

März 2018

# Computeralgebra Rundbrief

> Ausgabe 62

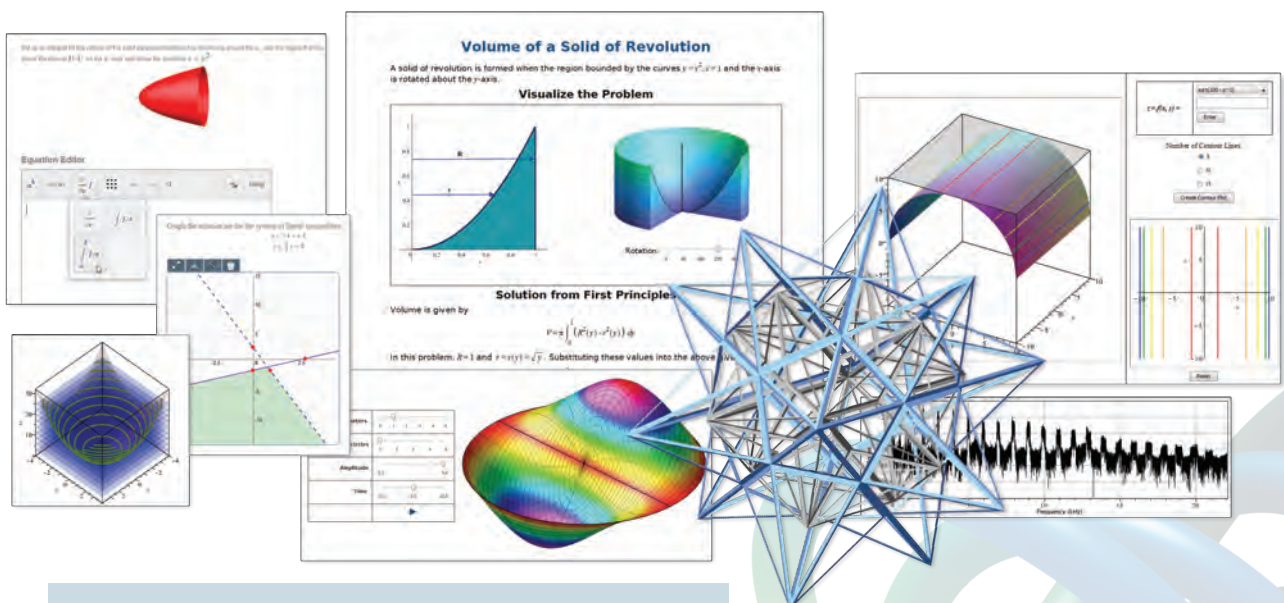
- ▶ Bindungspolynome
- ▶ Giac/Xcas new web interface
- ▶ Galois groups in Magma
- ▶ Scheinarchitektur
- ▶ Schnitte von Zylindern und Kegeln

# Herausforderungen und Lösungen für MINT-Programme

Wie kann Technologie die größten Herausforderungen in **MINT-Programmen** lösen und das Verständnis, die Merkfähigkeit und den Erfolg der Studierenden fördern?

Maple und MapleSim sind faszinierende Werkzeuge zur Visualisierung von MINT-Konzepten und zur Erkundung von Problemen. Maple T.A. ermöglicht es den Studierenden, ihre Fähigkeiten zu trainieren und gibt Ihnen ein Werkzeug an die Hand, um die Leistung der Studierenden in den MINT-Fächern zu bewerten.

Das Webinar „**Herausforderungen und Lösungen für MINT-Programme**“ bespricht die Herausforderungen der heutigen Zeit und untersucht die benutzerfreundlichen technologischen Lösungen von Maplesoft.



**Maple T.A. ist auf Deutsch verfügbar!**

Sehen Sie sich das Webinar „Herausforderungen und Lösungen für MINT-Programme“ an:

**[www.maplesoft.com/MINT](http://www.maplesoft.com/MINT)**



## Inhaltsverzeichnis

<b>Inhalt</b>	3
<b>Impressum</b>	4
<b>Mitteilungen der Sprecher</b>	5
<b>Tagungen der Fachgruppe</b>	6
<b>Themen und Anwendungen</b>	8
<i>Bindungspolynome</i> (J. W. R. Martini, Y. Ren, J. Torres)	8
<b>Neues über Systeme</b>	12
<i>Giac/Xcas new web interface</i> (B. Parisse)	12
<i>Galois groups in Magma</i> (N. Sutherland)	16
<b>Computeralgebra in der Schule</b>	22
<i>Scheinarchitektur</i> (C. Stauch)	22
<i>Schnitte von Zylindern und Kegeln</i> (J. Meyer)	27
<b>Berichte über Arbeitsgruppen</b>	29
<i>SFB/TRR 195 Symbolic Tools in Mathematics and their Application (Part 1/5)</i>	29
<b>Besprechungen zu Büchern der Computeralgebra</b>	30
<i>Pellikaan et. al.: Codes, Cryptology and Curves with Computer Algebra</i> (Martin Kreuzer)	30
<b>Promotionen in der Computeralgebra</b>	31
<b>Habilitationen in der Computeralgebra</b>	33
<b>Berichte von Konferenzen</b>	34
<b>Hinweise auf Konferenzen</b>	36
<b>Fachgruppenleitung Computeralgebra 2017–2020</b>	39

## Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und der GAMM (verantwortlicher Redakteur: Dr. Fabian Reimers [car@mathematik.de](mailto: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](mailto:gs@gi-ev.de)  
<http://www.gi-ev.de>



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



**GAMM** (Gesellschaft für Angewandte Mathematik und Mechanik e.V.)  
Technische Universität Dresden  
Institut für Statik und Dynamik der Tragwerke  
01062 Dresden  
Telefon 0351-463-33448  
Telefax 0351-463-37086  
[GAMM@mailbox.tu-dresden.de](mailto:GAMM@mailbox.tu-dresden.de)  
<http://www.gamm-ev.de>



---

## Mitteilungen der Sprecher

---

*Liebe Mitglieder der Fachgruppe Computeralgebra,*

*nachdem die Wahl der neuen Fachgruppenleitung, die Benennung neuer Vertreter der GI und der GAMM sowie das Jubiläum der Fachgruppe uns in den Mitteilungen der beiden letzten Hefte beschäftigt haben, ist es höchste Zeit, auch in den Mitteilungen der Sprecher wieder die inhaltliche Arbeit in den Vordergrund zu stellen.*

*In noch ganz frischer Erinnerung ist das Minisymposium zu Computeralgebrasystemen in der Hochschullehre, das wir in Zusammenarbeit mit dem Kompetenzzentrum Hochschuldidaktik der Mathematik (khdm) organisiert haben. Es fand Anfang März auf der gemeinsamen Jahrestagung GDMV18 einer unserer Trägerorganisationen, der DMV, mit der Gesellschaft für Didaktik der Mathematik (GDM) in Paderborn statt. Dort fügte es sich sehr gut in den Schnittstellenbereich zwischen Mathematik und Mathematikdidaktik – einer der vielen Aspekte der Interdisziplinarität der Computeralgebra. Einen ausführlichen Bericht über das Minisymposium finden Sie auf Seite 6.*

*Eine weitere Facette der Interdisziplinarität der Computeralgebra, Berührungspunkte zur Chemie, beleuchtet in diesem Heft der Artikel zu den Bindungspolynomen. Die Rubrik 'Neues über Systeme' bietet eine Vorstellung des Web Interfaces für das CAS Giac sowie einen Artikel zu Galois Gruppen in Maple. Geometrisch wird es dann in den beiden Artikeln zu 'Computeralgebra in der Schule', ehe sich der Bereich Zahlentheorie des Transregio 'Symbolische Werkzeuge in der Mathematik und ihre Anwendung' vorstellt als Auftakt zu einer kleinen Reihe von Beiträgen zu den dort vertretenen Forschungsthemen.*

*Auf der Frühjahrssitzung der Fachgruppenleitung in München haben wir unter anderem einen kritischen Blick auf unseren Internet-Auftritt geworfen und diesem dann eine gewisse Straffung seiner Struktur verordnet. Der aktuelle Zwischenstand ist schon unter*

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

*verfügbar. Wer regelmäßig auf der Seite unterwegs war, wird bemerken, dass wir nun für unsere Preisträger eine eigene Seite haben. Wie vermutlich die meisten Internet-Auftritte weltweit ist auch unserer niemals ganz fertig und optimal, so dass uns Kommentare und Anregungen von Ihnen sehr willkommen sind. Ebenfalls im Webauftritt wie auch hier im Heft auf Seite 32 finden Sie übrigens wieder die aktuellen Details zur Möglichkeit der Workshop-Förderung mit der aktuellen Antragsfrist 1.9.2018.*

*Nun möchten wir Sie aber nicht länger aufhalten und wünschen Ihnen eine angenehme und anregende Lektüre dieses Hefts.*

*Gregor Kemper*

*Anne Frühbis-Krüger*

### Minisymposium CAS in der Hochschullehre - ein Blick in die Praxis

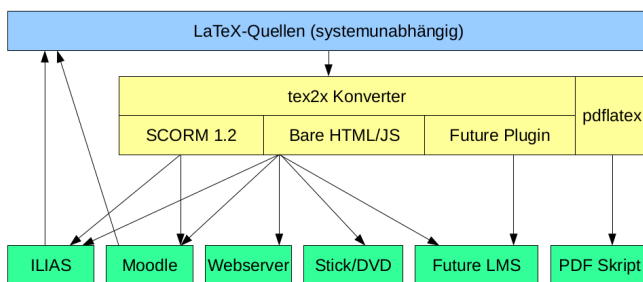
GDMV2018 Paderborn, 6. bis 9. März 2018

Computeralgebrasysteme (CAS) haben sich in den letzten Jahren und Jahrzehnten auf vielfältige Art in der Hochschullehre etabliert, auch wenn sie in der Regel nicht selbst im Fokus stehen. Mit technischem Verstand und didaktischem Geschick wurden Systeme und Einsatzformen entwickelt, die dieses Minisymposium der GMDV-Tagung näher beleuchtete. Die Einsatzmöglichkeiten beginnen dabei schon vor dem Studium.

In den Vorträgen von Volker Bach und Daniel Haase wurde deutlich, dass digitale Eingangstests und Vorkurse wie der OMB+ und VE&MINT, die zusammen den TU9-Brückenkurs bilden, für Studieninteressierte Orientierung zu den Anforderungen geben können. Vor allem aber geben sie konkretes Feedback zu fachlichen Lücken und bieten auch gleich Lerneinheiten zu diesen Inhalten an. Die Lerneinheiten können mit Animationen oder Videos angereichert werden, vor allem aber mit Aufgaben. Ein CAS im Hintergrund erlaubt unter anderem die randomisierte Auswahl von Aufgabentypen für einen Test. Konkrete Zahlen können als Koeffizienten ebenfalls zufällig bestimmt werden - evtl. unter Nebenbedingungen - und ermöglichen beliebig viele Trainings und Tests.



#### Technische Realisierung: Konvertersystem

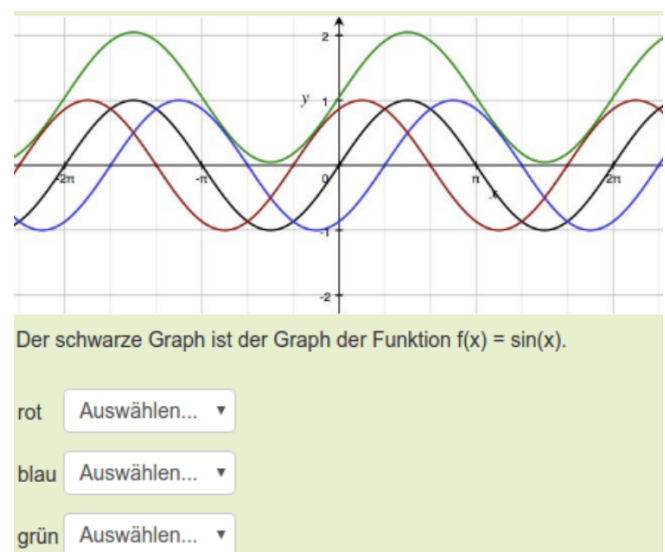


*Ein Blick hinter die Kulissen von VE&MINT (D. Haase)*

Die wichtigste Funktion des CAS ist das sofortige Feedback, weil es die Korrektheit der Eingaben prüfen kann. Die Nutzer sehen sofort, was sie richtig oder falsch gemacht haben. Als Eingaben sind dabei neben der Auswahl korrekter Lösungsvorschläge (multiple choice) auch Zahlen oder Terme möglich. Ein zweites, wichtiges Feedback geht an die Lehrenden: Sie sehen ebenfalls sofort und statistisch leicht auszuwerten, welche Themenbereiche besonders schwierig sind. Beide Kurse verzeichnen mehrere zehntausend registrierte Nutzer bzw. Abrufe – hier zeigt sich eindrucksvoll die mühelose Skalierbarkeit der Systeme. Allerdings be-

steht eine Herausforderung darin, die Nutzer bei der Stange zu halten. Nur ein einstelliger Prozentsatz der Nutzer arbeitet sich vollständig durch das Material. Mit Zertifikaten, Callcentern und moderierten Chatrooms einerseits und direkter Einbindung in Lehre vor Ort andererseits werden Anreize und Hilfestellungen gegeben. Neben der technischen Weiterentwicklung der Systeme wird auch an der didaktischen Weiterentwicklung gearbeitet.

Werden CAS in die Regellehre im Studium, in den Übungs- (oder sogar Prüfungs-) Betrieb integriert, so ergeben sich sofort weitere Möglichkeiten. Der Vortrag von Michael Kallweit konnte zeigen, dass ein CAS in diesem Kontext mehr als nur das Auslagern von Routineaufgaben erlaubt. Mit STACK im Hintergrund kann man Eingaben nicht nur auf die Form, sondern z. B. auch auf Eigenschaften testen: „Finde eine Funktion, die stetig, aber bei  $x = 4$  nicht differenzierbar ist“. Solche offenen Formate befeuern kreatives Arbeiten. Zudem zeigt sich, dass Studierende, die unterschiedlich randomisierte Aufgaben desselben Typs bearbeiten müssen, nicht einfach Lösungen austauschen können. Stattdessen werden Prinzipien besprochen, die hinter einer Aufgabe stehen, sodass stärker konzeptionelles Lernen stattfindet. Hier wie bei den Brückenkursen zeigt sich, wie wichtig gute Aufgaben und deren nahtloses Einfügen in ein Gesamtkonzept sind. Bei mehreren Lernplattformen stellen inzwischen Lehrende im Sinne guter Lehre ihre Aufgaben auch unter CreativeCommons-Lizenzen zur Verfügung. An geeigneten Plattformen zum Austausch solcher Aufgaben wird intensiv gearbeitet.



*STACK-Aufgaben im Praxiseinsatz (M. Kallweit)*

Doch der Einsatz von CAS muss nicht zwingend rein im Hintergrund erfolgen; Studierende können auch sehr viel lernen, wenn sie ein CAS wie Maple oder SAGE selbst in die Hand nehmen. Das wurde in den

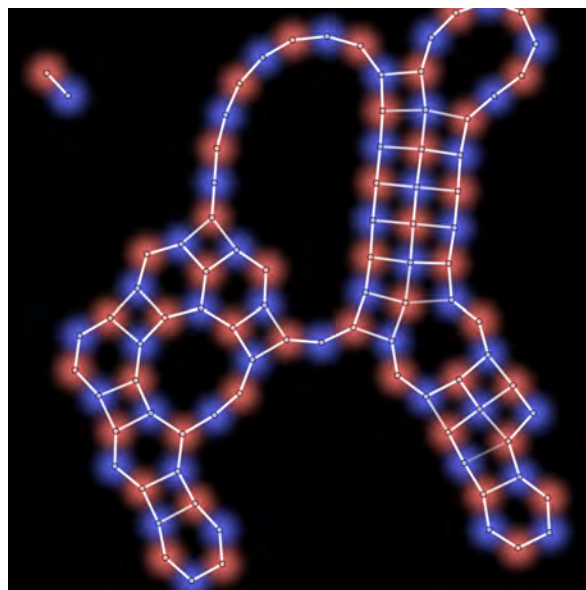


Vorträgen von Alice Niemeyer und Thorsten Jörgens deutlich, bei denen es um Lehrveranstaltungen in der Studieneingangsphase ging, die sich in ihrer Konzeption noch mehr unterscheiden als in der Auswahl der verwendeten CAS. Frau Niemeyer legte Ziele, Konzepte und Lernerfolge des Begleitpraktikums dar, das alle Mathematikstudierende der RWTH Aachen durchlaufen. Durch den Einsatz eines CAS (hier: Maple) werden dabei tiefere Verständnismöglichkeiten und Diagnosemöglichkeiten des Lernfortschritts geschaffen, als dies mit konventionellen Lehrmethoden möglich wäre. Frau Niemeyer schilderte, wie die Gewöhnung an ein CAS als Türöffner für den Einstieg in ganz verschiedene CAS wirkt. Herr Jörgens stellte das von Thorsten Theobald an der Universität Frankfurt entwickelte Modul „Einführung in die computerorientierte Mathematik“ vor, bei dem das CAS SAGE zum Einsatz kommt. Bei dem Vortrag kamen Themen des Moduls und Beispiele für den SAGE-Einsatz zur Sprache. Es zeigte sich, dass Vorkenntnisse in Python für die Syntax von SAGE von Vorteil sind.

Bei aller technischen Entwicklung darf man sich auch didaktisch und kritisch fragen, wie ihr Einsatz das Lernen am besten unterstützt. Diesen Blickwinkel nahm Rainer Kaenders in seinem Vortrag ein und demonstrierte daneben, wie CAS und Visualisierungen zur Begriffsbildung beitragen können, indem sie z. B. die Intuition befördern.

Jürgen Richter-Gebert lieferte mit seinem bunten Vortrag das Abschlussfeuerwerk des Minisymposiums. Seine Visualisierungen sind eindrucklich und tiefgänglich. Die Umsetzung von Modellen z. B. zu bewegten Fischeschwärmen nach wenigen, einfachen und transparenten Regeln lässt die Nutzer staunen, entdecken und macht Lust auf mehr. Dieses „mehr“ können sie ausprobieren und bestenfalls in sehr simpel gehaltenen Sprachen selbst programmieren. Jenseits der (Benutzer-)Oberfläche sind dabei Probleme zu umschiffen - etwa die Auswahl einer von mehreren Lösungen

durch das CAS - und Potentiale zu erschließen, die in Touch-Oberflächen und der Kommunikation mit modernen Grafikkarten zu sehen sind. Hier schließt sich der Kreis. CAS können helfen, Mathematik zu lernen. Und Mathematik ist notwendig, um CAS zu entwickeln.



*Strukturbildung als emergence Prozess in Partikelsystemen (J. Richter-Gebert)*

Auch wenn der Vortrag von Janko Böhm zum CAS-Einsatz bei mittleren Semestern leider kurzfristig der Grippe zum Opfer gefallen ist, war es insgesamt ein interessantes und ausgewogenes Minisymposium, dessen interessierte Teilnehmer nach jedem Vortrag sehr konstruktiv und engagiert im Plenum diskutierten, so dass das im knappen Zeitplan vorgesehene Raster eher als zu eng empfunden wurde.

