eingabeüberprüfung nutzbar. STACK kennt verschiedene Tests auf Gleichheit, wie z.B. algebraische Äquivalenz von Termen, Äquivalenz von Gleichungen (auch falls verschiedene Variablen verwendet wurden) und Gleichheit des Ableitungsbaums des CAS-Parsers. Hinzukommen Tests, die vielfältige formale Aspekte überprüfen können (u.a. ob ein Bruch vollständig gekürzt ist, oder die Eingabe komplett ausmultipliziert ist), verschiedene numerische Tests, Methoden für Zeichenketten (wie reguläre Ausdrücke) und

für einige spezielle Situationen (wie Differentiation und Integration):

Gleichheit: CasEqual, EqualComAss, AlgEquiv, SubstEquiv, SameType, SysEquiv

Form: LowestTerms, Expanded, FacForm, SingleFrac, PartFrac, CompletedSquare

Numerik: NumRelative, NumAbsolute, NumSigFigs, GT, GTE

Zeichenketten: String, StringSloppy, RegExp

Weitere: Factorisation of polynomials, Diff, Int

Damit wird ein breites Spektrum an tiefergehenden Aufgaben möglich. Die Beispielaufgabe in Abbildung 2 zeigt eine offene Modellierungsfrage, in der zu einem genannten Problem eine Gleichung aufgestellt und gelöst werden muss.

Eine Seite eines Rechtecks ist $5cm$ länger als die andere. Der Flächeninhalt ist $24cm^2$. Gesucht sind die Seitenlängen.

1. Geben Sie eine Gleichung ein, die die Seitenlängen mit dem Flächeninhalt in Beziehung bringt.
2. Geben Sie die Lösungen ihrer Gleichung aus 1 ein.
3. Geben Sie die Seitenlängen des Rechtecks ein.

Abbildung 2: *Eine Modellierungsaufgabe*

Auch Realisierungsaufgaben, in denen der Nutzer gefordert ist, ein Beispiel für einen mathematischen Sachverhalt einzugeben, lassen sich umsetzen, siehe Abbildung 3.

Geben Sie ein Beispiel für eine Funktion $f : \mathbb{R} \to \mathbb{R}$ an, die an der Stelle $x = 0$ stetig, aber nicht differenzierbar ist.

Abbildung 3: *Eine Realisierungsaufgabe*

Rückmeldebäume und individuelles Feedback
Doch mit dem Rückgriff auf das CAS lässt sich noch weit mehr bewerkstelligen. So ist es in dem STACK-Aufgabentyp möglich, zunächst mathematische Eigenschaften der Nutzereingabe als Zwischenergebnisse zu berechnen und von diesen die weitere Auswertung abhängig zu machen. STACK verwaltet die Struktur dieser Verzweigungen in einem sogenannten Rückmeldebaum, ein Beispiel ist in Abbildung 4 gegeben.

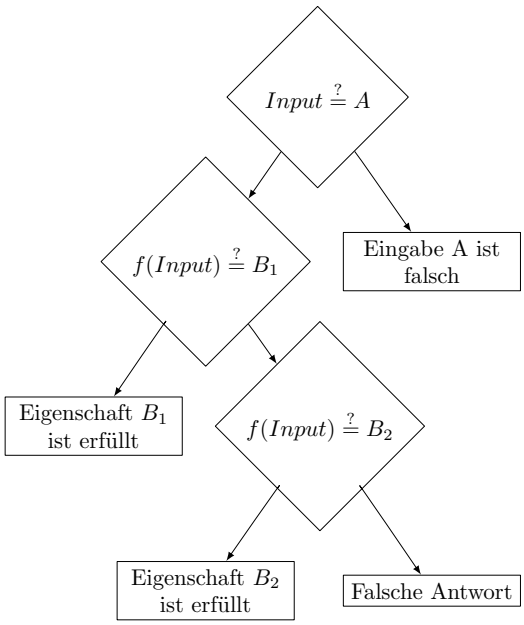


Abbildung 4: *Rückmeldebaum*

An jedem Knoten des Baums kann man wählen, welcher der oben genannten Tests mit der Eingabe oder den berechneten Zwischenergebnissen ausgeführt wird. Dabei lässt sich zusätzlich für jede der beiden Verzweigungen eines Knotens (und damit insbesondere bei den Blättern) eine (Teil-)Bepunktung festlegen. Beim formativen Assessment als Lernszenario kann dieses durch entsprechendes textliches Feedback ergänzt werden.

In der Beispielaufgabe 3 könnte man zur Bewertung der Nutzereingabe zunächst die beiden Eigenschaften *stetig bei 0* und *differenzierbar bei 0* mittels CAS bestimmen und dann entsprechende Bepunktungen ausgeben. Dabei kann die textliche Rückmeldung an den Nutzer, wenn gewünscht, diese Auswertungen aufgreifen und beispielsweise rückmelden, dass die eingegebene Funktion zwar stetig an der Stelle 0, aber auch differenzierbar ist (was sie laut Aufgabenstellung nicht sein sollte).

Die bei einer Auswertung durchlaufenen Knoten ergeben die Gesamtpunkte und das Gesamtfeedback.

Randomisierte Aufgaben
Neben den Vorteilen zur Auswertung und der Generierung eines individuellen Feedbacks, lässt das CAS sich auch direkt bei der Erzeugung der Aufgabe nutzen. In STACK können sogenannte Aufgabenvariablen definiert werden um Vorberechnungen vorzunehmen. Noch vor der Anzeige des Aufgabentextes werden diese Variablen durch die konkreten vom CAS berechneten Ergebnisse ersetzt. So lassen sich zur Laufzeit konstruierte mathematische Objekte für eine Aufgabe jedes mal neu erzeugen. Bereitgestellte Zufallsfunktionen ermöglichen eine automatische Variation, was gezielt für didaktische Zwecke genutzt werden kann. Jeder Lernende bekommt seine eigene Aufgabenvariante. Kooperation mit anderen ist nur noch über die zugrundeliegenden Konzepte möglich. Ein naives Abschreiben funktioniert nicht mehr.

Möchte der Lehrende Musterlösungen bereitstellen,

können diese direkt mit dem Gesamtfeedback ausgegeben werden. Auch hier kann das CAS genutzt werden, um diese aus der (randomisierten) individuellen Aufgabenstellung zu generieren.

Anforderungen an das CAS

Um die zuvor beschriebenen Funktionalitäten zu erreichen, muss das verwendete CAS einige Voraussetzungen erfüllen. Maxima hat sich hier als vielfach verwendete Lösung herauskristallisiert. Als weitere Beispiele neben STACK sind hier das System LON-CAPA [5] und das hochschulübergreifende Studienorientierungsportal Studifinder [8] zu nennen.

Im Praxiseinsatz sind die technischen Anforderungen von entscheidender Bedeutung. Ein schneller Start des CAS und zügige Verarbeitung der Anfragen des LMS müssen gewährleistet sein. Als Lisp-Programm kann Maxima zur Laufzeit mit vollständig geladenem STACK-Code als Snapshot gespeichert werden, was einen schnellen Wiederaufruf zur Folge hat. Soll das System zeitgleich viele parallele Nutzeranfragen verarbeiten, lässt es sich als Cluster mit mehreren Instanzen betreiben. Da sich Anfragen ans CAS mit der Zeit durchaus wiederholen können (insbesondere bei mehrfacher Verwendung der gleichen Aufgaben), kann ein ausgeklügeltes Cachingverfahren für weiteren Geschwindigkeitszuwachs sorgen.

Auf mathematisch funktioneller Seite bedingen die bereitgestellten Funktionen des CAS die Möglichkeiten zur Aufgabengestaltung.

Selbst bei dem einfachen Auswertungsalgorithmus in Abbildung 1 sind hohe Ansprüche an das CAS gestellt. Die Eingabe des Nutzers kann beliebig kompliziert und komplex sein. Terme mit geschachtelten Wurzeln und trigonometrische Funktionen zeigen oftmals die Grenzen der Vereinfachungsfunktionen[10] auf.

Tests, die die Form einer Eingabe bemessen, sind darauf angewiesen bestimmte automatische Berechnungen (wie Zusammenfassung gleicher Variablenpotenzen) lokal abzuschalten. Grundlegende Regeln wie die Kommutativität der Addition und Multiplikation, behindern die Überprüfung eines bestimmten geforderten Formates (z.B. eine Normalform für Funktionsklassen). Manchmal kann gerade die Anwendung einer elementaren Handlung wie Ausmultiplizieren das eigentliche Testziel sein.

Die Verzweigungen im Rückmeldebaum basieren vielfach auf mathematischen Eigenschaften der Eingabe. Hierfür sind Prädikatsfunktionen ungemein hilfreich. Methoden um z.B. die Stetigkeit oder Differenzierbarkeit einer Funktion oder die Diagonalisierbarkeit einer Matrix boolesch zu entscheiden werden benötigt.

Im Idealfall sind zwei zufällig generierte Aufgabenausprägungen didaktisch äquivalent, d.h. werden von Personen vergleichbarer Fähigkeit auch gleich gut bearbeitet. Dies stellt eine besondere Herausforderung für die Aufgabenkonstruktion dar. Zum Teil wird weitergehende Mathematik benötigt, um die gewünschten mathematischen Objekte mittels CAS zu erstellen (man denke hier an *schöne* Vektoren für das Gram-Schmidtsche Orthonormalisierungsverfahren). Funktionen, um solche zufälligen Objekte direkt zu erzeugen (bestenfalls mit Bedingungen), entlasten den Aufgabenersteller.

Ausblick

STACK findet zunehmend Verwendung in Schulen und Hochschulen. Übergreifende Nutzernetzwerke bauen gemeinsame Aufgabendatenbanken auf. Viele interessante Weiterentwicklungen, wie mehrsprachige Aufgaben oder Eingaben über dynamische Geometriesoftware, werden durch internationale Institutionen eingebracht. Es ist davon auszugehen, dass STACK oder ähnliche CAS-basierende Systeme in Zukunft noch weitere Verbreitung erfahren werden und den Bereich digitalen Testens weiter bereichern werden.

Literaturverzeichnis

[1] M. Kallweit, Mathematik-Kompetenzen überprüfen und fördern - Automatisiert Lehren und Lernen mit STACK, *Tagungsband zum Workshop der ASIM/GI-Fachgruppen*, Argesim Report AR 50, 2015

[2] C. J. Sangwin, Computer Aided Assessment of Mathematics, Oxford University Press 2013.

[3] `www.ilias.de/docu/goto_docu_cat_4119.html`

[4] `www.integral-learning.de`

[5] `www.lon-capa.org/cas.html`

[6] `maxima.sourceforge.net`

[7] `moodle.org/plugins/qtype_stack`

[8] `www.studifinder.de`

[10] Ganz zu Schweigen vom Fehlen einer klaren Definition von *einfach*.

