Oktober 2017

Computeralgebra Rundbrief

> Ausgabe 61

- 30 Jahre Fachgruppe Computeralgebra
- CoCoALib and CoCoA-5
- Neues aus Waterloo: Maple 2017
- Zur Symmetrie einiger Kurven

www.fachgruppe-computeralgebra.de

GI DMV GAMM

Computeralgebra-Rundbrief

Nr. 61 Oktober 2017

Fachgruppe Computeralgebra

Inhaltsverzeichnis

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)

Der Computeralgebra-Rundbrief erscheint halbjährlich, Redaktionsschluss 15.02. und 15.09. ISSN 0933-5994. Mitglieder der Fachgruppe Computeralgebra erhalten je ein Exemplar dieses Rundbriefs im Rahmen ihrer Mitgliedschaft. Fachgruppe Computeralgebra im Internet: http://www.fachgruppe-computeralgebra.de.

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

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

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

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

Mitteilungen der Sprecher

Liebe Mitglieder der Fachgruppe Computeralgebra,

genau 30 Jahre liegt die Gründung der Fachgruppe nun zurück. Das gibt uns einen Anlass zu feiern und zurückzublicken. Wer wäre berufener für diese Reminiszenz als Wolfram Koepf, der seit nunmehr 21 Jahren der Fachgruppenleitung angehört und neun Jahre davon der Sprecher der Fachgruppe war? So blickt er ab Seite 7 auf die Geschichte der Fachgruppe zurück und erinnert an einige der Highlights. Da wir diesem Artikel nichts vorwegnehmen wollen, wenden wir uns hier jetzt direkt dem Tagesgeschäft zu.

Im letzten Rundbrief haben wir bereits ausführlich über die Wahl der neuen Fachgruppenleitung und deren Zusammensetzung berichtet. Durch Veränderungen bei den Vertretern der Fachgesellschaften haben sich jedoch in der Zwischenzeit noch zwei weitere Änderungen ergeben: Herr Prof. Dr. Ernst Mayr, der neun Jahre lang der Fachgruppe als Vertreter der GI gedient hat, hat dieses Amt nun an Frau Prof. Dr. Erika Ábrahám (Aachen) übergeben, die zum 1. Juli von der GI hierfür benannt wurde. An dieser Stelle sei Herrn Mayr unser ganz besonderer Dank für seine tatkräftige Mitwirkung in der Fachgruppe ausgesprochen. Seinem Angebot folgend hat die Fachgruppe die ISSAC-Tagung 2010 nach München geholt, die er dann als Local Arrangements Chair mit großem Erfolg organisiert hat. Sein Rat und seine Expertise waren uns immer wertvoll, und das wird auch in Zukunft so bleiben. Ebenso begrüßen wir Frau Ábrahám herzlich in der Fachgruppenleitung und freuen uns sehr auf eine fruchtbare Zusammenarbeit und neue, von ihr ausgehende Impulse. Die zweite neue Entwicklung ist die Benennung von Frau Prof. Dr. Eva Zerz als Vertreterin der GAMM. Frau Zerz hier zu begrüßen erübrigt sich, da sie schon seit neun Jahren ein sehr aktives Mitglied der Fachgruppenleitung ist und eine Amtsperiode lang Sprecherin bzw. stellvertretende Sprecherin war. Da Frau Zerz damit nicht mehr als gewähltes Mitglied der Fachgruppenleitung angehören konnte, rückte für sie Herr Prof. Dr. Meinolf Geck (Stuttgart) nach. Diese Neuerungen führten dann auch zur Anpassung einiger Zuständigkeitsbereiche, deren Details Sie der Übersicht auf Seite 35 dieses Hefts entnehmen können.

Im Mai dieses Jahres fand wieder die 'Tagung der Fachgruppe' statt, über die wir in der Rubrik auf Seite 11 ausführlich berichten. An dieser Stelle sei nur erwähnt, dass uns die Tagung wieder einen merklichen Zuwachs an (vorwiegend jungen) Mitgliedern verschafft hat. Unsere einzige Aktivität in diesem Jahr war die Tagung allerdings nicht: Wir haben erneut auf der ISSAC, die in diesem Jahr in Kaiserslautern, also insbesondere in Deutschland, stattfand, Preise für das beste Poster und die beste Software-Demo gestiftet. Ein Bericht zur ISSAC findet sich auf Seite 31.

Erstmals hat die Fachgruppe im Rahmen der neu geschaffenen Workshop-Förderung eine Tagung unterstützt, einen Workshop über arithmetische Geometrie und Computeralgebra in Oldenburg. Ein Bericht findet sich auf Seite 30. Auch im kommenden Jahr besteht wieder die Möglichkeit einer Workshop-Förderung. Details hierzu finden Sie auf Seite 6.

Bei der gemeinsamen Jahrestagung der Deutschen Mathematiker-Vereinigung und der Gesellschaft für Didaktik der Mathematik im März 2018 in Paderborn wird unsere Fachgruppe mit einem Minisymposium zu Computeralgebrasystemen in der Hochschullehre vertreten sein, das wir in Zusammenarbeit mit dem Komptenzzentrum Hochschuldidaktik der Mathematik (khdm) organisieren. Näheres dazu gibt es ebenfalls auf Seite 12.

Nach so viel Information in eigener Sache darf natürlich die Computeralgebra selbst nicht zu kurz kommen, zu deren Aspekten wie immer interessante Artikel in den verschiedenen Rubriken auf Sie warten. Wir wünschen Ihnen eine angenehme und anregende Lektüre dieses Hefts.

Gregor Kemper *Anne Frühbis-Krüger*

Leserkommentar von J. Meyer zum Artikel ”Der goldene Schnitt und Polynome 4. Grades” (von A. Zitterbart) aus Rundbrief 60

In Zitterbart [2017] wird beschrieben, was ein Polynom vom Grad 4 mit dem Goldenen Schnitt zu tun hat. Ein anderer (und kürzerer) Weg findet sich in Meyer [2014]. Noch kürzer geht es so: Das Polynom hat (bis auf Verschiebungen längs der Achsen) die Gestalt

$$f(x) = x^4 - 6ax^2 + bx;$$

die Wendepunkte sind gegeben durch

$$x_{1,2} = \pm\sqrt{a} \quad \text{und} \quad y_{1,2} = -5a^2 \pm b\sqrt{a}.$$

Unabhängig von der Realität der Wendestellen hat die Gerade durch die Wendepunkte die Gleichung $y = bx - 5a^2$, hat also reelle Koeffizienten. Diese Gerade schneidet die Ausgangskurve an den Stellen $\pm\sqrt{a}$ und $\pm\sqrt{5a}$, woraus sich für positive Werte von a die Eigenschaft des Goldenen Schnittes sofort ergibt.

Literatur:

Meyer, J. [2014]: Über quartische Polynome mit zwei reellen Wendestellen. In: Der Mathematikunterricht 60 (1); S. 54-58.

Zitterbart, A. [2017]: Der goldene Schnitt und Polynome 4. Grades. In: Computeralgebra-Rundbrief 60, S. 22-26.

30 Jahre Fachgruppe Computeralgebra

Wolfram Koepf, Universität Kassel

koepf@mathematik.uni-kassel.de

Einführung

Wenn Sie diesen Computeralgebra-Rundbrief in den Händen halten, ist es genau 30 Jahre her, seit die Fachgruppe Computeralgebra gegründet wurde. Da ich die Fachgruppe 9 Jahre lang geleitet habe, bin ich auf der letzten Sitzung der Fachgruppenleitung gebeten worden, dieses Jubiläum hier im Rundbrief zu würdigen. Diesem Wunsch komme ich natürlich sehr gerne nach.

In diesem Artikel möchte ich noch einmal Revue passieren lassen, mit welchen Themen sich die Fachgruppe in den letzten 30 Jahren beschäftigt hat und wie sich unser Arbeitsgebiet Computeralgebra im Laufe der Zeit verändert hat. Damals eher noch eine Vision, hat sich die Computeralgebra inzwischen als eigenständiges Forschungsgebiet innerhalb von Mathematik und Informatik weltweit etabliert.

Gründung

Im ersten Computeralgebra-Rundbrief, der am 15. Dezember 1987 erschien und den wir – wie auch alle anderen Hefte – auf der Seite `http://www.fachgruppe-computeralgebra.de/rundbrief/` zum Download bereitgestellt haben, finden wir die folgenden Gründungsdetails.

> Liebe Kolleginnen und Kollegen,
>
> am 7. November wurde die Fachgruppe 2.2.1 Computer-Algebra der GI gegründet. Die Aufgabe dieser Fachgruppe wurde wie folgt definiert:
>
> Computer-Algebra befaßt sich mit Entwurf, Analyse und Implementierung von Algorithmen zum algebraischen und symbolischen Rechnen. Die Fachgruppe sieht es als ihre Aufgabe an, auf diesem Gebiet die Forschung und Entwicklung zu fördern sowie Informationen insbesondere über Softwaresysteme zu verbreiten.
>
> Die Fachgruppe wird gemeinsam von DMV, GAMM und GI getragen; die Federführung der Fachgruppe liegt bei der GI. Ihre Leitung besteht zunächst aus den Gründungsmitgliedern: Prof. Th. Beth, U. Karlsruhe, Prof. B. Buchberger, U. Linz, Prof. W. Degen, U. Stuttgart (Vertreter der GAMM), Prof. B. Fuchssteiner, U. Paderborn, Dr. R. Janßen, IBM Wissenschaftszentrum Heidelberg, Prof. R. Loos, U. Tübingen, Prof. B. H. Matzat, TU Berlin, Prof. J. Neubüser, RWTH Aachen (Vertreter der DMV), Dr. F. Schwarz, GMD St. Augustin (Vertreter der GI), Prof. H. Stoyan, U. Konstanz, Prof. V. Weispfennig, U. Passau, Prof. H. G. Zimmer, U. Saarbrücken. Als Sprecher wurde F. Schwarz gewählt, als Stellvertreter J. Neubüser.

Danach wird im ersten Heft die Struktur des Computeralgebra-Rundbriefs erläutert, die heute in modifizierter Form immer noch Gültigkeit hat.

Die Gründungsmitglieder waren daran interessiert zu erfahren, mit welchen Systemen die Mitglieder der Fachgruppe arbeiten und mit welchen Fragestellungen sie sich beschäftigen, und erstellten im Vorfeld der Gründung einen diesbezüglichen Fragenkatalog. Dessen Ergebnisse wurden im ersten Heft abgedruckt. Die Antworten der 360 ausgefüllten Fragebögen sind sehr aufschlussreich, zeigen sie uns doch heute, wie sehr sich die Details geändert haben: Die meisten der damaligen Betriebssysteme kennen wir heute gar nicht mehr, auch viele der damaligen Computeralgebrasysteme sind heute nicht mehr relevant. Lesen Sie selbst:

1. *Ich benutze die folgenden CA Pakete bzw. symbolischen Systeme:*

REDUCE	118	SAC-2/ALDES	18
μ-Math	68	Cayley	14
Macsyma	52	Scratchpad	9
Maple	34	SMP	6

Weitere Nennungen (5 oder weniger) folgen.

2. *Ich arbeite mit folgenden Rechnersystemen:*

VAX	93	Sun	15
IBM PC/AT/XT	69	Apple	13
IBM Mainframe	55	Atari	11
CDC	36	HP	11
Siemens	34	Symbolics	9
Cadmus	18	Unisys	7

Weitere Nennungen (5 oder weniger) folgen.

3. *Ich benutze Methoden der CA und symbolischen Manipulation für Anwendungen des folgenden Typs:*

Kombination mit Numerik	8
Algebraische Gleichungen und algebraische Geometrie	7
Erzeugung von Bew.-Gleichungen dyn. Systeme	6
Roboterkinematik	6
Kryptographie	5
Logikprogrammierung	5

Weitere Nennungen (4 oder weniger) folgen

4. *Dabei interessieren mich Methoden aus den folgenden Bereichen:*

Arithmetik f. Zahlen, Polynome, Potenzreihen	219
Lineare Algebra	162
Gruppentheorie	92
Termersetzungs-, Deduktionssysteme	112
Kombinatorik, Graphentheorie	110
Differentiation, Integration, Differentialgleichungen	180
Zahlentheorie	84
Geometrie	77
Mathematische Physik	113

Der erste Computeralgebra-Rundbrief, der seit der Gründung der Fachgruppe zweimal im Jahr erscheint, kam noch mit 7 Seiten aus, heute haben wir in der Regel ca. 30, manchmal sogar 40 Seiten pro Heft. Da der Computeralgebra-Rundbrief zunächst kostenlos verschickt wurde, steigt die Mitgliederzahl rasant an und liegt bereits nach einem Jahr im Oktober 1988 bei knapp 500 Mitgliedern. Was für eine Erfolgsgeschichte!

Dabei möchte ich daran erinnern, dass große Computeralgebrasysteme im Jahr der Fachgruppengründung auf IBM Personal Computern noch nicht genutzt werden konnten, da der maximal addressierbare Speicher nur 640 kByte betrug! Diese Schranke wurde erst im Jahr 1988 aufgeboben. Daher waren die ersten Computeralgebra-Nutzer entweder Nutzer kleiner Systeme oder sie hatten Zugriff auf einen Großrechner.

Damals war es auch noch sehr schwierig, Publikationen, in denen Ergebnisse, die mit Hilfe eines Computeralgebrasystems erzielt worden waren, eine Rolle spielten, positiv begutachtet zu bekommen. Die Gutachter schrieben häufig, dass die Ergebnisse ja nicht gesichert seien, wenn sie nicht von Hand berechnet oder verifiziert worden seien. Durch unser stetiges und unermüdliches Argumentieren hat sich dies bis heute grundlegend geändert: Inzwischen ist den meisten Gutachtern klar, dass die Ergebnisse komplexer Berechnungen eines Computeralgebrasystems wesentlich sicherer sind, als solch komplizierte Rechnungen per Hand durchzuführen.

Struktur der Fachgruppe

Die Fachgruppe Computeralgebra ist von Beginn an als eine Fachgruppe der GI konzipiert. Allerdings war den Gründern aufgrund der Interdisziplinarität der Computeralgebra wichtig, dass auch die DMV und die GAMM Trägerorganisationen der Fachgruppe sind. Seit die Fachgruppe ab Januar 1994 einen Mitgliedsbeitrag eingeführt hat, wurde die Organisation etwas komplizierter. Heute ist die GI alleinige Trägerorganisation der Fachgruppe Computeralgebra, die die Mitgliederdatenbank führt und den Mitgliedsbeitrag in Rechnung stellt. Unsere weitere Verbundenheit mit der DMV und der GAMM drückt sich unter anderem dadurch aus, dass beide Fachverbände – wie die GI – einen Vertreter in die Fachgruppenleitung entsenden, und dass die Fachgruppe sich immer wieder bei den Jahrestagungen der Gesellschaften engagiert. Zahlreiche Mitglieder der Fachgruppe sind zugleich Mitglieder der DMV oder der GAMM, wodurch der Jahresbeitrag – ebenso wie für ordentliche GI-Mitglieder – 7,50 € statt 9 € beträgt. Jedes Mitglied der Fachgruppe Computeralgebra wird bei der GI entweder als ordentliches GI-Mitglied oder als assoziiertes Mitglied geführt, s. `https://www.gi.de/mitgliedschaft.html`.

Die Fachgruppe hat eine Ordnung (s. `http://www.fachgruppe-computeralgebra.de/ordnung/`), welche im Januar 2014 unter der Leitung von Florian Heß aktualisiert wurde. Unsere Ordnung regelt u. a. die Zusammensetzung der Fachgruppenleitung. Die Leitung der Fachgruppe besteht aus

- 9 gewählten Mitglieder, welche alle 3 Jahre neu gewählt werden,
- je einem Vertreter von DMV, GAMM und GI, die von den Organisationen vorgeschlagen werden,
- und bis zu 3 weiteren Fachexperten, die von der Fachgruppenleitung gewählt werden.

Nach jeder Wahl bestimmt die Fachgruppenleitung aus ihrer Mitte einen Sprecher (eine Sprecherin) und einen stellvertretenden Sprecher (eine stellvertretende Sprecherin).

Die SprecherInnen im zeitlichen Verlauf der letzten 30 Jahre:

1987-1990	Fritz Schwarz, St. Augustin (Sprecher), Joachim Neubüser, Aachen (Stellv.)
1990-1993	Volker Weispfenning, Passau (Sprecher), Johannes Grabmeier, Heidelberg (Stellv.)
1993-1999	Johannes Grabmeier, Heidelberg (Sprecher), B. Heinrich Matzat, Heidelberg (Stellv.)