Anne Frühbis-Krüger  
Michael Liebendörfer

## Über die Anzahl entkoppelter Moleküle mit demselben Bindepolynom

J. W. R. Martini (KWS SAAT SE<sup>a</sup>)

Y. Ren (Max-Planck-Institut MIS, Leipzig)

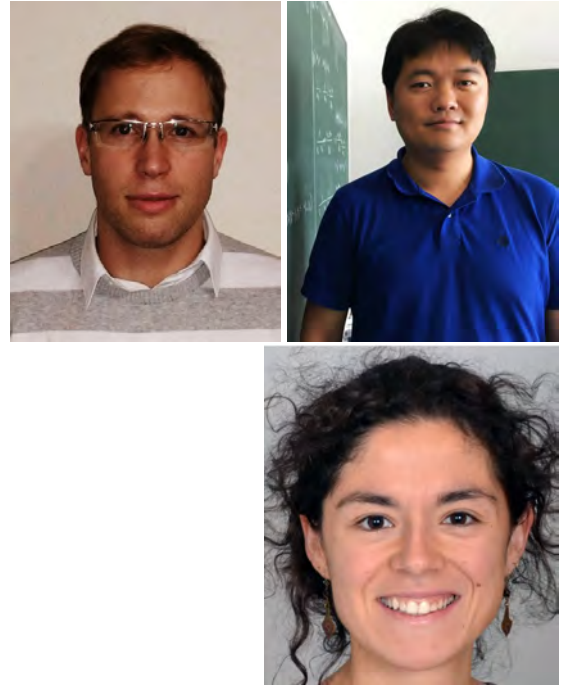
J. Torres (Max-Planck-Institut MIS, Leipzig)

johannes.martini@kws.com

yueren@mis.mpg.de

jtorres@mis.mpg.de

<sup>a</sup> dieses Projekt ist nicht mit KWS SAAT SE affiliert



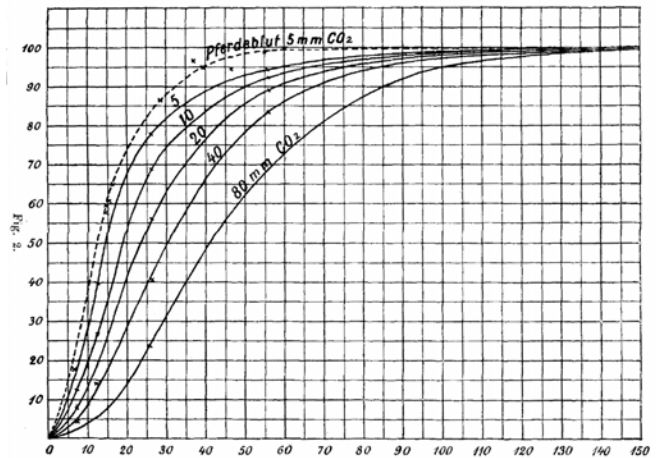
## Vorwort

Dies ist ein kurzer Bericht über den Inhalt der Artikel [8, 9] und vor allem [13]. Für genauere Details verweisen wir auf diese Artikel. Begleitmaterial zu den Computereperimenten ist auf <https://software.mis.mpg.de> zu finden.

## Einführung

Ein Ligand ist ein Stoff der reversibel an spezifische Bindestellen eines Biomoleküls binden kann. Dieses Biomolekül kann mehrere Bindestellen für unterschiedliche Liganden besitzen. Ein prominentes Beispiel hierfür ist Hämoglobin (siehe Abb. 1) welches vier Bindestellen für Sauerstoff, sowie eine weitere für 2,3-Bisphosphoglycerat besitzt. Letzteres moduliert hierbei die Affinität der vier Sauerstoffbindestellen.

Ein gängiges Modell, um Gleichgewichte und stationäre Zustände solcher Systeme zu beschreiben kommt vom sog. großkanonischen Ensemble der statistischen Mechanik [14]. Deren Partitionsfunktion, in unserem Kontext auch als *Bindungspolynom* bekannt, ist der Nenner der rationalen Funktion, welches die durchschnittliche Anzahl an belegten Bindestellen in Relation zur Ligandenaktivität beschreibt. Die Nullstellen dieses Polynoms spielen eine wichtige Rolle in der Charakterisierung des Bindungsverhaltens des Systems [4].



**Abbildung 1:** Sauerstoffkonzentration im Blut in Relation zur Sauerstoff- und Kohlenstoffdioxidkonzentration in der Luft, aus einer historischen Arbeit von Bohr, Hasselbalch und Krogh [3]

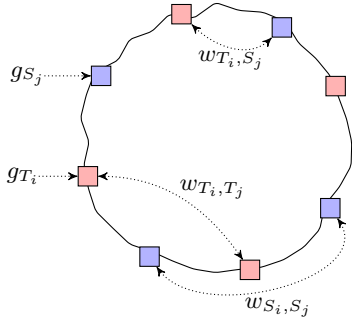
Der Einfachheit halber beschränken wir uns auf Systeme mit zwei Liganden, die wir  $T$  und  $S$  nennen. Wir sagen, ein Molekül  $M$  hat  $(n_1, n_2)$  Bindestellen, falls es  $n_1$  Bindestellen für  $T$  und  $n_2$  Bindestellen für  $S$  besitzt. Markieren wir diese mit  $T_1, \dots, T_{n_1}$  und  $S_1, \dots, S_{n_2}$ , so wird  $M$  beschrieben durch einen Punkt

$$M = \left( g_{T_1}, \dots, g_{T_{n_1}}, g_{S_1}, \dots, g_{S_{n_2}}, (w_P)_{P \subseteq \{T_i, S_j\}, |P|=2} \right) \\ \in (\mathbb{C}^*)^{n_1+n_2} \times (\mathbb{C}^*)^{\binom{n_1+n_2}{2}}$$

wobei (siehe Abb. 2):



- $g_{T_i}, g_{S_j}$  in Zusammenhang mit den Bindungsenergien an den Stellen  $T_i, S_j$  stehen,
- $w_P$  in Zusammenhang mit der Wechselwirkung zwischen den beiden Stellen in  $P$  steht.



**Abbildung 2:** Ein Molekül mit (4,4) Bindestellen.

Für ein solches Molekül  $M$  ist das Bindungspolynom  $p_M$  ein bivariates Polynom vom Grad  $(n_1, n_2)$ ,

$$p_M(X, Y) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} a_{i,j} X^i Y^j,$$

dessen Koeffizienten  $a_{i,j}$  wiederum selbst polynomiell von  $M$  abhängig sind (siehe Gleichungen (\*) und (\*\*)).

Normalerweise sind nur reellwertige, positive  $M$  von Interesse. Allerdings lässt sich im Fall von nur einem Liganden, jedes Molekül einzigartig durch ein komplexes entkoppeltes Molekül ohne Interaktion ( $w_P = 1$  für alle  $P$ ) repräsentieren [11]. “Repräsentieren” bedeutet hier, dass beide Moleküle dasselbe Bindungspolynom besitzen. Bei zwei verschiedenen Typen von Liganden bedeutet “entkoppelt”, dass keine Wechselwirkung zwischen den Bindestellen *vom gleichem Typ* existiert, genauer gesagt  $w_P = 1$  für  $P \subseteq \{T_1, \dots, T_{n_1}\}$  und  $P \subseteq \{S_1, \dots, S_{n_2}\}$ .

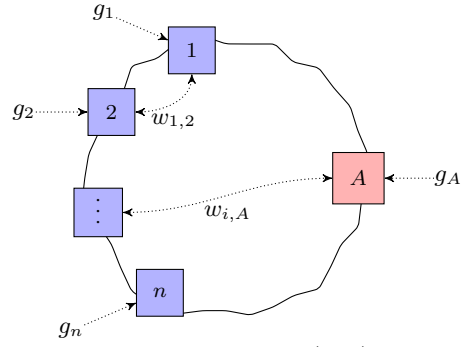
Wir beschäftigen uns mit der Frage, wie viele entkoppelte Moleküle sich in Systemen mit zwei Liganden ein Bindungspolynom teilen. Wir leiten explizite Formeln für Moleküle mit  $(n, 1)$  und  $(n, 2)$  Bindestellen her und untersuchen grössere Moleküle mittels Methoden der Computeralgebra.

## Moleküle mit $(n, 1)$ Bindestellen

Sei  $M$  ein Molekül mit  $(n, 1)$  Bindestellen:  $n$  Stellen für den ersten Liganden (als  $1, \dots, n$  markiert) und eine Stelle für den zweiten Liganden (als  $A$  markiert) (siehe Abb. 3).

Dann ist das Bindungspolynom von  $M$  ein bivariates Polynom vom Grad  $(n, 1)$ :

$$p_M(X, Y) = \sum_{i=0}^n \sum_{j=0}^1 a_{i,j} X^i Y^j.$$



**Abbildung 3:** Ein Molekül  $M$  mit  $(n, 1)$  Bindestellen. Ist  $M$  entkoppelt, so ist  $w_{i,j} = 1 \forall i, j \in \{1, \dots, n\}$  (aber nicht notwendigerweise  $w_{i,A} = 1$ ).

Ist nun  $M$  entkoppelt, oder suchen wir ein entkoppeltes Molekül  $M$  mit dem obigen Bindungspolynom, so müssen dessen Bindungsenergien  $g_1, \dots, g_n, g_A$  und Wechselwirkungen  $w_{i,A}$  zwischen den Bindestellen unterschiedlichen Typs folgendes algebraisches System lösen:

$$\begin{aligned} a_{0,0} &= 1 \\ a_{1,0} &= g_1 + \dots + g_n, \\ &\vdots \\ a_{n,0} &= g_1 \cdot \dots \cdot g_n, \\ a_{0,1} &= g_A, \\ a_{1,1} &= g_A(g_1 w_{1,A} + \dots + g_n w_{n,A}) \\ a_{2,1} &= g_A(g_1 g_2 w_{1,A} w_{2,A} + \dots \\ &\quad + g_{n-1} g_n w_{n-1,A} w_{n,A}) \\ &\vdots \\ a_{n,1} &= g_A g_1 \cdot \dots \cdot g_n w_{1,A} \cdot \dots \cdot w_{n,A}. \end{aligned} \quad (*)$$

Wir betrachten (\*) als ein parametrisiertes System polynomieller Gleichungen, mit Parametern  $a_{i,j}$ ,  $(i, j) \neq (0, 0)$ , und Variablen  $g_i, g_A, w_{i,A}$ .

Das Gleichungssystem (\*) ist symmetrisch unter der folgenden  $S_n \times S_1$ -Wirkung, was einer Ummarkierung der Bindestellen entspricht:

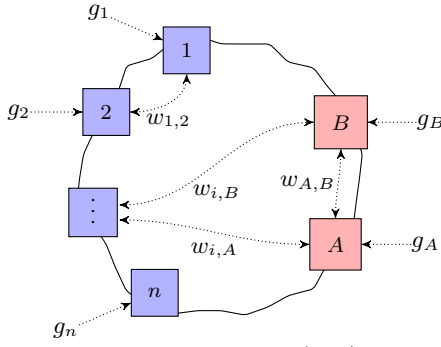
$$\begin{aligned} (\sigma, 1) \cdot (g_1, \dots, g_n, g_A, w_{1,A}, \dots, w_{n,A}) \\ = (g_{\sigma(1)}, \dots, g_{\sigma(n)}, g_A, w_{\sigma(1),A}, \dots, w_{\sigma(n),A}). \end{aligned}$$

Durch Umstellen und Anwendung von Vieta’s Formeln sieht man, dass (\*) für generische Parameter  $a_{i,j}$  immer  $(n!)^2$  Lösungen besitzt. Diese kommen in  $n!$  Orbits unter der  $S_n \times S_1$  Wirkung, weswegen es generisch  $n!$  verschiedene (unmarkierte) Moleküle gibt, die ein gegebenes Polynom als Bindungspolynom haben [8].

Ein simpler Trick, um die Gruppenwirkung für das Lösen konkreter Instanzen aus (\*) herauszuteilen, ist eine Wahl der  $g_1, \dots, g_n, g_A$  fest vorzugeben. So kann man, wenn man zum Beispiel das System für zufällig gewählte Parameter lösen will, einfach  $a_{i,1}$  für  $i > 1$  und  $g_1, \dots, g_n, g_A$  zufällig wählen, anstatt alle  $a_{i,j}$  zufällig zu wählen.

## Moleküle mit $(n, 2)$ Bindestellen

Sei  $M$  ein Molekül mit  $(n, 2)$  Bindestellen:  $1, \dots, n$  für den ersten Ligand und  $A, B$  für den zweiten (siehe Abb. 4).



**Abbildung 4:** Ein Molekül  $M$  mit  $(n, 2)$  Bindestellen. Ist  $M$  entkoppelt, so ist  $w_{i,j} = 1 = w_{A,B}$  für alle  $i, j$ .

Dann ist das Bindungspolynom von  $M$  ein bivariates Polynom vom Grad  $(n, 2)$ :

$$p_M(X, Y) = \sum_{i=0}^n \sum_{j=0}^2 a_{i,j} X^i Y^j,$$

Ist  $M$  entkoppelt, so sind die Koeffizienten gegeben durch:

$$\begin{aligned} a_{0,0} &= 1 \\ a_{1,0} &= g_1 + \dots + g_n, \\ &\vdots \\ a_{n,0} &= g_1 \cdot \dots \cdot g_n, \\ a_{0,1} &= g_A + g_B, \\ a_{1,1} &= g_A(g_1 w_{1,A} + \dots + g_n w_{n,A}) \\ &\quad + g_B(g_1 w_{1,B} + \dots + g_n w_{n,B}), \\ a_{2,1} &= g_A(g_1 g_2 w_{1,A} w_{2,A} + \dots \\ &\quad + g_{n-1} g_n w_{n-1,A} w_{n,A}) \\ &\quad + g_B(g_1 g_2 w_{1,B} w_{2,B} + \dots \\ &\quad + g_{n-1} g_n w_{n-1,B} w_{n,B}), \\ &\vdots \\ a_{n,1} &= g_A g_1 \dots g_n w_{1,A} \dots w_{n,A} \\ &\quad + g_B g_1 \dots g_n w_{1,B} \dots w_{n,B}, \\ a_{0,2} &= g_A g_B, \\ a_{1,2} &= g_A g_B (g_1 w_{1,A} w_{1,B} + \dots + g_n w_{n,A} w_{n,B}), \\ a_{2,2} &= g_A g_B (g_1 w_{1,A} w_{1,B} g_2 w_{2,A} w_{2,B} + \dots \\ &\quad + g_{n-1} w_{n-1,A} w_{n-1,B} g_n w_{n,A} w_{n,B}), \\ &\vdots \\ a_{n,2} &= g_A g_B g_1 w_{1,A} w_{1,B} \dots g_n w_{n,A} w_{n,B}. \end{aligned} \quad (**)$$

Betrachten wir in dem konkreten Fall  $n = 3$  ein zufällig gewähltes Polynom wie das folgende (beachte, dass  $g_i, g_A, g_B$  fest vorgegeben sind um die natürliche  $S_3 \times S_2$  Wirkung auszuteilen):

$$\begin{aligned} g_1 &= 2, & g_2 &= 3, & g_3 &= 5, & g_A &= 11, & g_B &= 13, \\ a_{1,1} &= 71, & a_{2,1} &= 73, & a_{3,1} &= 79, \\ a_{1,2} &= 101, & a_{2,2} &= 103, & a_{3,2} &= 107, \end{aligned}$$

so liefert uns BERTINI [1] 72 Lösungen, wobei keine davon reell ist (siehe Abb. 5 für die BERTINI Eingabe).

---

```

CONFIG
% refine endpoints to 20 digits
SharpenDigits: 20;

% used to classify sing vs nonsing
CondNumThreshold: 1e15;

% track paths more accurately
ODEPredictor: 2;
TrackTolBeforeEG: 1e-8;
TrackTolDuringEG: 1e-8;
FinalTol: 1e-10;

% values at infinity
SecurityMaxNorm: 1e8;
MaxNorm: 1e8;
END;
INPUT
variable_group w1,w2,w3,w4,w5,w6;
function f1,f2,f3,f4,f5,f6;

f1=(11*(2*w1+3*w3+5*w5)+13*(2*w2+3*w4+5*w6)-71)/70;
f2=(11*(6*w1*w3+10*w1*w5+15*w3*w5)+13*(6*w2*w4+10*w2*w6+15*w4*w6)-73)/70;
f3=(330*w1*w3*w5+390*w2*w4*w6-79)/300;
f4=(143*(2*w1*w2+3*w3*w4+5*w5*w6)-101)/150;
f5=(143*(6*w1*w2*w3+w4+10*w1*w2*w5+w6+15*w3*w4*w5*w6)-103)/120;
f6=(4290*w1*w2*w3*w4*w5*w6-107)/1000;
END;

```

---

**Abbildung 5:** BERTINI Eingabe für ein beliebig gewähltes Polynom mit  $(3, 2)$  Bindestellen. Beachte die spezielle Konfiguration, um numerischen Fehlern vorzubeugen.

Konzentrieren wir uns in der Nullstellenmenge  $\mathcal{M}$  des Systems  $(**)$  auf sog. normierte Moleküle, d.h. Moleküle mit

$$g_i = g_A = g_B = 1 \text{ und } w_{i,A} \cdot w_{i,B} = 1,$$

so kann man beweisen, dass diese eine Multiplizität von  $2n!$  besitzen und Verzweigungspunkte mit Verzweigungsgrad  $(2n!)^2$  von folgender Projektion sind:

$$\begin{aligned} \mathcal{M} &\subseteq \mathbb{C}^{\{a_{i,j}\}} \times (\mathbb{C}^*)^{\{g_i, g_A, g_B\}} \times (\mathbb{C}^*)^{\{w_{i,A}, w_{i,B}\}} \\ &\quad \downarrow \\ &(\mathbb{C}^*)^{\{g_i, g_A, g_B\}} \times (\mathbb{C}^*)^{\{w_{i,A}, w_{i,B}\}}. \end{aligned}$$

Demnach gibt es zu einem generisches Bindungspolynom vom Bigrad  $(n, 2)$  exakt  $4(n!)^3$  entkoppelte Moleküle, bzw.  $2(n!)^2$  Orbits unter der  $S_n \times S_2$  Wirkung [13]. Für den Fall  $n = 3$  macht das 864 Lösungen oder 72 Orbits unter der  $S_3 \times S_2$  Wirkung.

## Experimentelle Ergebnisse

Um eine obere Schranke für die Anzahl der Moleküle mit dem gleichen generischen Bindungspolynom von Grad  $(n, m)$  modulo der natürlichen  $S_n \times S_m$  Wirkung zu bestimmen, betrachten wir die gemischten Volumina der jeweiligen Gleichungssysteme. Diese stimmen nach Bernstein's Theorem mit den Anzahl der Lösungen überein, falls die Koeffizienten generisch sind [2].

Abb. 6 listet diese tabellarisch für kleine  $(n, m)$ , o.B.d.A.  $n \geq m$ . Diese wurden mit GFAN [6] berechnet, welches einen neuen Algorithmus basierend

auf tropischen Homotopiemethoden benutzt [7]. Zuerst erkennen wir, dass für  $(n, 1)$  und  $(n, 2)$  das gemischte Volumen mit den bewiesenen Werten in [8, 13] übereinstimmt.

	1	2	3	4	5	6
1	1	2	6	24	120	720
2		8	72	1152	28800	1036800
3			1944	162432	24624000	1349713408
4				52862976	-	-
5					-	-
6						-

Abbildung 6: gemischte Volumina für kleine  $(n, m)$

Als nächstes benutzen wir für die kleineren Fälle Gröbnerbasen um die Anzahl der Lösungen symbolisch zu bestimmen. Die roten Zahlen heben die mit SINGULAR [5] erfolgreich berechneten Fälle hervor. Hierdurch wird erstmals für den Fall  $(3, 3)$  die Anzahl der Lösungen bestimmt und bestätigt, dass diese ebenfalls mit dem gemischten Volumen übereinstimmt.

Schließlich wurde für die beiden nächstgrößeren, blau gekennzeichneten Fälle versucht die Anzahl der Lösungen numerisch mittels BERTINI [1] zu berechnen. Dies stellte sich als extrem rechenintensiv und numerisch schwierig heraus. Der Fall  $(3, 4)$  benötigte umgerechnet 6 CPU Jahre (Debian server with Intel Xeon E7-8837, 2.67GHz), und bereits der Fall  $(5, 2)$  bedurfte einer speziellen Konfiguration, um numerischen Instabilitäten vorzubeugen (siehe Abb. 5). Letzteres war uns in beiden Rechnungen leider nicht komplett möglich wegen eventuell nicht alle Lösungen gefunden wurden:

Für  $(5, 2)$  erhielten wir 28737 Lösungen, 63 weniger als oder 99.8% der bewiesenen 28800 Lösungen. Für  $(4, 3)$  erhielten wir 156966 Lösungen, 5466 weniger als oder 97% der durch die gemischten Volumina vermuteten 162432 Lösungen.

