Notes by J.H.Davenport

16–20 August 2021

Contents

1 16 August **3**
- 1.1 Le and Safey El Din . . . 3
- 1.2 Moustrou & Reiner: Specht polynomials . . . 3
  - 1.2.1 Q&A . . . 3
- 1.3 Davenport . . . 3
- 1.4 Schost: Bit Complexity . . . 4

2 August 17 **5**
- 2.1 Christina Katsamaki: Algorithms and Algebraic Operations on Parametric Curves . . . 5
  - 2.1.1 Topology of a parametric curve . . . 5
- 2.2 Agnes Szanto: Smooth Points on Real Algebraic Sets . . . 5
- 2.3 Alicia Dickenstein: Sign Conditions for the Existence of at Least One Positive Solution of a Sparse Polynomial System . . . 6

3 Thursday 19th August SC^2 I **7**
- 3.1 Matthew England — SC-Square: Past Successes and Future Progress with Machine Learning . . . 7
  - 3.1.1 Machine Learning . . . 7
  - 3.1.2 My work . . . 7
  - 3.1.3 . . . 7
- 3.2 Baptiste Vergain: Decidable Logics with Arithmetic and Uninterpreted Symbols for SMT . . . 8
  - 3.2.1 . . . 8
- 3.3 Grant Olney Passmore: An Introduction to the Imandra Automated Reasoning System . . . 8
  - 3.3.1 Q&A . . . 8
  - 3.3.2 "Widely used in Finance" . . . 9
- 3.4 Martin Brain: Further Steps Down The Wrong Path: Improving Multiplication in Bit-Blasting . . . 9
  - 3.4.1 Q&A . . . 9

4 Friday 20th August SC^2 II **10**
4.1 Michela Ceria: Degroebnerization and Its Applications: a New Approach for Reverse Engineering of Gene Regulatory Networks 10
4.1.1 Q&A . . . 10
4.2 Alexei Lisitsa: Constraint Satisfaction for Gauss Diagrams Enumeration . . . 10
4.2.1 Q&A . . . 11
4.3 Jasper Nalbach: Extending the Fundamental Theorem of Linear Programming for Strict Inequalities . . . 11
4.3.1 Q&A . . . 11
4.4 Philippe Specht: Level-Wise Construction of a Single Cell in Cylindrical Algebraic Decomposition . . . 11

5 Friday 20th August SC^2 III **13**
5.1 Zak Tonks: Practical Evaluation of QE Methods . . . 13
5.1.1 Q&A . . . 13
5.2 Akshar Nair: Equational Constraints, the Lazard Projection and the Curtain Problem . . . 13
5.2.1 Q&A . . . 13
5.3 Christopher Brown: Avoiding Work in CAD: NLSAT, CDCAC, NuCAD, etc. . . . 13
5.3.1 Q&A . . . 14
5.4 Vijay Ganesh: Logic Solvers and Machine Learning: The Next Frontier . . . 14
5.4.1 Q&A . . . 15

Chapter 1

16 August

1.1 Le and Safey El Din

Many examples, e.g. [MBD+18, LMP21]. Classically CAD, but doubly exponential in $n + t$.

Can reduce to parametric problem in one variable, but again doubly exponential in t =#parameters.

Wnt border polynomials/discriminant varieties. Skipping the boundaries gives us singly-exponential complexity.

We assume f generates a zero-dimensional radical ideal. Examples for random dense polynomials: much faster than Maple (root finding or regular chains). Let $B = nD(D-1)^n)$, then depends on B^{3t}. Use grevlex and generic staircase [MS03].

Main cost is given in [LSED20], depends on degree bound of minors of H. Again have much faster timings.

1.2 Moustrou & Reiner: Specht polynomials

Does a symmetric problem have a symmetric solution? Note the "half degree principle" from [Reiner2012].

1.2.1 Q&A

Q Does this give you faster algorithms?

A We haven't got there yet.

1.3 Davenport

See `https://staff.bath.ac.uk/masjhd/Slides/JHDSIAMAG21Monday.pdf`.

1.4 Schost: Bit Complexity

Some basic questions

- Is $V \cap \mathbf{R}^n$ empty
- Find a point in each real-connected component (we focus on this).
- Many operations have algebraic complexity $d^{O(n)}$ or $d^{O^\sim(n)}$.
- We would hope that complexity is $O^\sim(h)\times$algebraic complexity

There is good bit complexity for infinitesimal-based algorithms, but these are hard to analyse. We'll look at polar variety algorithms.

Assume radical and V is smooth and equidimensional.

