Oktober 2018

Computeralgebra Rundbrief

> Ausgabe 63

www.fachgruppe-computeralgebra.de

Maple™

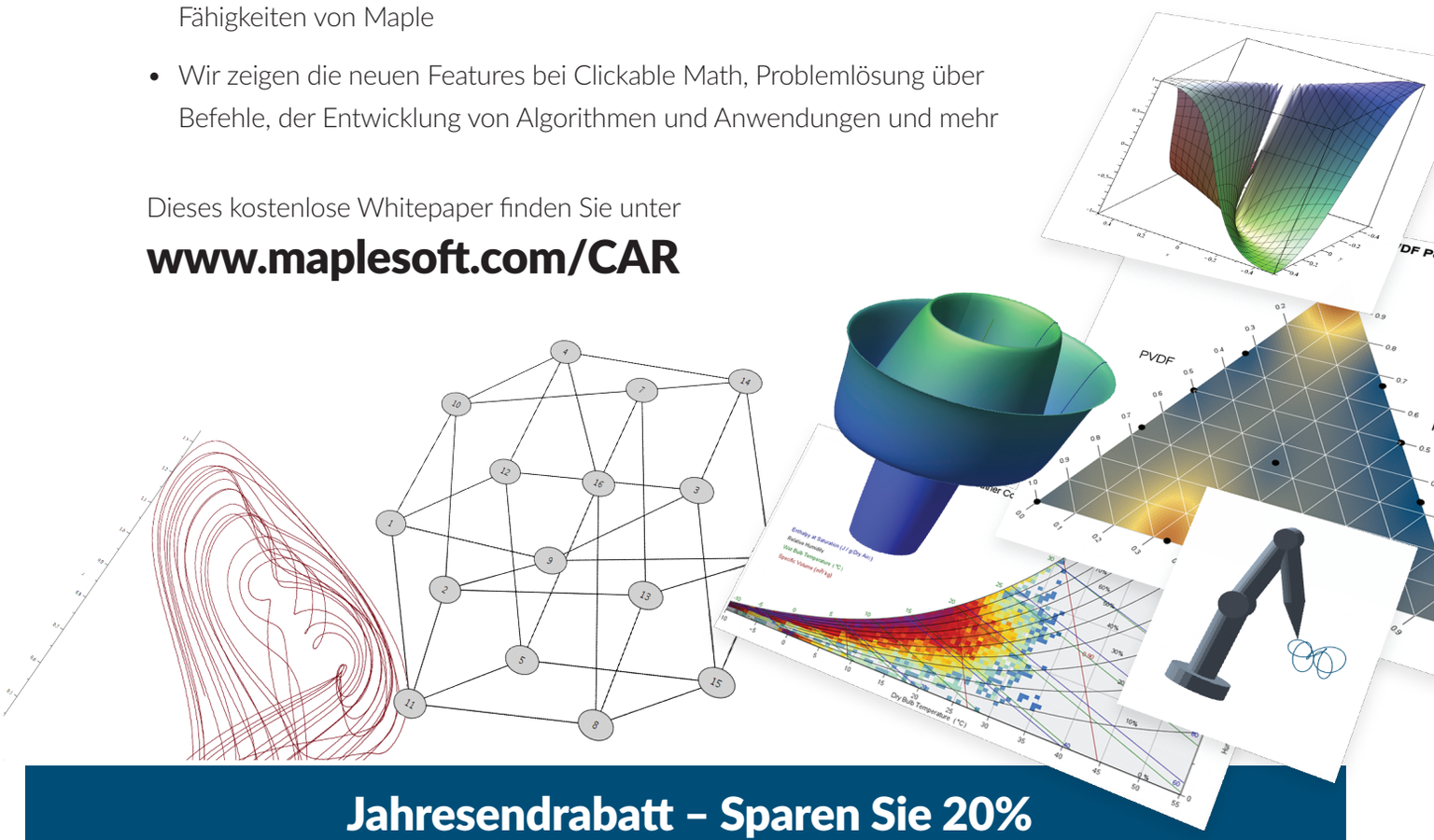
Sie nutzen Maple 2018 noch nicht?

Sehen Sie, was Sie bisher versäumt haben

- Entdecken Sie einige der Features, die bei Maple in den letzten drei wichtigen Upgrades hinzugekommen sind
- Erfahren Sie die Tiefe und Breite der weiterentwickelten mathematischen Fähigkeiten von Maple
- Wir zeigen die neuen Features bei Clickable Math, Problemlösung über Befehle, der Entwicklung von Algorithmen und Anwendungen und mehr

Dieses kostenlose Whitepaper finden Sie unter

www.maplesoft.com/CAR



Jahresendrabatt – Sparen Sie 20%

Maplesoft bietet einen Preisnachlass von **20% auf Lizenzen**, die bis zum 31.12.2018 gekauft werden.

Für ein Angebot wenden Sie sich bitte direkt an unsere deutsche Niederlassung unter **germany@maplesoft.com** oder telefonisch unter **+49 980919-30**

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

Computeralgebra-Rundbrief

Nr. 63 Oktober 2018

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,

auch diesmal konnten wir wieder eine ganze Reihe von interessanten Beiträgen für den Rundbrief gewinnen und mussten aus Platzgründen sogar einen Artikel auf die nächste Ausgabe vertagen. Doch ehe wir uns den Themen zuwenden können, möchten wir noch zwei Punkte in eigener Sache vermelden.

Im Frühjahr 2019 veranstaltet die Fachgruppe Computeralgebra wieder ihre zweijährliche Tagung. Eine ausführliche Ankündigung wird dazu im kommenden Rundbrief erscheinen, doch möchten wir Sie jetzt schon bitten, sich Zeit und Ort vorzumerken:

16. – 18. Mai 2019 in Kassel.

Da die Tagung wie immer auch reichlich Gelegenheit zu Vorträgen durch den wissenschaftlichen Nachwuchs bieten wird, wäre es schön, wenn Sie auch interessierte jüngere Kolleginnen und Kollegen darauf hinweisen könnten.

Die andere Nachricht in eigener Sache betrifft Christoph Thiel, unseren Fachexperten Industrie. Er scheidet demnächst aus der Fachgruppenleitung aus. Für die geleistete Arbeit und die gute Zusammenarbeit möchten wir ihm an dieser Stelle danken. Sein Ausscheiden reißt eine Lücke, die wir nun schnell schließen sollten. Gerade im Bereich 'Computeralgebra und Industrie' sind – nicht zuletzt wegen der oft engen Zusammenarbeit mit Unternehmen – Experten mit ausreichend zeitlichen Kapazitäten (und kreativem Freiraum) nicht ganz leicht zu finden.

Kommen wir nun zum vorliegenden Rundbrief zurück, der mit einem Rückblick über das 20-jährige Bestehen des Projekts SymbolicData *beginnt. Auf diesen folgt dann ein Artikel zum hochaktuellen Thema der Post-Quantum Kryptographie: Welche Algorithmen werden noch zur Verfügung stehen, wenn die bisher gängigen Kryptosysteme durch Quantencomputer gebrochen werden können?*
Die Rubrik 'Neues über Systeme' berichtet danach über ein neues System aus dem Bereich kommutative Algebra, ehe dann ein Artikel von Jörg Meyer, unserem früheren Fachexperten 'Computeralgebra in der Schule' (2008-11), der uns bis heute die Treue hält und zu unseren fleißigsten Autoren zählt, geometrische Überlegungen mit Anwendungen der Computeralgebra verknüpft.

Zu guter Letzt möchten wir noch auf einen Punkt in der Berichten von Konferenzen hinweisen: den Bericht auf Seite 34 über die ISSAC, eine der wichtigsten und größten internationalen Konferenzen über Computeralgebra, die in diesem Jahr in New York stattfand. Wir heben sie hier besonders hervor, da die Fachgruppe ihr auf zwei Arten besonders verbunden ist: durch unsere häufige institutionelle Mitgliedschaft im Steering Committee (so auch in diesem Jahr) und durch einige Preise, die die Fachgruppe auf der ISSAC stiftet.

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

Gregor Kemper *Anne Frühbis-Krüger*

20 Years SYMBOLICDATA

H.-G. Gräbe
(Leipzig University)

graebe@informatik.uni-leipzig.de

Introduction

At the OSCAR meeting[1] in Leipzig in December 2017 Bernd Sturmfels promoted a discussion "What to do with Big Old Data?"[2]. He addressed "a key problem in the development of an open source computer algebra system – the design of mechanisms and formats for dealing with big old data", i.e. "the output of a mathematical computation that is much larger than a few lines, and is intended for storage in a repository, or for further processing by a different program", and presented a list of 12 papers. In these papers authors from different areas of computational mathematics describe efforts, concepts and data stores to resolve for themselves or for a small computational mathematics subcommunity such a problem. As central goal each of these projects tries to establish from the very scratch a research infrastructure capable to present, access, inspect, exchange and maintain structured research data in both a powerful and sustainable way.

A similar situation was discussed 20 years ago at the *Special Session on Benchmarking*[3] at the 1998 ISSAC conference in Rostock. Heinz Kredel presented in his famous talk[4] also 12 examples of web resources that were set up and maintained by different people to keep track of the development concerning several of the challenges discussed at that time (Wester's CAS test suite, von zur Gathen's polynomial factorization challenge, Montgomery's number field sieve factorizations or the SATLIB benchmark suite – to name some of them). Most of the links provided by Heinz Kredel are outdated for a long time, of course. So, didn't the world change during the past 20 years?

This paper presents a comprehensive survey of the most important motivations, concepts, steps, efforts and practical achievements of the SYMBOLICDATA Project to contribute to the formation of a reliable and sustainable research infrastructure in the area of Computer Algebra. For a more detailed explanation of different aspects of the motivations, concepts and achievements of the SYMBOLICDATA Project we refer to our publication list[5].

We start with a short discussion about the importance of research data in science in general, touch the distinction between core concerns and cross cutting concerns in modern software engineering and discuss consequences of the emerging role of semantic technologies for research infrastructures as a great difference between the situation today and 20 years ago.

A cornerstone of such concepts is the increasing importance of metadata that have to accompany research data to organize their maintainability, searchability and exchangeability. Within SYMBOLICDATA we developed the concept of *semantic-aware fingerprints* to emphasize that the definition of such metadata is a process of social coordination between the "suppliers" and the "customers" of research data to agree upon a *domain ontology* and not merely a question of just technically presenting the data.

Our experience indicates that the importance to organize the formation of a research infrastructure as a social process is underestimated. Whereas in small subcommunities of Computer Algebra research infrastructures usually grow and develop in a "natural way", it requires great efforts to organize such a social process on the level of inter-subcommunity communication within Computer Algebra. Such inter- and infradisciplinary engagement is little recognized as beneficial for the scientific career of productive young researcher. We developed the concept of a *Computer Algebra Social Network* (CASN) [4] based on modern semantic technologies and maintained a prototypical distributed implementation as a showcase over several years.

In the last two years we concentrated our resources

[1] https://www.mis.mpg.de/calendar/conferences/2017/oscar2017.html.
[2] https://www.mis.mpg.de/fileadmin/pdf/slides_oscar2017_3198.pdf.
[3] http://bwcloud-108-017.bwcloud.uni-mannheim.de/cafgbench.html.
[4] http://bwcloud-108-017.bwcloud.uni-mannheim.de/cabench/issac98/.
[5] http://symbolicdata.github.io/Publications.

on GitHub and terminated several unstaffed activities. In particular we moved the wiki to GitHub pages, stopped to update the CASN data base and canceled our announcement mailing list as a probably outdated way of communication. It is due to the next generation to evaluate the SYMBOLICDATA heritage and update the parts that are worth to be continued.

Research Data in Science

Research data plays an important role within science in at least four dimensions: 1) artifacts as problem sources (in particular digitized artifacts in the humanities), 2) benchmark examples as well established challenges for different problem classes, 3) raw output of research to be analyzed and evaluated within the scientific community and 4) publications and other consolidated scientific output as part of a common scientific *social* infrastructure.

Traditionally, the Computer Algebra community focuses on dimension 4, in particular on algorithms, implementations and software. The swMATH project [15] and several big national or EU funded projects as PoSSo [13], FRISCO [2], the SPP 1489, OpenDreamKit [10] or OSCAR [11] aimed and aim at contributing to the formation of a common Computer Algebra research infrastructure in that direction.

At the 1998 ISSAC conference in Rostock, at the end of the projects PoSSo and FRISCO, dimension 2 and 3 started to play a more important role. Gert-Martin Greuel reported in his invited lecture *Computeralgebra and Algebraic Geometry, Achievements and Perspectives* in particular about the progress in the Gröbner business as one of the core algorithmic building blocks in advanced polynomial systems solving and algebraic geometry. In 1993, the FGLM algorithm was published. But there were rumors about new algorithmic ideas of a young guy in Paris, and Greuel's Singular group in particular was interested in evaluating these ideas. Unfortunately, after the end of the PoSSo project there was no established reference for the famous *PoSSo test suite* and different printed versions with a number of misprints were in circulation. It started a disputation about the timings of examples, since it was neither clear what version of the PoSSo examples was referenced nor what "timing" really means (clock time, wall time, processor time etc.). Heinz Kredel proposed to start a *Computer Algebra Benchmarking Initiative* and invited for a *Special Session on Benchmarking* that took place on August 15, 1998, 18:00 in room 110 in the main building of the Rostock University. "The initiative will discuss, develop, define, collect all facets of this challenging problem. It should analyse and develop test suites but also define standard examples for the various topics of computer algebra where algorithm and systems developers can test their newly developed and improved methods. Furthermore, all kinds of test examples should be collected and consolidated. "[6] Heinz Kredel presented a *Computer Algebra Benchmarks Collection from July 1998* and shortly explained the results of the *CASBENCH Computer Algebra Benchmarks* activities, started in 1995 by the German Fachgruppe. The CASBENCH setup[7] was inspired by the Parkbench activities[8] on *Public International Benchmarks for Parallel Computers* (nowadays known as the annual HPC challenge award competition[9]) but got much less support from the community. Unfortunately, even within the ISSAC Special Session on Benchmarking the community could not agree upon a further roadmap or even a commonly accepted process or dedicated resources to advance that matter. Once more the diversity of Computer Algebra challenges prevented awareness of common interests in promoting the development of a common research infrastructure.

The 1998 Special Session on Benchmarking was the starting point of the SYMBOLICDATA Project. Olaf Bachmann and me developed and implemented with great support by other members of the Singular team the basic concepts and a first version of a benchmarking environment in the area of polynomial systems solving, called POLYDATA and later on SYMBOLICDATA. We collected a considerable number of the benchmark examples used at that time for testing polynomial systems solvers (in particular the PoSSo test suite), made it publicly and reliably available in a digital exchange format and developed a standardized environment based on GNU make and GNU time to run, time and monitor test computations on such examples using different solvers. But the main conceptual goal of SYMBOLICDATA was a nontechnical one – to develop a research infrastructure that is independent of (permanent) project funding but operates based on overheads of its users. This approach was inspired by the rich experience of the Open Culture movement "business models" to run infrastructures. It was an early attempt to emphasize the advantage of an explicitly elaborated concept of a community-based solution to the "tragedy of the commons" within the Computer Algebra community and to apply such a concept to run as part of its research infrastructure.

Nowadays the awareness of the importance of digital research infrastructure increased both in the scientific communities and also in research politics. The development of research infrastructures coordinated by the European Research Infrastructure Consortium (ERIC) plays a crucial role in the EU funding program HORIZON 2020. *OpenDreamKit* is such a project within Computer Algebra. Unfortunately it concentrates on (technical) interoperability rather than the research data problems discussed above. "OpenDreamKit is a project that brings together a range of projects and associate software to create and strengthen virtual research environments. The most widely used research environment

[6] `http://bwcloud-108-017.bwcloud.uni-mannheim.de/cafgbench.html`.
[7] `http://bwcloud-108-017.bwcloud.uni-mannheim.de/cabench/casbench.html`.
[8] `http://www.netlib.org/parkbench/html/`.
[9] `http://www.hpcchallenge.org/`.

is the Jupyter Notebook from which computational research and data processing can be directed. The OpenDreamKit project provides interfaces to well established research codes and tools so that they can be used seamlessly and combined from within a Jupyter Notebook."

A different approach is pursued by the *Leibniz Network MMS* on modelling and simulation[10] with great focus on research data and strong cooperation with TIB Hannover[11] and the upcoming community of Research Software Engineers[12].

Several activities concerning research data are on the way in the "big scene": *FORCE-11*[13] and the *Research Data Alliance*[14] are international interdisciplinary initiatives to promote and feature research data management and the development of digitally supported research infrastructures. Much is on the way also within mathematics – international initiatives were started to digitally organize the mathematical heritage as a whole (WDML[15], IMKT[16]) and of mathematical software in particular (swMATH [15], the INRIA Software Code Archive[17]), see [7] for a survey. SYMBOLICDATA provides 20 years of experience, in particular with semantic technologies, for these broader initiatives.