Ein Begleitpraktikum für das erste Studienjahr Mathematik

W. Plesken
(RWTH Aachen)

plesken@mathb.rwth-aachen.de

Einleitung

Im Folgenden soll ein kurzer Erfahrungsbericht über ein Begleitpraktikum gegeben werden, welches an der RWTH seit einigen Jahren für die Bachelorstudenten des ersten Studienjahrs Mathematik durchgeführt wird. Es ist eng abgestimmt auf die Einführungsvorlesungen Analysis und Lineare Algebra und erfordert ein hohes Engagement aller Beteiligten. Das Praktikum war eine der Veranstaltungen, die von dem früheren Diplomstudiengang Computermathematik in das neue Bachelor-Master-Programm übernommen wurde, allerdings von drei auf zwei Semester gekürzt und auch im Stundenumfang leicht reduziert.

Ziele und Durchführung des Praktikums

Bei der Aufzählung der Ziele des Begleitpraktikums wird schnell klar, dass man diese Ziele nicht zum Nulltarif erreicht, aber es hat sich gezeigt, dass sie erreichbar sind:

- Heranführen des Studierenden an ein angemessenes Arbeitsniveau und regelmäßiges Arbeitspensum,
- Erzeugung von Begeisterung bei den geeigneten Studenten,
- schnelle Aufgabe des Mathematikstudiums bei ungeeigneten Studenten,
- anspruchsvolle Nachhilfe zu den Hauptvorlesungen,
- eng gekoppelte Verständniskontrolle ohne Ausweichmöglichkeit,
- Einübung des Umgangs mit angebotenen Hilfen,
- Entwicklung der sprachlichen Ausdrucksfähigkeit mathematischer Sachverhalte,
- zügige Sozialisierung und Hinführung zur Teamarbeit,
- Überwindung des saisonalen Lernens: Lern-für-die-Klausur-danach-vergiss,
- Momenterzeugung für Selbständigkeit nach einem Jahr.

Diese Ziele werden erreicht durch ein wöchentliches MAPLE-Worksheet, welches vom Dozenten nach Abstimmung mit den Dozenten der Hauptvorlesung mit einem Theorie- und einem Aufgabenteil sowie einem Musterlösungsteil mit Regieanweisungen für die Mitarbeiter vorbereitet wird. Zwar werden die Notationen aus den Hauptvorlesungen in der Regel nicht benutzt, aber es werden Verbindungen zwischen den Inhalten der beiden Vorlesungen aufgezeigt. Die Studierenden haben Anwesenheitspflicht für zwei Semesterwochenstunden, in denen sie die Worksheets bearbeiten, auf sehr intensive Verständnishilfe von älteren Studierenden zurückgreifen können und ihr wöchentliches Testat durch kompetente Mitarbeiter erhalten können. Die Verzahnung mit den Hauptvorlesungen ist eng, oftmals werden nützliche Zusätze geboten, jedoch wird möglichst nie im Stoff vorgegriffen. Angestrebt sind Verständnis und Problembewusstsein für den Stoff des ersten Jahres, nicht hingegen das einfache Abliefern schriftlicher Übungsaufgaben. Zwar können wir nicht verhindern, dass Studierende Lösungen ihrer Kommilitonen übernehmen, aber wir können sicherstellen, dass sie verstanden haben, was sie vorzeigen, indem sie Verständnisfragen zu ihrer Arbeit beantworten müssen. Die Studierenden werden angehalten, mit Hilfe von Maple zu visualisieren, zu rechnen und zu experimentieren.

Ich habe die Erfahrung machen müssen, dass die Ziele des Praktikums oft missverstanden werden. Deshalb stelle ich hier gegenüber, was das Praktikum erreichen will und tun soll, und was nicht. Es ist kein Maple-Programmierkurs, sondern dient zur Verinnerlichung mathematischer Grundkonzepte. Es ist kein Analysis- und/oder Lineare-Algebra-Praktikum, sondern vermittelt grundlegende Tatsachen und schlägt Brücken zwischen verschiedenen Gebieten. Es ist kein Klausurentraining der Hauptvorlesungen, sondern stellt sicher, dass grundlegende Begriffe mit angemessenen Vorstellungen im Gehirn abgespeichert werden. Es ist kein unverbindliches Zusatzangebot, sondern ist

auf intensivste Betreuung angewiesen, um Frustration zu vermeiden. Es ist nicht als Erholungsphase gedacht und wird auch nicht als solche wahrgenommen, erzeugt aber dankbare Studierende nach Abschluss des Praktikums. Es ist keine Nebenbeschäftigung für apathisch-gutmütige Mitarbeiter als Testatgeber, sondern erfordert hochdisziplinierten Einsatz und produziert Ermüdungserscheinungen nach einigen Jahren. Es ist mit Erstellung eines Jahrgangs von Worksheets für den Dozenten nicht erledigt, sondern muss jedesmal neu angepasst und erweitert werden.

Erfahrungen

Die Anzahl der Studierenden, die das erste Jahr angemessen überstehen, hat sich mit Einführung des Praktikums und einiger anderer Änderungen im Kurrikulum in etwa verdoppelt. Die kurze Rückkopplung von Stoffvermittlung und Verständnisüberprüfung ohne Ausweichmöglichkeit wird während des Praktikums teils als stressig, teils als heilsam empfunden. Es bilden sich schnell Arbeitgruppen. Die Hilfsassistenten sind hochmotiviert und werden von den Studenten immer sehr gelobt. Es werden keine Genies erzeugt, aber der Studierende lernt, sein eigenes Verständnis kritisch zu hinterfragen. Allgemeine Computerprobleme sind meist nach sechs Wochen überwunden. Generell wird eine hohe, jedoch deutlich leistungsabhängige Motivation erzeugt. Die anonyme Korrektur des Computers (Programm läuft nicht oder falsch) erzeugt wenig Reibung.

Aus der Sicht der Mitarbeiter ist Folgendes anzumerken. Die knapp bemessene Einzeltestatzeit erfordert große Disziplin: Der Testatgeber sagt falsch oder richtig und hat keine Zeit für Erklärungen. Der Organisationsaufwand ist erheblich. Die Zeit für Korrekturlesen und Umfangsanpassungen von Worksheets ist knapp. Es ergibt sich reichlich Kontakt mit Studierenden, was aber eine sehr strikte Regelung erfordert. Stimmung, Schwierigkeiten etc. müssen schnell an Dozenten weitergegeben werden. Die Gutmütigkeit des Testatgebers ist umgekehrt proportional zu seinem Nutzen. Schließlich muss der Dozent absolut von der Sache überzeugt sein.

Abschlusskommentare

Die Philosophie des Praktikums ist vergleichbar mit der des “jumpmath”-Projektes, welches leider noch nicht in Deutschland angekommen ist, vgl.

`http://jumpmath.org/`

zu dieser Methode und ihren sensationellen Erfolgen in der Grundschule: Am Anfang alles genau erklären, damit bei den Langsameren kein Frust entsteht, die Schüler alles selbst ausprobieren lassen, damit sie sich selbst begeistern können, nichts akzeptieren lassen, was nicht verstanden wurde. Leider ist man am Anfang des Studiums nicht am Anfang der Prägungsphase, sodass eingeschliffene Gewohnheiten eliminiert werden müssen. Aber die Erfahrung zeigt, dass doch noch positive Bewegungen möglich sind, die allerdings - wie immer in Spätphasen der Entwicklung - stark abhängig sind vom Hintergrund der Betroffenen.

Polynome mit ganzzahligen Null-, Extrem- und Wendestellen

J. Meyer
(Hameln)

j.m.meyer@t-online.de

Einleitung

Mitunter sucht man Polynome f, deren besondere Stellen ganzzahlig sind, d.h. bei denen f, f' und f'' jeweils ganzzahlige Nullstellen haben. Solche Polynome sollen hier „ganzheitlich“ genannt werden. Für den Grad 2 ist deren Auffinden völlig problemlos, Grad 3 lässt sich mit CAS-Hilfe schnell klären, und bei Grad 4 versagt ein (naiv aufgesetztes) CAS. Der Einsatz eines CAS verführt dazu, relativ theoriearm mit Hilfe von „brute force and ignorance“ die Maschine einfach rechnen zu lassen. Allerdings können die dadurch erzielten Resultate durchaus nachträglich zu Überlegungen führen, ob der große Rechneraufwand tatsächlich notwendig gewesen wäre. Dies wird anhand von Grad 3 erläutert.

Insofern will dieser Artikel einen Beitrag leisten zur Diskussion, ob im Schulunterricht der Einsatz von CAS mathematische Überlegungen sogar verhindern kann. Andererseits lässt sich dieser Artikel natürlich auch auffassen als Hilfe für Lehrende, die ganzheitliche Polynome suchen. Selbstverständlich kann man den Inhalt dieses Beitrages auch mit Lernenden, die das frei erhältliche CAS „Maxima“ lernen wollen, erarbeiten.

Kubische Polynome

Da die Nullstellen der gesuchten kubischen Polynome ganzzahlig sein sollen, bietet es sich an, diese in faktorisierter Form anzusetzen. Man erleichtert dem Rechner die spätere Sucharbeit, wenn 0 eine der Nullstellen ist, denn wenn man Polynome entlang der Argumentachse ganzzahlig verschiebt, ändert sich nichts an deren Ganzheitlichkeit. Sind u und v die beiden anderen Nullstellen, setzt man

$$f(x) = x \cdot (x-u) \cdot (x-v)$$

an. Ein anderer höchster Koeffizient als 1 würde an der Ganzheitlichkeit nichts ändern. Da zusammenfallende Nullstellen das Problem stark vereinfachen würden, soll

$$0 < u < v$$

vorausgesetzt werden. Nun kann man von einem CAS überprüfen lassen, für welche Wahlen von u und v die Ableitungen f' und f'' ganzzahlige Nullstellen haben. Das mag bei kubischen Polynomen f noch gut leistbar sein; bei Polynomen vom Grad 4 hat man es dann mit kubischen Gleichungen zu tun, und das sieht nach Programmieraufwand aus. Aber warum Gleichungen lösen? Stattdessen kann man das CAS auch alle Werte zwischen 0 und v überprüfen lassen, ob sie Nullstellen der beiden Ableitungen von f sind (Abb. 1).

```
n:50$
for u:1 thru n do
for v:u+1 thru n do
 block
  (f(x):=x*(x-u)*(x-v),
   define(f1(x), diff(f(x), x)),
   define(f2(x), diff(f1(x), x)),
   ESt:makelist(),
   WSt:makelist(),
   for x:1 thru n do
     if f1(x)=0 then ESt:endcons(x, ESt),
   if length(ESt)=2
      then for x:ESt[1] thru ESt[2] do
        if f2(x)=0 then WSt:endcons(x, WSt),
   if length(ESt)*length(WSt)=2
      then print(u, v, ESt, WSt));
9 24 [ 4, 18 ] [ 11 ]
15 24 [ 6, 20 ] [ 13 ]
18 48 [ 8, 36 ] [ 22 ]
21 45 [ 9, 35 ] [ 22 ]
24 45 [ 10, 36 ] [ 23 ]
30 48 [ 12, 40 ] [ 26 ]
```

Abbildung 1

Die vorletzte Eingabezeile verwendet die Einsicht, dass die Wendestelle zwischen den beiden Extremstellen liegen muss. Man braucht hier nicht von Extrem- und Wendestellen-Kandidaten reden, da jede kubische Funktion eine Wende- und höchstens zwei Extremstellen hat; hat man diese Stellen gefunden, sind die hinreichenden Kriterien automatisch erfüllt. Betrachtet man die Ergebnisse, fällt eine Symmetrie auf: Ist

$$(0, u, v)$$

ein geeignetes Tripel, so offenbar auch

$$(0, v-u, v),$$

und die Begründung ist einfach: Die Wendestelle ist durch

$$x_w = \frac{u+v}{3}$$

gegeben. Wegen

$$(u+v)+(v-u+v) = 3\cdot v$$

ist $u+v$ genau dann durch 3 teilbar, wenn auch $v-u+v$ es ist. Die Extremstellen sind durch

$$x_e = x_w \pm \sqrt{x_w^2 - \frac{u\cdot v}{3}} = x_w \pm \frac{\sqrt{u^2-u\cdot v+v^2}}{3}$$

gegeben. Wegen

$$(v-u)^2-(v-u)\cdot v+v^2 = v^2-u\cdot v+v^2$$

wird die Ganzheitlichkeit durch die Ersetzung von $(0,u,v)$ durch $(0,v-u,v)$ keineswegs beeinträchtigt.

