März 2021

Computeralgebra Rundbrief

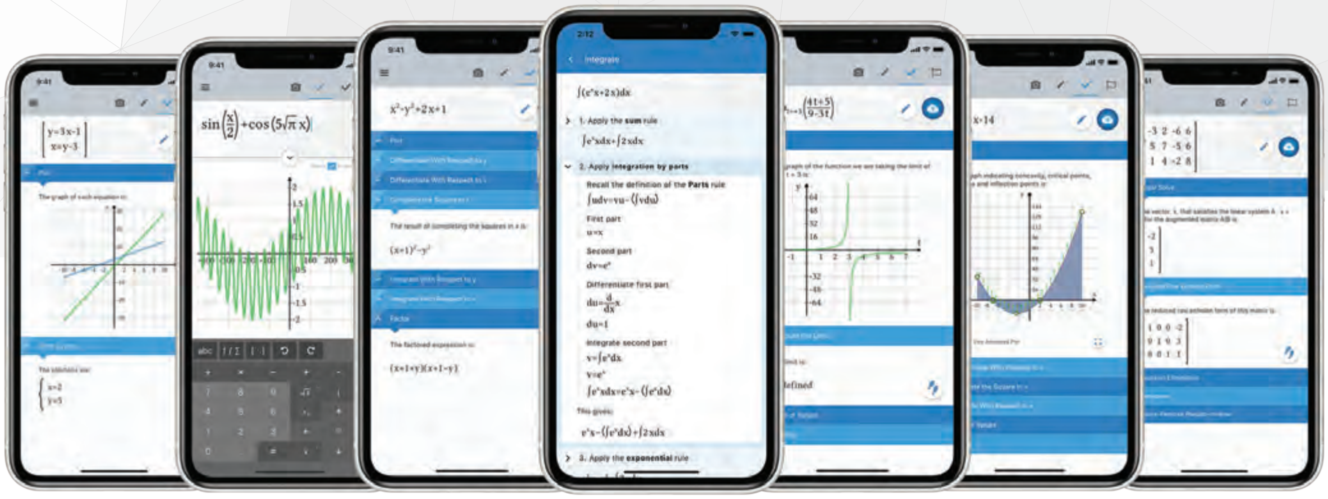
> Ausgabe 68

- Industrietagung der Fachgruppe
- Terao's freeness conjecture with computer algebra
- Computational Origami
- Square-1

www.fachgruppe-computeralgebra.de

GI DMV GAMM

Maple™ Calculator



Auch verfügbar in Deutsch

Warum warten?

Versuchen Sie es doch mal!

Available on the App Store | GET IT ON Google Play

www.maplesoft.com/calculator

Möchten Sie unsere Maple Calculator app ausprobieren, haben aber kein Maple?

Holen Sie sich eine 15-tägige Testlizenz und schauen Sie, was die Calculator app in Verbindung mit Maple leisten kann!

www.maplesoft.com/trial

Mathe auf Ihrem Smartphone!

Maple Calculator ist ein leistungsstarker Rechner zum Lösen mathematischer Aufgaben und ein vielseitiges Lernwerkzeug, das Ihnen Resultate, 2D- und 3D-Diagramme und sogar Schritt-für-Schritt-Lösungen liefert! Ob Sie nun einfache Berechnungen durchführen oder an mathematischen Problemen auf Universitätsniveau arbeiten, Maple Calculator kann alles. Verwenden Sie Maple Calculator, um Algebra-Probleme zu lösen, Ableitungen oder Integrale von Funktionen zu berechnen, Matrixmanipulationen durchzuführen, 2D- und 3D-Diagramme zu erkunden und vieles mehr!

- Mathe-Probleme mit einem Kamera-Klick oder dem eingebauten Mathe-Editor eingeben.
- Integrale finden, Polynome faktorisieren, Matrizen invertieren, Gleichungssysteme und gewöhnliche Differentialgleichungen lösen, und vieles mehr.
- Erhalten Sie Schritt-für-Schritt-Lösungen für eine Vielzahl von mathematischen Problemen, einschließlich dem Lösen von Gleichungssystemen, dem Finden von Grenzwerten, Ableitungen und Integralen sowie dem Durchführen von Matrixoperationen.
- Plots erstellen, um das Verständnis zu vertiefen.
- Mathe-Aufgaben in Maple hochladen, um die gesamte Leistung von Maple zu nutzen.
- Fehler vermeiden, die beim Abschreiben von mathematischen Ausdrücken zu Maple von Hand auftreten können.

Computeralgebra-Rundbrief

Nr. 68 März 2021

Fachgruppe Computeralgebra

Inhaltsverzeichnis

Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und der GAMM (verantwortlicher Redakteur: Dr. Fabian Reimers car@mathematik.de)

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

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

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

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

Mitteilungen der Sprecher

Liebe Mitglieder der Fachgruppe Computeralgebra,

zuerst einmal hoffen wir, dass Sie, Ihre Familien und auch die Menschen in Ihrem Arbeitsumfeld weiterhin gut durch die Unbilden der Corona-Pandemie gekommen sind, und wünschen Ihnen vor allem Gesundheit, Anpassungsfähigkeit an die Situation und gute Nerven.

Die Zeit seit dem Erscheinen des letzten Rundbriefs hielt für uns alle wieder große Herausforderungen bereit. Die noch immer grassierende Pandemie erzwang ein Jahr voller Online-Lehre und für die Hochschullehrer unter uns die schwierige Suche nach pandemiegeeigneten Prüfungsformen – und es bleibt spannend. Gerne können Sie uns gerade zum Thema Online-Lehren und Online-Prüfen (in der Schule ebenso wie in der Hochschule) Miniaturen mit Computeralgebrabezug zusenden – egal ob Anekdoten, warnende Pannen oder Best-Practice Beispiele. Diese werden gerne im nächsten Rundbrief als neue Corona-Miniaturen mit aufgenommen.

Nach all den Tagungsverschiebungen bei den Aktivitäten der Fachgruppe fand am 27. November 2020 doch zumindest noch eine Veranstaltung statt, ein Online-Workshop zu Computational Algebra mit Vorträgen von Christian Eder, Kathlen Kohn, Anna-Laura Sattelberger und Rainer Sinn. Mit über 50 Teilnehmern hat die Resonanz auch die Veranstalter positiv überrascht, was uns dazu bewogen hat, in dem Zeitraum, in dem eigentlich die Fachgruppentagung veranstaltet worden wäre, nochmals einen Online-Workshop anzubieten. Mehr dazu finden Sie in der Rubrik "Tagungen der Fachgruppe".

Bei der Industrietagung sehen wir weiterhin einen wichtigen Fokus auf dem Knüpfen von Kontakten zwischen Forschern, Industrie und Nachwuchs. Dies hatte uns in 2020 zu einer Verschiebung um ein Jahr bewogen. Nun jedoch, da selbst der Herbst 2021 nicht sicher Präsenztagungen erlauben wird, haben wir uns schweren Herzens für ein Hybridformat entschieden, damit wir uns flexibel an die dann aktuelle Pandemielage anpassen können. Auch hierzu findet sich eine Ankündigung in der bereits genannten Rubrik.

Natürlich darf die Computeralgebra in diesem Heft auch nicht zu kurz kommen. Unter "Themen und Anwendungen" findet sich diesmal ein Artikel zu Arrangements von Hyperebenen. In der Rubrik "Computeralgebra in der Hochschule" beschäftigt sich dann ein Artikel mit dem Zauberwürfelverwandten Square-I, Gruppentheorie und Algorithmen dazu.

Anlässlich der Bewilligung der zweiten Phase des SFB-TRR 195 haben wir die neu hinzugekommenen Projektleiter gebeten, die von Ihnen vertretenen, nun beginnenden Projekte in der Rubrik "Berichte über Arbeitsgruppen" kurz vorzustellen. Wir gratulieren zu diesem Erfolg und wünschen weiterhin gutes Gelingen.

Wir wünschen Ihnen eine angenehme und anregende Lektüre.

Anne Frühbis-Krüger *Gregor Kemper*

Tagungen der Fachgruppe

Industrietagung der Fachgruppe

Oldenburg, 29.09.-30.09.2021 (Fachtagung)
27.09.-28.09.2021 (Schule)

www.fachgruppe-computeralgebra.de/industrietagung2021

Nach zehnjähriger Pause veranstaltet die Fachgruppe Computeralgebra wieder eine Industrie-Fachtagung – eigentlich geplant für 2020, aber wie so vieles angesichts der Pandemie erst verschoben und nun als Hybridveranstaltung organisiert.

Im Zuge der großen Herausforderungen an digitale Sicherheit durch neue Technologien wie KI, hoch performante binäre Systeme und Quantentechnologien haben wir uns entschieden, den Fokus auf Kryptografie zu legen.

Aktuell befindet sich die NIST[1] in der zweiten Runde der Standardisierung der neuen kryptografischen Algorithmen, Post-Quanten sichere Kryptografie genannt. Diese werden besondere Herausforderungen an Implementationen stellen und damit Experten brauchen, die tiefes Verständnis entsprechender Bereiche aus Mathematik, Algorithmik und Computing mitbringen – ein lohnendes, sich rasch entwickelndes Feld mit Zukunftsperspektiven für junge Absolventen und Promovierte.

Die Fachtagung soll Vertretern der Industrie die Möglichkeit geben, ihre aktuellen Arbeitsgebiete darzustellen und Kontakte zu Forschern und wissenschaftlichem Nachwuchs zu knüpfen, während die Schule interessiertem Nachwuchs einen Einblick in die mit der 'neuen' Kryptografie verbundenen Themen geben und sie auf die Thematik der Fachtagung vorbereiten soll. Die Inhalte der Vorträge reichen von rein algorithmischen Fragen über Computing-Aspekte bis zu Kontext-Themen, wie z. B. der Notwendigkeit von Post-Quanten Kryptografie zur Realisierung von dynamischen QKD-Netzwerken.