Testing and Benchmarking in Modern Software Engineering

Testing and benchmarking is a common task in software engineering. Modern software development concepts for enterprise middleware provide architectural and technical support (git workflows, virtualization tools) for agile approaches as continuous integration, continuous development and continuous deployment of modular software pieces in distributed environments. All this makes software development more complex and requires a good theoretical understanding of the corresponding architectural concepts. To maintain such diverging goals within a single software, modern software engineering distinguishes between core and cross cutting concerns. *Core concerns* define the main goal of the implementation that is mission critical or requires special domain specific knowledge, insight and experience. *Cross cutting concerns* (logging, testing, profiling, data management, security concerns, inter process communication etc.) are implemented using well established generic approaches. This eases both the maintenance of the software and the training of the developers. Frameworks as Spring or EJB realize the concept of *context aware programming*. The core concerns are implemented in a runtime that is embedded into another generic runtime – the *context* – that provides cross cutting objects to be injected as dependencies into the core runtime. Such an approach allows to concentrate combined long time efforts on the development of the commonly used context environment and to realize short term core business in a more efficient way.

Such approaches are nowadays common also within the big Computer Algebra software projects SageMath, OpenDreamKit and OSCAR. The SYMBOLICDATA benchmark activities designed by Olaf Bachmann 20 years ago anticipated several such concepts still unknown at that time. Since 2013 this framework was enhanced, advanced and consequently used in the working group of Viktor Levandovskyy at RWTH Aachen, This *SDEval*[18] *Testing and Benchmarking Environment* provides an easy way to generate executable code for benchmarks of computer algebra systems (like Singular, Magma etc.) on SYMBOLICDATA benchmark data and a framework for trustfully reproducing computation results from current research papers. SDEval is intensively used in particular in projects from TRR 195 for, e.g., finitely presented associative algebras. Its current developer team includes Karim Abou Zeid and Viktor Levandovskyy; the beginnings were laid by Albert Heinle and Benjamin Schnitzler. A list of benchmark results created using SDEval and used in published papers, can be found at [8]. There is also a video tutorial/introduction[19] for SDEval on Youtube. It covers the main functionality of the provided scripts.

Andreas Nareike implemented within a project funded by the Saxonian E-Science Initiative *SDSage*[20] – a module for the SageMath [14] generic environment to access the SYMBOLICDATA database as injected dependency object. We don't know to what extend this embedding is used by the SageMath community. The documentation refers[21] only to an earlier implementation provided by Martin Albrecht.

Semantic Technologies

With the consolidation of concepts as Open Access, Open Data and the emerging semantic web the general understanding of the importance of community-based efforts to develop common research infrastructures matured. This development was accompanied with conceptual, technological and architectural standardization processes that had also impact on the development

[10] https://www.wias-berlin.de/research/Leibniz-MMS/index.jsp?lang=en.
[11] https://www.tib.eu/de/.
[12] https://www.de-rse.org/de/index.html.
[13] https://www.force11.org/.
[14] https://www.rd-alliance.org/.
[15] https://www.mathunion.org/ceic/library/world-digital-mathematics-library-wdml.
[16] https://imkt.org.
[17] https://www.softwareheritage.org/.
[18] https://symbolicdata.github.io/SDEval.
[19] https://www.youtube.com/watch?v=CctmrfisZso.
[20] https://symbolicdata.github.io/PolynomialSystems.Sage.
[21] http://doc.sagemath.org/html/en/reference/databases/sage/databases/symbolic_data.html.

of concepts and data structures within the SYMBOLICDATA Project. In 2009 we started to refactor the data along standard Semantic Web concepts based on the Resource Description Framework (RDF). With a new SYMBOLICDATA version released in September 2013 we completed the redesign of the data along RDF based semantic technologies, set up a Virtuoso based RDF triple store and an SPARQL endpoint as Open Data services along Linked Data standards[22]. The importance of the yet heavily growing Linked Open Data Cloud[23] is hardly to underestimate.

Semantic Aware Fingerprints

The main goal of SYMBOLICDATA is on data – structure, maintain and present research data in a digitally and publicly available way. 20 years ago we started with examples from polynomial systems challenges, with the PoSSo test suite and other sources. To make such a collection searchable one has to define and compile meta information about the different objects to cluster them or even identify a single one. This is challenging in particular for polynomial systems since the same example can be noted with different variable names and different term orders. Hence a pure string matching doesn't work. As a first approach we compiled invariants of polynomial normal forms and stored it together with the basis itself in a single information object. Such a strategy – to combine data and metadata in a single object – is commonly used also nowadays, e.g., within the LOM standard – the Learning Objects Metadata are tightly coupled with the Learning Objects themselves.

For our use case such a concept turned out to be suboptimal since it led to an explosion of data: polynomial systems can be interpreted in different ways, e.g., keeping a part of the variables as parameters, as homogenized ideals, bounding variables to special values etc.. Each such version had to be kept as a new data object since the metadata changed even if the basis was the same or could be easily (i.e., in polynomial time) generated from another example. In later versions we used the universal property of the ring $\mathbb{Z}[x_1, \ldots, x_n]$, decided to reduce the number of stored systems and keep track of the way how derived systems are generated from basic ones.

RDF strongly supports such a distinction between data (*resources* in the RDF terminology) and meta information (*resource descriptions*). Data is represented by URI's that can point even to remote locations. Hence RDF is well suited to describe also distributed research data even if the data is maintained by different stakeholders and only the metadata is federated in a common RDF store for search and data analytics.

SYMBOLICDATA operates such a central RDF store [17] and Andreas Nareike enhanced our metadata during his e-science project funded in 2012–2013 with metadata from two distinguished sources – the polytopes database of Andreas Paffenholz [12] and the transitive groups Database for Number Fields of Gunter Malle and Jürgen Klüners [9]. A central challenge was the definition of the metadata as a social process that requires not only sufficient domain specific insight but also resilient agreements about responsibilities to update and maintain the data and metadata. The first part of this challenge is reflected in our concept of *semantic-aware fingerprints* that focuses on the usage condition of *any* research data (awareness of the semantics) and our special use case for the meta information – search. The needle in the haystack, the fingerprint in the police file or the puzzle piece in the stack: To find it a clear domain model and a clear search strategy are required, and both are not independent from each other. For more details we refer to [5].

After a certain consolidation process on March 1, 2016, version 3.1 of the SYMBOLICDATA tools and data was released. The new release contained new resource descriptions ("fingerprints") of remotely available data on transitive groups (*Database for Number Fields* of Gunter Malle and Jürgen Klüners [9]) and polytopes (databases of Andreas Paffenholz [12] within the *polymake* project [3]), a recompiled and extended version of test sets from integer programming – work by Tim Römer (*normaliz* group [1]) –, an extended version of the *SDEval benchmarking environment* – work by Albert Heinle, Benjamin Schnitzler and Viktor Levandovskyy [6] – and a partial integration (SYMBOLICDATA People database, databases of upcoming and past conferences) of data from the CASN – the Computer Algebra Social Network subproject. Furthermore, our GitHub account[24] was transformed into an organizational account and the git repository structure was redesigned better to reflect the special life-cycle requirements of the different parts of our activities.

Research Infrastructure as a Social Project

So far we mainly discussed technical questions of structuring data, defining and compiling metadata and designing tools and workflows for local testing and benchmarking activities. But benchmarking – as *any* process of scientific evaluation – is primarily a *social* process. In other areas of science there are well established benchmark competitions for different algorithmic problem classes with clearly defined rules and places.

In 2012 we organized a workshop on benchmarking[25] with people from communities close to Computer Algebra. Satya Samal presented the PoCaB Project – Platform of Chemical and Biological Analysis Using Computer Algebra Methods – and explained in detail

[22] https://en.wikipedia.org/wiki/Linked_data.
[23] http://lod-cloud.net.
[24] https://github.com/symbolicdata.
[25] https://symbolicdata.github.io/Events.2012-12.

structural approaches within the PoCaB Databases and how data is generated within the PoCaB framework. PoCaB is interlinked with different communities within Computer Algebra (the polynomial systems solving and the polymake communities) and also beyond. It heavily exploits biological databases (BioModel Database, KEGG Database) that come with their own language SMBL and experiences how to express semantic aspects in a computer readable way. This example showed very clearly that communities are not interested in advice from outside how to reinvent wheels properly running for a long time within the community but acknowledge support and advice to organise intercommunity communication more smoothly in a world of evolving Linked Open Data standards.

Johannes Waldmann gave a talk about Benchmarks and Competitions in Theoretical Computer Science presenting best practices of three TCS Communities: Termination, SAT and SMT. For Termination he explained TPBD – the Termination Problems Data Base – and their way of benchmarking: They regularly organize termination competitions on previously agreed data from different problem categories in a similar way as the "Formula I" car race is organized: Upload tools to a single dedicated server that runs all tools on all problems and collects the results in aggregated form on a web page. Usually such a competition runs accompanying the annual large conference in the field. Similarly structured competitions take place in other areas of science, e.g., in High Performance Computing[26] or in the SAT Solver community[27].

The 1998 Special Session on Benchmarking stated that such contests with clear rules are lacking in the area of Computer Algebra. This did not change during the last 20 years. Evaluating the reason for such a longstanding deficit we observed that socially mounted benchmarking cultures live in certain Computer Algebra subcommunities but are rarely communicated beyond their scope. So what about communication between Computer Algebra subcommunities in general? RDF concepts are well suited not only to describe collections of benchmark data but also to support communication on other scientific activities and achievements between different subcommunities. Properly organized metadata generated by different stakeholders can easily be collected not only in a central store but also in a well organized distributed environment as a "scientific Facebook" – we called such a concept *Computer Algebra Social Network* (CASN) [4] – that could be implemented as a network of *CASN nodes* as part of a social research infrastructure within the Linked Open Data Cloud.

Since 2012 we tried to identify problem settings of common interest, implemented building blocks of such a network, tried to get showcases socially running and promoted our CASN idea. We report shortly about three of these showcases and refer to our wiki[28] for more details.

Conferences in Computer Algebra. Reporting about upcoming conferences seems to be a common need in many Computer Algebra subcommunities and could be a first class service of a CASN. The German Fachgruppe set up such a service for a long time in printed form within their Rundbrief. Upcoming conferences are listed independently on the websites of both SIGSAM and the German Fachgruppe. We maintained for several years such information about upcoming and (archiving the entries) past conferences in a structured RDF format that can be used to extract the different web and printed views from a single commonly maintained source. Defining such an exchange format the entries can even be produced by the subcommunities and the boards have merely to collect the information. We terminated that service due to limited staff capacity. A presentation of our past conferences collection can be found at our SYMBOLICDATA demonstration site [16].

RDF is well suited to combine such conference announcements with more detailed information about the conferences (tracks and sessions, papers and authors etc.) that is compiled anyway, e.g., for the web presentation of the conference. In many cases such information is already stored in a structured way and the web site of the conference is generated from that source. In particular, Serge Autexier as the publicity chair invented such a model for the CICM conferences[29] and compiled all information of each of the 12 conferences in publicly available XML files, thus arriving at level 3 of the 5 stars scale[30] for Open Data of Tim Berners-Lee.

As a showcase we transformed four of these presentations into RDF and stored it in our CASN node[31]. This is level 4 of the 5 stars scale since the data is available as RDF but not operated within a RDF store and thus not directly accessible for SPARQL query exploration. This could be part of an upcoming conference reporting structure within an emerging CASN.

The SYMBOLICDATA People Database. Conference announcements are a first class resource of information about people actively working in Computer Algebra. We attached to our conference records information about organizers, invited speakers, program committees etc. We have more than 1000 entries in our database and joint forces[32] with Wolfram Sperber and Uwe Schöneberg (Zentralblatt) to solve the problem of identification of those people in the Zentralblatt and partly also in the MathReviews. A presentation of this

[26] `http://www.hpcchallenge.org`, discontinued after 2014.
[27] `http://www.satcompetition.org/`.
[28] `https://symbolicdata.github.io/CASN`.
[29] `https://www.cicm-conference.org/cicm.php`.
[30] `https://5stardata.info/de/`.
[31] `http://symbolicdata.org/rdf`.
[32] See `https://symbolicdata.github.io/Events.2014-07` for details.

database can be found at our SYMBOLICDATA demonstration site [16].

Such a People Database maintains a set of established URI's and thus is a central building block to get activities in Computer Algebra recognized within the Linked Open Data world. It allows to embed the "stories" told within Computer Algebra and its subcommunities into a bigger world, to join forces with the author disambiguation projects of "big players" (Zentralblatt, Math Reviews, ACM digital library, Springer, Elsevier, ORCID, ResearchGate, VIAF, GND) and thus actively to promote the visibility of Computer Algebra research in the emerging digital world.

Computer Algebra Software. Another central problem within benchmarking Computer Algebra software is software disambiguation. SYMBOLICDATA started 20 years ago to maintain a consolidated list of Computer Algebra software. With the maturing swMATH project [15] we stopped in 2012 such activities and compiled together with Wolfram Sperber and Hagen Chrapary (Zentralblatt) a translation list between our URI's and those of swMATH. In the last years there was much discussion (What is a software, what a package? How to deal with libraries or different versions of the same software?) but little practical progress to prepare that collection for the Linked Open Data world of the 21st century. Being a first class reference of mathematical software swMATH achieves only 2 of the 5 stars of Tim Berners-Lee since it has no open interface to the data itself.

As a showcase we compiled in a common effort with Wolfram Sperber a consolidated RDF based version of Computer Algebra software (that is only a *part* of swMATH, since swMATH addresses mathematical software in general) combining URI's and descriptions from swMATH, the SIGSAM list of Computer Algebra software[33] and also the (very outdated) overview[34] on the website of the German Fachgruppe. We used an undocumented feature of swMATH to compile also links to Zentralblatt reviews of 10 papers related to that software. Since the data is also available from our RDF store it earns all 5 stars of Tim Berners-Lee. A presentation of this database can be found at our SYMBOLICDATA demonstration site [16].

SYMBOLICDATA as Non-Project

A project is usually defined by a *goal*, *attached resources* (money, web space, human resources) and a *time span* (as basis for planning, work packages, milestones etc.). 20 years ago SYMBOLICDATA grew up from the relicts remaining after the end of two such projects – PoSSo and FRISCO – and was designed from the very beginning as *non-project* – it was driven by casual volunteers, bringing in their own resources (time, web space), it was partly supported by different community structures (the Singular group, UMS Medicis, the German Fachgruppe) and it had never a defined project end but survived several "dry periods" almost without activities.

Such a situation is typical for research infrastructures and it is hard to allocate resources for such non-projects in a time of increasing importance of project-oriented research funding. The problems and workarounds are described on the pages of the OEIS Foundation as the goals of another old (since 1964) research infrastructure non-project – *The Online Encyclopedia of Integer Sequences* – in the following way: "1) own the intellectual property, 2) maintain the infrastructure as a service that is freely accessible by the general public, 3) act so as to maintain its own existence indefinitely, 4) collect and distribute funds in order to carry out the first three goals."[35]

During the last years the SYMBOLICDATA team spent efforts on goal 3 to prepare for another "dry period" since we didn't succeed with goal 4[36]. We concentrated the SYMBOLICDATA data and wiki at our GitHub account and terminated several of our ongoing activities (updating the record of upcoming conferences, advancing the alignment with swMATH or the dissertations project[37]).

The domain `http://symbolicdata.org` as a prefix of the SYMBOLICDATA ontologies is one of the core semantic web facilities of SYMBOLICDATA. By the RDF best practices it is of great importance to own that domain and to set up and operate an RDF store under that web address. This domain is owned and sponsored by the German Fachgruppe since 2005 and currently operated on a server at Leipzig University. Unfortunately, the current board of the German Fachgruppe doesn't understand well enough the importance to keep such an arrangement running "indefinitely" (private communication with Gregor Kemper).

What Else?

We acknowledge the strong support from the Board of the German Fachgruppe over many years who sponsors the domain `symbolicdata.org` since 2005 and was the power partner in our experiments towards a CASN. During the last years (2012–2017) we presented SYMBOLICDATA at several international conferences and submitted 6 papers for publication (2 accepted, 4 rejected[38]), not counting our contributions to the Rundbrief of the German Fachgruppe.

Stephen Watt asked in the discussion to my presentation in the *Work in Progress* session at CICM 2014

[33] `https://www.sigsam.org/Resources/Software.html`.
[34] `http://www.fachgruppe-computeralgebra.de/systeme/`.
[35] `http://oeisf.org/#GOALS`.
[36] For details we refer to `https://symbolicdata.github.io/New.html`.
[37] `https://symbolicdata.github.io/Dissertations`.
[38] For details we refer to `https://symbolicdata.github.io/Publications.html`.

"How will you sustainably attract resources for your project?" In my response I shortly explained our non-project philosophy, the role of casual volunteers and ended with the famous answer of Linus Torvalds on a similar question posed by Andrew Tannenbaum: "I won't." But time certainly changed, nowadays there is a big competition between projects resting on such "casual volunteers" and one has to spent much time in advertising the own projects.

We did so and tried to align SYMBOLICDATA not only with swMATH but also with other big community projects as OpenDreamKit (Michael Kohlhase), OSCAR (Wolfram Decker), SIGSAM (Ilias Kotsireas, Matthew England) or people showing interest in "big old data" (Bernd Sturmfels). Such advertisement could only be done with very restricted resources since the single volunteer actively developing SYMBOLICDATA at the moment is a specialist on semantic technologies but far away from core Computer Algebra for many years. The results were disappointing. Even the German Fachgruppe stopped with the relaunch of their website its direct cooperation with SYMBOLICDATA and moved the pages[39] with input from the CASN node of the German Fachgruppe[40] into the background.

A great number of people (Gert-Martin Greuel, Gerhard Pfister, Winfried Neun, Wolfram Sperber, Hannes Schönemann, me) involved in one way or another with SYMBOLICDATA already retired or will retire during the next years. Other people (Olaf Bachmann, Ralf Hemmecke, Andreas Nareike, Albert Heinle) timely involved in SYMBOLICDATA left Computer Algebra or are inactive with the project at the moment.