Nun kann man sich weiter überlegen, ob der Aufwand von Abb. 1 notwendig war. Die Wendestelle x_w ist ganzzahlig, wenn $u = 3\cdot a$ und $v = 3\cdot b$ oder wenn $u = 3\cdot a+1$ und $v = 3\cdot b+2$ (oder umgekehrt) ist. Im zweiten Fall ist aber

$$\frac{u\cdot v}{3} = \frac{9\cdot a\cdot b+6\cdot a+3\cdot b+2}{3}$$

nicht ganzzahlig.

Somit sollte $u = 3\cdot a$ und $v = 3\cdot b$ sein. Dann gilt

$$x_w = a+b,$$
$$x_e = a+b\pm\sqrt{a^2-a\cdot b+b^2};$$

es muss also

$$a^2-a\cdot b+b^2$$

ein Quadrat sein. Hier hilft wieder ein systematisches Durchsuchen (Abb. 2).

```
n:15$
L:makelist();
for a:1 thru n do
for b:a+1 thru n do
block
 (u:a^2-a*b+b^2,
  if (floor(sqrt(u)+0.5))^2=u
  then L:append(L,makelist([a,b])))$
L;
[]
[ [3, 8], [5, 8], [7, 15], [8,15] ]
```

Abbildung 2

Das erste Ergebnis in Abb. 2 ($a = 3$ und $b = 8$) entspricht dem ersten Ergebnis von Abb. 1 ($u = 9$ und $v = 24$). Natürlich kann man in Abb. 2 eine deutlich höhere Grenze als $n = 15$ wählen.

Quartische Polynome

Man könnte eine zu Abb. 1 analoge Suchstruktur aufstellen (Abb. 3).

```
n:500$
for u:1 thru n do
for v:u+1 thru n do
for w:v+1 thru n do
 block
 (f(x):=x*(x-u)*(x-v)*(x-w),
  define(f1(x), diff(f(x), x)),
  define(f2(x), diff(f1(x), x)),
  ESt:makelist(),
  WSt:makelist(),
  for x:u+1 thru n do
     if f1(x)=0 then ESt:endcons(x, ESt),
  if length(ESt)=3 then
    for x:ESt[1] thru ESt[3] do
     if f2(x)=0 then WSt:endcons(x, WSt),
  if length(ESt)*length(WSt)=6
     then print(u, v, w, ESt, WSt));
```

Abbildung 3

Leider stellt man fest, dass das CAS keine Lösungen findet. Es scheint auch so zu sein, dass bisher niemand weiß, ob es ganzheitliche quartische Polynome mit unterschiedlichen Nullstellen überhaupt gibt, vgl. etwa [1], die dort angegebene Literatur.

Da bleibt nur, mehrfache Nullstellen zuzulassen; aus Gründen der Einfachheit sei 0 die mehrfache Nullstelle. Es sind mehrere Fälle zu unterscheiden:

1. Eine doppelte und zwei einfache Nullstellen
Hier muss man lange suchen; der CAS-Code in Abb. 4 dürfte selbsterklärend sein.

```
n:500$
for u:1 thru n do
for v:u+1 thru n do
 block
  (f(x):=x*x*(x-u)*(x-v),
   define(f1(x), diff(f(x), x)),
   define(f2(x), diff(f1(x), x)),
   ESt:makelist(),
   WSt:makelist(),
   for x:0 thru n do
     if f1(x)=0 then ESt:endcons(x, ESt),
   if length(ESt)=3 then
     for x:ESt[1] thru ESt[3] do
       if f2(x)=0 then WSt:endcons(x, WSt),
   if length(ESt)*length(WSt)=6
      then print(u, v, ESt, WSt));
```

Abbildung 4

Das kleinste Ergebnis ist $u = 308$, $v = 360$ mit Extremstellen bei 0, 165 und 336 sowie Wendestellen bei 70 und 264. Das Suchverfahren lässt sich beschleunigen: Ist

$$f(x) = x^2\cdot(x-u)\cdot(x-v),$$

so hat f''' die Lösung

$$x_h = \frac{u+v}{4}$$

und f'' die Lösungen

$$x_w = x_h \pm \sqrt{x_h^2 - \frac{u \cdot v}{6}}$$

sowie f' die nichttrivialen Lösungen

$$x_e = \frac{3}{2} \cdot x_h \pm \sqrt{\left(\frac{3}{2} \cdot x_h\right)^2 - \frac{u \cdot v}{2}}.$$

Man bekommt Beispiele zur Ganzheitlichkeit, wenn x_h durch 2 teilbar ist (aber, wie das Beispiel $u = 308$ und $v = 360$ zeigt, muss das nicht sein!) und $u \cdot v$ durch 6; ferner müssen die Radikanden Quadrate sein (Abb. 5).

```
teilt(a,b):=if floor(b/a+0.5)=b/a
   then true else false$

istQua(a):=if (floor(sqrt(a)+0.5))^2=a
   then true else false$

n:1000$
L:makelist();
for u:500 thru n do
for v:u+1 thru n do
block
 (h:(u+v)/4,
  d:3*h/2,
  b1:teilt(2, h),
  b2:teilt(6, u*v),
  if (b1 and b2) then
  block
   (r1:h^2-u*v/6,
    b3:istQua(r1),
    r2:d^2-u*v/2,
    b4:istQua(r2),
    bb:(b3 and b4),
  if bb then
    L:append(L,makelist([u, v])))))$
L;
[ [616, 720] ]
```

Abbildung 5

Wenn man einschränkende Bedingungen macht, wird das Suchverfahren noch schneller: Es sei $u = 8 \cdot a$ und $v = 24 \cdot b$; die Radikanden sind

$$x_h^2 - \frac{u \cdot v}{6} = 4 \cdot \left((a + 3 \cdot b)^2 - 8 \cdot a \cdot b\right)$$

und

$$\left(\frac{3}{2} \cdot x_h\right)^2 - \frac{u \cdot v}{2} = 9 \cdot (a + 3 \cdot b)^2 - 96 \cdot a \cdot b.$$

Abb. 6 zeigt zwei Ergebnisse unter $n = 100$; das erste entspricht der dem Ergebnis von Abb. 5.

```
n:200$
L:makelist();
for a:1 thru n do
for b:1 thru n do
block
 (u:8*a, v:24*b, xh:(u+v)/4, d:3*xh/2,
  r1:xh^2-u*v/6,
  r2:d^2-u*v/2,
  b1:(floor(sqrt(r1)+0.5))^2=r1,
  b2:(floor(sqrt(r2)+0.5))^2=r2,
  bb:b1 and b2,
  if bb then
    L:append(L, makelist([a, b])))$
L;
[ [77, 30], [154, 60] ]
```

Abbildung 6

2. Zwei doppelte Nullstellen
Man sieht an Abb. 7, dass es hier keine ganzheitlichen Beispiele geben kann, denn f'' hat in keinem Fall ganzzahlige Nullstellen.

```
f:x^2*(x-6*u)^2$
f1:diff(f,x)$
f2:diff(f1,x)$

solve(f2,x);
[x=(3-sqrt(3))u, x=(sqrt(3)+3)u ]

solve(f1,x);
[x=3u, x=6u, x=0 ]
```

Abbildung 7

3. Eine dreifache Nullstelle
Dies ist ganz einfach: Abb. 8 zeigt, dass die einfache Nullstelle durch 4 teilbar sein muss.

```
f:x^3*(x-4*u)$
f1:diff(f,x)$
f2:diff(f1,x)$

solve(f2,x);
[ x=2u, x=0 ]

solve(f1,x);
[ x=3u, x=0 ]
```

Abbildung 8

Schlussbemerkung

Schon die Suche nach ganzheitlichen Polynomen vom Grad 4 erwies sich als unerwartet schwierig, so dass man sich nicht trauen wird, ganzheitliche Polynome vom Grad 5 ohne jeglichen Theorieaufwand zu suchen.

Literaturverzeichnis

[1] Ralph Buchholz und James MacDougall. When Newton met Diophantus : A study of rational-derived polynomials and their extension to quadratic fields. *Journal of Number Theory* 81 (2000), no. 2, pp. 210-233.

Berichte über Arbeitsgruppen

Neu im Arbeitsbereich Geometrie der Universität Tübingen

Hannah Markwig (Tübingen)

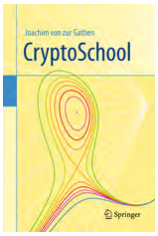
Der Arbeitsbereich Geometrie der Uni Tübingen ist mit der Berufung von Hannah Markwig um die tropische Geometrie bereichert worden. Tropische Geometrie ist Geometrie über dem Max-Plus-Halbkörper. Gleichzeitig kann sie als eine Degeneration der algebraischen Geometrie angesehen werden, bei der Polyederkomplexe produziert werden. Die sogenannte Tropikalisierung einer algebraischen Varietät erlaubt Rückschlüsse auf Eigenschaften der Varietät, so dass die tropische Geometrie als ein konvex-geometrisches Werkzeug in der algebraischen Geometrie behandelt werden kann. Tropikalisierung erlaubt in einigen Fällen explizite Berechnungen, die ohne dieses Werkzeug nicht zugänglich sind. Zudem birgt das Studium der tropischen Geometrie neue Herausforderungen algorithmischer Art. Markwig ist als Co-PI am DFG-Schwerpunktprojekts „Algorithmische tropische Schnitttheorie auf Modulräumen“ (gemeinsam mit Andreas Gathmann, TU Kaiserslautern) beteiligt. In einem weiteren DFG-Projekt geht es um Tropikalisierungen von tropischen Modulräumen und Konsequenzen in der enumerativen Geometrie. Postdoc Arthur Renaudineau unterstützt die Arbeiten an diesem Projekt. Gemeinsam mit Johannes Rau, Juniorprofessor im Arbeitsbereich Geometrie, und Postdoc Boulos El-Hilany, arbeitet Markwig im Rahmen des DFG-Projekts „Reelle Hurwitzzahlen“ an Fragen der reellen enumerativen Geometrie. Zur Arbeitsgruppe gehören außerdem noch der Doktorand Marvin Hahn, der an kombinatorischen Variationen des Hurwitzproblems sowie an Mustafin Varietäten arbeitet, und Doktorand Christoph Goldner, der sich mit tropischen Hirzebruchzykeln beschäftigt. Der Doktorand Christian Jürgens befasst sich mit tropischen Singularitäten.

