ISSAC 2012

Notes by J.H. Davenport — J.H.Davenport@bath.ac.uk

22–25 July 2012

Abstract

Because there were parallel sessions, these notes are very incomplete, and not every talk is even mentioned.

The proceedings were printed locally (JHD in fact only has an e-copy), but should be in the ACM Digital Library.

Contents

Chapter 1

Tutorials: 22 July 2012

1.1 Elements of Computer-Algebraic Analysis — Levandovskyy

List of literature and software, notably Singular/Plural [GPS01] with its D-module suite [ABL$^+$10], and work by Chyzak and Koutschan.

Algebraic Analysis arose in 1958 in the group of Mikio Sato (Japan). Discussed systems of linear PDEs with variable coefficients, and generalized functions.

1.1.1 Part 1a

Let K be an effective field, and let $\mathcal{F}$ be a K-vector space. If x is a local coordinate on $\mathcal{F}$. It induces an J-linear map.

$\partial(xf(x) = x\partial(f(x)) + f(x)$ so $\partial \circ x - x \circ \partial - 1 = 0$. Hence we get the first Weyl algebra:

$$D_1 = K\langle x, partial | \partial x = x\partial + 1\rangle$$

Let s be the shift operator: $s(g(k)) = g(k+1)$. The corresponding space is the first **shift algebra**:

$$S_1 = K\langle k, s || sk = (k+1)s\rangle.$$

We can mix the two:

$$A = D_1 \otimes S_1 = K\langle x, partial, k, s | \partial x = x\partial + 1, sk = (k+1)s, \mathcal{J}\rangle$$

where $\mathcal{J}$ indicates that all other generators commute.

The first q-Weyl algebra:

$$D_1^{(q)} = K\langle x, \partial_q | \delta_q x = qx\delta_q + 1\rangle,$$

and similarly for shifting.

Definition 1 *Algebra with linear (affine) relations:*

$$A^{(1)}(q,\alpha,\beta,\gamma) := K\langle xy|yx - q\cdot xy = \alpha x + \beta y + \gamma.$$

Theorem 1 (L–Koutschan–Motsak, 2011) *$A^{(1)}(q,\alpha,\beta,\gamma)$ is isomorphic to one of five model algebras:*

Theorem 2 (L–Makedonsky–Petravchuk, new) *For given $N \geq 2$, $A^{(2)}(q,c_0,\ldots,c_N,\alpha)$ is isomorphic to one of three model algebras:*

Hence various isomorphism theorems. Tangent algebra $K\langle \tan, \partial | \partial\cdot\tan = \tan\cdot\partial + \tan^2 + 1\rangle$. This is a subalgebra of the 1st Weyl algebra, generated by $Y = -x$ and $X = (x^2+1)\partial$, so $YX = XY + Y^2 + 1$.

Q But every subalgebra has two generators.

A No: every *ideal* does, but this is not an ideal.

1.1.2 Part Ib: G-algebras

If $R = K[x_1,\ldots,x_n]$, then we have standard monomials, well-orderings and monomials orderings: $\alpha \prec \beta \Rightarrow x^\alpha \prec x^\beta$ and compatible.

Suppose we have K, R and $c_{i,j} \in K^*, d_{i,j} \in R : 1 \leq i < j \leq n$. Assume also $\mathrm{lm}(d_{i,j} \prec x_i x_j$. Then we get an algebra

$$A = K\langle x_1,\ldots,x_n | x_j x_i = x_{i,j} x_i x_j + d_{i,j}\rangle.$$

Divisibility can be defined as usual: $\mathrm{x}\alpha | x_b eta \Leftrightarrow \forall i \alpha_i < \beta_i$. Then let $\gamma = \beta - \alpha$. Then $\mathrm{lm}(x^\alpha x^\gamma = x^\beta$, and hence (left) Gröbner bases.

Hence the Gröbner Trinity.

- left Gröbner basis of a submodule of a free module
- The left syzygy module of a given set of generators
- left transformation matrix, expressing elements of the Gröbner basis in terms of the original generators.

1.1.3 Part II: Dimension Theory

Again systems of equations become modules. But different matrices P can represent the same module. So let $\mathcal{F}$ be a left D-module (not necessarily finitely presented) and $\mathcal{P}$ a system of equations, then $\mathrm{Sol}_D(\mathcal{P},\mathcal{F}) := \left\{ f \in \mathcal{F}^{m\times 1} : P \bullet f = 0 \right\}$.

Theorem 3 (Noether–Malgrange Isomorphism) *There is an isomorphism of K-vector spaces*

$$\underset{D}{\mathrm{Hom}}(M,\mathcal{F}) = \underset{D}{\mathrm{Hom}}(D^{1\times m}/D^{1\times l}\mathcal{P},\mathcal{F}) \equiv \underset{D}{\mathrm{Sol}}(\mathcal{P},\mathcal{F}) :$$

$$(\phi : M \to \mathcal{F}) \mapsto (\phi([e_1]),\ldots,\phi([e_m])) \in \mathcal{F}^{m\times 1}.$$

$m \in \mathcal{F}$ is called a **torsion element** if $\mathrm{Ann}_D\, m \neq 0$. Many classical functions in common functional spaces are torsion.

This can model polynomial–exponential signals by linear systems. Operator algebras with polynomial coefficients is the right way to go (rather than larger operators with constant coefficients).

The generalization of Krull dimension is Krull–Rentschler–Gabriel dimension, which is not algorithmic. One can look at the projective dimension, which is algorithmic, but expensive. Global homological dimension is not algorithmic. Homological grade is slightly cheaper than the projective dimension, and implemented. There is also Gel'fand–Kirillov dimension, which is cheap to compute.

Let A be a K-algebra generated by $x_1, \ldots, x_m$. Let $V = Kx - 1 \oplus \cdots \oplus Kx_m$ be a vector space. Let $V_0 = K$, $V_1 = K \oplus V$ and $V_{k+1} = V_k \oplus V_{k+1}$. If $V_i \subseteq V_{i+k}$ and $V_i \cdot V_j \subseteq V_{i+j}$, $A = \bigcup_{i=0}^{\infty} V_k$ then $\{V_k\}$ is the **standard (ascending) filtration** of A.

Definition 2 *Let $\{H_d := V_d M_0\}$ be an ascending filtration of M induced by M_0 a finite-dimensional K-vector space spanned by the generators of M. The* **Gel'fand–Kirillov dimension** *of M is*

$$\mathrm{GKdim}(M) = \limsup_{d \to \infty} (\log_d(\dim_K H_d)$$

By convention $\mathrm{GKdim}(\mathbf{Q}) = 0$.

Lemma 1 *Let A be a K-algebra and a domain. If the standard filtration on A is compatible with the PBW basis, then*

$$\mathrm{GKdim}(A) = \limsup_{d \to \infty} \log_d \binom{d+m}{m} = m.$$

There is an algorithm in Singular:Plural, due to [Gomez-Torrecillasetal] which computed GKdim, via left Gröbner bases.

Theorem 4 (Bernstein's inequality) *Let A be the n-th Weyl algebra over K, with* $\mathrm{GKdim}(K) = 0$ *and* $\mathrm{char}(K) = 0$, *then* $\mathrm{GKdim}(A) = 2n$. $\mathrm{GKdim}_K(M) \geq n$ *where M is a non-zero A module.*

1.1.4 Part III — Ore localization

Let A be a **commutative** Noetherian domain, and S a multiplicatively closed set in A: $0 \notin S$.

Definition 3 *The localisation of A w.r.t. S is a ring $A_S = S^{-1}A$ together with an injective homomorphism $\phi : A \to A_S$ such that*

1. *$\forall s \in S$ $\phi(s)$ is a unit in A_S;*
2. *$\forall f \in A_S$ $\exists a \in A, s \in S$ such that $f = \phi(s)^{-1}\phi(a)$.*

Examples are monoidal, rational and geometric localisations.

Suppose now that A is a **non-commutative** Noetherian domain, and S a multiplicatively closed set in A: $0 \notin S$. The **Ore condition** is

$$\forall s_1 \in S \forall r_1 \in A \exists s_2 \in S, r_2 \in E : r_1 s_2 = s_1 r + 2$$

In particular the Weyl algebra is an Ore set.

Lemma 2 (L–Schinderal2011) *For the shift algebra $\mathcal{M}(S) = \{f^n(x \pm z) | n, z \in \mathbf{N}_0\}$ is an Ore set in A.*

Lemma 3 (L–Schinderal2011) *For the quantum shift algebra $\mathcal{M}(S) = \{f^n(q^{\pm z}x) | n, z \in \mathbf{N}_0\}$ is an Ore set in A.*

We can induce Gröbner basis theory in the localisations. We need the **elimination property**: $1 \prec x^\alpha \prec \partial_i$ for all α, i.

$\mathrm{Sol}_A(M, \tilde{\mathcal{F}}) \equiv \mathrm{Sol}_{S^{-1}A}(S^{-1}M, \tilde{\mathcal{F}})$.. Also, if $\tilde{\mathcal{F}} \subset \mathcal{F}$, then $\mathrm{Sol}_A(M, \tilde{\mathcal{F}}) \subseteq \mathrm{Sol}_A(M, \mathcal{F})$.

t is hard to determine the G-K dimension of localised algebras. We do know $\mathrm{GKdim}(S^{-1}A) \geq \mathrm{GKdim}(A)$.

Concepts of annihilators and complete annihilators. We don't know how to compute complete annihilators: computational D-module theory. Needs the global Bernstein–Sato polynomial.

Example 1 *Let A_1 be the polynomial and B_1 the rational Weyl algebra. ... Consider the matrix*

$$M = \begin{bmatrix} \partial^2 - 1 & \partial + 1 \\ \partial^2 + 1 & \partial = x \end{bmatrix}$$

The algorithm returns

$$D = \begin{bmatrix} x^2\partial^2 + 2x\partial^2 + \partial^2 - 2x\partial - 2\partial - x^2 - 1 & 1 \\ 0 & 1 \end{bmatrix}$$

and U and V.

So the strategy from localisation is to

- use the information from the localised situation.
- Perform fraction-free computations.

1.1.5 Part IV: Purity

In the handout, but not lectured.

1.2 Upper bounds on real roots and lower bounds for the permanent — Koiran

Agenda:

- Upper bounds on the number of real roots for certain sparse polynomial systems;
- Depth reduction for arithmetic circuits.

The motivating question is "what is the arithmetic complexity of the permanent" — described [Valiant1979] as 'the arithmetic version of $P \stackrel{?}{=} NP$. Note that a permanent of size n can be represented by a determinant of size $2^n - 1$ [Grenet].

Conjecture 1 *If* $\mathrm{per}(a) = \det(B)$ *then* $\mathrm{size}(B)$ *cannot be polynomial in* $\mathrm{size}(A)$.

Under Valiant's model, we ask $VP_k \stackrel{?}{=} VNP_k$. $L(f)$ i the size f the smallest arithmetic circuit computing f.