In this paper we described the main achievements and conceptual points of the SYMBOLICDATA project so far. It is up to the next generation to take over the baton, to evaluate the SYMBOLICDATA heritage and update the parts that are worth to be continued. If any.

Acknowledgement

We are grateful to the SIGSAM to help us reach a wider audience by additionally publishing this article in the "Communications in Computer Algebra".

References

[1] W. Bruns, B. Ichim, T. Römer, R. Sieg, C. Söger. Normaliz. Algorithms for Rational Cones and Affine Monoids. https://www.normaliz.uni-osnabrueck.de. [2018-09-02]

[2] FRISCO – A Framework for Integrated Symbolic/Numeric Computation, 1996–1999. https://cordis.europa.eu/project/rcn/31471_de.html. [2018-09-02]

[3] E. Gawrilow, M. Joswig. Polymake: a Framework for Analyzing Convex Polytopes. In: G. Kalai, G.M. Ziegler (eds.). Polytopes – Combinatorics and Computation (Oberwolfach, 1997), DMV Sem., 29, Birkhäuser, Basel 2000, pp. 43–73.

[4] H.-G. Gräbe, S. Johanning, A. Nareike. The SYMBOLICDATA Project – Towards a Computer Algebra Social Network. In: Workshop and Work in Progress Papers at CICM 2014, CEUR-WS.org, vol. 1186, 2014.

[5] H.-G. Gräbe. Semantic-aware Fingerprints of Symbolic Research Data. In: G.-M. Greuel, T. Koch, P. Paule, A. Sommese (eds.). *Mathematical Software – ICMS 2016*. LNCS 9725, 2016, pp. 411–418.

[6] A. Heinle, V. Levandovskyy. The SDEval Benchmarking Toolkit. ACM Communications in Computer Algebra, vol. 49.1, 2015, pp. 1–10.

[7] A. Heinle, W. Koepf, W. Sperber. Some steps to improve software information. Computeralgebra-Rundbrief 60 (March 2017) *and* Communications in Computer Algebra 51.1 (March 2017), pp. 1–11.

[8] A. Heinle: Benchmarks created using SDEval. https://cs.uwaterloo.ca/~aheinle/software_projects.html [2018-09-07]

[9] J. Klüners, G. Malle. A Database for Number Fields. http://galoisdb.math.uni-paderborn.de/. [2018-09-02]

[10] OpenDreamKit: Open Digital Research Environment Toolkit for the Advancement of Mathematics. http://opendreamkit.org/. [2018-09-01]

[11] The OSCAR project. https://oscar.computeralgebra.de/. [2018-09-01]

[12] A. Paffenholz. Polytope Database. http://www.mathematik.tu-darmstadt.de/~paffenholz/data/. [2018-09-02]

[13] The PoSSo Project. Polynomial Systems Solving – ESPRIT III BRA 6846, 1992–1995. https://cordis.europa.eu/project/rcn/9106_en.html. [2018-09-02]

[14] The SageMath Project. http://www.sagemath.org/. [2018-09-03]

[15] swMATH – an Information Service for Mathematical Software. http://swmath.org. [2018-09-02]

[16] The SYMBOLICDATA Demonstration site. http://symbolicdata.org/info. [2018-09-02]

[17] The SYMBOLICDATA RDF Data Store. http://symbolicdata.org/Data. [2018-09-02]

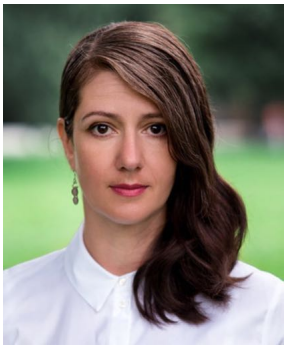
[39] See the overview at http://www.fachgruppe-computeralgebra.de/symbolicdata/.

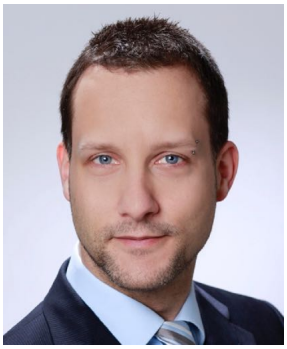
[40] http://www.fachgruppe-computeralgebra.de/rdf/.

Post-Quantum Secure Cryptographic Algorithms

X. Bogomolec (X4pi)
J. Gerhard (BearingPoint Software Solutions GmbH)

indigomind@protonmail.ch
jochen.gerhard@bearingpoint.com





Introduction

The expected dawn of a new technological era has certainly begun when IBM offered their first commercially available 20-Qubit Quantum Computers November 2017. While it was still discussed if it was necessary to take quantum technology into account in the IT industry during the last year, the estimations about their capability evolvement become much more specific now.

Luckily scientific researchers have specialized in the examinations of the various resulting challenges and questions since the beginning of this century. A series of conferences about post-quantum cryptography, the PQCrypto, started in 2006. Since 2010, they take place in another town of the world every year. The following article gives an overview of current developments in algorithmic solutions answering the upcoming threats posed by quantum computers as well as unsolved problems in the classical IT landscape.

Quantum Technologies

Quantum-mechanical phenomena, such as superposition and entanglement, are used for communication, computing, sensoring and simulation. While communication, sensoring and simulation have been realized in publicly announced projects or products, quantum computing was only a matter of research until last november. With the advent of 49 qubit processors quantum supremacy lies within reach, i.e. the potential ability of quantum computing devices to solve problems that classical computers practically cannot solve [2, 3]. IBM has announced to have built a 50 qubit prototype, Google participates in the race with their new record-breaking 72-qubit quantum processor Bristlecone.

Benefits

Quantum technologies offer and promise major benefits. So called adiabatic quantum computers, e.g. the D-Wave 2000Q with 2048 qubits from D-Wave Systems in Canada are able to solve optimization problems that would overburden a classical computer. Photon based quantum key distribution devices from ID Quantique in Switzerland are used by the government in Geneva and other institutions. China has built the 2000km quantum communication channel QUESS between Beijing and Shanghai for banks, the Xinhua News Agency and the government, whose nodes receive keys from their quantum communication satellite. Last year they denoted feasible distances up to 1200 km.
In the future quantum computers with enough stable qubits are expected to be able to help building complex materials as well as solve medical and environmental problems amongst other things.

Threats

It is long known that the security of currently used cryptographic algorithms relying on the hardness of integer factorization and finding discrete logarithms (DLOG systems) [1] will expire with potent enough quantum computers. All public parameters like public keys from asymmetric key pairs can then be used to compute the corresponding private keys. With the knowledge of those private keys, encrypted data, which was collected and assigned to the relevant key exchanges, will no longer remain secret. For technologies like public distributed ledgers, where encrypted data is publicly available, this threat is even more serious.

Solutions

Quantum Key Distribution

QKD is an implemented cryptographic protocol for key distribution involving components of quantum mechanics. The security of encryption that uses quantum key distribution relies on the foundations of quantum mechanics. In this context, the process of measuring a quantum system in general disturbs the system itself. So any third party trying to gain knowledge of the key would be detected by the original communication parties.

Quantum key distribution networks have already been established in China (QUESS), Austria (SECQC), Japan (Tokyo QKD Network), Switzerland (SwissQuantum) and the USA (DARPA). Disadvantages for widespread practical usage are limited distances between communication partners and the need of expensive hardware.
Rarely mentioned is the fact that message source authentication does not come with QKD genuinely. Man-in-the-middle attacks are also possible if the communication parties do not agree on an authentication protocol beforehand.

Post-Quantum Cryptography

The alternative to QKD are algorithms whose security rely on mathematical properties, like hardness of computing the inversion of a one way function even with a quantum computer. There are four mathematical areas which offer solutions for encryption, key exchanges and signatures. Some of them are still in the middle of the research process, others have been observed and challenged for years. The advantages of post-quantum cryptography are that they can run effectively on currently used devices such as smart phones, desktops and IoTs and they can be enabled by simple software updates.

Code-Based
Syndrome decoding of linear error-correcting codes is NP-complete considered as a decision problem if the number of errors is unbounded. On the other hand, some classes of linear codes have very fast decoding algorithms. The basic idea of a code-based crypto system is to choose a linear code with fast decoding algorithm and disguise it as a general linear code. Then the attacker has to use syndrome decoding for decrypting the message while the message receiver, who also set up the system, can remove the disguise and use the fast decoding algorithm.

MCELIECE and the NIEDERREITER cryptosystems are two basic encryption schemes built on this setup. MCELIECE was the first scheme using randomization in the encryption process. Both systems consist of three algorithms:

1) Probabilistic key generation algorithm producing an asymmetric key pair,
2) Probabilistic encryption algorithm,
3) Deterministic decryption algorithm.

The private key is an (n, k)-linear error correcting code represented by a generator matrix G, with a known efficient decoding algorithm. Originally binary Goppa Codes with the Patterson decoding algorithm were used. The public key is the generator matrix G perturbated by two randomly chosen invertible matrices S and P

$$G' = SGP$$

where S, a $(k \times k)$ matrix, functions as a scrambler and P is a $(n \times n)$ permutation matrix. Parameters proposed by MCELIECE [4] result in a public key of 2^{16} bytes size. The most effective attacks on MCELIECE use information-set decoding. To resist those in a quantum computing context, key sizes have to be increased by a factor of 4.

The NIEDERREITER scheme [5] applies the same idea to a parity check matrix H of a linear code. The encryption is about ten times faster than McEliece. McEliece was originally believed not to be usable for authentication or signature schemes because the encryption algorithm is not one-to-one and the total algorithm is truly asymmetric, meaning, encryption and decryption do not commute. However, a one-time signature scheme based on MCELIECE and NIEDERREITER was proposed at the Asiacrypt in 2001 [6]:

1) choose a hash function h and compute the hash value $h(d)$ of the document d which has to be be signed,
2) decrypt the hash value $h(d)$ as if it was an instance of the ciphertext,
3) append the decrypted hash value to the document as a signature.

As the second step in the signature scheme almost always fails, the system additionally specifies a deterministic way of tweaking d until a hash value $h(d)$ is found which can be decrypted. Verification then applies the public encryption function to the signature to the signature and compares it to the hash value of the document.

The most recently published code-based key exchange protocol is OUROBOROS [7]. It uses quasi-cyclic codes in Hamming metric in the encryption algorithm, efficient decoding is achieved through bit flipping in the Random Oracle Model. Encryption and decryption are faster than RSA for comparative benchmarks (`https://bench.cr.yp.to`). Ouroboros' integration into the OpenSSL/TLS library is planned and it is proposed as post-quantum secure algorithm at the NIST.

Hash-Based
This domain is limited to digital signatures schemes which rely exclusively on the security of the underlying hash functions so far. The signatures themselves reveal a part of the signing key and can only be used for one message, same as it is known from one-time pads such as visual cryptography shares.

Merkle tree signature schemes, introduced in 1979, combine a one-time signature scheme with a Merkle tree structure. Building blocks of the Merkle trees are one-time signature key pairs, with the node at the top being the global public key. This typically 256 bit large key can be verified with the path to another given public one-time key in the tree using a sequence of tree nodes, called the authentication path. The global private key is usually derived from a seed generated by a pseudo random number generator and has the size of 256 bits as well. Hereby, the number of possibilities for such signatures are all possible combinations of the simple one-time signatures within the tree structure. This procedure considerously enhances the security of the scheme against brute force attacks.

The latest performance improved hash-based signature scheme is SPHINCS$^+$[8], the advanced SPHINCS [9] scheme which was presented at EUROCRYPT 2015. Unlike its predecessors, XMSS and LMS, it is stateless, meaning that signing doesn't require updating the

secret key. It is a so called few-times scheme, where "few-times" means as much as after 2^{64} signatures it is necessary to reinitiate the complete scheme. Its signature sizes range from 8kb for NIST security level 1 to 30kb for NIST security level 5.

Lattice-Based
Lattice based codes come with the challenge of finding the nearest lattice point or a shortest basis for a given lattice. Both problems and their approximate adequates have been solved with NP-hard algorithms only. Given they are one of the longest known public key crypto systems, they can be fairly seen as the most promising post quantum crypto approaches. Low memory requirements and high speed computations let them run effectively on all currently and widely used devices. However, due to their significantly bigger key sizes they had not been as thoroughly researched and applied as RSA, EL GAMAL [10] or DLOG systems.

NTRU was the first successful lattice-based asymmetric cryptosystem. It was was proposed and patented in 1996 [11]. With the expiration of the patent in 2016, NTRU Prime [12], an improvement by eliminating worrisome algebraic structure could be published. Their security rely on the interaction of a polynomial mixing system with the independence of reduction modulo two relatively prime integers p and q.

Another popular ingredient of lattice-based algorithms is the Learning with Errors (LWE) problem. It was used in BCNS [13], which phrased Peikerts key encapsulation algorithm as a key exchange protocol. BCNS was the first lattice-based algorithm which was integrated into the OpenSSL library.

With NEW HOPE [14] an improvement was achieved by choosing more efficient parameters and shifting from LWE to Ring Learning with Errors (RLWE). The NEW HOPE protocol allows man-in-the-middle attacks, message authentication has to be implemented additionally. Google ran an experiment by using NEW HOPE embedded in an ECC procedure for a certain number of connections between the Chrome browser and their own servers in 2016. Since 2017, Infineon works on the first generation of contactless post-quantum chips with Pöppelmann, one of the authors of the NEW HOPE paper.

DILITHIUM [15], a module-lattice-based signature scheme was designed with the intention to be easy to implement against side-channel attacks, while offering comparable efficiency to previously developed lattice-based signature schemes. The key innovation is the replacement of Gaussian sampling by uniformly random sampling over a bounded domain. Furthermore, the public key sizes are reduced by more than a factor of 2.

All these algorithms except BCNS are submitted to the NIST post-quantum cryptography standardization process.

Multivariate
The proven NP-hardness and NP-completeness of solving multivariate polynomial equations over a finite field F are the reason why schemes with those asymmetric cryptographic primitives are considered good candidates for post-quantum security. Most of the published schemes use multivariate quadratics, namely polynomials of degree two.
The basic scheme consists of two affine transformations

$$S : F^n \to F^n$$

$$T : F^m \to F^m$$

and an easy to invert quadratic map

$$P' : F^m \to F^n$$

The trapdoor $(S^{-1}, {P'}^{-1}, T^{-1})$ represents the private key, whithout which the public key $P = S \circ P' \circ T$ is assumed to be hard to invert.
A first multivariate quadratic scheme, C^* [16], was presented at the EUROCRYPT CONFERENCE 1988. After it was broken [17], the general principal was used for stronger schemes, such as HIDDEN FIELD EQUATIONS [18] and QUAD [19].

Multivariate signature schemes provide the shortest signatures amongst post-quantum algorithms (GUI [20] 129 bit over $GF(2)$ for a quantum security level of 80 bit). The signature x of a message m is created by hashing m into a vector $y \in F^n$ and computing $x = P^{-1}(y) = T^{-1}(P'(S^{-1}(y)))$. The receiver can simply compute the hash y and check if $P(x) = y$.

MEDIUM FIELD SIGNATURE SCHEMES [21] with fewer equations and variables in the public key offer a further reduction in key sizes, greater efficiency and scalable levels of security. A proposal is submitted to the NIST standardization process of post-quantum signature schemes.

Isogeny-Based
One of the latest and most challenging post-quantum crypto ideas is the application of isogeny based encryption schemes like SUPERSINGULAR ISOGENY DIFFIE-HELLMANN (SIDH). With 2688-bit public keys at a 128-bit quantum security level, this scheme uses the smallest keys amongst post-quantum key exchanges. Additionally it supports perfect forward secrecy, a property which preserves the confidentiality of old communication sessions even if long-term keys have been compromised.

Although they are not as thoroughly researched as the previously mentioned schemes, Microsoft published an experimental VPN-library with a SUPERSINGULAR ISOGENY KEY ENCAPSULATION algorithm (SIKE) based on SIDH amongst a LWE key exchange and a signature algorithm using symmetric-key primitives and non-interactive zero-knowledge proofs [22]. SIKE is

also submitted to the NIST standardization process of post-quantum cryptography schemes

In a youtube video of a Microsoft research session where SIKE is presented to other researchers by Christophe Petit, he states at the end: "I wouldn't bet national security on it". On the other hand, SIDH was also denoted as "the hottest thing we have" in the key note of the pqcrypto conference 2017.

Amendment
Paramater choices are much more delicate for post-quantum crypto schemes than they are for classical ones. Furthermore classical asymmetric schemes mostly rely on number theory, a topic which has been studied in early courses at universities, where post-quantum algorithms include more mathematics from courses which are usually taught at later stages of study courses.

It will not only be a challenge to distinguish and weigh the complex influences on security of post-quantum encryption schemes, there will also be an increased need of cooperations between mathematicians, computer scientists and programmers to mitigate flaws in implementations, configurations and applications.

For someone who is not familiar with the concept of a mathematical conjecture, it is hard to understand on what ground the security of cryptography is built and what time can do to it, with or without regard to emerging technologies. Who can say for sure that there is no-one who generates one RSA key pair after another since decades and stores them in a huge database where he can simply assign a private key to its public key if it is present in his own collection? How many distinctive usable key pairs can even be expected within the range of a 4096-bit integer?

