Why is quantifier elimination doubly exponential?

James Davenport

University of Bath and DEWCAD group [BDE+21]
https://matthewengland.coventry.domains/dewcad/index.html
(supported by EPSRC under EP/T015713)

Notation

- d The maximum degree (in each variable separately) of the input polynomials. $\mathfrak{d} \leq dn$ total degree.
- l The maximum bit-length of the integer coefficients
- m The number of (distinct) polynomials.
- n The number of variables.
- a The number of alternations of quantifiers. $a \leq n - 1$.
- c The number of equational constraints.
- (M, D) At most M sets, each of combined degree $\leq D$ [McC84].

This is the standard theory setting. Real problems tend to involve rational functions, and rational, or even algebraic, numbers. See [UDE22].

The complexity of QE (and hence CAD) is doubly exponential in n, more precisely $d^{2^{e_d}} m^{2^{e_m}}$ where e_d and e_m depend non-trivially on n (or on a). What are e_d, e_m?

Quantifier Elimination

Given a quantified statement in $n = k + l$ variables

$$Q_1 x_1 \cdots Q_k \Phi(x_1, \ldots, x_k, y_1, \ldots, y_l), \qquad Q_i \in \{\exists, \forall\}$$

find an equivalent quantifier-free formula $\Psi(y_1, \ldots, y_l)$. Our applications will be either **C** (with $+, -, \times, =, \neq$) or **R** (with $+, -, \times, =, \neq, >, \geq, <, \leq$).
Note the absence of division: philosophical and practical issues here [UDE22].
R implies **C** (take real and imaginary parts, and you get $|z|$ and $\overline{z}$ for free). **C** with $\overline{z}$ implies **R**.
Solved by [Tar51, Sei54], but indescribable complexity.
First plausible solution [Col75], via cylindrical algebraic decomposition (CAD), constructed via projection/lifting.

CAD Sign invariant for P

Decomposition: $\mathbf{R}^n = \bigcup_i C_i$ and $i \neq j \Rightarrow C_i \cap C_j = \emptyset$

Cylindrical: If P_m is the projection onto the first m variables, then either $P_m(C_i) = P_m(C_j)$ or $P_m(C_i) \cap P_m(C_j) = \emptyset$.

Also A sample point in each cell, arranged cylindrically.

In fact This is slightly stronger than we need: can relax to "block-cylindrical", where m has to be where $Q_m \neq Q_{m+1}$, i.e. the quantifiers alternate.

(Semi-)Algebraic The boundaries of each cell are semi-algebraic functions, i.e. defined by polynomials and $=, \neq, >, \geq, <, \leq$.

N.B. This is the standard definition, but permits many pathological examples that "no sane algorithm would construct". See [DLS20].

Sign invariant In each C_i every $P_j \in \mathcal{P}$ is positive, or negative, or identically zero, so sample points suffice, and $\forall x_i$ translates to "$\forall x_i$ i-th coordinates of sample points".

Projection/Lifting for a property Z

Given $\mathcal{P}_v$ polynomials in v variables, construct a set $\mathrm{Proj}(P_v)$ in $v-1$ variables such that a CAD of $\mathbf{R}^{v-1}$ Z-invariant for $\mathrm{Proj}(P_v)$ can be lifted to a CAD of $\mathbf{R}^v$ Z-invariant for P_v.

[Col75] Z is "sign". Because a polynomial might vanish identically on a cell, also take subresultants, so $e_m \approx n \log_2 3$.

[McC85] Z is "order". Might fail if a polynomial vanishes identically on a cell. But $e_m \approx n$.

[Bro01] Z is "order", but projection is cheaper.

[Laz94, MPP19] Z is "lex-least", and (Lazard lifting) if a polynomial vanishes identically, divide out the obstruction. Again $e_m \approx n$.

[BM20] Z is "lex-least", but projection is cheaper.

Proj always involves $\mathrm{disc}_{x_v}(p_i)$ and $\mathrm{res}_{x_v}(p_i, p_j)$, hence both degree and number of polynomials squares with each projection.