- $(f_n) \in VP$ if the number of variables, $\det(f_n)$ and $L(f_n)$ are polynomials bounded. So $\det_n \in VP$, $X^{2^n} \notin VP$.
- $(f_n) \in VNP$ if $f_n(\overline{x}) = \sum_{\overline{y}} g_n(\overline{x}, \overline{y})$ for some $(g_n) \in VP$.

Outline:

1. Depth reduction for arithmetic circuits.
   - Reduction to depth $O(\log n)$ for arithmetic formulae
   - Reduction to depth $O(\log^2 n)$ for low-degree circuits
   - Reduction to depth 4 for low-degree circuits.
2. The real τ-conjecture.
3. Upper bound for the number of real roots:
   - Descartes: t monomials implies at most $t - 1$ positive roots.
   - Khovanskii: A system with t distinct exponent vectors has at most $(n+1)^t 2^{t(t-1)/2}$ non-degenerate roots in the upper orthant.
   - For certain sparse systems, the Wronskian determinant leads to better bounds.

   Example 2 *How many roots does* $fg = 1$ *have. Descartes implies* $O(t^2)$*, but this is* $y = f(x); yg(x) = 1$*, so the answer* might *be* $O(t)$.

1.2.1 Part 1

Definition 4 *A circuit is* weakly skew *if, for every multiplication gate $\alpha := \beta \times \gamma$, C_β and/or (at least one of) C_γ are otherwise independent of the rest of the circuits. A gate that is not in an independent sub-circuit is* reusable.

Arithmetic Trees and Skew Circuits[1] are both kinds of WSC (but the two are not comparable). Determinants are WSC, and every WSC can be written as a determinant (see lecture notes: the proof goes via Arithmetic Branching Programs), which gives lots of composability properties of determinants [KK08].

Note that if we convert a matrix into a graph G (see notes) the permanent is a cycle cover of G, and (up to signs) the determinant is the sum of the weights of cycle covers in G.

Theorem 5 ([KK08]) *A weakly skew circuit of size m has an equivalent skew circuit of size $2m$.*

Theorem 6 *Let G be a branching program of size m and depth δ. Then there is an equivalent circuit of depth $2\log\delta$, with $m^3\log\delta$ binary multiplication gates, and $m^2\log\delta$ addition gates of unbound fan-in.*

Theorem 7 ([VSBR83]) *Let C be a circuit of size s computing a polynomial $f(x_1,\ldots,x_n)$ of degree d. Then there is an equivalent circuit of size $O(d^6s^3)$ and depth $O(\log(ds)\log d+\log n)$.*

Showed proof of $VP \subseteq VNC^3$.

Theorem 8 ([AgrawalVinay2008]) *Let $P(x_1,\ldots,x_m)$ be a polynomial of degree $d=O(m)$. If there exists an arithmetic circuit of size $2^{o(d+d\log\frac{m}{d})}$ for P, then there is a depth 4 arithmetic circuit of size $2^{o(d+d\log\frac{m}{d})}$.*

Corollary 1 *If* per $\in VP$*, then it has depth 4 circuits of size $n^{O(\sqrt{n}\log n)}$.*

1.2.2 Part 2: the τ conjecture

Let $\tau(f)$ be the size of the smallest arithmetic circuit for $f \in \mathbf{Z}[x]$, with no constants allowed.

Conjecture 2 ([ShubSmale1995]) *There is a constant c such that f has at most $\tau(f)^c$ integer zeros (for a constant c).*

This is false for real roots — Chebyshev ($\tau(T_{2^n}) = O(n)$ by the duplication formula.). We have $c \geq 2$

Theorem 9 *This conjecture implies that $P_c \neq NP_c$.*

Theorem 10 ([Burgisser2007]) *This conjecture implies that there is no polynomial-size arithmetic circuits for the permanent.*

[1] One of β and γ is an input.

Conjecture 3 *Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{i,j}(X)$ where the $f_{i,j}$ are t-sparse. Then the number of real roots is polynomial in kmt.*

- $k = 1$ follows from Descartes
- It is enough to bound the number of integer roots.
- By expanding the products, we get $2kt^m - 1$ real roots.
- $k = 2$ is open even for $fg - 1$ — see introduction.

Theorem 11 ([AgrawalVinay2008]) *Any multilinear polynomial in n variables with an arithmetic circuit of size $2^{o(n)}$ also has a depth four $\Sigma|Pi\Sigma\Pi$ circuit of size $2^{o(n)}$.*

Sketch of Theorem 10. Use the Pochhammer–Wilkinson polynomials $PW_n := \prod_{i=1}^{n}(x - i)$ and the fact that is permanent is easy, PW_n has circuits of depth $(\log n)^{o(1)}$.

The limited power of powering

What if the number of distinct $f_{i,j}$ is very small, even constant? Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{i,j}}(X)$.

Theorem 12 ([GKKP10]) *If f is non-zero, it has at most $t^{O(m,2^k)}$ real roots.*

Note that Khovanskii-style bounds are exponential in k, m, t.

Theorem 13 ([KPT12]) *If f is non-zero, it has at most $t^{O(m,k^2)}$ real roots.*

1.2.3 Part 3: Wronskians

Linear dependence implies Wronskian is zero, but not the converse [Pea89a, Pea89b, B$\hat{0}$0a][2]. The converse *is* true for analytic functions [B$\hat{0}$0b].

Theorem 14 (Upper Bound Theorem) *Assume that the k Wronskians*

$$W(f_1), W(f_1, W_2), \ldots, W(f_1, f_2, \ldots, f_k)$$

have no zeros on I. Let $f = \sum a_i f_i$has at most $k - 1$ zeros on I, counted with multiplicity.

For $k = 2$ follows from Rolle's theorem.

Theorem 15 ([Bôcher1900]) *If $f_1, \ldots, f_k : I \to \mathbf{R}$ are analytic and $W(f_1, \ldots, f_k) \neq 0$ on I, then these functions are linearly independent.*

[2]For the history, see [EP11].

1.3 Algebraic Statistics — Sullivant

- Many statistical models are described by (semi-)algebraic constraints on a natural parameter space.
- Generators of the vanishing ideal can be useful for constructing algorithms or analysing properties of statistical models.

Example 3 *Hardy–Weinberg Equilibrium. If allele a occurs with probability* θ*, and alternative A* $1-\theta$*, then Hardy–Weinberg Equilibrium is* $P(aa) = \theta^2$*,* $P(aA) = 2\theta(1-\theta)$ *and* $P(AA) = (1-\theta)^2$*. The set of valid triples* $(P(aa), P(aA), P(AA))$ *is generated by* $p_{aa} + p_{aA} + p_{AA} - 1$*,* $p_{aA}^2 - 4p_{aa}p_{AA}$*, where the first is obvious and the second is interesting.*

1.3.1 Phylogenetics

Observing the DNA of living species, find the best tree that explains the origin. We assume the alignment problem is solved (non-trivial!). e.g. human/chimpanzee/gorilla. There are two parameters: possible tree shapes and mutation rates. We tend to assume independence of sites. Then summarise to the distribution of columns in the alignment.

Biologists tend to work with the exponentials of matrices, but this isn't necessary (or mathematically helpful). The random variables associated to the internal nodes are latent. Fixing a tree T and a model structure, we get $\phi^T : \theta \to \mathbf{R}^{4^n}$, where $\theta \subset \mathbf{R}^d$ is a parameter space of numerical parameters (transition matrices associated to each edge). For each $i_1, \ldots, i_n \in \{A, C, G, T\}$, $\phi^T_{i_1,\ldots,i_n}(\theta)$ gives the probability of the column in the alignment for θ.

Let $\mathbf{R}[p] := \mathbf{R}[p_{i_1,\ldots,i_n} : i_1, \ldots, i_n \in \{A, C, G, T\}]$.

Definition 5 *The* phylogenetic variety *is*

$$I_t := \langle \ldots \rangle.$$

A split of a tree is a partition of the leaves such that the induced trees do not intersect. Such a split induces a flattening of the probability tensor.

Proposition 1 *If* $A|B$ *is a valid split, then* $\operatorname{rank}(\operatorname{Flat}_{C|D}(P)) \leq 4$ *and the invariants in* I_t *are subdeterminants of* $\operatorname{Flat}_{A|B}(P)$*. Conversely, if* $C|D$ *is not valid, then generically* $\operatorname{rank}(\operatorname{Flat}_{C|D}(P)) > 4$*.*

Idea 1 ([Cavender–Felsenstein1987,Lake1987]) *Evaluate phylogenetically informative phylogenetic invariants at empirical distribution* $\hat{p}$ *to reconstruct phylogenetic trees.*

This was rubbished by [Huelsenbeck1995], but he only used linear invariants, approximating an $O(n^3)$ curved space by an $O(2^n)$ linear space.

1.3.2 Identifiability

Usual definition: the real question is "is the tree identifiable?". In fact, we settle for 'generic identifiability'.

Definition 6 *The tree parameter is* generically identifiable *if, for all n-leaf trees with $T \neq T'$.*

$$\dim(M_T \cap M_{T'}) < \min(\dim(M_T), \dim(M_{T'}))$$

Note that combinatorics says that the splits uniquely identify the tree.

Theorem 16 *The unrooted tree parameter of phylogenetic models is generically identifiable.*

However, the assumption the same parameters at every site, and this isn't really valid within a single gene. Furthermore, different whole genes might actually give different trees (a corollary of two-parent inheritance). Note that this can happen even if we use the same individuals for the genetic analysis. Let $M(T, r)$ denote the *same tree mixture model* with r classes of sites (previous case was $r = 1$). As $r \to \infty$ these models become unidentifiable (not surprising — too many parameters).

Theorem 17 ([RhodesSullivant2011]) *The unrooted tree and numerical parameters in a n-class, same tree phylogenetic mixture model on n-leaf trivalent trees are generically identifiable if $r < 4^{\lceil n/4 \rceil}$.*

In practice $r < 10$, so this theorem applies.

Can also look at group-based models. Then DFTs apply. These models are in fact toric varieties.

Theorem 18 (Draisma–Kuttler) *Let $T = T_1 \# T_2$ be obtained by joining two trees at a leaf. Suppose each ring is invariant under $\mathcal{G} + GL_r(\mathbf{C})^k$ acting on leaves.*

- $\mathbf{C}[p]/I_T \equiv (\mathbf{C}[p]/I_{T_1} \otimes_{\mathbf{C}} \mathbf{C}[p]/I_{T_2})^{\mathcal{G}}$
- $V_T = (V_{T_1} \times V_{T_2})//\mathcal{G}$ *(GiT Quotient).*

1.3.3 Generating Random Tables

Consider 2×3 tables with fixed rum and column sums. We can get from one to another by adding/subtracting one of $\begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{pmatrix}$ and two others. These form a lattice.

Definition 7

- *Let $A : \mathbf{Z}^n \to \mathbf{Z}^d$ be a linear transformation, $b \in \mathbf{Z}^d$.*
- $A^{-1}[b] = \{x \in \mathbf{N}^n : Ax = b\}$.