Wir haben industrielle Vertreter aus verschiedensten Bereichen eingeladen, um dem Nachwuchs die Weite der Einsatzmöglichkeiten außerhalb der konventionellen wissenschaftlichen Pfade nahe zu bringen.
Bereits fest zugesagt haben einen Vortrag aus dem außeruniversitären Bereich:

- Christian Wirth, www.privacybyblockchaindesign.com
- Peter Nonnenmann, DHBW, www.karlsruhe.dhbw.de

Wir bitten für diese kostenlose Veranstaltung um Registrierung per Mail an die Adresse

caindustrietagung2021@uol.de

Xenia Bogomolec (Hannover)
Anne Frühbis-Krüger (Oldenburg)
Florian Heß (Oldenburg)

[1]National Institute of Standard and Technologies, US Department of Commerce

Online-Workshop

21.05.2021, 13:00 – 17:00 Uhr

www.fachgruppe-computeralgebra.de/ca2021

Der erste Online-Kurzworkshop der Fachgruppe im Spätjahr 2020, dessen Idee als ein kleines Trostpflaster für die wegen der Pandemie entfallenen Minisymposia geboren wurde, wurde sehr positiv aufgenommen. Mit mehr als 50 Teilnehmern fand er ein vergleichsweise großes Publikum. Daher haben wir uns entschlossen, auch anläßlich der Verschiebung der Fachgruppentagung von 2021 auf 2022 wieder einen Online-Workshop mit 4 Vorträgen anzubieten.

Bereits fest zugesagt haben einen Vortrag:

- Simon Brandhorst (Saarbrücken)
- Jean-Philippe Labbé (Berlin)
- Timo de Wolff (Braunschweig)

Wir bitten für diese kostenlose Veranstaltung um Registrierung per Mail an die Adresse

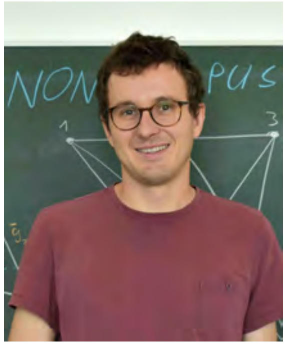
caworkshop2021@uol.de

Michael Cuntz (Hannover)
Anne Frühbis-Krüger (Oldenburg)
Sabrina Gaube (Oldenburg/Hannover)

Investigating Terao's freeness conjecture with computer algebra

Mohamed Barakat (University of Siegen)
Lukas Kühne (Max Planck Institute for Mathematics in the Sciences, Leipzig)

mohamed.barakat@uni-siegen.de
lukas.kuhne@mis.mpg.de



Introduction

In computational mathematics one often encounters the problem of scanning (finite but) large sets of certain objects. Here are two typical scenarios:

- Searching for a counter-example of an open conjecture among these objects.
- Building a database of such objects with some of their invariants.

A database is particularly useful when the questions asked are relational, i.e., involve more than one object. Recognized patterns and questions which a database answers affirmatively may lead to working hypotheses or even proofs by inspection.

This article describes an instance of this principle in an on-going research project in which we investigate hyperplane arrangements through a database [BK19b, BBJ19, BK21]. This is in part joint work with Reimer Behrends, Christopher Jefferson, and Martin Leuner.

The theory of hyperplane arrangements naturally lies at the crossroads of algebra, combinatorics, algebraic geometry, representation theory and topology. We are specifically interested in the interplay between the combinatorial and algebraic properties of an arrangement motivated by Terao's freeness conjecture, one of the central open conjectures in this area.

Hyperplane arrangements

We briefly give the relevant definitions together with an illustrating example in this section. We refer to the excellent textbook by Orlik and Terao for an in-depth introduction to the theory of hyperplane arrangements [OT92].

Definition 1 An *arrangement of hyperplanes* $\mathcal{A}$ is a finite set of hyperplanes in a vector space V over some field k. We assume that all hyperplanes contain the origin of V. We say that $\mathcal{A}$ is of rank ℓ if V is an ℓ-dimensional vector space.

Example 2 The rank 3 *braid arrangement* $\mathcal{A}_3$ is an arrangement in $\mathbb{R}^3$ defined via the linear equations

$$x = 0, y = 0, z = 0, x - y = 0, x - z = 0, y - z = 0.$$

Figure 1 shows a projectivized picture of $\mathcal{A}_3$, that is a slice of the arrangement with the hyperplane $z = 1$. The hyperplane $z = 0$ is shown as a circular line at infinity.

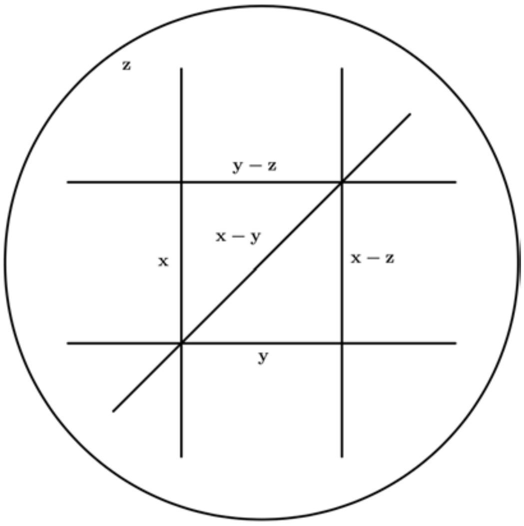


Figure 1: *The braid arrangement* $\mathcal{A}_3$.

Intersection lattice of an arrangement

Combinatorially, an arrangement gives rise to an intersection lattice:

Definition 3 Let $\mathcal{A}$ be an arrangement of hyperplanes. The *intersection lattice* $L(\mathcal{A})$ is defined as the set of all possible intersections of hyperplanes in $\mathcal{A}$:

$$L(\mathcal{A}) := \left\{ \bigcap_{H \in \mathcal{A}'} H \mid \mathcal{A}' \subseteq \mathcal{A} \right\}.$$

Partially ordering $L(\mathcal{A})$ by reverse inclusion yields a partially ordered set (poset).

The poset $L(\mathcal{A})$ is in fact essentially a *matroid*. In general, a matroid is a combinatorial abstraction of linear independence in vector spaces and forests in graphs. Since we will be exclusively working with rank 3 arrangements, we refrain from giving a general definition of a matroid and content ourselves with defining rank 3 matroids:

Definition 4 A matroid M of rank 3 is a bipartite graph[1] connecting atoms and coatoms such that every pair of atoms is jointly adjacent to exactly one coatom.

A matroid is called *representable over the field* k if there exists a rank 3 arrangement $\mathcal{A}$ such that there is a bijection between $L(\mathcal{A}) \setminus \{\emptyset \cup V\}$ and M respecting the poset structure such that hyperplanes and lines are mapped to atoms and coatoms, respectively.

The condition on a matroid demanding that every pair of atoms meets in exactly one coatom stems form the fact that every pair of hyperplanes defines exactly one line, i.e. 1-dimensional subspace.

Example 2 (continued) The intersection lattice of the braid arrangement $\mathcal{A}_3$ is depicted in Figure 2.

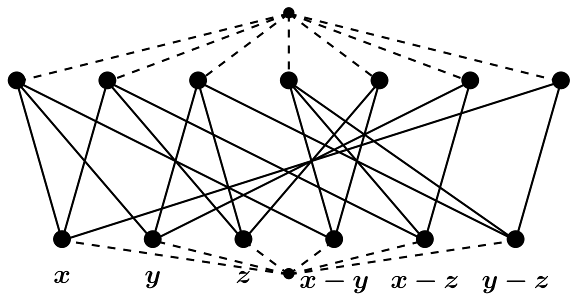


Figure 2: *The intersection* $L(\mathcal{A}_3)$.

Derivation module of an arrangement

Motivated from singularity theory, Saito introduced the notion of a logarithmic derivation of a hypersurface. Terao subsequently pioneered the study of such logarithmic derivation in the special case of hyperplane arrangements [Ter80].

Definition 5 Let $\mathcal{A}$ be an arrangement of hyperplanes in a vector space V over a field k of dimension ℓ. For every $H \in \mathcal{A}$, fix some linear defining equation $\alpha_H \in S := k[x_1 \dots, x_\ell]$. The S-module of *logarithmic derivations* $D(\mathcal{A})$ is defined as

$$D(\mathcal{A}) := \{\theta \in \operatorname{Der} S \mid \theta(\alpha_H) \in \alpha_H S \text{ for all } H \in \mathcal{A}\},$$

where $\operatorname{Der} S$ is the free S-module of all derivations of S.

The arrangement $\mathcal{A}$ is called *free* if $D(\mathcal{A})$ is a free S-module, that is, it admits a basis over S.

The module $\operatorname{Der} S$ is a free module over S generated by all partial derivatives $\partial_{x_1}, \dots, \partial_{x_\ell}$.

Example 2 (continued) The module of logarithmic derivations $D(\mathcal{A}_3)$ of the braid arrangement $\mathcal{A}_3$ is free with a basis

$$\begin{aligned}\theta_1 &:= x\partial_x + y\partial_y + z\partial_z,\\ \theta_2 &:= x^2\partial_x + y^2\partial_y + z^2\partial_z,\\ \theta_3 &:= x^3\partial_x + y^3\partial_y + z^3\partial_z.\end{aligned}$$

It is easy to verify that these derivations are in fact in $D(\mathcal{A}_3)$. We omit further arguments at this point which prove that these derivations are S-independent and in fact generate $D(\mathcal{A}_3)$.

Terao generalized this example by proving that in fact all reflection arrangements turn out to be free. Over the years, remarkable connections were established between the freeness of an arrangement and related arrangement properties. It is however fair to say that it is still quite mysterious to understand which arrangements are free and which are not. This is manifested in the main open conjecture in this area formulated by Terao about 40 years ago:

Conjecture 6 (Terao's freeness conjecture) The freeness of an arrangement $\mathcal{A}$ defined over a fixed field k depends only on its intersection lattice $L(\mathcal{A})$, that is its underlying matroid.

In other words, for two arrangements $\mathcal{A}$ and $\mathcal{A}'$ both defined over the same field k with $L(\mathcal{A}) = L(\mathcal{A}')$ Terao's conjecture asserts that $\mathcal{A}$ is free if and only if $\mathcal{A}'$ is free.