1999-2002	H. Michael Möller, Dortmund (Sprecher), Michael E. Pohst, Berlin (Stellv.)
2002-2005	Wolfram Koepf, Kassel (Sprecher), H. Michael Möller, Dortmund (Stellv.)
2005-2008	Wolfram Koepf, Kassel (Sprecher), Gerhard Hiss, Aachen (Stellv.)
2008-2011	Wolfram Koepf, Kassel (Sprecher), Elkedagmar Heinrich, Konstanz (Stellv.in)
2011-2014	Eva Zerz, Aachen (Sprecherin), Florian Heß, Oldenburg (Stellv.), Wechsel der Ämter nach der halben Periode
2014-2017	Florian Heß, Oldenburg (Sprecher), Gregor Kemper, München (Stellv.), Wechsel der Ämter nach der halben Periode
seit 2017	Gregor Kemper, München (Sprecher), Anne Frühbis-Krüger, Hannover (Stellv.in)

Highlights

Man kann schon mit Fug und Recht sagen, dass die Fachgruppe Computeralgebra zu jedem Zeitpunkt der letzten 30 Jahre sehr produktiv und erfolgreich gearbeitet hat. Dennoch gab es natürlich einige besondere Highlights, auf die wir uns hier konzentrieren wollen.

- Eine der ersten Aktivitäten der Fachgruppe war die Durchführung der internationalen Tagung ISSAC 1991 in St. Augustin bei Bonn. Diese Tagung war eine der ersten Tagungen der jährlichen Tagungsreihe ISSAC (International Symposium on Symbolic and Algebraic Computation), welche bis heute die führende Welttagung auf dem Gebiet der Computeralgebra ist. Auch später hat die Fachgruppe immer wieder ISSAC-Tagungen organisiert:
  - ISSAC 1991: 15.-17. Juli 1991, Bonn. Organisation: Wolfgang Lassner, Fritz Schwarz. Proceedings: `http://dl.acm.org/citation.cfm?id=120694`
  - ISSAC 1998: 13.-15. August 1998, Rostock. General Chair: Volker Weispfenning, lokale Leitung: Karl Hantzschmann. Proceedings: `http://dl.acm.org/citation.cfm?id=281508`
  - ISSAC 2010: 25.-28. Juli 2010, München. General Chair: Wolfram Koepf, lokale Leitung: Ernst W. Mayr. Proceedings: `http://dl.acm.org/citation.cfm?id=1837934`. Zu dieser Tagung wurde die Domain `http://www.issac-conference.org` eingerichtet. Das links abgedruckte Tagungsposter wurde vom Publicity Chair Peter Horn aus Fotos von ISSAC-Tagungen zusammengesetzt, die in Amerika, Europa und Asien stattgefunden hatten.

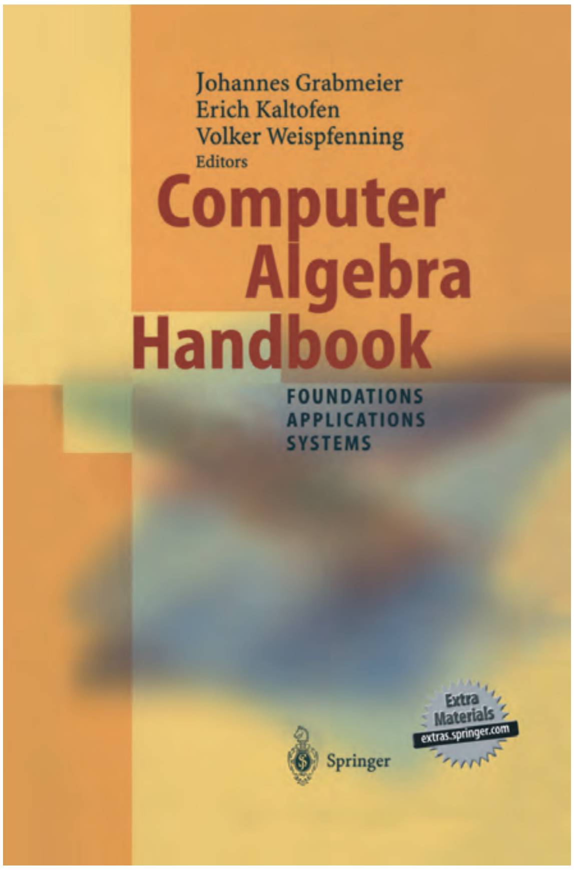
  Die folgenden internationalen Computeralgebra-Tagungen wurden von der Fachgruppe zwar nicht verantwortlich organisiert, hatten aber ebenfalls unsere Unterstützung und fanden in Deutschland statt:
  - EUROCAL 1987: 2.-5. Juni 1987, Leipzig. Organisation: Wolfgang Lassner. Proceedings: `https://link.springer.com/book/10.1007/3-540-51517-8`
  - ISSAC 2017: 25.-28. Juli 2017, Kaiserslautern. General Chair: Chee K. Yap, lokale Leitung: Wolfram Decker, Claus Fieker. Proceedings: `http://dl.acm.org/citation.cfm?id=3087604`
- Die Fachgruppe begann Anfang der 1990er Jahre, einen Report *Computeralgebra in Deutschland – Bestandsaufnahme, Möglichkeiten, Perspektiven* zu erstellen, der dann auch in Buchform erschien.

  Da es hierfür auch einen internationalen Markt gab, entstand aus diesem Projekt das Buch

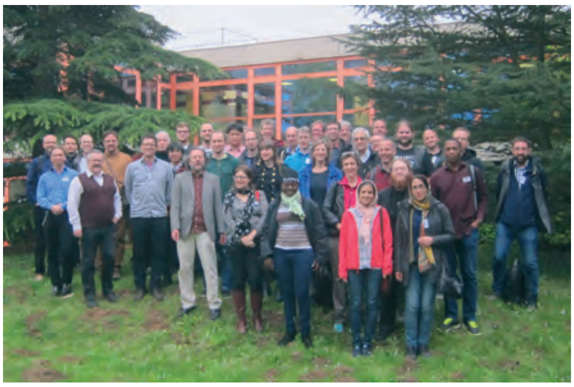
  J. Grabmeier, E. Kaltofen, V. Weispfenning (Eds.): *Computer Algebra Handbook*. Springer, 2003. `http://www.springer.com/de/book/9783540654667`

  das im Jahr 2003 veröffentlicht wurde.
- Zwischen 1998 und 2014 hat die Fachgruppe vier Tagungen zum Thema *Computeralgebra in Lehre, Ausbildung und Weiterbildung* organisiert, die 1998 und 2000 in Thurnau, 2002 im Bildungshaus Kloster Schöntal sowie 2004 im Tagungshaus Schönenberg

bei Ellwangen stattfanden, s. `http://www.fachgruppe-computeralgebra.de/konferenzen-claw/`. Zum selben Thema fanden zwischen 2006 und 2013 weitere 6 Tagungen statt, die gemeinsam mit dem GDM-Arbeitskreis Mathematikunterricht und Informatik (AKMuI) organisiert wurden.



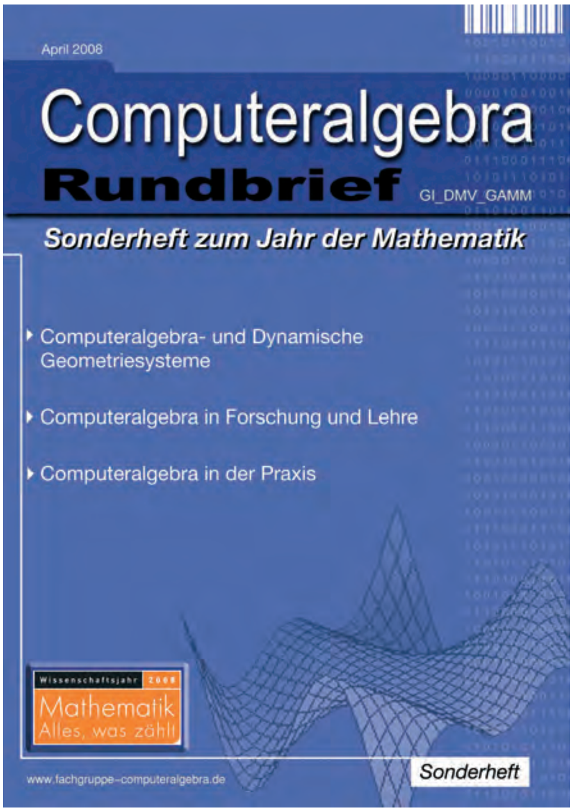
- Im Jahr der Mathematik 2008 erschien ein umfangreiches Sonderheft des Computeralgebra-Rundbriefs für die Zielgruppe Schüler und Lehrer, das an allen Gymnasien in Deutschland verteilt wurde. Dies war ein nicht ganz einfaches technisches, logistisches, organisatorisches und finanzielles Abenteuer! Aber alles klappte wirklich hervorragend.



Computeralgebra-Tagung 2017

- Die Fachgruppe organisiert und unterstützt Workshops, Seminare, Tagungen und andere Aktivitäten auf dem Gebiet der Computeralgebra. So wurde in den Jahren 2003, 2005, 2007, 2009, 2012, 2014 und 2017 jeweils eine dreitägige Computeralgebra-Tagung veranstaltet, bei der es neben eingeladenen Hauptvorträgen etablierter Wissenschaftler zahlreiche Doktorandenvorträge sowie mehrere Software-Präsentationen gab. Für den besten Vortrag eines Nachwuchswissenschaftlers wird regelmäßig ein Preis in Höhe von 500 € vergeben. Die nächste Computeralgebra-Tagung wird im Jahr 2019 stattfinden.

- Preise international: Auch bei der ISSAC-Tagungsreihe verleiht die Fachgruppe Preise für das beste Poster und die beste Software-Demonstration. Preise zu vergeben, ist aus mehreren Gründen eine sehr wertvolle Tätigkeit: Es bringt unser Fachgebiet voran, je mehr Preisträger es hervorbringt. Aber auch jeder einzelne Preisträger profitiert von einem Preis in seinem Lebenslauf. Bei Bewerbungen kann dies durchaus ein entscheidender Vorteil sein.



Ausblick

Wenn es die Fachgruppe Computeralgebra nicht gäbe, man müsste sie gründen! Die Fachgruppe Computeralgebra hat heute ungefähr 370 zahlende Mitglieder und ist damit die größte Computeralgebra-Vereinigung weltweit. In Deutschland gibt es nur wenige Fachgruppen innerhalb der Mathematik und Informatik, die so aktiv sind wie wir. Ich bin mir sicher, dass das große Engagement der Fachgruppe Computeralgebra entscheidend dazu beigetragen hat, unsere Fachgebiet zu dem zu machen, was es heute ist: Ein wichtiges, aktuelles Teilgebiet von Mathematik und Informatik, das diese beiden Disziplinen in hervorragender Weise verbindet.

Tagungen der Fachgruppe

Computeralgebra-Tagung der Fachgruppe

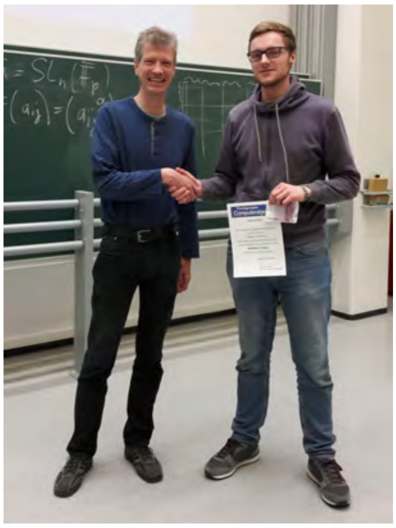
Kassel, 4. bis 6. Mai 2017

`www.fachgruppe-computeralgebra.de/tagung-kassel-2017`

In diesem Jahr fand wieder die Tagung der Fachgruppe Computeralgebra statt. Zum dritten Mal hintereinander war der Tagungsort Kassel, und die Leitung lag in den bewährten Händen von Wolfram Koepf. Ziel dieser Tagungsreihe ist es zum einen, Nachwuchswissenschaftlern die Möglichkeit zu geben, ihre Resultate vorzustellen, und zum anderen, durch einige Hauptvorträge Schlaglichter auf wichtige Gebiete der Computeralgebra zu werfen.

Insgesamt gab es 46 Teilnehmer, fünf Hauptvorträge, 18 eingereichte Vorträge und eine Software-Demo. Die Tagung begann mit einem Hauptvortrag von Daniel Robertz, der Eliminationsverfahren für Systeme von nicht-linearen partiellen Differentialgleichungen erklärte. Danach gab es eingereichte Vorträge in zwei Sektionen. Im zweiten Hauptvortrag berichtete Christian Eder, der Gewinner des Preises für den besten Nachwuchs-Vortrags der letzten Tagung, über aktuelle Softwareentwicklungen für Gröbnerbasen auf parallelen Rechnerarchitekturen. Im Hauptvortrag von Stephan Elsenhans ging es um Berechnungen zur Geometrie und Arithmetik algebraischer Flächen. Am Abschlusstag gab es noch zwei Hauptvorträge: Meinolf Geck berichtete über Rechnungen mit Strukturen aus der Lie-Theorie, und Alice Niemeyer führte in randomisierte Algorithmen in der Gruppentheorie ein.

Ebenso wie die Hauptvorträge trugen die eingereichten Vorträge zum Gelingen der Tagung bei, auch wenn sie hier nicht im Einzelnen genannt werden. Weil diese von durchweg hoher Qualität waren, hatten die Mitglieder der Fachgruppenleitung keine leichte Aufgabe, den Preisträger für den besten Vortrag eines Nachwuchswissenschaftlers zu ermitteln. Die Auswahl fand am Rande des gemeinsamen Abendessens in den Herkules Terrassen statt, das auch durch eine Polizeikontrolle des angemieteten Busses auf der Anfahrt nicht verhindert werden konnte.



Nachwuchspreis für Matthias Junge (rechts), mit Gregor Kemper (Sprecher der Fachgruppe)

Die Wahl fiel schließlich auf Matthias Junge (Oldenburg), der mit seinem Vortrag über schnelle Arithmetik in der Picardgruppe algebraischer Kurven überzeugte. Der Preis ist mit 500 Euro dotiert und beinhaltet die Einladung, auf der nächsten Tagung der Fachgruppe einen Hauptvortrag zu halten.

Die Tagung wurde getragen von der Fachgruppe mit finanzieller Unterstützung durch die Sponsoren Additive und Maplesoft. Zum ersten Mal war es diesmal möglich, einigen Teilnehmern eine Reisekostenbeihilfe zur Verfügung zu stellen. Die Fachgruppe dankt den Sponsoren, den Vortragenden und den weiteren Teilnehmern, ganz besonders aber Wolfram Koepf und seinem Team für die hervorragende Organisation der Tagung.

Gregor Kemper (München)

Tagungsfoto, Kassel 2017

Mini-Symposium: CAS in der Hochschullehre - Blick in die Praxis

GDMV 2018, Paderborn

www.gdmv2018.de/minisymposien

Der algorithmische Blickwinkel und der Einsatz vielfältiger moderner Computeralgebrasysteme hat in den letzten Jahren auch einer Reihe von Gebieten der Reinen Mathematik (wie z.B. der Algebra, Diskreten Mathematik, Geometrie und Zahlentheorie) einen experimentellen Zugang erschlossen, was die Forschung um etliche interessante Facetten bereichert und in Rückkopplung mit der Entwicklung von CAS inzwischen eine eigene Dynamik gewonnen hat. Zu nennen sind in diesem Zusammenhang das DFG-Schwerpunktprogramm 1489 "Algorithmische und experimentelle Methoden in Algebra, Geometrie und Zahlentheorie" (2010-2016), das DFG-Graduiertenkolleg 1632 "Experimentelle und konstruktive Algebra" (2010-2019) sowie der neu eingerichtete DFG Sonderforschungsbereich SFB/TR 195 "Symbolic Tools in Mathematics and their Application". Gerade in einem sich schnell entwickelnden Gebiet wie diesem ist es aber umso wichtiger, die heranwachsende Generation von Forschern nicht nur in der nötigen Tiefe mit der Theorie vertraut zu machen, sondern zusätzlich mit den entsprechenden Werkzeugen und experimentellen Techniken.

Auch in der Lehrerbildung kann die Auseinandersetzung mit CAS einen wichtigen Beitrag leisten. Neben den angesprochenen neuen Perspektiven besteht dort auch ein Mehrwert in der erworbenen Erfahrung im Umgang mit digitalen Werkzeugen, die in der Schule in den kommenden Jahrzehnten sicher ebenfalls eine Rolle spielen werden.

Im Gegensatz zum Schulkontext, wo der Einsatz von CAS im Unterricht bis heute kontrovers diskutiert wird und wo – wenn überhaupt – nur Systeme mit einer breiten Grundfunktionalität eingesetzt werden, gibt es in der Hochschullehre deutlich vielfältigere Einsatzszenarien und eine ganze Bandbreite von Systemen, die von sehr allgemeiner Funktionalität bis hin zu hoch spezialisierten CAS reicht. Trotz oder gerade wegen der Vielfalt lebt dieses Feld der universitären Lehre bis heute eher von ad hoc geschaffenen lokalen Vorgehensweisen; erfolgreiche Beispiele verbreiten sich oft durch zufällige Mund-zu-Mund Propaganda, eine systematische hochschuldidaktische Aufarbeitung dieses Feldes steckt noch in den Kinderschuhen.