References

[1] P. W. Shor Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms ona Quantum Computer `https://arxiv.org/abs/quant-ph/9508027`, 1995.

[2] J. Hsu Spectrum IEEE Tech Talk, January 9, 2018 `https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy`

[3] J. Kelly Google AI Blog, March 5, 2018 `https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html`

[4] R. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory, *DSN Progress Report 42-44*, 1978.

[5] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory, *Problems of Control and Information Theory*, 1886.

[6] N. T. Courtois, M. Finiasz and N. Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme, *Asiacrypt*, 2001.

[7] J. C. Deneuveville, P. Gaborit and G. Zémor. A Simple, Secure and Efficient Key Exchange Protocol Based on Coding Theory, *Springer, Post-Quantum Cryptography - PQCrypto 2017*

[8] J. Rijneveld and S. Kölbl. The SPHINCS$^+$ reference code, `https://github.com/sphincs/sphincsplus`.

[9] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe and Z. Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures, *Springer, Advances in Cryptology - EUROCRYPT 2015*.

[10] J. Hoffstein, J. Pipher and J. H. Silverman. An Introduction to Mathematical Cryptography, *Springer Science+Business Media*, 2008.

[11] J. Hoffstein, J. Pipher and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem, *SpringerLink International Number Theory Symposium*, 1998.

[12] D. Bernstein, C. Chuengsatiansup, T. Lange and C. van Vredendaal. NTRU Prime: reducing attack surface at low cost, *Cryptology ePrint Archive 2017*.

[13] J. W. Bos, C. Costello, M. Naehrig, and A. D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem, *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013.

[14] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe. Post-quantum key exchange – a new hope, *IEEE Security & Privacy 2015*.

[15] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler and D. Stehlé. CRYSTALS - Dilithium: Digital Signatures from Module Lattices, *Cryptology ePrint Archive 2017*.

[16] T. Matsumoto and H. Imai. Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption, *Springer EUROCRYPT '88*.

[17] J. Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88, *Springer CRYPTO '95'*.

[18] J. Patarin. Hidden Field Equations and Isomorphisms of Polynomials: Two New Families of Asymmetric Algorithms, *EUROCRYPT 1996*.

[19] C. Berbain, H. Gilbert and J. Patarin. A Practical Stream Cipher with Provable Security, *Springer, Advances in Cryptology - EUROCRYPT 2006*.

[20] M. S. E. Mohamed and A. Petzoldt. The Shortest Signatures Ever, *TU Darmstadt, Germany and Kyushu University, Fukuoka Japan*, 2015.

[21] A. Petzoldt, M. -S. Chen, J. Ding and B. -Y. Yang. HMFEv - An Efficient Multivariate Signature Scheme, *Springer, Post-Quantum Cryptography - PQCrypto 2017*.

[22] Microsoft Research Security and Cryptography Group. Microsoft PQCrypto VPN, `https://github.com/Microsoft/PQCrypto-VPN`.

Workshop-Förderung der Fachgruppe:

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

Anträge können bis mit einer kurzen Beschreibung des Workshops (ca. 1 DIN A4 Seite; kurze Beschreibung des Gebiets, Thema des Workshops, Zielgruppe, Budget-Planung) und einer Darstellung, inwiefern diese Förderung einen deutlich erkennbaren Beitrag zum Gelingen des Workshops und zur Nachwuchsförderung liefert, an den Sprecher der Fachgruppe gerichtet werden: **kemper@ma.tum.de**, bitte '**Workshop-Förderung**' im Betreff angeben.

Fachgruppe Computeralgebra

Rings: A JVM library for Commutative Algebra

S. Poslavsky
(Institute for High Energy Physics NRC "Kurchatov Institute" Protvino, Russia)

stvlpos@mail.ru

Introduction

RINGS is an open-source library written in Java and Scala which implements basic concepts and algorithms from computational commutative algebra. The goal is to provide a high-performance implementation packed into a lightweight *library* (not a full-featured CAS) with a clean API, which meets modern standards of software development.

Java is perhaps *the* most widely used language in industry today and combines several programming paradigms including object-oriented, generic, and functional programming. Scala, which is fully interoperable with Java, additionally implements several advanced concepts like pattern matching, an advanced type system, and type enrichment. Use of these concepts in RINGS made it possible to implement mathematics in a quite natural and expressive way directly inside the programming environment offered by Java and Scala.

RINGS is a cross-platform library compatible with modern JVM-based languages (Java, Scala, Closure, Kotlin, Groovy, etc.) or easily interacted with from native applications both on POSIX systems and Windows, either through the Java Native Interface (C/C++) or via simple pipes.

The Scala extension allows for expressive type-safe interaction with the library from within a Scala application or from the REPL console. Both Java and Scala are strongly and statically typed languages, and the mathematical structures used in RINGS also form a fully typed hierarchy, in contrast to many computer-algebra systems and libraries that operate either with untyped or weakly typed objects or use duck typing. The API provided by the library allows to write short and expressive code on top of the library, using both object-oriented and functional programming paradigms in a completely type-safe manner.

RINGS is hosted on github.com/PoslavskySV/rings. Installation instructions and comprehensive online documentation can be found at rings.readthedocs.io.

Overview

In a nutshell, RINGS allows to construct different rings and perform arithmetic in them, including both very basic math operations and advanced methods like polynomial factorization, linear systems solving, and construction of Gröbner bases.

The built-in rings include $\mathbb{Z}$, finite fields $\mathbb{Z}_p$ and $\mathbb{GF}(p,k)$ (with arbitrarily large p), field extensions $F(\alpha_1,\ldots,\alpha_s)$ (including e.g. Gaussian numbers $\mathbb{Q}(i)$), $\mathrm{Frac}(R)$ (including e.g. rationals $\mathbb{Q}$ and rational functions), $R[x]$ and the multivariate polynomial ring $R[\vec{X}]$, where R is an arbitrary ground ring which may be either one or any combination of the listed rings.

Below we illustrate the features of RINGS in a step-by-step example. All code snippets are in Scala and can be evaluated directly in REPL. Java examples can be found in the online manual http://rings.readthedocs.io.

Basic concepts

To start our illustration, let's take a finite field $\mathbb{GF}(17,3)$ and perform some basic math in it:

```
1  // Galois field GF(17,3) ("t" is the generator)
2  implicit val gf = GF(17, 3, "t")
3  // parse ring element from string
4  val t  = gf("t")
5  // do some basic math (+-*/)
6  val t1 = 3 + t - t.pow(22)/(1 + t + t.pow(9))
7  // compute minimal polynomial of t1
8  val mpoly = gf.minimalPolynomial(t1)
9  // assert that t1 is a root of mpoly
10 assert( gf(mpoly.composition(t1)).isZero )
```

This very basic example already reveals some important programming concepts implemented in Java and Scala.

The first key point is that each object from the above example has full compile-time type, which is just omitted in our example for shortness but inferred automatically by the compiler. The above lines are in fact equivalent to:

```
val gf : GaloisField64 = ...
val t  : UnivariatePolynomialZp64       = ...
val t1 : UnivariatePolynomialZp64       = ...
val mpoly : UnivariatePolynomialZp64    = ...
```

`GaloisField64` implements the Galois field $\mathbb{GF}(p, q)$ with $p < 2^{64}$ (machine word) and is a subtype of `Ring[UnivariatePolynomialZp64]`. (The interface `Ring[E]` is a supertype for all rings; it defines all mathematical operations on elements of type `E`, plus some methods inherent to rings like `.isField()`, `.characteristic()`, and `.cardinality()`.)

`UnivariatePolynomialZp64` represents univariate polynomials over $\mathbb{Z}_p$ ($p < 2^{64}$) and is used as the actual representation of elements of Galois fields.

The second key point, specific to Scala programming, concerns the concept of *type enrichment* which allows to "enrich" existing classes by adding new functionality. In RINGS it is used to add operator overloading for elements of arbitrary rings in an elegant way: all math operators (e.g. `*` or `+`) work for an arbitrary type `E`, provided that there is an `implicit` instance of `Ring[E]` in the scope:

```
implicit val ring : Ring[E] = ...
val t1 : E = ... ; val t2 : E = ...
t1 + t2 // compiles to ring.add(t1, t2)
t1 * t2 // compiles to ring.multiply(t1, t2)
```

The following example shows how the presence of an `implicit` ring changes the behavior of math operators:

```
// some arbitrary-precision integers
val t1 : IntZ = 12 ; val t2 : IntZ = 13
assert (t1 * t2 === 156) // multiply integers
{
  implicit val ring = Zp(2)
  assert (t1 * t2 === 0) // multiply modulo 2
}
{
  implicit val ring = Zp(17)
  assert (t1 * t2 === 3) // multiply modulo 17
}
```

The third point worth mentioning is the syntactic sugar applied in line 4, which is actually `gf.parse("t")` in full. In fact, there are several such methods for different conversions, which may all be called in the same way (this is a common pattern in Scala):

```
// from string
val elem = gf("1 + t^2")
// from Int
val unit = gf(1)
// from elements of other GF fields
val othFieldElement = GF(19, 5).randomElement()
val cast = gf(othFieldElement)
// syntactic sugar for multiple assignment
val (el1, el2) = gf("t + 1", "t + 2")
```

Polynomials, GCDs, Factorization

Our next step is to define some multivariate polynomial ring over the ground ring $\mathbb{GF}(17, 3)$. Below we define such a ring and perform some math operations in the same fashion we did above:

```
11 // multivariate ring GF(17,3)[x,y,z]
12 implicit val ring =
       MultivariateRing(gf, Array("x","y","z"),
       GREVLEX)
13 val (x, y, z) = ring("x", "y", "z")
14 // construct some multivariate polynomials
15 val p1 = (t.pow(2) + 1)*x*y.pow(2)*z +
        (t + 1)*x.pow(5) * z*y.pow(6) + 1
16 val p2 = p1.pow(2) + (t + 1)*x.pow(2)*y.pow(2) +
        (t.pow(9) + 1)*z.pow(7)
17 val p3 = (p1 + p2).pow(2) - 1
```

Again, the `ring` instance is defined `implicit`, hence all math operations on multivariate polynomials of type `MultivariatePolynomial[UnivariatePolynomialZp64]` will be delegated to that instance.

In line 12 we explicitly specified `GREVLEX` monomial ordering for multivariate polynomials. This choice affects algorithms like multivariate division and Gröbner bases. The explicit order may be omitted (`GREVLEX` will be used by default).

Polynomial greatest common divisors and polynomial factorization work for polynomials over all available built-in rings. Continue our example:

```
18 // GCD of polynomials from GF(17,3)[x,y,z]
19 val gcd1 = ring.gcd(p1 * p3, p2 * p3)
20 assert ( gcd1 % p3 === 0 )
21 val gcd2 = ring.gcd(p1 * p3, p2 * p3 + 1)
22 assert ( gcd2.isConstant )
23
24 // large polynomial from GF(17,3)[x,y,z]
25 // with more than 4 × 10^3 terms and degree 204
26 val hugePoly = p1 * p2.pow(2) * p3.pow(3)
27 // factorize it
28 val factors = ring.factor(hugePoly)
```

One of the key features of the RINGS library is that it does polynomial GCD and factorization of really huge polynomials over different ground rings robustly and fast.

To illustrate how the performance of e.g. polynomial GCD is manifested in applications, suppose we need to solve a system of linear equations with symbolic coefficients. In the continuation of our example, we use $\mathbb{GF}(17, 3)[x, y, z]$ for the ring of coefficients, so the solution belongs to the field of rational functions over 3 variables with coefficients from $\mathbb{GF}(17, 3)$. This can be accomplished in RINGS easily:

```
29 //field of rational functions Frac(GF(17,3)[x,y,z])
30 implicit val ratRing = Frac(ring)
31 // convert x, y, z and t to rationals
32 val (rx, ry, rz, rt) = ratRing(x, y, z, ring(t))
33 // lhs matrix
34 val lhs =  Array(
       Array(rt + rx + rz, ry * rz, rz - rx * ry),
       Array(rx - ry - rt, rx / ry, rz + rx / ry),
       Array(rx * ry / rt, rx + ry, rz / rx + ry))
35 // rhs column
36 val rhs = Array(rx, ry, rz)
37 // solve the system with Gaussian elimination
38 val solution = LinearSolver.solve[ratRing.
       ElementType](ratRing, lhs, rhs)
```

Standard Gaussian elimination needs $O(n^3)$ field operations, which means $O(n^3)$ multivariate polynomial GCDs (since each fraction should be reduced). The speed-up of the polynomial GCD by a factor two reduces the time to solve the entire system by nearly an order of magnitude.

Benchmarks

To compare the speed of GCD with other tools, the following benchmark was used. Polynomials a, b, and g were generated at random and the time needed to compute $\gcd(ag, bg)$ was measured. Each polynomial had

40 terms (so the products ag and bg had at most 1600 terms each), and monomial exponents were generated using two strategies. In the first one (uniform), the exponent of each variable in the monomial was taken uniformly in $0 \le \exp \le 30$. In the second strategy (sharp) the total degree of each monomial was fixed and equal to 50 (so input polynomials were homogeneous). The benchmark was run for different numbers of variables. The performance of RINGS 2.3.2 was compared with that of MATHEMATICA 11.1.1, SINGULAR 4.1.0 [1], FORM 4.2.0 [2], and FERMAT 6.19 [3].

Fig. 1 shows how the performance of different systems compares with increasing number of variables. In all considered problems the performance of RINGS was unmatched. Remarkably, its performance is almost independent of the number of variables in such sparse problems.

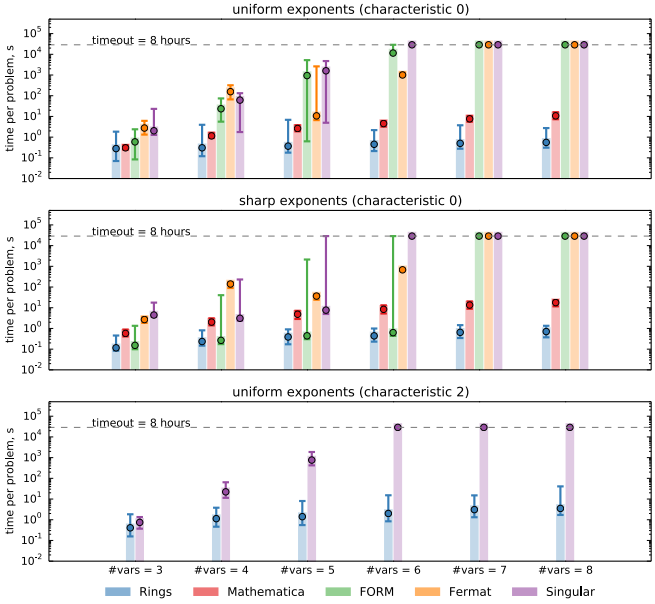


Figure 1: *Dependence of multivariate GCD performance on the number of variables. Each problem set contains 110 problems, points correspond to the median times, and the error bands correspond to the smallest and largest execution time required to compute the GCD within the problem set. If computation of a single GCD took more than 8 hours (timeout) it was aborted and the timeout value was adjoined to the statistics.*

Performance of polynomial factorization was tested using the following benchmark. Polynomials a, b, and c were generated at random and the time needed to compute $\mathrm{factor}(abc+1)$ (trivial) and $\mathrm{factor}(abc)$ (non-trivial) was measured. Each polynomial had 20 terms (so the products abc had at most 8000 terms each). The exponent of each variable in a monomial was chosen uniformly in $0 \le \exp \le 30$.

Fig. 2 shows how performance of multivariate factorization depends on the number of variables. It follows that the median time required to compute factorization changes quite slowly, while some outliers (typically ten times slower than median values) appear when the number of variables becomes large.

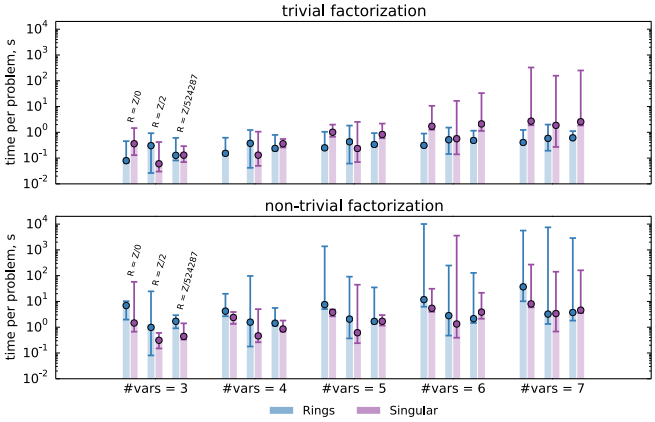


Figure 2: *Dependence of multivariate factorization performance on the number of variables. Each problem set contained 110 problems, points correspond to the median times, and the error bands correspond to the smallest and largest execution time required to compute the factorization within the problem set.*

The benchmarks shown above involve only sparse problems, which are more frequent in practice. The full set of benchmarks, including dense problems, is found at https://github.com/PoslavskySV/rings.benchmarks.

Implementation notes

To achieve the high performance of polynomial GCD and factorization, RINGS uses different algorithms depending on the type of input.

First of all, univariate and multivariate polynomials have different implementations: univariate are represented as dense arrays while multivariate are represented as sparse containers (implemented with a red–black `TreeMap`). Polynomials over the ring $\mathbb{Z}_p$ with $p < 2^{64}$, elements of which can be represented as 64-bit integers, have separate, highly optimized implementations.