Berufungen

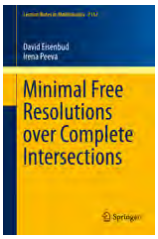
Dr. Jeroen Sijsling ist seit dem Sommersemester 2016 Juniorprofessor am Institut für Mathematik der Universität Ulm.

Publikationen über Computeralgebra

Neuerscheinungen:



Joachim von zur Gathen,
CryptoSchool,
Springer, 2016, 806 Seiten,
ISBN 978-3-662-48423-4



David Eisenbud, Irena Peeva,
Minimal Free Resolutions over Complete Intersections,
Springer, 2016, 107 Seiten,
ISBN 978-3-319-26436-3

Die Rubrik Publikationen ist nicht allein auf eine Liste von Neuerscheinungen und Neuauflagen beschränkt. Sie lebt vor allem von fundierten Rezensionen von Fachgruppenmitgliedern für Fachgruppenmitglieder, die wir an dieser Stelle gerne abdrucken. Sollte eines der oben genannten Bücher, insbesondere eine der Neuerscheinungen, Ihr Interesse geweckt haben, und Sie möchten dieses für den Computeralgebra-Rundbrief besprechen, nehmen Sie bitte Kontakt zu Anne Frühbis-Krüger (`fruehbis-krueger@math.uni-hannover.de`) auf.

Promotionen in der Computeralgebra

Dereje Kifle Boku: Gröbner Bases over Extension Fields of $\mathbb{Q}$

Betreuer: Wolfram Decker (Kaiserslautern)

Zweitgutachter: Anne Frühbis-Krüger (Hannover)

August 2016

Abstract: Gröbner bases are one of the most powerful tools in computer algebra and commutative algebra, with applications in algebraic geometry and singularity theory. From the theoretical point of view, these bases can be computed over any field using Buchberger's algorithm. In practice, however, the computational efficiency depends on the arithmetic of the coefficient field.

In this thesis, we consider Gröbner bases computations over two types of coefficient fields. First, consider a simple extension $K = \mathbb{Q}(\alpha)$ of $\mathbb{Q}$, where α is an algebraic number, and let $f \in \mathbb{Q}[t]$ be the minimal polynomial of α. Second, let K' be the algebraic function field over $\mathbb{Q}$ with transcendental parameters $t_1, \ldots, t_m$, that is, $K' = \mathbb{Q}(t_1, \ldots, t_m)$. In particular, we present efficient algorithms for computing Gröbner bases over K and K'. Moreover, we present an efficient method for computing syzygy modules over K.

To compute Gröbner bases over K, starting from the ideas of Noro, we proceed by joining f to the ideal to be considered, adding t as an extra variable. But instead of avoiding superfluous S-pair reductions by inverting algebraic numbers, we achieve the same goal by applying modular methods, that is, by inferring information in characteristic zero from information in characteristic $p > 0$. For suitable primes p, the minimal polynomial f is reducible over $\mathbb{F}_p$. This allows us to apply modular methods once again, on a second level, with respect to the modular factors of f. The algorithm thus resembles a divide and conquer strategy and is in particular easily parallelizable. Moreover, using a similar approach, we present an algorithm for computing syzygy modules over K.

On the other hand, to compute Gröbner bases over K', our new algorithm first specializes the parameters $t_1, \ldots, t_m$ to reduce the problem from $K'[x_1, \ldots, x_n]$ to $\mathbb{Q}[x_1, \ldots, x_n]$. The algorithm then computes a set of Gröbner bases of specialized ideals. From this set of Gröbner bases with coefficients in $\mathbb{Q}$, it obtains a Gröbner basis of the input ideal using sparse multivariate rational interpolation.

At current state, these algorithms are probabilistic in the sense that, as for other modular Gröbner basis computations, an effective final verification test is only known for homogeneous ideals or for local monomial orderings. The presented timings show that for most examples, our algorithms, which have been implemented in SINGULAR, are considerably faster than other known methods.

Johannes Braun: Maintaining Security and Trust in large scale Public Key Infrastructures

Betreuer: Johannes Buchmann (Darmstadt)

Zweitgutachter: Max Mühlhäuser (Darmstadt)

April 2015

Abstract: In Public Key Infrastructures (PKIs), trusted Certification Authorities (CAs) issue public key certificates which bind public keys to the identities of their owners. This enables the authentication of public keys which is a basic prerequisite for the use of digital signatures and public key encryption. These in turn are enablers for e-business, e-government and many other applications, because they allow for secure electronic communication. With the Internet being the primary communication medium in many areas of economic, social, and political life, the so-called Web PKI plays a central role. The Web PKI denotes the global PKI which enables the authentication of the public keys of web servers within the TLS protocol and thus serves as the basis for secure communications over the Internet.

However, the use of PKIs in practice bears many unsolved problems. Numerous security incidents in recent years have revealed weaknesses of the Web PKI. Because of these weaknesses, the security of Internet communication is increasingly questioned. Central issues are (1) the globally predefined trust in hundreds of CAs by browsers and operating systems. These CAs are subject to a variety of jurisdictions and differing security policies, while it is sufficient to compromise a single CA in order to break the security provided by the Web PKI. And (2) the handling of revocation of certificates. Revocation is required to invalidate certificates, e.g., if they were erroneously issued or the associated private key has been compromised. Only this can prevent their misuse by attackers. Yet, revocation is only effective if it is published in a reliable way. This turned out to be a difficult problem in the context of the Web PKI. Furthermore, the fact that often a great variety of services depends on a single CA is a serious problem. As a result, it is often almost impossible to revoke a CA's certificate. However, this is exactly what is necessary to prevent the malicious issuance of certificates with the CA's key if it turns out that a CA is in fact not trustworthy or the CA's systems have been compromised.

In this thesis, we therefore turn to the question of how to ensure that the CAs an Internet user trusts in are actually trustworthy. Based on an in depth analysis of the Web PKI, we present solutions for the different issues. In this thesis, the feasibility and practicality of the presented solutions is of central importance. From the problem analysis, which includes the evaluation of past security incidents and previous scientific work on the matter, we derive requirements for a practical solution.

For the solution of problem (1), we introduce user-centric trust management for the Web PKI. This allows to individually reduce the number of CAs a user trusts in to a fraction of the original number. This significantly reduces the risk to rely on a CA, which is actually not trustworthy. The assessment of a CA's trustworthiness is user dependent and evidence-based. In addition, the method allows to monitor the revocation status for the certificates relevant to a user. This solves the first part of problem (2). Our solution can be realized within the existing infrastructure without introducing significant overhead or usability issues. Additionally, we present an extension by online service providers. This enables to share locally collected trust information with other users and thus, to improve the necessary bootstrapping of the system. Moreover, an efficient detection mechanism for untrustworthy CAs is realized.

In regard to the second part of problem (2), we present a CA revocation tolerant PKI construction based on forward secure signature schemes (FSS). Forward security means that even in case of a key compromise, previously generated signatures can still be trusted. This makes it possible to implement revocation mechanisms such that CA certificates can be revoked, without compromising the availability of dependent

web services. We describe how the Web PKI can be transitioned to a CA revocation tolerant PKI taking into account the relevant standards.

The techniques developed in this thesis also enable us to address the related problem of "non-repudiation" of digital signatures. Non-repudiation is an important security goal for many e-business and e-government applications. Yet, non-repudiation is not guaranteed by standard PKIs. Current solutions, which are based on time-stamps generated by trusted third parties, are inefficient and costly. In this work, we show how non-repudiation can be made a standard property of PKIs. This makes time-stamps obsolete.

The techniques presented in this thesis are evaluated in terms of practicality and performance. This is based on theoretical results as well as on experimental analyses. Our results show that the proposed methods are superior to previous approaches.

In summary, this thesis presents mechanisms which make the practical use of PKIs more secure and more efficient and demonstrates the practicability of the presented techniques.

Rachid El Bansarkhani: On the Design and Improvement of Lattice-based Cryptosystems
Betreuer: Johannes Buchmann (Darmstadt)
Zweitgutachter: Tim Güneysu (Bremen)
Juni 2015