- $B \subset \ker_{\mathbf{Z}}(A)$.

Then let $A^{-1}[b]_{\mathcal{B}}$ *be the graph with vertices* $A^{-1}[b]$ *and edges* $u \leftrightarrow v$ *iff* $u - v \in \pm\mathcal{B}$

Problem 1 *Given* A *and* b*, find finite* $\mathcal{B} \subseteq \ker_{\mathbf{Z}} A$ *such that* $A^{-1}[b]_{\mathcal{B}}$ *is connected.*

Definition 8 *If* $\mathcal{B} \subseteq \ker_{\mathbf{Z}} A$ *is a set such that* $A^{-1}[b]_{\mathcal{B}}$ *is connected fro all* b*, then* $\mathcal{B}$ *is a* Markov basis *for* A*.*

Theorem 19 (Diaconis–Sturmfelds) *The set of moves* $\mathcal{B} \subseteq \ker_{\mathbf{Z}} A$ *is a Markov basis for* A *if and only if the set of binomials* $\{p^{b^+} - p^{b^-} : b \in \mathcal{B}\}$ *generates* I_A*.*

Hence the Hilbert Basis Theorem says that there is always a *finite* Markov basis.

$V(A)$ is a toric variety, which is what statisticians call a log-linear model. This variety is in fact generated by all the 2×2 minors of a generic matrix. See `www.4ti2.de`, which has a Macaulay2 interface, for computing Markov bases.

Theorem 20 *Every integer vector appears as part of a minimal Markov basis for* $3 \times k_2 \times k_3$ *tables. In fact,* k_2 *and* k_3 *are polynomial in the bit complexity of the vector.*

For $3 \times 4 \times 6$ tables, the minimal Markov basis has 355950 elements!

Given a move $\in \mathcal{B}$, we can associate a monomial $p^{m^+} - p^{m^-}$, and make these generate $I_{\mathcal{B}}$. Then u, v are in the same component iff $p^u - p^v \in I_{\mathcal{B}}$.

Theorem 21 ([EisenbudSturmfels1996]) *Every binomial ideal has a binomial primary decomposition (over an algebraically closed field).*[3]

See also Mesoprimary decompositions, which remove the last restriction.

Definition 9 (Graphical Models)

- *G is a graph with N vertices*
- *$d \in \mathbf{Z}^N$ with $d_i \geq 2$.*
- *We therefore get a set of margins of a $d_1 \times d_2 \times \cdots d_N$ array.*
- *let $\mathcal{C}G$ be the set of maximal cliques of G.*

Let

$$A_{G,d} : \mathbf{Z}^{d_1 \times d_2 \times \cdots d_N} \to \mathbf{Z}^k$$

be the linear computes the margins associated to all $C \in \mathcal{C}G$.

Suppose A, B, C partition $V(G)$ such that C separates A and B in G.

Theorem 22 (GeigerMeekSturmfels) *The separating moves are a Markov basis iff G is a* chordal graph*.*

Various theorems of the speaker indicate that appropriate laws on clique sums etc. work. Question: Is $I_{\mathcal{C}(G)}$ radical for all G, d? No for $K_{3,3}$ and $d = (2, 2, 2, 2, 2, 2)$. But this is not very constructive currently.

[3] In response to a question, the worst-case complexity is bad, since [MM82] applies directly: these are binomial ideals.

Chapter 2

23 July 2012

2.1 Solving Polynomial Systems over Finite Fields — Perret

NP-hard [GJ79]. But many applications (AES, GSM encryption, HFE-like systems, ECDLP [FaugereGaudryHuotRenault] etc.).

Given $f_1, \ldots, f_m \in \mathbf{F}_q[x_1, \ldots, x_n]$, find solutions in $\mathbf{F}_{q^m}$. Even over F_2 not clear. Best is [Bardetetal2012] $O(2^{0.841n})$.

Now consider $q > 2$ [BFP09]. Aim is to mix search and Gröbner bases: essentially specialising the last k variables. See Table 2.1, where we used F_5 for the Gröbner base computation.

Table 2.1: Results for Perret

System	k	Brute Force	Gröbner	Hybrid
OUV_{30}	10	2^{80}	2^{37}	2^{33}
OUV_{60}	20	2^{160}	$2^{??}$	$2^{59}(k=1)$
OUV_{60}	20	2^{160}	$2^{??}$	$2^{60}(k=1)$

But $k = 20$ is memory-feasible for OUV_{60}, whereas $k = 10$ is not.

Where should we trade-off? Experimentally, k is a constant, but in fact we see that k should be proportional to n, with $\beta_0 = \frac{1}{\nu_0^2}$, where ν_0 is a root of a given equation (depending on q, and ω). Seeing $2^{1.38\omega n}$ rather than pure Gröbner $2^{2\omega n}$.

In [FPPR12] apply to discrete logarithms.

2.2 The M4RIE library for dense linear algebra over small fields with even characteristic — Albrecht

2.2.1 Multiplication

- Linear algebra over $\mathbf{F}_{2^m}$ for $2 \leq m \leq 10$.
- We have two possible representations: a) matrices of polynomials and b) polynomials of matrices.

a) Identify polynomials (in the generator) with bit strings and hence with integers.

- Additions are cheap (XOR), but multiplications are much more expensive.

b) Or bitslice them. `mzd_slice_t`. Multiplications are a bit cheaper.

a) Multiply: precomputing all products of a vector gives $m2^e k$ operations. Or just compute the x^k·vector operations.

- We use 7 tables, filling the cache with these premultiplication tables.
- All this is the base case: still cubic, then leverage with Strassen–Winograd.

b) Use Karatsuba, since three matrix multiplications are cheaper than four!

There's a generalisation of this in LinBox for $\mathbf{F}_{p^e}$ for larger p.

2^8 results: Magma 2.15 was 104 seconds, GAP 84, SW-NJ (Strassen/precompute) 10.17 seconds, [Mon05] 27 (not seconds: apparently an expected efficiency ratio), bitslice 2.14.

2.2.2 Elimination

Definition 10 (PLE, [JPS11]) *Let A be a $m \times n$ matrix over a field K. A PLE decomposition of A is a triple of matrices P,L and E such that P is a $m \times m$ permutation matrix, L is a unit lower triangular matrix, and E is a $m \times n$ matrix in row-echelon form, and $A = PLE$.*

Use PLE decomposition. This needs efficient matrix products (done), efficient triangular solving (to be done) and a base case (to be done). Does 3 times better than LinBox (42 times better than Magma) as long as PLE fits: slower when switches to Gaussian elimination.

These asymptotically-fast algorithms are rank-sensitive. But there's also sparsity-sensitivity. SW-NJ is better when < 6 nonzeros/row.

2.3 Relaxed p-adic Hensel lifting for algebraic systems — Berthomieu

Work over p-adic integers, but valid for any $\mathfrak{p}$-adics. We have a choice for handling the fact that these are infinite objects.

Zealous double precision each time we run out.

Lazy Increase by 1, but represent a term as a flow of coefficients.

In particular, we only need the inverse of the Jacobian at precision 0.

A p-adic is recursive of order 1 if it is the solutions of $a = \Phi(a)$, where $\Phi(a)_N$ only depends on $a_1, \ldots, a_{N-1}$ for all $N \geq 1$. For example, quotient of two p-adics.

A *shifted algorithm* is where the shift (i.e. the fact that a_N only depends on $a_1, \ldots, a_{N-1}$) is made explicit. State Hensel's Lemma.

2.4 Hypergeometric functions, computational aspects — Beukers

Euler and Gauss studied

$$ {}_2F_1\left(\begin{array}{cc} \alpha & \beta \\ \gamma & \end{array} \middle| z \right) = \sum_{n=0}^{\infty} \frac{(\alpha)_m (\beta)_n}{n! (\gamma_n} z^n. $$

$$ {}_2F_1\left(\begin{array}{cc} 1/2 & 1/2 \\ 1 & \end{array} \middle| z \right) = \frac{2}{\pi} \int_0^1 \frac{dt}{\sqrt{(1-t^2)(1-zt^2)}}, $$

which doesn't have a closed form. We get higher hypergeometric functions

$$ {}_rF_{r-1}\left(\begin{array}{cccc} a_0 & a_1 & \ldots & a_r \\ b_0 & \ldots & b_{r-1} & \end{array} \middle| z \right) = \sum_{n=0}^{\infty} \ldots. $$

[BeukersHeckman1987] shows an ${}_8F_7$ which is algebraic of degree 483840. We can also get the Appell hypergeometric functions (confusingly F_i) of two variables (instead of z). These satisfy partial linear differential equations of order 2: shows $F_4(\alpha, \beta, \gamma, \gamma', x, y)$. [Lauricella1983] went further F_A etc., where there are m variables. When $m = 1$ they are all ${}_2F_1$.

2.4.1 A-hypergeometric approach

Consider $\mathbf{C}^4$ with coordinates v_i, $(\mathbf{C}^*)^3$ acting via

$$ (t_1, t_2, t_3), (v_1, v_2, v_3, v_4) \mapsto (t_1 v_1, t_2 v_2, t_3 v_3, (t_1 t_2 / t_3) v_4) \tag{2.1} $$

Let $\partial_1 \partial_2 \phi - \partial_3 \partial_4 \phi = 0$, and $\ldots$, then $\phi(1, 1, 1, z)$ is a Gauss hypergeometric. Homogeneity translates into Euler differential equations. This operator is homogeneous under the torus action (2.1).

We can write the exponents of the torus action (2.1) as 4 column vectors on $\mathbf{Z}^3$:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \tag{2.2}$$

Start with a finite subset $A \subseteq \mathbf{Z}^r$ and assume the $\mathbf{Z}$-span is $\mathbf{Z}^r$ and there is a linear form h such that $h(a_i) = 1$. Let $(Z_i) = A(v_i\partial_i)$ be the Euler operators. Let $square_{u-v} = \partial_u - \partial_v$ be the box operator. Then ${}_2F_1$ corresponds to the matrix above (2.2) with parameters $(-\alpha, -\beta, \gamma - 1)$.

Appell's F_1 is similar with $N = 6, r = 4$. Also Appell's F_4.

We can consider the system generated by $square_i$ and $Z_i - \alpha_i$ in $\mathbf{C}[v_1, \ldots, v_n, \partial_1, \ldots, \partial_n]$.

Theorem 23 (Kashiwara) *The rank ...*

There is a toric ideal generated by the box operators.

