Notes on ICMS 2020

Notes by J.H.Davenport

13–16 July 2020

Contents

0.1 Opening

Welcome To the largest TU in Northern Germany, the city of Henry the Lion, and the birthplace of Gauß.

TdW If ICMS cannot become a virtual conference, who can.

Chapter 1

A: Gröbner Bases

1.1 Levandovskyy

See [Lev20]. Fundamentally, the word problem in $K\langle X\rangle$ is undecidable. However, if the GN is finite it is decidable. A fundamental requirement is that, if the GB is finite, the algorithm finds it in finitely many steps. See our Letterplace 4-1-3 algorithm in Singular 4-1-3. Note also that NCAlgebra does things no other systems can. The system is also tractable if it can be graded by a well-ordered monoid. There are some bits of good news (slides 12,13). See also [CR99].

1.2 Tobias Metzlaff - Gröbner Bases over $K\langle X\rangle$ and $\mathbf{Z}\langle X\rangle$ in theory and practice

See [MLSAZ20]. LetterPlace is a subsystem of Singular. The name comes from a correspondence between $K\langle X\rangle$ and $K[K|\mathbf{N}] = K[\{x_i(j) : x_i \in X, j \in \mathbf{N}\}]$. Note what he calls the Gröbner Trinity

Example 1 *$G := \langle 6xy, 4yz\rangle$ is a GB over a field. But $2xyz = (6xy)\cdot z - x\cdot(4yz)$ is not reducible by either over $\mathbf{Z}$.*

Hence the concept of a *strong GB*. Hence we have S- and G-polynomials of the first kind, the usual ones, and also the second kind, based on $S(f,g)$ by reducing $\mathrm{lm}(f)\cdot w\cdot \mathrm{ln}(g)$ for arbitrary monomial w (hence infinitely many of these).

Criterion 1 *If $\mathrm{lm}(f)$ divides $\mathrm{lm}(g)$ then every S reduces to 0.*

Criterion 2 *If $\mathrm{lm}(f), \mathrm{lm}(g)$ and $\mathrm{lc}(f), \mathrm{lc}(g)$ are relatively prime, then every S reduces to 0.*

Criterion 3 (Chain Criterion) *Again need to add* lc *conditions.*

1.3 Session A: Q&A

CRLG Intersection of two matrix groups. One of them is preserving a space. But CRLG was inaudible: description to be written up.

Let G be the group that preserves a tensor decomposition of a vector space V of finite dimension, say dimension 20, over a finite field. So in this case the subspaces might be of dimensions 4 and 5. So take a symbolic 4 x 4 matrix $X = (x_{ij})$, and a 5 x 5 matrix $Y = (y_{ij})$. Then the Kronecker product X tensor Y is a 20 x 20 matrix. I have a subspace of the underlying space that this matrix should preserve. This gives a number of linear equations in the coefficients of the 20 x 20 matrix. So it is reasonable to expect Grobner basis techniques to solve such a set of equations?

1.3.1 Terui

Lego mindstorms manipulator.

Q–VL Biquadratic?

A At least in this situation.

Q: Miguel Marco Comparisons with other inverse kinematics software?

A We are interested in offline precomputations e.g. GCS

1.3.2 Matthias Bender

See [BFT20].

Q–VL Difference between mixed and unmixed sparsity.

A Don't necessarily get regularity in the mixed case, see slide 3.

Q How does this relate to Eder's work.

A

1.4 Session A: Q&A 2

1.4.1 Mora

There are four cornerstones of Buchberger Theory: Zacharis Canonical Representation, Spear's Theorem, Weispfenninf Multiplication and Möller's Lifting Theorem.

1.4.2 Ceria

I was looking at Involutive divisions and bases. Janet/Gerdt–Blinkov/Seiler. These are on the way to GB: often faster to compute, especially for PDE-based problems. Look at which cone a reducible monomial lives. Need a rule for constructing cones (which may or may not be disjoint.