Our strategy to investigate Conjecture 6 in dimension 3 consists of the following three steps:

1. We generate all possibly relevant matroids in some size-range and save them in a database.
2. For each such matroid we determine the (possibly empty) *moduli space* of all representations in a way that can be easily specialized to any field.
3. We compute the nonfree locus within the moduli space of each relevant matroid. That is, all arrangements representing one fixed matroid that are nonfree. A matroid with a nonfree locus that is a proper subset of the moduli space (after specializing to some field) would be a counter-example to Conjecture 6.

A matroid database

Given an arrangement $\mathcal{A}$, one can consider the so-called *characteristic polynomial* $\chi(\mathcal{A}, t)$ which is a combinatorial invariant, i.e., it only depends on the intersection lattice $L(\mathcal{A})$. A celebrated theorem of Terao states that for a free arrangement, the characteristic polynomial $\chi(\mathcal{A}, t)$ factors over the integers [Ter81]. In particular, it suffices to investigate Terao's freeness conjecture for those arrangements, and hence for those matroids with integrally splitting characteristic polynomial.

[1] A bipartite graph is a graph whose vertices can be partitioned into two sets U, V such that every edge joins a vertex in U to one in V.

This result is the starting point for our matroid enumeration efforts. Using a parallelized algorithm in HPC-GAP, the high-performance computing version of GAP, we generated all 815107 simple rank 3 matroids with integrally splitting characteristic polynomial with up to $n = 14$ atoms. The code used for this computation is part of the GAP package MatroidGeneration [BK20]. We stored them in a public ArangoDB database, which can be accessed here: [BK19b]. Figure 3 shows a screen shot of this database.

As a second application, we found through the database that there is exactly one representable rank 3 matroid that is divisionally free but not inductively free with up to 14 atoms, cf. [BK19c].

We invite the reader to use the database for their own matroid-related experiments. We computed several relevant properties of the matroids such as the moduli space, the automorphism group, the Tutte polynomial, and several notions of combinatorial freeness. We are happy to assist researchers interested in including other properties of the matroids for potentially different applications.

How to test matroid representability?

The space of *all* representations (over some unspecified field k) of a rank 3 matroid M is an *affine variety*, namely an affine subvariety $V(I) \subseteq \mathbb{A}^{3n+1}_{\mathbb{Z}}$, where n is its number of atoms.

More precisely, let $\mathbb{A}^{3n+1}_{\mathbb{Z}} := \operatorname{Spec} R[d]$, where $R := \mathbb{Z}[a_{ij} \mid i = 1, \ldots, 3, j = 1, \ldots, n]$ and d a further indeterminate. To describe the ideal I' set $A := (a_{ij}) \in R^{3\times n}$. For a subset $S \subseteq [n] := \{1, \ldots, n\}$ denote by A_S the submatrix of A with columns in S. Further, let $\mathcal{B}(M)$ be the set of triples in $[n]$ that are not contained in some coatom of M (these are the bases of M in classical matroid terminology). Then

$$I := \langle \det(A_D) \mid D \subseteq [n], |D| = 3, D \notin \mathcal{B}(M)\rangle + \left\langle 1 - d \prod_{B\in\mathcal{B}(M)} \det(A_B) \right\rangle \trianglelefteq R[d].$$

It follows that M is representable (over some field k) if and only if $1 \notin I$. This ideal membership problem can be decided by computing a Gröbner basis of I. This is basically the algorithm suggested in [Oxl11].

To make computations feasible, it is an essential step to embed the moduli space $V(I)$ into some smaller dimensional affine space. We used the GAP package ZariskiFrames [BKLH19] to compute such an embedding.

Application to Terao's conjecture

To investigate Terao's freeness conjecture it suffices to consider matroids that satisfy the following conditions:

- They are representable over some field: If a matroid does not admit any representation there is nothing to check.
- They are not inductively free: Inductive freeness is a combinatorial condition on the matroid which ensures that all representations are free arrangements [Ter80].
- They are balanced: Unbalancedness is another combinatorial condition for which it is a priori known whether all representations are free or nonfree.
- They are not uniquely representable: If a matroid only admits a unique representation (up to a coordinate-change and field extensions) it also cannot admit a free and a nonfree representation.

Somewhat surprisingly, it turns out that there are only 9 rank 3 integrally splitting matroids of size up to 14 satisfying all of these conditions. They have 9, 11, 12, or 13 atoms. This already verifies Terao's freeness conjecture for rank 3 arrangements with up to 14 hyperplanes with the above 9 exceptions. Figure 3 shows a screen shot of the query in our database of the matroids satisfying the above conditions of size 14. As shown at the bottom of the figure, this query does not yield any matroid.

Subsequently, we computed the nonfree locus within the moduli space of the 9 remaining matroids. This locus is a closed subvariety of the affine moduli space which we computed using Fitting ideals in GAP using SINGULAR.

We found that over a fixed field all representations are either all free or all nonfree. As previously observed by Ziegler [Zie90], we found new examples of matroids which admit free and nonfree representations over different fields. A new example of this type is the matroid underlying the so-called pentagon arrangement depicted in Figure 4. Projectively, this arrangement consists of the five sides of a regular pentagon, together with its five diagonals and the line at infinity. In [OT92] it is explained that this arrangement is in fact free. Our methods prove that any arrangement representing the same matroid over a field of characteristic not 2 is free. However, the matroid is also representable over fields of characteristic 2 containing the field $\mathbb{F}_4$ with 4 elements and all of these representations are nonfree.

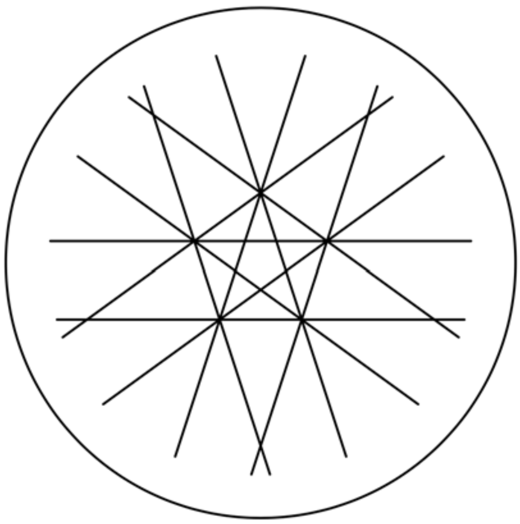


Figure 4: *The pentagon arrangement.*

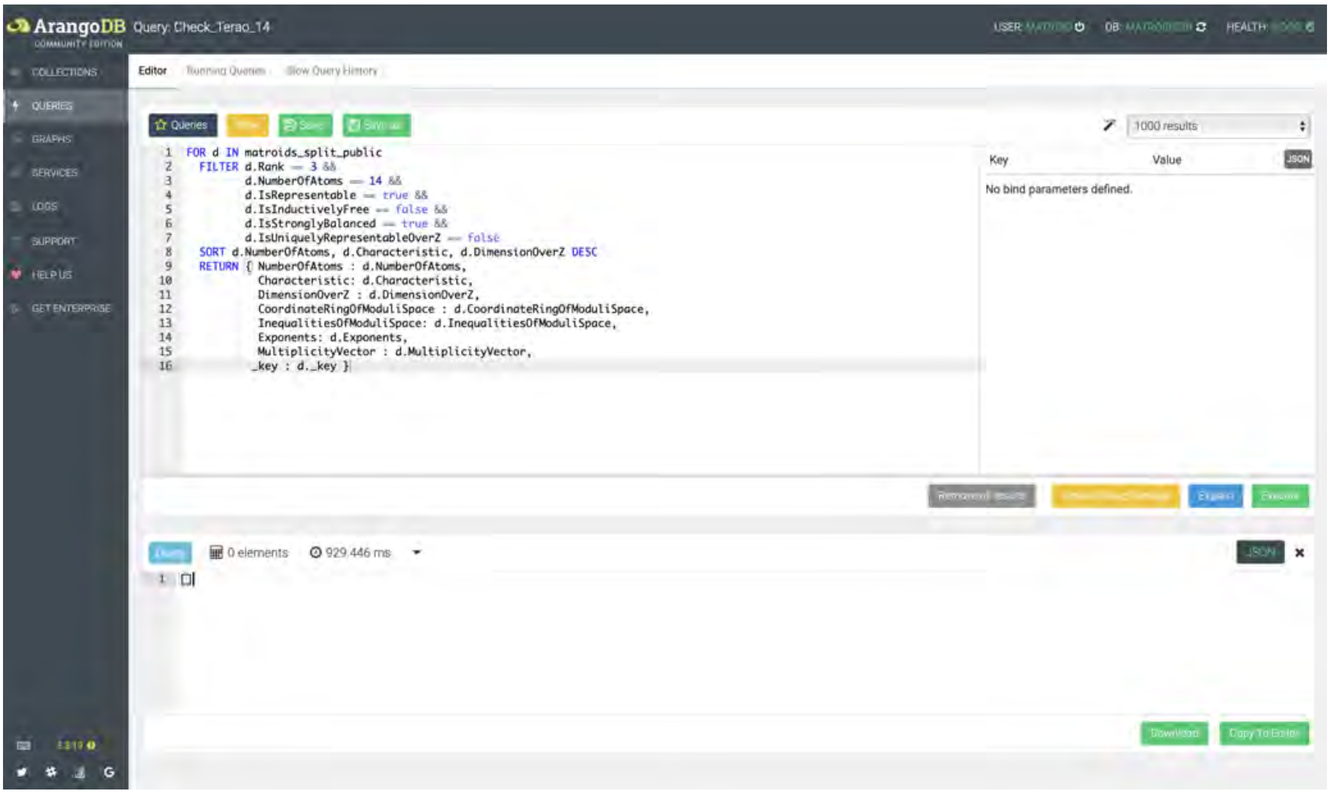


Figure 3: *A screenshot of the matroid database showing the query that proves Terao's freeness conjecture for rank* 3 *arrangements with* 14 *hyperplanes.*

In total we obtained the following main result:

Theorem 7 Terao's freeness conjecture is true for arrangements in dimension 3 with up to 14 hyperplanes in arbitrary characteristic.