Theorem 1 *With probabiliyt $> 1-\epsilon$, we can compute at least one point in each connected component with $O^\sim(hd^n \log^2(1/\epsilon)$.*

Idea based on [SEDS03]. In generic coordinates, our system has finitely many solutions. need to prove that the unlucky substitutions lie in a suitable subvariety etc. Need a constructive version of Thom's transversality theorem.

Need a construction from [Giustietal1997].

Chapter 2

August 17

2.1 Christina Katsamaki: Algorithms and Algebraic Operations on Parametric Curves

Two problems

2.1.1 Topology of a parametric curve

Well-studied in 2D [KS14].

2.2 Agnes Szanto: Smooth Points on Real Algebraic Sets

$\dim_{\mathbf{C}}(V)$ is $d^{O(n)}$, but $r := \dim_{\mathbf{R}}(V)$ is $d^{O(r(n-r))}$ — gap here.

Finding smooth points in each component: many works. [BPR06, Chapter 13] in general. Critical point methods are only guaranteed to work for smooth varieties. For real dimension, see [LSED21] for the latest.

Theorem 2 *Assume $V(f_1, \ldots, f_s)$ is equidimensional of dimension $n-s$. Suppose the polynomial g has Sing(V)$\cap \mathbf{R}^n \subset V(g)$ and $\dim(V \cap V(g)) < n-s$. Then the set of poiynts where $g|_{V\cap\mathbf{R}^n}$ attains its extreme values on every component of $V \cap \mathbf{R}^n$.*

[MorkPiene2008] example where we can't get smooth points on all four components.

Example 1 (Kuramoto $n = 4$) *Discriminant $D(w)$ has degree 48. Bézout is $47^3 > 100K$, reduced by symmetry to 6400, and verified real solutions numerically.*

Theorem 3 ([Marshall2008]) $\dim_{\mathbf{R}} = \dim_{\mathbf{C}}$ *iff there is a smooth point on $V \cap \mathbf{R}^n$.*

If this doesn't work, replace V by a smaller V' with the same real points: limit of perturbed (in the sense of small $\epsilon > 0$) polar varieties.

Future work is removing some of these assumptions. See [HHS20].

2.3 Alicia Dickenstein: Sign Conditions for the Existence of at Least One Positive Solution of a Sparse Polynomial System

See [BDG19]. Let $a_1, \ldots, a_n$ be the support of the sparse system. We would like an algorithm with sufficient conditions to predict the answer. Let C be the matrix of coefficient. We need $\mathbf{0} \in$ positive cone spanned by columns of C. [MulleretalFOCM2016] gives a condition for ≤ 1 positive root. Hence the case of > 1 root is more complicated.

[Conradietal, PLoS C 2017]. This generalises the Intermediate Value Theorem from $d = 1$.

We get a bijection between positive solutions of the original system and the solutions of the Gale dual system.

Theorem 4 ([BDG19], Theorem 1) *Under these assumptions, $n_A(C) > 0$.*

[FisherShapiro1996] has definitions of mixed and dominating matrices.

But note that when the ideal is not CM, there is no simple connection between the algebra and the geometry.

Chapter 3

Thursday 19th August SC^2 I

3.1 Matthew England — SC-Square: Past Successes and Future Progress with Machine Learning

Introduces SC^2, and slide from [Á15]. EU project, and success at [DEG+20]. Mentions NLSAT, NuCAD, REDLOG+VerIT. MathCheck. Also many others.

3.1.1 Machine Learning

Note that ML is fallible, for a variety of reasons, but for us this is bad, but not a disaster. Note that my work is not the same as the Facebook work on actually computing integrals my machine learning [KTYF21, ?]. Note [XHHLB08] on Satzilla. Many successes.

3.1.2 My work

Own example of 3 cells versus 59 in 2D. [HEW+14] showed ML can do better than any one. Still don't do as well as a human heuristic. But problem is having enough, and varied enough, datasets.

3.1.3

PF Can you get hieristics a human can understand, e.g. reverse engineering?

A Tried this — not particualrly successful. We had a human-understandable set of features, though.

3.2 Baptiste Vergain: Decidable Logics with Arithmetic and Uninterpreted Symbols for SMT

Note work in progress. We are at the frontier of decidability. In particular difference logic with unary predicates. $x - y\sigma c$ with $\sigma \in \{=, \neq, <\ .\ >, \leq, ge\}$. Therefore can define parity. On **N** this is equivalent to S1S (Büchi automata). Existential quantification is automaton projection.