Let $M(A)$ be the monomial set. We have an involutive division L on $MA(A)$ iff

- If $w \in L(u, U)$ and $v|w$ then $v \in L(u, U)$
- If $u, v \in U$ and $uL(U, U) \cap vL(uv, U) \neq \emptyset$ then either $u \in vL(v, U)$ or $v \in uL(u, U)$.
- If $v \in U$ and $v \in \ldots$
- $\ldots$

1.5 Session A: Q&A 3

1.5.1 Eder by Levandovskyy

Convert to lex etc in usual method. But let's invert this. Find $B \in |N$ such that all positive roots are in $[0, B)$, and rescale to $[0, 1)$. Then Descartes splitting, wit 2-adic truncation. But big polynomials with big coefficients. Examples for Henrion and Katsura. USe Wiedeman and Berlekamp–Massey, or B–M–Sakura is for in general position. Get a Hankel matrix and use Brent's solution technique. Isolate the dense parts and use AVX2 here. Good/bad prime issues, and rational reconstructions. Note that his slides promise an aggressive timeline.

1.5.2 Metzlaff

See [MLSAZ20]. Note the LT-definition of GBs.

1.5.3 Levandovskyy

Q

A

1.5.4

Q

A

Chapter 2

Plenary: Ábrahám

2.1 History

Note that there are theorem provers, constraint systems, computer algebra systems and SMT solvers. I can't talk about them all, so just SMT solvers. Timeline slide for SAT. Notable CDCL and watched literals: GRASP'97, zChaff'04. Note the importance of a standard format, and of competitions. Then SMT timeline. Z3, Yices, etc. SMT-RAT. Note SMT-LIB (quite an achievement across multiple theories) and contests here to. Benchmark library with $\sim$ 250K examples. In 2019 there were 23 teams and 71 solvers. Note many theories: `QF_UF` (uninterpreted functions), `QF_BV` (bit vectors) etc. Example of `(check-sat)` and `(get-unsat-core)` [slide 14]. Some solvers support `(maximize)`. Graph of Google Scholar citations.

2.2 The Combination

DPLL+CDCL. Start with enumeration: n variables is 2^n combinations. Better if we do propagation. Also CDCL. Then look at a SAT solver calling out to a theory solver.

We use CArL for our arithmetic library. Then a description of whole structure of SMT-RAT with many modules which can be plugged together. Results from SMT competition 2019, which shows SMT-RAT as doing reasonably well. Note also many theses supported on top of this, also ERASMUS+ projects.

2.3 Q&A

Q Constructible sets so complex geometry. Certificates?

A Resolution gives a proof of the Boolean side, but for the theory side we still need automatically0checkable certificates, ideally polynomially-checkable.

Q–MJ Semi-definite programming?

A Some in CArL, but I need to check.

Q Order theory? Lattices etc.

A Good question.

Comment There's been work on parallel correctness which relates to orders.

Q SAT is a question of non-emptyness in a set? But can you get normal forms for future use?

A Good question.

Q Prolog?

A Not well-established, but a good question.

Chapter 3

Real Algebraic Geometry: session B

3.1 Changbo Chen

Consider our 2011 ncremental algorithms [CM11]. We geta graph with vertices for variables. So how to make this graph chordal?

Q-MM Woudl even Normal Forms in the sense of GB break chordality?

A They could do.

Q Would our `RemoveZero` break chordality?

A Yes and No. JHD didn't follow.

3.2 Delaram Talaashrafi

Looking for a triangualr form for inequalities. Big problem is redundant inequalities. Our aim is eliminating these via Fourier–Motzkin, bringing doubly-exponential down to single. We see alower complexity: theoretical and practical. Claims that we detect all the redundancies. MM commented that this is in (a future release of) Maple as part of RegularChains.

Q–Chen Other software comparisons?

A In the plan.

MM Many of the others use floating point, which brings its own complications.