Univariate GCD uses the Half-GCD algorithm for polynomials over finite fields and modular algorithms in other cases (i.e. $\mathbb{Q}[x]$ and $\mathbb{Q}(\alpha)[x]$). Univariate polynomial factorization is implemented with the use of the Cantor–Zassenhaus method with optional use of Shoup's baby-step giant-step algorithm [4] (for large polynomials or for finite fields with large characteristic).

Multivariate GCD switches between Zippel's sparse algorithms and Enhanced Extended Zassenhaus algorithm (EEZ-GCD). The latter is used only on very dense problems. Zippel's algorithms require that the ground ring contains a sufficient number of elements (so they will always fail in e.g. $\mathbb{Z}_2[\vec{X}]$). When the cardinality of a ground ring is not sufficiently large, RINGS switches to a Kaltofen–Monagan generic modular algorithm [5]. For polynomials over algebraic number fields, the modular approach with either sparse (Zippel) or dense (EEZ-GCD) interpolation is used with further rational number reconstruction.

Multivariate factorization uses Kaltofen's algorithm [6], with major modifications due to Lee [7]. For factoring bivariate polynomials, the very efficient Bernardin

algorithm [8] is used. Additionally, RINGS performs some fast early checks based on Newton polygons to ensure that there is a nontrivial factorization pattern. Multivariate Hensel lifting is done via a Zippel-like sparse method: the problem of lifting is reduced to a system of (in general non-linear) equations which may be solved efficiently in many cases. For factorization of polynomials over algebraic number fields $\mathbb{Q}(\alpha)$ RINGS uses Trager's algorithm [9].

Ideals and Gröbner bases

The concept of a mathematical ideal is implemented by the `Ideal` class, which computes the corresponding Gröbner basis automatically at instantiation. The following code snippet continues our example with polynomial ring $\mathbb{GF}(17,3)[x,y,z]$ and illustrates the main methods provided by the `Ideal` class:

```
// define a set of polynomial generators
val (i1, i2, i3) = (x + y + z, x - y - z,
                    y.pow(2) - z.pow(2))
// construct Ideal from a set of generators
// (Groebner basis with GREVLEX order will be
    automatically computed)
val ideal = Ideal(Seq(i1, i2, i3))
// print Groebner basis
println( ideal.groebnerBasis )
// print dimension of ideal
println( ideal.dimension )
// print degree of ideal
println( ideal.degree )
// print Hilbert series of ideal
println( ideal.hilbertSeries )
// reduce poly modulo ideal
val p4 = p2 %% ideal
```

RINGS also provides built-in algorithms for manipulating ideals:

```
val othIdeal = Ideal(Seq(p1, p2, p3))
// union of ideals
val union = ideal + othIdeal
// product of ideals
val prod = ideal * othIdeal
// intersection of ideals
val in = ideal intersection othIdeal
// quotient of ideals
val quot = othIdeal :/ ideal
```

Table 1: *Time required to compute Gröbner basis in graded reverse lexicographic order. In case of $\mathbb{Z}_p$ the coefficient-ring value of $p = 1000003$ was used.*

Problem	Ring	RINGS	MMA	SINGULAR
c-7	$\mathbb{Z}_p$	3s	26s	N/A
c-8	$\mathbb{Z}_p$	51s	897s	39s
c-9	$\mathbb{Z}_p$	14603s	∞	8523s
k-7	$\mathbb{Z}_p$	0.5s	2.4s	0.1s
k-8	$\mathbb{Z}_p$	2s	24s	1s
k-9	$\mathbb{Z}_p$	2s	22s	1s
k-10	$\mathbb{Z}_p$	9s	216s	9s
k-11	$\mathbb{Z}_p$	54s	2295s	65s
k-12	$\mathbb{Z}_p$	363s	28234s	677s
k-7	$\mathbb{Q}$	5s	4s	1.2s
k-8	$\mathbb{Q}$	39s	27s	10s
k-9	$\mathbb{Q}$	40s	29s	10s
k-10	$\mathbb{Q}$	1045s	251s	124s

Implementation and benchmarks

RINGS implements Faugère's F4 and Buchberger's algorithms for computing Gröbner bases. These implementations show sufficient performance on small and medium problems. Table 1 shows the time needed to compute Gröbner bases of classical Katsura and cyclic systems for RINGS, MATHEMATICA (MMA), and SINGULAR. Timings are in general comparable between RINGS and SINGULAR for polynomial ideals over $\mathbb{Z}_p$ while for $\mathbb{Q}$ RINGS behaves worse. It should be noted that for very hard problems much more efficient dedicated tools like FGB [10] (proprietary) or OPENF4 [11] (open source) exist.

Programming with RINGS in Scala

The important feature of RINGS is that it allows to write short and expressive code on top of it using both object-oriented and powerful functional programming. Consider the following short example, which implements a solver for Diophantine equations, i.e. a straightforward generalization of the extended GCD on more than two arguments:

```
/**
 * Solves equation $\sum f_i s_i = \gcd(f_1,\ldots,f_N)$ for
     given $f_i$ and unknown $s_i$
 * @return a tuple (gcd, solution)
 */
def solveDiophantine[E](fi: Seq[E])
                       (implicit ring: Ring[E])=
  fi.foldLeft((ring(0), Seq.empty[E])) {
    case ((gcd, seq), f) =>
      val xgcd = ring.extendedGCD(gcd, f)
      (xgcd(0), seq.map(_ * xgcd(1)) :+ xgcd(2))
  }
```

With this function it is very easy to implement, for example, an efficient algorithm for partial fraction decomposition with just a few lines of code. The resulting function will work with elements of arbitrary fields of fractions:

```
/** Computes partial fraction decomposition of
    given rational */
def apart[E](frac: Rational[E]) = {
  implicit val ring: Ring[E] = frac.ring
  val facs = ring
               .factor(frac.denominator)
               .map {case (f, exp) => f.pow(exp)}
  val (gcd,  nums) = solveDiophantine(facs.map(
    frac.denominator / _))
  val (ints, rats) = (nums zip facs)
    .map { case (num, den) =>
       Rational(frac.numerator * num, den * gcd)
    }
    // extract integral parts
    .flatMap(_.normal)
    // separate integrals and fractions
    .partition(_.isIntegral)\

  // return the result
  rats :+ ints.foldLeft(Rational(ring(0)))(_+_)
}

// partial fraction decomposition for rationals
val qFracs = apart( Q("1234213 / 2341352") )

// partial fraction decomposition for functions
val ufRing = Frac(UnivariateRingZp64(17, "x"))
val expr = ufRing("1 / (3 - 3*x^2 - x^3 + x^5)")
val pFracs = apart(expr)
```

The function `apart[E]` is defined as a generic function which can be applied to fractions over elements of arbitrary rings (that should be Euclidean rings of course). Returning to our initial example where we've constructed a field $\mathrm{Frac}(\mathbb{GF}(17,3)[x,y,z])$, let us adjoin a new variable, say W, and construct the partial fraction decomposition in this complicated field:

```
105 // partial fraction decomposition of rational
        functions
106 // in the ring Frac(GF(17,3)[x,y,z])[W]
107 implicit val uRing = UnivariateRing(ratRing,"W")
108 val W = uRing("W")
109 val fracs = apart(Rational(W + 1,
        (rx/ry + W.pow(2)) * (rz/rx + W.pow(3))))
```

The function call on the last line involves nearly all main components of RINGS library: from very basic algebra to multivariate factorization over sophisticated rings.

The above examples show how powerful features of RINGS in a combination with expressive and type-safe Scala syntax may be used to implement quite non-trivial and generic functionality with little effort.

Summary and outlook

RINGS is a high-performance and lightweight library for commutative algebra that provides both basic methods for manipulating with polynomials and high-level methods including polynomial GCD, factorization, and Gröbner bases over sophisticated ground rings. Special attention was paid to high performance and a well-designed API. High performance is crucial for today's computational problems that arise in many research areas including high-energy physics, commutative algebra, cryptography, etc. The API provided by the library allows to write short and expressive code on top of the library, using both object-oriented and functional programming paradigms in a completely type-safe manner.

Some of the planned future work for RINGS includes improvement of Gröbner bases algorithms (better implementation of "change of ordering algorithm" and some special improvements for polynomials over $\mathbb{Q}$), optimization of univariate polynomials with more advanced methods for fast multiplication, specific optimized implementation of $\mathbb{GF}(2,k)$ fields which frequently arise in cryptography, and better built-in support for polynomials over arbitrary-precision real numbers ($\mathbb{R}[\vec{X}]$) and over 64-bit machine floating-point numbers ($\mathbb{R}64[\vec{X}]$).

RINGS is an open-source library licensed under Apache 2.0. The source code and comprehensive online manual can be found at http://ringsalgebra.io.

References

[1] W. Decker, G.M. Greuel, G. Pfister, H. Schönemann (2018), SINGULAR — A computer algebra system for polynomial computations, `http://www.singular.uni-kl.de`.

[2] B. Ruijl, T. Ueda, J. Vermaseren (2017), FORM version 4.2, arXiv:1707.06453.

[3] R. Lewis (2018), Fermat, `http://home.bway.net/lewis`.

[4] V. Shoup (1995), A new polynomial factorization algorithm and its implementation, *J. Symb. Comput.* 4 (20) 363–397.

[5] E. Kaltofen, M.B. Monagan (1999), On the genericity of the modular polynomial gcd algorithm, in: *Proceedings of ISSAC'99* 59–66, ACM Press.

[6] E. Kaltofen (1985), Sparse hensel lifting, in: *Proceedings of EUROCAL'85* (2) 4–17, Springer.

[7] M.M. Lee (2013), Factorization of multivariate polynomials, PhD thesis, University of Kaiserslautern.

[8] L. Bernardin, M. Monagan (1997), Efficient multivariate factorization over finite fields, on: *Proceedings of AAECC'97* (1255) 15–28, Springer.

[9] B.M. Trager (1976), Algebraic factoring and rational function integration, in: *Proceedings of SYMSAC '76* 219–226, ACM Press.

[10] J.C. Faugère (2010), FGb: a library for computing Gröbner bases, in: *Mathematical Software – ICMS 2010* 84–87.

[11] V.V. Titouan Coladon, A. Joux (2018), OpenF4, `https://github.com/nauotit/openf4`.

Spiegelungen am Kreis: A case for a CAS

J. Meyer
(Hameln)

j.m.meyer@t-online.de

Einführung

Man kann nicht nur an Punkten oder Geraden spiegeln, sondern auch an Kurven. Möchte man das tun, so liegt folgendes Verfahren nahe: Man sucht sich zum Originalpunkt P den nächstgelegenen Kurvenpunkt K und führt anschließend eine Punktspiegelung von P an K aus.
Dies Verfahren wird hier für den Fall untersucht, dass die Kurve ein Kreis um den Ursprung O mit dem Radius r ist und dass eine Gerade abgebildet wird. Es ist zu beachten, dass die hier vorgestellte Spiegelung am Kreis etwas anderes bedeutet als die übliche Kreisinversion.

Ermittlung der Kurvenpunkte

Man bekommt den zu P nächstgelegenen Punkt K_n (der Index n steht für „nah“) des Kreises, indem man die Ursprungsgerade mit dem allgemeinen Punkt $X = t \cdot P$ mit dem Kreis schneidet; das Ergebnis ist $t = \frac{r}{\sqrt{P \cdot P}}$ und damit

$$K_n = \frac{r \cdot P}{\sqrt{P \cdot P}}.$$

Der Kreispunkt K_f (der Index steht für „fern“) mit der größten Entfernung zu P ist $K_f = -K_n$. Die jeweiligen Bildpunkte sind

$$Q_n = 2 \cdot K_n - P$$

und

$$Q_f = 2 \cdot K_f - P = -2 \cdot K_n - P \quad \text{(Abb. 1).}$$

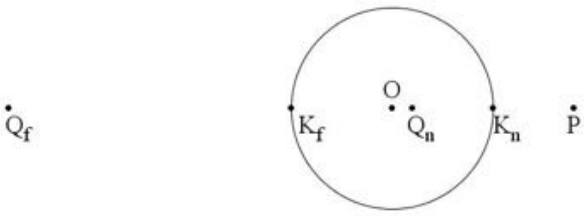


Abbildung 1: *Spiegelung am Kreis*

Nun durchlaufe P eine Gerade. Da beide Achsen jeweils auf sich abgebildet werden, kann man als Geradengleichung $x = x_0$ mit $x_0 > 0$ annehmen. Der allgemeine Punkt der Gerade ist $P(t) = \begin{pmatrix} x_0 \\ t \end{pmatrix}$, und

$$Q_n(t) = \left(\frac{2r}{\sqrt{x_0^2 + t^2}} - 1 \right) \cdot \begin{pmatrix} x_0 \\ t \end{pmatrix}$$

und

$$Q_f(t) = \left(\frac{-2r}{\sqrt{x_0^2 + t^2}} - 1 \right) \cdot \begin{pmatrix} x_0 \\ t \end{pmatrix}$$

durchlaufen Kurven.

Nullstellen

Betrachten wir

$$Q_n(t) = \left(\frac{2r}{\sqrt{x_0^2 + t^2}} - 1 \right) \cdot \begin{pmatrix} x_0 \\ t \end{pmatrix}.$$

Stets führt $t = 0$ zu einer Nullstelle, nämlich zu

$$Q_n(0) = \left(\frac{2r}{x_0} - 1 \right) \cdot \begin{pmatrix} x_0 \\ 0 \end{pmatrix}.$$

Es gibt weitere Nullstellen, wenn der Vorfaktor

$$2r - \sqrt{x_0^2 + t^2}$$

verschwindet; dann liegen diese Nullstellen im Ursprung.
Ist $2r < x_0$, so schneidet die Kurve die x-Achse nur für $t = 0$ (Abb. 2; die dicken Punkte sind Wendepunkte und werden weiter unten erläutert).

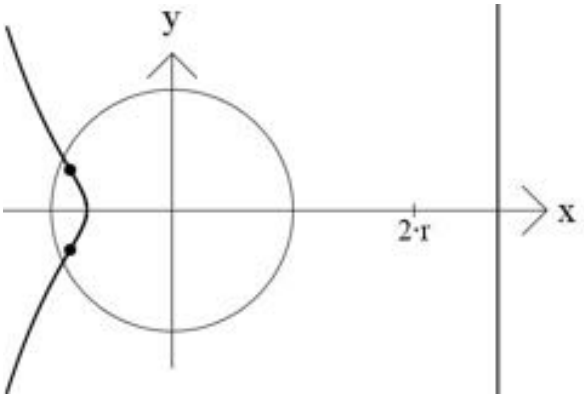


Abbildung 2: *Nur eine Nullstelle*

Ist $0 < x_0 < 2r$, so schneidet die Kurve die x-Achse außerdem für

$$t = \pm\sqrt{4r^2 - x_0^2};$$

die Kurve hat im Ursprung einen *Doppelpunkt* (Abb. 3).

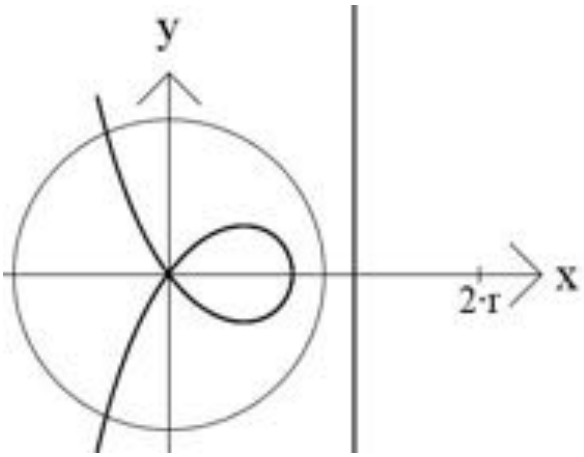


Abbildung 3: *Ein Doppelpunkt*

Ist $x_0 = 2r$, so hat der allgemeine Kurvenpunkt die Gestalt

$$Q_n(t) = \left(\frac{2r}{\sqrt{4r^2 + t^2}} - 1\right) \cdot \begin{pmatrix} 2r \\ t \end{pmatrix}.$$

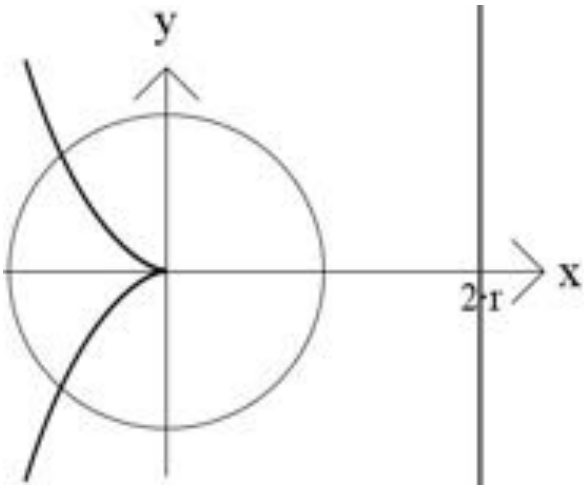


Abbildung 4: *Eine Spitze*

Die Tangente durch $Q_n(t)$ und den Ursprung hat die Steigung $\frac{t}{2r}$; sie hat für $t \to 0$ den Grenzwert 0. Daher hat die Kurve im Ursprung eine *Spitze* (Abb. 4).