Angesichts der Vielfalt der Einsatzszenarien von CAS in der Hochschullehre haben wir uns für 4 Themenschwerpunkte entschieden, bei denen wir jeweils versuchen mehr als einen Aspekt und mehr als ein System abzudecken:

- CAS als Backend – Brückenkurse und mehr
- CAS in der Studieneingangsphase Mathematik
- CAS in der Lehrerbildung
- CAS für mittlere Semester

Für dieses Minisymposium an der Schnittstelle zwischen Mathematik und Didaktik bietet gerade die gemeinsame Jahrestagung von DMV und GDM eine bestens geeignete Platform. Wir, d.h. die Organisatoren aus Fachgruppe Computeralgebra und khdm, hoffen auf einen fruchtbaren Erfahrungsaustausch und angeregte Diskussionen und würden uns freuen, wenn viele Fachgruppenmitglieder zumindest zu dem ein oder anderen Themenschwerpunkt vorbeischauen würden.

Anne Frühbis-Krüger (Hannover)

CoCoALib and CoCoA-5

J. Abbott, Universität Kassel
A. M. Bigatti, Università degli Studi di Genova

abbott@dima.unige.it
bigatti@dima.unige.it


Introduction

The CoCoA project dates back to 1987, with the first public release of its interactive system in 1989. The aim has always been to provide a user-friendly software laboratory for studying Computational Commutative Algebra, specifically ideals of multivariate polynomials (*e.g.* Gröbner bases).

Starting in the early 2000s the CoCoA software has undergone a profound change in its internal design: its "mathematical expertise" resides in *CoCoALib*, a C++ software library [5]; there is also an interactive system *CoCoA-5* [9] which uses an interpreter to grant easy access to CoCoALib's capabilities. All code is free and open source (licence: GPL-3).

We give an overview of the latest developments in the library and system: specifically for the recent release of CoCoA-5.2.0/CoCoALib-0.99550 (May 2017), and the upcoming release (autumn 2017).

The maturity of the software means that the overall design of CoCoALib and CoCoA-5 is now quite stable, so the main changes are the addition of numerous new functions. There are, nevertheless, a few handy changes in the user experience: *e.g.* verbose mode, interrupt handling, a time-out mechanism, and a procedure for "graceful obsolescence".

Verbose mode

There are several circumstances where it can be useful to know what is happening inside a function call: *e.g.* to understand how the computation is progressing (and whether it will likely finish in a reasonable time), for developers to check that intermediate results are correct, and even for educational purposes.

The new function `SetVerbosityLevel(N)` sets the global verbosity level to `N`. Various functions defined in CoCoALib and in the CoCoA-5 packages will print out some internal progress messages when the global verbosity level is higher than a certain threshold value: higher levels trigger greater verbosity. For instance, the lowest level giving information on the progress of Gröbner bases is 100: every time a new basis polynomial is found, an informative line is printed:

```
/**/ SetVerbosityLevel(100);
/**/ GBasis(ideal(x^4-z^2, x^3-y));
myDoGBasis: New poly: len(GB) = 1  len(pairs) = 1
myDoGBasis: New poly: len(GB) = 2  len(pairs) = 1
...
myDoGBasis: New poly: len(GB) = 5  len(pairs) = 2
```

Besides `GBasis`, verbose information can be produced by numerous other functions, and can also be activated within user defined functions.

Since Gröbner basis computation is quite pervasive in CoCoA its verbosity threshold level is quite high, whereas in general other functions have levels in the range 10–99. The values 1-9 are effectively reserved for user defined functions, so that they may be used without triggering any verbosity from CoCoA internal functions.

The number of functions which respond to the verbosity setting is steadily increasing — details are in the documentation (*e.g.* in CoCoA-5 type "`?verbose`").

Interrupt handling

In some hard Gröbner basis computations, by setting the verbosity level, one may see that the number of pairs yet to be processed is getting unfeasibly high. The user may then choose to interrupt the computation by typing `Ctrl-C`: the computation will be interrupted as soon as the reduction of the current S-polynomial terminates.

This interruption cancels the incomplete Gröbner basis computation, and returns the computer to the state it was in just before the Gröbner basis computation was begun. This allows the user to continue using the CoCoA-5 session safely, *e.g.* using values computed and stored before the interrupted function call. In other words: there is no need to fall back on the popular technique "turn it off and on again" ;-)

Implementing this correctly is not trivial: thanks to its clean, exception-safe design, CoCoALib guarantees that no memory has been lost, and no data has been modified.

All code written in the CoCoA-5 language is intrinsically interruptible thanks to the design of the interpreter. Those who write C++ and use CoCoALib directly have a similarly comfortable situation. The only point to note when writing C++ code is that a normal function in C++ "ignores" interrupts, so to make it interruptible there has to be an explicit check for a pending interrupt signal. The design for inserting such a check in potentially critical loops in C++ code using CoCoALib is quite simple: it is enough to add a line like this

```
  CheckForInterrupt("myFunctionName");
```

this function checks if the "interrupt flag" has been set, and if so, throws an exception. Thus it may be used in any function in CoCoALib (even those defined by users of CoCoALib).

A small aside: one might like to interrupt a computation, perform some checks and then decide whether to continue the computation from that stage. This option requires keeping in memory a (possibly huge) partial computation for further access, and is not (yet) implemented.

Time-Outs

Gröbner basis computations are notorious for being unpredictably lengthy: two seemingly similar inputs can lead to wildly varying computation times. A new feature in CoCoALib (and thus also in CoCoA-5) is the ability to set a *time-out*, an upper limit for the amount of time allowed to complete a computation: if it takes longer than the specified limit then a *time-out error* is generated.

The time-out mechanism is already available in an interim release, and will be in the next stable release CoCoA-5.2.2/CoCoALib-0.99560. The design allows time-outs to be applied easily to almost any C++ function using CoCoALib.

One application of time-limited Gröbner basis computations is in SMT solving with polynomial equalities. In this context, speed of computation is important; if a Gröbner basis can be obtained *quickly* then it can be used to assist future decisions and computations (or perhaps to prove unsatisfiability).

To illustrate the ease of using the time-out feature, here is the CoCoALib code for a time-limited Gröbner basis computation (based on the time-unlimited function `GBasis`). This function will either return the Gröbner basis or throw a `TimeoutException` if the computation is taking longer than `MaxTime` seconds.

```
auto GBasisTimeout(const ideal& I, double MaxTime)
{
  CpuTimeLimit timeout(MaxTime);
  return GBasis(I);
}
```

Graceful Obsolescence

As software develops, sometimes a few old functions become obsolete because a better design has evolved: *e.g.* perhaps a new more general function, or a more expressive syntax. For some users this "progress" is a deterrent to updating to the latest software release, for fear that their programs would stop working.

In CoCoA-5 and CoCoALib we have opted to first declare such functions *obsolescent*, *i.e.* they will become *obsolete* in a few years time. Such functions continue to work, but produce helpful warning messages about how the call should be updated. Here is an example in CoCoA-5. Imagine that a user runs these lines from an old file

```
/**/ Weights := matrix([[1,2,3,4]]);
/**/ M := CompleteToOrd(Weights);
```

The variable `M` will be assigned the expected result, but a warning message is also printed:

```
--> WARNING: "CompleteToOrd" is obsolescent;
--> WARNING: use "MakeTermOrd" instead.
```

We use this approach to allow users time to update their programs, and to provide useful guidance on how to do this. The actual implementation of the obsolescent functions resides in the special files: `obsolescent.cpkg5` for CoCoA-5 functions, and `obsolescent.[HC]` for CoCoALib functions.

News specific for CoCoALib

For simplicity we describe the new CoCoALib functions below in the following section on CoCoA-5, since most of the new functions added to CoCoA-5 are actually part of CoCoALib.

Special effort has always been invested in making CoCoALib clean and portable. One new feature specific to CoCoALib is the improved, more portable *configure-build-install* procedure. This makes it easier to install CoCoALib on a wider range of computers. The C++ source code for CoCoALib has also been made cleaner and more portable.

CoCoALib is currently written using standard C++03; this does mean that compilation with a C++11 compiler may produce some harmless warnings about "deprecated" features (esp. `auto_ptr`). Now that C++ compilers which can handle the newer versions of C++ are widely available, we shall soon make a proper update to CoCoALib to use the newer C++ standards.

New functions in CoCoA-5

Here we mention some of the more significant mathematical additions. The website download page for CoCoA-5 includes a complete list of all the new functions and features; the same list can also obtained by running the function `RelNotes()` from inside CoCoA-5. Furthermore, a fully detailed description of the work and decisions made for each software release may be found on the CoCoA *Redmine* website `https://cocoa.dima.unige.it/redmine`.

Most of the new functions added to CoCoA-5 have actually been added to CoCoALib, and then made available to CoCoA-5; the remaining few will be ported shortly.

- several efficient operations specifically for zero-dimensional ideals, mostly based on an efficient algorithm for computing minimal polynomials in zero-dimensional affine algebras (see *Minimal polynomials* below)
- more operations with algebraic extensions (even without a primitive element)
- faster conversion from a `string` to a `RingElem` — this is now the best way to read a large polynomial with very many terms.

Sectional matrix

The *sectional matrix* of an ideal I was introduced in [14] unifying the concepts of the Hilbert function of a homogeneous ideal I (along the rows) and its hyperplane sections (along the columns), and has recently been revived in [15]. Its theory and computation for a homogeneous ideal I is related to its `rgin`, (DegRevLex-gin).

```
/**/ use P ::= QQ[w,x,y,z];
/**/ I := ideal(x^4 -x*y^3,  x^2*(y -z),
                y^2*(y -z),  x*z^2 -y^3);
/**/ rgin(I);
ideal(w^3,  w^2*x,  w*x^2,  x^4,  x^3*y,  w^2*y^2,
      w*x*y^3,  x^2*y^3)
/**/ PrintSectionalMatrix(P/I);
  0   1   2   3   4   5   6
  -   -   -   -   -   -   -
  1   1   1   0   0   0   0
  1   2   3   1   0   0   0
  1   3   6   7   5   3   3
  1   4  10  17  22  25  28
```

The computation of `gin` and `rgin` in CoCoA, via a change of coordinates with random coefficients from a wide range, $[-10^6, 10^6]$, is based on the implementation of `RingTwinFloat` (see [3]).

Ideal of projective points

The efficient C code which was originally written for CoCoA-4 (see [7]) has now been made accessible from CoCoALib and CoCoA-5:

```
/**/ use P ::= QQ[x,y,z];
/**/ I := IdealOfProjectivePoints(P,
            matrix([[0,0,1],[1/2,1,1],[0,1,0]]));
/**/ I;
ideal(x*z +(-1/2)*y,  x*y +(-1/2)*y,
      x^2 +(-1/4)*y,  y^2*z -y)
```

Resolutions and graded Betti numbers

The current implementation for computing resolutions is rather naive, but will soon be improved — in this case porting the efficient but old C code for CoCoA-4 is impractical.

```
/**/ use R ::= QQ[x,y,z];
/**/ I := ideal(x^2+z^2,  x^2-y^2,  x*y-z^2);
/**/ resI := res(I);  PrintRes(resI);
0 --> R[-5] --> R[-3](+)R[-4]^2 --> R[-2]^3
```

A new addition is the ability to compute multigraded resolutions. There are functions to extract the Betti numbers in several formats (namely `BettiNumbers`, `BettiMatrix`, `BettiDiagram`, and functions to print them nicely: type `?betti` to see their manual pages)

```
/**/ -- first make a poly-ring with ZZ^2-grading
/**/ M := MakeTermOrd(matrix([[5,5,5,1,1],
                              [1,1,1,0,0] ]));
/**/ P := NewPolyRing(QQ, "t1,t2,t3,x,y", M, 2);
/**/ use P;
/**/ I := ideal(t1^6 -t3^6,  t1*t3*x^8 -t2^2*y^8);
/**/ resPI := res(P/I);  PrintRes(resPI);
0 --> R[-48, -8] --> R[-18, -2](+)R[-30, -6] --> R

/**/ PrintBettiNumbers(resPI);
----- 1 -----------
----- 2 -----------
  [18,  2]: 1
  [30,  6]: 1
----- 3 -----------
  [48,  8]: 1
-------------------
```

Gröbner Fan and Gfan library

There are several functions for accessing the operations of the `Gfan` library [18] (type `?gfan` for their manual pages). Using them A. Jensen implemented in CoCoA-5 a function for computing all distinct Gröbner bases:

```
/**/ use QQ[x,y,z];
/**/ I := ideal(x^3 +x*y -z,  x^2 -y*z);
/**/ GF := GroebnerFanIdeals(I);  indent(GF);
[
  ideal(x^2-y*z,  x*y*z+x*y-z,  y^2*z^2+y^2*z-x*z),
  ideal(x^2-y*z,  x*z-y^2*z^2-y^2*z,  y^3*z^2+...),
  ...
```

Since the Gröbner fan can sometimes be quite large, the function `GroebnerFanIdeals` will print a "$*$" as each new Gröbner basis is found if the verbosity level is 10 or higher.

Another problem with ideals having large Gröbner fans (with thousands of different orderings and bases) is that storing all the different ideals becomes impracticable. However, typically we are interested only in those bases satisfying a certain property. The function `CallOnGroebnerFanIdeals` calls a given function successively on each of the distinct Gröbner bases, without storing them in a huge list. This needs a little technical ability, but might make the difference between getting an answer or filling up the computer's memory.

Here is an example showing how to print out only those Gröbner bases comprising 3 elements (and the corresponding order matrix):

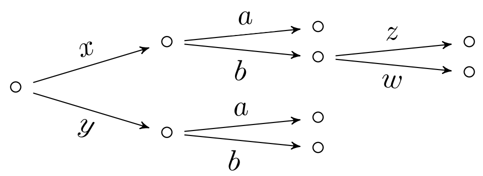
```
/**/ define GBOfLen3(I)
       if len(ReducedGBasis(I)) = 3 then
         println OrdMat(RingOf(I));
         println ReducedGBasis(I);
       endif;
     enddefine;
/**/ I := ideal(a^5+b^3+c^2-1,
               b^2+a^2+c-1,
               c^3+a^6+b^5-1);
/**/ CallOnGroebnerFanIdeals(I, GBOfLen3);
matrix(ZZ,
 [[1, 1, 1],
  [0, 0, -1],
  [0, -1, 0]])
[x^2 -y*z,  x*y*z +x*y -z,  y^2*z^2 +y^2*z -x*z]
....
```

Staged trees

A surprising application of monomial ideals in statistics is described in [17]. Given a polynomial whose power-products index all the cases in a data-set, find all *staged trees*

```
/**/ use QQ[a,b,x,y,z,w];
/**/ c_T := b*x*z +b*x*w +a*x +b*y +a*y;
/**/ trees := StagedTrees(c_T);
/**/ PrintTrees(trees);
-- number of trees = 2 -----------------------
-------- tree 1 ------------------------------
-- florets: [[x,  y],  [a,  b],  [z,  w]]
<
 x <
      a
      b <
           z
           w
 y <
      a
      b
-------- tree 2 ------------------------------
-- florets: [[a,  b],  [x,  y],  [z,  w]]
...
```

The first part is a textual representation for the tree



Knowing all such trees is a new key for the investigation of putative causal interpretations of datasets.

Hypersurface Implicitization

An efficient implementation of implicitization of hypersurfaces was recently added to CoCoALib. The CoCoA-5 interface has been made more user-friendly for rational parametrizations:

```
/**/ K := NewFractionField(RingQQt(1));
/**/ use K;
/**/ ParamDescr:=[ (1-t^2)/(1+t^2), 2*t/(1+t^2) ];
/**/ ImplicitHypersurface(ParamDescr);
x[1]^2 +x[2]^2 -1
```

Full details of the algorithms can be found in [8].

Minimal polynomials

Let $P = K[x_1, \ldots, x_n]$ be a polynomial ring with coefficients in a field K, and let I be a zero-dimensional ideal in P. It is well-known that the zero-dimensional affine K-algebra $R = P/I$ is a finite-dimensional K-vector space. Consequently, it is not surprising that minimal and characteristic polynomials from Linear Algebra can be successfully used to detect properties of R. For example, in the quotient ring $R = \mathbb{Q}[x, y]/(x^2, y^2)$ the minimal polynomial for the residue class of $x + y$ is z^3.

This point of view was taken systematically in the book [19] where the particular importance of minimal polynomials (rather greater than that of characteristic polynomials) emerged quite clearly. We studied the theory of minimal polynomials, developing efficient algorithms to compute them, and finding practical and efficient applications; here we recall the main results we described in [10] and in [20].

The first step was to implement in CoCoALib algorithms for computing the minimal polynomial of an element of R or that of a K-endomorphism of R. These functions are also accessible from CoCoA-5: type `?MinPolyQuot` for documentation.