Previously, Dimca, Ibadula, and Măcinic verified Terao's conjecture for arrangements with up to 13 hyperplanes in $\mathbb{C}^3$.

References

[BBJ19] Mohamed Barakat, Reimer Behrends, Christopher Jefferson, Lukas Kühne, and Martin Leuner, *On the generation of rank 3 simple matroids with an application to Terao's freeness conjecture*, (`arXiv:1907.01073`), to appear in SIAM J. Discrete Math. 2021.

[BK19b] Mohamed Barakat and Lukas Kühne, `matroids_split_public` – *a database collection for rank* 3 *integrally split simple matroids*, 2019, (`https://homalg-project.github.io/pkg/MatroidGeneration`).

[BK19c] Mohamed Barakat and Lukas Kühne, *Smallest divisionally free not inductively free arrangement*, 2019, (`https://homalg-project.github.io/nb/DivFreeNotIndFree`).

[BK20] Mohamed Barakat and Lukas Kühne, `MatroidGeneration` – *Generate low-rank matroids*, 2017–2020, (`https://homalg-project.github.io/pkg/MatroidGeneration`).

[BK21] Mohamed Barakat and Lukas Kühne, *Computing the nonfree locus of the moduli space of arrangements and Terao's freeness conjecture*, in preparation.

[BKLH19] Mohamed Barakat, Tom Kuhmichel, and Markus Lange-Hegermann, `ZariskiFrames` – *(Co)frames/Locales of Zariski closed/open subsets of affine, projective, or toric varieties*, 2018–2019, (`https://homalg-project.github.io/pkg/ZariskiFrames`).

[DIM19] Alexandru Dimca, Denis Ibadula, and Anca Măcinic, *Freeness for 13 lines arrangements is combinatorial*, Discrete Mathematics **342** (2019), no. 8, 2445 – 2453.

[OT92] Peter Orlik and Hiroaki Terao, *Arrangements of hyperplanes*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 300, Springer-Verlag, Berlin, 1992.

[Oxl11] James Oxley, *Matroid theory*, second ed., Oxford Graduate Texts in Mathematics, vol. 21, Oxford University Press, Oxford, 2011.

[Ter80] Hiroaki Terao, *Arrangements of hyperplanes and their freeness. I*, J. Fac. Sci. Univ. Tokyo Sect. IA Math. **27** (1980), no. 2, 293–312.

[Ter81] Hiroaki Terao, *Generalized exponents of a free arrangement of hyperplanes and Shepherd-Todd-Brieskorn formula*, Invent. Math. **63** (1981), no. 1, 159–179.

[Zie90] Günter M. Ziegler, *Matroid representations and free arrangements*, Trans. Amer. Math. Soc. **320** (1990), no. 2, 525–541.

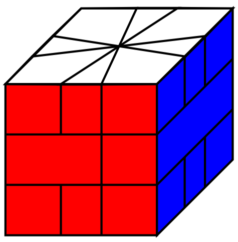
Square-1: Wie man Weihnachten 2020 verbringt

Benjamin Sambale (Hannover)

sambale@math.uni-hannover.de

Jenseits des Zauberwürfels

Im Rahmen meiner Gruppentheorie-Vorlesung beschäftigte ich mich zum wiederholten Male mit Rubiks Zauberwürfel. Auch wenn immer wieder neue mathematische Erkenntnisse, wie die NP-Vollständigkeit von Gottes-Algorithmus für den $n \times n \times n$-Würfel [2], bewiesen werden, ist doch das meiste bereits zu oft gesagt und geschrieben worden. Da ich zur großen Mehrheit derer zähle, die den Zauberwürfel nicht ohne fremde Hilfe lösen konnten, entstand der Wunsch ein anderes, weniger bekanntes Permutationspuzzle selbstständig zu analysieren und zu lösen. Die $n\times n\times n$-Würfel mit $n = 2, 4, 5, \ldots$ (namentlich *Pocket Cube*, *Rubik's Revenge*, *Professor's cube*, ...) sind bezüglich der Mechanik dem Original zu ähnlich (abgesehen davon, dass ich die Lösung bereits kannte). Der Tetraeder *Pyraminx* ist so einfach, dass man ihn mit Glück lösen kann. Meine Wahl fiel daher auf das 1990 von Hršel und Kopský erfundene *Square-1* (im Englischen gibt es die an „Mensch ärgere dich nicht“ erinnernde Redewendung „back to square one“). Im Ausgangszustand ähnelt Square-1 in seiner Würfelform mit verschieden gefärbten Seiten dem Zauberwürfel:



Ober- und Unterseite sind in je vier Ecksteine und vier Kantensteine unterteilt. Im Gegensatz zum Zauberwürfel gibt es keine Seitenmittelsteine (diese sind dort ohnehin irrelevant). Stattdessen bilden die Ecksteine 60°-Sektoren, während die Kantensteine 30°-Sektoren bilden. Die mittlere Ebene ist nur einmal asymmetrisch unterteilt, wodurch sich jeder Zeit Ober- und Unterseite identifizieren lassen (beim Zauberwürfel gelingt dies durch die Seitenmittelflächen, während beim Pocket Cube oder Rubik's Revenge keine solche Identifizierung möglich ist). Die drei Ebenen lassen sich frei und unabhängig drehen. Zusätzlich gibt es die Möglichkeit an einer festen vertikalen Ebene um 180° zu drehen (andere Drehwinkel zwischen 0° und 360° sind zwar zulässig, aber führen unmittelbar in eine Sackgasse). Nach einer solchen Drehung werden einige Eck- und Kantensteine der Oberseite mit der Unterseite ausgetauscht. Dabei verliert das Puzzle seine Würfelgestalt (die mittlere Ebene wird sechseckig).

Im Folgenden möchte ich den kurzweiligen Prozess beschreiben, der mich an Weihnachten 2020 zur Lösung des Square-1 geführt hat. Anstatt der Lektüre sei dem Leser empfohlen sich an diesem oder einem ähnlichen Puzzle selbst zu probieren.

Erste Schritte

Ab jetzt halten wir den Square-1 an der linken Hälfte der mittleren Ebene fest. Dies ist offensichtlich keine wesentliche Einschränkung. Anfangs habe ich nur einige vermeintlich übersichtliche Züge gewagt, um das Puzzle jederzeit rekonstruieren zu können. Dazu eignen sich Kombinationen aus 180°-Drehungen. Zum besseren Verständnis führen wir folgende Bezeichnungen ein:

$$\begin{aligned} u &= \text{30°-Drehung der Oberseite gegen den Uhrzeigersinn,} \\ d &= \text{30°-Drehung der Unterseite gegen den Uhrzeigersinn mit Sicht von oben,} \\ s &= \text{180°-Drehung an der vertikalen Ebene.} \end{aligned}$$

Durch Probieren entdeckt man:

$$\begin{aligned} su^6d^6s &= \text{Ober- und Unterseite werden vollständig getauscht,} \\ su^6su^6su^6 &= \text{Mittlere Ebene verändert die Form.} \end{aligned}$$

Die Drehungen sollen dabei von links nach rechts ausgeführt werden, wobei die Reihenfolge in diesem speziellen Fall noch keine Rolle spielt. Da wir nun die beiden Zustände der mittleren Ebene ineinander überführen

können ohne den Rest zu verändern, werden wir diese Ebene zukünftig vernachlässigen.

Als Nächstes konzentrieren wir uns auf Manöver, die die Würfelgestalt erhalten. Dafür ist es nützlich die Unterseite um 30° zu versetzen, d. h. wir führen d aus. Dieser Zustand bleibt nämlich unter s, u^{3k} und d^{3k} mit $k \in \mathbb{Z}$ erhalten. Den Effekt dieser Züge beschreiben wir durch Permutationen, indem wir die insgesamt 16 Eck- und Kantensteine nummerieren. Die Steine der Oberseite seien $1, \dots, 8$ und die der Unterseite $9, \dots, 16$ jeweils gegen den Uhrzeigersinn (Blick von oben) nummeriert. Die Ecksteine bilden die ungeraden Ziffern. Direkt unter Ecke 1 sei Kante 10 wegen des 30°-Versatzes. In Zyklenschreibweise gilt nun

$$\begin{aligned} s &= (1,13)(2,12)(3,11)(4,10), \\ u^3 &= (1,3,5,7)(2,4,6,8), \\ d^3 &= (9,11,13,15)(10,12,14,16). \end{aligned}$$

Damit lässt sich bereits erheblicher „Schaden“ anrichten. Bei der Suche nach Zugfolgen mit überschaubarem Effekt empfiehlt es sich *Kommutatoren* zu betrachten, also Produkte der Form $[x,y] := xyx^{-1}y^{-1}$. Oft kürzen sich dabei unerwünschte Effekte heraus. Zum Beispiel:

$$\begin{aligned} [s,u^3] = su^3su^{-3} &= (1,7,13,11,3)(2,8,12,10,4), \\ A := [s,u^3]^2 s &= (3,7)(4,8)(10,12)(11,13) \end{aligned}$$