3.3 Akshar Nair

[Had to speak unexpectedly.]

Aim to understand "curtains" and relevance to CAD. Shows Terminology: Curtains slide. These are always a problem, which McCallum described as nullification. With lex-least we need to attack these. At ISSAC2019 [NDSM19] our implementation gave up on curtains.

Q–CWB Multiple equational constraints?

A Possibly, but there is a growth of number of CADs

Q But is this more expensive: I think not.

A Needs studying.

3.4 Gereon Kremer

Basically a conflict-driven description of CAD. This works quite well in the SMT context.

Q–ZPT Nullification?

A In the SMT-RAT implementation we never encountered it, but we used liftng adapted to Lazard. In the CVC4 we didn't check.

ME McCallum had for us fewer polynomials than Lazard in the projection.

Q–CWB Equational constraints?

A In the plan. For SMT instances we had, linear elimination gave us all the benefits.

Q–ME CV4 implementation?

A Same algorithm. The details like value selection are different. Also the polynomial arithmetic is different. And we are only called after linear technology (replace every monomial by a fresh variable) has failed.

Q–Changbo Real root isolation?

A Built in. Bisection and Sturm.

3.5 Zak Tonks

VTS + Lazard CAD. Use Maple's `RootOf` indexed by intervals, which are good for output.

Q–CWB Tarski formulae in the output?

A The output is in terms of `RootOf`, and the input can be as well. So the VTS can handle these as well.

CWB Not totally convinced.

Q–ME Available?

A Will be as my thesis finishes.

Q–Changbo Maple 2020 using RegularChains.

A Yes, problems with name clash conflicts.

3.6 Li-Yong Shen

Rational curves

Q

A

Chapter 4

Edelman: The Power of Language (Julia)

[JHD Chaired]. See [BEKS17] and `https://julialang.org/blog/2012/02/why-we-created-julia/`.

Everyone talks about "reproducible" these days. But that same is true of papers as well, and a software implementation can make the dependencies explicit. "If computing for maths is good, HPC for maths should be better". Power of language in many ways: I was surprised by its power for community building.

HPC people talk about performance, but the conversation is moving towards impact. Composability is also an issue. Portability (e.g. GPU to distributed machine) is needed. PVM was the thing in 1990, but now we have have MPI, and its showing its age. "With the right definitions, the theorems prove themselves" — Sottile quoting Grothendieck.

At SC, 1988 to 2010, there was always a section on programming languages, but now there's a dark decade. In 2005, we had Chapel/Fortress/X10. What went wrong. Note that we a good serial language, and a good user base.

So we decided about 10 years ago to start a new programming language. We 4 won the James H. Wilkinson prize. We said then that it would take ten years to build a user base. Look at Python. Buch of slides about Julia at the various national labs. Hard to measure usage of open source software like Julia. Various usage stats (# StackOverflow etc.), which show growth rates.

Examples of distributed arrays and GPU arrays. Note that no code needed rewriting. The key is abstraction. Also need a good LLVM behaviour, and good interatcions with legacy codes (and Python). Note a fresh climate model being built in Julia. They said "take these kernels". Note also slide about the mathematical notation.

Q Why not `2/3`?

A–VC `2//3`. Hence debate in the chat.

AE I had a matrix of ± 1 and I knew the answer was -1, but the answer was $+1$. MATLAB rounds determinants of integer matrices to integres, and this was basically a random number. I disagree, since floaing point should be visible. Julia also has intervals.

Q Which opensource algebra can I use?

MJ-Chat OSCAR: `https://oscarcomputeralgebra.de`.

AE Interesting.

Q `2/3 == 2//3` gives false!

A Indeed, but what you wanted was `2/3 \approx 2//3`. Noting that `\approx` renders as $\approx$.

Q Parametrised types?

A Yes, really what we're all about.

Q Unicode?

A Yes, we're great.

Chapter 5

Business Meeting

JHD chaired.