Theorem 24 (GKZ) *Let* $\mathrm{Vol}(A)$ *be the volume of the convex hull of* A*. Then* $\mathrm{rank}(H(A\alpha)) \geq \mathrm{Vol}(A)$*, with equality if* I_A *is Cohen-Macaulay.*

Consider the initial ideal, and replace the ∂_i by commuting ξ_i. The zero locus of this ideal, projected onto the x_i is the singular locus. For ${}_2F_1$ and matrix (2.2), then we get the solution $z = 1$ as the singular locus: correct. 0 and ∞ are always there. Appell's F_4 gives $1 - 2x - 2y + x^2 - 2xy + y^2$: a well-known singular locus. Remarkable (and proved) numeric properties of the coefficients, e.g. 13^{13}. Can also get formal series solutions. Let L be the lattice of relations between the a_i. Choose $\gamma_i \subset \mathbf{R}$ such that $\sum \gamma_i a_i = 0$. Then

$$\sum_{l \in L} \frac{v_1^{l_1+\gamma_1} \cdots}{\Gamma(l_1 + \gamma_1 + 1) \cdots}$$

is a formal solution. Since $\gamma_i = 0$ for $i > r$, there is a region of convergence. Similarly four solutions to Appell's F_4.

Challenge: find an algorithm in the non-resonant case ($>$ in Theorem 24).

2.5 An efficient implementation of the algorithm computing the Borel-fixed points of a Hilbert scheme — Lella

2.6 Effective de Rham Cohomology - The Hypersurface Case — Scheiblechner

This question is relevant to computing Betti numbers in singly-exponential time.

2.7 GNU TEXMACS: a scientific editing platform —van der Hoeven

A system for writing mathematical papers, and using CA systems. Looks like LaTeX, but is not based on it! Has style files for most CA-related outlets. Ongoing work on to/from LaTeX converters. Note that LaTeX is a programming language, not a document processor.

There is more semantics than LaTeX. Also highlights (using MathJax) items that did not parse properly. From the demo., matrix input looks distinctly less painful. Also claims that dynamic objects allow export of documents created in, say, Mathematica to those who don't have it?! "There is no such thing as source code" — can be saved in various formats, including XML.

2.8 POLY: a new polynomial data structure for MAPLE — Pearce

Amdahl's Law:

$$\text{speedup} \leq \frac{1}{S + (1-S)/N},$$

where S is the sequential fraction. And in fact this isn't limited to parallelism, applied to all optimisations. Maple has `degree`, `coeffs`, `subs` etc. in the kernel, and the rest in the interpreter.

Our new data structure represents sparse polynomials in graded lexicographic order. The exponent vector is coded and packed into one machine word (if possible?).

Q What about polynomial $+ e^x$?

A We don't handle this: it's a general Maple structure.

Q Overflow?

A Graded lex, so checking is only done once.

2.9 Parallel programming support in GAP — Behrends

HPC–GAP project. Demonstrated starting a shell in an independent thread. Shared regions have to be locked before you can access them (either RW or RO). However, `RunTask`, `WaitTask` and `WaitAnyTask` mean that one doesn't need to worry about the low-level regions. Also calls like `ShareObj`, `LockAndMigrateObj` etc., which he demonstrated inside a parallel matrix multiply. The Garbage Collector is parallelised for both the sequential and parallel case.

Base types are atomic list (resizable), atomic list (fixed), atomic record, atomic positional object etc. There are still issues with loading and saving

workspaces.. There is a challenge with memory barriers in macros, needed for ARM

Q Hardware threads?

A POSIX threads!

Q If you migrate an object, do you *have* to copy it?

A Choice between copying and "moving".

Q Which GC.

A Currently "stop the world", and not generational — on the list of things to improve.

2.10 `MONOMIALIDEAL.LIB` — Bermejo

A Singular Library. Computes irreducible and primary decompositions. Showed `QuotientMon` and `:` operations. `isprimeMon`, `isprimaryMon` and `isirreducibleMon`. `irreddecMon` (after [Vas98]). Also the slice algorithm, using the `sr`keyword: see [Rou09]. These have applications in more general settings (the demonstration failed), primary decomposition of general ideals seems to be assisted here: much faster (27 versus 102 seconds) than [GTZ88].

2.11 Sparse Polynomial Interpolation and Berlekamp/Massey Algorithms That Correct Outlier Errors in Input Values — Pernet

Berlekamp–Massey algorithm with errors (bounds on decoding capacity).

Dense Reed–Solomon codes/CRT codes; Number of evaluation points can be made adaptive on error impact and degree [KPR$^+$10].

Sparse Our work, based on [BOT88].

$$\forall j \geq 0 \qquad a_{j+t} = \sum_{i=0}^{t-1} c_i a_{j+i}.$$

Suppose a_i is generated by $\Lambda(z)$ of degree t, but we have $(b_i) = (a_i) + \epsilon$, where ϵ has E errors. How to recover? His (shown) example shows that it may not be unique ($t = 2$, $E = 1$, $n = 11$). More generally $n - 3t(2E + 1) - 1$ is still ambiguous, but can prove this is maximal. Shows `SequenceCleanUp`.

Theorem 25 *If $n \geq t(2E + 1)$ then a deceptive segment will necessarily be exposed.*

Also works if we are looking for polynomials of degree *at most* t.

[BOT88]: recover a r-sparse polynomial f given a black box computing evaluations of it. $a_i = f(p^i)$. Needs $2t$ input values, or $2T(2E+1)$ with $e \leq E$ errors and $t \leq T$.

Reed–Solomon codes can be regarded as evaluation codes. Vandermonde can essentially be run either way.

There's more, based on [GLL09], for sparse interpolation with noise *and* outliers. In the dense case Padé approximants work well with Reed–Solomon codes, but the sparse case is open.

Q What happens if you have error probabilities, rather than a fixed number of errors?

A Not considered.

2.12 Parallel sparse polynomial multiplication on modern hardware architectures — Biscani

Perturbative methods in celestial mechanics. Laurent, Puiseux, Fourier series. High sparsity and large number of terms (10^5 or 10^6). Truncated arithmetic, series expansions, differential operations, substitution, evaluation etc.

Tend to use classical methods because of the high sparsity. Main challenge is high memory bandwidth requirements ($O(n^2)$). Memory hierarchy plus problems of parallel machines. Use hashing techniques, and cache-friendly hash table implementation. Homomorphic hashing via the Kronecker transform. Simple and effective parallelisation. Implementation in C++.

Separate chaining has unpredictable memory access, open addressing hash suffers from clustering, and there are problems of concurrency. So we have a cache-friendly one: inline cache heads, separate chaining with 2^n sizes, first element of each bucket is stored within the array. This minimises heap allocations, and memory access becomes more predictable. The Kronecker mapping preserves addition and subtraction. Works for signed integers (Laurent polynomials etc.). We pack vectors of exponents as `long` or `long long`. Use primes for the ranges, then cast the KS signed integer and reduce modulo 2^n = table size. Note that hash of sum = sum of hashes. After sorting, we are writing into consecutive addresses.

We use the occurrence of the first exponent collision to infer the size of the result, by Ramanujan's Q-function $\approx \frac{2}{\pi}Q^2$. Randomise the term order of the operands, multiply together and repeat (?). Divide (sorted) operands into blocks, and assign to threads. A thread only multiplies blocks if no-one else is storing into the target block.

2.12.1 Benchmarks

Dense Fateman — 10× faster than Maple (SDMP=Monagan/Pearce).

Sparse Monagan–Pearce. 4× faster than Maple (SDMP).

Parallelisation shows 6× on 8 processors for dense, and 4× for sparse.

Q–Pearce Any other operations besides multiplication?

A I don't really need division, and multiplication is my real bottleneck.

2.13 On the complexity of multivariate blockwise polynomial multiplication — Lecerf

An unsolved problem for $R[x_1, \ldots, x_n]$. Many problems (such as division -JvdH) come down to this.

Dense Here it is solved as $\tilde{O}(n)$ by Kronecker and [SS71].

Sparse Well-tuned naïve algorithm. Note that the naïve algorithm is $\tilde{O}(\#output)$ in the absence of combinations. Karatsuba is not suitable in general. Blockwise methods have been suggested [vdH02]. Evaluation/Interpolation is good *when the support R is known in advance*.

Functional SLPs or black boxes. Important, but not treated here.

Write polynomials "blockwise", use polynomials whose coefficients are polynomials in the *same* variables, but where the outer exponents grow in blocks.

Theorem 26 *Algorithm* `BockMultiply` $N(b)s_{\overline{P}}s_{\overline{Q}} + \cdots$ *operations.*

N assume that R supports evaluation–interpolation schemes with $N(d) = 2d-1$ and $E(d) = O(d \log \nu(d+1))$. Then $O(|S_{n,d}|^{1.5337})$ operations, choosing blocks of size 1, 4 or $\lceil d/n \rceil$, depending on d/n.

The bottom line is that there is no one best algorithm.

2.14 ISSAC Business Meeting

Chair Franz Winkler.

2012 Sponsoring 1512 euros. We did a weaker affiliation to ACM, which reduced the guarantor's share from 16% to 10% of the budget (42000 euros): INRIA was the guarantor, and supplied more administrative support. The proceedings were printed locally (still in ACM Digital Library), and this saved 18 euros/copy[1]. In the future we should increase sponsorship, and

[1] But more ACM sponsorship would have made the proceedings cheaper.

try to help the few people who have financial issues with attending. Maple is a recurrent sponsor (1640 euros). There was an issue with referees who demand benchmarks on proprietary systems. EK reported that the Simon Foundation had purchased Magma for US academics. The Proceedings are electronically given to participants on USB, and the local printer charged 20 euros each. Fees are about the same as 2010, and historically low. 175 participants, and 80 for the tutorials. Current plan is 2000 euro surplus.

2013 Gene Cooperman spoke. This will be at NorthEastern University (Boston, Mass.). Northeastern has 25 Faculty in CS and 100 in Maths. June 26–29, which is the inter-summer break. Hotels are expensive, and dorms should be $85/night. Aims or a registration fee of $350 based on 2007.

CICM 2013 JHD announced as week of 8 July, in Bath.

2014 The only bid was from Kobe[2] University (Japan). Bid supported by JSSAC (397 members) and annual meetings. Kansia International Airport, and train/limousine (85 minutes, $18). also Osaka and Kobe (domestic only). Same location as CASC 2009 and ICMS 2010. Date July 21–25. This will be arranged as a satellite of ICM 2014 (Seoul, 13–21 August). Temperature 25–30C. Kobe University has exchange agreements with RISC and Maplesoft. Exchange rate is currently high (100/euro versus 170 for 2008).

[2]SW of Tokyo, 1.5M inhabitants.

Chapter 3

24 July 2012

3.1 On the Complexity of Solving a Bivariate Polynomial System — Emeliyanenko

Assume $f, g \in \mathbf{Z}[x, y]$, total degree $\leq n$, length $\leq L$. Resultants are complicated, and $f(z_i, y), g(z_i, y)$ have irrational coefficients. Note [Rouillier], [CGL09], [DET07]. [BES11]. Resultant and gcds (over $\mathbf{Z}$) are the only symbolic methods, outsourced to GPUs.

1. Compute $R(x) = \mathrm{res}_x(f, g)$ and $\tilde{R}(y)$. $\tilde{O}(n^5 L)$.
2. Isolate the roots $\tilde{O}(n^8 + n^7 L)$.
3. Refine the regions $\tilde{O}(n^8 + n^7 L)$.