Here is an example where we verify that the algebraic extension made by $\{\sqrt{2}, \sqrt[3]{3}, \sqrt[5]{5}, \sqrt[7]{7}\}$ is actually a field, and that $\sqrt{2} + \sqrt[3]{3} + \sqrt[5]{5} + \sqrt[7]{7}$ is a primitive element for the extension:

```
/**/ use P ::= QQ[a,b,c,d];
/**/ I := ideal(a^2-2, b^3-3, c^5-5, d^7-7);
/**/ IsMaximal(I); --> Is P/I is a field?
true
/**/ multiplicity(P/I);
210
/**/ MinPolyQuot(a+b+c+d, I, a);
a^210 -210*a^208 -210*a^207 + 21840*a^206 + ...
```

For computing minimal polynomials of elements of an algebra over $\mathbb{Q}$ we use an efficient modular approach and the rational reconstruction described in [4]. In particular, we dealt with the notion of *reduction of an ideal modulo* p, introducing the related notions of *ugly, usable, good and bad primes*. This is further investigated in a forthcoming paper [11].

As mentioned earlier, an efficient way of computing minimal polynomials lets us compute quickly a number of other important invariants of zero-dimensional affine K-algebras. For instance, in [10] we described algorithms for testing whether a zero-dimensional ideal is radical, maximal or primary, and others for computing the radical and the primary decomposition of a zero-dimensional ideal. In particular, it is noteworthy that in the case of coefficient fields of small finite characteristic, Frobenius spaces play a fundamental role, as described in [19, 20]. All the algorithms described have been implemented in CoCoA. Their efficiency is exhibited by the tables of timings in [10].

The computation of the primary decomposition of a zero-dimensional ideal computes minimal polynomials "behind the scenes":

```
/**/ use P ::= QQ[x,y,z];
/**/ I := ideal(3*y^2*z^2 +3*x*y^2 +1,
                x^4 +y*z^3,  y^4 +y*z^3)
/**/ PD := PrimaryDecomposition0(I);
/**/ len(PD);
9
```

The efficient computation of minimal polynomials is also a key ingredient in solving polynomial systems, and together with the other related algorithms we believe we can develop good tools for the SC^{2}community (see below).

News about SC^2

The CoCoA group is a member of the EU project "SC^2, Satisfiability Checking and Symbolic Computation" [1], so we are now collaborating with various members of the *Satisfiability Modulo Theory* community. In particular, we have built a first interface between CoCoALib and the software MathSAT [16] in close collaboration with A. Griggio. There is also a separate collaboration with the developers of SMTRAT [13].

A simple example of calling the MathSAT function `msat_solve` from CoCoALib can be found in the example program `ex-MathSat1`. A more advanced interface, with dedicated CoCoALib C++ wrapper class for the data type `msat_env`, is part of the forthcoming CoCoALib 0.99560 distribution (September 2017).

The current interface permits CoCoA to use MathSat capabilities for solving linear inequalities; in the future the interface will be bi-directional allowing MathSAT to use CoCoA for tackling polynomial inequalities.

We illustrate the interface via an example. The linear inequalities are represented as matrices: each row representing one inequality. Consider the following system of inequalities:

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4 \leq 0 \\ 9x_1 + 8x_2 + 7x_3 \leq 0 \\ x_1 \neq 0 \\ x_1 + x_2 + 4 = 0 \\ x_2 < 0 \end{cases}$$

```
/**/ constraints :=
       record[ leq0 := matrix([ [1,2,3,  4],
                                [9,8,7,  0] ]),
               neq0 := matrix([ [1,0,0,  0] ]),
               eq0 := RowMat([1,1,0,  4]),
               lt0 := RowMat([0,1,0,  0])   ];
/**/ soln := MSatLinSolve(constraints);
/**/ GetCol(soln,1);
[-2, -2, -2/7]
```

This interface is still a "proof-of-concept" prototype; more MathSAT features will soon be added. The CoCoA-5 manual search `?MathSAT` will list all capabilities available in the current version.

Thanks

This research was partly supported by the project H2020-FETOPEN-2015-CSA_712689 of the European Union.
J. Abbott is an INdAM-COFUND Marie Curie Fellow.

References

[1] Erika Ábrahám, John Abbott, Bernd Becker, Anna M. Bigatti, Martin Brain, Bruno Buchberger, Alessandro Cimatti, James H. Davenport, Matthew England, Pascal Fontaine, Stephen Forrest, Alberto Griggio, Daniel Kroening, Werner M. Seiler, Thomas Sturm: SC^2*: Satisfiability checking meets symbolic computation (Project Paper)* LNCS 9791, pp. 28-43

[2] J. Abbott.: *The Design of CoCoALib* In N. Takayama, A. Iglesias (eds.) Proceedings of ICMS 2006, LNCS 4151:205–215. Springer (2006)

[3] J. Abbott: *Twin-Float Arithmetic* Journal of Symbolic Computation, in press: http://dx.doi.org/10.1016/j.jsc.2011.12.005 (2011)

[4] J. Abbott, *Fault-Tolerant Modular Reconstruction of Rational Numbers* J. Symb. Comp. 80P3, pp. 707–718, (2017).

[5] J. Abbott, A.M. Bigatti: *CoCoALib: a C++ library for doing Computations in Commutative Algebra* `http://cocoa.dima.unige.it/cocoalib/`

[6] J. Abbott, A.M. Bigatti, *CoCoA and CoCoALib: Fast prototyping and flexible C++ library for Computations in Commutative Algebra* Proc. 1st Workshop on Satisfiability Checking and Symbolic Computation, SC-Square 2016, CEUR Workshop Proceedings 1804, pp. 1–3, (2016).

[7] J. Abbott, A.M. Bigatti, M. Kreuzer, L. Robbiano, *Computing Ideals of Points*, Journal of Symbolic Computation 30, pp. 341–356 (2000).

[8] J. Abbott, A.M. Bigatti, L. Robbiano, *Implicitization of Hypersurfaces* Journal of Symbolic Computation 81, 20–40 (2017).

[9] J. Abbott, A.M. Bigatti, G. Lagorio: *CoCoA-5: a system for doing Computations in Commutative Algebra* `http://cocoa.dima.unige.it/`

[10] J. Abbott, A. Bigatti, E. Palezzato, L. Robbiano, *Computing and Using Minimal Polynomials* `arXiv:1704.03680`, (2017).

[11] J. Abbott, A. Bigatti, L. Robbiano, *Ideals modulo p* In preparation.

[12] J. Abbott, C. Fassino, M.L. Torrente: *Stable border basis for ideals of points* Journal of symbolic computation 43, pp. 883–894 (2008).

[13] E. Ábrahám *SMTRAT* Software available from `https://github.com/smtrat/smtrat/wiki`

[14] A. M. Bigatti, L. Robbiano. *Borel sets and sectional matrices* Ann. Comb., 1(1):197–213, (1997).

[15] A. M. Bigatti, E. Palezzato, M. Torielli. *Extremal behaviour in sectional matrices* `arXiv:1702.03292`, (2017).

[16] A. Cimatti, A. Griggio, B. Schaafsma, and R. Sebastiani: *The MathSAT5 smt solver* Proceedings TACAS 2013, volume 7795 of LNCS, pp. 93–107, (2013).

[17] C. Görgen, A.M. Bigatti, E. Riccomagno, J. Smith *Discovery of statistical equivalence classes using computer algebra* `arXiv:1705.09457` (2017).

[18] A.N. Jensen, *Gfan, a software system for Gröbner fans and tropical varieties.* Available from `http://home.math.au.dk/jensen/software/gfan/gfan.html`

[19] M. Kreuzer and L. Robbiano, *Computational Linear and Commutative Algebra* Springer, Heidelberg 2016.

[20] E. Palezzato, *Minimal Polynomials, Sectional Matrices, and Applications*; PhD thesis, Università degli Studi di Genova, (2017).

Neues aus Waterloo: Maple 2017

Thomas Richard, Maplesoft Europe GmbH

`trichard@maplesoft.com`

Mit etwas Verspätung gegenüber den Vorjahren ist Anfang Mai diesen Jahres Maple 2017 erschienen. Wie immer wurden Benutzeroberfläche, Mathematik-Library und Konnektivität (Verbindung mit anderer Software) überarbeitet. Wir wollen einige dieser Neuerungen kurz vorstellen. Die komplette Dokumentation ist wie gewohnt per Menüpunkt *Help* > *What's New* oder durch Eingabe von **?updates,v2017** zugänglich. Wer die neue Version noch nicht hat, kann all dies auf der Seite `https://www.maplesoft.com/products/maple/new_features/` nachlesen, wo die einzelnen Themen als PDF-Dateien verlinkt sind.

Oberfläche

Auf vielfachen Wunsch insbesondere von Industriekunden können Worksheets innerhalb von Workbooks (dem im Vorjahr eingeführten Container-Format mit der Dateiendung `.maple`) nun per Passwort vor Lesezugriff geschützt werden, obwohl sie per Befehl **DocumentTools:-RunWorksheet** für jeden Empfänger ausführbar bleiben und Parameter entgegen nehmen können. So lässt sich deren Funktionalität zur Verfügung stellen, ohne das eigene Know-How preiszugeben.

Stark vereinfacht wurde der Zugang zu Anwenderpaketen über die MapleCloud. Hier lassen sich Workbook-basierte Pakete nun mit einem Klick herunterladen und installieren. Alternativ geht dies per Befehl **PackageTools:-Install**. Die Webseiten der Cloud sind per Browser erreichbar unter `https://maple.cloud`, sodass man sich auch ohne Maple einen Überblick der verfügbaren Pakete verschaffen kann. Sie stammen zum Teil von Maplesoft selbst, zunehmend jedoch von Drittanbietern. Einige der ersten hochgeladenen Pakete zeigt Abb. 1. Langfristig soll diese Sammlung das eher traditionell aufgebaute Maple Application Center ablösen. Alle interaktiven MathApps lassen sich im Browser direkt bedienen – beispielsweise werden Gleichungen im Formelsatz mit einem simplen Editor eingegeben, ganz ohne Syntaxkenntnisse vorauszusetzen. Die restlichen herkömmlichen Worksheets lassen sich zumindest statisch ansehen. Insofern ähnelt dieser browser-basierte Zugang dem Maple Player.

Aus Maple 2017 heraus erreicht man die Cloud am einfachsten über das neue Wolken-Symbol oben rechts in der Button-Leiste. Hier kann man sehr komfortabel eigene Dateien verwalten, etwa Aufgabensammlungen für verschiedene Lehrinhalte. Die MapleCloud-Palette bleibt weiterhin verfügbar, um Benutzerkonten und Anwendergruppen (Kursteilnehmer, Arbeitskollegen, usw.) zu verwalten.

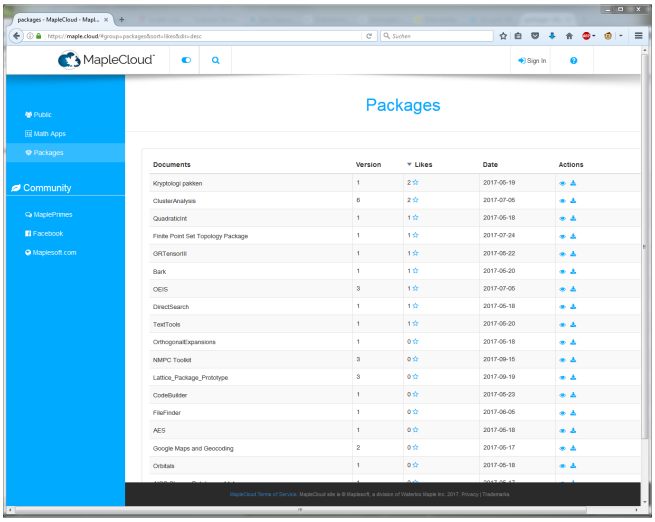


Abbildung 1: *Packages in der Maple Cloud*

Der völlig neu gestaltete interaktive Plot Builder ist besser durch ein Bild als durch 1000 Worte beschrieben. Wie man in Abb. 2 sieht, wird er als Panel rechts neben dem Plotbereich eingeblendet. Dieses Prinzip soll künftig auf die meisten Operationen der Kontextmenüs ausgedehnt werden, um Platz zu sparen und gleichzeitig Parameter eingeben zu können.

Etwas mehr Eyecatcher-Potential hat die eingebaute Datenbank **GeoNames** mit geografischen Informationen. Sie beantwortet Fragen der Art

- Wieviele Städte heißen Waterloo?
- Zu welcher Provinz gehören sie jeweils?
- Auf welchem Längen- und Breitengrad liegt das Waterloo in Ontario?
- Welche deutschen Bundesländer haben mehr als 5 Millionen Einwohner?

Dazu gibt es verschiedenste Arten von Weltkarten, Ausschnitten und Projektionen, die mit Ortsnamen und Ländergrenzen versehen und individuell (thematisch) eingefärbt werden können. Das Einzeichnen von Graphen und sonstigen Diagrammen ist einfach. Die Verknüpfung solcher Karten mit statistischen oder kombinatorischen Fragestellungen liegt nahe, etwa zur Visualisierung des Traveling-Salesman-Problems. Eine vergleichbare Aufgabenstellung zeigt Abb. 3.

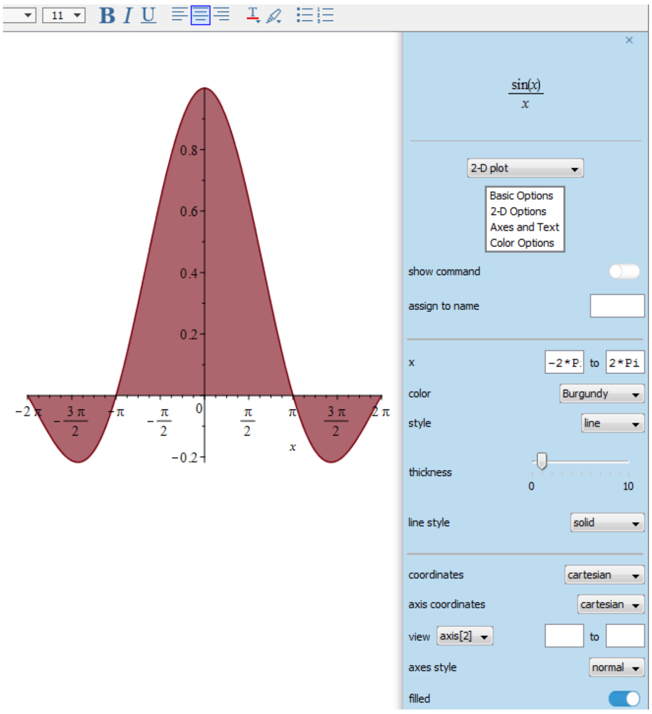
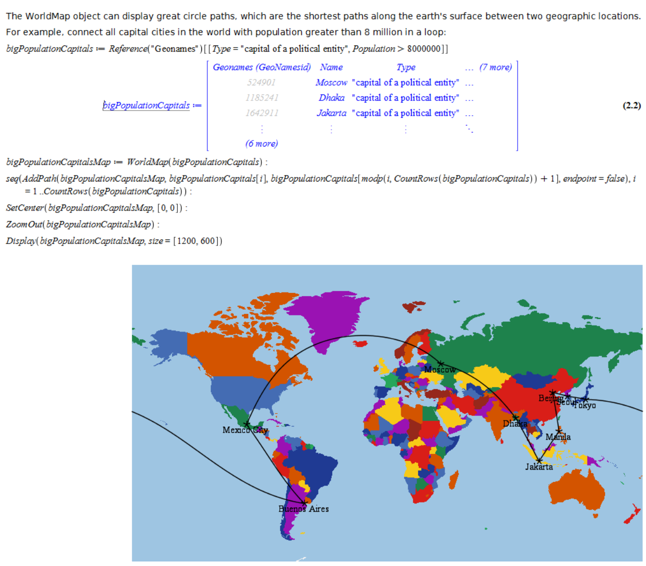


Abbildung 2: *Der interaktive Plot Builder.*

Das **Units**-Paket zum Rechnen mit Einheiten hat neben den bekannten Umgebungen **Standard** und **Natural** (sowie der Default-Umgebung) eine weitere erhalten. Sie heißt **Simple** und ermöglicht das flexible Kombinieren von Ausdrücken mit und ohne Einheiten. Dies war bisher nur umständlich realisierbar, sofern man keine Fehlermeldung aufgrund inkompatibler physikalischer Größen provozieren wollte. Damit ist nun auch die Verwendung von Symbolen ohne zugewiesenen Wert in solchen Ausdrücken möglich.