(man kann diese Rechnungen leicht im Computeralgebrasystem GAP [3] verifizieren). Das Manöver A vertauscht auf der Ober- und Unterseite je ein Eckpaar und ein Kantenpaar. Insbesondere wird nicht mehr zwischen den Ebenen getauscht. Möchte man nur auf einer Seite Veränderungen, sagen wir auf der Unterseite, so kann man den Kommutator mit d^3 bilden:

$$B := [A, d^3] = (9,11,13)(10,12,16)$$

(bei der praktischen Durchführung hilft es $A = A^{-1}$ zu wissen). In der Tat bleibt nun die Oberseite invariant, während auf der Unterseite je drei Ecken und drei Kanten zyklisch permutiert werden. Die bisher benutzten Züge s, u^3 und d^3 können eine Ecke der Oberseite nicht von ihrer rechten Nachbarkante trennen. Es ist daher Zeit Neues zu wagen. Die Kombination $ud^{-1}sdu^{-1}$ erhält ebenfalls die Würfelgestalt. Günstiger ist jedoch der Kommutator

$$t := [ud^{-1}, s] = (4,10)(8,14).$$

Dies vertauscht ein gegenüberliegendes Kantenpaar von Ober- zu Unterseite. Durch Kommutatorbildung kann man Ober- und Unterseite trennen

$$C := [t, d^6] = (4,8)(10,14)$$

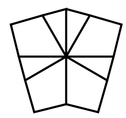
und schließlich nur „unten“ arbeiten

$$D := [B, C] = (10,14,16).$$

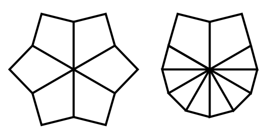
Ein Algorithmus

Wir verdrehen das Puzzle nun endgültig durch beliebige Anwendung von s, u und d, sodass die Würfelgestalt verloren geht. Man beachte, dass nicht alle Kombinationen dieser drei Erzeuger möglich sind (bereits $u^{-1}s$ ist nicht realisierbar). Im Gegensatz zum Zauberwürfel bilden die Zustände des Square-1 also keine Permutationsgruppe in der naheliegenden Weise. Eine genaue Analyse der Zustandsmenge erfolgt im letzten Abschnitt. Wir lösen das Puzzle durch folgende Schritte:

Schritt 1: Sortieren von Ecken und Kanten Es bereitet erhebliche Schwierigkeit das Puzzle zurück auf Würfelgestalt zu bekommen. Dies liegt vermutlich daran, dass man die Würfelform nur über genau eine andere Form erreicht. Bei dieser haben Ober- und Unterseite die Form

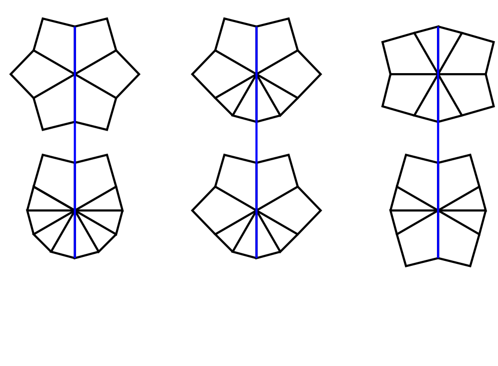


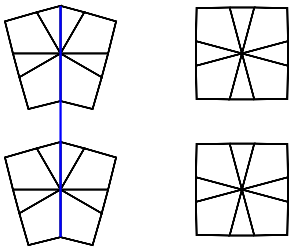
(ohne Berücksichtigung von Farben). Man muss also zwangsläufig diesen Zustand anstreben. Dies geht nur von zwei weiteren Formen (welchen?). Nach einigen Probieren finde ich es leichter einen sortieren Zustand anzustreben, der von vielen weiteren Zuständen erreichbar ist. Es bereitet keinerlei Probleme sechs Ecken (und folglich keine Kante) auf der Oberseite zu einem Stern zu versammeln. Für die Unterseite bleiben noch fünf mögliche Formen, die sich durch die Anzahl der Kanten zwischen den zwei verbleibenden Ecksteinen unterscheiden. Allein mittels s- und d-Zügen lässt sich zwischen diesen fünf Formen wechseln. Man erreicht somit:



(Ober/Unterseite)

Schritt 2: Herstellen der Würfelform Eine leicht zu merkende Folge symmetrischer Züge verteilt die Kantensteine solange gleichmäßig bis die Würfelform aus der Sterngestalt entsteht:





Die blaue Linie gibt den s-Zug an.

Schritt 3: Anordnung der Ecksteine Mit den im letzten Abschnitt eingeführten Zugfolgen s, u^3 und d^3 lassen sich die vier Ecksteine der Oberseite mit Leichtigkeit weiß färben ohne die Würfelgestalt zu zerstören. Jede Permutation der unteren Ecken $9, 11, 13, 15$ lässt sich als Produkt der Zyklen $(9, 11, 13, 15)$ und $(9, 11, 13)$ schreiben. Mit den Manövern d^3 und B lassen sich daher die Ecksteine der Unterseite korrekt anordnen ohne die Oberseite zu verändern. Analog kann man auch die Ecksteine der Oberseite in die richtige Position bringen (wir wissen bereits, dass man Ober- und Unterseite komplett tauschen kann).

Schritt 4: Anordnung der Kantensteine Wir benutzen ab jetzt nur noch Zugfolgen, die die Ecksteine fest lassen. Mit der Sequenz t können wir oben und unten je zwei weißen Kantensteine produzieren. Nach geeigneter Anwendung von D, liegen die Kanten der Unterseite gegenüber. Wie immer erreicht man das auch auf der Oberseite. Durch nochmalige Benutzung von t färbt man alle Kanten der Oberseite weiß. Mit den Zügen C und D bringt man die Kanten der Unterseite in die richtige Anordnung. Leider werden bei Anwendung von C Kanten der Oberseite permutiert. Es entstehen daher zwei Fällen: Bilden die oberen Kantensteine eine gerade Permutation, so lassen sie sich mit dem Analogon von D lösen. Anderenfalls können wir das Puzzle mit den bisherigen Methoden gar nicht lösen, denn u^3, d^3, s, t sind allesamt gerade Permutationen (auf 16 Ziffern). Wir benötigen in diesem Fall den nächsten Schritt.

Schritt 5: Lösen des Paritätproblems Ich habe mich lange gefragt, ob man ungerade Permutationen überhaupt als Zustände erreichen kann. Beim Nachdenken über die Sterngestalt aus Schritt 1 wurde mir klar, dass es durchaus möglich ist mit s je drei Ecksteine oben und unten zu tauschen. Dies realisiert ein Produkt von drei Transpositionen; also eine ungerade Permutation! Das war vielleicht das schönste Aha-Erlebnis während der „Arbeit“ an dem Puzzle. Diese Erkenntnis zeigt gleichzeitig, wie man mit wenig Aufwand vom ungeraden in den gerade Zustand wechseln kann: Wir benötigen eine Folge f, die fast den Sternzustand erreicht, führen dann einmalig s aus und anschließend f^{-1}. Von der Würfelgestalt funktioniert zum Beispiel folgende Sequenz aus Schritt 2:

$$f = su^3d^{-3}sud^{-2}su^2d^2.$$

Anschließend sind einige Ecksteine vertauscht, sodass man wieder bei Schritt 3 einsteigen muss. Man wird nun aber garantiert mit Schritt 4 fertig.

Selbstverständlich ist die beschriebene Lösung in keiner Weise optimal, dafür aber leicht nachvollziehbar.

Eine Analyse

Spätestens nach Schritt 5 des obigen Algorithmus ist klar, dass man alle $(8!)^2 = 1.625.702.400$ Permutationen der Eck- und Kantensteine durch legale Verdrehungen des Würfels realisieren kann. Damit sind allerdings nur Zustände in Würfelgestalt gezählt. Für alle andere Zustände muss man zuerst eine „Normalform“ definieren. Wir vereinbaren, dass Ober- und Unterseite so ausgerichtet sein sollen, dass auf der (roten) Vorderseite ein senkrechter Strich rechts der Mitte zu erkennen ist (das bedeutet noch nicht, dass der Zug s möglich ist). Wir zählen nun die verschiedenen Formen ohne Berücksichtigung von Farben. Ober- und Unterseite enthalten jeweils zwischen 2 und 6 Ecksteine. Enthält die Oberseite e Ecken, so besteht sie insgesamt aus $12 - e$ Steinen. Es gibt dann $\binom{12-e}{e}$ Möglichkeiten die Position dieser Ecken zu wählen. Auf der Unterseite kommen $\binom{4+e}{8-e}$ mögliche Anordnungen hinzu. Theoretisch ergeben sich somit

$$\sum_{e=2}^{6} \binom{12-e}{e}\binom{4+e}{8-e} = 8518$$

Formen. Unterscheidet man zwischen den beiden Zuständen der mittleren Ebene, so besitzt das Puzzle insgesamt höchstens

$$(8!)^2 \cdot 2 \cdot 8518 = 27.695.466.086.400 \qquad (0.1)$$

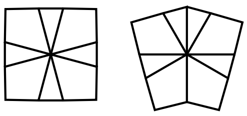
Zustände. Allerdings ist keineswegs klar, ob wirklich alle Formen angenommen werden können. Als Beitrag des Computeralgebra-Rundbriefs sei es erlaubt den Computer zu befragen. Da 8518 eine überschaubare Zahl ist, genügt ein naiver Algorithmus. Wir modellieren die Formen von Ober- und Unterseite durch zwei Vektoren v_u und v_d mit Einträgen 1 (für eine Kante) und 2 (für eine Ecke). Drehungen von Ober- und Unterseite entsprechen Shifts dieser Vektoren, wobei zu beachten ist, dass die Länge der Vektoren im Laufe des Algorithmus variiert. Im Ausgangszustand gilt

$$v_u = v_d = (1, 2, 1, 2, 1, 2, 1, 2)$$

und wir setzen $L_1 := \{(v_u, v_d)\}$ und $M := \varnothing$. Induktiv sei L_i für ein $i \geq 1$ bereits definiert. Für jede Form $(v_u, v_d) \in L_i$ berechnen wir alle möglichen Shifts und fügen diese der Menge M hinzu. Für jeden Shift prüfen wir, ob die ersten $k \in \{3, 4, 5, 6\}$ Einträge von v_u und v_d in der Summe 6 ergeben. Ist dies der Fall, so lässt sich s durchführen. Dabei werden die entsprechenden Teile von v_u und v_d vertauscht (man kann vernachlässigen,

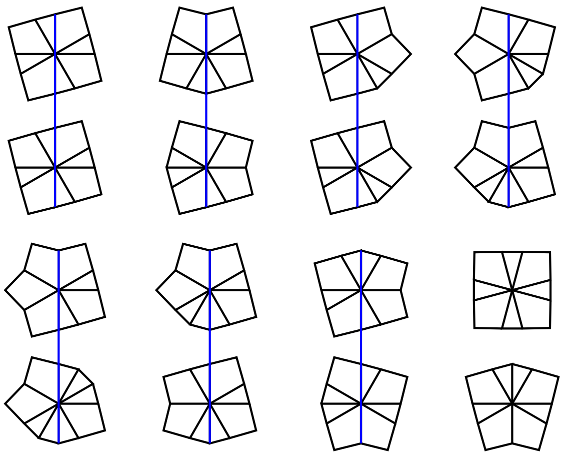
dass s die Reihenfolge umkehrt). Die neu konstruierten Paare (v'_u, v'_d) speichern wir in L_{i+1} ab, falls sie noch nicht in M liegen. Man erhöhe nun i und wiederhole so lange, bis L_i leer ist. In der Tat enthält M nach acht Durchläufen alle 8518 Formen. Damit ist (0.1) die korrekte Anzahl an Zuständen des Square-1. Der große Primfaktor 4259 von 8518 bestätigt erneut, dass sich die volle Zustandsmenge nicht zweckmäßig als Permutationsgruppe beschreiben lässt.