Zur Gleichung

Es war

$$Q_n(t) = \left(\frac{2r}{\sqrt{x_0^2 + t^2}} - 1\right) \cdot \begin{pmatrix} x_0 \\ t \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

Mit $T := t^2$ bekommt man aus der ersten Zeile die Gleichung

$$\frac{x}{x_0} + 1 = \frac{2r}{\sqrt{x_0^2 + T}},$$

woraus sich nach Quadrieren T ermittelt. Die quadrierte zweite Zeile

$$y^2 = (\frac{x}{x_0})^2 \cdot T$$

liefert die *quartische* Gleichung

$$(x + x_0)^2 \cdot \left(x^2 + y^2\right) = 4r^2x^2.$$

Man sieht, dass es auch beim Einsatz eines CAS einige mathematische Ideen braucht.
An der Gleichung ist zu erkennen, dass es maximal vier Nullstellen geben kann. Das liegt daran, dass auch die *Fernpunkte* $Q_f(t)$ diese Gleichung erfüllen. Die Kurve der Fernpunkte liefert allerdings wenig Variation (Abb. 5, die auch die weiter unten behandelten Wendepunkte zeigt).

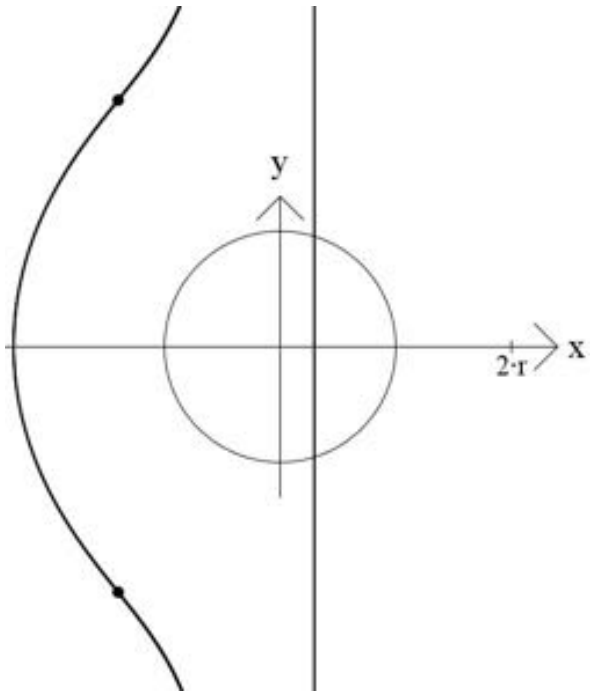


Abbildung 5: *Die Kurve der Fernpunkte*

Exkurs zum Krümmungskreismittelpunkt

Es sieht so aus, als hätten manche Kurven Wendepunkte (wie etwa Abb. 2 oder Abb. 5 zeigen). Dies soll hier näher untersucht werden.

Gegeben sei eine Kurve mit dem allgemeinen Punkt $K(t)$. Man bekommt den *Krümmungskreismittelpunkt* für den Parameter a, indem man die Kurvennormalen zu $K(a)$ und zu $K(a+h)$ miteinander schneidet und anschließend h gegen 0 laufen lässt.

Die Kurvennormale zu $K(a)$ hat den allgemeinen Punkt

$$X = K(a) + \lambda \cdot K'(a)^{\perp},$$

und der allgemeine Punkt X der Kurvennormale zu $K(a+h)$ erfüllt die Gleichung

$$X \cdot K'(a+h) = K(a+h) \cdot K'(a+h) \quad \text{(Hesseform)},$$

was zu

$$\lambda = \frac{(K(a+h) - K(a)) \cdot K'(a+h)}{K'(a+h) \cdot K'(a)^{\perp} - \underbrace{K'(a) \cdot K'(a)^{\perp}}_{0}}$$

$$= \frac{\frac{K(a+h)-K(a)}{h} \cdot K'(a+h)}{\frac{K'(a+h)-K'(a)}{h} \cdot K'(a)^{\perp}}$$

führt, das für $h \to 0$ gegen

$$\frac{K'(a) \cdot K'(a)}{K''(a) \cdot K'(a)^{\perp}}$$

strebt. Damit erhält man den Krümmungskreismittelpunkt

$$K(a) + \frac{K'(a) \cdot K'(a)}{K''(a) \cdot K'(a)^{\perp}} \cdot K'(a)^{\perp}.$$

Ist $K''(a) \cdot K'(a)^{\perp} = 0$ und $K'(a) \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, so ist $K(a)$ ein *Wendepunkt*.

Für welche Parameter liegt ein Wendepunkt vor?

Es war

$$Q_n(t) = \left(\frac{2r}{\sqrt{x_0^2 + t^2}} - 1 \right) \cdot \begin{pmatrix} x_0 \\ t \end{pmatrix}.$$

Die Wendepunktsbedingung führt mit CAS-Hilfe auf die Gleichung

$$\left(x_0^2 - 2t^2\right) \cdot \sqrt{x_0^2 + t^2} - 2r x_0^2 = 0,$$

woraus mit $z := x_0^2$ und $T := \frac{2t^2}{z}$ einerseits die Forderung

$$(1 - T) \cdot \sqrt{z + \frac{z \cdot T}{2}} = 2r$$

und damit $0 < T < 1$ sowie andererseits nach Quadrieren die kubische Gleichung $g(T) = d$ mit

$$g(T) := T^3 - 3T$$

und

$$d := \frac{8 \cdot r^2}{z} - 2$$

folgt. Hier ist das Lösungsverhalten übersichtlich (Abb. 6).

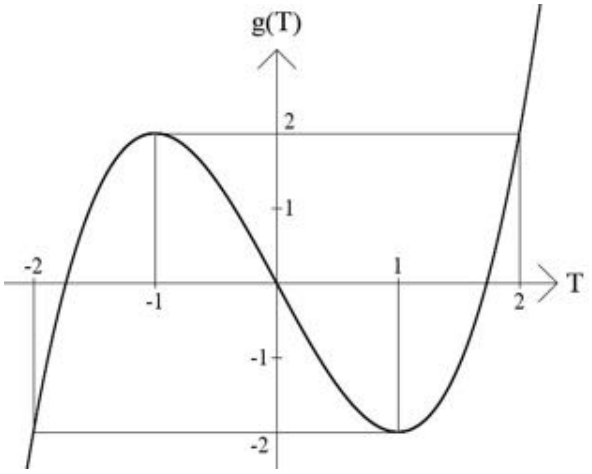


Abbildung 6: *Graph zur kubischen Gleichung*

Ist $0 < T < 1$, so muss $-2 < d < 0$ bzw. $0 < 4r^2 < z$ sein; eine notwendige Bedingung für die Existenz von Wendepunkten ist also $2r < x_0$. Die Lösungen von $T^3 - 3T = d$ sind bekanntlich durch

$$T = 2 \cdot \cos\left(\frac{\arccos(\frac{d}{2})}{3} + k \cdot 120^{\circ} \right)$$

gegeben; für unsere Zwecke ist $0 < T < 1$ notwendig. Die dicken Punkte in Abb. 2 sind die so berechneten Wendepunkte. Auch hier sieht man, dass die mathematischen Überlegungen auch mit Einsatz eines CAS nicht trivial werden.

Die Wendepunkte der Fernkurve

Die Wendepunktsbedingung führt auf

$$\left(x_0^2 - 2t^2\right) \cdot \sqrt{x_0^2 + t^2} + 2r x_0^2 = 0$$

und mit $z := x_0^2$ und $T := \frac{2t^2}{z}$ auf

$$(T - 1) \cdot \sqrt{z + \frac{z \cdot T}{2}} = 2r$$

und somit auf die notwendige Bedingung $T > 1$. Die kubische Gleichung ist dieselbe wie bei den Nahpunkten. Ist $|d| > 2$, so muss man in der Lösungsformel nur cos durch cosh und arccos durch arcosh ersetzen. Es gibt stets Wendepunkte.

Abschlussbemerkungen

Hier bieten sich sofort viele Anschlussfragen an, etwa nach der Kurve, auf der alle Wendepunkte liegen, oder danach, welche Kurven sich ergeben, wenn man nicht eine Gerade, sondern eine Parabel am Kreis spiegelt (Abb. 7 zeigt das Bild einer nach rechts geöffneten Parabel) oder einen anderen Kreis (Abb. 8 zeigt das Bild eines nach rechts verschobenen Kreises).

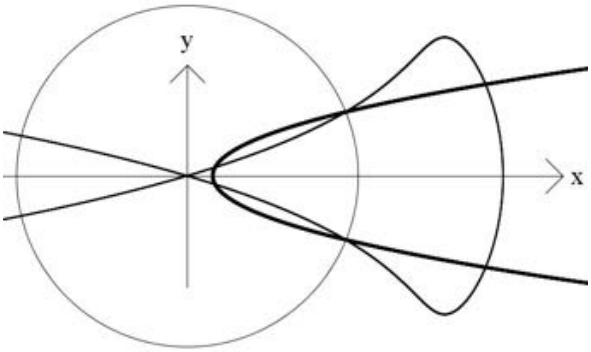


Abbildung 7: *Spiegelung einer Parabel*

Auch wenn die Kurvengleichungen höhergradig werden und nicht mehr mit elementaren Mitteln untersucht werden können, kann der unvermutete Formenreichtum der Kurven zum Erstaunen führen.

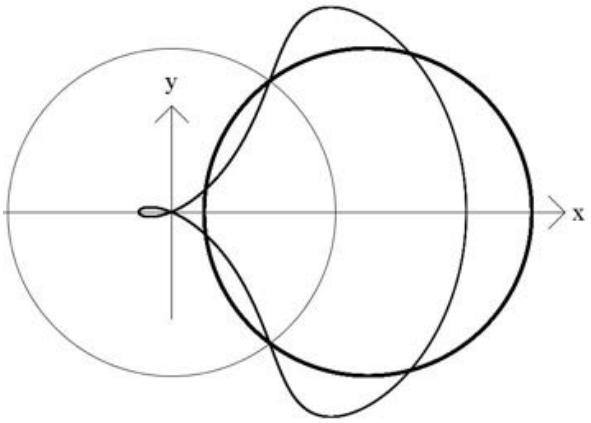


Abbildung 8: *Spiegelung eines Kreises*

SFB/TRR 195 Symbolic Tools in Mathematics and their Application (Part 2/5)

Group and Representation Theory

"Group and Representation theory" is one of the five core areas in the SFB-TRR. The use of computational methods has a long trandition in this area, especially in dealing with finite simple groups. According to Aschbacher, when faced with a problem about general finite groups, it nowadays seems best to reduce the problem to a question about simple groups and groups closely related to simple groups. The classification of finite simple groups (one of the greatest achievements of 20th century mathematics) then supplies an explicit list of groups which can be studied in detail using the effective description of the groups. One of the long term goals is to create something like an ATLAS which serves two main purposes: on the one hand it collects useful information about simple groups (about ordinary and modular irreducible representations, maximal subgroups and so on) in a systematic and easily accessible (electronic) way; on the other hand, it provides the software tools for performing experiments with these data which are essential in theoretical investigations. Thus, this ATLAS will substantially enhance and go beyond the famous Cambridge ATLAS of finite groups (which contains tables for individual groups, like the 26 sporadic simple groups) because it will also contain algorithms and computer programs, drawn from various systems (e.g., GAP, CHEVIE) and integrated within the new OSCAR system, for dealing with infinite families of finite simple groups (like the various families of finite groups of Lie type).

The projects in the SFB concerned with group and representation theory all operate at this close interplay between theoretical developments and new software. They focus on questions related to the subgroup structure of finite simple groups (Hiss), the determination of modular characters (Malle), unipotent subgroups of groups of Lie type (Geck, Malle), Lusztig's theory of character sheaves and the computation of generic character tables (Geck, Lübeck), and trivial source character tables of finite groups (Lassueur). In almost each of these projects, the application of new theoretical methods leads to new data concerning finite simple groups; conversely, the systematic generation of data in substantial examples is extremely helpful in shaping the general theory and—in the spirit of Aschbacher—establishing theorems on general finite groups (e.g., concerning counting conjectures on characters of finite groups).

Meinolf Geck (Stuttgart)

Nonlinear algebra at the MPI MiS Leipzig

F. Arici, Y. Ren

francesca.arici@mis.mpg.de
yue.ren@mis.mpg.de

Overview

The theory, algorithms, and software of linear algebra are familiar tools across mathematics, engineering and other sciences. This ubiquity masks the recent growth of *nonlinear* algebra in mathematics and its applications. The proliferation of nonlinear methods, notably for systems of multivariate polynomial equations, has been fueled by recent theoretical advances, efficient software, and an increased awareness of these tools.

The research group on Nonlinear algebra at the Max Planck Institute for Mathematics in the Sciences (MPI MiS) was established in March 2017. It currently consists of one director (Bernd Sturmfels), three affiliated junior group leaders (Christiane Görgen, Mateusz Michalek, and André Uschmajew), as well as 16 Postdocs, 6 PhD students and 6 long term visitors.

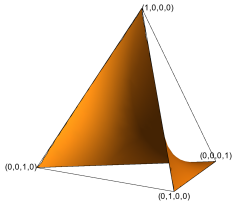
The group focuses on fundamental problems in algebra, geometry and combinatorics that are relevant for nonlinear models. This involves algebraic geometry (complex and real), commutative algebra, combinatorics, polyhedral geometry, and more. On the application side, we are interested in statistics, optimization and the life sciences.

Due to the nature of our research, the use of mathematical software is very prominent. Notably, our group is actively involved in the development of HOMOTOPYCONTINUATION.JL [4], MACAULY2 [8] and SINGULAR [7].

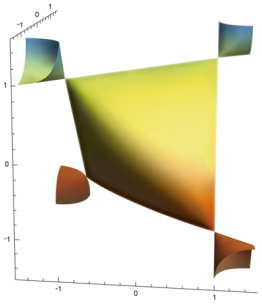
Research topics

The mathematical research that is carried out in our group spans across a broad range of topics, of which we give a short overview below. The diversity of expertise and research interests within the group has been fostering frutiful inter- and intradisciplinary collaborations, united by the common interests for non-linear methods and their applications to mathematics and the applied sciences.

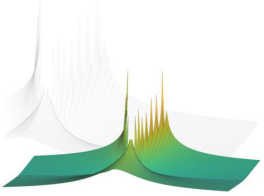
Algebraic statistics uses tools from algebra, combinatorics and geometry to study statistical models. One of the main guiding principles in this area is that statistical models can be thought of as algebraic varieties. This approach leads to new insight about their properties and to new algorithms in statistics based on computer algebra. These results are relevant for topics like conditional inference, phylogenetic models and maximum likelihood estimates. Of particular interest for our group are *Chain Event Graphs* [6].



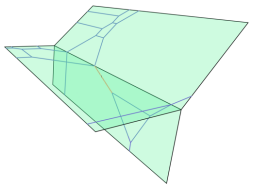
Computational and numerical algebraic geometry [12, 2] brings back the explicit computational core of algebraic geometry to the forefront of research using the computational power of computers and dedicated software. As a result, we are granted access to an ever expanding range of examples for theoretical exploration and industrial application.



Polynomial optimization. Several problems in engineering and in the sciences require minimizing polynomial functions. Though finding this minimum is in general a very hard problem, quite often the solution can be approximated efficiently through a technique known as semidefinite programming [3]. At the heart of this method lies Hilbert's 17th problem: understanding which nonnegative polynomials can be written as sum of squares [10].



Tensor optimization [9] focuses on solving high-dimensional problems using low-rank approximation. Such problems arise from PDEs in quantum chemistry or statistics. In the last decade, new tensor decomposition formats have been proposed that allow for Riemannian optimization on smooth manifolds. Tensors have also recently been applied in Deep Learning [5].



Tropical geometry [11] studies piecewise linear structures in the language of algebraic geometry. These structures arise naturally from polynomials and from max/min-plus functions alike. As such, it enjoys applications in algebraic geometry, such as enumerative geometry or numerical algebraic geometry, or related to optimization, such as biology and economics.

Activities

The research group distinguishes itself through the abundance of activities organised inside and around the institute. In the first 18 months its existence, in addition to a weekly seminar, regular joint seminars with Berlin, and joint reading groups with Magdeburg, the group has co-organized or hosted the following events:

- Computing in tropical geometry
  `11–12 May 2017`
- Interactions between algebra and the sciences
  `27 May 2017`
- Reading group on real algebraic geometry
  `3–7 July 2017`
- Algebraic statistics day, `10 November 2017`
- Open source computer algebra research
  `11–12 December 2017`
- Optimization day, `12 February 2018`
- TAGS: Linking topology to algebraic geometry and statistics
  `19–23 February 2018`
- Mathematical biology day, `9 April 2018`
- Combinatorics day, `23 May 2018`
- Introduction to algebraic statistics
  `25–27 May 2018`
- Macaulay2 workshop, `4–8 June 2018`
- A Tropical Panorama, `9–10 July 2018`
- Summer School on Numerical computing in algebraic geometry
  `13–17 August 2018`
- NoGAGS 2018 — North German Algebraic Geometry Seminar 2018
  `8–9 November 2018`

Finally, the research group is also one of the nodes in the *Mathematics of Data* initiative [1], recently established at the MPI MiS. Recognising the ubiquity of data across various disciplines and the broad scientific expertise present at the MPI MiS, the initiative aims at development of mathematically rigorous and efficient techniques for the integration of large data sets. Recently, the initiative has been further strengthened by the arrival at the institute of Guido Montúfar, who holds an ERC Starting Grant on *Deep Learning*.