The WorldMap object can display great circle paths, which are the shortest paths along the earth's surface between two geographic locations. For example, connect all capital cities in the world with population greater than 8 million in a loop:

bigPopulationCapitals := Reference("Geonames")[[Type = "capital of a political entity", Population > 8000000]]

Geonames (GeoNamesid)	Name	Type	... (7 more)
524901	Moscow	"capital of a political entity"	...
1185241	Dhaka	"capital of a political entity"	...
1642911	Jakarta	"capital of a political entity"	...
⋮	⋮	⋮	⋱
(6 more)			

(2.2)

bigPopulationCapitalsMap := WorldMap(bigPopulationCapitals) :

seq(AddPath(bigPopulationCapitalsMap, bigPopulationCapitals[i], bigPopulationCapitals[modp(i, CountRows(bigPopulationCapitals)) + 1], endpoint = false), i = 1 ..CountRows(bigPopulationCapitals)) :

SetCenter(bigPopulationCapitalsMap, [0, 0]) :

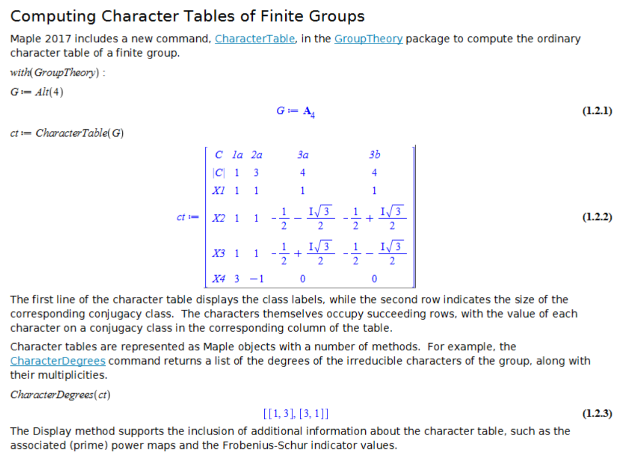
ZoomOut(bigPopulationCapitalsMap) :

Display(bigPopulationCapitalsMap, size = [1200, 600])

Abbildung 3: *Rundreise durch Städte ab 8 Mio. Einwohnern*

Mathematik

Das Paket **GroupTheory** wurde um einen Befehl zum Berechnen von Charaktertafeln erweitert, stößt also in Richtung Darstellungstheorie vor. Implementiert wurde dazu der Algorithmus von Burnside–Dixon–Schneider. In Abb. 4 ist die einfache Ausgabe zu sehen; mit einem **Display**-Befehl und diversen Optionen kann man eine schöner formatierte Tafel mit zusätzlichen Informationen ausgeben. Dies erfordert natürlich mehr Platz auf dem Bildschirm bzw. Papier. Weitere Verbesserungen betreffen die mitgelieferte Datenbank kleiner Gruppen und zugehörige Befehle wie **SearchSmallGroups** und **NumGroups**. Deren Namen dürften wohl selbsterklärend sein.


Computing Character Tables of Finite Groups

Maple 2017 includes a new command, CharacterTable, in the GroupTheory package to compute the ordinary character table of a finite group.

with(GroupTheory) :

G := Alt(4)

$$G := \mathbf{A}_4 \qquad (1.2.1)$$

ct := CharacterTable(G)

$$ct := \begin{bmatrix} C & 1a & 2a & 3a & 3b \\ |C| & 1 & 3 & 4 & 4 \\ X1 & 1 & 1 & 1 & 1 \\ X2 & 1 & 1 & -\frac{1}{2}-\frac{I\sqrt{3}}{2} & -\frac{1}{2}+\frac{I\sqrt{3}}{2} \\ X3 & 1 & 1 & -\frac{1}{2}+\frac{I\sqrt{3}}{2} & -\frac{1}{2}-\frac{I\sqrt{3}}{2} \\ X4 & 3 & -1 & 0 & 0 \end{bmatrix} \qquad (1.2.2)$$

The first line of the character table displays the class labels, while the second row indicates the size of the corresponding conjugacy class. The characters themselves occupy succeeding rows, with the value of each character on a conjugacy class in the corresponding column of the table.

Character tables are represented as Maple objects with a number of methods. For example, the CharacterDegrees command returns a list of the degrees of the irreducible characters of the group, along with their multiplicities.

CharacterDegrees(ct)

$$[[1, 3], [3, 1]] \qquad (1.2.3)$$

The Display method supports the inclusion of additional information about the character table, such as the associated (prime) power maps and the Frobenius-Schur indicator values.

Abbildung 4: *Das Berechnen von Charaktertafeln.*

Die enge Verbindung von Gruppen- und Graphentheorie zeigt sich in neuen Befehlen zur Berechnung der Automorphismengruppe eines Graphen sowie zur Visualisierung seiner Automorphismen durch eine Animation des Graphen. Weitere Neuerungen betreffen die Befehle **CanonicalGraph**, **Eccentricity**, **Radius**, außerdem sechs zusätzliche Graphen im Unterpaket **SpecialGraphs** und die Unterstützung für das Digraph6-Format mit der Dateiendung `.d6`.

Das seit Maple 2016 neue Paket **NumberTheory** wurde nun komplettiert, insbesondere mit dem verallgemeinerten Chinesischen Restsatz (bei dem die „Moduli“ nicht paarweise teilerfremd sein müssen) und einer Datenbank zum Faktorisierungsstatus der Fermatschen Zahlen. Es kann somit das alte **numtheory**-Paket ablösen, geht aber in der Funktionalität klar darüber hinaus. Beispielsweise findet **SimplestRational** die einfachste rationale Approximation einer reellen Zahl, d.h. mit minimalem Zähler und Nenner.

Höhere Geschwindigkeit gegenüber Maple 2016 bei der Berechnung lexikographischer Gröbnerbasen verspricht die neue Implementierung des FGLM-Algorithmus in einer C-Library. Sie wird vom Befehl **Basis** des **Groebner**-Pakets und ggfs. vom allgemeineren Top-Level-Befehl **solve** genutzt. Auch viele grundlegende Operationen in der Polynomarithmetik und beim Arbeiten mit rationalen Funktionen profitieren von neuen und effizienteren Methoden im Maple-Kernel.

Die meisten Beispiele demonstrieren Steigerungen der Geschwindigkeit (teils um mehrere Größenordnungen), einige hingegen Speicherplatz-Ersparnisse.

Kommen wir nun von der diskreten Mathematik zur kontinuierlichen, also zur Analysis. Zu den häufigst gebrauchten Top-Level-Befehlen gehören **int**, **limit** und **sum**. Wie schon in den Vorjahren findet das überarbeitete **int**-Kommando wiederum mehr bestimmte und unbestimmte Integrale. Der **limit**-Befehl wurde abermals um neue Methoden für Grenzwerte bivariater Funktionen ergänzt. Soweit uns bekannt, ist Maple derzeit das einzige System mit dieser Fähigkeit. Das **sum**-Kommando liefert seine Ergebnisse möglichst mittels Binomialkoeffizienten, wenn die Summanden ebenfalls von dieser Form sind – etwas vereinfachend gesagt. Für Details sei jeweils auf die in der Update-Dokumentation eingebauten Beispiele und Kommentare verwiesen. Manche Verbesserungen sind recht speziell, etwa asymptotische Entwicklungen der unvollständigen Gammafunktion oder die exakte Auswertung der Lambertschen W-Funktion mit bestimmten logarithmischen Ausdrücken im Argument.

Erstmals in Maple implementiert wurden die vier Appell-Funktionen F_1 bis F_4, die man als Verallgemeinerung der hypergeometrischen Funktionen auf zwei Veränderliche beschreiben kann. Dazu gibt es ein umfangreiches Framework namens **MathematicalFunctions** mit seinem Unterpaket (und gleichnamigem Befehl) **Evalf**, welches zur numerischen Auswertung dient, inklusive geeigneter Transformationen und der Bestimmung von Singularitäten. Dazu können automatisch Plots generiert werden, welche diese Vorgänge veranschaulichen.

Der **pdsolve**-Befehl zum Lösen partieller Differentialgleichungen erfasst immer mehr Randwertprobleme auf endlichen oder halb-unendlichen Gebieten und wendet – wo erlaubt und zielführend – Fourier- und Laplacetransformation an. Mit der Option *generalsolution* kann man die Suche nach der allgemeinsten Form einer Lösung veranlassen. Ob sie gefunden werden konnte, zeigt der Solver auch an, falls vorher **infolevel[pdsolve]** auf einen ausreichend hohen Wert gesetzt wurde. Bisher wurde **infolevel** nur genutzt, um eine textbasierte Fortschrittsanzeige und die aufgerufenen Methoden ausgeben zu lassen. Die dabei generierte Datenmenge kann den Anwender jedoch leicht überfordern, daher sollte er den **infolevel** nur so hoch wählen wie unbedingt nötig.

Dieser Themenbereich (wie auch Spezielle Funktionen, bestimmte **simplify**-Routinen u.v.m.) wird fortlaufend mit Updates versorgt, welche außerhalb der regulären Maple-Updates erscheinen und von den Projektseiten zu „Mathematical Functions & Differential Equations“ heruntergeladen werden können. Entsprechendes gilt für das mittlerweile sehr umfangreiche Paket **Physics**. Hier bietet Maple 2017 u.a. eine Klassifikation aller Lösungen der Einstein-Gleichungen, neue Routinen für Tensoren in der Speziellen und der Allgemeinen Relativitätstheorie und ein Unterpaket zum Rechnen im Standardmodell der Teilchenphysik. Der Hauptentwickler dieser Themengebiete beschreibt die Neuerungen regelmäßig auf der Community-Seite MaplePrimes und diskutiert dort auch Feature-Wünsche der Anwenderschaft – nicht nur aus der Physik.

Connectivity

Wie schon in den Vorjahren hat der Quelltext-Generator eine neue Zielsprache hinzu gewonnen: diesmal ist es Swift, was die Firma Apple als Nachfolger von ObjectiveC einstuft. Wegen des stark modularisierten Aufbaus des **CodeGeneration**-Pakets war der Implementierungsaufwand relativ gering. Im Prinzip muss man nur die Unterschiede zu einer bereits vorhandenen Zielsprache spezifizieren, und ein Dutzend solcher Sprachen waren bereits abgedeckt.

Als Nebenprodukt der Kooperation im EU-Projekt SC^2 (Satisfiability Checking and Symbolic Computation, siehe `http://www.sc-square.org/`) wurde ein kleines Paket implementiert, welches Maple-Ausdrücke in das SMT-LIB-Format konvertiert (Satisfiability Modulo Theories). Außerdem wurde an der in Maple 2016 gestarteten MiniSat-Anbindung in den Befehlen **Satisfy** und **Satisfiable** des **Logic**-Pakets gearbeitet: sie ist nun komplett und mit allen Optionen dokumentiert. Weitere Schritte in Richtung SAT-Solving sind hier in den nächsten Jahren zu erwarten.

Technisches

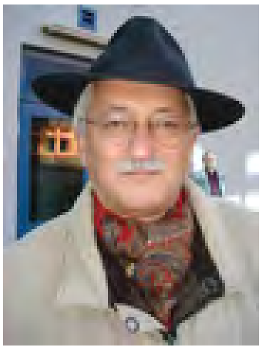
Für den Herbst ist ausnahmsweise noch ein drittes Update – also Maple 2017.3 – geplant, u.a. um Anpassungen an das gleichzeitig erscheinende MapleSim 2017 vorzunehmen.

Den Administratoren von Lizenzservern wird der Umstieg auf die neue Version 2017 der „Maplesoft Network Tools“ empfohlen, weil der Lizenzmanager FlexNet darin auf Version 11.13 aktualisiert worden ist. Die vorher mitgelieferte Version 11.7 enthielt eine Sicherheitslücke und bereitete außerdem Schwierigkeiten mit bestimmten VPN-Anbindungen. Auch die Hilfsprogramme und der „Vendor Daemon“ maplelmg wurden angepasst an neue Zusatzprodukte. Dies betrifft jedoch eher MapleSim-Anwender.

Voraussichtlich im vierten Quartal soll ein Update des kostenlosen Maple Players auf Version 2017 erscheinen, welches dann auch e-Books in bestimmten Formaten unterstützen wird.

Bei der Plattform-Unterstützung für alle hier genannten Produkte hat sich wenig getan, d.h. es werden weiterhin die gängigen 64-bit-Betriebssysteme unterstützt. Lediglich für Windows gibt es derzeit noch eine separate 32-bit-Variante. Die laufenden (teils mit Seiteneffekten behafteten) Änderungen an Windows 10, aber auch die jüngsten Versionen von Mac OS X und einigen Linux-Distributionen stellen eine gewisse Herausforderung für Software-Hersteller dar. Bei etwaigen Problemen sowie bei Feature-Vorschlägen und sonstigen technischen Fragen sollte man sich auf Deutsch oder Englisch an die Adresse `de_support@maplesoft.com` wenden.

Zur Symmetrie einiger Kurven



J. Meyer
(Hameln)

j.m.meyer@t-online.de

Zusammenfassung

Die Parameterdarstellung eines Kreises wird variiert, um Kurven zu erhalten, deren Symmetrieverhalten den erzeugenden Termen nicht unmittelbar anzusehen ist. Schülerinnen und Schüler erhalten erstens eine nichttriviale Veranlassung, Matrizen zu verwenden, und sind zweitens mitunter angehalten, den Punkten auf der Kurve Parameter zuzuordnen, um den Nachweis von Punkt- oder Achsensymmetrie führen zu können. Einer der Kurventypen liefert im rotationssymmetrischen Fall sogar eine Veranlassung, sich etwas mit Kongruenzrechnung zu befassen.

1. Einleitendes Beispiel

Bekanntlich beschreibt $r \cdot \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}$ für $t \in \mathbb{R}$ einen Kreis mit Radius r. Was für eine Kurve erhält man, wenn man den Term etwas aufmischt, also zu

$$P(t) = \begin{pmatrix} r \cdot \cos(a \cdot t) + s \cdot \sin(b \cdot t) \\ r \cdot \sin(a \cdot t) + s \cdot \cos(b \cdot t) \end{pmatrix}$$

übergeht? Abb. 1 zeigt das Resultat für $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ sowie $s = 0,7 \cdot r$.

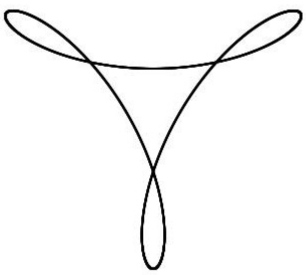


Abbildung 5

Die Symmetrie ist auffällig und nicht direkt aus der Gestalt für $P(t)$ ersichtlich. Wie lässt sie sich nachweisen? Offenbar muss eine Drehung von $P(t)$ um $\varphi = \frac{2\pi}{3}$ dasselbe liefern wie eine Parameterverschiebung um φ, d. h. es muss mit der Drehmatrix $D_\varphi = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$ die Gleichung

$$D_\varphi \cdot P(t) = P(t + \varphi) \tag{1}$$

gelten. Deren Gültigkeit ist mit einem CAS schnell bestätigt.

2. Didaktische Anmerkungen

Hinter dem Nachweis der Gültigkeit von (1) stecken natürlich die Additionstheoreme, die schon lange kein Element der Schulcurricula mehr sind; wären sie es, bräuchte man für (1) kein CAS. Auch die komplexen Zahlen sind schon lange aus der Schule verschwunden; hätte man sie, könnte man $P(t)$ als

$$P(t) = r \cdot e^{i \cdot a \cdot t} + s \cdot e^{i \cdot \left(\frac{\pi}{2} - b \cdot t\right)}$$

schreiben und sofort sehen, dass mit $\varphi = \frac{2\pi}{3}$ für $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ die Beziehung $P(t+\varphi) = e^{i \cdot \varphi} \cdot P(t)$ gilt. In beiden Fällen kompensiert der Einsatz eines CAS also fehlende mathematische Kenntnisse. Eine wirkliche Einsicht liefert nur der Weg über komplexe Zahlen.

3. Anschlussfragen

Die Beziehung (1) ist unabhängig von der Wahl von r und s gültig. Variiert man hingegen a und b, so hat man schnell die Vermutung, es mit einer $(a + b)$-zähligen Symmetrie zu tun zu haben. In der Tat gilt mit $\varphi = \frac{2\pi}{a+b}$ stets

$$P(t + \varphi) = D_{a \cdot \varphi} \cdot P(t) = D_{-b \cdot \varphi} \cdot P(t), \tag{2}$$

denn aus

$$P(t) = r \cdot e^{i \cdot a \cdot t} + s \cdot e^{i \cdot \left(\frac{\pi}{2} - b \cdot t\right)}$$

folgt

$$P(t+\frac{2\pi}{a+b})=e^{i\cdot a\cdot\frac{2\cdot\pi}{a+b}}\cdot P(t).$$

Für konkrete Werte von a und b ist die Überprüfung von (2) mit einem einfachen CAS wie Maxima kein Problem.

4. Eine andere Kurvenschar

Eine andere Kurvenschar ist gegeben durch eine leichte Veränderung: Nun sei

$$P(t)=\begin{pmatrix}r_1\cos(t)+r_2\cos(at)+r_3\sin(bt)\\ r_1\sin(t)+r_2\sin(at)+r_3\cos(bt)\end{pmatrix}. \quad (3)$$

Abb. 2 zeigt den Fall $\begin{pmatrix}a\\ b\end{pmatrix}=\begin{pmatrix}5\\ 8\end{pmatrix}$; die r_i spielen für das Symmetrieverhalten keine Rolle.