The problem with iterated resultants

Consider $f_1, f_2, f_3 \in \mathbf{Q}[x, y, z]$ of degree d in each variable.
Then $\mathrm{res}_z(f_1, f_2)$ etc. have degree $2d^2$, and
$R := \mathrm{res}_y(\mathrm{res}_z(f_1, f_2), \mathrm{res}_z(f_1, f_3))$ has degree $8d^4$.
[And so $e_d \approx n$]
But (Bézout) $f_1 = f_2 = f_3 = 0$ has $\le 27d^3$ points (x, y, z).
The problem is that R has as roots

$$\begin{aligned}
\text{(true)}\quad & x : \exists y \exists z f_1(x, y, z) = f_2(x, y, z) = f_3(x, y, z) = 0 \\
\text{(spurious)}\quad & x : \exists y\, [\exists z_1 f_1(x, y, z_1) = f_2(x, y, z_1) = 0] \wedge \\
& \qquad [\exists z_2 f_1(x, y, z_2) = f_3(x, y, z_2) = 0].
\end{aligned}$$

In this case, a Gröbner base [EBD20], or even
gcd $(R, \mathrm{res}_y(\mathrm{res}_z(f_1, f_2), \mathrm{res}_z(f_2, f_3)))$, will solve the problem. Goes some way to explain [McC99a]'s observation that iterated resultants tend to factor.

But in the general case, those "spurious" roots are where the projected topology of $V(f_i)$ changes.

Equational Constraints

[Col98] What if our formula Φ is $f = 0 \wedge \hat{\Phi}$, where $\hat{\Phi}$ involves $m-1$ polynomials g_i?

[McC99b] Answers this: we only need $O(m)$ $\mathrm{res}_x(f, g_i)$, not $O(m^2)$ $\mathrm{res}_x(g_i, g_j)$, since

$$\mathrm{res}_x(g_i, g_j)|_{f=0} \propto \mathrm{res}_y(\mathrm{res}_x(f, g_i), \mathrm{res}_x(f, g_j)). \qquad (1)$$

Means that, after the x projection, we only have $O(m)$ polynomials not $O(m^2)$.

[McC01] Generalises to $f_1 = 0 \wedge \cdots \wedge f_c = 0 \wedge \hat{\Phi}$.

$+$ Reduces e_m from n to $n-c$, nothing for e_d.

[BDE+16] Generalises to where only part of the formula has equational constraints: "truth-table invariant CAD"

[EBD20] Can use Gröbner bases, rather than just iterated resultants, to reduce degree growth, *ideally* e_d becomes $n-c$.

But All this is for the McCallum projection, i.e. well-oriented.

Doesn't Lazard projection/lifting eliminate "well-oriented"?

- $+$ Yes, for straight cylindrical algebraic decomposition
- **But** if $f(x, y, z, \ldots)$ vanishes identically on some surface $S(y, z, \ldots)$, the constant of proportionality in (1) is 0, and we learn nothing about $\mathrm{res}_x(g_i, g_j)$ from $\mathrm{res}_x(f, x_i)$.
- ⚠ "Nullification" has come back to bite us, but only nullification of f, not the g_i.
- **Call** S the *foot* of the "curtain": the "vertical" part of $f = 0$ [NDS20].
- **$\dim(S)$** The case $\dim(S) = 0$ is tractable [Nai21] — see that thesis for more details of $\dim(S) > 0$.

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.

Problem

Recognise, and treat effectively, this case, also "nearly disconnected" (see next)

Graph Theory to the rescue continued

A graph $\mathcal{G}$ is *chordal* if 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: minimal will probably do.

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* (PEO) if $\forall i$ x_i and its neighbours $x_j : x_j \prec x_i$ form a clique. This, and chordality, can be found efficiently [RTL76].

Let n' be the maximal length of a path from x_1 to x_n (as reordered) in $\mathcal{G}$ following $\succ$.

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 $\mathcal{G}$.

Triangular Chordality preserving is proved in [MBL21].

CAD [LXZZ21] consider chordality, ordering x_i in a perfect elimination ordering, then essentially use the same algorithm.

e_d is now n' rather than n (polynomials "drop through" layers!).

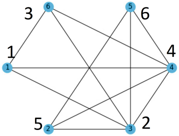
The quantifier structure may be incompatible with the perfect elimination ordering.

What we currently lack is any view of how common in practice these non-trivial chordal structures are, but they are related to "nearly disconnected" $\mathcal{G}$.