Abstract: Digital signatures and encryption schemes constitute arguably an integral part of cryptographic schemes with the goal to meet the security needs of present and future private and business applications. However, almost all public key cryptosystems applied in practice are put at risk due to its vulnerability to quantum attacks as a result of Shor's quantum algorithm. The magnitude of economic and social impact is tremendous inherently asking for alternatives replacing classical schemes in case large-scale quantum computers are built. Lattice-based cryptography emerged as a powerful candidate attracting lots of attention not only due to its conjectured resistance against quantum attacks, but also because of its unique security guarantee to provide worst-case hardness of average-case instances. Hence, the requirement of imposing further assumptions on the hardness of randomly chosen instances disappears, resulting in more efficient instantiations of cryptographic schemes. The best known lattice attack algorithms run in exponential time. In this thesis we contribute to a smooth transition into a world with practically efficient lattice-based cryptographic schemes. This is indeed accomplished by designing new algorithms and cryptographic schemes as well as improving existing ones. Our contributions are threefold. First, we construct new encryption schemes that fully exploit the error term in LWE instances. To this end, we introduce a novel computational problem that we call Augmented LWE (A-LWE), differing from the original LWE problem only in the way the error term is produced. In fact, we embed arbitrary data into the error term without changing the target distributions. Following this, we prove that A-LWE instances are indistinguishable from LWE samples. This allows to build powerful encryption schemes on top of the A-LWE problem that are simple in its representations and efficient in practice while encrypting huge amounts of data realizing message expansion factors close to 1. This improves, to our knowledge, upon all existing encryption schemes. Due to the versatility of the error term, we further add various security features such as CCA and RCCA security or even plug lattice-based signatures into parts of the error term, thus providing an additional mechanism to authenticate encrypted data. Based on the methodology to embed arbitrary data into the error term while keeping the target distributions, we realize a novel CDT-like discrete Gaussian sampler that beats the best known samplers such as Knuth-Yao or the standard CDT sampler in terms of running time. At run time the table size amounting to 44 elements is constant for every discrete Gaussian parameter and the total space requirements are exactly as large as for the standard CDT sampler. Further results include a very efficient inversion algorithm for ring elements in special classes of cyclotomic rings. In fact, by use of the NTT it is possible to efficiently check for invertibility and deduce a representation of the corresponding unit group. Moreover, we generalize the LWE inversion algorithm for the trapdoor candidate of Micciancio and Peikert from power of two moduli to arbitrary composed integers using a different approach. In the second part of this thesis, we present an efficient trapdoor construction for ideal lattices and an associated description of the GPV signature scheme. Furthermore, we improve the signing step using a different representation of the involved perturbation matrix leading to enhanced memory usage and running times. Subsequently, we introduce an advanced compression algorithm for GPV signatures, which previously suffered from huge signature sizes as a result of the construction or due to the requirement of the security proof. We circumvent this problem by introducing the notion of public and secret randomness for signatures. In particular, we generate the public portion of a signature from a short uniform random seed without violating the previous conditions. This concept is subsequently transferred to the multi-signer setting which increases the efficiency of the compression scheme in presence of multiple signers. Finally in this part, we propose the first lattice-based sequential aggregate signature scheme that enables a group of signers to sequentially generate an aggregate signature of reduced storage size such that the verifier is still able to check that each signer indeed signed a message. This approach is realized based on lattice-based trapdoor functions and has many application areas such as wireless sensor networks. In the final part of this thesis, we extend the theoretical foundations of lattices and propose new representations of lattice problems by use of Cauchy integrals. Considering lattice points as simple poles of some complex functions allows to operate on lattice points via Cauchy integrals and its generalizations. For instance, we can deduce for the one-dimensional and two-dimensional case simple expressions for the number of lattice points inside a domain using trigonometric or elliptic functions.

Tommy Hofmann: Integrality of representations of finite groups
Betreuer: Claus Fieker (Kaiserlautern)
Zweitgutachter: Werner Bley (München)
Juni 2016

Abstract: Since the early days of representation theory of finite groups in the 19th century, it was known that complex linear representations of finite groups live over number fields, that is, over finite extensions of the field of rational numbers. While the related question of integrality of representations was answered negatively by the work of Cliff, Ritter and Weiss as well as by Serre and Feit, it was not known how to decide integrality of a given representation. In this thesis we show that there exists an algorithm that given a representation of a finite group over a number field decides whether this representation can be made integral. Moreover, we provide theoretical and numerical evidence for a conjecture, which predicts the existence of splitting fields of irreducible characters with integrality properties.

In the first part, we describe two algorithms for the

pseudo-Hermite normal form, which is crucial when handling modules over ring of integers. Using a newly developed computational model for ideal and element arithmetic in number fields, we show that our pseudo-Hermite normal form algorithms have polynomial running time. Furthermore, we address a range of algorithmic questions related to orders and lattices over Dedekind domains, including computation of genera, testing local isomorphism, computation of various homomorphism rings and computation of Solomon zeta functions.

In the second part we turn to the integrality of representations of finite groups and show that an important ingredient is a thorough understanding of the reduction of lattices at almost all prime ideals. By employing class field theory and tools from representation theory we solve this problem and eventually describe an algorithm for testing integrality. After running the algorithm on a large set of examples we are led to a conjecture on the existence of integral and nonintegral splitting fields of characters. By extending techniques of Serre we prove the conjecture for characters with rational character field and Schur index two.

Tobias Moede: Coclass theory for nilpotent associative algebras

Betreuer: Bettina Eick (Braunschweig)

Zweitgutachter: Andrea Caranti (Trento)

Juli 2016

Abstract: The coclass of a finite p-group of order p^n and class c is defined as $n - c$. In 1980 Leedham-Green and Newman suggested to use coclass as the primary invariant in a possible classification of finite p-groups. Coclass theory has become very fruitful approach yielding many interesting results and it still is an area of active research.

The central aim of this thesis was to develop a coclass theory for nilpotent associative algebras over fields and hence to gain further insight into their structure. A central tool in our investigation are the so-called coclass graphs associated with the nilpotent associative F-algebras of a fixed coclass. We have developed and implemented an algorithm to construct finite parts of these graphs over finite fields. The implementation is in GAP. Based on the experimental evidence obtained in this way, we prove several interesting structural results for the associated coclass graphs. This yields results in the flavor of the coclass theorems for finite p-groups. However, the proofs in the nilpotent associative algebra case are completely different.

The most striking observation in all of our experimental data is that for finite fields all of these graphs seem to exhibit a periodic pattern. Based on this evidence, we conjecture a certain periodicity for all coclass graphs associated to nilpotent associative algebras over finite fields. We prove this conjecture for coclasses 0 and 1 and give a very detailed conjecture on the structure of coclass graphs for coclass 2.

Michael Schweinfurter: Deterministic Genericity and the Computation of Homological Invariants

Betreuer: Werner Seiler (Kassel)

Zweitgutachter: Wolfram Koepf (Kassel)

Juli 2016

Abstract: The main goal of this thesis is to discuss the determination of homological invariants of polynomial ideals. Thereby we consider different coordinate systems and analyze their meaning for the computation of certain invariants. In particular, we provide an algorithm that transforms any ideal into strongly stable position if char k = 0. With a slight modification, this algorithm can also be used to achieve a stable or quasi-stable position. If our field has positive characteristic, the Borel-fixed position is the maximum we can obtain with our method. Further, we present some applications of Pommaret bases, where we focus on how to directly read off invariants from this basis.

In the second half of this dissertation we take a closer look at another homological invariant, namely the (absolute) reduction number. It is a known fact that one immediately receives the reduction number from the basis of the generic initial ideal. However, we show that it is not possible to formulate an algorithm – based on analyzing only the leading ideal – that transforms an ideal into a position, which allows us to directly receive this invariant from the leading ideal. So in general we can not read off the reduction number of a Pommaret basis. This result motivates a deeper investigation of which properties a coordinate system must possess so that we can determine the reduction number easily, i.e. by analyzing the leading ideal. This approach leads to the introduction of some generalized versions of the mentioned stable positions, such as the weakly D-stable or weakly D-minimal stable position. The latter represents a coordinate system that allows to determine the reduction number without any further computations. Finally, we introduce the notion of -maximal position, which provides lots of interesting algebraic properties. In particular, this position is in combination with weakly D-stable sufficient for the weakly D-minimal stable position and so possesses a connection to the reduction number.

Martin Vigil: Trustworthy and Efficient Protection Schemes for Digital Archiving

Betreuer: Johannes Buchmann (Darmstadt)

Zweitgutachter: Ricardo Custódio (Santa Catarina)

Juli 2015

Abstract: The amount of information produced in the last decades has grown notably. Much of this information only exists in the form of electronic documents and it has often to be stored for long periods. Therefore, digital archives are increasingly needed. However, for the documents to remain trustworthy while they are archived, they need to be protected by the archivists. Important protection goals that must be guaranteed are integrity, authenticity, non-repudiation, and proof of existence.

To address these goals, several protection schemes for digital archives have been designed. These schemes are usually based on cryptographic primitives, namely digital signatures and hash functions. However, since documents can be archived for decades or even indefinitely, the used cryptographic primitives can become insecure during the archival time. This is a serious issue because it can be exploited by attackers to compromise the protection goals of the archived documents. Therefore, a requirement for long-term protection schemes is to address the aging of cryptography, i.e. replacing the used primitives properly before they become insecure.

In this work we analyze and improve long-term protection schemes for digital archives. More precisely, we aim at answering three questions. (1) How do long-term protection schemes compare with respect to trustworthiness? (2) How do they differ in performance? (3) Can new schemes be designed, which generate more efficient and trustworthy evidence needed to establish the protection goals?

Although several protection schemes can be found in the literature, many of them fail in addressing the aging of cryptography. Therefore, our first step is to identify which existing

schemes provide long-term protection with respect to integrity, authenticity, non-repudiation, and proof of existence.

Afterwards, to answer question (1) we analyze the trustworthiness of the long-term protection schemes using two approaches. In the first approach, we initially identify the required trust assumptions. Then, based on these assumptions, we compare the protection schemes.

In the second approach, we turn to quantifying the trustworthiness of the evidence generated by time-stamping and notarial schemes. To this end, we use a belief trust model and design a reputation system. This leads to two further, more detailed answers to question (1). First, that trustworthiness depends on the reputation of the involved parties rather than the protection schemes themselves. Second, the trustworthiness of evidence tends to degrade in the long term. Therefore, we propose to use the reputation system to create incentives for the involved parties to build good reputation. This raises the trustworthiness of generated evidence, hence addressing question (3).

Next, we address question (2) by analyzing how schemes differ in performance using an analytical evaluation and experiments. More precisely, we measure the times needed to create and verify evidence, the space required to store evidence, and the communication necessary to generate evidence. Moreover, this analysis shows that while verifying evidence most of the time is spent on checking certificate chains.

The findings in the performance analysis provide us with directions for addressing question (3). We propose three new solutions that provide more efficient evidence. The first solution is a new notarial scheme that generates smaller evidence and that communicates less data than the existing notarial scheme. Novelties in our scheme include balancing the numbers of signatures that users and notaries verify, and using notaries as time-stamp authorities to provide proof of existence.

The second solution is based on the time-stamping scheme Content Integrity Service (CIS) and allows for faster evidence verification. To the best of our knowledge, CIS is the only scheme designed for an archive where documents are submitted and time-stamped sequentially but share the same sequence of time-stamps. However, in this case the validities of several time-stamps in this sequence may overlap. Consequently, many of these time-stamps need not be checked when verifying the time-stamp sequence for one document. We address this issue in our new scheme by using a data structure called skip list. The result is a time-stamp sequence where users can skip the time-stamps that are not necessary to guarantee the protection goals of one document. Using an analytical evaluation and experiments, we show that our scheme is notably faster than CIS.

The third solution is intended to reduce time spent on checking certificate chains when verifying evidence generated by time-stamping schemes. More precisely, we improve an existing public key infrastructure-based solution where the root certification authority generates smaller verification information for time-stamps. This verification information can be used to replace the certificate chains needed to verify time-stamps. However, this solution requires extra work from time-stamp authorities and the root certification authority, especially when the number of time-stamps grows significantly. In our solution, this issue is addressed such that this extra work is independent of the number of time-stamps. Using an analytical evaluation we demonstrate the advantage of our solution.

Finally, we provide our conclusions and future work. In this thesis we design new solutions that allow for more efficient and trustworthy evidence of protection for archived documents. As future work, we suggest conducting more research in the direction of developing methods that address the decay of the trustworthiness of evidence over time.