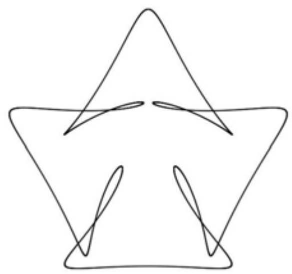


Abbildung 6

Die Kurve ist achsensymmetrisch, wie sich leicht nachprüfen lässt; es ist nämlich

$$\begin{pmatrix}-1&0\\ 0&1\end{pmatrix}\cdot P(t)=P(\pi-t).$$

So ganz trivial ist der Umgang mit dem CAS dann doch nicht, da man die Parameter geeignet wählen muss.

5. Erste Variation

Wählt man statt dessen $\begin{pmatrix}a\\ b\end{pmatrix}=\begin{pmatrix}5\\ 9\end{pmatrix}$, so ergibt sich Abb. 3 mit zwei Achsensymmetrien.

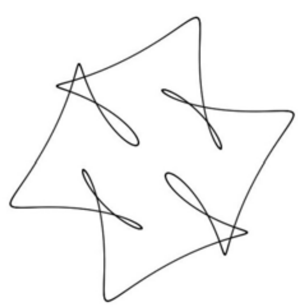


Abbildung 7

Dreht man die Kurve mit einer Matrix R um $45°$, so ist in der Tat

$$\begin{pmatrix}1&0\\ 0&-1\end{pmatrix}\cdot R\cdot P(t)=R\cdot P(\frac{3}{2}\pi-t)$$

sowie

$$\begin{pmatrix}-1&0\\ 0&1\end{pmatrix}\cdot R\cdot P(t)=R\cdot P(\frac{\pi}{2}-t).$$

6. Zweite Variation

Mit $\begin{pmatrix}a\\ b\end{pmatrix}=\begin{pmatrix}3\\ 5\end{pmatrix}$ erhält man Abb. 4, die wegen $\begin{pmatrix}-1&0\\ 0&-1\end{pmatrix}\cdot P(t)=P(\pi+t)$ punktsymmetrisch ist.

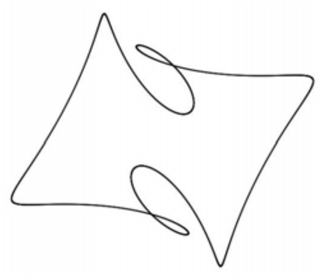


Abbildung 8

7. Dritte Variation

Interessanter ist der Fall $\begin{pmatrix}a\\ b\end{pmatrix}=\begin{pmatrix}6\\ 4\end{pmatrix}$, der eine 5-zählige Symmetrie liefert (Abb. 5). Dies lässt sich mit Hilfe eines CAS wieder leicht überprüfen, liefert allerdings wiederum keinerlei Einsicht. Schreibt man jedoch den zugehörigen Kurvenpunkt (3) als

$$P(t)=r_1\cdot e^{i\cdot t}+r_2\cdot e^{i\cdot a\cdot t}+r_3\cdot e^{i\cdot\left(\frac{\pi}{2}-b\cdot t\right)},$$

so bekommt man eine 5-zählige Symmetrie genau dann, wenn mit $\varphi=\frac{2\pi}{5}$ die Beziehung

$$e^{i\cdot\varphi}\cdot P(t)=P(t+\varphi)$$

gilt. Das ist der Fall für $a\cdot\varphi=\varphi$ und $-b\cdot\varphi=\varphi$. Nun ist aber $6\cdot\varphi=\varphi$ und $-4\cdot\varphi=\varphi$. Nun weiß man auch, wie man weitere rotationssymmetrische Kurven vom Typ (3) erzeugen kann.

Abbildung 9

8. Schlussbemerkung

Man könnte immer so weitermachen. Es ist bemerkenswert, welcher Formen- und Symmetriereichtum in (3) steckt, wie man etwa an Abb. 6 mit $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$ und an Abb. 7 mit $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 5 \\ 12 \end{pmatrix}$ sehen kann.

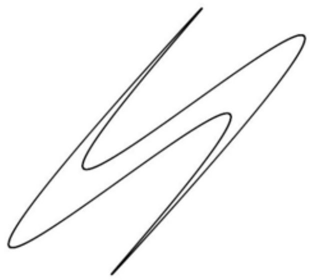


Abbildung 10

Abbildung 11

Allerdings weisen nicht alle derartige Kurven Symmetrien auf, wie Abb. 8 mit $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$ zeigt.

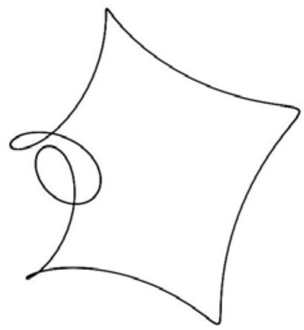


Abbildung 12

Berichte über Arbeitsgruppen

Explizite Methoden in der Arithmetischen Geometrie

Jeroen Sijsling (Ulm)

Zum Institut für Reine Mathematik an der Universität Ulm gehören Irene Bouw und Stefan Wewers (Professoren), und weiter zurzeit Jeroen Sijsling (Juniorprofessor), Angelos Koutsianas (Postdoc), Jeroen Hanselman, Roman Kohls, Tudor Micu und Christian Steck (Doktoranden).

Das aktuelle Hauptthema der Forschung dieser Arbeitsgruppe ist die Bestimmung von arithmetischen Invarianten von Kurven von kleinem Geschlecht. Die Forschung von Bouw, Wewers, Koutsianas und Ihren Doktoranden beschäftigt sich mit der expliziten Berechnung von Führern von solchen Kurven, und der Bestimmung von Kurven mit möglichst kleinem Führer. Hanselman und Sijsling arbeiten an der expliziten Bestimmung von Endomorphismenringen von Jakobischen sowie umgekehrt an der Konstruktion von Jakobischen mit einem interessanten Endomorphismenring.

Eine wichtige Anwendung dieser Forschung ist ihr Beitrag zur LMFDB, wofür man in Zusammenarbeit mit unter anderen Andrew Sutherland (MIT) bald eine große Datenbank von Kurven vom Geschlecht 3 konstruieren wird, zusammen mit allen ihren interessanten arithmetischen Invarianten.

Weitere Themen sind das Studium von glatten ebenen Kurven mit Reynald Lercier und Christophe Ritzenthaler (Rennes) und die Zerlegung von Jakobischen mit Davide Lombardo (Pisa) und Elisa Lorenzo García (Rennes). Auch diese Themen sind orientiert auf explizite und wenn möglich algorithmische Ergebnisse; ihre Implementierung findet hauptsächlich in Magma und Sage statt.

Stefan Wewers arbeitet zusammen mit Julian Rüth (Bogota) an einem Sage-Packet 'mclf' (Models of Curves over Local Fields) zur Bestimmung von regulären und semistabilen Modellen von Kurven über p-adischen Körpern. Dieses Projekt benutzt ganz wesentlich die Sage-Packete 'mac_lane' und 'completion' von Julian Rüth.

Algorithmisierung abstrakter algebraischer und geometrischer Konstruktionen

Mohamed Barakat (Siegen)

Unsere Arbeitsgruppe beschäftigt sich seit einigen Jahren, auch vor dem Wechsel nach Siegen, zunehmend mit der Algorithmisierung von abstrakten algebraischen und geometrischen Konstruktionen. Die Schlüsselerkenntnis kann wie folgt formuliert werden:

Die Kategorientheorie ist eine Programmiersprache.

Genauer: Die Kategorientheorie ist sowohl eine Sprache zur flexiblen Modellbildung in der Computeralgebra, als auch die korrekte Programmiersprache für die Implementierung dieser, oft hochabstrakten Modelle. In der Tat ist die Kategorientheorie eine der mächtigsten organisierenden Theorien der modernen Mathematik. Dadurch dass viele mathematische Invarianten rein kategorieller Natur sind, erlaubt die Kategorientheorie den Wechsel zwischen scheinbar sehr unterschiedlichen, jedoch äquivalenten Kategorien. Durch diese Erkenntnis und ihre Umsetzung wird die algorithmische Mathematik vom engen Korsett der mengentheoretischen Modellbildung befreit und um völlig neue Dimensionen erweitert. Dies ermöglicht uns Datenstrukturen der klassischen Modellbildung durch wesentlich verschiedene, im Idealfall deutlich kompaktere zu ersetzen, die sich auf Laufzeiten von Algorithmen positiv auswirken. Ist z. B. R ein berechenbarer kommutativer Ring und $\mathfrak{p}$ ein endlich erzeugtes Primideal in R, so ist das Rechnen in der $\mathfrak{p}$-Lokalisierung der Kategorie der endlich präsentierten R-Moduln viel effizienter als in der Modulkategorie über dem lokalisierten Ring $R_{\mathfrak{p}}$. Damit dies gelingt, muss man einen algorithmischen Rahmen schaffen, in dem die Lokalisierung abelscher Kategorien ausdrückbar und durchführbar ist. Ein solcher Zugang erlaubt sogar die Algorithmisierung von "endlich präsentierten Moduln über nicht-affinen Schemata", sprich von kohärenten Garben.

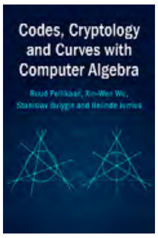
Die Entwicklung, der vielfältige Einsatz, und die Verbreitung dieser Programmiersprache wird unsere Marschrichtung in Forschung und Lehre für die nächsten Jahre bestimmen. Langfristig sollen abstrakte Konstruktionen wie

- derivierte Kategorie und derivierte Äquivalenzen
- Quillen-Modell Kategorien
- Fourier-Mukai Transformationen
- Grothendiecks 6-Operationenkalkül

zusammen mit weiteren Anwendungen in der algebraischen Geometrie dem Computer zugänglich gemacht werden.

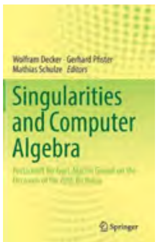
Publikationen über Computeralgebra

Neuerscheinungen:

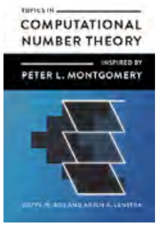


Ruud Pellikaan, Xin-Wen Wu, Stanislav Bulygin, Relinde Jurrius,
Codes, Cryptology and Curves with Computer Algebra: Volume 1,
Cambridge University Press, 2017, 600 Seiten,
ISBN 978-0521520362

Tufan Şirin,
Computer Algebra Applications for Electrostatics,
Scholars' Press, 2017, 176 Seiten,
ISBN 978-3659845987



Wolfram Decker, Gerhard Pfister, Mathias Schulze (Editors),
Singularities and Computer Algebra 2017,
Springer, 2017, 389 Seiten,
ISBN 978-3319288284



Joppe W. Bos, Arjen K. Lenstra (Editors),
Topics in Computational Number Theory Inspired by Peter L. Montgomery,
Cambridge University Press, 2017, 263 Seiten,
ISBN 978-1107109353

Die Rubrik Publikationen ist nicht allein auf eine Liste von Neuerscheinungen und Neuauflagen beschränkt. Sie lebt vor allem von fundierten Rezensionen von Fachgruppenmitgliedern für Fachgruppenmitglieder, die wir an dieser Stelle gerne abdrucken. Sollte eines der oben genannten Bücher, insbesondere eine der Neuerscheinungen, Ihr Interesse geweckt haben, und Sie möchten dieses für den Computeralgebra-Rundbrief besprechen, nehmen Sie bitte Kontakt zu Florian Heß oder Martin Kreuzer (`florian.hess@uni-oldenburg.de`, `martin.kreuzer@uni-passau.de`) auf.

Besprechungen zu Büchern der Computeralgebra

Martin Kreuzer, Lorenzo Robbiano
Computational Linear and Commutative Algebra

Springer International Publishing Switzerland (2016), 321 Seiten, ISBN 978-3-319-43599-2, € 44,02

This book is an lovely blend of commutative and linear algebra. The authors previously collaborated on the two-volume work *Computational Commutative Algebra*, published by Springer in 2000 and 2005. Their new book has a different structure (no exercises or tutorials) but fortunately preserves the irreverent humor of the earlier volumes. It is an interesting and original text. The motivation for the book comes from the study of systems of polynomial equations

$$(1) \qquad f_1(x_1,\dots,x_n) = \cdots = f_m(x_1,\dots,x_n) = 0$$

that have finitely many solutions over the algebraic closure of the coefficient field K. In this situation, the quotient ring $R = K[x_1,\dots,x_n]/\langle f_1,\dots,f_m\rangle$ is a finite-dimensional vector space over K, and an element $f \in R$ gives the linear map $\theta_f : R \to R$ defined by $\theta_f(g) = fg$. This is where the authors quote a Woody Allen joke that involves the biblical injunction "be fruitful and multiply." Since R is commutative, the mutliplication maps θ_f form a commuting family of endomorphisms of a finite-dimensional vector space. One of the central themes of the book is that such families are indeed a fruitful object of study.

Chapter 1 considers a K-linear map $\varphi : V \to V$, where $\dim_K(V) < \infty$. Since the authors want K to be arbitrary, the eigenvalues and eigenspaces of φ are less important. Rather, the minimal polynomial of φ takes center stage, and its irreducible factors, called *eigenfactors*, assume the role of eigevalues. An eigenfactor has an eigenspace and a generalized eigenspace, and V is the direct sum of the generalized eigenspaces, as one would expect. Let me also comment on terminology. In linear algebra, a linear map $V \to V$ whose minimal and characteristic polynomials agree is *non-derogatory*. This is relevant because the existence of a non-derogatory $\theta_f : R \to R$ has strong consequences for the structure of R. The authors introduce the term *commendable* to replace non-derogatory, arguing that "Would you call a beautiful painting 'non-ugly'?" My own feeling about commendable vs. non-derogatory is that it is similar to smooth vs. non-singular in algebraic geometry. I think there is place for both.

In Chapter 2, things get more sophisticated as the focus shifts to families of commuting endomorphisms of V. Such a family $\mathscr{F}$ is a K-algebra with V as a finitely-generated $\mathscr{F}$-module. The authors note that "a considerable dose of commutative algebra" is required, but this allows them to generalize the classical fact that *commuting diagonalizable matrices can be simultaneously diagonalized*, even when no eigenvalues lie in K. The maximal ideals of the algebra $\mathscr{F}$ and their associated kernels in V are essential tools. The primary decomposition of $\{0\} \subseteq \mathscr{F}$ is also important.

Chapter 3 is devoted to special types of families, including ones that contain a commendable element, naturally called commendable families. As an example of what this means algebraically, the authors note that when $\mathscr{F}$ is a field, it is commendable if and only if it has a primitive element in the sense of field theory. Another interesting family uses the dual map $\varphi\check{} : V^* \to V^*$ of $\varphi \in \mathscr{F}$ to create that dual family $\mathscr{F}\check{}$ that features in the next chapter.

Chapter 4 is for me the heart of the book. The focus is on $R = K[x_1, \ldots, x_n]/\langle f_1, \ldots, f_m \rangle$, $\dim_K(R) < \infty$. The commuting family $\mathscr{F} = \{\theta_f \mid f \in R\}$ is naturally isomorphic to R, so Chapters 2 and 3 apply to R. For example, if $\{0\} = \bigcap_i \mathfrak{q}_i$ is the primary decomposition of $\{0\} \subset R$, then the factorization into local rings

$$R \simeq \textstyle\prod_i R/\mathfrak{q}_i \tag{2}$$

expresses R as the product of the joint generalized eigenspaces of $\mathscr{F}$. Also, the dual family $\mathscr{F}\check{}$ acts natually on the dualizing module ω_R. By Chapter 3, $\mathscr{F}$ is commendable if and only if R is Gorenstein, a splendid link between linear algebra and commutative algebra.

The decomposition (2) highlights the importance of computing primary decomposition. This is a challenging problem, and Chapter 5 presents of state-of-the-art discussion for the rings and ideals considered here.

The book culminates in Chapter 6, which is devoted to the problem of finding the solutions of the system (1). The basic idea is that if $p \in K^n$ is a solution of (1) and $f \in R$, then $f(p)$ is an eigenvalue of both θ_f and $\theta_f\check{}$. This corresponds to 1-dimensional joint eigenspaces of $\mathscr{F}$ and $\mathscr{F}\check{}$, leading to some nice solution methods. The authors also explore what happens over finite fields and extension fields.

The book is well-written and includes many examples. Each chapter begins with a summary that motivates the (often substantial) mathematics to follow, and every method is accompanied by an algorithm, justifying the word "Computational" in the title. Some parts of the book are self-contained, while others require an acquaintance with their earlier volumes. For example, Algorithm 2.14 will baffle someone not familiar with the Buchberger-Möller algorithm. The book contains many new results and concepts, along with known ideas drawn from a widely scattered literature. Here they appear in a coherent general context. Overall, this book is a worthy contribution to both linear and commutative algebra.