But [DM22] in CASC 2022 (being digested)

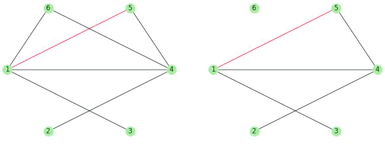
- [MBL21] proves that (sparse) triangular decomposition following a PEO preserves chordal structure.
- But when run in practice, they observe new edges.

Original Chordal Graph



?swap 2,3

Graphs of triangular decompositions



(a) (b)

Extra lines in red

There are four issues:

- Simplifying a Polynomial Set with Its Binomials;
- Simplifying a Polynomial System with Binomials;
- Reducing Inequation Polynomials with a Polynomial in the TS;
- Reducing a TS with a Polynomial in the TS.

? But is it safe to do these as a post-process?

CGB=Comprehensive Gröbner Bases (I) [Wei98, FIS15]

The key idea is this. We consider an "innermost block" in this form:

$$\exists \overline{x} \left(\begin{array}{c} f_1(\overline{y},\overline{x}) = 0 \wedge \cdots f_r(\overline{y},\overline{x}) = 0 \wedge \\ p_1(\overline{y},\overline{x}) > 0 \wedge \cdots p_s(\overline{y},\overline{x}) > 0 \wedge \\ q_1(\overline{y},\overline{x}) \neq 0 \wedge \cdots q_t(\overline{y},\overline{x}) \neq 0 \end{array} \right)$$

where $\overline{y}$ represents the remaining variables, and $f_i, p_j, q_k \in \mathbf{Q}[\overline{y},\overline{x}] \setminus \mathbf{Q}[\overline{y}]$. We introduce new variables $\overline{z}$ and $\overline{w}$, with $\overline{z},\overline{w} \succ \overline{x}$, and consider the polynomials

$$\{f_1,\ldots,f_r,\underbrace{z_1^2p_1-1,\ldots,z_s^2p_s-1}_{\text{forcing positive}},\underbrace{w_1q_1-1,\ldots,w_tq_t-1}_{\text{forcing nonzero}}\}.$$

Let $\mathcal{G} = (S_i, G_i)$ be a Comprehensive Gröbner System (with parameters $\overline{y}$) for this so that $\overline{y}$ space is partitioned by the S_i. We claim each G_i will be $\{f'_1,\ldots,f'_{r'},u_1z_1^2-p'_1,\ldots,u_sz_s^2-p'_s,v_1w_1-q'_1,\ldots,v_tw_t-q'_t\}$. Our answer will be $\bigvee_i \Psi_i(S_i,G_i)$: next two slides explain Ψ_i.

G_i zero-dimensional ($\overline{z}, \overline{w}$ irrelevant for dimension)

If $G_i = (1)$ then we return `false`. Otherwise recall
$G_i = \{f'_1, \ldots, f'_{r'}, u_1 z_1^2 - p'_1, \ldots, u_s z_s^2 - p'_s, v_1 w_1 - q'_1, \ldots, v_t w_t - q'_t\}$.
Let $I = \langle f'_1, \ldots, f'_{r'} \rangle$,

$$\chi(x) = \prod_{(e_1,\ldots,e_s)\in\{0,1\}^s} \chi^I_{(p'_1/u_1)^{e_1},\cdots,(p'_s/u_s)^{e_s}}(x) = x^{2^s d} + \sum_0^{2^s d-1} a_i x^i.$$

The answer is $\Psi_i := \mathcal{F}(S_i) \wedge I_{2^s d}(a_i)$.
JHD: at least that's my reconstruction. I can't see where the w_i (the $\neq 0$) terms come in. Also, the subscript of $\chi^I_{\ldots}$, the characteristic polynomial of $M^I_{\ldots}$, is not a polynomial.