## Offene Fragen

(1) Für Bindungspolynome vom Grad  $(n, 1)$  und  $(n, 2)$  ist die Zahl dazugehöriger entkoppelter Moleküle durch einfache Formeln beschreibbar. Nehmen wir an, dass die gemischten Volumina des Gleichungssystems und die Anzahl der Lösungen übereinstimmen, zeichnet Abbildung 6 ein komplizierteres Muster für die Zahl entkoppelter Moleküle vom Grad  $(n, 3)$ . So enthält beispielsweise die vermutete Anzahl entkoppelter Moleküle für den Fall  $(3, 4)$  (162432) den Primfaktor 47. Die Anzahl entkoppelter Moleküle wäre hier also nicht einfach ein Produkt von Fakultäten der Grade.

(2) Für univariate Bindungspolynome wird die Existenz von nicht-reellen Nullstellen als Indikator für Kooperativität angesehen [10, 12]. Es ist weder klar, wie dieses Konzept auf Moleküle mit zwei Typen von Liganden übertragen werden könnte, noch welche gemeinsamen Eigenschaften verschiedene entkoppelte Moleküle teilen. Um ein Verständnis hierfür zu entwickeln, wäre es beispielsweise nützlich, die Zahl reeller, positiver Lösungen für kleine Fälle zu berechnen.

## Literatur

- [1] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: Software for numerical algebraic geometry. Available at [bertini.nd.edu](http://bertini.nd.edu).
- [2] D. N. Bernstein. The number of roots of a system of equations. *Funkcional. Anal. i Prilozhen.*, 9(3):1–4, 1975.
- [3] C. Bohr, K. Hasselbalch, and A. Krogh. Ueber einen in biologischer Beziehung wichtigen Einfluss, den die Kohlensäurespannung des Blutes auf dessen Sauerstoffbindung übt. *Skandinavisches Archiv Für Physiologie*, 16(2):402–412, 1904.
- [4] W. E. Briggs. The relationship between zeros and factors of binding polynomials and cooperativity in protein-ligand binding. *J Theor Biol*, 114(4):605–614, 1985.
- [5] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 4-1-0 — A computer algebra system for polynomial computations. Available at [www.singular.uni-kl.de](http://www.singular.uni-kl.de), 2016.
- [6] Anders N. Jensen. Gfan, a software system for Gröbner fans and tropical varieties. Available at [home.imf.au.dk/jensen/software/gfan/gfan.html](http://home.imf.au.dk/jensen/software/gfan/gfan.html).
- [7] Anders Nedergaard Jensen. Tropical homotopy continuation, 2016.
- [8] J. W. R. Martini, M. Schlather, and G. M. Ullmann. On the interaction of two different types of ligands binding to the same molecule part I: basics and the transfer of the decoupled sites representation to systems with  $n$  and one binding site *J Math Chem*, 51(2):672–695, 2013.
- [9] J. W. R. Martini, M. Schlather, and G. M. Ullmann. On the interaction of different types of ligands binding to the same molecule Part II: systems with  $n$  to 2 and  $n$  to 3 binding sites *J Math Chem*, 51(2):696–714, 2013.
- [10] J. W. R. Martini, L. Diambra and M. Habeck. Cooperative binding: a multiple personality. *J Math Biol*, 72(7): 1747–1774, 2016.
- [11] A. Onufriev, D. A. Case and G. M. Ullmann A novel view of pH titration in biomolecules. *Biochem*, 40(12):3413–3419, 2001.
- [12] A. Onufriev, D. A. Case and G. M. Ullmann. Decomposing complex cooperative ligand binding into simple components: connections between microscopic and macroscopic models. *J Phys Chem B*, 108 (30):11157–11169, 2004.
- [13] Y. Ren, J. W. R. Martini, and J. Torres. Decoupled molecules with binding polynomials of bidegree  $(n, 2)$ . arXiv:1711.06865, Submitted.
- [14] J. A. Schellman. Macromolecular binding. *Biopolymers*, 14 (5): 999–1018, 1975.

### A browser interface to the Giac/Xcas CAS

**Bernard Parisse**  
(Université de Grenoble I)

bernard.parisserie@univ-grenoble-alpes.fr



---

#### Introduction

---

Giac is a general purpose computer algebra system library written in C++. Xcas is a user interface to Giac, it is popular in French education but Giac/Xcas is not very well known outside France. Giac covers most symbolic computations from high-school to university levels, and has good performance for fast multivariate polynomial arithmetic (including Groebner basis computations) and linear algebra.

Many CAS have a small kernel and a large math library written in a user language. Giac implements a user language (with syntactic sugar for people used to Maple, Python, or TI CAS calculators), but all built-in commands of Giac are implemented in C++ and can be interfaced to all languages interfacing to C++ (or even C, interacting with the CAS kernel using C strings `char *`). This makes interfacing with other projects easy, and Giac is already interfaced with some codes (free and commercial):

- Xcas (GPL): it is the user interface I developed for Giac,
- Geogebra (GPL): Giac is the math kernel for the CAS window,
- the HP Prime calculator is running Giac as CAS,
- some applications, like PocketCAS on iOS (commercial), are using Giac for their math kernel.

I will discuss here a new Javascript interface to Giac that can be used as an alternative to a classic CAS. It is the math kernel of Xcas for Firefox, a CAS GUI for desktops and mobile devices, and may also be used in  $\text{\LaTeX}$  documents to produce interactive math-enabled documents. The main difference with many web-based CAS user interfaces is that the computations are done *locally* by the web browser instead of requiring net access for each computation (sending the computation to a server and getting back the answer). It is therefore possible to install Xcas for Firefox on a mobile device and

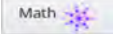



run it in airplane mode – this might be an alternative to calculators in education.

---

#### Xcas for Firefox

---

You can test Xcas for Firefox at [www-fourier.ujf-grenoble.fr/~parisse/xcasen.html](http://www-fourier.ujf-grenoble.fr/~parisse/xcasen.html)

The  button opens wizards for common CAS operations. For programming use . Click on , then  to see some examples of sessions.

#### Differences with native application

The feature set of Xcas for Firefox is not as complete as the Xcas native application feature set, for example it is not possible to speed up parallelizable algorithms (like modular determinant, modular Groebner basis, ...) on multi-core architectures. This does not impact educational use.

On the other hand, Xcas for Firefox is much more transparent to the user, because it does not require installation. Exchanging a worksheet is as easy as writing an email, because a worksheet can be encapsulated in an HTML link along with metadata like the e-mail address of the sender. Clicking on the Mail link of Xcas for Firefox will therefore automatically open the email client of the user with the e-mail address filled, either with the sender e-mail or with the default from Settings.

#### Technical details and performances

The C++ CAS code is compiled to Javascript (instead of machine language) using emscripten, then the user-interface code written in Javascript interacts with a Javascript entry point to the C++ code using strings. More precisely, the C-like Giac function

```
const char * caseval(const char *)
```

is exported with a compile flag

```
-s EXPORTED_FUNCTIONS=["['_caseval']"]
```



and is made callable from Javascript with



```
var docaseval = Module.cwrap(
  'caseval', 'string', ['string'] );
```

We will see that the performance of floating-point operations (with `double`) is not far from a natively compiled single-threaded application, thanks to `asmjs`, a subset of Javascript that the browser can translate to native code on the fly. Unfortunately, Javascript does not provide a type for 64-bit integers, which is a much bigger performance hit for exact computations, especially for operations like Groebner-basis computations (for example, `cyclic7` takes more than 15 seconds instead of about 2 s natively). Recent versions of desktop browsers implement a new standard named web-assembly, and there the performance loss is not as severe, Groebner-basis computations like `cyclic7` are 2 to 3 times slower than single-threaded native programs, and this is acceptable unless you run really large computations.

Following are a few benchmarks with Xcas 1.4.9-53 and Firefox 58 on a Mac (Core i5) using the interface at [www-fourier.ujf-grenoble.fr/~parisse/xcasen.html](http://www-fourier.ujf-grenoble.fr/~parisse/xcasen.html)

Click on the Settings button , then on  to switch between `asmjs` and web-assembly (if your browser supports it).

### Approx. linear algebra benchmarks

#### LU decomposition:

```
n:=500;
m:=ranm(n,n,uniformd,-1,1);
time(p,l,u:=lu(m));
maxnorm(l*u-permu2mat(p)*m)
native: 0.05s, wasm: 0.045s, asmjs: 0.055s
```

#### Giac built-in QR decomposition:

```
n:=500;
m:=ranm(n,n,uniformd,-1,1);
time(q,r:=qr(m,-1));
maxnorm(m-q*r);
native: 0.08s, wasm: 0.09s, asmjs: 0.1s
```

#### Schur decomposition:

```
n:=500;
m:=ranm(n,n,uniformd,-1,1);
time(p,q:=schur(m));
maxnorm(m-p*q*trn(p));
native: 0.45s, wasm: 0.5s, asmjs: 0.5s
```

### Exact benchmarks from the Nemo test suite

#### Series expansion:

```
n:=200; series("t");
u:= t + O(t^n);
time(r:=(u/(exp(u)-1))*exp(x*u));
native: 0.6s, wasm: 3.6s, asmjs: 2.9s.
```

#### Power of a polynomial with coefficient in a field extension of $\mathbb{Q}$ :

```
x:=rootof(cyclotomic(20));
f:=(3x^7 + x^4 - 3x + 1)*y^3 +
(2x^6-x^5+4x^4-x^3+x^2-1)*y +
(-3x^7+2x^6-x^5+3x^3-2x^2+x);
p:=symb2poly(f,y,[]);
time(s:=p^800);
native: 0.3s, wasm: 1.9s, asmjs: 1.7s
```

#### Determinant of a matrix with coefficients in a field extension of $\mathbb{Q}$ :

```
purge(x);
n:=80;
a:=rootof(x^3+3x+1);
m:=ranm(n,n,a);
time(d:=det(m));
native: 6s, wasm: 24s, asmjs: 24s
```

#### Determinant of a matrix with coefficients in a field extension of $\mathbb{Z}/p\mathbb{Z}$ :

```
p := 2003*1009; n:=40;
f(j,k):={
  k:=rand(6);
  return randpoly(x,k) mod p;
};
m:=matrix(n,n,f);
time(det(m));
native: 0.45s, wasm: 0.4s, asmjs: 0.7s
```

#### Characteristic polynomial of a random matrix with integer coefficients:

```
n:=100; m:=ranm(n,n,-21);
time(charpoly(m));
native: 0.05s, wasm: 0.11s, asmjs: 0.9s
```

#### Minimal and characteristic polynomial of a random matrix with coefficients in a non-prime finite field. A large random matrix is constructed by blocks from a small one and similarity transforms are applied:

```
restart(1);
n1:=30; n:=2*n1; GF(103,2,g);
m:=ranm(n1,n1,g);
M:=blockmatrix(2,2,[m,0*m,0*m,m]);
for j from 1 to 10 do
  p:=idn(n);
  q:=p;
  r:=rand(n);
  d:=ranv(1,-4)[0];
  p[r]:=seq(d,n);
  p[r,r]:=1;
  q[r]:=seq(-d,n);
  q[r,r]:=1;
  M:=q*M*p;
od;
time(p:=pmin(M)); time(q:=pcar(m));
p-q;
native: 0.38 and 0.06, wasm: 0.55 and 0.08,
asmjs: 0.63 and 0.09
```



## Groebner-basis benchmark (available in Doc, Examples in Xcas for Firefox, remove `mod p` for computation on $\mathbb{Q}$ )

*cyclic7 modulo prevprime(2<sup>24</sup>):*

native: 0.3s, wasm: 0.4s, asmjs: 1.9s

*cyclic7 on  $\mathbb{Q}$ :*

native: 1.7s, wasm: 5.3s, asmjs: 16s

---

## Interactive documents written in $\text{\LaTeX}$

---

Combining  $\text{\LaTeX}$  rendering quality and CAS computing is not new:

1. many math softwares provide converters to export data to a  $\text{\LaTeX}$  file,
2. some programs handle both  $\text{\LaTeX}$ -like rendering and computation, e.g. `texmacs` or `lyx`.

In the first case, however, if the writer changes some inputs in his computation, he must export again the result and include it in his  $\text{\LaTeX}$  file, and in the second case the data format is not standard  $\text{\LaTeX}$  and requires additional software to be installed on the reader device or a net access to a server to run the computations.

The solution presented here is new in that the writer edits a standard  $\text{\LaTeX}$  file, adds a few simple commands like `\giacinputmath{factor(x10-1)}` or `\giacinput{plot(sin(x))}` and compiles it to produce an HTML5+MathML document. The reader can see the document in any browser (it is optimized for Firefox), and he can modify computation command-lines and run them.

The writer can also compile the  $\text{\LaTeX}$  file to PDF for printing purposes by running `giac file.tex`, where `giac` will precompute the Giac commands in an intermediate tex file and call `pdflatex` on it.

### On the writer side

The writer must install `hevea` (`hevea.inria.fr`) or `hevea-mathjax`, `Giac/Xcas` for computing-enabled output and `heveatomml` for MathML output. The files `giac.tex`, `hevea.sty`, `mathjax.sty`, and `giac.js` must be copied to the  $\text{\LaTeX}$  working directory.

The writer opens a  $\text{\LaTeX}$  file with his usual editor. In the preamble he adds the following lines:

```
\makeindex
\input{giac.tex}
\giacmathjax
```

For support of interactive CAS  $\text{\LaTeX}$  commands, the writer should add

```
\begin{giacjshare}
\tableofcontents
\printindex
```

just after `\begin{document}` and

```
\end{giacjshare}
```

just before `\end{document}`. Printing the table of contents and index before the first  $\text{\LaTeX}$  section command is recommended, otherwise the HTML output Table and Index buttons will not link correctly.

The rest of the source file is standard  $\text{\LaTeX}$  except that new commands are available for interactive CAS support, for example:

- `\giacinputmath{cmdline}` and `\giaccmdmath{cmd}{args}` embed an interactive command (in the second form only `args` is interactive), whose MathML or plot output is typeset in  $\text{\LaTeX}$ 's inline math style ( $\$...\$$ ). Both take an optional HTML 'style' argument for detailed formatting control.  
Example:  
`\giacinputmath{factor(x10-1)}`  
`\giaccmdmath{factor}{x4-1}`
- `\giacinputbigmath`, `\giaccmdbigmath` the same but typeset in  $\text{\LaTeX}$ 's display math style ( $\$\$...\$\$$ ).
- `\begin{giacprog}...\end{giacprog}` holds a program or multi-line command. The program is run whenever the user presses the ok button. To have the program executed already at load time, replace `giacprog` with `giaconload`.
- `\giacslider{v}{vmin}{vmax}{ $\Delta v$ }{v0}{cmd}` adds a slider. When the user modifies the slider interactively, the new value is stored in variable `v` and `cmd` is executed.

Example:

```
\giacslider{a}{-5}{5}{0.1}{0.5}
{plot(sin(a*x))}
```

Once the source file is written, it is compiled to HTML5 with the command

```
hevea2mml sourcefile.tex
```

The HTML output and the `giac.js` files should be available to the Web server in the same directory.

For a more precise description, please refer to `www-fourier.ujf-grenoble.fr/~parisse/giac/castex.html`

### On the reader side

The reader's browser opens an HTML5+MathML file (linking to the `giac.js` Javascript). The MathML is rendered natively on Firefox or Safari, while Chrome or Internet Explorer will automatically load MathJax to render MathML – this is of course noticeably slower if the document is large. Computations are run by the reader's browser (the CAS is Javascript code). This is

```

\newcommand{\giacinput}[2][style="width:400px;height:20px;font-size:large"]{
  \ifhevea
    \@print{<textarea onkeypress="UI.ckenter(event,this,1) "}
    \@getprint{#1>#2}
    \@print{</textarea>
      <button onclick="
        previousSibling.style.display='inherit';
        var tmp=UI.caseval(previousSibling.value);
        tmp=UI.rmquote(tmp);
        nextSibling.innerHTML='&nbsp;'+tmp;
        UI.render_canvas(nextSibling);
      ">ok</button>
      <span></span>
      <br>}
  \else
    \lstinline@#2@
  \fi
}

```

**Figure 1:** The definition of *giacinput* with HTML code highlighted in blue and Javascript in red.

slower than native code but faster than network access to a server and it does not require setting up a specific server for computations.

### How this is done

All `\giac...` commands are defined in `giac.tex`. An example definition is shown in Fig. 1.

In general, if `hevea` compiles the file, the `\ifhevea` part is active, and the command will output an HTML5 `<textarea>` element and a OK `<button>`, with a callback to Javascript code that will evaluate the CAS command through `UI.caseval` and fill the next HTML5 `<span>` field with the result.

The CAS evaluation is performed by the same method as inside Xcas for Firefox.

## References

- [1] L. Marandet, Hevea: LaTeX to HTML5 compiler, <http://hevea.inria.fr>, 2017.
- [2] B. Parisse and R. D. Graeve, Giac/Xcas Computer Algebra System, <http://www-fourier.ujf-grenoble.fr/~parisse/giac.html>, 2017.
- [3] A. Zakai, Emscripten: C/C++-to-Javascript compiler, <http://kripken.github.io/emscripten-site>, 2017.

# Computations with Galois groups in Magma

Nicole Sutherland  
(University of Sydney)

nicole.sutherland@sydney.edu.au




---

## Introduction

---

This article discusses some computations which use Galois groups. The computations I will mention are available in MAGMA [1] for some coefficient rings. The main Galois group algorithm has been previously discussed in [3] and the computer algebra system MAGMA has also been previously discussed in [4]. We will first consider the computation of the Galois group itself, look at some examples and then at further computations. Both algebraic number fields and function fields will be considered.

Elements of the Galois group of a polynomial will permute the roots of that polynomial. In the late 1990s to early 2000s there were a number of algorithms which improved on each other by increasing the degree of polynomials they could handle. The algorithm we use and build on is that of Fieker and Klüners [2] which removed degree bounds altogether. This is because they do not use tabulated information which was used before.

The Magma transcripts in this article have been edited for better readability, with user input shown in blue. See [9] and [8] Chapters 7–9 for further details of the Galois group algorithm mentioned below.

---

## Main algorithm

---

We provide here a brief description of the algorithm. An extended description of this algorithm can be found in [9]. We start by computing a splitting field for the polynomial – a local one works well – and the roots of the polynomial.

Since we use the top–down approach of Stauduhar [6] we compute a group  $G$  which will contain the Galois group of  $f$ , the smaller the better, by considering the Galois groups of the subfields of the extension of  $F$  by a root of  $f$ .

Then we look through the maximal subgroups of  $G$  and check whether any of these contain the Galois group of  $f$ . To do this efficiently, we would like to check the cheapest subgroup first. So we compute polynomials invariant under a maximal subgroup of  $G$  but not invariant under  $G$  and also determine a cost for attempting a descent into each of these subgroups.

Once we have decided which subgroup  $H$  to check next we evaluate the invariant for this subgroup at the

roots of  $f$ , which must be known to some precision in order for us to make a proven decision. This precision depends on the index of  $H$  in  $G$  as well as the degree of the invariant. If the evaluation of the invariant at the roots of  $f$  is in the coefficient field for a representative  $\tau$  of some coset of  $H$  in  $G$ , then we have a smaller group which contains the Galois group of  $f$  by a theorem of Stauduhar [6]. We start the loop again and check whether the Galois group of  $f$  is contained in any of the maximal subgroups of this new group  $G$ .

This process stops when we run out of maximal subgroups of the smallest group known to contain the Galois group which shows that this group is the Galois group of  $f$ .

As simply stated this algorithm works for all number fields and global function fields. The differences between the coefficient fields are in the details which are explained in [9].