David A. Cox (Amherst)

Promotionen in der Computeralgebra

Mario Albert: Computing Quot Schemes
Betreuer: Werner M. Seiler (Kassel)
Zweitgutachter: Margherita Roggero (Turin)
Februar 2017

Abstract: The main goal of this thesis is to develop computational methods which allow effective computations on Hilbert and Quot schemes. At first we introduce marked bases over modules. They may be considered as a form of Gröbner basis which do not depend on a term order. Instead, one chooses for each generator some term as head module term such that the head module terms generate a prescribed monomial module. We show that the involutive normal form algorithm with respect to Pommaret division will terminate if the prescribed monomial module is quasi-stable. For the introduction of marked bases over modules we introduce the concept of resolving decompositions which provide a unifying framework for computing free resolutions Then we investigate into the Hilbert function and the Hilbert polynomial. Furthermore, we analyse the important persistence and regularity theorem for ideals of Gotzmann. We provide for both theorems new alternative proofs which are much simpler to understand than previous proofs. They are based on the theory of Pommaret bases. We recall the definition of the Hilbert, Quot and Grassmann functors. Then we construct the "quasi-stable covering" of the Grassmann functor and show that we can restrict this covering to the Quot functor. The covering is represented by subfunctors. In an additional step we show that every subfunctor can be represented by a marked scheme. A marked scheme parametrizes all marked bases which belong to a prescribed quasi-stable module. Furthermore, we develop algorithms for the concrete computation of Quot schemes. At first we investigate the computation of saturated quasi-stable or p-Borel fixed monomial modules. At second we present two algorithms for computing marked schemes. By using these algorithms we show that the Hilbert scheme of 4 points in the projective 3-space is reduced. Moreover, we compute for the first time a concrete representation for a non-trivial Quot scheme and give an example for Hilbert schemes over fields of finite characteristic.

Simon Brandhorst: Existence and uniqueness of certain automorphisms on K3 surfaces
Betreuer: Matthias Schütt (Hannover)
Zweitgutachter: Keiji Oguiso (Tokyo), Ichiro Shimada (Hiroshima)
Mai 2017

Abstract: A projective K3 surface over an algebraically closed field $k = \overline{k}$, is a smooth, projective surface with $h^1(X, \mathcal{O}_X) = 0$ and $\omega_X \cong \mathcal{O}_X$. In this thesis we study the existence and uniqueness of K3 surfaces with certain automorphisms, both over $\mathbb{C}$ and in positive characteristic. Via a Torelli theorem, one can reformulate this problem in terms of isometries of indefinite lattices and certain convex cones. The results are obtained by a mixture of formal proofs and computer aided calculations. The main tools applied (and developed) are lattice theory, gluing as well as linear and quadratic programming.

In the first part, we treat automorphisms of finite order. For example we give a classification of purely non-symplectic automorphisms of high order and find projective models for all cases. Further, we prove that if $(\mathbb{Z}/5\mathbb{Z})^2$ acts faithfully on a K3 surface, then both the surface and the group action are unique (up to isomorphism). However, dropping the commutativity or rather finiteness of the group, we prove the existence of an infinite sequence of K3 surfaces admitting both a symplectic and a non-symplectic automorphism of order 5.

In the second part of the thesis, we turn to automorphisms of infinite order. They fall into two classes depending on whether they are compatible with an elliptic fibration or not. In the first case the algebraic (or topological) entropy of the automorphism is zero, while in the second case it is the logarithm of a Salem number. That is an algebraic integer $\lambda > 1$ which is Galois conjugate to $1/\lambda$ and all whose other conjugates lie on the unit circle. We give a strategy to prove or disprove the existence of an automorphism on a supersingular K3 surface realizing a given Salem number. The strategy is then applied in characteristic 5 to prove that the minimal Salem number λ_d of degree d is realized if and only if $d \leq 22$ is even and $d \neq 18$. As a by product, we close a case left open by McMullen - the existence of a complex projective K3 surface automorphism realizing λ_{12}.

Jasper Van Hirtum: Computational Aspects of Classical and Hilbert Modular Forms
Betreuer: Gabor Wiese (Luxembourg), Wim Veys (Leuven), Jan Tuitman (KU Leuven)
Komittee: Lassina Dembélé (Bonn), Andrew Sutherland (MIT), Martin Schlichenmaier (Luxembourg), Walter Van Assche (KU Leuven)
Juni 2017

Abstract: The main topic of this thesis is the study of classical and Hilbert modular forms and computational aspects of their q-expansions. The coefficients of q-expansions of eigenforms are particularly interesting because of their arithmetic significance. The thesis consists of two parts: the first part concerns the distribution of the coefficients of a given classical eigenform; the second part studies computational aspects of the adelic q-expansion of Hilbert modular forms of weight 1. Part I of this thesis presents a heuristic model that settles the following question related to the Sato-Tate and Lang-Trotter conjectures: given a normalised eigenform of weight 2 with quadratic coefficient field, what is the asymptotic behaviour of the number of primes p such that the p-th coefficient of this eigenform is a rational integer? This thesis contributes to this problem in two ways. First, it provides an explicit heuristic model that describes the asymptotic behaviour in terms of the associated Galois representation. Secondly, it is shown that this model holds under reasonable assumptions and numerical evidence supporting these assumptions is presented. Part II concerns the study of (adelic) q-expansions of Hilbert modular forms. The main achievements are the design, proof and implementation of several algorithms that compute the adelic q-expansions of Hilbert modular forms of weight 1 over the complex numbers and over finite fields. One reason for studying such q-expansions is that their coefficients (conjecturally) describe the arithmetic of Galois extensions of a totally real number field with Galois group in $\mathrm{GL}_2(\overline{\mathbb{F}}_p)$ that are unramified above p. Using the adelic q-

expansions of Hilbert modular forms of higher weight, these algorithms enable the explicit computation of Hilbert modular forms of any weight over $\mathbb{C}$ and the computation of Hilbert modular forms of parallel weight both over $\mathbb{C}$ and in positive characteristics. The main improvement to existing methods is that this algorithm can be applied in (partial) weight 1, which fills the gap left by standard computational methods. Moreover, the algorithm computes in all characteristics simultaneously. More precisely, it is proved that, under certain conditions in higher weight, the output of the algorithm for given level N and quadratic character ϵ includes a finite set of primes L such that all Hilbert modular forms of given parallel weight, level N and the quadratic character ϵ over $\mathbb{F}_p$ are liftable for all primes p outside the set L. In particular, testing primes in the set L enabled the computation of examples of non-liftable Hilbert modular forms of weight 1.

Sebastian Posur: Constructive Category Theory and Applications to Equivariant Sheaves
Betreuer: Mohamed Barakat (Siegen)
Zweitgutachter: Frank-Olaf Schreyer (Saarbrücken)
Juni 2017

Abstract: In this thesis we create a purely categorical framework for cohomology computations of G-equivariant coherent sheaves on projective space for a finite group G. For this, we develop three different sub-frameworks: First, we construct a skeletal tensor category $\mathrm{SRep}(G)$ equivalent to the representation category $\mathrm{Rep}(G)$ of G. Second, we design, in the context of an arbitrary abelian category, an algorithm for computing spectral sequences which is suitable for a direct computer implementation, i.e., it only uses categorical constructions provided by the axioms of an abelian category. Last, we describe how to internalize the exterior algebra E and its modules in a tensor category.

Combining our three sub-frameworks yields an algorithm for computing spectral sequences within the category of E-modules internal to $\mathrm{SRep}(G)$. Thanks to an equivariant version of the famous BGG-correspondence, we can use such an algorithm for computing cohomology groups of G-equivariant sheaves on projective space. Furthermore, this algorithm allows us to compute a new invariant called spectral cohomology table which in this thesis is proven to be stronger than the classical cohomology table.

Since our framework can be described in purely categorical language, a software project in GAP facilitating the implementation of abstract categories and categorical algorithms was born during the writing of this thesis: CAP (Categories, Algorithms, Programming). The categorical framework along with all algorithms presented in this thesis is implemented in CAP.

Stefan Toman: Radicals of Binomial Ideals and Commutative Thue Systems
Betreuer: Ernst W. Mayr (München)
Zweitgutachter: Bruno Buchberger (Linz))
July 2017

Abstract: Solving systems of polynomial equations is one of the most fundamental problems of computer algebra. The theory of Gröbner Bases allows for algorithmic solutions of many problems of polynomial ideals, but their computation takes an exponential amount of space in the worst case. Therefore, such computations are only feasible for subclasses of systems with small generators or ones that exhibit a special structure. One of the most interesting subclasses is the set of binomial ideals as they have more structure than general polynomial ideals, but still comprise the full complexity. We present a new algorithm for computing the radical of a binomial ideal which uses binomials as intermediate results of the computations only, but matches the running time of the best known algorithms. With this algorithm we can define radicals of commutative Thue systems.

Nils Amend: Restrictions of Reflection Arrangements and Asphericity
Betreuer: Gerhard Röhrle (Bochum)
Zweitgutachter: Ivan Marin (Amiens), Björn Schuster (Bochum)
July 2017

Abstract: In 1962, Fadell and Neuwirth showed that the configuration space of the braid arrangement is aspherical. Having generalized it to many real reflection groups, Brieskorn conjectured this for all Coxeter groups in 1971. In 1972, this was proved by Deligne in his seminal work showing that the complement of a complexified simplicial real arrangement is a $K(\pi;1)$ space. In the 1980s Nakamura, Orlik and Solomon established this property for all but six irreducible unitary reflection groups. These outstanding cases were resolved only recently in 2015, in stunning work utilizing Garside theory by Bessis. In this thesis we pursue the $K(\pi;1)$ property for restrictions of reflection arrangements. Our aim is to show that this property also holds for restrictions of the reflection arrangements of monomial complex reflection groups. Then only a finite list of restrictions of reflection arrangements remains for which the $K(\pi;1)$ property is undecided. This yields a new three-parameter family of $K(\pi;1)$ arrangements. In the case of the infinite families of Coxeter groups, these restrictions are again simplicial so that in this instance the result follows from Deligne's theorem.

Georg Loho: Combinatorics of Tropical Linear Programming
Betreuer: Michael Joswig (Berlin)
Zweitgutachter: Stéphane Gaubert (INRIA Saclay)
September 2017

Abstract: Der Schwerpunkt dieser Arbeit liegt auf der Untersuchung von Ungleichungssystemen, in denen nur die Operationen „min“ und „+“ vorkommen und die wir tropische, lineare Ungleichungssysteme nennen. Ein wichtiges algorithmisches Problem besteht darin, einen Vektor zu finden, der alle Ungleichungen erfüllt. Dieses Problem tritt in Verbindung mit Scheduling sowie Mean-Payoff-Spielen auf, und es ist ein Analogon zum klassischen Zulässigkeitsproblem für lineare Ungleichungssysteme. Wir untersuchen diese Systeme mittels Mengen bipartiter Graphen. Methoden aus der kombinatorischen Optimierung liefern eine Charakterisierung dieser bipariten Graphen, die wir Kovektor-Graphen nennen, mittels minimaler Matchings. Wir beginnen mit einer gründlichen Untersuchung von Polyedern, die auf natürliche Weise mit kürzesten Wegen in einem Graph assoziiert werden können. Das führt zu neuen Kovektorzerlegungen des tropischen projektiven Raums. Wir zeigen die Verbindung zwischen Ordnungs-Polytopen und dem Rand des tropischen projektiven Raums auf. Ferner lösen wir eine Vermutung von Develin und Yu (2007). Ein weiterer Ansatz zur Analyse von tropischen linearen Ungleichungssystemen entsteht

durch ihre Darstellung als Bilder linearer Ungleichungssysteme über Puiseux-Reihen unter einer Bewertungsabbildung. Wir führen den Teilkörper der Puiseux-Brüche ein, welcher im Gegensatz zu Puiseux-Reihen geeignet ist für Berechnungen am Computer, und wir stellen eine Verbindung zwischen linearen Ungleichungssystemen über dem Körper der Puiseux-Brüche mit linearen Ungleichungssystemen über den reellen Zahlen her. Verbindet man die Ideen zu klassischen Halbraum-Konfigurationen mit den Kovektor-Zerlegungen, so führt dies zu einer Abstraktion, welche durch vorzeichenbehaftete tropische Matroide ausgedrückt wird, die wiederum ein Analogon zu klassischen orientierten Matroiden darstellen. Dies ermöglicht es uns tropische lineare Ungleichungssysteme mit Mitteln polyedrischer Unterteilungen zu studieren. Während tropische lineare Ungleichungssysteme in Korrespondenz mit regulären Unterteilungen von einem Produkt zweier Simplizes stehen, lassen sich unsere Argumente direkt auch auf nicht-notwendigerweise reguläre Unterteilungen übertragen. Wir konzipieren einen Algorithmus, der die Zulässigkeit eines tropischen linearen Ungleichungssystems bestimmt und auch für die verallgemeinerte Struktur funktioniert, die wir anhand nicht-notwendigerweise regulärer Unterteilungen definiert haben. Für reguläre Unterteilungen ist die Komplexität durch die Einträge eines minimalen ganzzahligen Vektors in einem Kegel des Sekundärfächers des Produkts zweier Simplizes beschränkt.

Habilitationen in der Computeralgebra

Tobias Rossmann (Bielefeld):
Zeta Functions of Groups, Algebras, and Modules
Betreuer: Christopher Voll (Bielefeld)
Gutachter: Andrei Jaikin-Zapirain (Madrid), Benjamin Klopsch (Düsseldorf), Cheryl Praeger (Perth).
Juni 2017

Abstract: The six research articles collected in this thesis contain contributions to the theory of zeta functions of groups, algebras, and modules. The zeta functions under considerations arise e.g. from the enumeration of subgroups, subalgebras, submodules, or representations of suitable algebraic structures. The majority of the work is devoted to local and topological zeta functions. The former naturally occur as Euler factors of Dirichlet generating functions and the latter arise as suitable limits of the former.

The first four chapters are devoted to the development of practical methods for explicitly computing various types of local and topological zeta functions. These methods are practical, as demonstrated by an implementation in the software package Zeta. An overview of Chapters 1 to 4 (including an introduction to the area) is included as Chapter 0. Apart from providing techniques for their explicit computation (and numerous applications of these), Chapters 1 to 4 contain several theoretical results of independent interest, e.g. a proof that topological representation zeta functions have degree zero.

Chapter 5 constitutes a thorough analysis of a special class of submodule zeta functions, namely those enumerating submodules invariant under a single matrix with entries in the ring of integers of a number field. Applications include a proof that non-abelian nilpotent Lie lattices of maximal class have quadratic ideal growth.

The main result of Chapter 6 establishes that in many cases of interest, the behaviour of local zeta functions under variation of the prime within a set of density 1 completely determines these functions for almost all primes and, moreover, also determines their behaviour under local base extensions.

Alexander D. Rahm (Luxembourg/Paris):
Torsion Subcomplex Reduction
Président de Jury: Alain Valette (Neuchâtel)
Jury: Nicolas Bergeron (Paris), Mladen Dimitrov (Lille), Paul E. Gunnells (Amherst, reviewer), Günter Harder (Bonn), Hans-Werner Henn (Strasbourg, reviewer), David Kohel (Aix-Marseille), Ralf Köhl (Giessen).
Juni 2017

Abstract: This thesis describes works involving a technique called Torsion Subcomplex Reduction (TSR), which was developed by the author for computing torsion in the cohomology of discrete groups acting on suitable cell complexes. TSR enables one to skip machine computations on cell complexes, and to access directly the reduced torsion subcomplexes, which yields results on the cohomology of matrix groups in terms of formulas. TSR has already yielded general formulas for the cohomology of the tetrahedral Coxeter groups as well as, at odd torsion, of SL_2 groups over arbitrary number rings (in joint work of M. Wendt and the author). The latter formulas have allowed Wendt and the author to refine the Quillen conjecture. Furthermore, progress has been made to adapt TSR to Bredon homology computations. In particular for the Bianchi groups, yielding their equivariant K-homology, and, by the Baum-Connes assembly map, the K-theory of their reduced C*-algebras, which would be very hard to compute directly. As a side application, TSR has allowed the author to provide dimension formulas for the Chen–Ruan orbifold cohomology of the complexified Bianchi orbifolds, and to prove (jointly with F. Perroni) Ruan's crepant resolution conjecture for all complexified Bianchi orbifolds.

3C in G Workshop on Computational Algebra

King's College. Cambridge, Großbritannien, 18.04. – 21.04.2017

`http://people.ds.cam.ac.uk/amt69/3CinGApril17/3CinGApril17.html`

The workshop on Computational Algebra brought together experienced researchers and young mathematicians under the broad overall topic of "Classification, Computation, and Construction: New Methods in Geometry" of the underlying 5-year EPSRC-funded program 3CinG. After a first day devoted to a SINGULAR tutorial session in the morning and short talks by PhD-students in the afternoon, the conference touched on a plethora of aspects of geometry as e.g. Discrete Morse theory, Tropical Geometry, Surfaces of general type, but also Complexity and the role of software and its development in research in geometry.

A common point and connecting element to the 20 talks and 40 participants was the algorithmic/constructive point of view on the respective geometrical questions and in some cases also the revelation of finding the same algorithmic task in different guises.

