Structure in Polynomial Systems

James Davenport

University of Bath and DEWCAD group
https://matthewengland.coventry.domains/dewcad/index.html
EPSRC Grants EP/T015748/1 and EP/T015713/1

1 July 2021

Plan of Talk

1. Effective Algebra requires choices
2. Choices of Orderings
3. Graph Theory?
4. Conclusions and Thanks

Effectiveness imposes choices

For mathematicians, commutative algebra is in $k[x_1, \ldots, x_n]$, with no attention paid to the ordering of the x_i. Most definitions and theorems live in this world. Operations, from the basic $+, -, \times$ to $\sqrt{}$ (finding the radical of an ideal), are well-defined.

But the computer scientist lives in a world of data structures, and wants accessors such as "leading coefficient". Furthermore, the search for algorithms leads us (Thanks, Bruno) to concepts like Gröbner base.

The most fundamental question:

- Distributed : $k[x_1, \ldots, x_n]$, which is typically how the mathematician defines the multivariate polynomials — Gröbner bases;
- Recursive : $k[x_1] \ldots [x_n]$, which is typically how one proves that polynomials over a Noetherian ring are Noetherian (for example) — Regular Chains, Cylindrical Algebraic Decomposition.

Choice of variable order

Even in the recursive format, we have to choose an order: is it $k[x_1]\dots[x_n]$, or $k[x_n]\dots[x_1]$, or any of the $n!$ orders.

Abstractly: the choice doesn't matter, as polynomial rings, they are all isomorphic.

Often: it doesn't matter computationally

Sometimes: it is fundamental [BD07, Theorem 7]: a polynomial p in $3n+4$ variables such that *any* CAD, w.r.t. one order, of R^{3n+4} sign-invariant for p has $O\left(2^{2^n}\right)$ cells, but w.r.t. another order has 3 cells.

Hence numerous heuristics to choose the order [DSS04, Bro04]

And: an interest in machine learning for orders [HEW$^+$19].

The Polynomial

$$p := x^{n+1}\left(\left(y_{n-1}-\frac{1}{2}\right)^2+(x_{n-1}-z_n)^2\right)\left((y_{n-1}-z_n)^2+(x_{n-1}-x_n)^2\right.$$
$$+\sum_{i=1}^{n-1} x^{i+1}\left((y_{i-1}-y_i)^2+(x_{i-1}-z_i)^2\right)\left((y_{i-1}-z_i)^2+(x_{i-1}-x_i)^2\right.$$
$$+x\left((y_0-2x_0)^2+\left(\alpha^2+(x_0-\frac{1}{2})\right)^2\right)\times$$
$$\left((y_0-2+2x_0)^2+\left(\alpha^2+(x_0-\frac{1}{2})\right)^2\right)+a.$$

- The bad order (eliminating x, then $y_0, \alpha, x_0, z_1, y_1, z_1, \ldots,$ x_n, a) needs $O\left(2^{2^n}\right)$ (Maple: 141 when $n=0$) cells.
- Any order eliminating a first says that R^{3n+3} is undecomposed, and the only question is $p=0$, which is linear in a, and we get three cells: $p<0$, $p=0$ and $p>0$.
- However, if we replace a by a^3, the topology is essentially the same, but the discriminant is no longer trivial, and the "good" order now takes 213 cells in Maple.

Choice of monomial order

In the distributed case, we need to do more than order the variables — we have to order the monomials.
For example, does x^2y come before or after xy^{10}?. x^2y wins lexicographically, but xy^{10} wins with total degree. As we know, there is more to ordering than just the variables and degree/lexicographic.
So how do you explain the difference between degree/lexicographic and degree/reverse lexicographic with the variables reversed?

Explaining monomial order (Thanks, Franz)

For three variables, the monomials of degree three are ordered as

$$x^3 > x^2y > x^2z > xy^2 > xyz > xz^2 > y^3 > y^2z > yz^2 > z^3$$

under `grlex`, but as

$$x^3 > x^2y > xy^2 > y^3 > x^2z > xyz > y^2z > xz^2 > yz^2 > z^3$$

under `tdeg`.
One way of seeing the difference is to say that `grlex` with $x > y > z$ discriminates *in favour of* x, whereas `tdeg` with $z > y > x$ discriminates *against* z. This metaphor reinforces the fact that there is no difference with two variables.

Choice of monomial order isn't all

Buchberger's Algorithm requires us to test all pairs $S(g_i, g_j)$, but the *order* in which we do this can be critical for performance. [Buc79, generalised in [BF91]] gives useful criteria for eliminating some pairs, and maximal effectiveness of these imposes some constraints, and we say that we have a *normal selection strategy* if, at each iteration, we pick a pair (i, j) such that $\mathrm{lcm}(\mathrm{lm}(g_i), \mathrm{lm}(g_j))$ is minimal with respect to the ordering in use. Given a tie between (i, j) and (i', j') (with $i < j$, $i' < j'$), we choose the pair (i, j) if $j < j'$, otherwise (i', j') [GMN$^+$91].
A variant is to use a "sugar" strategy, where we consider, not the actual degree of a polynomial, but its "sugar" [GMN$^+$91], i.e. the degree it would have had if we'd homogenised.