References

[1] MATHEMATICS OF DATA, an MPI MiS initiative. More information available at `https://www.mis.mpg.de/math-of-data/`.

[2] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25. SIAM, 2013.

[3] G. Blekherman, P. A. Parrilo, and R. R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

[4] P. Breiding and S. Timme. HOMOTOPYCONTINUATION.JL — A julia package for solving systems of polynomial equations by numerical homotopy continuation. Available at `https://www.juliahomotopycontinuation.org/`.

[5] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[6] R. A. Collazo, C. Goergen, and J. Q. Smith. *Chain Event Graphs*. CRC Press, 2018.

[7] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 4-1-0 — A computer algebra system for polynomial computations. Available at `https://www.singular.uni-kl.de`.

[8] D. R. Grayson and M. E. Stillman. MACAULAY2 — A software system for research in algebraic geometry. Available at `https://www.math.uiuc.edu/Macaulay2/`.

[9] W. Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42 of *Springer series in computational mathematics*. Springer, Heidelberg, 2012.

[10] J.-B. Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2010.

[11] D. Maclagan and B. Sturmfels. *Introduction to tropical geometry*, volume 161. American Mathematical Soc., 2015.

[12] H. Schenck. *Computational algebraic geometry*, volume 58. Cambridge University Press, 2003.

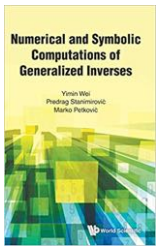
Berufungen

Prof. Dr. Ghislain Fourier hat zum 24. Juli 2018 den Lehrstuhl B für Mathematik (Algebra) an der RWTH Aachen übernommen. Sein Arbeitsgebiet liegt im Zusammenspiel von Darstellungstheorie, Algebraischer Geometrie und Kombinatorik.

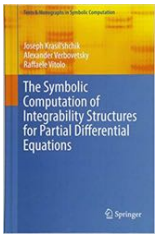
Dr. Andreas-Stephan Elsenhans (Universität Paderborn) hat zum 1. September 2018 eine Professur für Computeralgebra an der Universität Würzburg angenommen.

Publikationen über Computeralgebra

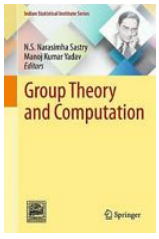
Neuerscheinungen:



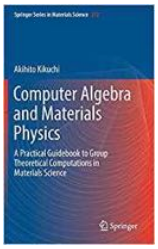
Yimin Wei, Predag Stanimirovic, Marco Petkovic,
Numerical and Symbolic Computations of Generalized Inverses,
World Scientific Publ., 2018,
470 Seiten,
ISBN 978-9813238664



Joseph Krasilshchik, Alexander Verbovetsky, Raffaele Vitolo,
The Symbolic Computation of Integrability Structures for Partial Differential Equations,
Springer Verlag 2018, 263 Seiten,
ISBN 978-3319716541



N.S. Narasimha Sastry, Manoj Kumar Yadav (Hrsg.),
Group Theory and Computation,
Springer Verlag, 2018, 206 Seiten,
ISBN 978-9811320460



Akihito Kikuchi,
Computer Algebra and Materials Physics,
Springer Verlag, 2018, 159 Seiten,
ISBN 978-3319942254

Karl Josef Fuchs, Simon Plangg,
Computer Algebra Systeme in der Lehrer(innen)bildung,
WTM Verlag, 2018, 110 Seiten,
ISBN 978-3959870849

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.

Joppe W. Bos, Arjen K. Lenstra (Hrsg.)

Topics in Computational Number Theory Inspired by Peter L. Montgomery

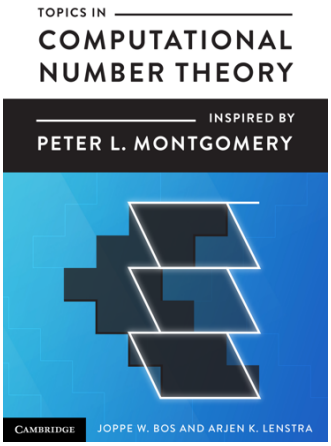
Cambridge University Press, Cambridge 2017, 266+ix Seiten, ISBN 978-1-107-10935-3, € 81,84

Der Name „Montgomery" ist wahrscheinlich am besten bekannnt für die Montgomery Multiplikation, ein Verfahren zur schnellen Multiplikation in großen endlichen Körpern. Der voliegende Band über Themen aus dem Gebiet der algorithmischen Zahlentheorie, die durch die Arbeiten von Peter L. Montgomery inspiriert sind, zeigt aber, dass die Beiträge Montgomerys wesentlich breiter gestreut sind und eine Vielzahl von Bereichen sehr wesentlich befruchtet haben. Nachdem in den Kapiteln 2 und 3 die Montgomery Multiplikation sowohl von der Software-Perspektive als auch von der Hardware-Implementierung her intensiv betrachtet wurden, wendet sich dieser Sammelband jenen weiteren Themen zu. Jedes davon wird von einem oder mehreren einschlägigen Spezialisten ausführlich und verständlich dargestellt.

In Kapitel 4 wird dabei die Montgomery-Leiter diskutiert, die schnelle Arithmetik auf elliptischen Kurven erlaubt. Für moderne kryptographische Verfahren, die auf elliptischen Kurven beruhen, ist eine möglichst effiziente Umsetzung der Vielfachenbildung eines Punkts essentiell. Ausgehend von besonders geeigneten Normalformen elliptischer Kurven, insbesondere der Montgomery Kurven, basiert die Montgomery-Leiter auf einem *add-double-add* Verfahren, das auf einer einfachen Verdopplungsformel beruht.

Ein weiteres Gebiet wird in den Kapiteln 5 und 6 angesprochen, nämlich das Zahlkörpersieb zur schnellen Faktorisierung großer ganzer Zahlen. Ältere Implementierungen verwenden dabei das *Multiple Polynomial Quadratic Sieve*, das hochparallele Implementierungen erlaubt und 1994 zur Lösung der 129-stelligen Herausforderung aus der August 1977 Ausgabe des *Scientific American* führte. Ein zentraler Beitrag ist dabei die schnelle Konstruktion bestimmter Polynome für das Zahlensieb, die von Montgomery wegweisend vereinfacht wurde.

Im Endergebnis führt das Zahlensieb auf riesengroße, dünn besetzte lineare Gleichungssysteme über dem Körper $\mathbb{F}_2$. Auch für die Lösung dieser System hat Montgomery einen wichtigen Vorschlag beigetragen, nämlich den Block Lanczos Algorithmus, der in Kapitel 7 des Bands vorgestellt wird. Dieser Algorithmus hat für neue Rekordfaktorisierungen bis etwa 2005 eine wichtige Rolle gespielt.



Für Faktorisierungsmethoden, die die Glattheit von Gruppenordnungen ausnutzen, also die Tatsache, dass manche Gruppenordnungen nur vergleichsweise kleine Primfaktoren besitzen, kann man sowohl die Pollardsche $(p-1)$-Methode als auch die EC Faktorisierung mit Hilfe der FFT (Fast Fourier Transformation) sehr effizient implementieren. Die Beiträge Montgomerys zu diesem Thema sind der Gegenstand von Kapitel 8. Das neunte und letzte Kapitel behandelt schließlich das hochaktuelle Gebiet der paarungsbasierten ECC (Elliptic Curve Cryptography). Die geschickte Berechnung solcher Paarungen stellt eine schwierige Aufgabe dar, die durch eine Reihe von Tricks wie die schnelle modulare Inversion, die simultane Inversion, und die Verwendung projektiver Koordinaten optimiert werden kann.

Trotz der Beiträge sehr unterschiedlicher Autoren oder Autorenteams ist das Buch sehr homogen und gut lesbar geschrieben. Ein Vielzahl von Überblicken über aktuelle algorithmische Methoden und die zugehörigen ausführlichen Literaturverweise machen es für jeden Praktiker, der aktuelle computeralgebraische Verfahren für die Kryptographie und die Kryptoanalysis implementieren will, zu einem unverzichtbaren Werkzeug und Nachschlagewerk. Der Kaufpreis von etwa 80 Euro erscheint voll und ganz gerechtfertigt.

Martin Kreuzer (Passau)

Promotionen in der Computeralgebra

Ulrike Faltings: On the characters of the Sylow 2-subgroups of $F_4(2^n)$ and decomposition numbers

Betreuer: Gunter Malle (Kaiserslautern)

Zweitgutachter: Gerhard Hiss (Aachen)

März 2018

Abstract: Ziel der Arbeit war die Bestimmung von Zerlegungszahlen der Chevalley-Gruppen vom Type $F_4(2^n)$ in nicht-definierender Charakteristik. Frau Faltings hat dazu für alle Gruppen dieses Typs die Konjugiertenklassen und große Teile der komplexen Charaktertafel einer Sylow 2-Untergruppe U sowie die Fusion der Klassen in die Klassen der Chevalley-Gruppen bestimmt. Damit hat sie die Charaktere von U induziert und in die unipotenten Charaktere zerlegt und so projektive Charaktere erhalten. Zur Durchführung der Rechnungen hat sie vorhandene Algorithmen aus GAP verwendet, aber diese wo notwendig auch erweitert.

David Geis: On the combinatorics of Tits arrangements

Betreuer: Michael Cuntz (Hannover)

Zweitgutachter: Piotr Pokora (Krakau), Bernhard Mühlherr (Giessen)

Mai 2018

Abstract: In this thesis, we study simplicial arrangements of hyperplanes and certain generalizations of those which are called Tits arrangements. Among other things, we give a combinatorial characterization of pairs of Tits arrangements differing by only one hyperplane. Moreover, special emphasis is put on simplicial line arrangements whose characteristic polynomials have only real roots: we give a classification of such arrangements having multiplicity at most four. Similarly, we show that such an arrangement with multiplicity five consists of at most forty lines. Furthermore, we show how to construct certain sporadic arrangements via duality from finite reflection groups.

Christian Neurohr: Efficient integration on Riemann surfaces & applications

Betreuer: Steffen Müller (Oldenburg)

Zweitgutachter: Nils Bruin (Vancouver), Florian Heß (Oldenburg)

Mai 2018

Abstract: This thesis is concerned with developing, analyzing and implementing algorithms that compute periods of compact Riemann surfaces given by an affine equation. Various methods for numerical integration are employed (and compared) to approximate integrals of holomorphic differentials along paths on a Riemann surface to high numerical precision. In this way, objects of central interest associated to a Riemann surface are obtained, such as period matrices and the Abel-Jacobi map, leading to numerous applications which are motivated by problems in arithmetic geometry. The author presents two different sets of algorithms, depending on the defining equation for the Riemann surface. Firstly, algorithms for the special case of superelliptic curves of the form $y^m = p(x)$ with separable $p \in \mathbb{C}[x]$ are based on a joint paper with Pascal Molin. Secondly, the more general class of Riemann surfaces defined by $f = 0$ with geometrically irreducible $f \in K[x, y]$ over a number field K. All algorithms have been implemented in MAGMA and will be available in its next distribution as a Riemann surface class.

Cornelia Rottner: Algorithmic Methods for Mixed Hodge Modules

Betreuer: Mathias Schulze (Kaiserslautern)

Zweitgutachter: Francisco J. Castro Jimenez (Sevilla)

Juli 2018

Abstract: In modern algebraic geometry solutions of polynomial equations are studied from a qualitative point of view using highly sophisticated tools such as cohomology, D-modules and Hodge structures. The latter have been unified in Saito's far-reaching theory of mixed Hodge modules, that has shown striking applications including vanishing theorems for cohomology. A mixed Hodge module can be seen as a special type of filtered D-module, which is an algebraic counterpart of a system of linear differential equations. We present the first algorithmic approach to Saito's theory. To this end, we develop a Groebner basis theory for a new class of algebras generalizing PBW-algebras.

The category of mixed Hodge modules satisfies Grothendieck's six-functor formalism. In part these functors rely on an additional natural filtration, the so-called V-filtration. A key result of this thesis is an algorithm to compute the V-filtration in the filtered setting. We derive from this algorithm methods for the computation (extraordinary) direct image functors under open embeddings of complements of pure codimension one subvarieties. As side results we show how to compute vanishing and nearby cycle functors and a quasi-inverse of Kashiwara's equivalence for mixed Hodge modules.

Describing these functors in terms of local coordinates and taking local sections, we reduce the corresponding computations to algorithms over certain bifiltered algebras. It leads us to introduce the class of so-called PBW-reduction-algebras, a generalization of the class of PBW-algebras. We establish a comprehensive Groebner basis framework for this generalization representing the involved filtrations by weight vectors.

PCA 2018

St. Petersburg, Russland, 16.04. – 20.04.2018

pca-pdmi.ru/2018/

Die elfte Ausgabe der internationalen Konferenz “Polynomial Computer Algebra” fand wie üblich am Euler Institut für Mathematik statt, das zum Steklov Institut in St. Petersbrug gehört. Die Tagung unter der erfahrenen Leitung von Nikolai Vasiliev bietet alljährlich die Möglichkeit, viele russische Kollegen aus der Computeralgebra zu treffen, ihre Arbeitsgebiete kennen zu lernen und sich über aktuelle Entwicklungen auszutauschen. Naturgemäß stehen dabei Computeralgebramethoden für die Gruppentheorie, für Differentialgleichungen und für die mathematische Physik im Vordergrund. Doch auch die klassischen Bereiche wie die algorithmische algebraische Geometrie, algorithmische kommutative Algebra und algorithmische Zahlentheorie kamen nicht zu kurz.

Eine spezielle Sitzung in Gedenken an den verstorbenen langjährigen Teilnehmer Sergei Baranov fand überwiegend in russischer Sprache statt. In den Hauptvorträgen beschäftigten sich u.a. Vladimir Gerdt (Dubna) mit der Implementierung der Thomas-Zerlegung differentieller Systeme, Lorenzo Robbiano (Genua) mit dem Buch „Computational Linear and Commutative Algebra” und Mark Spivakovsky (Toulouse) mit der Pierce-Birkhoff Vermutung. Das wissenschaftliche Programm wurde abgerundet durch ein Kammermusikkonzert am Euler Institut, eine Stadtrundfahrt mit Besichtigung der Isaakskathedrale sowie einem Konferenzdinner mit russischem Essen und Gesängen. Ein Blick auf die Teilnehmerlisten der vergangenen Jahre zeigt, dass die meisten Teilnehmer, die diese Konferenz erst einmal entdeckt haben, ihr fortan die Treue halten und jedes Jahr gerne wieder kommen.

Martin Kreuzer (Passau)

Teilnehmer PCA 2018

ACA 2018

Santiago de Compostela, Spanien, 18.06. – 22.06.2018

www.usc.es/regaca/aca2018

Die Konferenzserie „Applications of Computer Algtebra (ACA)” bringt es mittlerweile auf stolze 24 Ausgaben, deren letzte im Juni in der Hauptstadt Galiziens durchgeführt wurde. Zwei der vier Hauptvorträge, nämlich die von James H. Davenport (Bath) und Vijay Ganesh (Waterloo) beschäftigten sich dabei mit den aktuellen Entwicklungen zu den Interaktionen zwischen Computeralgebrasystemen und den SAT-Lösern aus der Logik. Desweiteren berichteten Laureano Gonzalez-Vega (Santander) über Experimente mit und Anwendungen von reellen algebraischen Kurven und Flächen sowie Dingkang Wang (Beijing) über automatische Beweise in der Geometrie.

Konferenzfoto ACA 2018

Daneben gab es sage und schreibe 12 spezielle „Sessions" von denen jeweils bis zu fünf gleichzeitig stattfanden. Diese bestanden aus drei, vier, oder auch bis zu 23 Vorträgen. Wegen der hohen Parallelität war den Teilnehmern somit eine entsprechend hohe Selektivität abverlangt. Deshalb kann der Berichterstatter im Wesentlichen auch nur Positives über die Session „Algorithms for Zero-Dimensional Ideals" berichten, die erstmals stattfand und sich eines guten Zuspruchs sowie hoher Qualität erfreute. Wenn man sich zum Beispiel das Programm des letzten Konferenztages ansieht, das bis Mittag währte und auch nur zwei weniger gut besuchte Sessions beinhaltete, wird schnell ein Trend erkennbar, dass immer mehr Vorträge parallel gelegt werden und die Teilnehmer nur noch für einen Teil des Konferenzzeitraums anreisen.

Die Konferenz war mustergültig organisiert. Neben einem Konferenzausflug in die Küstenregion Galiziens war den Teilnehmern auch ein beeindruckendes Konferenzdinner in einem der ältesten Hotels der Welt, dem *Hostal de los Reyes Catolicos*, vergönnt.

Martin Kreuzer (Passau)

FLoC 2018, SC^2 2018

Oxford, Vereinigtes Königreich, 01.07. – 19.07.2018

`www.sc-square.org/CSA/workshop3.html`

Die Federated Logic Conference (FLoC) (Webseite: `https://www.floc2018.org`) ist nicht nur eine, sondern eine Ansammlung verschiedener Konferenzen und Workshops zur allgemeinen Thematik der Verbindungen zwischen der mathematischen Logik und der Informatik. Um genau zu sein: die FLoC 2018 brachte neun größere internationale Konferenzen und 74 Workshops an einen Ort, nämlich die altehrwürdige Universität von Oxford.