Interessanterweise gibt es nur 24 Formen, die erst nach acht Durchläufen des Algorithmus entstehen. Sie unterscheiden sich nur durch Shifts und Vertauschen von Ober- und Unterseite. Ein Repräsentant ist



(Ober/Unterseite).

Man erhält ihn durch die folgenden sieben s-Anwendungen:



In Analogie zu *God's Number* für den Zauberwürfel kann man fragen, wie viele Züge nötig sind, um einen solchen Zustand zu lösen. Ein *Zug* sei hierbei eine Anwendung von s oder einer Potenz von u oder d. Da nur eine gerade Anzahl an s-Zügen die mittlere Ebene intakt lässt, muss es Zustände geben, deren Lösung mindestens acht s-Züge benötigen. Da zwischen zwei s-Zügen ein u- oder d-Zug erfolgen muss, erfordern manche Zustände mindestens 15 Züge. Tatsächlich reichen selbst 15 Züge nicht immer, denn es gibt nur 14^7 Sequenzen der Form $sx_1^{i_1}sx_2^{i_2}\ldots sx_7^{i_7}s$ mit $x_1,\ldots,x_7 \in \{u,d\}$, aber $24 \cdot (8!)^2 > 14^7$ „kritische" Zustände. Ein 16. Zug lässt sich an neun Positionen sinnvoll einfügen. Damit kommt man auf höchstens

$$14^7 + 2 \cdot 14^8 + 7 \cdot 14^6 \cdot 63 > 24 \cdot (8!)^2$$

Möglichkeiten, was immer noch zu wenig ist. Noch einmal klappt der Trick allerdings nicht. Wir können somit festhalten, dass God's Number für den Square-1 mindestens 17 ist. Shuang Chen [1] hat 2017 (offenbar auch über Weihnachten) per brute force den genauen Wert von God's Number bestimmt: 31 (zumindest wenn man sich auf Zustände beschränkt, bei denen s anwendbar ist). Leider wurde dieses Ergebnis in keiner Zeitschrift publiziert.

Literatur

[1] S. Chen, *Square one can be solved in 31 moves in face turn metric*, `https://www.speedsolving.com/threads/square-one-can-be-solved-in-31-moves-in-face-turn-metric.67363`.

[2] E. D. Demaine, S. Eisenstat and M. Rudoy, *Solving the Rubik's cube optimally is NP-complete*, in: 35th Symposium on Theoretical Aspects of Computer Science, Art. No. 24, 13, LIPIcs. Leibniz Int. Proc. Inform., Vol. 96, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

[3] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.11.0*; 2020, (`http://www.gap-system.org`).

SFB-TRR 195 Symbolic Tools in Mathematics and their Application

Im September 2020 hat die DFG die zweite Phase des SFB-TRR 195 bewilligt. Im Folgenden werden einige in der zweiten Runde neu hinzugekommenen Projekte vorgestellt.

Project A11: Tropical linear degenerate flag varieties and applications

Flag varieties and Grassmannians are important moduli spaces in algebraic geometry, parametrising (flags of) subspaces in a linear space. The tropical Grassmannian parametrises realisable tropical linear spaces, that is, tropical linear spaces which arise as the tropicalisation of a linear space defined over the ground field. The tropical Grassmannian is contained in the *Dressian* which parametrises all tropical linear spaces, not necessarily realisable ones. The interplay of the tropical Grassmannian and the Dressian remains an active area of research.

In this project, we propose a new approach towards tropical flag varieties. We generalise the tropical approach to linear degenerate flag varieties, whose homology rings are being studied by PI Fourier. That is, we consider all subrepresentations of a fixed dimension vector within the equioriented type A quiver with a representation of constant dimension vector. This move offers at the same time a new perspective on tropical flag varieties. This is true since tropical linear degenerate flag varieties can be viewed as interpolations between products of tropical Grassmannians and tropical flag varieties. Furthermore, we extend the study of tropical flag varieties to analogues of Dressians. Dressian flag varieties should parametrise flags of tropical linear spaces, without requiring their relative realisability.

With the complete combinatorial characterisation of all tropical linear degenerate flag varieties and Dressian flag varieties as our distant goal of dreams, we propose concrete steps to obtain crucial partial information (e.g. on particular cones) about tropical linear degenerate flag varieties and Dressian flag varieties.

Furthermore, we work towards applications in *network coding*: by preliminary work of PI Nebe with Liebhold and Vazquez-Castro (2018), flag varieties offer a promising approach to transmit codes efficiently. The methods which we intend to develop to attack tropical flag varieties and Dressian flag varieties – in particular linear degenerate flag varieties – allow us to encode information even more efficiently. First steps have been taken very recently by PIs Fourier and Nebe (2020).

The project offers a new constructive approach to a meaningful class of higher-dimensional tropical varieties. Throughout this research, the working hypotheses will be built on the computation of relevant examples. These calculations involve fans and polyhedral complexes (for which the natural cornerstone system is POLYMAKE) and initial ideals and Gröbner bases (using SINGULAR). For that reason, the computations will necessitate the further integration in OSCAR.

Principal investigators: G. Fourier (Aachen), H. Markwig (Tübingen), G. Nebe (Aachen)

Project A22: Automorphisms of irreducible holomorphic symplectic manifolds via hermitian forms

An irreducible holomorphic symplectic manifold (IHSM) is a compact, complex Kähler manifold X which is simply connected and which admits a nowhere degenerate symplectic 2-form σ_X unique up to scaling

$$\pi_1(X) = 1, \quad \mathrm{H}^0(X, \Omega_X^2) = \mathbb{C}\sigma_X.$$

IHSMs are one of the fundamental building blocks of Kähler manifolds with trivial first Chern class. Namely, these manifolds admit a finite étale cover which is the product of a torus, IHSMs and Calabi–Yau manifolds. For example Enriques manifolds are quotients of IHSMs by automorphisms. Similarly, Borcea–Voisin CY3-folds appearing in mirror symmetry are constructed as the quotient of the product of a K3 surface and an elliptic curve by a suitable symmetry.

Goal A: This research project aims to classify finite groups of symmetries of irreducible holomorphic symplectic manifolds up to deformation.

IHSMs have proved a fertile testing ground for conjectures in arithmetic and algebraic geometry, as is illustrated by the spectacular proofs of the Tate conjecture for K3 surfaces, the 2-dimensional IHSMs. In order to formulate conjectures and test them, one needs to be able to compute examples. Frequently, this is only possible in highly symmetric situations.

Since the pioneering work of Nikulin and Vinberg in the 70s, the study of automorphisms of K3 surfaces has become a subject in its own right. For instance Mukai's beautiful theorem on symplectic groups of automorphisms and the Mathieu group has triggered the discovery of the Mathieu-Moonshine phenomena coming from theoretical physics. Automorphisms have provided insights to non-liftability from positive characteristic, modularity of Galois representations, dynamical systems, Salem numbers, numerical period computations via Picard–Fuchs differential equations, to name a few. Automorphisms of K3 surfaces have proved to be both fascinating and useful. We expect that the same holds true for their higher dimensional incarnations, the IHSMs. Hence a better understanding and a classification of (finite) groups of automorphisms of IHSMs is of great interest.

Automorphisms of K3 surfaces are usually classified by first determining their fixed point sets. Then this is used to derive an equivariant projective model, for example an elliptic fibration, which can be studied explicitly. This approach breaks down on higher dimensional IHSMs. We circumvent this by proposing to start with the natural representation

$$\mathrm{Aut}(X) \to O(\mathrm{H}^2(X, \mathbb{Z}), q)$$

by isometries of the second cohomology lattice equipped with the Beauville–Bogomolov–Fujiki quadratic form q. This allows to translate Goal A to

Goal B: Given a quadratic $\mathbb{Z}$-lattice (L, q) classify isometries with a given characteristic polynomial up to conjugacy in the orthogonal group $O(L, q)$.

Our path towards Goal B is a combination of classical tools from number theory (hermitian forms) and modern algorithmic methods from group theory via OSCAR. Once a classifying algorithm is available, we plan to use Markman Verbitsky's Torelli type theorem to show that an isometry comes from an automorphism. A given isometry can then be used to further understand the geometry of the corresponding automorphism. This turns the conventional approach for K3 surfaces, which starts from the geometry and uses it to classify the representation, around. The advantage of our approach is that it is rather independent of the dimension and applies to any deformation type of IHSMs.

Principal investigator: S. Brandhorst (Saarbrücken)

Project B7: Computations with matrix groups

Designing algorithms for matrix groups defined over finite fields is an international research focus pursuing two central goals:

1. given a finite set of matrices in $\mathrm{GL}(d, q)$, "recognize" the group they generate;
2. exploit available structural information about a "recognized" group to perform effective computations in this group.

The practical performance of the resulting algorithms frequently is the decisive factor in whether or not mathematical conjectures can be tested experimentally or proved by handling small dimensional cases computationally.

The overall strategy to achieve the first central goal, *matrix group recognition*, is to compute a data structure called *composition tree* for the input group which reflects a structural decomposition of the group (specifically a subnormal series). The composition tree is of fundamental importance as it allows many basic computations in the given group to be carried out efficiently, such as testing whether it contains a given matrix. Algorithms that answer structural questions about a matrix group such as testing subgroups for conjugacy, computing normalisers, and many more, require a composition tree of the input group.