Thomas Stadler: Eine Anwendung der Invariantentheorie auf das Korrespondenzproblem lokaler Bildmerkmale
Betreuer: Martin Kreuzer (Passau)
Zweitgutachter: Gregor Kemper (München)
Juli 2016

Bericht über die Dissertation: Die Dissertation von Thomas Stadler befasst sich mit einer neuartigen Anwendung von Verfahren der algorithmischen Invariatentheorie auf ein wichtiges Teilproblem in der Bildverarbeitung, nämlich das Korrespondenzproblem. Dabei geht es darum festzustellen, ob zwei lokale Pixelfenster in etwa dasselbe Objekt oder Szenario darstellen. Die beiden Bilder, aus denen die Pixelfenster stammen, sind dabei Aufnahmen desselben Objekts oder Szenarios, die von verschiedenen Kamerapositionen aus entstanden sind. Die beiden Pixelfenster stellen das Objekt also nicht in etwa gleich dar, sondern das Objekt muss erst einer 3-dimensionalen Transformation unterzogen werden, um zu der zweiten Abbildung zu passen.

Thomas Stadler führt die Aufgabe, solchermassen korrespondierende Pixelfenster in grosser Zahl und schnell zu finden, nun in folgender Weise auf ein Problem der algorithmischen Invariantentheorie zurück. Das in einem Pixelfenster dargestellte Grauwertebild wird durch eine Polynomfunktion niedrigen Grades approximiert. Die beiden Polynomfunktionen, die die zu vergleichenden Pixelfenster approximieren, unterscheiden sich dann durch eine Koordinatentransformation ihrer Variablen. Wenn man die erlaubten Veränderungen der Kameraposition, des Blickwinkels und der Vergrösserung nun durch eine Gruppe erlaubter Koordinatentransformationen darstellt, so bedeutet die Korrespondenzfindung, das man prüfen muss, ob die beiden Polynome approximativ im derselben Bahn der Operation dieser Gruppe auf dem Polynomring liegen. Dies ist nun die approximative Version einer Standardaufgabe der algorithmischen Invariantentheorie, da die Bahnen (in guten Situationen) gerade durch den Invariantenring repräsentiert werden.

Wie sich aus dieser stark abgekürzten Inhaltsbeschreibung bereits ergibt, ist die Arbeit von Herrn Stadler ein inhaltsreiches, sorgfältig und tief ausgearbeitetes Werk. Obwohl viele der Grundlagen aus den verwendeten Bereichen bereits „im Prinzip" bekannt waren, ist die Ausarbeitung in ihrem Detaillierungsgrad und ihrer Präzision neu. Die eigentliche Neuerung ist die im letzten Kapitel eingeführte Anwendung auf das Korrespondenzproblem für lokale Bildmerkmale. Obwohl sich Herr Stadler letztendlich mit einer Heuristik zufrieden geben muss, bietet sein neues Verfahren gegenüber dem Bekannten große Vorzüge und stellt so einen wichtigen Beitrag in diesem aktuellen und anwendungsrelevanten Gebiet dar. (Bericht von Martin Kreuzer)

Berichte von Konferenzen

1. International Congress of Mathematical Software (ICMS) 2016

Berlin, 11.-14. Juli 2016

http://icms2016.zib.de

In diesem Jahr fand der *International Congress of Mathematical Software* als Satellitenkonferenz des 7ECM in Berlin statt. Unter Leitung von Gert-Martin Greuel, Peter Paule, Andrew Sommese und Thorsten Koch bot er ein abwechslungsreiches Programm mit über 150 Vorträgen zu einer Vielzahl von Aspekten mathematischer Software, wobei über die Hälfte der 20 Sessions direkt oder indirekt mit Themen der Computeralgebra befasst war. So gab es neben klassischen Themen von Zahlentheorie und Algebraischer Geometrie bis hin zu Automatischem Beweisen und Symbolischer Integration auch Vorträge zu Anwendungen etwa bei symbolisch-numerischem Lösen oder in der Elementarteilchenphysik. Einziger Wermutstropfen des vielfältigen Programms war, dass bis zu 5 Sessions gleichzeitig tagten und so die Entscheidung zwischen verschiedenen Vorträgen dann manchmal schwer fiel. Wie inzwischen üblich wurde das Vortragsprogramm auch diesmal abgerundet von Poster Sessions und Software Demos.

Anne Frühbis-Krüger (Hannover)

2. 7ECM

Berlin, 18.-22. Juli 2016

http://www.7ecm.de

Auf dem siebten European Congress of Mathematics in Berlin war die Computeralgebra angesichts der Vielfalt an mathematischen Gebieten zwar nicht sehr auffällig, aber doch gut vertreten: Neben dem Minisymposium 'Computer Algebra and Applications', das sich direkt diesen Themen widmete und in sieben Vorträgen zu ganz unterschiedlichen Aspekten ein Panorama aktueller Entwicklungen bot, standen symbolische Algorithmen und Berechnungen auch im Mittelpunkt mehrerer Vorträge in den Minisymposien zur kombinatorischen algebraischen Geometrie, zur Tropischen Geometrie und zur Kombinatorik Polynomialer Systeme.

Anne Frühbis-Krüger (Hannover)

3. ISSAC 2016

Waterloo, Ontario, Kanada, 20.–22. Juli 2916

http://www.issac-symposium.org/2016/

Die 41. Auflage des renommierten International Symposium on Symbolic and Algebraic Computation (ISSAC) fand vom 20. bis 22. Juli 2016 an der Wilfrid Laurier University in Waterloo (Provinz Ontario in Kanada) statt. Die Konferenzreihe feierte damit ihr 50jähriges Bestehen, was beim Konferenz-Bankett ausführlich gefeiert wurde. Zunächst seien aber einige Daten genannt.

Die Konferenz wurde geleitet von Eugene Zima (General Chair und Local Arrangements Chair) und Sergei Abramov (General Chair). Als Vorsitzender des Programmkomitees fungierte Xiao-Shan Gao. Die relativ moderaten Tagungsgebühren zogen diesmal eine große Zahl von Teilnehmern an. Auch in diesem Jahr wurde einiges interessantes geboten. Als Auftakt gab es drei Tutorials, gehalten von Clemens G. Raab, Daniel Robertz und Georg Regensburger. Das Herzstück der Konferenz bildeten drei eingeladene Vorträge, gehalten von Ioannis Z. Emiris, J. Ian Munro und Carsten Schneider, und 50 vom Programmkomitee ausgewählte Vorträge, die in zwei parallelen Sitzungen gehalten wurden. Daneben gab es 25 Poster und neun Software-Demos.

Auch diesmal gab es eine ganze Reihe von Preisen, die alle bei dem Bankett vergeben wurden: Den Distinguished Paper

Der Konferenzort. Foto: ISSAC'2016

Award, den Distinguished Student Author Award, den Distinguished Poster Award, den Distinguished Software Demonstration Award und den Distinguished Female Student Award. An den Preisen für das beste Poster und für die beste Software-Demo war die Fachgruppe direkt beteiligt, indem sie das Preisgeld von je 250 Euro zur Verfügung stellte (außerdem gab es eine Maple-Lizenz). Dabei wurde der Poster-Preis zweigeteilt. Die prämierten Poster sind:

- „Fibonacci-Mandelbrot Polynomials and Matrices“ von Eunice Y. S. Chan und Robert M. Corless.
- „Linearization of a Specific Family of Bézout Matrices“ von Leili Rafiee Sevyeri und Robert M. Corless.

Auch der Software-Preis wurde geteilt:

- „Motion Polynomials and Planar Linkages“ von Christoph Koutschan.
- „Development of automatic reasoning tools in GeoGebra“ von Miguel Abánades, Francisco Botana, Zoltán Kovács, Tomás Recio und Csilla Sólyom-Gecse.

Software-Preis für Christoph Koutschan (Mitte), mit Jürgen Gerhard (Maplesoft), Gregor Kemper, Eugene Zima und Roman Pearce (Sofware Chair). Foto: ISSAC'2016

Für die Preisträger der übrigen Preise sei auf die oben angegebene Webseite verwiesen.

Bei dem ISSAC Business Meeting, an dem wie immer alle Konferenzteilnehmer (mit Stimmrecht) teilnehmen durften, wurde der Ort für die ISSAC 2018 bestimmt. Es gab zwei Bewerbungen, von denen sich die City University in New York durchsetzte. Als neues Mitglied im ISSAC Steering Committee wurde Frédéric Chyzak (INRIA) gewählt. Erneut wurde bei dem Business Meeting über die Vergabe der Preise gesprochen. Ein Antrag mit dem Ziel, den Auswahlprozess der Preisträger zu regeln und den jeweiligen Konferenz-Komitees zu übertragen, wurde kontrovers diskutiert.

Stephen Watt, Keith Geddes, Joachim von zur Gathen und Erich Kaltofen beim Maplesoft-Empfang. Foto: ISSAC'2016

Der diesjährige Konferenzstandort war gewissermaßen vor der Haustür von Maplesoft, der Firma, die das Computeralgebrasystem Maple entwickelt. So war es nicht verwunderlich, dass es einen Empfang bei Maplesoft gab.

Bei dem diesjährigen Konferenz-Bankett war neben den Preisverleihungen eine Ansprache von Stephen Watt ein Highlight, in der er mit feinem Humor das 50jährige Bestehen der Konferenzreihe feierte und die letzten 50 Jahre Revue passieren ließ.

Konferenz-Bankett. Foto: ISSAC'2016

Die nächste ISSAC-Konferenz findet vom 25. bis 28. Juli 2017 in Kaiserslautern statt.

Gregor Kemper (München)

4. Applications of Computer Algebra (ACA)

Kassel, 1.–4. August 2016

`http://www.mathematik.uni-kassel.de/ACA2016/index.php`

Die 22. Auflage der Konferenzreihe *Applications of Computer Algebra (ACA)* fand dieses Jahr erstmalig in Deutschland statt unter der Leitung von Wolfram Koepf und Werner M. Seiler. Rund 110 Teilnehmer aus aller Welt präsentierten über 100 Vorträge und diskutierten in vier “round table” Runden. Dazu kamen drei eingeladene Hauptvorträge: Thomas Sturm (Nancy) sprach über *Real Problems over the Reals: From Complete Elimination Procedures to Subtropical Decisions*, David Jeffrey (London, Ontario) über *Computer Algebra Systems and the Lambert W Function* und Pedro Real (Sevilla) über *Exploring a Homotopy Approach to the Science of Data: Huge Scenarios, Topological Scintigraphy and Flagellate Structures*. Ferner stellten Vertreter von MAPLE und MATHEMATICA in zwei Sponsorenvorträgen aktuelle Entwicklungen ihrer Produkte vor. Neben diesen beiden Sponsoren unterstützte auch die Deutsche Forschungsgemeinschaft die Tagung durch einen finanziellen Zuschuß.

Gemäß der Prinzipien der ACA-Konferenzreihe wurden keine thematischen Schwerpunkte vorgegeben, sondern die einzelnen Sektionen wurden von der ACA Working Group aus Vorschlägen von Gruppen potentieller Organisatoren ausgewählt. Die Sektionsleiter waren dann auch verantwortlich für die Kontaktierung geeigneter Redner.