Extension to **Z** uses the susual $\mathbf{N} \leftrightarrow \mathbf{Z}$ bijection: the automata are bigger. But **R** is different. Example which is true $< a$ and chaos $> a$. Examples of operators on behaviours.

3.2.1

MJB Examples in this subset? It is implementable with existing quantifier technqiues?

A–PF Not really example-driven.

MJB If this automaton can do quantifier instantiation, this would be helpful.

Q–JHD **Q**?

A Probably like **R**. Grant Passmore claims equisatisfiability.

3.3 Grant Olney Passmore: An Introduction to the Imandra Automated Reasoning System

Note IJCAR systems paper. A new(ish) theorem prover based on OCAML[1]. Sales pitch: "A new cloud-native automated reasoning system".

We have an axiomatic semantics for a subset of OCAML. Extreme focus on automation. A key idea is "first class counterexamples": common in algebra. A conjecture about a program is itself a program. Hence we ask "does this always evaluate to true". Hence a counter-example is a finite call graph. We have a avriety of mechanisms for connecting tools.

Interactive example, showing finding a counterexample of a polymorphic statement falsified: synthesising a function and its type. Another example (reverse of reverse) : "no counterexamples < 100, so ask for a proof". So does a proof.

3.3.1 Q&A

Q–MJB What theories?

A We are guided by UNSAT cores in SMT, so need that. We have some SMT solvers ourselves, but also a plug-in system for other solvers.

[1] "Widely used in finance" — see §3.3.2: also Coq.

3.3.2 "Widely used in Finance"

JHD asked Grant to expand on this.

"Re OCaml, yes it is widely used in finance. A few examples:

1. Bloomberg Terminals ship with the LexiFI suite for describing and pricing structured products (derivatives). This is written in OCaml, and the DSL for describing derivatives is actually a collection of OCaml combinators.

2. Jane Street is a major prop trading shop which uses OCaml

3. Goldman Sachs, Citi, Itiviti, OneChronos, etc., all use OCaml through their use of Imandra :-)."

JHD regards (1) as particularly telling. `https://en.wikipedia.org/wiki/Bloomberg_Terminal` claims "As of October 2016, there were 325,000 Bloomberg Terminal subscribers worldwide". The current version of Bloomberg's corporate information makes the same claim.

3.4 Martin Brain: Further Steps Down The Wrong Path: Improving Multiplication in Bit-Blasting

Bitblasting takes QF-bitvectors (i.e. $\mathbf{Z}/2^n$) into booleans. Tables of various operators, and whether propositionally-complete. Multipliers are quadratic,and not propositionally complete, which is a worry. The smallest encoding is $O(n \log n)$ but propositionally complete is 2^n. However, I will talk about it. Example reducing $975 \times N$ from seven additions to two. Good general example. Then shows a real example of a NN with 36 multiplies by constants (note that we're allowed to share subterms). But it's NP-complete, so we have an NP to feed to an NP SAT solver!

Multiplication by Polynomial Interpolation. Note that we can pick nice evaluation points, e.g. 0, 1, -1. Substantial savings.

3.4.1 Q&A

Q–PF How much can you leave out if ....

A You can leave some out, but I'd like to have a better theory.

Chapter 4

Friday 20th August SC^2 II

4.1 Michela Ceria: Degroebnerization and Its Applications: a New Approach for Reverse Engineering of Gene Regulatory Networks

Only need linear algebra. See [LS04]. Have various coordinate functions, which can be viewed as polynomials. Need normal forms of polynomials, which they propose to get via GB. Quadratic in #variables, exponential in #Points.

4.1.1 Q&A

Brandilyn What is the role of the bars?

A The bar (kth bar) represents degree k of the corresponding monomial. This essentially computes a lex basis.

Q Other basis?

A Not yet.

Brandilyn

4.2 Alexei Lisitsa: Constraint Satisfaction for Gauss Diagrams Enumeration

How to extract Gauss diagram from a curve. The interlacement graph: nodes connected if corresponding edges in Gauss diagram intersect. A diagram is prime iff interlacement graph is connected. Diagrams are equivalent if diagram graphs isomorphic. Then Gauss asked which chord diagrams are realisable by planar graphs. Guass had a necessary conditin. Sufficient conditions by [Dehn1936].

Claims that these can be formulated in terms of interlacement graph: [Rozenthiel1976]. The GL- and B- conditins have no second-order quantification.

But even with simple symmetry breaking, time grows fast (miniZinc solver).

Hence we have an on-the-fly checker in SWi-Prolog. Direct and incremental algortihms implemented. There is a bijection between S_n and even/odd matchings. A prime Gauss diagram n which is not a wheel can be generated from an $n-1$ prime diagram by a teepee move.