4.1

4.2

5.1 Take a lower bound L_i for $|R|_{\partial\Delta_i}$: $\approx 2^{-\tilde{O})n^4+n^3 L)}$ — $\tilde{O}(n^8 + n^7 L)$.

5.2 Amortisation leads to $\prod_i L_i = 2^{-\tilde{O}(n^4+n^3 L)}$

Theorem 27

$$\sum_i -\mathrm{mult}(\ldots) \log L_i \leq \tilde{O}(n^4 + n^3 L).$$

8 Verification.

8.1 Use interval arithmetic with precision ρ

8.2 Apply inclusion to test that Δ contains a root

8.3 If not, double the precision.

Worst case complexity $\tilde{O}(n^8 + n^7 L)$, which in theory is two/four orders better than [DET07]. In practice this is clearly better. This produces best known complexity for topology of algebraic curves.

It should generalise to higher dimensions as well.

Q Worst-case achievable?

A Don't know — we normally treat resultants as arbitrary polynomials, e.g. a Mignotte polynomial, but is this really possible?

3.2 Critical Points and Gröbner Bases: the Unmixed Case — Spaenlehauer

Critical points are when $F = \{f_1, \ldots, f_p\} = 0$ and the maximal minors of J (Jacobian matrix less the $\frac{\partial}{\partial x_1}$ column) are zero. If F is a regular reduced sequence and $\mathrm{Var}(F)$ is smooth then these are the critical points of the restriction to $\mathrm{Var}(F)$ of projection onto x_1. Applications to sampling points, connectivity queries, quantifier elimination. Software RAGlib based on Gröbner bases.

Assume "unmixed", i.e. $\deg(f_i) = D \forall i$. Generically, there are $DEG := \binom{n-1}{p-1} D^p (D-1)^{n-p}$ critical points. Experimentally, Gröbner bases do very well, but there's no theory. We focus on 0-dimensional case. We first compute the Gröbner bases with a grevlex ordering, giving row echelon forms of the Macaulay matrices up to degree d_{reg}. Then convert this to lex order. $O(\binom{n+d_{\mathrm{reg}}}{d_{\mathrm{reg}}}^{\omega} \ldots$ for the first phase.

What about bounds for d_{reg}? If the f_i are generic and homogeneous, then $d_{\mathrm{reg}} = D(p-1) + (D-2)n + 2$. When generic, $d_{\mathrm{reg}} \leq D(p-1) + (D-2)n + 2$. We use the Hilbert Series $\mathrm{HS}(I)$, which is generated by the rank defects of the successive Macaulay matrices. Start with the $p \times (n-1)$ matrix $U = (u_{i,j})$ with $1 \leq i \leq p; 2 \leq j \leq n$ where the $u_{i,j}$ are new indeterminates. Since the $u_{i,j} - \frac{\partial f_i}{\partial x_j}$ are not homogeneous, we weight $D-1$ the variables $u_{i,j}$ to get a quasi-homogeneous ideal. Note that $K[U]/\mathcal{U}$ is a Cohen-Macaulay domain. Algebraic Sard's Theorem will get us that $\dim(I) = 0$ under genericity assumptions.

d_{reg} is $\deg(\mathrm{HS}_I(t))_1$. $\deg(I) = \mathrm{HS}_I(1) = \binom{n-1}{p-1} D^p (D-1)^{n-p}$. In the non-homogeneous case, we consider the homogeneous part of highest degree and get the same bounds.

If $d = 2$, $O(n^{2p\omega})$, which is polynomial in n. If $D > 2$, the complexity is $O(\frac{1}{\sqrt{n}} 2^{(D-1)n\omega h_2(1/(D-1))}) = O((D-1)^{3.57n})$, which is no longer polynomial in n.

Experimentally $10^{-8} DEG^{2.48}$, when modulo 65521. When $D = 2$, $p = 4$ we see $n^{10.55}$ (polynomial, as expected). One example with $\deg(I) = 18240$ took three hours (and broke the nice straight line: he blamed cache effects).

Q What happens if we remove genericity?

A It's OK in dimension zero.

3.3 Solving Polynomial Systems over Semi-algebraic Sets Represented by Cylindrical Algebraic Formulas — Strzeboński

So we need to allow algebraic functions in the input. Definition of a semi-algebraic

$$S(x_1, \ldots, x_n) = \bigvee \bigwedge \cdots$$

Quantified systems and Tarski's theorem. A Cell is a slice or a sector. "Algebraic function" means a specific root of a polynomial. Example: $-5(y-3)^3+7x^2(y-3) < 1 \wedge x^2 + 2(y-4)^2 < 21$.

These CAF (Cylindrical Algebraic Formulae) representations let us decide if A is empty, find volume, integrate over etc. We could like to compute set-theoretic operations over these [Str10]. More formally

Input A system S, and CAF F and a sequence of quantifiers $Q_1, \ldots, Q_k$

Output A cylindrical G equivalent to $Q_1, \ldots, Q_k(F \wedge S)$

***** Only compute S within the truth of F.

Problem of intersection of an oval with a rounded astroid was insoluble in 2 hours, now takes seconds. Another example of intersecting a ball and a rose (>> 24 hours. Ball is trivial. Rose is 1000 seconds. "Rose inside Ball" is 2.5 seconds.

- projection phase (McCallum's or Hong). $ProjMC(P)$ is coefficients, discriminants and resultants. $ProjMC(P,Q) = ProjMC(P) \cup \{\text{res}_{x_k}(f,g) : f \in P, g \in Q \setminus P\}$. Similar saving for Hong.
- Lifting phase. Only lift those cells that are valid for F, i.e. in $\Pi_k(F)$. When lifting $C \in \Pi_k(F)$, only use relevant projection polynomials.

  While we cannot give an *a priori* criterion for deciding whether the incremental method will be faster than the direct method, it is still useful to have an alternative method to try in cases that are hard for the direct method.

Q Could this be incorporated into general CAD.

A Good question?

Q Automatic methods?

A Note that switching the rose and the ball is not a good idea!

Q How sensitive is the computation to the input, e.g. making the ball larger?

A Very: if I made the ball large enough, I'd have the whole rose!

Q Is the CAD simpler, or are you just saving time by not computing?

A The latter.

3.4 Asymptotic spectrum and matrix multiplication — Strassen

The cost of multiplying $m \times m$ matrices is $O(m^{\omega+o(1)})$ and ω controls most of linear algebra, in at least one sense. The crucial problem in the other direction.

Theorem 28 ([BS83]) *If C is the (straight-line) cost of computing, then*

$$C(f, \partial f/\partial x_1, \ldots, \partial f/\partial x_n) \leq 4C(f). \tag{3.1}$$

Note that this is independent of n!

The derivatives of the determinant of a matrix are the minors, and from the minors, we can compute the inverse, and inverse implies multiplication, hence everything is all inter-related.

$$b : U \times V \to W \text{ assumed bilinear (fundamental) and concise (technical).} \tag{3.2}$$

- Diagonal map is vector product
- Maschke: group algebras are direct sums of matrix multiplication

$$\langle 2m, 2m2m \rangle \approx \langle 2, 2, 2 \rangle \otimes \cdots$$

The notion of restriction $a \leq b$:

$$\exists \alpha, \beta, \gamma \text{ linear } a(u, v) = \gamma b(\alpha u, \beta v)$$

This is a pre-order. Define the *rank* as $R(b) := \min\{r : b \leq \langle r \rangle\}$.

$$R(\langle m, m, m \rangle) = m^{\omega+o(1)}.$$

History illustrated by analogy with bicycles.

- Strassen $\omega = 2.81$
- Pan, Bini *et al.* $\omega < 2.78$
- $\omega < 2.55$ Schönhage.
- $\omega < 2.50$ CW
- $\omega < 2.48$ "laser method".
- $\omega < 2.38$ CW (laser+diagonal)
- Also CUKS: techniques from loops etc., which now equal CW.

History of the wren and the eagle.

- Williams $\omega < 2.373$.

Theorem 29 (Schönhage's τ-theorem) $\otimes_i \langle m_i, m_i, m_i\rangle <_{\asymp} \langle r\rangle \Rightarrow \sum_i m_i^{\omega} \leq r$.

The isomorphism classes of bilinear maps for a commutative semiring, and $\leq$ is a partial order compatible with addition and multiplication.

Theorem 30 *That any semiring $\mathcal{S}$ of bilinear maps, then there are*

- *a compact space Δ*
- *A homomorphism $\phi : S \to C^+(\Delta)$*

such that $\phi(S)$ separates points and $a <_{\asymp} b \Leftrightarrow \phi(a) \leq \phi(b)$.

$$\max \phi(b) \quad = \quad \min \ldots \tag{3.3}$$

There is a logarithmic embedding $\Delta(b_1, \ldots, b_q) \subset \mathbf{R}^q$, with $\phi(b_i) = 2^{\tau_i}$. $\Delta(\langle 2,2,2\rangle) \subset \mathbf{R}$.

$\omega = \max \Delta_m \in \Delta_m$.

$$\begin{array}{rcl} \otimes_i \langle m_i, m_i, m_i\rangle & <_{\asymp} & \otimes_j \langle n_j, n_j, n_j\rangle \\ \to \sum_j n_j^{\tau} - \sum_i m_i^{\tau} & \geq & 0 \text{ on } \Delta_m \\ \to \sum_j n_j^{\omega} - \sum_i m_i^{\omega} & \geq & 0 \end{array}$$

Theorem 31 (CohnUmans) $\langle m, m, m\rangle \leq_{CU} \mathbf{C}[G]$ *iff there are S_i such that*

CKSU may get non=-traditional spectral differences.

His example (not necessarily real)

$$(2^{4\tau} + 2^8) - (6 \cdot 2^{3\tau} + 5 \cdot 2^{2\tau} + 2^{\tau} + 1) > 0$$

This would imply $\omega < 2.323$.

3.4.1 General matrix multiplication

i.e. not necessarily square. Note that we can't do as well as we can for square matrices ("the umbrella lies above the straight plane").

Restrict our discussion to oblique maps. There exist bases such that the support is an anti-chain. Let Θ be the standard 2-simplex. Let $\sigma_b : \Theta \to \mathbf{R}$

$$\sigma_b(\theta) := \max\{\sum_{i=1}^{3} \theta_i H(P_i) : \ P \text{ a probability on } supp(b)\}.$$

Then

$$\min \Delta \subset \sigma(\Theta) subset \Delta$$

by CW.

Two research problems.

1. Prove $\omega < 3$ by a non-traditional restriction.
2. Prove that $\Delta(b)$ is an *interval* for any tight b.

Q Does this speed up the eigenvalue problem?

A Not that I know of: spectrum is meant in a very different sense.

Q Can you say something about Boolean Matrix multiplication?

A It's very different!

3.5 Univariate real root isolation in multiple extension fields — Tsigaridas

Dedicated to Werner Krandick.