$\exists\phi$: $G_i > 0$-dimensional ($\overline{z}, \overline{w}$ irrelevant for dimension)

$\overline{u} :=$ maximal independent variables $(\overline{x}, G_i, \succ)$. (B)
If $\overline{u} = \overline{x}$ return SYNRAC$(\mathcal{F}(S) \wedge \exists\overline{x}\phi)$ [Wei98]
$\overline{x}' := \overline{x} \setminus \overline{u}$; $\phi_1 := \texttt{Free}(\phi, \overline{x}')$; $\phi_2 := \texttt{NonFree}(\phi, \overline{x}')$;
$\varphi := \phi_1 \wedge \text{Recurse}(S_i, \exists\overline{x}'\phi_2)$ (1)(A)
JHD: I think this means φ now only contains $\overline{u}$-variables
Let $\varphi_1 \vee \cdots \vee \varphi_l$ be a disjunctive normal form of φ. (C)
for $1 \le j \le l$ **do**
$\quad\varphi_j^{(1)} := \texttt{Free}(\varphi, \overline{u})$; $\varphi_j^{(2)} := \texttt{NonFree}(\varphi_j, \overline{u})$;
$\quad\psi_j := \varphi_j^{(1)} \wedge \text{Recurse}(S_i, \exists\overline{u}\phi_j^{(2)})$ (2)(E)
Return $\Psi := \mathcal{F}(S_i) \wedge (\psi_1 \vee \cdots \vee \psi_l)$
JHD: "Recurse" goes right back to the MainQE, note that call (1) has pushed the $\overline{u}$-variables into being parameters (I think) (D). But somehow S_i gets lost in these recursions: I hope I've added it in the right place. Their Theorem 16 states that this does terminate — far from obvious (F).

CGB=Comprehensive Gröbner Bases (IV) [Wei98, FIS15]

- A. Recursing with S is, I think, my interpolation to make sense of the recursions we'll see later. S initially is $\mathbf{R}^{\#\overline{y}}$.
- B. There's a lot of freedom here: ML?
- C. Note that our main recursion is on ϕ in conjunctive normmal form (CNF), whereas here we convert to disjunctive normal form (DNF) and implicitly back at the end of the block. Since CNF$\leftrightarrow$DNF naïvely is exponential, this would provide an exponential blowup at each $\exists/\forall$ boundary, similar to [DH88].
- D. Therefore this recursion is on strictly fewer variables, since dim > 0.
- E. Therefore this recursion is on strictly fewer variables, since $\overline{u} \neq \overline{x}$. $\varphi_j^{(1)}$ is free of $\overline{u}$ by construction, and free of $\overline{x}'$ since it comes from ϕ_1, so actually belongs in an outer block. We might ask why such things exist, but they could be generated by the recursion.
- F. But the two previous notes are probably key.

Complexity of CGB

I know no results on the complexity of Comprehensive Gröbner Bases/Systems.
Since we are doing Gröbner Bases, we might *hope for* singly exponential behaviour at each block, and hence $e_d = O(a)$ rather than $O(n)$, but worst-case Gröbner bases can be doubly exponential [MR13]. *If* we get $O(a)$ behaviour, though, this does not depend on having a lot of equational constraints.
We are doing CNF/DNF conversions at each quantifier alternation, as with VTS, so this could be expected to give us $e_m = O(a)$ rather than $O(n)$.

it's not $\mathbf{R}/\mathbf{C}$: it's quantifiers (and alternations)

[DH88, BD07] Are really about the combinatorial complexity of quantifier alternations

Let $S_k(x_k, y_k)$ be the statement $x_k = f(y_k)$ and then define recursively $S_{k-1}(x_{k-1}, y_{k-1}) := x_{k-1} = f(f(y_{k-1})) :=$

$$\underbrace{\exists z_k \forall x_k \forall y_k}_{Q_k} \underbrace{\left(\underbrace{(y_{k-1} = y_k \wedge x_k = z_k}_{L_k^1}) \vee (\underbrace{y_k = z_k \wedge x_{k-1} = x_k}_{L_k^2}) \right)}_{L_k} \Rightarrow S_k(x_k, y_k).$$

We can transpose this to the complexes, and get zero-dimensional QE examples in $\mathbf{C}^n$ with $2^{2^{O(n)}}$ isolated point solutions, roots of an irreducible polynomial of that degree [DH88]. Or can get that many even though the equations are all linear and the solution set is zero-dimensional [BD07].