Diesmal gab es 12 Sektionen zu den folgenden Themen:

- *S1 – Computer Algebra for Modeling in Science and Engineering*
- *S2 – Computer Algebra in Education*
- *S3 – Human-Computer Algebra Interaction*
- *S4 – Applied and Computational Algebraic Topology*
- *S5 – Difference Computer Algebra and its Applications*

- *S6 – Computer Algebra for Dynamical Systems and Celestial Mechanics*
- *S7 – Information Services for Mathematical Software, Models, and Research Data*
- *S8 – Algebraic and Algorithmic Aspects of Differential and Integral Operator Session*
- *S9 – Automated Theorem Proving in Dynamic Geometry: Current Achievements*
- *S10 – Computer Algebra in Coding Theory and Cryptography*
- *S11 – SC-Square: Symbolic Computation and Satisfiability Checking*
- *S12 – General Session*

Etliche der Sektionen wie z.B. S2, S4 oder S8 finden fast jedes Jahr statt und stellen einen wichtigen Treffpunkt für Wissenschaftler auf den betreffenden Gebieten dar. Die beiden Sektionen S9 und S11 markierten dabei den Beginn zweier neuer europäischer Projekte im Bereich der Computeralgebra. Am stärksten besucht waren diesmal die Sektionen S2, S6 und S8.

Im Rahmenprogramm gab es einen Ausflug zu den berühmten Wasserspielen im Bergpark Wilhelmshöhe, die mittlerweile zum UNESCO Weltkulturerbe gehören.

Leider kam dabei auch einiges Wasser von oben, was die Begeisterung der Teilnehmer aber nur minimal beeinträchtigte. Das Konferenzdinner fand im Geburtshaus von Dorothea Viehmann statt, von der die Gebrüder Grimm viele ihrer Märchen erzählt bekamen.

Werner M. Seiler (Kassel)

Tagungsfoto ACA 2016 (Foto: Jonathan Alainis)

Tagungsposter (Hendrikje Schmidtpott) mit Bergpark Wilhelmshöhe (Foto: Ralf Schaper)

5. ANTS XII

Kaiserslautern, 29.8. - 2.9.2016

`www.mathematik.uni-kl.de/~thofmann/ants/`

Vom 29.8. bis 2.9. fand an der TU-Kaiserslautern die diesjährige ANTS Tagung, organisiert von Claus Fieker, Tommy Hofmann und Bill Hart, statt. ANTS, das Algorithmic Number Theory Symposium, findet alle 2 Jahre statt, das letzte mal in 2014 in Gyeongju, Korea. Ungefähr 100 Teilnehmer hörten 26 eingereichte Vorträge, 5 eingeladene Hauptvorträge sowie neun Kurzvorträge bei der Rumpsession. Ferner gab es noch eine Postersession mit 6 eingereichten Plakaten. Die Hauptvortragenden waren Chris Peikert (Finding Short Generators of Ideals, and Implications for Cryptography), Wei Ho (Arithmetic invariant theory and distributions of invariants for number fields and elliptic curves), Biance Viray (Brauer groups and the Brauer-Manin obstruction on geometrically abelian surfaces and geometrically Kummer surfaces), Ghaith Hiary (Analytic algorithms to compute L-functions) und Henri Cohen (Using trace formulas to construct modular form spaces).

Die eingereichten Vortäge zeigten die ganze Brandbreite der algorithmischen Zahlentheorie: von Gittern und diskreten Logarithmen, über arithmetische Geometrie bis zu L-Reihen.

Der Selfridge Preis für den besten eingereichten Artikel ging an Jan Steffen Müller und Michael Stoll für „Computing canonical heights on elliptic curves in quasi-linear time“. Tagungshöhepunkt war zweifelsohne die Nachtwanderung mit Fackeln nach dem Tagungsessen.

Die Teilnehmer. Foto: ANTS'2016

Die nächste Tagung, ANTS-XIII 2018 wird wahrscheinlich im mittleren Westen der USA, in Madison (Wisconsin) stattfinden.

Claus Fieker (Kaiserslautern)

6. The 18th International Workshop on Computer Algebra and Scientific Computing (CASC 2016)

Bucharest, September 19–23, 2016

`http://www.casc.cs.uni-bonn.de/2016`

One of the main goals of the International Workshops on Computer Algebra in Scientific Computing, which started in 1998 and since then have been held annually, is to highlight cutting-edge advances in all major disciplines of Computer Algebra (CA). And the second goal of the CASC workshops is to bring together both the researchers in theoretical computer algebra and the engineers as well as other allied professionals applying CA tools for solving problems in industry and in various branches of scientific computing.

This year the 18th CASC conference was held in Bucharest (Romania) and very well organized by the University of Bucharest with the Romanian Mathematical Society, with the local organizing committee by Luminita Dumitrica, Mihaela Mirulet, and Silviu Vasile being headed by Doru Stefanescu. Computer Algebra is popular among scientists in Romania. Researchers from many institutions, such as the University of Bucharest, the Institute of Mathematics "Simion Stoilow" of the Romanian Academy, the West University of Timisoara, the University "Al. I. Cuza" of Iasi, the Institute of Computing "Tiberiu Popoviciu" from Cluj-Napoca, the "Horia Hulubei" National Institute for Research and Development in Physics and Nuclear Engineering (Bucharest-Magurele), and "Ovidius" University in Constanta, are working on subjects such as numerical simulation using computer algebra systems, symbolic-numeric methods for polynomial equations and inequalities, algorithms and complexity in computer algebra, application of computer algebra to natural sciences and engineering. In Romania there are several international conferences on Computer Algebra and related topics such as the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC, in Timisoara, or the series of conferences on commutative algebra and computer algebra held in Constanta and Bucharest. The above has affected the choice of Bucharest as a venue for the CASC 2016 workshop.

Talks on 29 full papers submitted to the workshop by the participant and accepted by the Program Committee after a thorough reviewing process with usually three independent referee reports were given. Additionally, two invited talks were given. The accompanying papers are contained in the proceedings, as well as 30 full papers—one accepted paper could not be presented due to visa problems. Polynomial algebra, which is at the core of computer algebra, is represented by contributions devoted to improved algorithms for computing Janet and Pommaret bases, the dynamic Gröbner bases computation, the algorithmic computation of polynomial amoebas, refinement of the bound of Lagrange for the positive roots of univariate polynomials, computation of characteristic polynomials of matrices whose entries are integer coefficient bivariate polynomials, finding the multiple eigenvalues of a matrix dependent on a parameter, the application of a novel concept of a resolving decomposition for the effective construction of free resolutions, enhancing the extended Hensel construction with the aid of a Gröbner basis, a hybrid symbolic-numeric method for computing a Puiseux series expansion for every space curve that is a solution of a polynomial system, numerical computation of border curves of bi-parametric real polynomial systems, and the application of sparse interpolation in Hensel lifting and pruning algorithms for pretropisms of Newton polytopes. Polynomial algebra also plays a central role in contributions concerned with elimination algorithms for sparse matrices over finite fields, new algorithms for computing sparse representations of systems of parametric rational fractions, and quadric arrangement in classifying rigid motions of a 3D digital image.

Several papers are devoted to using computer algebra for the investigation of various mathematical and applied topics related to ordinary differential equations (ODEs): application of the Julia package `Flows.jl` for the analysis of split-step time integrators of nonlinear evolution equations, the use of the CAS MAPLE 18 for the derivation of operator splitting methods for the numerical solution of evolution equations, and the complexity analysis of operator matrices transformations as applied to systems of linear ODEs. Three papers deal with applications of symbolic and symbolic-numeric computations for investigating and solving partial differential equations (PDEs) and ODEs in mathematical physics and fluid mechanics: the construction of a closed form solution to the kinematic part of the Cosserat partial differential equations describing the mechanical behavior of elastic rods, symbolic-numeric solution with Maple of a second-order system of ODEs arising in the problem of multichannel scattering, and symbolic-numeric optimization of the preconditioners in a numerical solver for incompressible Navier-Stokes equations.

Applications of CASs in mechanics and physics are represented by the following themes: qualitative analysis of the general integrable case of the problem of motion of a rigid body in a double force field, investigation of the influence of aerodynamic forces on satellite equilibria with the aid of the Gröbner basis method, and generation of irreducible re-

presentations of the point symmetry groups in the rotor + shape vibrational space of a nuclear collective model in the intrinsic frame. The first invited talk by Th. Hahn focused on the application of computer algebra in high-energy physics, in particular, the MATHEMATICA packages `FeynArts` and `FormCalc`. The second invited talk by C.S. Calude dealt with the problems of incompleteness and undecidability. These are important problems related to the foundations of mathematics and lead to lively discussions among participants.

CASC 2016 features for the first time a full blown Topical Session. In this fairly new feature of the CASC series, up to six talks around a common theme are invited.

The authors have an extended page limit, but their submissions are refereed according to the same principles and with the same rigour as normal submissions. This time the topic was *Satisfiability Checking and Symbolic Computation* (SC^2) and the session also marks the beginning of a European FET-CSA project with the same title (see `http://www.sc-square.org` for more information about this project and its objectives). There is a large thematic overlap between the fields of satisfiability checking (traditionally more a subject in computer science) and of symbolic computation (nowadays mainly studied by mathematicians). However, the corresponding communities are fairly disjoint and each has its own conference series. The central goal of the SC^2-project consists of bridging this gap and of bringing together people from both sides. Thus, the 2016 Topical Session intends to familiarize the CASC participants with the many interesting problems in this domain. It was well organized by E. Abraham, J. Davenport, P. Fontaine, and Th. Sturm and comprised five talks. One was a one-hour survey talk by D. Monniaux on satisfiability modulo theory. The other four talks concern the investigation of the complexity of cylindrical algebraic decomposition with respect to polynomial degree, efficient simplification techniques for special real quantifier elimination, the description of a new SAT + CAS verifier for combinatorial conjectures, and a generalized branch-and-bound approach in SAT modulo nonlinear integer arithmetic.

The program of the CASC 2016 may be found at the web site

`http://www.casc.cs.uni-bonn.de/2016/index.php/program-casc-2016`

The scientific program of the CASC 2016 workshop was combined by the local organizers in an excellent way with the cultural program including an exciting excursion to the Dealu Mare Region with visits of the Bellu Manor Museum and the Rotenberg Winery.

The online version of the Proceedings (LNCS 9890) is available at

`http://link.springer.com/book/10.1007%2F978-3-319-45641-6`

Vladimir Gerdt (Dubna)
Andreas Weber (Bonn)

CASC 2016 at excursion to Bellu Manor

Hinweise auf Konferenzen

1. **Polynomial Computer Algebra 2017**

   Euler International Mathematical Institute, St. Petersburg, Russland, 17.04.2017 – 22.04.2017

   http://pca.pdmi.ras.ru/2017

   The annual conference Polynomial Computer Algebra is devoted to polynomial algorithms in Computer Algebra. This field has a lot of applications both in theoretical and applied mathematics as well as in Computer Science. The conference PCA'2017 is the eighth in the series . The first one PCA'2008 commemorated Eugene Pankratiev who was a brilliant specialist in the field of Computer Algebra and Differential Algebra.