### Example over $\mathbb{F}_q(t)$ ([8] Ex. 12, [9] Ex. 1)

Let  $F = \mathbb{F}_7(t)$  and  $f = x^8 + t + 1 \in F[x]$ ,  $\text{Gal}(f) \subseteq S_8$  with order 40320. We are able to compute subfields of the extension of  $F$  defined by  $f$  and this gives us a group of order 64 to begin with which contains the Galois group instead of the order-40320 symmetric group. This group has 6 maximal subgroups.

```
> SetVerbose("GaloisGroup", 3);
> F<t> := FunctionField(GF(7));
> P<x> := PolynomialRing(F);
> G, R, S := GaloisGroup(x^8 + t + 1);
Degrees of subfields [ 4, 2 ]
Computing group of subfield given by
x^4 + t + 1
```

```
Proven subfield group (D_4) of order
8 found. Reduced order of starting
group using subfields to 64, TGI: 8T26
```

```
Have to consider 6 subgroups (classes)
```

```
Lifting roots in Power series ring
over GF(7^16) to precision 10
```

```
Further reduce to 4 (using rejected)
Further reduce to 2 (using sieve)
```

Some other information means we need to consider only 2 of them. The first group we look at does not contain the Galois group so we move onto the other group which we find does contain the Galois group.

```
Stauduhar: group index 2 (TGI: 8T15)
no cosets remain, group impossible
```

```
Stauduhar: group index 2 (TGI: 8T15)
Found 2 cosets as simple zeros and 0
cosets as multiples DESCENT
```

```
Try to descend from group of order 32
Have to consider 6 subgroups (classes)
Further reduce to 4 (using rejected)
```

We need to consider whether the Galois group is contained in any of 4 of its maximal subgroups.

```
Stauduhar: group index 2: D_8
no cosets remain, group impossible
```

```
Stauduhar: group index 2 (TGI: 8T8)
```

```
Stauduhar: group index 2: (TGI: 8T8)
no cosets remain, group not possible
```

```
Stauduhar: group index 2 = D_8
Stauduhar: group index 2: (TGI: 8T8)
```

The Galois group is not contained in the first subgroup we look at but then we consider 3 of the other subgroups and find that it is not contained in a conjugacy class of  $8T8$  but it is contained in another conjugacy class of  $D_8$ .

```
Stauduhar: group index 2 (D_8)
Found 2 cosets as simple zeros and
0 cosets as multiples DESCENT
```

We now have a group of order 16 containing the Galois group and we can use other information to determine that none of its maximal subgroups contain the Galois group so the Galois group is this group of order 16 which is the dihedral group  $D_8$ . We see some roots of  $f$  in the splitting field  $Z$  computed from a chosen prime.

```
Time: 0.360
> TransitiveGroupDescription(G); G;
D(8)
Permutation group G acting on a set
of cardinality 8 Order = 16 = 2^4
      (2, 8) (3, 7) (4, 6)
      (1, 2) (3, 8) (4, 7) (5, 6)
      (1, 3, 5, 7) (2, 4, 6, 8)
      (1, 5) (2, 6) (3, 7) (4, 8)
> Z<z> := Universe(R);
> W<w> := CoefficientRing(Z);
> WW<ww> := Parent(Eltseq(
>           Eltseq(R[1])[1])[1]);
> Z, R;
Power series ring in z over GF(7^16)
[ (5*ww + 5)*w^3 + ... + O(z^4),
  (5*ww + 2)*w^3 + ... + O(z^4),
  (3*ww + 1)*w^3 + ... + O(z^4),
  (4*ww + 1)*w^3 + ... + O(z^4),
  .
  . ]
```

## Example over an extension of $\mathbb{F}_q(t)$

We can do the same computation when the polynomial is over an algebraic instead of a rational function field.

```
> F<t> := FunctionField(GF(7));
> P<x> := PolynomialRing(F);
> FF<a> := FunctionField(x^2 + t);
> P<x> := PolynomialRing(FF);
> time G := GaloisGroup(x^8 + a + 1);
Time: 0.460
> TransitiveGroupDescription(G); G;
D(8)
Permutation group acting on a set of
cardinality 8 Order = 16 = 2^4
      (1, 8) (2, 7) (3, 6) (4, 5)
      (1, 8, 7, 6, 5, 4, 3, 2)
      (1, 3, 5, 7) (2, 4, 6, 8)
      (1, 5) (2, 6) (3, 7) (4, 8)
```

## Examples of polynomials with degree $> 23$

This example shows the ability of this algorithm to compute Galois groups of some polynomials of fairly large degree – much larger than the degree limits imposed by most earlier algorithms.

```
> F<t> := FunctionField(GF(7));
> P<x> := PolynomialRing(F);
> f := x^143 + t + 4;
> time G := GaloisGroup(f); G;
Time: 1338.900
Permutation group G acting on a set
of cardinality 143
Order = 8580 = 2^2 * 3 * 5 * 11 * 13
> f := x^201 + t + 4;
> time G := GaloisGroup(f); G;
Time: 3554.240
Permutation group G acting on a set
of cardinality 201
Order = 13266 = 2 * 3^2 * 11 * 67
```

---

## Reducible polynomials

---

Since Galois groups permute the roots of a polynomial they can also be computed for reducible polynomials. There are a few adjustments we need to make to the algorithm – the main one being our choice of starting group. Since the Galois group will not be transitive, it may not be contained in the symmetric group. Instead, the Galois group of a reducible polynomial will be contained in the direct product of the Galois groups of its factors.

While we cannot find a smaller group containing the whole Galois group, we can apply some knowledge of ramified and unramified extensions to consider how the splitting fields of the irreducible factors interact. If we can determine that the splitting fields of some factors do not overlap with all the others, we do not need to include the associated Galois groups in the starting group for the descent, but compute the product of them with the result of the smaller descent afterwards. The more the splitting fields of the factors overlap the smaller the splitting field of the reducible polynomial and the Galois group.

Over  $\mathbb{Q}$  there are no non-trivial unramified extensions. In the function-field case we know ramification is related to the extension of the constant field since  $\mathbb{Q}$  and  $\mathbb{F}_q$  are perfect [7]. For function fields constant field extensions are unramified and extensions not extending the constant field are totally ramified. We can investigate the ramified extensions further using discriminants to determine whether there is any non-trivial intersection of splitting fields of factors. If all pairs of splitting fields of factors intersect trivially then the degree of the splitting field will be the product of the degrees of the splitting fields of the factors but as this is the order of the direct product of the Galois groups the Galois group of the product can be no smaller.

### Example of reducible polynomial over $\mathbb{F}_q(t)$ ([8] Ex. 17, [9] Ex. 6)

We will look at an example of a Galois group computation of a reducible polynomial.

After computing the Galois groups of the 4 irreducible factors we first look at the orders of these groups. Since the order of the second group is coprime to the orders of the others its splitting field has trivial intersection with the splitting fields of all the other factors. We run a few other tests based on the ramification of the splitting fields and find that the 1st splitting field also has trivial intersection with the splitting fields of all the other factors.

```
> SetVerbose("GaloisGroup", 3);
> F<t> := FunctionField(GF(101));
> P<x> := PolynomialRing(F);
> f := (x^2 + x + 3*t)^5*(x^5 + 5*t)
>      *(x^7 + 7*t)*((x + 1)^7 + 7*t);
> time G := GaloisGroup(f);
Factor 1 = C_2 = S_2
Factor 2 = C_5
Factor 3 = TGI: 7T4
Factor 4 = TGI: 7T4
```

```
maybe_coprime by order 1, 3, 4
disc_non_coprime 3, 4
maybe_coprime after stem check 1
non_coprime after stem check 3, 4
After cyclic subgroup check: 1
maybe_coprime after fixed field test
Final non_coprime: 3, 4
```

So when we look through subgroups to find the Galois group we do this only for the direct product of the 3rd and 4th factors and take the direct product of the Galois groups of the first two factors, which has order 10, with the result. This reduced the size of the groups being considered by a factor of 10.

```
starting group order 17640
Try descent from group of order 1764
```

```
Permutation group Gc acting on a set
of cardinality 21 Order = 10 = 2 * 5
Permutation group _gnc acting on a
set of cardinality 21
Order = 42 = 2 * 3 * 7
Time: 2.640
```

```
> G;
Permutation group G acting on a set
of cardinality 29
Order = 420 = 2^2 * 3 * 5 * 7
```

---

## Fixed Fields

---

Now that we can, in theory, compute Galois groups of all polynomials we move on to computing fixed fields of subgroups  $U$  of these Galois groups. The computation is similar. We compute an invariant and roots to some useful precision and from this we can compute a polynomial with at least one root fixed by  $U$  and with degree that of the index of  $U$  in  $G$ . If  $U$  is smaller than the Galois group, then none of the roots will lie in the coefficient ring of  $f$ . This polynomial will define the fixed field of the given subgroup and can be mapped back to be over the coefficient ring of the original polynomial.

We see that a defining polynomial for a fixed field can be computed fairly quickly.

```
> P<x> := PolynomialRing(Rationals());
> K<a> := NumberField(x^3 + 2);
> P<y> := PolynomialRing(K);
> time G, R, S := GaloisGroup(
>      y^8 + a + 3); G;
Time: 0.330
Permutation group G acting on a set
of cardinality 8 Order = 32 = 2^5
```

We take a normal subgroup of  $G$  in a conjugacy class with  $C_8$ .

```
> subg := NormalSubgroups(G);
> time Polynomial(K, GaloisSubgroup
>      (S, subg[12]'subgroup));
y^4 + (32*a + 128)*y^3 + ...
Time: 0.050
```

---

## Splitting Fields

---

There are two ways we can compute a splitting field from a known Galois group. The splitting field will be the field fixed by only the trivial subgroup. But using the fixed field algorithm above this will compute the splitting field as a single extension of the coefficient field of the polynomial.

We can also compute the splitting field as a tower of extensions of the coefficient field. We first compute a chain of subgroups and then compute their fixed fields but we need to find defining polynomials over a field in the tower rather than over the coefficient field of the original polynomial.

Continuing the previous example we see here the splitting field computed as a direct extension of  $K$  which was the coefficient field of the polynomial above. The coefficient ring is not restricted to a rational field.

```
> time G, R, S := GaloisGroup(
>      y^8 + a + 3); G;
> tG := sub<G | G.0>;
> time NumberField(Polynomial(K,
>      GaloisSubgroup(S, tG))): Maximal;
```



```

$1
|
K<a>
|
Q

$1 : y^32 - (3452a + 10356)y^24 + ...
K : x^3 + 2
Time: 0.070

```

The splitting field is computed below as a tower of extensions over  $K$  using the second method explained above. We can see the field defined by the input polynomial as well as the further extensions are required to find the splitting field.

```

> time GSF := GaloisSplittingField(
>      y^8 + a + 3);
Time: 0.840
> KK<aa> := CoefficientField(GSF);
> <yy> := PolynomialRing(KK);
> GSF:Maximal;
$1
|
KK<aa>
|
K<a>
|
Q
$1 : yy^4 + aa^4
KK : y^8 + a + 3
K : x^3 + 2

```

## Solution of polynomials by radicals

Galois Theory has its beginning in the attempt to solve polynomial equations by radicals. It is reasonable to expect then that we could use the computation of Galois groups for this purpose. We first compute the Galois group and check it is solvable. If so, since a solvable finite group is a group with a composition series all of whose factors are cyclic groups of prime order, we can use this series  $C$  of subgroups to compute a splitting field as a tower of cyclic extensions which we can then convert to radical extensions. This conversion requires handling the necessary roots of unity.

### Example of a solution of polynomial by radicals

Below is a degree-6 polynomial which can be solved by radicals, though many cannot be. Computing the `GaloisSplittingField` of this polynomial results in a degree-2 extension of the degree-6 extension defined by this polynomial. Solving by radicals results in this splitting field being expressed as 3 extensions. The difference comes from the chain of subgroups used – in this case using only cyclic subgroups we used more subgroups and so there are more extensions.

```

> S<s>,rt:=SolveByRadicals(
>      x^6+15*x^4+4*x^3+75*x^2-60*x+129);
> CS<cs> := CoefficientRing(S);
> Qa<qa> := CoefficientRing(CS);
> S:Maximal;

```

```

S<s>
|
S : $.1^3 + 648qa*csr - 4536qa
CS<cs>
|
CS : $.1^2 + 5
Qa<qa>
|
Qa : x^2 + 3
Q
> qa^2 + 3, cs^2 + 5,
>      s^3 + 648*qa*cs - 4536*qa;
-3 -5 -648*qa*csr + 4536*qa
> rt;
[
(1/972*cs - 1/486)*s^2 +/- cs,
(1/1944*(-/+qa - 1)*cs + ...
(1/1944*(-/+qa - 1)*cs + ...
]

```

## Explicit Hilbert Irreducibility

It is also possible to compute Galois groups over function fields of characteristic 0.

Hilbert's Irreducibility Theorem says that for any polynomial over  $\mathbb{Q}(t)$  the specializations of the polynomial at infinitely many rational numbers factor the same way as the original polynomial and the Galois group of the specialization is isomorphic to the Galois group of the original. What my co-author David Krumm was interested in was what happens when the specialization has a different type of factorization and non-isomorphic Galois group. He has proven this theorem in our submitted paper [5] in which we consider the question:

Let  $P \in \mathbb{Q}[t, x]$ ,  $G = \text{Gal}(P)$  be the Galois group of  $P$  over  $\mathbb{Q}(t)$ . For most rational  $c$ ,  $P_c = P(c, x) \in \mathbb{Q}[x]$  has  $\text{Gal}(P_c) \cong G$  and factors in the same way as  $P$ . For which  $c$  does this not occur?

We require that  $c$  not be a root of the discriminant or leading coefficient. Also  $c$  must not make the discriminants of the defining polynomials of the fixed fields identically zero. Then we have the equivalence of the non-isomorphic Galois groups and the specialization of one of the fixed field defining polynomials having a rational root. If there is no rational root, then the Galois groups are isomorphic and the factorization types are the same.

But in order to do the necessary computations we need to be able to compute Galois groups over  $\mathbb{Q}(t)$ , including when  $P$  is reducible which had not been implemented previously, and compute fixed fields of subgroups of the Galois group. Once we obtain the  $f_i$  we may then find suitable  $c$  using rational points on the curves defined by  $f_i$ .

Some of the Galois group computations necessary are shown in this example.

### Example over $\mathbb{Q}(t)$ ([5] Sect. 4.1)

```

> Qt<t> := FunctionField(Rationals());
> P<x> := PolynomialRing(Qt);
> f := x^6 + t^6 - 1;
> time G, r, S := GaloisGroup(f);
Time: 0.190
> Max := Reverse(MaximalSubgroups(G));

```

```

> // smallest index first
> for M in Max do
for> GaloisSubgroup(S, M`subgroup);
for> end for;
x^2 + 6*x + 9*t^6
x^2 + 1728*t^12 - 3456*t^6 + 1728
x^2 - 62208*t^30 + 311040*t^24 -
      622080*t^18 + 622080*t^12
      - 311040*t^6 + 62208
x^3 + 12*x^2 + 48*x - 8*t^6 + 72

```

As in [5] it can be shown theoretically using known elliptic curves that none of the curves defined by these subfield polynomials have a rational root when  $c$  is not 0, 1 or  $-1$ . Therefore the Galois group of  $f$  specialised at  $c$  is isomorphic to  $G$  unless  $c$  is 0, 1, or  $-1$  and  $f$  specialised at  $c$  must be irreducible when  $c$  is not 0, 1, or  $-1$ . It can be directly seen that  $f$  specialised at 0, 1,  $-1$  are reducible.

### Example of reducible polynomial over $\mathbb{Q}(t)$

We now consider an example involving a Galois group computation for a reducible polynomial over  $\mathbb{Q}(t)$ .

```

> k<t> := FunctionField(Rationals());
> <x> := PolynomialRing(k);
> Phi4 := x^12 + 6*t*x^10 + x^9 +
> (15*t^2 + 3*t)*x^8 + 4*t*x^7 +
> (20*t^3 + 12*t^2 + 1)*x^6 +
> (6*t^2 + 2*t)*x^5 + (15*t^4 +
> 18*t^3 + 3*t^2 + 4*t)*x^4 +
> (4*t^3 + 4*t^2 + 1)*x^3 +
> (6*t^5 + 12*t^4 + 6*t^3 + 5*t^2
> + t)*x^2 + (t^4 + 2*t^3 + t^2 +
> 2*t)*x + t^6 + 3*t^5 + 3*t^4 +
> 3*t^3 + 2*t^2 + 1;
> ep4 := Polynomial([Evaluate(y,
> (4 - 3*t - t^3)/(4*t)) :
> y in Coefficients(Phi4)]);
> #GaloisGroup(Phi4);
384
> #GaloisGroup(ep4);
128
> eep := Polynomial([Evaluate(y,
> (t^2-1)/t) :
> y in Coefficients(ep4)]);
> #GaloisGroup(eep4);
64

```

Phi4 is a polynomial of interest in arithmetic dynamics having roots of period 4 under iteration of  $x^2 + t$ . Using the procedure above we can determine that Phi4 has a different factorization type and Galois group when evaluated at something of the form  $(4 - 3v - v^3)/4v$ . But to determine what Galois group Phi4 specialised at  $c$  has in these cases we need to compute the Galois group of the reducible polynomial ep4, a product of a degree-8 and a degree-4 polynomial. Further we can apply the procedure above to ep4 which tells us that when  $v$  has the form  $(s^2 - 1)/s$  then the Galois group and factorization type is different again. In order to be able to do these calculations we needed to be able to compute Galois groups of reducible polynomials over char-0 rational function fields and their associated fixed fields.

## Geometric Galois groups

**Definition 1** The *geometric Galois group* of a polynomial  $f \in \mathbb{Q}(t)[x]$ ,  $\text{GeoGal}(f)$ , is the Galois group of  $f$  considered as a polynomial over  $\mathbb{C}(t)$ ,  $\text{Gal}(f/\mathbb{C}(t))$ .

A geometric Galois group is a subgroup of the Galois group over  $\mathbb{Q}(t)$  – as the field is larger the group is smaller. Geometric Galois groups are connected to inverse Galois theory and are an alternate approach to computing absolute factorizations.

Again we start with computing a Galois group – this time over  $\mathbb{Q}(t)$ . Using HIT we know we can find  $t_1, t_2$  at which the specialization has Galois group isomorphic to  $G$ . We compute the Galois group over  $\mathbb{Q}$  of the product of the specializations, which is a subgroup of the direct product of  $G$  with itself.

We have some bounds on the index in  $G$  and the order of the geometric Galois group using the fixed field of the geometric Galois group which is also the algebraic closure of  $\mathbb{Q}$  in the splitting field of  $f$  over  $\mathbb{Q}(t)$ . We compute this fixed field at the same time as the geometric Galois group.

We consider normal subgroups that satisfy these bounds, compute their fixed fields and then look at which of these define constant field extensions. This gives us the subgroup of the geometric Galois group and its fixed field which is the algebraic closure of  $\mathbb{Q}$  in the splitting field of  $f$  over  $\mathbb{Q}(t)$ .

We summarise the algorithm as follows:

### Algorithm 1 (The geometric Galois group of $f$ )

Let  $f \in \mathbb{Q}(t)[x]$ .

1. Compute  $G = \text{Gal}(f)$ .
2. Choose  $t = t_i, i = 1, 2$ , such that  $\text{Gal}(f(t_i, x)) = \text{Gal}(f) = G$ .
3. Compute  $H = \text{Gal}(f(t_1, x)f(t_2, x))$ .
4. For normal subgroups  $X$  of  $G$  having index less than that of  $H$  in  $G \times G$  and order dividing  $\#G/c$  where  $c$  is the degree of the full constant field of  $\mathbb{Q}(t)[x]/f$ ,
  - (a) Compute the defining polynomial of the field  $K'$  fixed by  $X$ .
  - (b) Check whether this is a polynomial over  $\mathbb{Q}$  or whether this defines a constant field extension. If so  $X$  contains  $\text{GeoGal}(f)$  and  $K \supseteq K'$ .
5. The subgroup  $X$  containing  $\text{GeoGal}(f)$  with the largest index in  $G$  and smallest order corresponds to the largest constant field extension in  $\Gamma$ , and is  $\text{GeoGal}(f)$ .