What is the Boolean complexity of the isolating the real roots of

$$B_\alpha = \sum_i b_i(\alpha_1, \ldots, \alpha_l)x^i.$$

- Can norm out, but get a much larger polynomial with spurious roots [EmirisTsigaridas2007] with univariates and continued fractions. $\log|\gamma|$ bounds; complexity $\tilde{O}_B(N^{4l+4})$.
- Direct — Sturm–Habicht. Need signs in $\alpha - i$, so recurse. No known bounds, but we have $\tilde{O}_B(m^{2l^2}b^4(\sigma^2 + \cdots))$.
- Precision-based approximations. Use fast approximations [KS11]. Construct a system for the algebraic numbers *and* a given coefficients. Needs a better gap theorem [EGT10]. $\tilde{O}_B(N^{2l+4})$.

Generalised Laguerre: can do $l = 3$, $n = 50$, $m = 10$ in 4 seconds. Generalised Wilkinson polynomials $\prod_k(y - k\lambda$ where λ is a sum of roots: 4/50/10 in 78 seconds.

We get [ST11] for the case $l = 1$.

But have no examples of upper (worst-case) bounds. This is related to the question of whether the Euclidean TP is NP-complete, since we can write this as a polynomial problem! [GareyJohnson1974] $2^{-n^2 2^\tau}$.

We should look at towers of extensions

Q-JHD Is your generalised Wilkinson as nasty as the original?

A Yes — it's really only the original scaled by λ.

3.6 When Newton meets Descartes: A Simple and Fast Algorithm to Isolate the Real Roots of a Polynomial — Sagraloff

$m \leq v$ and $m \equiv v \pmod 2$, where $v = \mathrm{Var}(f, I)$. If $I_1, I_2 \subset I$ disjoint then $\mathrm{Var}(f, I) \geq \mathrm{Var}(f, I_1) + \mathrm{Var}(f, I_2)$. This is a generalisation of results by Obreshkoff.

Descartes Illustrated with initially $v = 9$ or five actual roots.. Also a tree which can never be wider than initial v. Distance between roots is $2^{-\tilde{O}(nL)}$, but this can't be true for them all, and total tree size is $\tilde{O}(nL)$. Total cost $\tilde{O}(n^4L^2)$. We don't actually need full precision: certified nL precision will work — see Maple's solve. [Sag10].


All we have is bisection.

* Say that a node is a milestone if it is starting point, or if "something happens" —- a split or var decreases. There are $\leq v$ milestones.

Newton Similar to [Abb06]. Reduces the length of the chains to $O(\log(nL))$. Note that this reduction is even better for sparse polynomials. Compute the Newton approximations, decide which small interval they are in, and check that this is right. If not, but we've split the roots, use each half. Get $\tilde{O}(n^3L)$ — matches Schönhage, but not Pan's improvement. Working with Rouillier to improve Rs. Would also like to work with the bitstream model.

3.7 Solving Polynomial Systems Globally Invariant Under an Action of the Symmetric Group and Application to the Equilibria of N vortices in the Plane — Svartz

Showed a 2003 hurricane satellite image. Also vortices in a rotating superfluid. A particular case of N-body problem. Naïvely can do $N = 4$, 180 seconds and ideal of degree 96. With symmetries, we can do $N = 4$ by hand, and up to $N = 7$.

$$\forall i \quad \frac{\partial^2 z_1}{\partial t^2 \sum_{j \neq i} m_j U'(|z_i - z_j|^2)(|z_i - z_j)}.$$

Q Were all the solutions real?

A No, but quite a few were.

3.8 Algorithms for the universal decomposition algebra — Lebreton

Let K have characteristic 0 or sufficiently large. Fix $f = X^n + \sum_{i=1}^{n} (-1)^i f_i X^{n-i}$ be separable of degree n. Roots $alalpha_i$. Universal Decomposition Algebra $A = k[X_1, \ldots, X_n]/I$. Degree $\delta = n!$. Absolute Lagrange resultant:

$$L_P(T) = \prod_{\sigma \in S_n // StabP} (T - P(\alpha_{\sigma(1)}, , \ldots, \alpha_{\sigma(n)}) \in k[T].$$

Triangular representation: $A_1 = k[X_1]/(C_1), \ldots, A_j]k = X_1, \ldots, X_j]/(C_1, \ldots, c_j)$, ... Alternative is a univariate representations. Cost $\tilde{O}(\delta^3)$ by FGLM, or $\tilde{O}(\delta^2)$ by geometric representation. In the multivariate representation, there is quasi-linear multiplication [BCvHP11], but not implemented and has a significant constant; division is worse. But in the univariate case both are efficient.

Theorem 32 *Conversion in quasi-linear time.*

- Newton sums method.

  Define $f \oplus g$ bethe polynomial whose rots are the sums o the roots of f and g, and $\otimes$.

  Theorem 33 *Let $\Lambda_j = \lambda_1 X_1 + \cdots$, then can compute ...*

  Lemma 4 $S_i(T) = -\left(\frac{\partial Chi_\Lambda}{\partial \lambda_i}\right) / \left(\frac{\partial Chi_\Lambda}{\partial T}\right)$.

  In Magma, and 100secs versus 6 hours with FGLM. But sill not quite optimal.

- Resultant method. Regard

  $$Resultant_{X_{i+1}} : A_i[T][X_{i+1}] \times A_i[T][X_{i+1}] \to A_i[T].$$

  $\tilde{O}(n^?)$ [Shoup1994].

Theoretically we have the first quasi-linear change of representations, hence characteristic polynomial and division. Practically, we have working Magma code, and better timings.

Q–J-CF : the weak point with FGLM is ...

A Where can we get this code?

Q–EK $\delta = n!$, but you are claiming quasi-linear!

A Quasi-linear in δ, which is the size of the output.

3.9 Symbolic computation for ordinary boundary problem in MAPLE — Korporal

New version of `IntDiffOp_ package. Allows 'complicated' boundary conditons such as integral co` which has the same syntax as, and is based on, Maple's `dsolve`.

- Simple ODE with two boundary conditions.
- Green's operator.
- `u(0)+2*(D(U))(0)=0` produces a solution in terms of non-elementary integrals. `dsolve` can' do this.
- finding specific RHS.
- Interfaces with `plot3D`.
- A fourth-order example.
- Can give a fundamental system as an additional argument, demonstrates by entering complex exponentials, rather than `dsolve`'s sin/cos.
- Singular boundary conditions, which are not always uniquely soluble.
- Composition of boundary conditions due to composition of Green's operators. `bdfactor`, again based on `dfactor`. Returns a list of solutions.

`IntDiffOperators` provides overloaded operators, notably `d &* f(x)` for differentiation and `a` for integration.

3.10 ISOLDE — A MAPLE package for linear functional equations — Pfluegel

Acknowledgements and ovation to Evelyne Tournier.

1990s Started during his Grenoble PhD. DESIR/ DESIR II were precursors.

2004 Sourceforge

2011 Maple modules. Joined by Flavia Stan.

Grail formal reduction, for differential, difference and q-difference systems.

K a field of characteristic 0. $F = K((X))$ or $K(X)$. ϕ a K-automorphism of F. δ a pseudo-derivation w.r.t. ϕ, with the usual three interpretations.

Sets up a matrix `A`, then `B:=mat_convert(A,x,0)` returns `A1`, which Isolde, knows is a key for `A`. Functions such as `Mat_Eval` and `Mat_Description` will find out information. Can use `linalg[charpoly]` on these to produce characteristic polynomials.

The new functionality is to make the difference case work as well as the differential one. "super reduction" is now Moser reduction. `http://isolde.sourceforge.net`.

3.11 The DECODING library for list decoding — Quintin

Decoding over fields is well-understood — rings less so. [Sudam1997] can decode $n-\sqrt{2nk}$ errors, in polynomial time. Improved [GuruswamiSudan1998], but still only returns a list of possible messages. We suppose $((f(x_i))$ was transmitted and $(y+i)$ received. Find a "curve"passing through (x_i, y_i) with appropriate multiplicity.

We let the user choose the alphabet, and the underlying fast arithmetic, e.g. `gmp`, `NTL` etc.

3.12 Computing Puiseux Series for Algebraic Surfaces — Verschelde

Given a sparse polynomial system: $f(x) = \sum_{a \in A} c_a x^a$. Cyclic n.

Lemma 5 (Backelin) *If $m^2|n$, then cyclic n-th roots has solutions of dimension $m-1$.*

The sparsity structure is modeled by its Newton polytope., the convex hull of A.

Definition 11 *A* tropism *consists of he leading powers $(v_0, \ldots, v_{n-1})$ of a Puiseux series of the curve.*

[Ber75] is critical. Also `cddlib` which enables us to enumerate facets. `Gfan`, via `Sage`, computes Gröbner fans, and uses `cddlib`.

Consider binomial systems. Represented by $A \in \mathbf{Z}^{N \times n}$ and a coefficient vector in $(\mathbf{C}^*)^N$. Solution sets are related to toric varieties. Cyclic-4 has binomial solutions. a pre-tropism. For cyclic 9, we found two-dimensional solutions. 4840 normals, 276 pretropisms, 17 generators. cyclic12 was 907923/38229/290 and 148 hours.

1. Select d linearly independent generators
2. If $\in_{v_0} (\ldots \in_{v_{d-1}} (\ldots)$
3.
4.

The pretropisms are small and sparse, and we solve these numerically. Have a tropical interpretation of Backelin's Lemma. Can write down an explicit structure for cyclicm^2 (but it's not necessarily complete). This is a proof of concept for a polyhedral solution to general dim>0 problems.

3.13 A Root Isolation Algorithm for Sparse Univariate Polynomials — Alonso

Developing idea from [GVLM98] on *virtual roots* of univariate polynomials. [Galligo2011]. [CLLLR05]. Concept of an F-derivative. $f(x) := \sum_{i=0}^{d} a_i x^{r_i}$. Let $g_d = f/x^{r_0}$. Can differentiate this and proceed: $g_{d-1} = g_d'/x^{r_1}$ etc. Then we want to look at V_f, the sign variations of the sequence of the (g_i). Near a root of multiplicity k of f, which is not a root of another one of these, the multiplicity changes by k as we pass through this. Hence:

Theorem 34 (Generalised Budan-Fourier Theorem) *The number of real roots of f in $]a, b[$, counted with multiplicities, $\leq V_f(a) - V_f(b)$, the difference being even.*

The F-Budan table is the union of $d+1$ infinite rectangles (rows of height 1): $L_i := \mathbf{R}_+ \times [i - 1/2, i + 1/2]$. These are coloured grey/black, and the connected components of the union of the closures of the gray (black) rectangles. Apparently, if two components stop at the same "x" coordinate we have multiple roots.

Assume f and every F-derivative has only simple roots, and these are pairwise disjoint: an (FP)-polynomial. Then can produce a table (truncated Budan-Fourier). (sign difference) − (root count at ends) = 2× number of virtual roots, which are locations of the BF table. We get a case analysis.