Obwohl in diversen anderen Konferenzen und Workshops durchaus für die Computeralgebra relevante Vorträge stattfanden, wie zum Beispiel über eine Formalisierung des LLL-Algorithmus oder pseudo-deterministische Algorithmen, soll hier hauptsächlich über den dritten und letzten Workshop des FETOPEN-CSA Projekts SC^2 (Satisfiability Checking and Symbolic Computation) berichtet werden, der am 11.7.2018 abgehalten wurde und zwölf Vorträge umfasste. Das Thema dieses EU-Projekts und des Workshops war die Verbindung zwischen den Algorithmen der Computeralgebra, insbesondere dem Lösen polynomialer Gleichungssysteme über $\mathbb{F}_2$ und den SAT-Solvern, die die Erfüllbarkeit logischer Formeln mit hunderttausenden von Variablen und Millionen von Klauseln schnell testen können.

Der Hauptvortrag *„Hard Combinatorial Problems: A Challenge for Satisfiability"* von Ilias Kotsireas (Waterloo) zeigte bereits die Mächtigkeit dieser Verbindung auf, denn mittels moderner SAT-Solver konnten extrem harte kombinatorische Probleme untersucht und teilweise gelöst werden. Weitere Vorträge beschäftigten sich mit der Implementierung dieser Verbindung in Computeralgebrasystemen, mit neuen Anwendungsmöglichkeiten der SAT-Solver in der Computeralgebra, und mit neuen Kalkülen, die die guten Eigenschaften des Clause Learnings aus der SAT-Welt mit denen des Buchberger-Algorithmus zu kombinieren versuchen.

Mit knapp 2000 (!) Teilnehmern war die FLoC ein wahres Großereignis, so dass es wirklich beeindruckt, wie reibungslos und effizient die Veranstaltungen abliefen. Die beachtlichen Konferenzgebühren (450 GBP für jeden der beiden Konferenzblöcke, dazu 85 GBP pro Workshop) erlaubten den Organisatoren aber eine durchweg professionelle und großzügige Betreuung und die herrliche Stadt Oxford machte ein besonderes touristisches Rahmenprogramm überflüssig.

Martin Kreuzer (Passau)

EACA 2018

Zaragoza, Spanien, 04.07. – 06.07.2018

`eventos.unizar.es/15634`

The 16th edition of the biennial conference EACA was held in Zaragoza in early July and brought together around 50 researchers in computational algebra and applications. True to the title, the 5 main lectures focused around topics from computational algebra and its applications. More on the theoretic side were the talks of Ana Romero on computational aspects of spectral sequences and of Fatemeh Mohammadi on toric degenerations of Grassmanians and flag varieties from tropical geometry. The talk of Anne Frühbis-Krüger on parallel computing in algebraic geometry opened the scope toward a new generation of algorithms. Applications outside of mathematics were in focus of the other two talks, where Javier Arsuaga spoke on algebraic analysis of cancer genomes and Javier Lobillo gave an insight into Error Correcting Codes and the McEliece Cryptosystem.

Konferenzfoto EACA 2018

Taking advantage of the immediately preceding 94th sage days, many young and very young researchers attended both meetings and took the opportunity to contribute a talk in one of the parallel sessions of EACA. This contributed to a very good mix of different topics and to lively interaction between experience and new ideas.

Jorge Martín Morales (Zaragoza)

ISSAC 2018

New York City, USA, 16.07. – 19.07.2018

`www.issac-conference.org/2018`

Die ISSAC (ausführlich: International Symposium on Symbolic and Algebraic Computation) ist eine der wichtigsten internationalen Konferenzreihen über algebraisches und symbolisches Rechnen (kurz: über Computeralgebra), die seit ihren Ursprüngen eine enge Verbindung zur Fachgruppe Computeralgebra pflegt. In diesem Jahr fand sie in New York statt.

Mit 140 Teilnehmerinnen und Teilnehmern war die ISSAC wieder gut besucht, was vielleicht auch dem attraktiven Standort zu verdanken war. Die lokale Leitung übernahm Victor Pan (City University of New York), und als General Chair fungierten Manuel Kauers (Johannes Kepler Universität Linz) und Alexey Ovchinnikov (City University of New York). Alle Entscheidungen über die Vorträge lagen wir immer bei dem Programmkomitee, das von Éric Schost (University of Waterloo, Canada) geleitet wurde. Nach dem durch das Programmkomitee organisierte Begutachtungsprozess wurden 47 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. Außerdem wurden 15 Poster und diesmal nur drei Software-Demos präsentiert. Für weitere Details, Titel und Autoren sei auf die wie immer gut organisierte Homepage der Konferenz verwiesen.

New York

Bei dem ISSAC Business Meeting, an dem wie immer alle Konferenzteilnehmer (mit Stimmrecht) teilnehmen durften, wurde der Ort für die ISSAC 2020 bestimmt. Die Wahl fiel allerdings nicht allzu schwer, da (vor dem Hintergrund einer zurückgezogenen Bewerbung) der einzige Vorschlag aus Kalamata in Griechenland kam. Gemäß der immer wieder kompliziert erscheinenden Bylaws der ISSAC wurde außerdem ein neues Mitglied im Steering Committee gewählt, um die durch das turnusgemäße Ausscheiden des Vorsitzenden Dan Roche entstandene Vakanz aufzufüllen. Unter vier Kandidaten setzte sich hierbei Éric Schost (University of Waterloo, Canada) durch. Unter den sonst behandelten Themen stach eines als untypisch heraus: ob man gemäß eines Vorschlags aus den Reihen der Teilnehmer künftig die USA wegen der durch die Regierung verhängten Einreisesperre für verschiedene Nationalitäten als Standort ausschließen sollte. Hier gelangte das Business Meeting zu der Auffassung, dass solche Entscheidungen jeweils der Weisheit der Wähler überlassen werden sollen, die ja in jedem Fall über den Austragungsort abstimmen.

Auch diesmal gab es eine ganze Reihe von Preisen, die 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, die sich diesmal komplett aus den ohnehin bestehenden Poster- und Software-Komitees der Konferenz rekrutierten.

Der Preis für das beste Poster ging an:

Autoren: Simon Telen, Bernard Mourrain und Marc Van Barel.
Titel: Truncated Normal Forms for Solving Polynomial Systems.

Als beste Software-Demo wurde ausgezeichnet:

Autoren: Victor Magron und Mohab Safey El Din.
Titel: RealCertify: a Maple Package for Certifying Nonnegativity.

In Ermangelung von Bildern der Konferenz und zur Vermeidung einer Bleiwüste sei hier eines der Wahrzeichen des Austragungsortes abgebildet. Dies mag auch daran erinnern, dass die Konferenz im Herzen Manhattans, schräg gegenüber des Empire State Buildings stattfand.

Die nächste ISSAC-Konferenz findet vom 15. bis 18. Juli 2019 in Beijing statt.

Gregor Kemper (München)

CAP Days 2018

Siegen, 28.08. – 31.08.2018

`homalg-project.github.io/capdays-2018/`

CAP (Categories, Algorithms, Programming) is a software project for constructive category theory written in GAP. The main idea is to use categorical abstraction as a computational tool: CAP facilitates both the realization of specific instances of categories and the implementation of generic categorical algorithms.

The this year's CAP Days workshop took place at the University of Siegen, lasted four days and brought together 12 users, developers, and newcomers.

On the first day, several participants gave talks about features and applications of CAP in both mathematical and physical research. Topics included the usage of CAP in the GAP package QPA for quivers and path algebras (Øystein Skartsæterhagen), connections of CAP with machine learning and string theory (Martin Bies), and categorical algorithms for some mixed Tate motives of moduli spaces (Daniel Juteau).

Participants of the CAP Days 2018.

On the second day, all participants eagerly solved the exercises given in tutorial worksheets. The exercises served both as an introduction to CAP and constructive homological algebra, and inspired even the advanced users to try out new ideas and tricks in their various applications. These worksheets are freely available as Jupyter notebooks on the conference's webpage.

On the third and fourth day, everybody participated in a coding sprint. After the tutorials, even first time users were able to either integrate CAP in their research computations or make significant contributions to core system itself, like the implementation of the subobject classifier, or the creation of the Karoubi envelope as a category constructor.

We are grateful for the financial support by the SFB-TRR 195 "Symbolic Tools in Mathematics and their Application" that made this workshop possible.

Sebastian Gutsche und Sebastian Posur (Siegen)

CASC 2018

Lille, Frankreich, 17.09. – 21.09.2018

www.casc.cs.uni-bonn.de/2018

This year the 20th International Workshop on Computer Algebra in Scientific Computing (CASC) conference was held in Lille (France). The local organizing committee headed by François Boulier, François Lemaire and Adrien Poteaux of the University of Lille did a superb job in doing all of the local arrangements. For the first time François Boulier joined Vladimir P. Gerdt and Werner M. Seiler as a general chair of the conference.

For the second time the conference started with a day of tutorials on Monday. In two morning sessions Nathalie Verdière (Le Havre) gave an overview on "Applications of computer algebra in the identifiability study and the parameter estimation. Application in neurosciences". In two afternoon sessions Marc Morena Maza (University of Western Ontario) lectured on "Computing limits with the RegularChains and PowerSeries libraries: from rational functions to topological closures".

On Tuesday the nice lecture halls of the LILLIAD Learning Center Innovation in the center of the "Scientific City" of the University of Lille became available for the conference. The morning session started with an overview talk on sparse polynomial factorization by Michael Monagen (Simon Fraser University) in an ad hoc change of program, as unfortunately the invited talk by Jean-Guillaume Dumas had to be cancelled for medical reasons. Three sessions on polynomial solving, methods for real and integer solving as well as tropical methods and presenting 8 accepted papers formed the major scientific part on Tuesday. In the business meeting on Tuesday evening the location of next CASC, the Plekhanov Russian University of Economics in the city center of Moscow was presented. A shift of the traditional September date of CASC to the last week of August was suggested.

On Wednesday morning two sessions on software and system aspects and on algebraic methods presenting 5 papers were the content of the scientific program. The tradition of CASC of having exciting excursions was instantiated by an excursion and social dinner to the French Flanders on Wednesday afternoon and evening.

The second invited talk on "Quantum information and quantum computing: an overview and some mathematical aspects" by Maurice R. Kibler (Lyon) started the scientific program on Thursday. The scientific program on Thursday was completed by two sessions presenting 6 papers on groups and complexity related topics and on applications in physics. A sponsor session presenting new features in Maple 2018 was part of the afternoon program on Thursday.

Participants of CASC at the excursion to Cassel in the French Flanders, the "preferred village of the French 2018"

The conference was concluded by two sessions presenting work on symbolic-numeric methods (2 papers) and papers involving combinatorial aspects (3 papers) on Friday morning.

The program of the CASC 2018 may be found at the web site http://www.casc.cs.uni-bonn.de/2018/index.php/program

The online version of the Proceedings (LNCS 11077) is available via the conference web site at http://www.casc.cs.uni-bonn.de/2018/index.php/proceedings.

Andreas Weber (Bonn)

Annual meeting SFB TRR 195

Tübingen, 24.09. – 28.09.2018

www.math.uni-tuebingen.de/arbeitsbereiche/geometrie/annual-meeting-sfb-trr-195-1

Dies war das zweite Jahrestreffen des SFB TRR 195 "Symbolic tools in Mathematics and their Application". Das Programm war eine gelungene Mischung aus längeren Übersichtsvorträgen und Kurzbeiträgen diverser Art. Eingeladen war eine Reihe von hochkarätigen Rednern: Nils Bruin, Alicia Dickenstein, Derek Holt, Ion Nechita und Bernd Sturmfels. Zusammen mit den Vorträgen von Mitgliedern des TRR umfassten die Themengebiete sämtliche Disziplinen, die auch den TRR selbst ausmachen: Gruppen- und Darstellungstheorie, algebraische Geometrie, kommutative und nicht-kommutative Algebra, tropische und polyhedrale Geometrie, Zahlentheorie, Zufallsmatrix-Theorie und freie Probabilität. Darüberhinuas gab es kurze Snapshots, in denen zum Beispiel Doktoranden und Doktorandinnen sich der Herausforderung stellen konnten, in weniger als 10 Minuten ihr Arbeitsgebiet zu präsentieren. In mehreren Vorträgen wurde über neueste Fortschritte bei OSCAR berichtet, dem integrierten Computer-Algebra-System, dessen Entwicklung das zentrale Software-Projekt des TRR ist. Im Begleitprogramm gab es eine "women's night", eine Wanderung am Mittwochnachmittag sowie eine öffentliche Aufführung des Films "The Discrete Charm of Geometry" von Ekaterina Eremenko.

Meinolf Geck (Stuttgart)

Hinweise auf Konferenzen

Nikolaus conference 2018

Aachen, 07.12. – 08.12.2018

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.

Computeralgebra-Tagung der Fachgruppe

Kassel, 16.05. – 18.05.2019

www.fachgruppe-computeralgebra.de

In Fortsetzung der erfolgreichen Tagungen 2003, 2005, 2009, 2012, 2014, 2017 in Kassel und 2007 in Kaiserslautern führt die Fachgruppe im Mai 2019 wieder eine derartige Tagung in Kassel durch. Ziel ist es, ein Forum zu bieten, das es erstens Nachwuchswissenschaftlern ermöglicht, ihre Ergebnisse vorzustellen, andererseits aber auch einige Hauptvortragende zu gewinnen, die Übersichtsvorträge über wichtige Gebiete der Computeralgebra und über Computeralgebra-Software geben sollen.

MEGA 2019

Madrid, Spanien, 17.06. – 21.06.2019

eventos.ucm.es/12097/detail/mega-2019.html

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

AAG 2019

Bern, Schweiz, 09.07. – 13.07.2019

mathsites.unibe.ch/siamag19

The purpose of the SIAM Activity Group in Algebraic Geometry is to bring together researchers who use algebraic geometry in industrial and applied mathematics. „Algebraic geometry“ is interpreted broadly to include at least: algebraic geometry, commutative algebra, noncommutative algebra, symbolic and numeric computation, algebraic and geometric combinatorics, representation theory, and algebraic topology. These methods have already seen applications in: biology, coding theory, cryptography, combustion, computational geometry, computer graphics, quantum computing, control theory, geometric design, complexity theory, machine learning, nonlinear partial differential equations, optimization, robotics, and statistics. We welcome participation from both theoretical mathematical areas and application areas not on this list which fall under this broadly interpreted notion of algebraic geometry and its applications.

ISSAC 2019

Peking, China, 15.07. – 18.07.2019

www.issac-conference.org/2019

The International Symposium on Symbolic and Algebraic Computation (ISSAC) is the premier conference for research in symbolic computation and computer algebra. ISSAC 2019 will be the 44th 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 center-piece of contributed research papers.
ISSAC 2019 will be held on 15-18 July 2019, at Beihang University, Beijing, China

ACA 2019

Montreal, Kanada, 16.07. – 20.07.2019

aca2019.etsmtl.ca

The 25th Conference on Applications of Computer Algebra (ACA) will be held in Montréal, Canada. This event will take place from Tuesday July 16 to Saturday July 20, 2019. École de technologie supérieure will host the international conference for a second time, 10 years after ACA 2009.
The ACA conference series is devoted to promoting all kinds of computer algebra applications, and encouraging the interaction of developers of computer algebra systems and packages with researchers and users (including scientists, engineers, educators, and mathematicians).
Topics include, but are not limited to, computer algebra in the sciences, engineering, communication, medicine, pure and applied mathematics, education, business and computer science.

DMV-Jahrestagung 2019

Karlsruhe, 23.09. – 26.09.2019

dmv2019.math.kit.edu

Die DMV-Jahrestagung 2019 findet vom 23. bis 26. September 2019 am Karlsruher Institut für Technologie (KIT) statt. Organisatoren sind Christian Wieners und Wilderich Tuschmann. Weitere Informationen auf der Konferenzwebseite.

GI-Jahrestagung 2019

Kassel, 23.09. – 27.09.2019

www.informatik2019.de

Die Jahrestagung INFORMATIK 2019 wird vom 23. – 26. September 2019 an der Universität Kassel stattfinden. Die Gesellschaft für Informatik wurde 1969 gegründet. In Kassel soll der 50. Geburtstag mit einem attraktiven Programm für und mit „Jung und Alt“ aus Wissenschaft und Praxis gewürdigt werden. Das Leitthema der Tagung wird sein „50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft“. Gegenüber früheren Jahrestagungen wird sich die Tagungsstruktur ändern: Themen, die aktuell besondere Aufmerksamkeit in Wissenschaft, Praxis und Gesellschaft erfahren, werden im Rahmen vorgegebener thematischer „Tracks“ behandelt, die jeweils von ausgewiesenen Expertinnen und Experten aus der Informatik-Community gestaltet werden.

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

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

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: .....

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
Institut für Algebra und Zahlentheorie
Universität Stuttgart
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
Fachbereich Campus Minden der FH Bielefeld
Artilleriestr. 9, 32427 Minden
0571-8385-258
christoph.thiel@fh-bielefeld.de
https://www.fh-bielefeld.de/minden/ueber-uns/personenverzeichnis/christoph-thiel

Stellvertretende Sprecherin:
Prof. Dr. Anne Frühbis-Krüger
Institut für Algebraische Geometrie
Welfengarten 1, 30167 Hannover
0511-762-3592
fruehbis-krueger@math.uni-hannover.de
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
https://math.uni-paderborn.de/ag/klueners/

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/

TI-Nspire™ macht Schule.

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

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

Nehmen Sie Fahrt auf!

education.ti.com/de/rover

Texas Instruments