The second central goal, *efficient matrix group computations*, aims to develop algorithms for computing in groups which have already been recognized. Structural questions about a group are often investigated by the soluble radical method, which again relies on the infrastructure supplied by a composition tree. Efficient algorithms exploiting the soluble radical model exist, see for example [4, Chap. 10] or [1, Sec. 12].

This project contributes to both central goals, by pursuing the following aims:

Aim 1: Develop a novel second generation constructive recognition algorithm for finite classical groups of Lie type, initially for the natural representation, albeit with a view to generalising this to a 'black box' setting for arbitrary representations.

Aim 2: Implement highly efficient methods to facilitate computations in finite groups of Lie type.

Aim 1 is a contribution to the first central goal. The outcomes of this project will be new mathematical results and new algorithms, accompanied by correctness proofs, tight complexity bounds and efficient implementations. Faster constructive recognition algorithms for finite classical groups immediately lead to greater efficiency in the overall matrix group recognition machinery.

Aim 2 contributes to the second central goal. In particular, it will make the methods from [2] for untwisted finite groups of Lie type available. In the long term, we intend to achieve the same goal for the twisted case, taking into account the (not yet published) results in [3].

Together, the outcomes of our aims will provide an effective solution to the word problem in finite classical matrix groups, by providing a rewriting procedure which takes an arbitrary element of a finite classical matrix group as input and expresses it as a word in the input generators.

Our implementations will be open source, written in OSCAR and GAP, and exploit features of OSCAR such as the ability to write parallel algorithms.

Principal investigators: M. Horn (Kaiserslautern),
A. Niemeyer (Aachen)

References

[1] H. Bäärnhielm, D. Holt, C.R. Leedham-Green, and E. A. O'Brien, *A practical model for computation with matrix groups*, J. Symb. Comput. 68 (2015), 27–60.

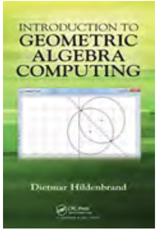
[2] A. M. Cohen, S. H. Murray, and D. E. Taylor, *Computing in groups of Lie type*, Math. Comp. 73 (2004), 1477-1498.

[3] A. M. Cohen, and D. E. Taylor, *Row reduction for twisted groups of Lie type*, Preprint.

[4] D. F. Holt, B. Eick, and E. A. O'Brien, *Handbook of computational group theory*, Discrete Mathematics and its Applications, Chapman & Hall/CRC, 2005.

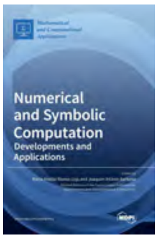
Publikationen über Computeralgebra

Neuerscheinungen:

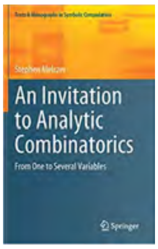


Dietmar Hildenbrand,
Introduction to Geometric Algebra Computing,
CRC Press, Boca Raton 2020,
212 Seiten,
ISBN 978-0-367-57132-0

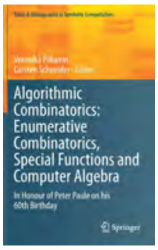
Akihito Kikuchi,
Quantum Mechanics built on Algebraic Geometry: Emerging Physics through Symbolic Computation,
Eliva Press, Chisinau 2021,
285 Seiten,
ISBN 978-1-636-48071-8



Maria A.R. Loja,
Joaquim I. Barbosa (Hrsg.),
Numerical and Symbolic Computation: Developments and Applications,
MDPI AG, Basel 2020,
140 Seiten,
ISBN 978-3-039-36952-2



Stephen Melczer,
An Invitation to Analytic Combinatorics
Springer-Verlag, Berlin 2021,
436 Seiten,
ISBN 978-3-030-67079-5



Veronika Pillwein,
Carsten Schneider (Hrsg.),
Algorithmic Combinatorics, In Honour of Peter Paule on His 60th Birthday,
Springer-Verlag, Berlin 2020,
427 Seiten,
ISBN 978-3-030-44558-4

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 Jürgen Klüners oder Martin Kreuzer (`klueners@math.uni-paderborn.de`, `martin.kreuzer@uni-passau.de`) auf.

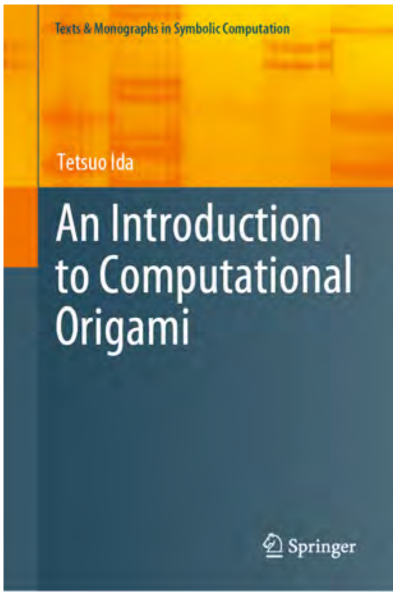
Besprechungen zu Büchern der Computeralgebra

Tetsuo Ida
An Introduction to Computational Origami

Die Tatsache, dass zwischen Origami, der alten japanischen Kunst des Papierfaltens, und der Algebra interessante Beziehungen bestehen, wurde mir erstmals vor 20 Jahren durch den Artikel [1] klar. Die durch gewisse Origami-Operationen konstruierbaren komplexen Zahlen bilden Körper, welche je nach den erlaubten Operationen den quadratischen Abschluss von $\mathbb{Q}$ (also den Körper der mit Zirkel und Lineal konstruierbaren Zahlen) oder den quadratischen und kubischen Abschluss von $\mathbb{Q}$ mit einschließen. Und in letzterem kann man die Winkeldreiteilung und das delische Problem der Würfelverdoppelung ja bekanntlich lösen. Obwohl mache solcher Resultate bereits seit über 120 Jahren bekannt sind (vgl. [2]), hat die Formalisierung der Origami-Kunst durch die Einführung von Methoden der Computeralgebra und der algorithmischen Logik in den letzten 30 Jahren bedeutend an Fahrt aufgenommen, und *Computational Origami* ist zu einem inhaltsreichen und spannenden Forschungsthema geworden.

Mit dem nun nach diversen Verzögerungen endlich erschienenen Band von Tetsuo Ida ist erstmals auch eine Monographie verfügbar, mit deren Hilfe der Leser sich systematisch in dieses Gebiet einarbeiten und einen Überblick über den aktuellen Stand der Forschung erhalten kann. Das Werk verfolgt dabei einen durch und durch praktischen Ansatz: alle Algorithmen und Resultate werden mithilfe des E-Origami-Systems `EOS`, einem Paket für `Mathematica`, ausprobiert und illustriert.

Nachdem in den ersten beiden Kapiteln die Geschichte und die Formalisierung der Origami-Operationen vorgestellt wurden, beginnt in Kapitel 3 die Reise durch die Welt des *Computational Origami* in Form der anfangs erwähnten algebraischen Analyse der Origami-Operationen und ihrer Beziehungen zur Körpertheorie. Danach folgen zwei Kapitel über die logische Formalisierung der Origami-Geometrie sowie ihrer Anwendung auf geometrische Beweise. Ein weiterer überraschender Bezug zwischen Origami und der Knotentheorie wird in Kapitel 6 diskutiert, wo es um computergestützte Konstruktionen sowie automatische Verifikationen polygonaler Knoten mittels Origami-Theorie geht. Im siebten und letzten Kapitel wird Origami schließlich vollends abstrahiert und durch ein Reduktionssystem (rewriting system) beschrieben. Diese Formalisierung erlaubt bereits einen Ausblick auf 3-dimensionale Verallgemeinerungen und Origamis mit Überlappungen (wie sie z. B. bei der klassischen Kranich-Konstruktion vorkommen). In den Anhängen wird dann noch das System `EOS` näher vorgestellt.



Dieser Band nimmt in der Reihe der *Texts and Monographs in Symbolic Computation* des Springer-Verlags eine sehr spezielle Stellung ein, denn etwas abseits von den Hauptströmungen der modernen Forschung wird hier ein wunderschöner, tiefer und fruchtbarer Bezug zwischen der Computeralgebra und einer klassischen japanischen Kunstform beschrieben. Da ist es kein Wunder, dass mit platzsparenden Faltungen für Airbags oder einem faltbaren Weltraumteleskop auch bereits erste konkrete Anwendungen in Sicht sind.

Das Buch ist in einer sehr klaren und lesbaren Art und Weise geschrieben. Es eignet sich für Spezialvorlesungen (zu jedem Kapitel gibt es eine Sammlung von Übungsaufgaben) oder Seminare (z. B. über Anwendungen der Computeralgebra), aber auch zum Selbststudium und nicht zuletzt als erbauliche Lektüre für Computeralgebraiker, die gerne einen Blick über den Tellerrand hinaus werfen.

Martin Kreuzer (Passau)

Literatur:

[1] R. C. Alperin, A mathematical theory of Origami numbers and constructions, New York J. Math. 6 (2000), 119-133.

[2] T. S. Row, Geometrical exercises in paper folding, Addison and Co., Madras 1893.

Hinweise auf Konferenzen

GAMM-Jahrestagung 2021
(mit Minisymposium zur Computeralgebra)

Kassel, 15.03. – 19.03.2021

jahrestagung.gamm-ev.de

Wegen der Corona-Krise musste die für Mitte März 2020 geplante GAMM-Jahrestagung in Kassel leider abgesagt werden. Dafür wird nun die GAMM-Jahrestagung 2021 in Kassel abgehalten. Auch das geplante Minisymposium zur Computeralgebra wurde auf 2021 verschoben. Die nachfolgenden GAMM-Jahrestagungen sollen in Aachen (2022), Dresden (2023) und Magdeburg (2024) stattfinden.

MEGA 2021

Tromsö, Norwegen, 07.06. – 11.06.2021

puremath.no/mega2021

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.

The 16th edition of MEGA will take place at UiT (The Arctic University of Norway) including 10 invited talks as well as contributed talks. Information on submission and deadlines will be available on the Web-Site.