Most geometric Galois groups are equal to the Galois group of the polynomial but I show here an example where that is not the case.

## A Non-Trivial Example

Let  $f = x^9 - 3x^7 - (6t+6)x^6 + 3x^5 + (12t-6)x^4 + (12t^2 - 84t + 11)x^3 - (6t+6)x^2 + (-12t^2 + 12t + 24)x - 8t^3 - 24t^2 - 24t - 6 \in \mathbb{Q}(t)[x]$ .

1. We compute the Galois group over  $\mathbb{Q}(t)$ ,  $\text{Gal}(f)$  as  $9T8$ , equivalently,  $S_3 \times S_3$  of order 36.
2. Specialising  $f$  at  $t = 1, 2$  gives
$$f_1 = x^9 - 3x^7 - 12x^6 + 3x^5 + 6x^4 - 61x^3 - 12x^2 + 24x - 62,$$

$$f_2 = x^9 - 3x^7 - 18x^6 + 3x^5 + 18x^4 - 109x^3 - 18x^2 - 214$$
with  $\text{Gal}(f_1), \text{Gal}(f_2)$  conjugate to  $9T8$ .
3.  $H = \text{Gal}(f_1 f_2)$  is an intransitive group of order 216.

4. We compute a bound using the order 36 of  $G$  and the order 216 of  $H$  to give us an index bound for  $\text{GeoGal}(f)$  of  $36 \times 36/216 = 6$ . The exact constant field of  $\mathbb{Q}(t)[x]/f$  has degree 3 so the order of  $\text{GeoGal}(f)$  must divide  $36/3 = 12$ .
  - (a) There are two normal subgroups of  $\text{Gal}(f)$ , both isomorphic to  $S_3$  of order 6, which satisfy this index and order restriction.
  - (b) The corresponding fixed fields are defined by  $x^6 + 78732$  and  $x^6 - 54x^4 + 729x^2 + 78732t^2 - 2916$ .

We have 2 possible defining polynomials. One is over  $\mathbb{Q}$  so obviously defines an extension of  $\mathbb{Q}$ . The second polynomial contains a  $t$  so does not obviously define an extension of  $\mathbb{Q}$ . In fact,  $\mathbb{Q}$  is algebraically closed in the extension defined by this second polynomial.

Therefore the geometric Galois group is the first subgroup, which is isomorphic to  $S_3$ , and the extension defined by the first polynomial is the algebraic closure of  $\mathbb{Q}$  in the splitting field of  $f$ .

These calculations have been carried out by calling `GeometricGaloisGroup(f)` in MAGMA 2.23 [1], and took 1.22s.

## The Absolute Factorization

We can also compute a factorization of  $f$  over  $\mathbb{C}$  by computing a factorization of  $f \in \mathbb{Q}(t)[x]$  over  $\mathbb{Q}(t)[\alpha]$ ,

$$f = x^9 - 3x^7 - (6t+6)x^6 + 3x^5 + (12t-6)x^4 + (12t^2 - 84t + 11)x^3 - (6t+6)x^2 + (-12t^2 + 12t + 24)x - 8t^3 - 24t^2 - 24t - 6.$$

- Factor  $f$  over  $\mathbb{Q}(t)[\alpha] = \mathbb{Q}(t)[x]/\langle x^6 + 78732 \rangle$ .
- There are 3 cubic factors of  $f$  over  $\mathbb{Q}(t)[\alpha]$ :

$$y^3 - 1/486 \alpha^4 y^2 + (-1/9 \alpha^2 - 1)y + 1/1458 \alpha^4 - 2t - 2,$$

$$y^3 + (1/972 \alpha^4 - 1/2 \alpha)y^2 + (-1/2916 \alpha^5 + 1/18 \alpha^2 - 1)y - 1/2916 \alpha^4 + 1/6 \alpha - 2t - 2,$$

$$y^3 + (1/972 \alpha^4 + 1/2 \alpha)y^2 + (1/2916 \alpha^5 + 1/18 \alpha^2 - 1)y - 1/2916 \alpha^4 - 1/6 \alpha - 2t - 2.$$

This factorization can be carried out in MAGMA using `Factorization(Polynomial(Qta, f))` where `Qta` is the extension  $\mathbb{Q}(t)[\alpha]$ .

## References

- [1] J. J. Cannon, W. Bosma, C. Fieker, and A. Steel (eds.), *Handbook of Magma functions (V2.23)*, Computational Algebra Group, University of Sydney, 2017, <http://magma.maths.usyd.edu.au>.
- [2] C. Fieker and J. Klüners, *Computation of Galois groups of rational polynomials*, London Mathematical Society Journal of Computation and Mathematics **17** (2014), no. 1, 141 – 158, <http://arxiv.org/abs/1211.3588>.
- [3] C. Fieker and J. Klüners. Galoisgruppen in Magma. *Computeralgebra-Rundbrief*, 43:19–20, Oktober 2008.
- [4] C. Fieker. Magma. *Computeralgebra-Rundbrief*, 43:16–18, March 2003.
- [5] David Krumm and Nicole Sutherland, *Galois Groups over Rational Function Fields and Explicit Hilbert Irreducibility*, Submitted to JSC, arXiv:1708.04932, 2017
- [6] Richard P. Stauduhar, *The determination of Galois groups*, Mathematics of Computation **27** (1973), 981–996.
- [7] H. Stichtenoth, *Algebraic function fields and codes*, Springer–Verlag, 1993.
- [8] N. Sutherland, *Algorithms for Galois extensions of global function fields*, Ph.D. thesis, The University of Sydney, 2015.
- [9] N. Sutherland, *Computing Galois groups of polynomials (especially over function fields of prime characteristic)*, Journal of Symbolic Computation **71** (2015), 73–97.

## Scheinarchitektur

C. Stauch  
(Coswig)

stauch@gymnasiumcoswig.de



---

### Einführung

Mathematik und Kunst müssen keine unüberbrückbaren Gegensätze bilden, bestes Beispiel dafür ist der Goldene Schnitt. Die Entdeckung der perspektivischen Darstellung in der Malerei u. a. durch A. Dürer führte zur Entwicklung eines neuen mathematischen Zweiges: der projektiven Geometrie. In der Kunst wurden die perspektivischen Erkenntnisse zunächst genutzt, um durch die Einbeziehung der dritten Dimension, also der Tiefe der dargestellten Räume, realistischere Gemälde zu schaffen. Später ging man auch dazu über, durch geschickten Einsatz zeichnerischer Mittel Scheinarchitekturen zu erzeugen, z. B. die nicht existierende Kuppel der Jesuitenkirche in Wien, die illusionistisch überhöhten Kuppeln des Innsbrucker Doms oder die ebenfalls nur virtuell kuppelartige Decke des Kaisersaales zu Kremsmünster. In einfacheren Fällen wurde perspektivische Effekte genutzt, um Räume scheinbar zu vergrößern, virtuelle Nischen oder Gewölbe zu gestalten.

Im folgenden Artikel wird ein mathematischer Zugang zur beispielhaften Gestaltung von Scheinarchitekturen skizziert. Die Aufgaben sind geeignet für einen Leistungskurs Mathematik - und selbst hier ist Platz für die Binnendifferenzierung. Die Berechnung von Bildpunkten ist ein Standardproblem, die Erzeugung einer virtuellen Kuppel dagegen sehr anspruchsvoll. Die Beispielberechnungen wurden mit einem ClassPad 400 von Casio durchgeführt, der an meiner Schule allen Schülern zur Verfügung steht.

---

### Raumerweiterung

Zunächst soll eine einfache Raumerweiterung mathematisch beschrieben werden, d. h. es soll eine scheinbare rechteckige Nische entstehen. Als Beispiel wird eine quadratische Fläche mit 3 Metern Seitenlänge scheinbar um 1 Meter nach hinten versetzt. Das Koordinatensystem wird so gewählt, dass die Wand, also die Zeichenebene  $Z$ , die  $y$ - $z$ -Ebene bildet. Die scheinbare Wand,

die entstehen soll, ist die Ebene  $S : x = -1$ . Der Augpunkt  $A$  (von dem aus das Bild betrachtet wird) ist (vom Zeichner) frei wählbar, z. B.  $(3/1/2)$ . Jeder Punkt  $P$  der Ebene  $S$  wird auf die Zeichenebene  $Z$  abgebildet, so entsteht  $P'$  als Schnittpunkt der Verbindungsgerade  $g(AP)$  mit  $Z$ . Sei etwa  $P(-1/2/1)$ , dann gilt die Gleichung der Geraden  $g$  durch die Punkte  $A$  und  $P$

$$g_{AP} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} + t \begin{pmatrix} -4 \\ 1 \\ -1 \end{pmatrix}.$$

Für den Schnittpunkt  $P'$  von  $g$  und  $Z$  folgt  $P' = (0/1.75/1.25)$ .

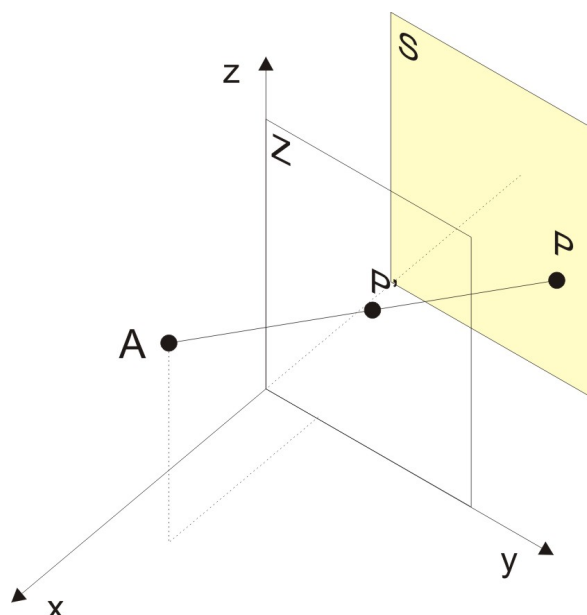


Abbildung 1

**Beispiel für Schüleraufgabe 1:** Berechnen Sie die Bildpunkte  $Q$  und  $R$  zu den Punkten  $P_1(-1/3/3)$  und  $P_2(-1/3/0)$ .

Mit dieser Idee ist es möglich, zu zwei beliebigen Punkten  $P_1$  und  $P_2$  die zugehörigen Bilder  $Q$  und  $R$  in der Zeichenebene  $Z$  zu ermitteln und somit

auch das Bild  $g(QR)$  der Verbindungsgeraden  $g(P_1P_2)$ . Führt man in  $Z$  ein ebenes Koordinatensystem ein, lässt sich die Gerade  $g(QR)$  als lineare Funktion darstellen. Dabei übernimmt die (Raum-)  $y$ -Koordinate eines Bildpunktes  $P'$  in diesem Fall die Funktion der  $x$ -Koordinate im ebenen Koordinatensystem, entsprechend wird die  $z$ -Koordinate des räumlichen Koordinatensystems zur  $y$ -Koordinate des ebenen Koordinatensystems. Mit  $P_1(0/0/0)$  und  $P_2(-1/0/0)$  (also die linke untere Kante unserer Scheinnische) findet man die Gleichung der Bildgeraden  $y = 2x$ . Analog findet man die Gleichungen der anderen Kanten und erhält das perspektivische Bild einer Nische, wenn man die gefundenen Funktionen geeignet einschränkt.

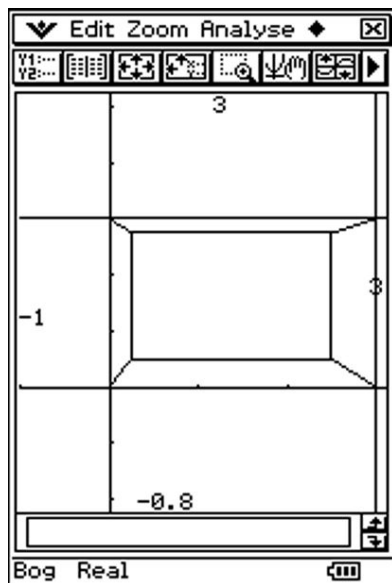


Abbildung 2

In Abb. 3 erkennt man, dass die linearen Funktionen, die die Kanten repräsentieren, sich alle in einem Punkt schneiden. Da es sich um eine perspektivische Darstellung handelt, ist dies nicht überraschend.

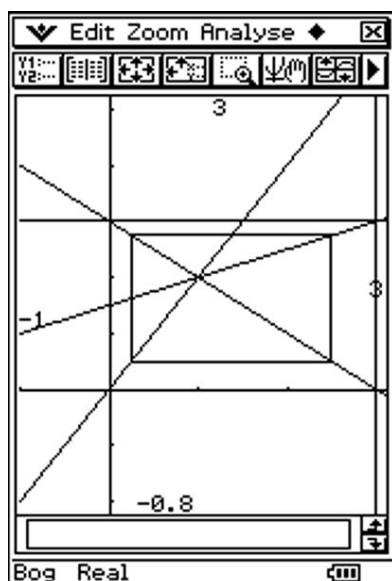


Abbildung 3

**Schüleraufgabe 2:** Berechnen Sie die linearen Funktionen, die die drei fehlenden Kanten beschreiben sowie die notwendigen Einschränkungen. Stellen Sie die Scheinnische grafisch auf Ihrem GTR dar.

**Schüleraufgabe 3:** Ermitteln Sie den Schnittpunkt der linearen Funktionen. Finden Sie den Zusammenhang zum Augpunkt A! Begründen Sie!

**Schüleraufgabe 4 (classPad):** Erstellen Sie eine eActivity, die zu zwei beliebigen Punkten  $P_1$  und  $P_2$  die lineare Funktion der Bildgeraden ermittelt.

Als oberer Abschluss der Nische kann auch ein parabelförmiger Bogen verwendet werden.

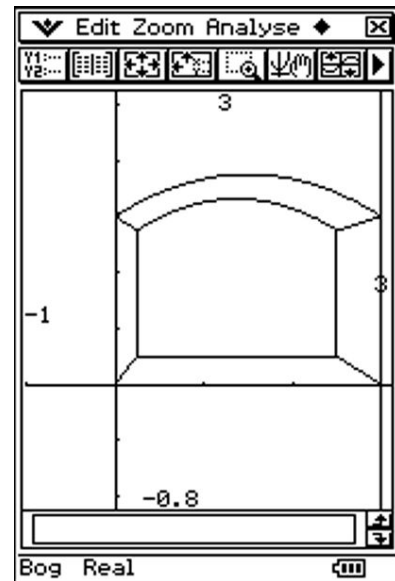


Abbildung 4

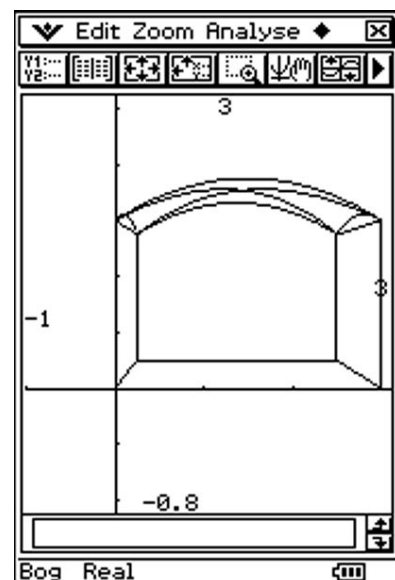


Abbildung 5

**Schüleraufgabe 5:** Ermitteln Sie die Gleichung einer Bildparabel zu  $y = -\frac{1}{3}x^2 + x + 3$  (vgl. Abb. 4).

**Schüleraufgabe 6:** Die Parabelbögen können auch senkrecht zur Abbildungsebene verlaufen oder diagonal als Kreuzgewölbe (vgl. Abb. 5). Ermitteln Sie je-



weils (näherungsweise) geeignete Parabelbögen und Gleichungen der Bildparabeln.

## Scheinkuppeln

Statt einer Nische soll jetzt eine virtuelle Kuppel entstehen. Die Herangehensweise ist prinzipiell die gleiche wie im vorherigen Abschnitt. Man wählt einen Augpunkt  $A$ , von dem aus jeder Punkt  $P$  der Scheinkuppel durch Schnitt der Verbindungsgeraden  $g_{AP}$  mit  $Z$  auf die Zeichenebene  $Z$  abgebildet wird. Als Beispiel wählen wir eine Kuppel  $K$  mit dem Radius  $r = 10$  Metern, die scheinbar auf der ebenfalls in 10 Meter Höhe befindlichen Decke erzeugt werden soll. Um die Rechnung zu vereinfachen, wählen wir das räumliche Koordinatensystem so, dass der Fußboden in der  $xy$ -Ebene liegt, der Kuppelmittelpunkt sich über dem Koordinatenursprung befindet und der Augpunkt im Fußboden liegt, etwa  $A(0/-15/0)$  und

$$K : x^2 + y^2 + (z - 10)^2 = 100.$$

Es ist nun nicht weiter schwierig, zu einem Punkt  $P \in K$  den Bildpunkt  $P' \in Z$  zu finden bzw. zu berechnen. Für eine allgemeinere Darstellung wäre es günstig, die „Breitenkreise“ bzw. die „Meridiane“ der Kuppel in die Zeichenebene  $Z$  abzubilden. Dazu führen wir eine Hilfsebene  $E_a$  mit  $E_a \parallel E_{xy}$  und der Höhe  $a$  ein. Für  $10 \leq a \leq 20$  schneidet  $E_a$  die Kuppel  $K$  in einem „Breitenkreis“: Es gilt

$$K : \left| \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 10 \end{pmatrix} \right| = 10$$

und  $E_a : z = a$  mit  $10 \leq a \leq 20$  und  $E_{10} = Z$ .  $E_a \cap K$  sei der Kreis  $k_a$  mit Radius

$$r' = \sqrt{100 - (a - 10)^2} = \sqrt{20a - a^2}.$$

In Parameterdarstellung gilt

$$k_a : \vec{x} = \begin{pmatrix} \sqrt{20a - a^2} \cdot \sin(t) \\ \sqrt{20a - a^2} \cdot \cos(t) \\ a \end{pmatrix},$$

d. h. jeder Punkt  $P$  auf der Kuppel ist durch einen Wert  $a \in [10, 20]$  und  $t \in [0, 2\pi]$  festgelegt und hat einen Bildpunkt  $P'$ , der durch die Projektion von  $A$  nach  $P$  entsteht.

Es gilt  $g(AP)$ :

$$\vec{x} = \begin{pmatrix} 0 \\ -15 \\ 0 \end{pmatrix} + s \begin{pmatrix} \sqrt{20a - a^2} \cdot \sin(t) \\ \sqrt{20a - a^2} \cdot \cos(t) \\ a \end{pmatrix}.$$

Für  $g(AP) \cap E_{10}$  gelten

$$\begin{aligned} x' &= \frac{10\sqrt{20a - a^2}}{a} \sin(t); \\ y' &= \frac{10(\sqrt{20a - a^2} \cos(t) + 15)}{a} - 15; \\ z' &= 10. \end{aligned}$$

Besser als mit der Höhe  $a$  lassen sich die Punkte in Polarkoordinaten beschreiben: Zu jeder Höhe  $a$  gehört ein „Breitenwinkel“

$$\varphi \in [0; \frac{\pi}{2}]$$

mit  $a = r \cdot \sin(\varphi) + 10$ . Da uns nur die Zeichenebene  $Z$  interessiert, betrachten wir nur die  $x$ - und die  $y$ -Koordinate:

$$\begin{aligned} x &= \frac{10\sqrt{1 - \sin^2(\varphi)}}{1 + \sin(\varphi)} \sin(t); \\ y &= \frac{10\sqrt{1 - \sin^2(\varphi)} \cdot \cos(t) + 15}{1 + \sin(\varphi)} - 15. \end{aligned}$$

**Schüleraufgabe 7:** Verifizieren Sie die Parameterdarstellung

Damit lässt sich mit jedem grafikfähigem Taschenrechner, der Kurven in Parameterdarstellung zeichnen kann, ein Bild der Scheinkuppel erzeugen. Für einen Breitenkreis setzen wir für  $\varphi$  einen beliebigen Winkel ein und erhalten die Parameterdarstellung

$$\begin{aligned} x(t) &= \frac{10 \cos(\varphi)}{1 + \sin(\varphi)} \sin(t); \\ y(t) &= \frac{10 \cos(\varphi) \cdot \cos(t) + 15}{1 + \sin(\varphi)} - 15 \end{aligned}$$

für  $0 \leq t \leq 2\pi$  und  $0 \leq \varphi \leq \frac{\pi}{2}$ .