Note that we have mistakes in main theorems of papers in IJKTR (? were these GL and B?): counterexamples start at $n = 9$ and many for larger n. See [KLLV21].

But quantified SAT would allow for direct implementation of [Rozenthiel1976].

4.2.1 Q&A

MJB Would "search over non-isomorphic graphs" be a useful primitive?

A Yes, very much.

4.3 Jasper Nalbach: Extending the Fundamental Theorem of Linear Programming for Strict Inequalities

Simplex for lazy SMT [DdM06] but no proof. [KingPhD2014] gave a proof. We generalise this.

Lazy SMT and Theory solvers. Because of negation, we need to handle weak constraints. Formally $\leq$ sufices. $\max_{\mathbf{x}} c^T.x | u.x \leq b$ is our optimisation problem. C is satisfiable iff there's a satisfying vertex.

Add a weakening ϵ. What to prove $\alpha \vDash C \Leftrightarrow \alpha \vDash C_w \wedge \epsilon > 0$.

But there can be some combinatorial explosion if adding ϵ increased $\tilde{\text{v}}$ertices.

4.3.1 Q&A

Q–JHD MIP?

A Related, but we generally need more theory.

4.4 Philippe Specht: Level-Wise Construction of a Single Cell in Cylindrical Algebraic Decomposition

Given a set of constraints, and a sample point, construct a cell which is sign-invariant for the polynomials. Consider mcSAT [JdM13].

We start levelwise, substituting all of teh sample point except for x_n, so we get a cell in the nth dimension. Then iterate.

Also refinement-based CAD [BK15]. Used 11512 benchmarks from SMTLIB QF_NRA. They solve 9293 and 9303 problems respectively, but intersection in 4227. On these level-wise is 7% faster.

Chapter 5

Friday 20th August SC^2 III

5.1 Zak Tonks: Practical Evaluation of QE Methods

Talk given by Ali Uncu.

5.1.1 Q&A

Q–ME What is the ECheuristic — how is this relevant to the others, which are variable orderings not EC-ordering.

A–JHD ECHeuristic is Algorithm 36 in Zak's thesis. It's a variant of Brown, aimed at cases where there are ECs as well.

5.2 Akshar Nair: Equational Constraints, the Lazard Projection and the Curtain Problem

5.2.1 Q&A

Q–CWB If there are no equational constraints, do you need all the lc, tc and disc from A rather than E (which you certainly need)?

A I think that's right.

5.3 Christopher Brown: Avoiding Work in CAD: NLSAT, CDCAC, NuCAD, etc.

Very inspired by NLSAT [JdM13] and CAC [ADEK20]. How does this relate to NuCAD [Bro15]? CAD: Projection gives us 2K polynomials, degree up to 27,

1986 cells in $\mathbf{R}^1$. But NLSAT has 8 open cells, others 15 or 31. Note also that if you're looking for open cells, we can ignore end-points.

Example 2 (NLSAT) *Choose $x = 0$, $y = 0$ gives UNSAT (odd negation of the contradiction), so $y = -1$, and get another cell. Eventually rules out $x = 0$, and we try a different x.*

Example 3 (CDCAC) *Having failed $x = 0$, we look at the projections of the polynomials, and get a bad interval for x. Note that there are (implicit) cells. But at a new $x = 1$, say, we also sample $y = 0$ and find the same conflict (whereas NLSAT would have generalised).*

Example 4 (NuCAD) *We keep a priority queue of unanalysed cells. As CDCAC have to do the splits multiple times.*

All three are conflict-guided, which means we're lazy w.r.t. projection. But note that they all trade this for a more complicated SMT-like control mechanism, especially NLSAT. NLSAT builds complete cells as it goes, whereas CDCAC will only have a complete cell with SAT. NuCAD doesn't allow overlap, but it doesn't have cylindricity constraints.

There are commonalities: fixed variable ordering, and some cell orientation.

5.3.1 Q&A

Q-JHD Always end up with the same SAT sample point?

A Not quite a coincidence - I made the same "arbitrary" choices.

5.4 Vijay Ganesh: Logic Solvers and Machine Learning: The Next Frontier

CWB pointed out the range of arbitrary choices, and ML is a way of addressing this. SAT solvers currently are a soup of heuristics: which actually matter? I want to view solvers as proof systems, with a natural way of using ML. There are a range of branching heuristics. We regard these as heuristics to maximise global learning rate.