Two iterations of Heintz:

$$\exists z_1 \forall x_1 \forall y_1 \left[\left(L_1^1 \vee L_1^2 \right) \Rightarrow \exists z_2 \forall x_2 \forall y_2 \left[\left(L_2^1 \vee L_2^2 \right) \Rightarrow \Phi \right] \right]$$

which becomes

$$\exists z_1 \forall x_1 \forall y_1 \exists z_2 \forall x_2 \forall y_2 \left(L_1^1 \vee L_1^2 \right) \Rightarrow \left[\left(L_2^1 \vee L_2^2 \right) \Rightarrow \Phi \right].$$

The quantified part is then

$$\left(\neg L_1^1 \wedge \neg L_1^2 \right) \vee \left(\neg L_2^1 \wedge \neg L_2^2 \right) \vee \Phi.$$

We will get singly-exponential blow-up as we convert this to Conjunctive Normal Form

Other questions than QE: multistaionarity

Consider ([BDE+17]) a single semi-algebraic set defined by

$$f_1(x_1,\ldots,x_{n-1},k_1)=0 \land f_2(x_1,\ldots,x_{n-1},k_1)=0\land\cdots$$
$$f_{n-1}(x_1,\ldots,x_{n-1},k_1)=0\land x_1>0\land\cdots\land x_{n-1}>0$$

and ask the question "How does the number of solutions vary with k_1?" The f_i are multilinear ($d=1$ but $\mathfrak{d}=2,3,4$) and primitive, and are pretty "generic".

Of course, this doesn't guarantee that all the iterated resultants in [EBD15], or the Gröbner polynomials in [ED16], are primitive, but in practice they are.

In practice can handle k_1, k_2 and looking at k_1, k_2, k_3. But note we want $k_1,\ldots,k_{19}$ for the real biochemical application.

Questions

1. Can we actually say anything about the complexity of GB methods [EBD20]?
2. What happens when we have equational constraints that don't involve the first projection variable?
3. Can we actually say anything about the complexity of CGB-based methods for QE?
4. Can CGB methods, which do QE, actually produce block-cylindrical algebraic decompositions? If so, this would be the first real construction here.
5. Chordality: understand [DM22].
6. Are there any "weak average case complexity" [AL17] results? The examples of [BD07, DH88] seem very special.
7. Understand "which no sane algorithm would construct".

Bibliography

D. Amelunxen and M. Lotz.
Average-case complexity without the black swans.
J. Complexity, 41:82–101, 2017.

A.M. Bigatti, J. Carette, J.H. Davenport, M. Joswig, and T. de Wolff, editors.
Mathematical Software — ICMS 2020, volume 12097 of *Springer Lecture Notes in Computer Science*. Springer, 2020.

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.

Bibliography II

R.J. Bradford, J.H. Davenport, M. England, S. McCallum, and D.J. Wilson.
Truth table invariant cylindrical algebraic decomposition.
J. Symbolic Comp., 76:1–35, 2016.

R.J. Bradford, J.H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Kosta, O. Radulescu, T. Sturm, and A. Weber.
A Case Study on the Parametric Occurrence of Multiple Steady States.
https://arxiv.org/abs/1704.08997, 2017.

Bibliography III

R.J. Bradford, J.H. Davenport, M. England, A. Sadeghimanesh, and A. Uncu.
The DEWCAD Project: Pushing Back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition.
ACM Comm. Computer Algebra, 55(3):107–111, 2021.

C.W. Brown and S. McCallum.
Enhancements to Lazard's Method for Cylindrical Algebraic Decomposition.
Computer Algebra in Scientific Computing. CASC 2020, pages 129–149, 2020.

C.W. Brown.
Improved Projection for Cylindrical Algebraic Decomposition.
J. Symbolic Comp., 32:447–465, 2001.

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.
In Bigatti et al. [BCD+20], pages 27–38.

G.E. Collins.
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition.
In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

V

G.E. Collins.
Quantifier elimination by cylindrical algebraic decomposition — twenty years of progess.
In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 8–23. Springer Verlag, Wien, 1998.

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

J.H. Davenport and J. Heintz.
Real Quantifier Elimination is Doubly Exponential.
J. Symbolic Comp., 5:29–35, 1988.

J.H. Davenport, A.F. Locatelli, and G.K. Sankaran.
Regular cylindrical algebraic decomposition.
J. LMS., 101:43–59, 2020.

Mingyu Dong and Chenqi Mou.
Analyses and Implementations of Chordality-Preserving Top-Down Algorithms for Triangular Decomposition.
In François Boulier, Matthew England, Timur M. Sadykov, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing CASC 2022*, volume 13366 of *Lecture Notes in Computer Science*, pages 124–142, 2022.