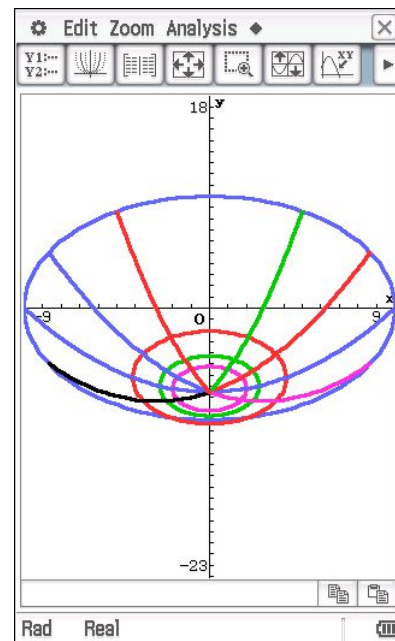


Abbildung 6

Um einen „Meridian“ bzw. einen „Viertelmeridian“ zu erhalten, tauschen die Parameter die Rollen:  $t$  erhält einen beliebigen aber festen Wert und  $\varphi$  wird zur Laufvariable. Da in Taschenrechnern im Allgemeinen  $t$  als

Laufvariable festgelegt ist, muss die Parameterdarstellung angepasst werden zu

$$x(t) = \frac{10 \cos(t)}{1 + \sin(t)} \sin(\alpha);$$

$$y(t) = \frac{10 \cos(t) \cdot \cos(\alpha) + 15}{1 + \sin(t)} - 15$$

für  $0 \leq t \leq \frac{\pi}{2}$  und  $0 \leq \alpha \leq 2\pi$ .

**Schüleraufgabe 8:** Die Scheinkuppel in der Wiener Jesuitenkirche hat einen Durchmesser von 7 Metern; die Decke des Mittelschiffs hat eine Höhe von 25 Metern, der Augpunkt hat einen horizontalen Abstand von 27 Metern vom Kuppelmittelpunkt. Geben Sie eine geeignete allgemeine Darstellung der Bildpunkte der Kuppel an!

### Lösungshinweise

Zur Raumvergrößerung:

- 1.)  $Q(0/2.5/2.75); R(0/2.5/0.5)$ .
- 2.)  $y = -x + 3$  für  $0 \leq x \leq 0.25$  und  $2.5 \leq x \leq 3$  bzw.  $y = 0.5x + 1.5$  für  $2.5 \leq x \leq 3$ .
- 3.)  $(0/1/2)$  ist die senkrechte Projektion des Augpunktes  $A$  auf  $Z$ .
- 4.) Beispiellösung:

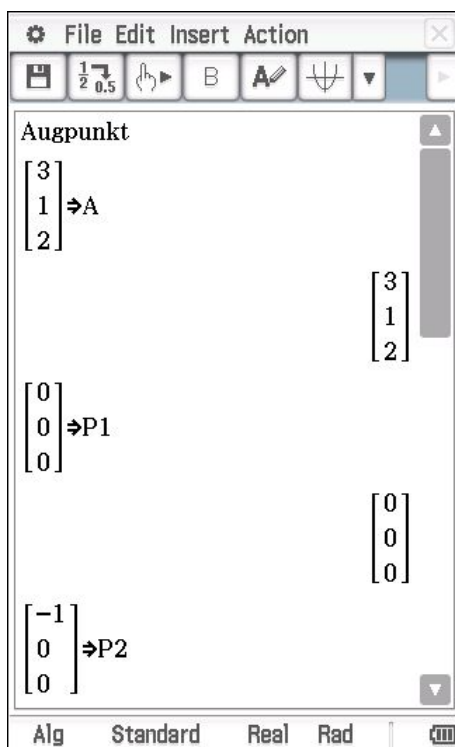


Abbildung 7

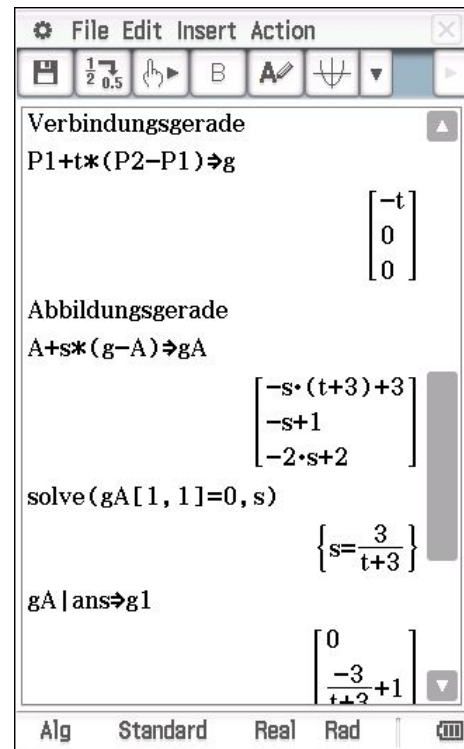


Abbildung 8

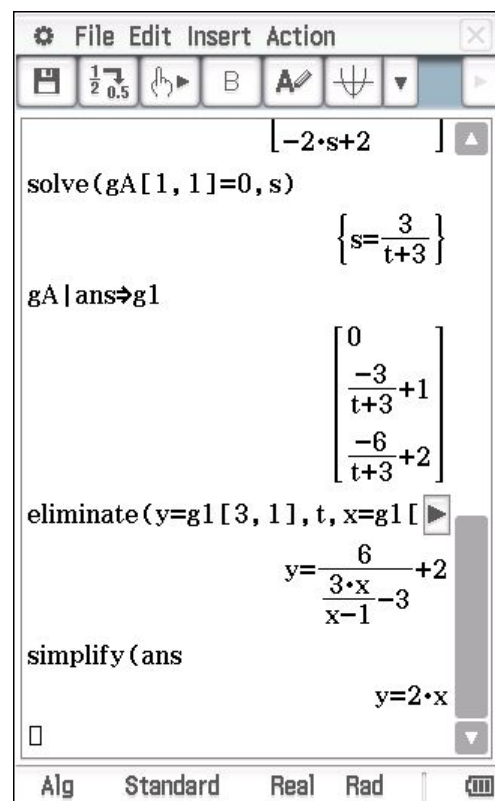


Abbildung 9

- 5.)  $y = -\frac{4}{9}x^2 + \frac{11}{9}x + \frac{89}{36}$ .
- 6.) Senkrecht zur Zeichenebene: z. B.

$$y = -x^2 - x + 3,$$

$$y' = \frac{-5.5x^2 + 34 - 46.5}{x - 1},$$

diagonal: z. B.

$$y \approx -0.32x^2 + x + 3.00,$$

$$y' \approx \frac{3.69x^2 - 5.49x - 30}{x - 10}.$$

7.) und 8.), zur Scheinkuppel siehe die beiden folgenden Abbildungen:

[illegible]

The screenshot shows the TI-Nspire CX CAS interface with the following content:

- Menu Bar:** Edit, Action, Interactive
- Toolbar:** Includes icons for undo, redo, undo redo, simplify, derivative, integral, and a dropdown menu.
- Input Area:**
  - Initial expression:  $\cos(t) \cdot (-a^2 + 50 \cdot a - 612.75)^{0.5}$
  - Command:  $A + s \cdot (P - A) \Rightarrow g$
  - Expression:  $\left[ \begin{array}{l} 0.5 \cdot s \cdot \sin(t) \cdot (-4 \cdot a^2 + 200 \cdot a - 2451)^{0.5} \\ s \cdot (0.5 \cdot \cos(t) \cdot (-4 \cdot a^2 + 200 \cdot a - 2451)^{0.5} + 27) - 27 \end{array} \right]$
  - Command:  $\text{expand}(g | s=25/a) \Rightarrow P1$
  - Expression:  $\left[ \begin{array}{l} \frac{12.5 \cdot \sin(t) \cdot (-4 \cdot a^2 + 200 \cdot a - 2451)^{0.5}}{a} \\ \frac{12.5 \cdot \cos(t) \cdot (-4 \cdot a^2 + 200 \cdot a - 2451)^{0.5}}{a} + \frac{675}{a} - 27 \end{array} \right]$
  - Command:  $P1 | a=25+.5 \cdot \sin(\varphi)$
  - Final Expression:  $\left[ \begin{array}{l} \frac{12.5 \cdot \sin(t) \cdot (-\sin(\varphi)^2 + 49)^{0.5}}{0.5 \cdot \sin(\varphi) + 25} \\ \frac{12.5 \cdot \cos(t) \cdot (-\sin(\varphi)^2 + 49)^{0.5}}{0.5 \cdot \sin(\varphi) + 25} + \frac{675}{0.5 \cdot \sin(\varphi) + 25} - 27 \end{array} \right]$
- Status Bar:** Alg, Decimal, Real, Rad

# Schnitte von Zylindern und Kegeln

J. Meyer  
(Hameln)

j.m.meyer@t-online.de



## Einführung

Wenn man einen Zylinder mit einer Ebene schneidet, bekommt man als Schnittkurve eine Ellipse, und wenn man den Zylinder abwickelt, wird aus der Ellipse eine Sinuskurve. Warum ist das so? Und was passiert, wenn man den Zylinder durch einen Kegel ersetzt?

## Schnitte von Zylindern

Was für eine Schnittkurve erhält man, wenn man einen Zylinder mit einer Ebene schneidet (Abb. 1)?

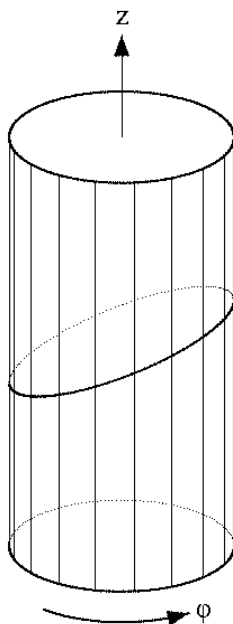


Abbildung 1: Zylinderschnitt

Der Zylinder habe die Gleichung

$$x^2 + y^2 = 1 \quad (z \text{ beliebig})$$

(die  $z$ -Achse ist also die Symmetrieachse des Zylinders), die Ebene habe die Gleichung  $z = mx$  mit  $m =$

$\tan \alpha$ . Im Zylindermantel sind die Geraden mit dem allgemeinen Punkt  $\begin{pmatrix} \cos \varphi \\ \sin \varphi \\ z \end{pmatrix}$  enthalten; ihr Schnitt mit der Ebene liefert den allgemeinen Punkt der Schnittkurve  $\begin{pmatrix} \cos \varphi \\ \sin \varphi \\ m \cdot \cos \varphi \end{pmatrix}$ . Was ist das für eine Kurve? Dazu ist es hilfreich, die Schnittebene und damit die Schnittkurve mithilfe der Matrix

$$M = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

in die Ebene mit  $z = 0$  hineinzudrehen (die  $y$ -Achse ist also Drehachse).  $M$  bildet den allgemeinen Punkt  $\begin{pmatrix} x \\ y \\ m \cdot x \end{pmatrix}$  der Schnittebene auf  $\begin{pmatrix} x / \cos \alpha \\ y \\ 0 \end{pmatrix}$  ab und den allgemeinen Punkt der Schnittkurve auf  $\begin{pmatrix} \cos \varphi / \cos \alpha \\ \sin \varphi \\ 0 \end{pmatrix}$ . Man erkennt, dass es sich um einen mit dem Faktor  $\frac{1}{\cos \alpha}$  gestreckten Kreis und damit um eine *Ellipse* handelt. Ist  $\alpha = 0^\circ$ , bekommt man als Schnittkurve natürlich einen ungestreckten Kreis. Schneidet man den Zylinder parallel zu seiner Symmetrieachse auf, bekommt man ein Rechteck; die Schnittkurve hat nun den allgemeinen Punkt  $\begin{pmatrix} \varphi \\ m \cdot \cos \varphi \end{pmatrix}$ , ist also eine *Sinuskurve* (und für  $m = 0$  eine Gerade).

## Schnitte von Kegeln

Nun liegt es nahe, die analoge Fragestellung für Kegel zu untersuchen. Die Kegelspitze sei  $Z = \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix}$ ; der Grundkreis habe den Radius  $r$  und den allgemeinen Punkt  $\begin{pmatrix} r \cdot \cos \varphi \\ r \cdot \sin \varphi \\ -h \end{pmatrix}$ . Die Gesamthöhe beträgt al-

so  $2 \cdot h$  (das hat den Grund, um später die Schnittebene möglichst einfach halten zu können). Der allgemeine Punkt der Mantelfläche ist durch

$$\begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} + \lambda \cdot \begin{pmatrix} r \cdot \cos \varphi \\ r \cdot \sin \varphi \\ -2 \cdot h \end{pmatrix}$$

mit  $\lambda \geq 0$  gegeben.

Schneidet man die Mantelfläche mit der Ebene zu  $z = m \cdot x$ , so ist

$$\lambda = \frac{h}{2 \cdot h + m \cdot r \cdot \cos \varphi};$$

der allgemeine Punkt der Schnittkurve (Abb. 2) ergibt sich als

$$\begin{aligned} P(\varphi) &:= \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} + \frac{h}{2h + mr \cos \varphi} \cdot \begin{pmatrix} r \cos \varphi \\ r \sin \varphi \\ -h \end{pmatrix} \\ &= \frac{hr}{2h + mr \cos \varphi} \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \\ m \cos \varphi \end{pmatrix}. \end{aligned}$$

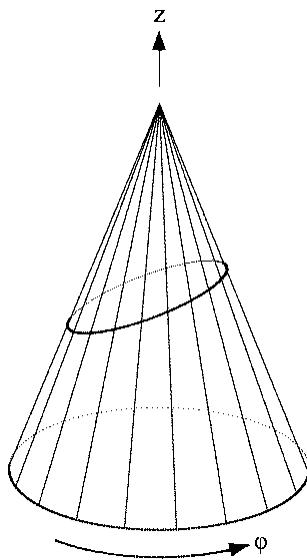


Abbildung 2: Kegelschnitt

Für  $m = 0$  bekommt man natürlich einen Kreis; im Folgenden sei also  $m > 0$ .

Wie beim Zylinder ist es auch hier hilfreich, Schnittebene und Schnittkurve mit der obigen Matrix  $M$  in die Ebene mit  $z = 0$  hineinzudrehen; das Resultat ist

$$Q(\varphi) := \frac{r}{2 \cos \alpha \left(1 + \frac{mr}{2h} \cos \varphi\right)} \cdot \begin{pmatrix} \cos \varphi \\ \cos \alpha \sin \varphi \\ 0 \end{pmatrix}$$

(hier ist ein CAS hilfreich).

Mit  $\varepsilon := \frac{m \cdot r}{2 \cdot h}$  ist  $Q(\varphi)$  bis auf Achsenstreckungen gegeben durch

$$\frac{1}{1 + \varepsilon \cos \varphi} \cdot \begin{pmatrix} \varepsilon \cos \varphi \\ \sin \varphi \\ 0 \end{pmatrix} =: \begin{pmatrix} \xi \\ \eta \\ 0 \end{pmatrix}.$$

Wegen  $\xi^2 + \varepsilon^2 \cdot \eta^2 = \frac{\varepsilon^2}{(1 + \varepsilon \cdot \cos \varphi)^2}$  und  $1 - \xi = \frac{1}{1 + \varepsilon \cdot \cos \varphi}$  ist  $\xi^2 + \varepsilon^2 \cdot \eta^2 = \varepsilon^2 \cdot (1 - \xi)^2$ , woraus

$$\xi^2 \cdot \left(\frac{1 - \varepsilon^2}{\varepsilon^2}\right) + 2 \cdot \xi + \eta^2 = 1$$

folgt. Für  $\varepsilon = 1$  (wenn also die Schnittebene zu einer Mantellinie parallel ist) hat man eine *Parabel*, für  $0 < \varepsilon < 1$  eine *Ellipse* und für  $\varepsilon > 1$  eine *Hyperbel*.

Nun möchte man den Kegel *abwickeln*; dies ist nur im Ellipsenfall sinnvoll, da im Hyperbelfall der Doppelkegel berücksichtigt werden müsste. Das Resultat ist ein Kreissektor mit dem Mittelpunkt  $Z$ , dem Radius  $s = \sqrt{4 \cdot h^2 + r^2}$  und dem Teilumfang  $2\pi r$ ; der zugehörige Mittelpunktswinkel ermittelt sich wegen  $\frac{\mu}{360^\circ} = \frac{2\pi r}{2\pi s}$  zu  $\mu = 360^\circ \cdot \frac{r}{s}$  (Abb. 3).

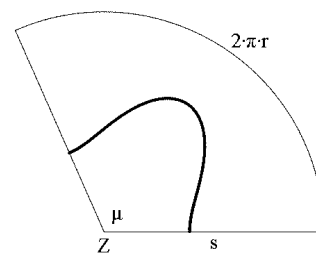


Abbildung 3: Abgewickelter Kegelschnitt

Man bildet die Punkte

$$P(\varphi) = \frac{hr}{2h + mr \cos \varphi} \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \\ m \cos \varphi \end{pmatrix}$$

der Schnittkurve ab auf den abgewinkelten Mantel; das Resultat heiße  $R(\varphi)$ . Dem Winkel  $\varphi$  entspricht der Mittelpunktswinkel  $\psi$  auf dem Kreissektor, den man wegen  $\frac{\varphi}{360^\circ} = \frac{\psi}{\mu}$  zu  $\psi = \frac{\mu}{360^\circ} \cdot \varphi$  bestimmt.

Der Abstand  $d$  von  $P(\varphi)$  zur Kegelspitze  $Z$  muss genauso groß sein wie der Abstand zwischen  $Z$  und  $R(\varphi)$ . Daher ist  $R(\varphi) = \begin{pmatrix} d \cdot \cos \psi \\ d \cdot \sin \psi \end{pmatrix}$  (hier kommt wieder ein CAS zum Tragen). Abb. 3 zeigt ein mögliches Resultat.

Natürlich kann man die Parameter  $m$ ,  $r$  und  $h$  beliebig variieren (unter Beachtung von  $\varepsilon = \frac{m \cdot r}{2 \cdot h} < 1$ ).



### SFB/TRR 195 Symbolic Tools in Mathematics and their Application (Part 1/5)

#### Number Theory group

Number theory constitutes one of the five core areas in the SFB and is represented through both mathematical and computer algebra problems. The problems are chosen to produce numerical evidence and examples for important problems as well as to force the development of tools for number theoretical computations.

Our two projects, by Claus Fieker (Kaiserslautern) and Gabriele Nebe (Aachen), are the explicit determination of solvable fields and the computation of unit groups of orders in division algebras. The necessary number theoretic infrastructure that will be developed here will be distributed through the new OSCAR system.

Fieker's project is the construction of solvable number fields with a given fixed soluble group. This explicit counting problem has a long tradition in number theory — statistical analysis of invariants of suitable families of fields led for example to the Cohen–Lenstra heuristic, but explicit lists of number fields (provided by Malle) were also crucial in Prasad and Yeung's recent classification of fake projective planes.

Number fields with a solvable Galois group are naturally obtained via a tower of relative abelian (or even cyclic fields). Relative abelian fields are parametrized by class field theory, which as a first step has already been developed in the new system.