2. **Computeralgebra-Tagung der Fachgruppe**

   Kassel, 04.05.2017 – 06.05.2017

   http://www.fachgruppe-computeralgebra.de/tagung-kassel-2017

   In Fortsetzung der erfolgreichen Tagungen 2003, 2005, 2009, 2012, 2014 in Kassel und 2007 in Kaiserslautern führt die Fachgruppe im Mai 2017 wieder eine derartige Tagung in Kassel durch. Ziel ist es, ein Forum zu bieten, das es erstens Nachwuchswissenschaftlern ermöglicht, ihre Ergebnisse vorzustellen, andererseits aber auch einige Hauptvortragende zu gewinnen, die Übersichtsvorträge über wichtige Gebiete der Computeralgebra und über Computeralgebra-Software geben sollen.

3. **MEGA 2017**

   Nizza, Frankreich, 12.06.2016 -– 16.06.2017

   http://mega2017.inria.fr/

   MEGA is the acronym for Effective Methods in Algebraic Geometry (and its equivalent in Italian, French, Spanish, German, Russian, etc.). This series of biennial international conferences, with the tradition dating back to 1990, is devoted to computational and application aspects of Algebraic Geometry and related topics, over any characteristics.

4. **FoCM 2017**

   Barcelona, Spanien, 10.07.2016 -– 19.07.2017

   http://www.ub.edu/focm2017

   The computer has profoundly changed the relationship between mathematics and computation. Besides its invaluable role in numeric, symbolic, and experimental applications, computation is an important object of mathematical study in its own right and a fundamental theoretical tool. It is a source of new and exciting problems for mathematics. The FoCM conference, held every three years, covers the entire spectrum of mathematical computation.

5. **ACA 2017**

   Jerusalem, Israel, 17.07.2017 – 21.07.2017

   http://www.math.unm.edu/∼aca/

   The ACA - Applications of Computer Algebra - conference series is devoted to promoting all kinds of computer algebra applications, and encouraging the interaction of developers of computer algebra systems and packages with researchers and users (including scientists, engineers, educators, and mathematicians). Topics include, but are not limited to, computer algebra in the sciences, engineering, communication, medicine, pure and applied mathematics, education and computer science.

6. **PASCO 2017**

   Kaiserslautern, 23.07.2017 – 24.07.2017

   http://www.sigsam.org/PASCO/2017/

   The 8th International Workshop on Parallel Symbolic Computation (PASCO) is the latest instance in a series of workshops dedicated to the promotion and advancement of parallel algorithms and software in all areas of symbolic mathematical computation.

7. **ISSAC 2017**

   Kaiserslautern, 25.07.2017 – 28.07.2017

   http://www.issac-conference.org/

   The International Symposium on Symbolic and Algebraic Computation (ISSAC) is the premier annual conference to present and discuss new developments and original research results in all areas of symbolic mathematical computation.

Antrag auf Mitgliedschaft in der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und GAMM und auf Bezug des Computeralgebra Rundbriefs

Rückfragen: Telefon: + 49 (0)228-302-151/-149 / Telefax: + 49 (0)228-302-167/
E-Mail: mitgliederservice@gi.de / http://www.gi.de
Bitte zurücksenden an: Prof. Dr. Wolfram Koepf, Institut für Mathematik, Heinrich-Plett-Str. 40, 34132 Kassel

Name:	Vorname:
Akadem. Grad:	Geburtsjahr:
Privatanschrift:	
Straße / Postf.:	PLZ Ort:
Telefon:	Telefax:
Dienstanschrift:	
Firma / Inst.:	Abteilung:
Straße / Postf.:	PLZ Ort:
Telefon:	Telefax:
E-Mail:	
Gewünschte Postanschrift: ☐ Privatanschrift	☐ Dienstanschrift
Gewünschte Regionalgruppenzuordnung (www.gi.de/regionalgruppen/):	
Regionalgruppe:	

☐ Ich bin persönliches Mitglied der GI und beantrage die Mitgliedschaft in der Fachgruppe Computeralgebra sowie den Bezug des Rundbriefs

☐ Ich beantrage assoziierte Mitgliedschaft in der GI und Mitgliedschaft in der Fachgruppe Computeralgebra sowie den Bezug des Rundbriefs

☐ ab 1. Januar. ☐ rückwirkend zum 1. Januar des laufenden Jahres (bis zum 30. September möglich).

Ich ordne mich folgender Jahresbeitragsklasse zu:

- ☐ 7,50 Euro für Mitglieder von ☐ DMV ☐ GI ☐ GAMM Mitgliedsnummer:
- ☐ 7,50 Euro. Ich beantrage gleichzeitig Mitgliedschaft in ☐ DMV ☐ GI ☐ GAMM und bitte um Zusendung der dazu erforderlichen Unterlagen.
- ☐ 9,00 Euro für Nichtmitglieder. Ich bitte um Zusendung von Informationen über ☐ DMV ☐ GI ☐ GAMM

☐ Ich bitte lediglich um Aktualisierung meiner Adressdaten sowie meiner Angaben über die Zusendung von Informationen.

Datennutzung

Meine oben angegebenen personenbezogenen Daten werden im Rahmen meiner Mitgliedschaft soweit gesetzlich erlaubt oder aufgrund meiner Einwilligung durch die GI oder durch Dritte nach Weitergabe durch die GI wie folgt genutzt:

- ☐ für alle GI gesellschaftsinternen Aussendungen
- ☐ sowie weitere von der GI gesondert ausgewählte Informationen mit Bezug zur Informatik, wie u.a. Weiterbildungsangebote (z.B. der DIA), Informatik Veranstaltungen oder Kongresse mit und ohne GI-Beteiligung sowie Publikationen mit Informatik-Bezug.

Soweit Sie uns Ihre E-Mail Adresse angegeben haben, wird die oben angegebene Kommunikation soweit möglich elektronisch ausgeführt.

- ☐ Der Nutzung meiner E-Mail Adresse zu Zwecken, die über die satzungsgemäßen Ziele der GI hinausgehen (wie z.B. Werbung, Markt- und Meinungsforschung) stimme ich zu.

Natürlich können Sie Ihre Zustimmung jederzeit widerrufen oder Ihre E-Mail-Adresse in unserem System löschen lassen. Eine kurze Nachricht an mitgliederservice@gi.de, per Post oder Fax genügt.

Ich nehme zur Kenntnis, dass die Aufnahme in die Fachgruppe Computeralgebra zum 1.1. erfolgt und dass die Mitgliedschaft zum 31.12. mit Frist 30.11. schriftlich gekündigt werden kann.

_____ _____

(Datum) (Unterschrift)

Jetzt und in Zukunft.
Wir sind für Sie da.

Sie wählen das Werkzeug, wir liefern die passende Lösung.
Ob numerischer Graphikrechner oder Computeralgebrasystem,
ob Computer (Win/Mac®) oder Tablet (Windows 8/iPad®) –
mit der TI-Nspire™ Technologie sind Sie bestens ausgestattet.

Die ausgezeichnete TI-Nspire™ CAS App für iPad®
erhielt das Comenius EduMedia Siegel 2016!

Bei Fragen oder Interesse an
einer unverbindlichen Produkt-
vorführung kontaktieren Sie
bitte unsere TI Schulberater:
schulberater-team@ti.com

education.ti.com/deutschland

Texas Instruments

Fachgruppenleitung Computeralgebra 2014-2017

Sprecher:
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

Fachexperte Redaktion Rundbrief:
Prof. Dr. Michael Cuntz
Institut für Algebra, Zahlentheorie und Diskrete Math.
Leibniz Universität Hannover
Welfengarten 1, 30167 Hannover
0511-762-4252
cuntz@math.uni-hannover.de
http://www.iazd.uni-hannover.de/~cuntz

Fachreferent CA-Systeme und -Bibliotheken:
Prof. Dr. Claus Fieker
Fachbereich Mathematik
Technische Universität Kaiserslautern
Gottlieb-Daimler-Straße, 67663 Kaiserslautern
0631-205-2392, -4427 (Fax)
fieker@mathematik.uni-kl.de
http://www.mathematik.uni-kl.de/~fieker

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

Fachreferent Schwerpunktprogramm 1489:
Prof. Dr. Jürgen Klüners
Mathematisches Institut der Universität Paderborn
Warburger Str. 100, 33098 Paderborn
05251-60-2646, -3516 (Fax)
klueners@math.uni-paderborn.de
http://www2.math.uni-paderborn.de/people/juergen-klueners.html

Fachreferent Themen und Anwendungen:
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 Schule und Didaktik:
OStR Jan Hendrik Müller
Rivius-Gymnasium der Stadt Attendorn
Westwall 48, 57439 Attendorn
02722-5953 (Sekretariat)
jan.mueller@math.uni-dortmund.de
www.mathebeimueller.de

Stellvertretender Sprecher:
Prof. Dr. Florian Heß
Carl-von Ossietzky Universität Oldenburg
Institut für Mathematik, 26111 Oldenburg
0441-798-2906, -3004 (Fax)
florian.hess@uni-oldenburg.de
http://www.staff.uni-oldenburg.de/florian.hess

Fachreferentin Themen und Anwendungen:
Prof. Dr. Bettina Eick
Institut Computational Mathematics
Fachbereich Mathematik und Informatik
Technische Universität Braunschweig
Braunschweig
0531-391-7525, -7414 (Fax)
beick@tu-bs.de
http://www.icm.tu-bs.de/~beick/

Fachreferentin Publikationen und Promotionen:
Prof. Dr. Anne Frühbis-Krüger
Institut für Algebraische Geometrie
Welfengarten 1, 30167 Hannover
0511-762-3592
fruehbis-krueger@math.uni-hannover.de
http://www.iag.uni-hannover.de/~anne

Fachreferentin Computational Engineering, Vertreterin der GAMM:
Prof. Dr.-Ing. Sandra Klinge
Institute of Mechanics
TU Dortmund
Leonhard-Euler-Str. 5, D-44227 Dortmund
0231-755-5790
sandra.klinge@tu-dortmund.de
http://www.im.mb.tu-dortmund.de/typo3/en/institute

Vertreter der DMV:
Prof. Dr. Wolfram Koepf
Institut für 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

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 CA an der Hochschule:
Prof. Dr. Eva Zerz
Lehrstuhl D für Mathematik
RWTH Aachen
Pontdriesch 14/16, 52062 Aachen
0241-80-94544, -92108 (Fax)
eva.zerz@math.rwth-aachen.de
http://www.math.rwth-aachen.de/~Eva.Zerz/