Choice of S-polynomial order is still active

[PSHL20] did substantial machine-learning experiments on Buchberger's Algorithm as applied to binomial ideals. They observed that "the agent prefers pairs whose S-polynomials are low degree".

As they stated, this is a new strategy, and seems, on their data, to be an improvement, but this result is subject to confirmation on larger runs.

Graph Theory to the rescue?

Instead of considering degrees of the polynomials in F, consider the graph $\mathcal{G}(F)$ on $\{x_1, \ldots, x_n\}$ with an edge betwen (x_i, x_j) iff there is a polynomial in F contaning both x_i and x_j.

Connectedness?

Gröbner If $\mathcal{G}(F)$ is not connected, the problems are independent, and [Buc79, Criterion 1] will treat them as such.

CAD Essentially independent, but this is hard to describe: we have "the outer product" of the two (or more) CADs. We definitely need to project one component at a time.

Graph Theory to the rescue continued

A graph $\mathcal{G}$ is *chordal* if every every $>$ 3-cycle has a chord. Equivalently, every induced cycle has length 3. Every graph $\mathcal{G}$ has a chordal completion $\overline{\mathcal{G}}$.

Minimum chordal completion is NP-complete [Yan81], but that doesn't really worry me.

If this is the complete graph, then graph theory doesn't seem to help us: the exciting case is when $\overline{\mathcal{G}}$ is smaller.

An ordering $\succ$ on the vertices $x_1, \ldots, x_n$ is a *perfect elimination ordering* if $\forall i$ x_i, x_i and its neighbours $x_j : x_j \prec x_i$ form a clique.

This, and chordality, can be found efficiently [RTL76].

Graph Theory to the rescue continued

Non-trivial chordality has been exploited.

Regular Chains [Che20] shows how it can be exploited efficiently.

Gröbner Bases [CP16] consider "chordal elimination". The challenge here is that an S-polynomial can introduce new edges in $\mathcal{G}$.

CAD [LXZZ21] consider chordality here, ordering x_i in a perfect elimination ordering.

What we currently lack is any view of how common in practice these non-trivial chordal structures are.

Thanks and Conclusions

- Thanks for Franz for many years of interaction,
- and his explanations to me,
- and his service to the computer algebra community in Linz, in Austria and in the world.
- But there are still many unsolved problems for him to look at it in his “retirement”.

Bibliography I

C.W. Brown and J.H. Davenport.
The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition.
In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

J. Backelin and R. Fröberg.
How we proved that there are exactly 924 cyclic 7-roots.
In S.M. Watt, editor, *Proceedings ISSAC 1991*, pages 103–111, 1991.

C.W. Brown.
Tutorial handout.
http://www.cs.usna.edu/~wcbrown/research/ISSAC04/handout.pdf, 2004.

B. Buchberger.
A Criterion for Detecting Unnecessary Reductions in the Construction of Groebner Bases.
In *Proceedings EUROSAM 79*, pages 3–21, 1979.

Changbo Chen.
Chordality Preserving Incremental Triangular Decomposition and Its Implementation.
volume 12097 of *Springer Lecture Notes in Computer Science*, pages 27–38. 2020.
URL: https://www.researchgate.net/publication/342758264_Chordality_Preserving_Incremental_Triangular_Decomposition_and_Its_Implementation.

Bibliography III

D. Cifuentes and P. Parrilo.
Exploiting chordal structure in polynomial ideals: A grobner bases approach.
SIAM Journal on Discrete Mathematics, 30:1534–1570, 2016.

A. Dolzmann, A. Seidl, and Th. Sturm.
Efficient Projection Orders for CAD.
In J. Gutierrez, editor, *Proceedings ISSAC 2004*, pages 111–118, 2004.

A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso.
One sugar cube, please, or selection strategies in the Buchberger algorithm.
In S.M. Watt, editor, *Proceedings ISSAC 1991*, pages 49–54, 1991.

Bibliography IV

Z. Huang, M. England, D. Wilson, J.H. Davenport, and L.C. Paulson.
Using Machine Learning to Improve Cylindrical Algebraic Decomposition.
Mathematics in Computer Science, 11:461–488, 2019.

H. Li, B. Xia, H. Zhang, and T. Zheng.
Choosing the Variable Ordering for Cylindrical Algebraic Decomposition via Exploiting Chordal Structure.
https://arxiv.org/abs/2102.00823, 2021.

D. Peifer, M. Stillman, and D. Halpern-Leistner.
Learning selection strategies in Buchberger's algorithm.
International Conference on Machine Learning, pages 7575–7585, 2020.

Bibliography V

Donald J Rose, R Endre Tarjan, and George S Lueker.
Algorithmic aspects of vertex elimination on graphs.
SIAM Journal on computing, 5(2):266–283, 1976.

Mihalis Yannakakis.
Computing the minimum fill-in is np-complete.
SIAM Journal on Algebraic Discrete Methods, 2(1):77–79, 1981.