ISSAC 2021

St. Petersburg, Russland, 18.07. – 22.07.2021

www.issac-conference.org/2021

The International Symposium on Symbolic and Algebraic Computation (ISSAC) is the premier conference for research in symbolic computation and computer algebra. ISSAC 2021 will be the 46th meeting in the series, which started in 1966 and has been held annually since 1981.

The international situation makes it impossible to hold ISSAC'21 the way usual ISSAC conferences are held. To offer a reasonable experience to our community, ISSAC'21 will be organized according to the following exceptional format:

- ISSAC'21 will be run in a hybrid mode, that is, with largest attendance online and a possibility to attend in person for those able to travel.
- The working times will be from 14:00 to 20:00 in Saint Petersburg, so as to accommodate as many time zones as possible.
- The conference will spread over 5 days, including tutorials (Mon 19 to Fri 23 July).

Rings and Polynomials

Graz, Österreich, 19.07. – 24.07.2021

integer-valued.org/rings2020

There will be a conference on "Rings and Polynomials" at Technische Universitaet Graz (TU Graz) in Graz, Austria, July 19-24, 2021.

Topics include: integer-valued polynomials, polynomial functions, multiplicative ideal theory, topological methods in ring theory, Zariski-Riemann spaces of valuation domains, factorization theory in rings and semigroups, Pruefer and Krull domains and generalizations.

The conference at the Institute of Analysis and Number Theory of Technische Universität Graz (TU Graz) continues the series of biennial ring-theory conferences held alternatingly at the two math departments in Graz since 2012 - at TU in 2012, at KFU in 2014, at TU in 2016, and at KFU in 2018. We hope to see the participants of the previous events again in Graz 2021, as well as many new participants.

CICM 2021

Timişoara, Rumänien, 26.07. – 31.07.2021

synasc.ro/2020

The Conference on Intelligent Computer Mathematics (CICM) is an organization and a conference dedicated to promoting the advancement of machine-supported reasoning, computation, and knowledge management in Science, Technology, Engineering, and Mathematics.

CICM-14 (CICM 2021) will be held during July 26-31, 2021 in Timisoara, Romania, co-chaired by Fairouz Kamareddine and Claudio Sacerdoti Coen, and organized by Madalina Erascu.

SIAM AG 2021

College Station, Texas, USA, 16.08. – 20.08.2021

www.siam.org/conferences/cm/conference/ag21

This is the meeting of the SIAM Activity Group on Algebraic Geometry.

The purpose of the SIAM Activity Group on 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.

SCSS 2021

Online, 08.09. – 10.09.2021

www3.risc.jku.at/conferences/scss2021

SCSS 2021 is the 9th International Symposium on Symbolic Computation in Software Science. Its purpose is to promote research on theoretical and practical aspects of symbolic computation in software science, combined with modern artificial intelligence techniques. Due to COVID-19 pandemic, SCSS 2021 will be held as a virtual event, organized by the Research Institute for Symbolic Computation, Johannes Kepler University Linz, Austria.

The International Symposium on Symbolic Computation in Software Science (SCSS '21) will take place online, hosted by RISC/JKU in Linz, Austria.

INFOS 2021: Fachtagung des Fachausschusses „Informatische Bildung in Schulen" der GI

Wuppertal, 08.09 – 10.09.2021

www.infos2021.de

Die 19. GI-Fachtagung Informatik und Schule (INFOS) des Fachausschusses ≫Informatische Bildung in Schulen≪ findet in der Bergischen Universität Wuppertal statt. Die Tagung steht unter dem Motto ≫Lehrerbildung≪.

Sommerschule Computeralgebra des SFB-TRR 195

Aachen, 06.09. – 10.09.2021

oscar.computeralgebra.de/meetings/2021-09/

The summer school of the SFB-TRR 195 on Computeralgebra has been scheduled. It will take place from September 6th to September 10th in Aachen. The main organizers are Thomas Breuer, Claus Fieker, Tommy Hofmann and Max Horn. More information will follow soon.

Jahrestagung des SFB-TRR 195

Aachen, 13.09. – 16.09.2021

www.computeralgebra.de/news

The first annual conference of the SFB-TRR 195 (second period) will take place at the RWTH Aachen from 13th to 16th September 2021.

Main organizer is Ghislain Fourier. More information will follow soon.

CASC 2021

Sotschi, Russland, 13.09. – 17.09.2021

www.casc-conference.org

The 23nd International Workshop on Computer Algebra in Scientific Computing, CASC 2021, will be held in Sochi, Russia, September 13 - 17, 2021.

At the moment the organisers are optimistic to hold the conference in person in September 2021. However, should this not be possible due to the ongoing pandemic then the conference will go ahead as an online event.

The tools of Scientific Computing play an important role in the natural sciences and engineering. Computer Algebra Systems and the underlying algorithms for Symbolic Computation play an increasingly important role within Scientific Computation. The CASC workshop series has been running for over two decades to explore the interaction of these topics, their implementation, and their application.

Industrietagung der Fachgruppe Computeralgebra

Oldenburg, September 2021

www.fachgruppe-computeralgebra.de

Nach elfjähriger Pause veranstaltet die Fachgruppe Computeralgebra der Gesellschaft für Informatik wieder eine Industrie-Fachtagung. Im Zuge der großen Herausforderungen an digitale Sicherheit durch neue Technologien wie KI, hoch performante binäre Systeme und Quantentechnologien haben wir uns entschieden, den Fokus auf Kryptografie zu legen.

Siehe ausführliche Ankündigung auf Seite 6.

DMV-Jahrestagung 2021 (mit ÖMG) (inkl. Sektion Computeralgebra)

Passau, 27.09. – 01.10.2021

www.uni-passau.de/dmv-oemg-2021/

Die DMV-Jahrestagung 2021 wird vom 27.9.-1.10. in Passau stattfinden. Das Organisationsteam plant, die Veranstaltung sofern es geht in Präsenz durchzuführen. Es besteht allerdings die pandemiebedingte Möglichkeit, dass auf eine Online-Tagungsform ausgewichen werden muss. Alle Fachgruppenmitglieder sind herzlich eingeladen, an der Sektion Computeralgebra teilzunehmen, die von Martin Kreuzer (Passau) und Manuel Kauers (Linz) geleitet wird. Bewerbungen um einen der Vorträge sind baldmöglichst an Martin Kreuzer zu richten.

GI-Jahrestagung 2021

Berlin & Online, 27.09 – 01.10.2021

informatik2021.de

Die INFORMATIK ist die offizielle Jahrestagung der Gesellschaft für Informatik e.V. (GI), der größten Vereinigung der Informatikerinnen und Informatiker im deutschsprachigen Raum und wird jährlich an wechselnden Orten veranstaltet. Im Jahr 2021 findet die INFORMATIK vom 27.09. bis 01.10.2021 Online und in Berlin statt, ganz im Zeichen der Nachhaltigkeit.

Computeralgebra-Tagung der Fachgruppe

München, 10.03. – 12.03.2022

www.fachgruppe-computeralgebra.de

Die Jahrestagung der Fachgruppe Computeralgebra sollte turnusgemäß im Frühjahr 2021 stattfinden. Aufgrund der COVID-19-Pandemie wurde sie auf das Jahr 2022 verschoben. Nach den vielen erfolgreichen Tagungen 2003, 2005, 2009, 2012, 2014, 2017, 2019 in Kassel und 2007 in Kaiserslautern wird die Tagung im Jahr 2022 in München stattfinden. Eine ausführliche Ankündigung erfolgt im Oktober-Rundbrief.

Das Ziel dieser Tagungsreihe 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.

CoCoA - School and Conference on Computational Commutative Algebra

Hue (Vietnam), März 2022

cocoa.dhsphue.edu.vn

Die internationale Doktorandenschule und Tagung COCOA 2020 ist aufgrund der Pandemie nochmals verschoben wurden und findet jetzt im März 2022 in Hue (Vietnam) statt. Die genauen Daten werden im nächsten Heft bekannt gegeben.

CCAAGS-22

Seattle, USA, 27.06. – 01.07.2022

sites.google.com/view/ccaaggs-22/home

CCAAGS-22 aims to bring together researchers working in the creative mixture of combinatorial and computational ideas in applied algebraic geometry. As part of the event we will celebrate Bernd Sturmfels and his contributions to the field.

Details of the conference will be posted at the website as they become available.

Fachgruppenleitung Computeralgebra 2020–2023

Sprecherin:
Prof. Dr. Anne Frühbis-Krüger
Carl-von Ossietzky Universität Oldenburg
Institut für Mathematik
Carl-von-Ossietzky-Straße 11, 26129 Oldenburg
0441 798-3233
`anne.fruehbis-krueger@uni-oldenburg.de`
`https://uol.de/anne-fruehbis-krueger`

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

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

Fachreferent CA-Systeme und -Bibliotheken:
Prof. Dr. Max Horn
Fachbereich Mathematik
Technische Universität Kaiserslautern
Gottlieb-Daimler-Straße, 67663 Kaiserslautern
0631 205-2730, -4427 (Fax)
`horn@mathematik.uni-kl.de`
`https://www.quendi.de/de/mathe`

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`

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/`

Stellvertretender 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.groups.ma.tum.de/algebra/kemper`

Fachreferentin Industrie:
Xenia Bogomolec
Coding Services Hannover
Engelbosteler Damm 15, 30167 Hannover
0173 3031816
`indigomind@protonmail.ch`
`https://quant-x-sec.com`

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`

Vertreter der DMV:
Prof. Dr. Florian Heß
Carl-von Ossietzky Universität Oldenburg
Institut für Mathematik, 26111 Oldenburg
0441 798-2906, -3004 (Fax)
`florian.hess@uni-oldenburg.de`
`https://uol.de/florian-hess`

Fachreferent Themen, Anwendungen und Publikationen:
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 Redaktion Rundbrief:
Dr. Fabian Reimers
Zentrum Mathematik – M11
Technische Universität München
Boltzmannstr. 3, 85748 Garching
089 289-17474
`reimers@ma.tum.de`
`http://www.groups.ma.tum.de/algebra/reimers`