The second number theoretical project by Nebe spans number theory as well as group theory: here the structure of certain infinite matrix groups that arise naturally from number theory, as the unit groups of divi-

sion algebras or more general  $S$ -arithmetic groups, will be investigated. The main role model here is Dirichlet's unit theorem that gives a precise structural description of the unit group of an order in an algebraic number field. For the non-commutative situation, where the field is generalised to a semisimple algebra and the ring of integers to a (maximal) order in this algebra, not much is known. There are theoretical results going back to the 1960s that unit groups of orders are finitely presented. Nebe aims to turn these theoretical results into practical algorithms and implementations that allow experimental studies of such unit groups (or more general  $S$ -unit groups). This project will find units and compute unit groups through methods based on a wide range of areas: non-commutative orders, lattices over number rings, cell complexes, group theory, and representation theory. As the ultimate results will be finite presentations for those unit groups, constructive group theory in **GAP** is one of the key tools, which is also going to be part of OSCAR.

Claus Fieker (Kaiserslautern)



### **Ruud Pellikaan, Xin-Wen Wu, Stanislav Bulygin, Relinde Jurrius** **Codes, Cryptology and Curves with Computer Algebra**

Cambridge University Press, Cambridge 2018, 597+ix Seiten, ISBN 978-0-521-52036-2, € 58,57

Wie der Name bereits andeutet, befasst sich dieses Buch mit Anwendungen der Computeralgebra in der Kodierungstheorie und der Kryptologie. Es ist ein fast 600 Seiten mächtiges und recht umfassendes Kompendium, wobei das Hauptaugenmerk sicherlich auf der Seite der Kodierungstheorie liegt und die Kryptologie mit weniger als 100 Seiten eher stiefmütterlich behandelt wird.

Etwa die erste Hälfte des Werks stellt eine klassische Einführung in die Kodierungstheorie dar, die z. B. sehr gut als Grundlage für eine Vorlesung zu diesem Thema dienen könnte. Neben mathematische sauberen Definitionen und Sätzen finden sich viele explizit durchgerechnete Beispiele und ein reichhaltiger Schatz an Übungsaufgaben. Alle grundlegenden Themen zu fehlerkorrigierenden Codes, Konstruktionen von Codes, Fehlerschranken und Gewichtszählerpolynomen werden eingehend behandelt. Viele wichtige Beispielklassen wie Hamming Codes, zyklische Codes, Reed-Muller Codes oder Rees-Salomon Codes werden sorgfältig studiert. Einen besonderen Mehrwert dieses Teils des Buches stellt das Kapitel zur algebraischen Dekodierung dar, welche vielerorts nur recht verwaschen präsentiert wird. Hier ist insbesondere die Komplexitätsanalyse verschiedener Aufgaben der Kodierungstheorie positiv hervorzuheben.

Im weiteren Verlauf wird der Rahmen der Thematik ausgedehnt. Neben Bezügen der Kodierungstheorie zu

klassischen Themen der diskreten Mathematik wie Matroiden oder Gittern finden sich eine kurze Einführung in die Kryptologie, ein Kapitel über Anwendungen der Gröbner-Basen in der Kodierungstheorie und der Kryptologie sowie ein Kapitel über die moderne Herangehensweise an die Kodierungstheorie mittels algebraischer Kurven und dem Satz von Riemann-Roch.

Der Band wird abgerundet durch ein Kapitel über die Realisierung verschiedener Algorithmen und Berechnungen in diversen Computeralgebrasystemen, wobei sich auch viele mithilfe von GAP oder MAGMA vorgerechnete Beispiele finden.

Gerade für algebraische oder algorithmisch orientierte Leser bietet der Band ein Fülle an interessantem und hochwertig präsentiertem Material, das insbesondere die oftmals existierende Lücke zwischen der Theorie und der konkreten Berechnung sehr gut überbrückt. Deswegen ist er gerade für Interessenten an der Kodierungstheorie und ihren Algorithmen sehr zu empfehlen. Wer sich tiefere Einblicke in die Kryptologie oder die Theorie und Algorithmik algebraischer Kurven erhofft, sei allerdings auf andere einführende Werke verwiesen. Für meine nächste Vorlesung zum Thema Kodierungstheorie wird das vorliegende Werk auf jeden Fall eine wärmstens empfohlene Lektüre darstellen.

Martin Kreuzer (Passau)

**Patrick Wegener: Hurwitz action in Coxeter groups and elliptic Weyl groups**

**Betreuer: Barbara Baumeister (Bielefeld)**

**Zweitgutachter: Jon McCammond (Santa Barbara)**

**Juli 2017**

**Abstract:** Gegeben eine Gruppe  $G$ , eine Erzeugermenge  $T \subseteq G$  von  $G$ , welche abgeschlossen unter Konjugation ist, und eine natürliche Zahl  $n > 1$ . So erhält man durch

$$(t_1, \dots, t_n) \mapsto (t_1, \dots, t_{i-1}, t_{i+1}, t_{i+1}^{-1} t_i t_{i+1}, t_{i+2}, \dots, t_n)$$

eine Wirkung der Artinschen Zopfgruppe  $\mathcal{B}_n$  auf  $T^n$ , genannt Hurwitzwirkung. Zwei Elemente aus  $T^n$  heißen Hurwitz-äquivalent, wenn sie in der selben Bahn unter dieser Wirkung liegen. Die Fragestellung ob zwei beliebige Elemente Hurwitz-äquivalent sind, ist dabei eng mit dem Wortproblem in endlich präsentierten Gruppen verbunden. Im Jahr 2005 zeigten Liberman und Teicher, dass diese Frage (genau wie das Wortproblem) im Allgemeinen nicht entscheidbar ist. Es ist leicht festzustellen, dass zwei Hurwitz-äquivalente Elemente jeweils ein Wort (in der Erzeugermenge  $T$ ) für ein und dasselbe Gruppenelement liefern. Daher kann man die Hurwitzwirkung auch als Wirkung auf den (reduzierten) Worten für ein bestimmtes Gruppenelement betrachten. Motiviert durch Anwendungen in der algebraischen Geometrie und der Darstellungstheorie von Algebren, wurde das Hurwitz-äquivalenz Problem für verschiedene Gruppen untersucht. Zum einen wurden (endliche, affine und beliebige) Coxetergruppen betrachtet. Zum anderen wurden elliptische Weylgruppen (natürliche Verallgemeinerungen von Coxetergruppen) betrachtet. Dabei zeigte sich, dass diese Wirkung transitiv ist, wenn sie als Wirkung auf den reduzierten Wörtern eines (quasi-)Coxeter Elementes betrachtet wird. Im Falle der endlichen Coxetergruppen wurden die Resultate für die Ausnahmetypen  $E_6, E_7, E_8, F_4, H_3$  und  $H_4$  (teilweise) mit Hilfe von GAP nachgewiesen.

**Benjamin Schröter: Matroidal subdivisions, Dressians and tropical Grassmannians**

**Betreuer: Michael Joswig (Berlin)**

**Zweitgutachter: Alex Fink (London), Hannah Markwig (Tübingen)**

**November 2017**

**Abstract:** In algebraic geometry a basic case of a moduli space are Grassmannians, which parametrize all linear subspaces of fixed dimension. In this thesis we study various aspects of their tropical equivalents, the tropical Grassmannians and Dressians. A matroid is the combinatorial abstraction of a linear space and a tropical linear space is a valuated version of those. Tropical linear spaces are dual to matroid subdivisions. Motivated by the concept of splits, the simplest case of a subdivision, a new class of matroids is introduced, which can be studied via techniques from polyhedral geometry. This class is very large as it strictly contains all paving matroids. The structural properties of these split matroids can be exploited to obtain new results in tropical geometry, especially on the rays of the tropical Grassmannians and the dimension of the Dressian. In particular, a relation between matroid realizability and certain tropical linear spaces is elaborated. The rays

of a Dressian correspond to facets of the secondary polytope of a hypersimplex. A special class of facets is obtained by a generalization of splits, called multi-splits or originally, in Herrmann's work,  $k$ -splits. We give an explicit combinatorial description of all multi-splits of the hypersimplex. These are in correspondence to nested matroids and, via the tropical Stiefel map, also to multi-splits of products of simplices. Hence, we derive a description for all multi-splits of a product of simplices. Moreover, a computational result leads to explicit lower bounds on the total number of facets of secondary polytopes of hypersimplices. Other computational aspects are also part of our research: A new method for computing tropical linear spaces and more general duals of polyhedral subdivisions is developed and implemented in the software polymake. This is based on Ganter's algorithm (1984) for finite closure systems. Additionally, we describe the implementation of a subfield of the field of formal Puiseux series. This is employed for solving linear programs and computing convex hulls depending on a real parameter. Moreover, this approach is useful for computations in convex and algebraic tropical geometry. Tropical varieties, as for example tropical linear spaces or tropical Grassmannians, are intersections of finitely many tropical hypersurfaces. The set of corresponding polynomials is a tropical basis. We give an explicit upper bound for the degree of a general tropical basis of a homogeneous polynomial ideal. Furthermore, various examples illustrate differences between Gröbner bases and tropical bases.

**Carlos Améndola Cerón: Algebraic Statistics of Gaussian Mixtures**

**Betreuer: Bernd Sturmfels (Leipzig), Christian Haase (Berlin)**

**Zweitgutachter: Reinhold Schneider (Berlin)**

**November 2017**

**Abstract:** We study the statistical models known as Gaussian mixtures from an algebraic point of view and we illustrate how algebraic techniques can be useful to address fundamental questions on the shape and inference of mixtures of Gaussian distributions. These include the maximum number of modes, algebraic complexity of the methods of maximum likelihood and moment matching, and the problem of identifiability.

**Merlin Mouafo Wouodjie: On the solutions of holonomic third-order linear irreducible differential equations in terms of hypergeometric functions**

**Betreuer: Wolfram Koepp (Kassel)**

**Zweitgutachter: Mark van Hoeij (Tallahassee), Mama Foupouagnigni (Limbe, Cameroon)**

**Februar 2018**

**Abstract:** Let  $k$  be an extension field of  $\mathbb{Q}$  which is algebraically closed and has characteristic zero, and  $k(x)[\partial]$  be the ring of differential operators with coefficients in  $k(x)$ . Let  $L \in k(x)[\partial]$  be an irreducible third-order linear differential operator without Liouvillian solutions. Let  $E = \{B_\nu^2, {}_1F_1^2, {}_0F_2, {}_1F_2, {}_2F_2\}$  where  $B_\nu$  is the Bessel function, and  ${}_pF_q$  with  $p \in \{0, 1, 2\}$ ,  $q \in \{1, 2\}$ , the generalized hypergeometric function. The goal of this thesis is to find a solution (if that exists) of  $L$  in terms of  $S \in E$ , change of variables,

algebraic operations and exponential integrals. That means to find a solution (if it exists) of the form

$$\exp\left(\int r dx\right) \left(r_0 S(f(x)) + r_1 (S(f(x)))' + r_2 (S(f(x)))''\right)$$

where  $r, r_0, r_1, r_2 \in k(x)$ , and  $f^2 \in k(x)$  when  $S = B_\nu^2$  or  $f \in k(x)$  when  $S \in E \setminus \{B_\nu^2\}$ . We have implemented in Maple five solvers for  ${}_1F_1^2$ ,  ${}_0F_2$ ,  ${}_1F_2$ ,  ${}_2F_2$  and two solvers for  $B_\nu^2$ : when  $f \in k(x)$ , and when  $f^2 \in k(x)$  but  $f \notin k(x)$ . We complete the work by providing explicit examples for each solver.

**André Wagner: Computer Vision and Computer Algebra**  
**Betreuer: Michael Joswig (Berlin)**  
**Zweitgutachter: Didier Henrion (Toulouse)**  
**November 2017**

**Abstract:** In multiview geometry, a field of computer vision, images of a three-dimensional scene are taken by several cameras from various perspectives. We dedicate ourselves to studying multiview geometry by means of computer algebra. Three-dimensional scene and camera parameter reconstruction is at the core of computer vision. This thesis obtains novel results about these two fundamental topics in cer-

tain cases if additional information about the original three-dimensional scene is available.

The multiview variety encodes the space of three-dimensional points seen through various views. We extend the knowledge about the multiview variety to two generalizations of it, the rigid multiview variety and the unlabeled multiview variety. These two varieties are inspired by specific applications in computer vision. They can be used to improve and speed-up the identification of unlabeled marker configurations. We give a set-theoretical description of the rigid multiview variety and design a triangulation algorithm for the unlabeled multiview variety. The 8-point algorithm is one of the most important algorithms in multiview geometry, and the most commonly used algorithm for fundamental matrix estimation. It is known that the unit cube defeats the 8-point algorithm. We extend this result to all combinatorial cubes. Two perspective projections of a combinatorial cube defeat the 8-point algorithm independent of the position of the cameras. In this case we describe a new algorithm that drastically improves the quality of reconstruction of the fundamental matrix compared to the 7- and 8-point algorithm.

Finally we determine subintersections with omitted single intersectands of the primary decomposition of the set-theoretic equations of the Veronese variety. As the Veronese variety is an binomial ideal, these can actually be described by combinatorial means.

## Workshop-Förderung der Fachgruppe:

Sie veranstalten einen Workshop zu einem Thema aus dem Bereich der Computeralgebra und könnten mit einer kleinen finanziellen Unterstützung den Workshop deutlich interessanter oder effektiver gestalten? Die Fachgruppe Computeralgebra möchte 2018 wieder einen Workshop mit bis zu 1000,- Euro unterstützen.

Anträge können bis **1. September 2018** mit einer kurzen Beschreibung des Workshops (ca. 1 DIN A4 Seite; kurze Beschreibung des Gebiets, Thema des Workshops, Zielgruppe, Budget-Planung) und einer Darstellung, inwiefern diese Förderung einen deutlich erkennbaren Beitrag zum Gelingen des Workshops und zur Nachwuchsförderung liefert, an den Sprecher der Fachgruppe gerichtet werden:

**kemper@ma.tum.de**, bitte 'Workshop-Förderung' im Betreff angeben. Eine Entscheidung über die Vergabe dieser Mittel wird den Antragstellern ca. Mitte September mitgeteilt. (Anträge, die nach Ende der Frist eintreffen, können ggf. für die Vergabe von Restmitteln berücksichtigt werden.)



**Christoph Koutschan (Linz):**

**Quod Erat Demonstrandum: Proofs by Computer**

Jury: James H. Davenport (Bath), Christian Krattenthaler (Wien), Marko Petkovsek (Ljubljana), Bruno Salvy (Lyon), Carsten Schneider (Linz), Nobuki Takayama (Kobe)

**Juni 2017**

**Abstract:**

In today's mathematics, the computer has become an indispensable tool, not only as a number cruncher, but also as a proof assistant. There are different approaches to constructing mathematical proofs with the computer: general theorem provers versus special-purpose algorithms that are designed for a particular type of problem or a particular class of objects. We follow the second approach. The objects we deal with are holonomic functions and sequences, a rather large class of mathematical functions, that are given as solutions of certain systems of linear differential equations or recurrences. The type of problem that we address is to prove identities among these objects, possibly involving integrals and sums.

The foundations of the proof theory for holonomic functions have been laid by Wilf and Zeilberger in the early 1990s, and it is heavily based on the method of creative telescoping. When applied to a parametrized integral, this method is also called "differentiating under the integral sign", and it allows, under certain assumptions, to derive a differential equation satisfied by the integral. The basic idea is the following: given  $F(x) = \int_a^b f(x, y) dy$ , one aims at finding a creative telescoping relation for the integrand of the form  $p_r(x)\partial_x^r f + \dots + p_0(x)f = \partial_y g$ , where the coefficients  $p_i$  are free of  $y$  and where  $g$  is an explicit function in terms of  $f$ . Integrating both sides of this equation yields the desired differential equation  $p_r(x)F^{(r)}(x) + \dots + p_0(x)F(x) = g(b) - g(a)$ . An analogous strategy can be used to derive recurrence equations for definite sums. Creative telescoping

can be executed algorithmically, i.e., by a computer, if the input is a holonomic function, for example. In a nutshell this means that it satisfies a maximally overdetermined system of partial differential equations.

Our contributions to this area are two-fold: on the one hand, we have developed new algorithmic approaches for constructing creative telescoping relations, which in many situations are more efficient than previously known methods. On the other hand, we have implemented a large collection of relevant algorithms in our Mathematica package HolonomicFunctions: noncommutative Gröbner basis computation in Ore algebras, rational solutions of linear systems of PDEs or recurrences, closure properties for D-finite functions, and several creative telescoping algorithms.

Finally, we present a selection of applications from various fields of mathematics that demonstrate the power and usefulness of our approach. For example, we can find (and prove) closed-form evaluations of determinants whose dimension is a symbolic parameter. Many counting problems in enumerative combinatorics can be stated in terms of such determinants. This way, we were able to prove the qTSPP conjecture, a long-standing conjecture concerning the q-enumeration of totally symmetric plane partitions. Another application is the verification of special function identities, as they are listed, for example, in the Digital Library of Mathematical Functions (DLMF). In the context of knot theory, holonomic methods turned out to be useful for studying the colored Jones polynomial, a powerful knot invariant that is known to satisfy a q-holonomic recurrence equation. In mathematical physics we have studied random walks in face-centered cubic lattices, recurrence equations for relativistic Coulomb integrals, and integrability conditions for homogeneous potentials of degree -1. Last but not least, we mention some collaborations with colleagues from numerical analysis, where we employed our software to derive recursive schemes for efficient implementations of finite element solvers.



### AIMS-Volkswagen Stiftung Workshop on Introduction to Computer Algebra and Applications

Douala, Kamerun, 06.10. – 13.10.2017

[www.aims-volkswagen-workshops.org](http://www.aims-volkswagen-workshops.org)

This workshop brought together experts from various different areas of computer algebra and over 80 researchers and students from 18 different countries in Africa. The main organisers of the workshop, Wolfram Koepf (Universität Kassel) and Mama Foupouagnigni (AIMS Cameroon), did an excellent job in running this workshop. The workshop was funded by the Volkswagen Foundation and was based on the AIMS (African Institute of Mathematical Sciences). The AIMS Cameroon is one of the currently 6 such centers for mathematical research and teaching distributed over Africa.

The core of this workshop were the 8 mini-courses in different topics of computer algebra combined with hands-on tutorial sessions on different computer algebra systems. These mini-courses were given by John Abbott (University of Genoa, Italy), Bruno Buchberger (RISC Linz, Austria), Bettina Eick (TU Braunschweig, Germany), Mama Foupouagnigni (AIMS Cameroon, Cameroon), Kenza Guenda (University of Algier, Algeria), Wolfram Koepf (Universität Kassel, Germany), Martin Kreuzer (Universität Passau, Germany), Georg Regensburger (Universität Linz, Austria) and they addressed the computer algebra systems ApCoCoA, CoCoA, GAP, Mathematica, Maxima and Sage. The workshop also contained a variety of interesting talks given by researchers and students attending the workshop.



*Hauptvortragende und Ehrengäste (v.l.): Hans Fotsing Tetsing, Kenza Guenda, Christian Kouam (Volkswagen AG Kamerun), Merlin Mouafo Wouodjie, John Anthony Abbott, Mama Foupouagnigni, Daniel Duviol Tcheutia, Hans-Dieter Stell (Botschafter in Kamerun), Evans Ocansey, Wolfram Koepf, Martin Kreuzer, Bettina Eick, Georg Regensburger, Bruno Buchberger*

Cameroon as a country is a highly interesting place to host such a workshop. It is in the middle of Africa and it displays a lot of African culture. This became prominent during the conference dinner which did not only feature excellent food, but also a very entertaining dancing session. The workshop also included a tourist visit to places in Douala and this, again, was a very interesting experience.

Bettina Eick (Braunschweig)



*Tagungsfoto der Teilnehmer des Workshops in Douala*

## Nikolauskonferenz 2017

Aachen, 08.12. – 09.12.2017

[www.math.rwth-aachen.de/Nikolaus2017/](http://www.math.rwth-aachen.de/Nikolaus2017/)

Die Nikolauskonferenz findet seit mehr als 25 Jahren jährlich in der Regel am Freitag und Samstag am oder unmittelbar nach dem 6. Dezember am Lehrstuhl D für Mathematik der RWTH Aachen statt (historische Anmerkungen zur Konferenz findet man auf [www.math.rwth-aachen.de/~Frank.Luebeck/misc.html](http://www.math.rwth-aachen.de/~Frank.Luebeck/misc.html)).