5.1 ICMS 2020: MJ/TdW

Registered participants almsot 200, 140 talks. 5 parallel software demos (as requested in ND). Proceedings in LNCS 12097 for download. The server is a virtual Amazon machine. There is an offer from Leibnitz (?) library in Hannover to host these permanently: MJ is still digesting these.

Q Eventbrite?

A Worked for us: German universities are not well set up.

5.2 ICMS 2022

MJ: many people have advocated not deciding yet. JHD gave background. Yue Ren gave a bid for Swansea. Having lived in both, the beaches of Swansea are comparable to Cape Town. Suggests all in computational foundry. Good quotes from a summer school (B&B £50) though no guarantees.

Q

A

5.3 AOB

JHD This has been a great success, as one of the first conferences, at least in my field. We should work out "lessons learned".

YR We have more people here than at any physical ICMS, so we should work out what to do virtually in the future. General agreement.

5.4 Museum Tour

Led by Sophia, a Museum Tour guide and research student studying architectural monuments.

1. Picture of a woman, from "early Islamic Art".

2. Also picture of Berlin 1780 (not called that in the centre, but Köln apparently). The idea for German museums came from the Britis h Museum and the Louvre. Then image of Berlin in 1920s with these. Pergamon was the last museum to be built on Museum Island, finished in 1930s, but looking very classical from a 1909 plan. The rebuild will be finished in 6 years.

3. The Pergamon altar (2nd C BC) is now in the eponymous museum. A beautiful reconstruction drawing. This all happened at end of 19th century, when complicated relatioships between Ottoman and German empires. This is the second-longest surviving Greek frieze after the Elgin Marbles. Long discussion of Pergamon itself and the history.

4. Ishtar gate, leading to inner city of Babylon. Built by Nebuchadnezzar. Another Ottoman/German story. 200 workers for 15 years, paid for by the Germans.

JHD had to dip out to supervise an MSc student

5. Stucco panels from Abbasid caliphate 8xx AD.

6. A room for "Asia Minor 12-14 century". But not updated, e.g. title is "Turkey is 13th century" though Turkey didn't exist.

7. Contrasts a griffin from a Roman sarcophagus in Thesaloniki with pre-Islamic Iranian image.

8. Various Turkish carpets, which were brought to Europe to be used in churches etc. In fact they were apparently make for exports, and sometimes bespoke orders with coats of arms woven in. Picture of one in a 13th-century Madonna+Child painting. The guide questioned the term "Islamic Art", which seems to be a modern invention.

Chapter 6

Algebraic Geometry via Numerical Computation

6.1 Nicolas?

Étale cohomology. Get a galois cohomology: $\rho : Gal(\overline{\mathbf{Q}}/\mathbf{Q}) \to GL_f(\mathbf{Z}/\ell\mathbf{Z})$ and the goal is to compute this explicitly.

6.2 Francesca

p-adic approximation of integral points. We have a necessary but not sufficient condition on ranks.

6.3 Netan Dogra:Algorithms for Frobenius

Has algorithms for Coleman Integration. Refers to Francesca's thesis.

6.4

6.5

Q

A

Chapter 7

Plenary: Shoup on NTL

Introduced by Anna Bigatti: noted that NTL won the Jenks Prize.

Open source C++ library for $\mathbf{Z}, \mathbf{Z}_p, \mathbf{Z}[x], \mathbf{Z}_p[X]$. p could, but needn't be prime. Also extension rings/fields. Really only doesn univariate polynomials. Matrices and lattice-basis reductions. Therefore various floating point implementations. Raw line count ∼140K. Aim for fast performance. Portability key (in 1990s I was changing jobs and environments, so no machine code). Now thread-safe and have multi-core facilities. Can take advantage of SIMD etc., mostly AVX. Also an early user of C++ templates when support became really available.