Anne Frühbis-Krüger (Hannover)

7th Int. Conference on Algebraic Informatics: CAI 2017

Kalamata, Griechenland, 25.06. – 28.06.2017

`www.cargo.wlu.ca/CAI2017`

Die siebte Ausgabe der internationalen Konferenz "Algebraic Methods in Informatics" fand diesmal in Kalamata (Griechenland) statt und hatte 5 Tracks. Dabei wurde "Track 3: Computer Algebra" von Rafael Sendra (Alcala/Spanien) und Franz Winkler (Linz/Österreich) geleitet und trug den Hauptvortrag "Computing Difference Algebraic Relations Among Solutions of Linear Differential Equations" von Michael Wibmer (Univ. of Pennsylvania/USA) zum Konferenzgeschehen bei.

Eine besondere Anwendung mathematischer Algorithmen zeigte Lila Kari (Univ. of Waterloo/Kanada) mit ihrem Hauptvortrag "Was Pegasus a Mammal or a Bird? How to Measure and Visualize Species' Relatedness" auf, in dem verschiedene mathematische Algorithmen kombiniert wurden, um mit Hilfe des Genoms die Verwandtschaft von Tierarten zu klassifizieren und so einen Teil der Zoologie möglichst exakt berechenbar zu machen. Einen weiteren prominenten Beitrag aus der Computeralgebra lieferte Stephen Watt (Univ. of Waterloo/Kanada) mit seinem Vortrag "Specialization of Symbolic Polynomials".

Die Konferenz war von Ilias Kotsireas (Univ. of Waterloo/Kanada) als Hauptorganisator hervorragend geplant und durchgeführt. Das wissenschaftliche Programm wurde durch einen interessanten Ausflug ins alte Messene, einem erst in den letzten Jahren intensiv ausgegrabenen Zentrum der griechischen Kultur, ergänzt und fand im hervorragend ausgestatteten Hotel "Elite City Resort" direkt am Mittelmeerstrand statt.

Martin Kreuzer (Passau)

Workshop Arithmetic Geometry and Computer Algebra

Oldenburg, 29.06. – 01.07.2017

`www.uol.de/math/wagca`

Vom 29.6. bis zum 1.7.2017 fand in Oldenburg ein Workshop zum Thema „Arithmetic Geometry and Computer Algebra" statt. Organisiert wurde dieser von Florian Heß (Oldenburg), Andreas Stein (Oldenburg) und Steffen Müller (Oldenburg, jetzt Groningen).

Ziel des Workshops war es, Entwickler von Computeralgebra für den Einsatz in der arithmetischen Geometrie mit Anwendern zusammenzubringen. Zu diesem Zweck wurden ca. 30 nationale und internationale Experten eingeladen, insgesamt gab es ca. 50 Teilnehmer.

Bei der Erstellung des Programms wurde darauf geachtet, neben einem abwechslungsreichen Vortragsprogramm auch genügend Zeit für Diskussionen und Kooperationen zu reservieren. Dies wurde von den Teilnehmern reichlich genutzt. Thematischer Fokus der meisten Vorträge waren Algorithmen zur expliziten Untersuchung von algebraischen Kurven, abelschen Varietäten und Modulformen. Außerdem wurde den Entwicklern der gängigsten in der arithmetischen Geometrie verwendeten Computeralgebrasysteme (Pari/GP, Magma, Sage sowie das neue System Nemo) die Möglichkeit gegeben, eine Einführung in die Verwendung dieser Systeme sowie neuere Funktionalität derselben zu präsentieren. Nachwuchswissenschaftler und Doktoranden konnten ihre Ergebnisse auf einer Postersession vorstellen.

Der Workshop wurde finanziell von der DFG, der Universitätsgesellschaft Oldenburg sowie der Fachgruppe Computeralgebra unterstützt.

Steffen Müller (Groningen)

Die Teilnehmer in Oldenburg. Foto: S. Schlüters

ISSAC 2017

Kaiserslautern, 25.07. – 28.07.2017

`www.issac-symposium.org/2017`

Die ISSAC (Langtitel: International Symposium on Symbolic and Algebraic Computation) fand in diesem Jahr wieder einmal in Deutschland statt. Der Austragungsort war Kaiserslautern, sicherlich eine der Hochburgen der Computeralgebra in Deutschland. Die ISSAC ist eine der wichtigsten internationalen Konferenzreihen über algebraisches und symbolisches Rechnen (anders gesagt: über Computeralgebra), die seit ihren Ursprüngen eine enge Verbindung zur Fachgruppe Computeralgebra pflegt.

Mit 129 Teilnehmerinnen und Teilnehmern war die ISSAC auch in diesem Jahr wieder gut besucht. Die lokale Leitung lag diesmal in den Händen von Wolfram Decker und von Claus Fieker, der auch Mitglied der Fachgruppenleitung ist. Als General Chair fungierte Chee K. Yap (New York), und das Programmkomitee, bei dem alle Entscheidungen über die Vorträge liegen, wurde von Mohab Safey El Din (Paris) geleitet. Nach dem durch das Programmkomitee organisierte Begutachtungsprozess wurden 55 der eingereichten Artikel akzeptiert. Neben diesen — jeweils in zwei Sektionen abgehaltenen — eingereichten Vorträgen gab es auch auf dieser ISSAC drei eingeladene Hauptvorträge und, als Auftakt zur Konferenz, drei Tutorials. Daneben wurden zwölf Poster und sechs Software-Demos präsentiert. Für die Namen aller Beitragender sei auf die wie immer gut organisierte Homepage der Konferenz verwiesen.

Auf dem ISSAC Business Meeting wurden neben einigen Berichtspunkten der Austragungsort für das Jahr 2019 und ein neues Mitglied im Steering Committee bestimmt, jeweils durch Wahlen, bei denen wie immer alle Teilnehmerinnen und Teilnehmer der Konferenz stimmberechtigt waren. Beim Konferenzort fiel die Wahl allerdings nicht schwer, da die einzige Bewerbung aus Beijing kam. Für die Besetzung einer Vakanz im Steering Committee gab es mehrere Kandidaten, und die Wahl fiel auf Thomas Sturm (Nancy).

Beim Konferenz-Bankett

Auch diesmal gab es eine ganze Reihe von Preisen, die alle bei dem Bankett vergeben wurden: Den Distinguished Paper Award, den Distinguished Student Author Award, den Distinguished Poster Award und den Distinguished Software Demonstration Award. An den Preisen für das beste Poster und für die beste Software-Demo war die Fachgruppe direkt beteiligt, indem sie das Preisgeld von je 250 Euro zur Verfügung stellte (außerdem stiftete Maplesoft eine Maple-Lizenz). Für die Ermittlung der Preisträger dieser beiden Preise wurden jeweils spezielle Komitees gebildet: Ganz ohne Mühe ist auch die Vergabe von Preisen nicht gut zu machen, auch wenn das Gewinnen die weit schwerere Aufgabe bleibt.

Das prämierte Poster ist:

Autoren: Dean Kieffer und Luca De Feo.
Titel: Isogeny-based cryptography in Julia/Nemo: a case study.

Als beste Software-Demo wurde ausgezeichnet:

Autoren: Sardar Haque, Xin Li, Farnam Mansouri, Davood Mohajerani, Marc Moreno Maza und Wei Pan.
Titel: CUMODP: A CUDA Library for Modular Polynomial Computation.

Zum Schluss sei noch erwähnt, dass das Konferenzbankett — ganz den Erwartungen an die Fußball-Stadt Kaiserslautern entsprechend — im Restaurant des Fritz-Walter Stadiums stattfand.

Die nächste ISSAC-Konferenz findet vom 16. bis 19. Juli 2018 an der City University of New York statt.

Gregor Kemper (München)

GAP Days Fall 2017

Siegen, 30.08. – 08.09.2017

`gapdays.de/gapdays2017-fall`

GAP Days are meetings where developers and users with GAP programming experience are invited to influence the future development of GAP. The meetings are also used to advertise recent developments in core GAP and packages via short talks. However, the focus of the meeting is on code development. As enough GAP experts will be around for technical support, the meetings offer good opportunities for people to work on their own packages.

The GAP Days Fall 2017 were split in two parts:

During the first week (August 30 - September 1), a workshop with several talks took place. Most of the talks reported about new developments in GAP including the core system, new packages, and upgraded functionality of older packages (`http://gapdays.de/gapdays2017-fall/program/`). Our former colleague Max Neunhöffer from the company ArangoDB gave an invited talk about their open source database (`https://www.arangodb.com`), which some of the participants are already using via a new GAP interface.

In the second week (September 4-8), we ran a coding sprint. Topics being worked on included:

- preparing the GAP 4.9 release (planned for November 2017)
- continued work on the MatrixObj project, a modern, unified matrix interface for GAP
- work on several packages (`MatricesForHomalg`, `cvec`, `FinInG`) to comply with the new matrix interface
- integration of GAP and Julia (part of the OSCAR project)
- libgap (making GAP usable as a shared library, e.g. for SageMath and OSCAR)
- splitting the libraries of small, transitive and primitive groups into proper GAP packages
- releases of several GAP packages such as `numericalsgps`, `sgpviz`, `intpic`, `AutoDoc`, `NormalizInterface`, ...
- new GAP package `InfiniteLists` which was extracted from the package `ComplexesForCAP` providing lazy and infinite complexes for the `homalg`/CAP project
- lots of bugs were fixed, pull requests reviewed and merged.

Mohamed Barakat (Siegen), Max Horn (Gießen)

The 19th International Workshop on Computer Algebra and Scientific Computing (CASC 2017)

Beijing, China, 18.09. – 21.09.2017

`www.casc.cs.uni-bonn.de/2017`

This year the 19th CASC conference was held in Beijing (China). The organizing committee at the Key Laboratory of Mathematics Mechanization, Chinese Academy of Sciences, headed by Jin-San Cheng, did a superb job in doing all of the local arrangements. All talks took place in the new South Building of AMSS, with lunch and dinner in a nearby hotel, in which the majority of the participants also had their rooms.

Banner of CASC 2017 at conference location, the new South Building of AMSS

For the first time the conference started with a day with tutorials on Monday. In two morning sessions Jan Verschelde (Chicago) spoke on "Numerical Algebraic Geometry in the Cloud". In the afternoon Wen-Shin Lee (Antwerp) gave an overview on "Sparse interpolation and its connection to Padé approximation, signal processing and tensor decomposition".

Tuesday started with one of the two invited talks on "Linear Differential Systems with Infinite Power Series Coefficients" by Sergei Abramov. Computations of integer sequences and of integer points of a Polyhedron were the topics of the morning sessions. Special homotopy continuation methods, the computation of witness points on a penalty function based critical point approach and full rank approximations of real algebraic sets and its applications were presented in the first afternoon session. The second afternoon session started with another classical topic of CASC conferences: Using computer algebra for the investigation of various mathematical and applied topics related to ordinary differential equations (ODEs). In the business meeting next years location of CASC, Lille, was presented. Brief presentations of potential subsequent locations were given: Moscow and the campus of the King Abdullah University of Science and Technology (KAUST) at the Red Sea near Jeddah in Saudi Arabia.

The second invited talk "On Simple Multiple Zeros of Polynomial Systems" by Lihong Zhi started the scientific program on Wednesday morning. Contributed talk in the area of sparse interpolation, which was previously addressed on the level of a tutorial, gave some state of the art advances.

The tradition of CASC of having exciting excursions was instantiated by an excursion to the nearby Summer Palace on Wednesday afternoon. The conference dinner was held at a roasted duck restaurant giving insight into the original way of preparing a "Beijing duck" meal.

Excursion to the Summer Palace

Four sessions presenting 13 contributed papers were held on Thursday. Topics included the applications of symbolic and symbolic-numeric computations for investigating and solving partial differential equations (PDEs) and ODEs in mathematical physics and fluid mechanics, focussing on, for example, the symbolic-numeric integration of the dynamical Cosserat partial differential equations describing the mechanical behavior of elastic rods; the symbolic-numeric solution with Maple of the parametric self-adjoint 2D elliptic boundary-value problem with the aid of a high-accuracy finite element method; and a new symbolic-numeric preconditioned solver for incompressible Navier–Stokes equations using the integral form of collocation equations. Applications of CASs in mechanics, physics, and biology were represented by the following themes: investigation of the asymptotic stability of a satellite with a gravitational stabilizer; satellite dynamics subject to damping torques; Mathematica-based analysis of the relative equilibria stability in a problem of celestial mechanics; stationary motions of the generalized Kowalewski gyrostat and their stability; and symbolic versus numerical computation and visualization of parameter regions for multistationarity of biological networks.

A one day workshop on differential algebra and polynomial system solving on Friday, September 22 followed the conference, in which several participants presented other recent work of theirs (e. g. the work on "Algorithmic Verification on Linearizability of Ordinary Differential Equations" that was recently given the best paper award of ISSAC 2017).

Andreas Weber (Bonn)

Konferenzfoto, CASC 2017

Hinweise auf Konferenzen

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

Douala, Kamerun, 06.10. – 13.10.2017

www.aims-volkswagen-workshops.org

The first edition of the AIMS-Volkswagen Stiftung Workshop on Introduction to Computer Algebra and Applications is aimed globally at promoting capacity building in terms of research and training in computer algebra and applications, discussions and development of new ideas, development and enhancement of networking including south-south cooperation. This workshop is funded within the framework of the Volkswagen Foundation Funding Initiative "Symposia and Summer Schools".

MACIS 2017

Wien, Österreich, 15.11. – 17.11.2017

www.macis2017.sba-research.org

MACIS is a series of biennial conferences focusing on research in mathematical and computational aspects of computing and information science. It is broadly concerned with algorithms, their complexity and their embedding in larger logical systems. At the algorithmic level, there is the rich interplay along the Numerical/Algebraic/Geometric/Topological axes. At the logical level, there are issues of data organization, interpretation and associated tools. These issues often arise in scientific and engineering computation where we need experimental and case studies to validate or enrich the theory. MACIS is interested in outstanding and emerging problems in all these areas.

Nikolaus conference 2017

Aachen, 08.12. – 09.12.2017

www.math.rwth-aachen.de

The Nikolaus conference is an annual meeting at Lehrstuhl D für Mathematik (RWTH Aachen). The main aim is to bring together people who have recently finished a thesis on a topic in group or representation theory and some established people in this area. Particularly welcome are reports on projects with a computational aspect.

GDMV 2018

Paderborn, 05.03. – 09.03.2018

www.gdmv2018.de

Im Jahr 2018 richtet Paderborn die gemeinsame Jahrestagung der Gesellschaft für Didaktik der Mathematik (GDM) und der Deutschen Mathematiker-Vereinigung (DMV) aus. Zuletzt gab es diese Kombination 2010 in München. Neben den beiden Jahrestagungen werden Themen auf der Schnittstelle in gemeinsamen Hauptvorträgen, sowie in speziellen Schnittstellensektionen adressiert. Beispielsweise sind dies Themen wie Übergang Schule/Hochschule, Hochschuldidaktik sowie die Mathematikausbildung von Lehrkräften. Eine der Sektionen innerhalb der DMV wird Diskrete Mathematik und Computeralgebra sein, welche von Florian Heß (Oldenburg) und Alexander Pott (Magdeburg) geleitet wird. Im Schnittstellenbereich wird ein Minisymposium "CAS in der Hochschullehre - Ein Blick in die Praxis" stattfinden, welches von Anne Frühbis-Krüger (Hannover), Gregor Kemper (München), Wolfram Koepf (Kassel) und Michael Liebendörfer (Hannover) organisiert wird (siehe Seite 12).

Die Anmeldung und die Einreichung von Vorträgen startet am 1.10.2017. Die Deadline für Beiträge in Minisymposien ist der 10.11.2017 und für sonstige Vorträge ist dies der 1.12.2017. Alle Informationen befinden sich auf der Tagungshomepage.

ACA 2018

Santiago de Compostela, Spanien, 18.06. – 22.06.2018

www.usc.es/regaca/aca2018

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

ISSAC 2018

New York City, USA, 16.07. – 19.07.2018

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 43nd 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

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.

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:
Privatanschrift: Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
Dienstanschrift: Firma/Institut:	Abteilung:
Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
E-Mail:	
Gewünschte Postanschrift: ❑ Privatanschrift ❑ 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.

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 hinausgeben (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, per Post oder Fax genügt.

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

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

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

Dieses Formular gibt es auch im Internet: http://fachgruppe-computeralgebra.de/anmeldeformular

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`
`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`
`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`
`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`
`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`
`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`
`www.mathebeimueller.de`

Fachexperte Industrie:
Prof. Dr. Christoph Thiel
Fachhochschule Bielefeld
Fachbereich Campus Minden
Artilleriestr. 9, 32427 Minden
0571-8385-258
`christoph.thiel@fh-bielefeld.de`
`https://www.fh-bielefeld.de/fb2/personen/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`
`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`
`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`
`http://wwwth.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`
`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`
`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`
`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`
`http://www.math.rwth-aachen.de/~Eva.Zerz/`