A I' isolates one root

B I' such that $BF(f, I') = 2$

E ...

Example of a Mignotte-like polynomial $x^{100} - 625x^4 + 1500x^3 + \cdots$. Shows it has three real roots and a "virtual root" (pair?).

3.14 Near Optimal Tree Size Bounds on a Simple Real Root Isolation Algorithm — Sharma

Model of computation is interval arithmetic. Looking for common roots in $\mathbf{R}^2$.

- Quadtree subdivision of B_0
- Apply exclusion predicates on f and g
- Inclusion predicate (contains a unique common root)
- sub-divide and recurse.

What is the size of the sub-division tree? [Kearfott1987] assumes all 1/(root sep). This doesn't happen in practice. There are many other contexts in which we do subdivision as well.

First look at 1-D case. Exclusion is that f is demonstrably big, inclusion is the same for f'. We would like $O(d(L + \log d))$.

[Yakoubson2005] $O(d^4(L + \log d))$.

[BurrKrahmerYap2009] $O(d^3(L + \log d))$.

[SY11] $O(d(L + \log d)(\ldots)$.

us $O(d(L + r))$, where r is the number of real roots in the interval.

The idea from [BurrKrahmerYap2009] is *charging function* $G : \mathbf{R} \to \mathbf{R}_{\geq 0}$ such that $\#P(I) \leq \int_I G$. Also need a stopping function [BK12].

Note that in the worst case we could have a d^2 term — would like to reduce this to $d \log d$.

3.15 Maple Demo

Plots Rubber banding, zooming tool etc. More control over colours, user-defined palette. "Smart view" handles discontinuities in the plottand better. `ColorTools` package. Snippets palette.

"Math Apps" — essentially demos. Accessible from the tools menu. Example Central Limit Theorem, and a slider controlling number of samples.

There's a Maple Player for Ipad.

`CodeTools:=Usage` seems a useful tool. Shows CPU time, therefore cpu/real=2 implies dual core usage etc.

Fast polynomial arithmetic, including good divisibility test.

Better ($\times 50$) factorisation for polynomials with algebraic entries. This spills over into radical ideals, prime decomposition etc.

Real solutions of polynomial systems, via `RegularChains`. New `SuggestVariableOrder`. Examples in stability analysis, identity verification.

More differential equations, including unknown functions.

Programming language has automatic type conversion: declare argument as `a::~Vector` — please convert the argument to a vector.

Objects (an extension of Modules).

Chapter 4

25 July 2012

Proposition 2

4.1 Border basis representation of general quotient algebra — Trebuchet

Quotient Algebra → Resultants **R**H-Bases → Janet Bases.

Aim: Construct $AsubsetR$ with a basis B and a project $\pi : R \to A$ such that the sequence is exact. Properties of border bases.

+ Extend all previous normal form techniques
+ Stable w.r.t numeical uncertainty
+ control coefficient growth/fill-iin
+ efficient
- Need new definition of bases
- Hard to definite on positive dumension bases.

B is the set of monomials connected to 1. $B^+[B \cup x_1B \cup x_2B \cup \cdots x_nB.$, with $\partial B = B^+ \setminus B$.

Theorem 35 *Let $D \geq 2$ let B be asubset connected to 1, and let $\pi \ldots$*

If I is zero-dimensional, then A is a f.d. K-vector space, and we get an explicit description of *all* the normal forms of ∂B. Define Castelnuoove–Mumford regularity.

Theorem 36 *1. $\nu_{i+1} = \nu_i^{(i,1)} - \nu_s$ [Goztmann regularity]*

Goztmann ??

We have an algorithm analogous to Gröbner bases using C-polynomials rather than S-polynomials. Cyclic–5 has degree 70, took 0.16 second and 20MB. We can how do positive dimension, so dropping one equation from cyclic–5 gives 0.25 seconds and 5MB.

Future directions include the addition of signatures, and parallelization.

4.2 Practical Groebner Basis Computation — Roune

See `http://arxiv.org/abs/1206.6940` ([RS12]) for the full version. Signature algorithms are just like Buchberger except that everying is lifted frm R to R^m. Fix a term order on R^m. Classic reduction of f by g is replacing f by $f - mg$ if $\text{in}(f) = \text{in}(mg)$ for a monomial m. Signature reduction says that $\text{in}(\overline{\alpha}) = \text{in}(\overline{m\beta})$ in R^m (m appears overloaded here!), and $\mathfrak{s}(\alpha) > \mathfrak{s}(m\beta)$. Want to construct all the Koszul signatures. Naïvely, this takes too much memory.

What is the best way to check ?? So these signature-lead ratios are very useful.

In any Buchberger-like algorithm, we need to order S-pairs according to some $\prec$. Ordering is either time- or space-intensive. Let $f(u)$ be a 0/1-vector (divmask): $f(u)_i = 1$ iff $d_i | u$, then $f(u) \not\leq f(v) \Rightarrow u \not| v$. Embed this into Kd-trees.

In reduction, we reduce the maximal term at each step, but what is maximal? Priority queues for reduction, maybe with a hash table to merge. Generally $2\times$ faster ($1.5\times$ for Geobuckets [Yan98]).

Q–JHD Why does hashing improve Geobuckets — I thought they merged?

A My hashing merges even if things aren't (yet) compared.

4.3 A Signature-Based Algorithm for Computing Gröbner Bases in Solvable Polynomial Algebras — Ma

Quotes [Fau02] as the start of signature-based algorithms. Also Gröbner bases in non-commutative rings: [Gal85] etc.

We propose a signature-based Gröbner-base algorithm in solvable polynomial algebras. This gives simpler proofs of correction and termination for [Gaoetal2010].

Let R, a K-algebra have the fact that $\mathbb{M}$, the set of monomials, is a k-basis of R and $x_j x_i = c_{i,} x_i x_j + p_{i,j}$ with every term of $p_{i,j}$ less than $x_i x_j$. Let $I = \langle f_1, \ldots, f_m \rangle$ be a left ideal in R.

Let $t = \text{lcm}(\text{pp}(f), \text{pp}(f))$ and $g^{[u]} < g^{[v]}$ iff ....

Step 1 Initialisation.

Step 2 Choose *any* critical pair. If it is regular and cannot be rewritten, we reduce the S-poly of the pair by G and create the signature

Step 3

Remark 1

Remark 2

Note that, due to non-commutativity $S(f,g) \neq -S(g,f)$ in general, hence some reductions to zero cannot be avoided. Showed various comparisons: 95% of CPs rejected by the reduction criterion.

Q Signature-based algorithms tend to compute more critical pairs: have you compared with non-signature-based ones?

A Not yet.

4.4 Complexity of deciding connectivity in semi-algebraic sets: recent results and future research directions — Roy

The number of connected components is $O(d)^k$ for equations of degree d in $\mathbf{R}^k$. This comes from Oleinok, Petrowski, Thom, Milnor etc. This talk — see [BRSEDS12].

[SS83], which has doubly-exponential complexity. To control adjacencies between cells, make all polynomials monic with respect to the elimination variable. Note that projection takes d/k to $d^2/(k-1)$ to $d^4/(k-2)$ hence the doubly-exponential (in k) is not surprising. Canny (and others) produced algorithms of singly-exponential complexity, based on the *roadmap* idea. The roadmap is one-dimensional, connected in every connected component of the set. Construction of the roadmap is based on recursive calls to itself on several $O(d)^k$ in $(k-1)$-dimensional slices. *But* these are calls on polynomials with degree d in $)k-1)$-variables. Hence the number of recursive calls is $d^{O(k^2)}$, but there are some genericity issues. can be solves [Basuetal] using infinitesimal deformations, and hence general theory of real-closed fields.

- Number of components is $O(d)^k$.
- Can test emptiness and compute Euler-Poincaré characteristic on $D^{O(k)}$.
- [D'AcuntoKurdyka] geodesic diameter of any connected component of a real variety is $d^{O(k)}$.
- New constructions for roadmaps, due to Safey and Schost, successfully applied to smooth real hypersurfaces.

- New recursive scheme where dimension drops by $\sqrt{k}$ (baby-step/giant-step), so we might expect $d^{O(k\sqrt{k})}$. We need generic coordinates since non-singularity of polar varieties is only true for a Zariski-closed set. We know of no way of doing this deterministically, hence the algorithm is randomised.
- But taking sums of squares [BPR00] avoids genericity requirements. Infinitesimal deformations are valid in the original (non-generic) coordinate frame.

Example. $Q := X_2^2 + \cdots$. $\mathrm{Def}(Q, \zeta) := D^2 - \zeta(X_1^{10} + x_2^{10} + X_3^{10} - 1)$.

Definition 12 *$S \subset \mathbf{R}^k$, $M \subset S$ a finite set of points. A* roadmap *for (S, M), $RM(S, M)$ is a semi-algebraic set such that*

1.
2.

Theorem 37 ([BRSEDS12]) *We have*

1. *Algorithm for constructing a roadmap for V in $d^{O(k\sqrt{k})}$.*
2. *Algorithm for counting the number of connected components of V in $d^{O(k\sqrt{k})}$.*
3. *Algorithms for deciding if two points are in same component $d^{O(k\sqrt{k})}$.*

4.4.1 The classical algorithm

The classic algorithm takes a bounded,non-singular generic hypersurface $Zer(Q, R^k)$. Construct a finite set of points intersecting every connected component of $Zer(Q, R^k)$: X_1-critical points of $Zer(Q, R^k)$.

For the classical roadmap, we take the X_2-pseudo-critical points parameterised by X_1. Construct the "silhouette": set of X_2-pseudo-critical points Distinguished values D: union of the X_1-pseudo-critical points and ...

Definition 13 *$S^0, S^1 \subset S$. Has* good connectivity property *if for every connected component C of S, $C \cap (S^) \cup S^1)$ is connected.*

We have to repeat the construction in every distinguished hyperplane ($O(d)^k$ of these) H_i defined by $X_1 = v_i$ with input $Q(v_i, X_2, \ldots, X_n)$ and distinguished points M_i.

4.4.2 Baby-step/Giant-step

Instead of considering curves in the X_1-dimension, we consider a p-dimensional subset V^0 which is a closed semi-algebraic set of dimension p,a nd for every $y \in R^p$, V_y^0 is a finite set of points having non-empty intersection with every connected component with V_y. Then $M^0 \subset V$ is a finite set such that the

intersection of M^0 with every connected component of S^0_a is on-empty for $a \in D^0 = \pi_1(M^0)$. Moreover, for every interval $[a,b]$ and $c \in [a,b]$ .... Let $N = \pi_{[1.p]}(M \cup M_0)$. Then $(V, V^0, V_N,)$ has good connectivity property. So the baby steps are computing a classic roadmap f V^0 passing through V^0_N, and the giant steps compute the roadmap ....

Fixing a whole block of $\sqrt{k}$ variables at a time necessitates a new kind of algebraic representation, known as a *real block representation*. Complexity issues here, and over limiting processes.