Early 2000s implemented [vH02]. Look at artithmetic in $\mathbf{Z}_p[X]$ for *large* p. For $\log_2 p \leq 24000$ we precompute powers and are better than Borodin. For the CRT, with $\log_2 p \leq 18000$ we use a cunning implementation of the naive algorithm. Quadratic, but very small constants. Small-prime FFT (word-size). Precompute `1/double(n)`, hence limited to 50 bits of integer. Note also that correction steps can cause the hardware to stall, but it turns out that branch prediction does a good job. Or there's a shift/mask solution, or "conditional move".

7.1 Barrett reduction

Computing $a \cdot b \pmod{n}$ with b fixed. Let $b' := \lfloor b2^w/n \rfloor$. Then $q := \lfloor ab'/2^w \rfloor$; $r := ab - qn$ and subtract n if too great: bound to be in $[0, 2n-2)$. Documented by David Harvey [Har12], more ideas in [Har17]. There's also Montgomery reduction, which is roughly equal performance.

Basic FFT works on transforms whose sizes are powers of 2. This leads to quantization effects. Use [vdH04] which is much closer to a smooth line.

7.2 Empirical effects

Shows four algorithms, and with degree $> 2^7$ multimodular always does better than Karatsuba nd Schönhage–Strassen; for 256-bit prime. Moving to 1024-bit prime, SS is better,and at 4096-bit prime equals multi-modular. Multi-modular is really easy to parallelise. Can either give lots of primes to each core, or can split the FFT if not many primes.

Note Intel has specialist hardware for polynomials over $\mathbf{Z}/2\mathbf{Z}$!. Also a lot of use of Kronecker given his fast univariates.

7.3 Lattices

Integer LLL, Floating LL in double, doubledouble, extended exponents, arbitary precision. Can use Greamm-Schmidt, Givens reduction (better stability). Also using Block Korkin-Zolotarev (better quality basis). Many cryptographic applications.

7.4 HElib

Homomorphic Encryption Library uses NTL. If x and y encrypt C and D, then have $x \oplus y$ encrypts $C + D$, also $\otimes$, for suitable defitnions of $\oplus, \otimes$. With Shai Halevi. Makes heavy use of small-prime FFT. Multiplication of 8-bit numbers is 2.4ms, similar to UNIVAC-1 (1951).

Shows some FLINT/NTL comparisons. In general NTL is significantly faster, or marginally slower.

7.5

Q-MM What next?

A Mostly HElib. But also want to make it GitHub and more maintainable. Also improve matrix arithmetic for large primes (multiplication is multi-modular, but not the rest).

Chapter 8

Thursday 16 July

8.1 Software demo: Zak Tonks

This is the QE package in Maple: see also my talk. Has multiple functions. CAD, Partial CAD. For an existential problem, we produce witnesses.

Q Can I get the description of the true cells?

A Yes, demonstrated. In general they would be interval-indexed `RootOf`

A major part is incrementality. Feature called "lifting constraints" which will built a decomposition of only this ssubset of $\mathbf{R}^n$. These are used purely during lifting?

Questions of whether the roots are paraemtric. Answer offline.

Also has an example where Gröbner makes the equational constraints more complicated.

Input doesn't have to be prenex.

Q Consider the interior of the unit circle: don't you need $y < \sqrt{1-x^2}$? But this doesn't display as that.

A Looks like a bug.

8.2 Terui: Solving System of Nonlinear Equations with te Genetic Algorithm and Newton's Method

Q How do you *know* that you have all the roots?

A? Experimentally.

Q–MJ How does fitness evolve?

A See slide 14.

8.3 Hamada: Multiple Choice Exercises for Computer Algebra Systems

See [HNT20]. Uses LuaTEX, and "Auto Multiple Choice".

Q–MJ Whichtype of classes?

A 1st/2nd year for biologists etc.?

Q How is it available?

A GitHub.

Q Webwork?

A I've heard of it, but not used it.

Q Is this for examinations, or just exercises?

A It seemed to be exercises.