Example of variable choice as multi-armed bandits. Look at results from various arms, and use exponential moving averages. Then apply RL here. Example: between "A assigned" nad "A unassigned" is is used in 2 out of 3 learns. Then do exponential moving averages on this ratio, to decide best variables. Shows cactus plot, with his MiniSAT+LRB beating standard MiniSAT.

But is global learning rate the best thing? Seems to be.

This is not the whole reason why SAT solvers do well, especilaly on industrial instances.

5.4.1 Q&A

Q

A

Bibliography

[Á15] E. Ábrahám. Building Bridges between Symbolic Computation and Satisfiability Checking (slides). *Invited Talk at ISSAC 2015*, 2015.

[ADEK20] E. Ábrahám, J.H. Davenport, M. England, and G. Kremer. Deciding the Consistency of Non-Linear Real Arithmetic Constraints with a Conflict Driven Search Using Cylindrical Algebraic Coverings. *To appear in Journal of Logical and Algebraic Methods in Programming*, 2020.

[BDG19] F. Bihan, A. Dickenstein, and M. Giaroli. Sign conditions for the existence of at least one positive solution of a sparse polynomial system. `https://arxiv.org/abs/1908.05503`, 2019.

[BK15] C. Brown and M. Košta. Constructing a single cell in cylindrical algebraic decomposition. *J. Symbolic Computation*, 70:14–48, 2015.

[BPR06] S. Basu, R. Pollack, and M.-F. Roy. Algorithms in Real Algebraic Geometry, 2nd ed. *Springer*, 2006.

[Bro15] C.W. Brown. Open Non-uniform Cylindrical Algebraic Decompositions. In *Proceedings ISSAC 2015*, pages 85–92, 2015.

[DdM06] B. Dutertre and L. de Moura. A fast linear-arithmetic solver for DPLL (T). *In International Conference on Computer Aided Verification*, pages 81–94, 2006.

[DEG+20] 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, 2020.

[HEW+14] Z. Huang, M. England, D. Wilson, J.H. Davenport, L.C. Paulson, and J. Bridge. Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition. In S.M.Watt *et al.*, editor, *Proceedings CICM 2014*, pages 92–107, 2014.

[HHS20] K. Harris, J.D. Hauenstein, and A. Szanto. Smooth Points on Semi-algebraic Sets. https://arxiv.org/abs/2002.04707, 2020.

[JdM13] D. Jovanović and L. de Moura. Solving non-linear arithmetic. *ACM Communications in Computer Algebra*, 46(3/4):104–105, 2013.

[KLLV21] A. Khan, A. Lisitsa, V. Lopatkin, and A. Vernitski. Circle graphs (chord interlacement graphs) of Gauss diagrams: Descriptions of realizable Gauss diagrams, algorithms, enumeration. https://arxiv.org/abs/2108.02873, 2021.

[KS14] A. Kobel and M. Sagraloff. On the Complexity of Computing with Planar Algebraic Curves. *To appear in J. Complexity*, 2014.

[KTYF21] H. Kubota, Y. Tokuoka, T.G. Yamada, and A. Funahashi. Symbolic integration by integrating learning models with different strengths and weaknesses. https://arxiv.org/abs/2103.05497, 2021.

[LMP21] H.P. Le, D. Manevich, and D. Plaumann. Computing totally real hyperplane sections and linear series on algebraic curves. https://arxiv.org/abs/2106.13990, 2021.

[LS04] R. Laubenbacher and B. Stigler. A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology*, 229:523–537, 2004.

[LSED20] H.P. Le and M. Safey El Din. Solving parametric systems of polynomial equations over the reals through Hermite matrices. https://arxiv.org/abs/2011.14136, 2020.

[LSED21] P. Lairez and M. Safey El Din. Computing the dimension of real algebraic sets. https://arxiv.org/abs/2105.10255, 2021.

[MBD+18] C.B. Mulligan, R. Bradford, J.H. Davenport, M. England, and Z. Tonks. Quantifier Elimination for Reasoning in Economics. https://arxiv.org/abs/1804.10037, 2018.

[MS03] G. Moreno-Socías. Degrevlex Gröbner bases of generic complete intersections. *Journal of Pure and Applied Algebra*, 180:263–283, 2003.

[SEDS03] M. Safey El Din and É. Schost. Polar Varieties and Computation of One Point in Each Connected Component of a Smooth Real Algebraic Set. In J.R. Sendra, editor, *Proceedings ISSAC 2003*, pages 224–231, 2003.

[XHHLB08] L. Xu, F. Hutter, H.H. Hoos, and K. Leyton-Brown. SATzilla: Portfolio-based Algorithm Selection for SAT. *Journal of Artificial Intelligence Research*, 32:565–606, 2008.