What next? Divide and conquer is a very natural idea. Consider a $k/2$-dimensional subset S^0 of S and make truly recursive calls at S^0 and S^1 which will be the union of certain $(k/2)$-dimensional linear spaces intersected with S. We would need to prove that (S, S^0, S^1) has good connectivity. [BRSEDS12] — 50 page paper.

4.5 2-closed Majorana representations — Seress

$M \equiv Aut(V_M)$: V_M is the Griess algebra — 196884-dimensional algebra over $\mathbf{R}$ with a commutative non-associative algebra and a positive definite bilinear form. This satisfies axioms

M1 $(u, vw) = (\ldots)$

-
-
-

M6 and M7 are equivalent to fusion rules. M3–M7 idempotents are known as M-axes. In M there are two conjugacy classes of involutions: 2a and 2b in Atlas. $\phi : M \to GL(V)$. For a 2a t, there is an M-axis $a_t \in V$

There is a Moonshine module $V^\#$ which is a Vertex Operator Algebra whose automorphism group is M. In many VOA, there are Miyamoto involutions, with associated idempotents satisfying M1–M7. The involutions $\tau(a) | a \in A$ are the restrictions of the Miyamoto involutions of $V^\#$. Ivanov used the phrase ....

What happens if we want M-representations of arbitrary groups. Let $G = \langle T \rangle$ where the generating set T is the union of conjugacy classes on involutions.

Theorem 38 (Norton) *In V_M there are two types of dihedral subalgebras (i.e. algebras generated by two M-axes, associated to 2A involutions.*

Theorem 39 (Sakuma2007) *If $G - D_{2n}$ is dihedral, then ... (essentially Norton in general)*

The algebras are named from the monster: 1A, 2A/B, 3A/C, 4A/B, 5A 6A.

Sakuma's theorem shows that M1-M7 captures some essential features of the Monster. We can therefore look at sub-algebras of the Griess algebra without needing to consider the 200,000-d matrices of the whole algebra.

Fix the *Shape* of the representation: the types of the dihedral subalgebras which must respect certain rules; Introduce a set C of algebra generators needed in the dihedral subalgebras. If two $D_{2n} < G$ have the same C_n subgroup then the algebra elements are the same.

We have a fully automated GAP program, which succeeds if the representation is 2-closed.

Q

A

4.6 An Efficient Programming Model for Memory-Intensive Recursive Algorithms using Parallel Disks — Cooperman

Note — really about permutations. Discs offer 100 times the storage of RAM, and on a cluster they basically come for free. Our *programming model* in agnostic over MapReduce, Hadoop etc.

Case Study: NFA $\rightarrow$ DFA $\rightarrow$ minimal DFA. [Knu68, Vol 1, 2.2.1] All permutations that do not contain (231) can be sorted by an infinite stack, so (231) is the *forbidden permutation*. And much else, Atkinson etc. `http://en.wikipedia.org/wiki/Permutation_pattern`.

An NFA is converted to a DFA just by multiplying states as appropriate. If we can collapse a pair of DFA states, then this may rigger more collapse.

- 3-buffer followed by 10-stack has a minimal set of 12,636 forbidden permutations of lengths between 7 and 18. [Linton2011]
- In fact for 11 or 12 stack depth get the same results, hence we conjecture it has stabilised. 10 needed 22GB and 162min; 11 needed 81GB and 560 minutes, 12 needed 295GB and 32hours.

Bibliography

[Abb06] J.A. Abbott. Quadratic Interval Refinement for Real Roots. *Poster at ISSAC 2006*, 2006.

[ABL+10] D. Andres, M. Brickenstein, V. Levandovskyy, J. Martín-Morales, and H. Schönemann. Constructive D-module Theory with SINGULAR. `http://arxiv.org/abs/1005.3257`, 2010.

[Bôo0a] M. Bôcher. Of linear dependence of functions of one variable. *Bull. Amer. Math. Soc.*, 7:120–121, 1900.

[Bôo0b] M. Bôcher. The theory of linear dependence. *Ann. Math.*, 2:81–96, 1900.

[BCvHP11] A. Bostan, F. Chyzak, M. van Hoeij, and L. Pech. Explicit formula for the generating series of diagonal 3D rook paths. `http://arxiv.org/abs/1105.4456`, 2011.

[Ber75] D.N. Bernshtein. The Number of Roots of a System of Equations. *Functional Analysis and Applications (trans. from Russian)*, 9:183–185, 1975.

[BES11] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *Proceedings ALENEX 11*, pages 35–47, 2011.

[BFP09] L. Bettale, J.-C. Faugère, , and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Crypt.*, 3:177–197, 2009.

[BK12] M. Burr and F. Krahmer. SqFreeEVAL: An (almost) optimal real-root isolation algorithm. *J. Symbolic Comp.*, 47:153–166, 2012.

[BOT88] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings 20th. Symp. Theory of Computing*, pages 301–309, 1988.

[BPR00] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *J. AMS*, 13:55–82, 2000.

[BRSEDS12] S. Basu, M.-F. Roy, M. Safey El Din, and É. Schost. A baby step-giant step roadmap algorithm for general algebraic sets. `http://arxiv.org/abs/1201.6439`, 2012.

[BS83] W. Baur and V. Strassen. The Complexity of Partial Derivatives. *Theor. Comp. Sci.*, 22:317–330, 1983.

[CGL09] J. Cheng, X. Gao, and J. Li. Root isolation for bivariate polynomial systems with local generic position method. In *Proceedings ISSAC 09*, pages 103–110, 2009.

[CLLLR05] M. Coste, T. Lajous-Loaeza, H. Lombardi, and M.-F. Roy. Generalized Budan-Fourier theorem and virtual roots. *J. Complexity*, 21:479–486, 2005.

[DET07] D.I. Diochnos, I.Z. Emiris, and E.P. Tsigaridas. On the complexity of real solving bivariate systems. In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 127–134, 2007.

[EGT10] I.Z. Emiris, A. Galligo, and E.P. Tsigaridas. Random Polynomials and Expected Complexity of Bisection Methods for Real Solving. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 235–242, 2010.

[EP11] S.M. Engdahl and A.E. Parker. Peano on Wronskians: A Translation. `http://mathdl.maa.org/mathDL/46/?pa=content&sa=viewDocument&nodeId=3642&pf=1`, 2011.

[Fau02] J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F_5). In T. Mora, editor, *Proceedings ISSAC 2002*, pages 75–83, 2002.

[FPPR12] J.-C. Faugère, L. Perret, C. Petit, and G. Renault. Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Field. In *Proceedings Eurocrypt 2012*, pages 1–15, 2012.

[Gal85] A. Galligo. Some Algorithmic Questions on Ideals of Differential Operators. In *Proceedings EUROCAL 85*, pages 413–421, 1985.

[GJ79] M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. *W.H. Freeman*, 1979.

[GKKP10] B. Grenet, E. Kaltofen, P. Koiran, and N. Portier. Symmetric Determinantal Representation of Formulas and Weakly Skew Circuits. `http://arxiv.org/abs/1007.3804`, 2010.

[GLL09] M. Giesbrecht, G. Labahn, and W. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comp.*, 44:943–959, 2009.

[GPS01] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR — a computer algebra system for polynomial computations. In M. Kerber and M. Kohlhase, editors, *Proceedings Calculemus 2000*, pages 227–234, 2001.

[GTZ88] P. Gianni, B.M. Trager, and G. Zacharias. Gröbner Bases and Primary Decomposition of Polynomial Ideals. *J. Symbolic Comp. 6*, pages 149–167, 1988.

[GVLM98] L. Gonzalez-Vega, H. Lombardi, and L. Mahé. Virtual roots of real polynomials. *J. Pure Appl. Algebra*, 124:147–166, 1998.

[JPS11] C.-P. Jeannerod, C. Pernet, and A. Storjohann. Rank-profile revealing Gaussian elimination and the CUP matrix decomposition. `http://arxiv.org/abs/1112.5717`, 2011.

[KK08] E. Kaltofen and P. Koiran. Expressing a Fraction of Two Determinants as a Determinant. In D.J.Jeffrey, editor, *Proceedings ISSAC 2008*, pages 141–146, 2008.

[Knu68] D.E. Knuth. The Art of Computer Programming, Vol. I, Fundamental Algorithms. *Addison-Wesley*, 1968.

[KPR+10] M. Khonji, C. Pernet, J.-L. Roch, T. Roche, and T. Stalinski. Output-Sensitive Decoding for Redundant Residue Systems. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 265–272, 2010.

[KPT12] P. Koiran, N. Portier, and S. Tavenas. A Wronskian approach to the real τ-conjecture. `http://arxiv.org/abs/1205.1015`, 2012.

[KS11] M. Kerber and M. Sagraloff. Root Refinement for Real Polynomials. `http://arxiv.org/abs/1104.1362`, 2011.

[MM82] E. Mayr and A. Mayer. The Complexity of the Word Problem for Commutative Semi-groups and Polynomial Ideals. *Adv. in Math.*, 46:305–329, 1982.

[Mon05] P.L. Montgomery. Five, Six, and Seven-Term Karatsuba-Like Formulae. *IEEE Trans. Computers*, 54:362–369, 2005.

[Pea89a] G. Peano. Sur le déterminant Wronskien. *Mathesis*, 9:75–76, 1889.

[Pea89b] G. Peano. Sur les wronskiens. *Mathesis*, 9:110–112, 1889.

[Rou09] B.H. Roune. The Slice Algorithm for irreducible decomposition of monomial ideals. *J. Symbolic Comp.*, 44:358–381, 2009.

[RS12] B.H. Roune and M. Stillman. Practical Groebner Basis Computation. `http://arxiv.org/abs/1206.6940`, 2012.

[Sag10] M. Sagraloff. On the Complexity of Real Root Isolation. `http://arxiv.org/abs/1011.0344`, 2010.

[SS71] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:282–292, 1971.

[SS83] J.T. Schwartz and M. Sharir. On the "Piano-Movers" Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[ST11] A.W. Strzeboński and E. Tsigaridas. Univariate real root isolation in an extension field. `http://arxiv.org/abs/1101.4369`, 2011.

[Str10] A. Strzeboński. Computation with Semialgebraic Sets Represented by Cylindrical Algebraic Formulas. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 61–68, 2010.

[SY11] M. Sagraloff and C.K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *Proceedings ISSAC 2011*, pages 353–360, 2011.

[Vas98] W.V. Vasconcelos. Computational methods in commutative algebra and algebraic geometry. *Springer*, 1998.

[vdH02] J. van der Hoeven. Relax, but Don't be Too Lazy. *J. Symbolic Comp.*, 34:479–542, 2002.

[VSBR83] L.G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast Parallel Computation of Polynomials using Few Processors. *SIAM J. Comp.*, 12:641–644, 1983.

[Yan98] T. Yan. The geobucket data structure for polynomials. *J. Symbolic Comp.*, 25:285–294, 1998.