8.4 Grasegger: Flexrilog: Sagemath package for motions of graphs

Check out his Jupyter notebook.

8.5 Abbott: $\mathbf{Q}[x]$

- Degree Set Analysis, e.g. $(3,1)$ and $(2,2)$
- Large Prime Factor Witness. Fixed divisors, to which his answer is Möbius transforms. For eight sqrts, he uses $x \mapsto \frac{26}{105}x$. My prime is ~ 1000 digits.

MM I did this, and my certificate was ~ 1800 digits.

8.6 Monagan: Tangent Graeffe root finding

$P \in F_p[x]$ known to have d distinct roots. cantor–Zassenhaus $O(M(d)(\log p + \log d)\log d)$. 2015 algorithm. out TG is 100 times faster than Magma CZ for polynomials of degree 2^{16}. We have a $\times 2$ in the paper.

Graeffe is $P(x) \mapsto -P(x)P(-x)|_{x^2 \mapsto x}$.

We can factor a polynomial of degree 10^9 for $P = 5 \cdot 2^{55} + 1$ ina 10-core machne with 128GB RAM.

8.7 Hellstrom

Q–MJ JSON or XML?

A Equivalent

Q–MJ Which systems support this?

A Maple.

Bibliography

[BEKS17] J. Bezanson, A. Edelman, S. Karpinski, and V.B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59:65–98, 2017.

[BFT20] M.R. Bender, J.-C. Faugère, and E. Tsigaridas. Gröbner Bases and Sparse Polynomial Systems. `https://videoalbum59f7f8abc4694903b3b1f0a4b1f23fcc91005-master.s3-eu-west-1.amazonaws.com/public/upload/2a05dfe2-dc1f-4a64-adc9-2b7c700cefab.pdf`, 2020.

[CM11] C. Chen and M. Moreno Maza. Algorithms for Computing Triangular Decompositions of Polynomial Systems. In *Proceedings ISSAC '11*, pages 83–90, 2011.

[CR99] P.M. Cohn and C. Reutenauer. On the construction of the free field. *International journal of Algebra and Computation 03n04*, 9:307–323, 1999.

[Har12] D. Harvey. Faster arithmetic for number-theoretic transforms. `http://arxiv.org/abs/1205.2926`, 2012.

[Har17] D. Harvey. Faster truncated integer multiplication. `https://arxiv.org/abs/1703.00640`, 2017.

[HNT20] T. Hamada, Y. Nakagawa, and M. Tamura. Method to Create Multiple Choice Exercises for Computer Algebra System. In *Proceedings ICMS 2020: Mathematical Software*, pages 419–425, 2020.

[Lev20] V. Levandovskyy. The state of the art of Gröbner bases and Gröbner technology in mid-2020. `https://videoalbum59f7f8abc4694903b3b1f0a4b1f23fcc91005-master.s3-eu-west-1.amazonaws.com/public/upload/3f9d6d5c-2a02-4cb8-ba99-0ebb3e82df15.pdf`, 2020.

[MLSAZ20] T. Metzlaff, V. Levandovskyy, H. Schönemann, and K. Abou Zeid. Gröbner Bases over $K\langle X\rangle$ and $\mathbf{Z}\langle X\rangle$ in theory and practice. `https://videoalbum59f7f8abc4694903b3b1f0a4b1f23fcc91005-`

master.s3-eu-west-1.amazonaws.com/public/upload/370d977a-0a28-44b4-b2a9-556f312919e5.pdf, 2020.

[NDSM19] A. Nair, J. Davenport, G. Sankaran, and S. McCallum. Lazard's CAD exploiting equality constraints. *ACM Comm. Computer Algebra*, 53:138–141, 2019.

[vdH04] J. van der Hoeven. The Truncated Fourier Transform and Applications. In J. Gutierrez, editor, *Proceedings ISSAC 2004*, pages 290–296, 2004.

[vH02] M. van Hoeij. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95:167–189, 2002.