M. England, R. Bradford, and J.H. Davenport.
Improving the Use of Equational Constraints in Cylindrical Algebraic Decomposition.
In D. Robertz, editor, *Proceedings ISSAC 2015*, pages 165–172, 2015.

M. England, R.J. Bradford, and J.H. Davenport.
Cylindrical Algebraic Decomposition with Equational Constraints.
In J.H. Davenport, M. England, A. Griggio, T. Sturm, and C. Tinelli, editors, *Symbolic Computation and Satisfiability Checking: special issue of Journal of Symbolic Computation*, volume 100, pages 38–71. 2020.

VIII

M. England and J.H. Davenport.
The Complexity of Cylindrical Algebraic Decomposition with Respect to Polynomial Degree.
In V.P. Gerdt, W. Koepf, W.M. Seiler, and E.V. Vorozhtsov, editors, *Proceedings CASC 2016*, volume 9890 of *Springer Lecture Notes in Computer Science*, pages 172–192. Springer, 2016.

R. Fukasaku, H. Iwane, and Y. Sato.
Real Quantifier Elimination by Computation of Comprehensive Gröbner Systems.
In D. Robertz, editor, *Proceedings ISSAC 2015*, pages 173–180, 2015.

D. Lazard.
An Improved Projection Operator for Cylindrical Algebraic Decomposition.
In C.L. Bajaj, editor, *Proceedings Algebraic Geometry and its Applications: Collections of Papers from Shreeram S. Abhyankar's 60th Birthday Conference*, pages 467–476, 1994.

H. Li, B. Xia, H. Zhang, and T. Zheng.
Choosing the Variable Ordering for Cylindrical Algebraic Decomposition via Exploiting Chordal Structure.
ISSAC '21: Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation, pages 281–288, 2021.

Bibliography

C. Mou, Y. Bai, and J. Lai.
Chordal Graphs in Triangular Decomposition in Top-Down Style.
Journal of Symbolic Computation, 102:108–131, 2021.

S. McCallum.
An Improved Projection Operation for Cylindrical Algebraic Decomposition.
PhD thesis, University of Wisconsin-Madison Computer Science, 1984.

S. McCallum.
An Improved Projection Operation for Cylindrical Algebraic Decomposition.
Technical Report 578, Computer Science University Wisconsin at Madison, 1985.

XI

S. McCallum.
Factors of iterated resultants and discriminants.
J. Symbolic Comp., 27:367–385, 1999.

S. McCallum.
On Projection in CAD-Based Quantifier Elimination with Equational Constraints.
In S. Dooley, editor, *Proceedings ISSAC '99*, pages 145–149, 1999.

S. McCallum.
On Propagation of Equational Constraints in CAD-Based Quantifier Elimination.
In B. Mourrain, editor, *Proceedings ISSAC 2001*, pages 223–230, 2001.

S. McCallum, A. Parusiński, and L. Paunescu.
Validity proof of Lazard's method for CAD construction.
J. Symbolic Comp., 92:52–69, 2019.

E.W. Mayr and S. Ritscher.
Dimension-dependent bounds for Gröbner bases of polynomial ideals.
J. Symbolic Comp., 49:78–94, 2013.

A.S. Nair.
Curtains in Cylindrical Algebraic Decomposition.
PhD thesis, University of Bath, 2021.

Bibliography XIII

A.S. Nair, J.H. Davenport, and G.K. Sankaran.
Curtains in CAD: Why Are They a Problem and How Do We Fix Them?
In Bigatti et al. [BCD+20], pages 17–26.

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.

A. Seidenberg.
A new decision method for elementary algebra.
Ann. Math., 60:365–374, 1954.

A. Tarski.
A Decision Method for Elementary Algebra and Geometry.
2nd ed., Univ. Cal. Press. Reprinted in *Quantifier Elimination and Cylindrical Algebraic Decomposition* (ed. B.F. Caviness & J.R. Johnson), Springer-Verlag, Wein-New York, 1998, pp. 24–84., 1951.

A.K. Uncu, J.H. Davenport, and M. England.
SMT-Solving Combinatorial Inequalities.
To appear in Proc. SCSC 2022, 2022.

V. Weispfenning.
A New Approach to Quantifier Elimination for Real Algebra.
In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 376–392. Springer-Verlag, 1998.

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