Thematische Schwerpunkte sind die Gruppen- und Darstellungstheorie und insbesondere rechnerische Aspekte in diesen Gebieten. Die Konferenz 2017 wurde von 43 externen Teilnehmerinnen und Teilnehmern besucht; hinzu kamen noch die Teilnehmerinnen und Teilnehmer aus Aachen. Das Programm begann am frühen Freitagnachmittag (8.12.) und endete am Samstagabend (9.12.).

Es wurden 16 jeweils 20-minütige Vorträge gehalten, in denen sowohl theoretische als auch rechnerische und algorithmische Fragestellungen und Forschungsergebnisse vorgestellt wurden. Alle Vorträge hatten Bezüge zur Gruppentheorie oder zur Darstellungstheorie von Gruppen und Algebren, wobei ein breites Themenspektrum abgedeckt wurde. Die Vortragsthemen entstammten zum Beispiel den Bereichen: computergestützte Gruppentheorie, geometrische Gruppentheorie, endlich präsentierte Gruppen, Coxetergruppen,  $W$ -Graphen, Darstellungen endlicher Gruppen, nilpotente Algebren, Differentialgeometrie und Kohomologie diskreter Gruppen.

Die Konferenz war hervorragend organisiert (Hauptorganisator: Frank Lübeck). Zusätzlich zu den Vorträgen boten die Kaffee- und Diskussionspausen sowie die gemeinsamen Abendessen am Freitag und Samstag Gelegenheit zum wissenschaftlichen Austausch.

Frank Himstedt (München)

## GDMV 2018 - Sektion Diskrete Mathematik und Computeralgebra

Paderborn, 05.03. – 09.03.2018

[www.gdmv2018.de](http://www.gdmv2018.de)

Die gemeinsame Jahrestagung der Gesellschaft für Didaktik der Mathematik und der Deutschen Mathematiker-Vereinigung fand dieses Jahr Anfang März in Paderborn statt. Das sehr umfangreiche Programm der Tagung beinhaltet auch eine Sektion zur Diskreten Mathematik und Computeralgebra, die von Alexander Pott (Magdeburg) und Florian Heß (Oldenburg) organisiert wurde.

In der Sektion wurden insgesamt vier 50-minütige Hauptvorträge zu den Themen Analoga von Designs und Codes von Alfred Wassermann (Bayreuth), dem Computeralgebrasystem OSCAR von William Hart (Kaiserslautern), zur Dekodierung binärer linearer Codes und Anwendungen in der Kryptographie von Alexander May (Bochum) sowie zu einfachen Semiringen und Postquantenkryptographie von Jens Zumbrägel (Passau) gehalten.

Darüberhinaus gab es 19 jeweils 20-minütige Vorträge mit einem diversen Spektrum an Themen von Kombinatorik und algorithmischer algebraischer Geometrie zur Entwicklung beziehungsweise Anwendung von Computeralgebrasystemen (Hecke und Maple) und über Open-Source Gröbner Basen Algorithmen bis hin zur optimierten Festlegung von Wahlbezirken.

Die Sektion war hervorragend in die GDMV Tagung und das weitere Angebot an Vorträgen, Kaffeepausen, Unterhaltungsprogramm am Mittwochnachmittag und Konferenzdinner am Donnerstagsabend integriert.

Florian Heß (Oldenburg)

## SYNASC 2017: International Symposium on Symbolic and Numeric Algorithms for Scientific Computing

Timișoara, Rumänien, 21.09. – 24.09.2017

[synasc.ro/2017](http://synasc.ro/2017)

SYNASC is a series of annual events in Timișoara, Romania, that aim to stimulate the interaction between the scientific communities of symbolic and numeric computing and to present interesting applications of the algorithms developed in the areas both in theory and in practice. The choice of the symposium topic was motivated by the belief of the organizers that the dialogue between the two communities is very necessary for accelerating the progress in making the computer a truly intelligent aid for mathematicians and engineers.

Started in 1999 as a workshop, SYNASC has established itself as an international forum for researchers and practitioners interested in symbolic and numeric computing. SYNASC is organized by the Department of Computer Science at West University of Timișoara in cooperation with the Research Institute for Symbolic Computation at Johannes Kepler University of Linz, Austria and with the Research Institute e-Austria in Timișoara. Having its unique venue in Timișoara, a historical and multicultural city, SYNASC has become a fixed meeting point where researchers from Romania and the rest of the world get together to present, to discuss, and to exchange their research results, new findings, and work in progress. It has helped promote scientific research and development in Romania and enhance the contacts between foreign and Romanian researchers, in addition to its significant contributions to international scientific exchange in computer science, with emphasis on computer algebra, symbolic computation and their possible relations to numerical computing.

The symposium is structured in six tracks: *Symbolic Computation*, *Numerical Computing*, *Logic and Programming*, *Artificial Intelligence*, *Distributed Computing* and *Advances in the Theory of Computation*. Each track has its own program subcommittee chaired by at least two experts who coordinate the paper reviewing process. In addition to these six tracks, several related workshops are organized each year. For insuring a high quality of the reviewing and of the paper selection process, the general program chair is chosen each year from the representative researchers in the international community. Over the years, the general program chairs of SYNASC have been: Bruno Buchberger, Ștefan Mărușter, Viorel Negru, Tudor Jebelean, Stephen Watt, Tetsuo Ida, Dongming Wang, Andrei Voronkov, Nikolaj Björner, Daniela Zaharie, Laura Kovacs, James Davenport, and for the 2018 edition Erika Abraham. The proceedings of the symposium are published by the IEEE Computer Society Press and contain selected papers from those presented at the event, after the improvements suggested by the programme committee.

The organizers of SYNASC have always tried to attract representative researchers in fields related to the main topics of the symposium. In 2017 the invitation has been honored by seven well-known scientists who gave remarkable plenary talks: Armin Biere, Bruno Buchberger, Panagiotă Fatourou, Tetsuo Ida, Gheorghe Păun, Klaus-Dieter Schewe, Dongming Wang, Erika Abraham and Wolfgang Windsteiger. As every year, an important number of invited speakers are well known researchers in Computeralgebra and related fields.

Tudor Jebelean (RISC-Linz)

---

## Hinweise auf Konferenzen

---

### Symmetry and Computations

Marseille, Frankreich, 03.04. – 07.04.2018

[conferences.cirm-math.fr/1772.html](http://conferences.cirm-math.fr/1772.html)

Symmetry appears as a desirable feature to preserve through numerical computations, as a property to take advantage of in efficiency considerations, or as an organizing principle for computations. In either cases, sophisticated schemes that require group-theoretic foundations have been developed, as the results of research at the frontier of pure mathematics, computer science and applied mathematics.

The topics will range across Geometric Integration, Symbolic Analysis, Computational Algebraic Geometry, Orthogonal Polynomials and Special Functions but with a focus on the exploitation of symmetry and group theoretic methods.

### ACA 2018

Santiago de Compostela, Spanien, 18.06. – 22.06.2018

[www.usc.es/regaca/aca2018](http://www.usc.es/regaca/aca2018)

The ACA 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.

### SC-square 2018

Oxford, Vereinigtes Königreich, 11.07.2018

[www.sc-square.org/CSA/workshop3.html](http://www.sc-square.org/CSA/workshop3.html)

Symbolic Computation is concerned with the algorithmic determination of exact solutions to complex mathematical problems; more recent developments in the area of Satisfiability Checking are starting to tackle similar problems but with different algorithmic and technological solutions. The two communities share many central interests, but researchers from these two communities rarely interact. Also, the lack of common or compatible interfaces of tools is an obstacle to their fruitful combination. Bridges between the communities in the form of common platforms and road-maps are necessary to initiate an exchange, and to support and direct their interaction. The aim of this workshop, along the SC-square H2020 FETOPEN Coordination and Support Activity project, is to provide a time to discuss, share knowledge and experience across both communities.

### ISSAC 2018

New York City, USA, 16.07. – 19.07.2018

[www.issac-conference.org/2018](http://www.issac-conference.org/2018)

The International Symposium on Symbolic and Algebraic Computation is the premier conference for research in symbolic computation and computer algebra. ISSAC 2018 will be the 43rd meeting in the series, which started in 1966 and has been held annually since 1981. The conference presents a range of invited speakers, tutorials, poster sessions, software demonstrations and vendor exhibits with a centerpiece of contributed research papers.

### ICMS 2018

South Bend (Indiana), USA, 24.07. – 27.07.2018

[www.icms-conference.org/2018](http://www.icms-conference.org/2018)

The meeting will provide researchers like yourself a forum for sharing challenges, achievements and progress in mathematical software research, design, development and use. We welcome work on any aspect of mathematical software in any area of mathematics, science and engineering, and applications.

### AEC 2018

Hagenberg, Österreich, 30.07. – 3.08.2018

[www.risc.jku.at/conferences/aec2018](http://www.risc.jku.at/conferences/aec2018)

Within the framework of the SFB “Algorithmic and Enumerative Combinatorics”, the summer school AEC 2018 will be held at the Research Institute for Symbolic Computation (Johannes Kepler University Linz) in Hagenberg. It is jointly organized with the algebra group at the Johannes Kepler University, the Johann Radon Institute for Computational and Applied Mathematics, and the combinatorics groups at the University of Vienna and the Vienna University of Technology.

The goal of this summer school is to put forward the interplay between the fields of Enumerative Combinatorics, Analytic Combinatorics, and Algorithmics. This is a very active research area, which, aside from the three fields fueling each other mutually, receives as well constant impetus from outside, by its interaction with algebra, probability, statistical physics, and computer science.

### CICM 2018

Hagenberg, Österreich, 13.08. – 17.08.2018

[www.cicm-conference.org/2018](http://www.cicm-conference.org/2018)

Digital and computational solutions are becoming the prevalent means for the generation, communication, processing, storage and curation of mathematical information. Separate communities have developed to investigate and build computer based systems for computer algebra, automated deduction, and mathematical publishing as well as novel user interfaces. While all of these systems excel in their own right, their integration can lead to synergies offering significant added value. The Conference on Intelligent Computer Mathematics (CICM) offers a venue for discussing and developing solutions to the great challenges posed by the integration of these diverse areas.

The conference is organized by Wolfgang Windsteiger, takes place at the RISC.

### CAP Days 2018

Siegen, 28.08. – 31.08.2018

[homalg-project.github.io/capdays-2018/](http://homalg-project.github.io/capdays-2018/)

CAP (Categories, Algorithms, Programming) is a software project for constructive category theory written in GAP. It facilitates both the realization of specific instances of categories and the implementation of generic categorical algorithms.

The workshop aims at mathematicians who want to learn about the CAP project, categorical programming, and structuring implementations in a categorical way. The workshop is suitable for both GAP newcomers and veteran GAP programmers.

## **AISC 2018**

Suzhou, China, 16.09. – 19.09.2018

[aisc2018.cc4cm.org](http://aisc2018.cc4cm.org)

The aim of the conference is to provide a forum for the exchange of ideas and the presentation of new tools and solutions. Another goal is to foster personal contacts among researchers from different fields related to AI and Symbolic Computation. The conference is concerned with all aspects of research, including theory, implementations and applications. Conferences in this series are usually held every two years. The previous five ones took place in Sevilla (Spain), Paris (France), Birmingham (United Kingdom), Beijing (China) and Linz (Austria). AISC 2018 will take place in Suzhou, China

## **CASC 2018**

Lille, Frankreich, 17.09. – 21.09.2018

[www.casc.cs.uni-bonn.de/2018](http://www.casc.cs.uni-bonn.de/2018)

The 20th International Workshop on Computer Algebra in Scientific Computing, CASC 2018, will be held in the city of Lille, France, September 17 - 21, 2018. The deadline for submission is April 15, 2018.

The topics addressed in the workshop cover all the basic areas of scientific computing as they benefit from the application of computer algebra methods and software.

## **SYNASC 2018**

Timișoara, Rumänien, 20.09. – 23.09.2018

[synasc.ro/2018](http://synasc.ro/2018)

International Symposium on Symbolic and Numeric Algorithms for Scientific Computing is an international conference that aims to stimulate the interaction between the two scientific communities of symbolic and numeric computing and to exhibit interesting applications of the areas both in theory and in practice. The choice of the topic is motivated by the belief of the organizers that the

dialogue between the two communities is very necessary for accelerating the progress in making the computer a truly intelligent aid for mathematicians and engineers.

## **Annual meeting SFB TRR 195**

Tübingen, 24.09. – 28.09.2018

[www.math.uni-tuebingen.de/arbeitsbereiche/geometrie/annual-meeting-sfb-trr-195-1](http://www.math.uni-tuebingen.de/arbeitsbereiche/geometrie/annual-meeting-sfb-trr-195-1)

The second annual meeting of the TRR 195 will take place in Tübingen, Sept 24-28, 2018. The meeting starts on Monday at 2pm and ends on Friday after lunch.

## **INFORMATIK 2018 — 48. Jahrestagung der Gesellschaft für Informatik**

26.09. - 27.09.2018

[informatik2018.gi.de](http://informatik2018.gi.de)

Die INFORMATIK 2018 ist die Jahrestagung der Gesellschaft für Informatik e.V. und findet am 26./27. September unter dem Motto „Zukunft der Arbeit – Zukunft der Informatik“ statt. Im Austausch mit Experten aus Wissenschaft, Wirtschaft und Politik werden folgende vier Fragestellungen erörtert:

- Wie wird sich Arbeit durch die Digitalisierung verändern, wie gestalten wir die Arbeitswelt der Zukunft und welche Anforderungen stellt das an die informatischen Systeme?
- Wie muss eine gute Bildung in der digital vernetzten Welt aussehen und wie muss sich das deutsche Bildungssystem verändern?
- Welche Herausforderung an Sicherheit, Schutz und Vertrauen bringen zunehmend digital vernetzte Arbeits- und Produktionsprozesse?
- Bedarf es einer neuen Ethik in der digitalen Welt, wie kann Regulierung aussehen und welche Rolle muss die Informatik dabei spielen?

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

Bitte zurücksenden an:

Prof. Dr. Wolfram Koepf  
Universität Kassel  
FB Mathematik/Informatik  
Heinrich-Plett-Str. 40  
D-34132 Kassel



Name:	Vorname:
Akadem. Grad:	Geburtsjahr:
<i>Privatanschrift:</i>	
Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
<i>Dienstanschrift:</i>	
Firma/Institut:	Abteilung:
Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
E-Mail:	
Gewünschte Postanschrift: <input type="checkbox"/> Privatanschrift <input type="checkbox"/> Dienstanschrift	
Gewünschte Regionalgruppenzuordnung: (http://regionalgruppen.gi.de)	

- ☐ 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 der ☐ GI ☐ DMV ☐ GAMM,

Mitgliedsnummer: .....

- ☐ 7,50 Euro. Ich beantrage gleichzeitig Mitgliedschaft in der ☐ GI ☐ DMV ☐ GAMM und bitte um Zusendung der dazu erforderlichen Unterlagen.

- ☐ 9,00 Euro für Nichtmitglieder. Ich bitte um Zusendung von Informationen über ☐ GI ☐ DMV ☐ GAMM.

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

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.

## 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,
- ☐ für von der GI ausgewählte Informationen mit Bezug zur Informatik, z.B. Weiterbildungsangebote, Informatikveranstaltungen oder -kongresse mit und ohne GI-Beteiligung sowie Publikationen mit Informatikbezug.

Wenn Sie uns Ihre E-Mail-Adresse angegeben haben, wird die 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, kurze Nachricht an [mitgliederservice@gi.de](mailto:mitgliederservice@gi.de), per Post oder Fax genügt.

Datum: ..... Unterschrift: .....

Rückfragen: Telefon +49 (0)228-302-151/-149 Telefax +49 (0)228-302-167 E-Mail: [mitgliederservice@gi.de](mailto:mitgliederservice@gi.de) <http://gi.de>



---

## Fachgruppenleitung Computeralgebra 2017–2020

---

**Sprecher:**

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

**Vertreterin der GI:**

Prof. Dr. Erika Ábrahám  
Fachgruppe Informatik  
RWTH Aachen University  
Ahornstr. 55, 52056 Aachen  
0241-80-21242, -22243 (Fax)  
[abraham@cs.rwth-aachen.de](mailto:abraham@cs.rwth-aachen.de)  
<https://ths.rwth-aachen.de/people/erika-abraham/>

**Fachreferent Sonderforschungsbereich 195:**

Prof. Dr. Meinolf Geck  
Universität Stuttgart  
Institut für Algebra und Zahlentheorie  
Pfaffenwaldring 57, 70569 Stuttgart  
0711 685-65367  
[meinolf.geck@mathematik.uni-stuttgart.de](mailto:meinolf.geck@mathematik.uni-stuttgart.de)  
<http://www.mathematik.uni-stuttgart.de/~geckmf/>

**Fachreferent Themen, Anwendungen und Publikationen:**

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

**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](mailto:koepf@mathematik.uni-kassel.de)  
<http://www.mathematik.uni-kassel.de/~koepf>

**Fachreferent Schule und Didaktik:**

StD Jan Hendrik Müller  
Rivius-Gymnasium der Stadt Attendorn  
Westwall 48, 57439 Attendorn  
02722-5953 (Sekretariat)  
[jan.mueller@math.uni-dortmund.de](mailto:jan.mueller@math.uni-dortmund.de)  
[www.mathebeimueeller.de](http://www.mathebeimueeller.de)

**Fachexperte Industrie:**

Prof. Dr. Christoph Thiel  
Fachbereich Campus Minden der FH Bielefeld  
Artilleriestr. 9, 32427 Minden  
0571-8385-258  
[christoph.thiel@fh-bielefeld.de](mailto:christoph.thiel@fh-bielefeld.de)  
<https://www.fh-bielefeld.de/minden/ueber-uns/personenverzeichnis/christoph-thiel>

**Stellvertretende Sprecherin:**

Prof. Dr. Anne Frühbis-Krüger  
Institut für Algebraische Geometrie  
Welfengarten 1, 30167 Hannover  
0511-762-3592  
[fruehbis-krueger@math.uni-hannover.de](mailto:fruehbis-krueger@math.uni-hannover.de)  
<http://www.iag.uni-hannover.de/~anne>

**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](mailto:fieker@mathematik.uni-kl.de)  
<http://www.mathematik.uni-kl.de/~fieker>

**Fachreferent Physik:**

Dr. Thomas Hahn  
Max-Planck-Institut für Physik  
Föhringer Ring 6, 80805 München  
089-32354-300, -304 (Fax)  
[hahn@feynarts.de](mailto:hahn@feynarts.de)  
<http://www2.mpp.mpg.de/members/hahn>

**Fachreferent CA an der Hochschule:**

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

**Fachreferent Themen, Anwendungen und Publikationen:**

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](mailto:martin.kreuzer@uni-passau.de)  
<http://www.fim.uni-passau.de/~kreuzer>

**Fachexperte Redaktion Rundbrief:**

Dr. Fabian Reimers  
Zentrum Mathematik – M11  
Technische Universität München  
Boltzmannstr. 3, 85748 Garching  
089-289-17474  
[reimers@ma.tum.de](mailto:reimers@ma.tum.de)  
<http://www-m11.ma.tum.de/reimers>

**Vertreterin der GAMM:**

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](mailto:eva.zerz@math.rwth-aachen.de)  
<http://www.math.rwth-aachen.de/~Eva.Zerz/>



# TI-Nspire™ macht Schule.

Der TI-Innovator™ Rover bringt Mathematik in Bewegung: Spielerisch lassen sich die Grundlagen der Programmierung erfahren.

Zur Steuerung verwenden Sie den TI-Innovator™ Hub und das TI-Nspire™ CX CAS Handheld.

Nehmen Sie Fahrt auf!



[education.ti.com/de/rover](http://education.ti.com/de/rover)