International Congress of Mathematical Software 2014

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

5–9 August 2014

Contents

Chapter 1

5 August 2014

1.1 Math Soft/Soft Math: Buchberger

This will be a "political" talk, and subjective (possibly provocative).

In mathematics we think deeply, once to produce insight, in order not to think about infinitely many instances of the insight.

Example 1 *If $t|x$ and $t|y$ then $t|x-y$, and this insight gives us the Euclidean algorithm.*

Note the importance of meta-mathematics, mathematics applied to itself. This gives us metamatics.

Example 2 *The Roman calculation XIII·IX (=117=CXVII). Apply the distributive law, which is easier in a positional system, and then we get a multiplication algorithm.*

Example 3 *F is a Gröbner basis iff $\forall f, g \in FS(f,g) \to^F 0$, and this solves the problem of recognising a Gröbner basis, and more subtly an algorithm for computing one.*

Example 4 *Show that Newton iteration converges to an interval including $\sqrt{x}$: $a := \frac{ab+x}{a+b}$, $b := \frac{x+b^2}{2b}$. But suppose we vary the iteration. See [EH13].*

Example 5 (Generalise Example 3) *Idea.*

1. *Attempt a correctness proof for P using an algorithm scheme*
2. *When this doesn't verify (it won't) extract specifications for sub-algorithms*
3. *Recurse*

Claims that, had Gröbner had this paradigm, Buchberger himself would have been redundant.

Note that Gödel incompleteness means that the (meta-) hierarchy will never terminate. Hence "our works always becomes more exciting, and we leave more interesting problems to the next generation".

1.1.1 We live in a mathematical world

Example 6 *My luggage is in Hong Kong, identified by mathematics, and it is hard to explain that baggage clerks live by mathematics.*

Professors have a direct responsibility to explain the nature of mathematics, and do that in the language of our customers. In particular we have an obligation to explain this via programming languages, as the computer is the ultimate to the idiot.

Example 7 *NIST tables [OLBC10] can now have much more verification applied to them.*

It is up to the mathematicians, not school teachers, not government, to lead the recognition of mathematics. We must not give up the use of software as part of this explanation process.

This progress will lead to a great change in the way mathematical knowledge is disseminated: moving beyond digitised formulae to actual digitised knowledge.

There is a two-fold risk:

People no need to teach the theory: just press the button;

Purists we shouldn't use technology, because of this risk.

Hence I have a "white-box, black-box" principle: no time to explain.

Q We need an orchestra: we can't all be equally good at software,politicians etc.

A Yes and no. I play in a band, and I need to appreciate what the other members do.

1.2 Flyspecking Flyspeck: Adams

I work now for FireEye Dresden: a firm producing verified anti-malware. Note that mathematical proofs are getting longer (e.g. FSG) and use computers (e.g. Four-colour, Kepler). Theorem-prover is a program that implements a formal logic to recreate proofs.

1. Four Colour Theorem [Gonthier, Coq]
2. Feit-Thompson lemma [Gonthier, Coq] [Gon12]
3. Kepler Conjecture [Hales *et al.*, HOL Light] [Hal12].

So why should we trust these formalisations? Flyspeck breaks task down into lemmas, which are farmed out to workers who get paid a bounty for each lemma. Almost finished, and we have 500Kloc of HOL Light. Note that the theorem is 4-lines, uses `ball`, `packing` (defined as a set of points ≥ 2 apart) etc. Note

that we need to know that all added axioms (if any) are consistent. Also, do we trust the theorem prover. There is "Pollack-consistency" — is the statement printed out what the internal state believes?

HOL Light Doesn't capture basic type definitions (Hales uses this to import results[1]), Pollack inconsistency, OCaml vulnerabilities.

Coq Large trusted kernel (occasional unsoundness reported), complex formal logic, Pollack inconsistency, OCaml vulnerabilities.

Hence auditing the proof is nontrivial. Am trying to produce a theorem-checker HOL Zero that is just that much smaller. I think the Coq kernel have made too many compromises.

1.3 Symbolic Computer Package for Mathematica for Versatile Manipulation of Mathematical Expressions

I particular we want to talk about unevaluated derivatives etc. Note that we allow multiple notations for derivatives.

1.4 Representing, Archiving and Searching the Space of Mathematical Knowledge: Kohlhase

Claims that mathematics does this wrong, by being document-focused. Note that documents don't appear in Buchberger's creativity spiral. In the actual doing of mathematics, we abstract away from syntactic differences, recognise known parts, and identify unknown parts.

> It is less work, and more pleasant, to program a graduate student than a theorem prover.

Claims that the mathematical knowledge space is the structured space of represented and induced mathematical knowledge. Current search engines work in the represented space, not the induced space.

MatWebSearch showed

Mizar Applicable Theorem Search, but depends on knowing what is free/substitutable.

MMT Shows the modular tree [Rab13]. Note that the import statements may rename, but are truth-preserving. A drawback of this is that the axioms of ring, say, are not in one place. This flattening processes induces a actor of 40 in the LATIN database. Searching on this is the first thing MK has sees that searches the induced knowledge.

[1] Some lemmas take thousands of hours.

1.5 Gröbner Cover: Montes

The C-representation of a segment S is $\mathbf{V}(p) \setminus \mathbf{V}(q)$. The P-representation of S is a set $\bigcup_j \left(\mathbf{V}(p_j) \setminus \bigcup_j \mathbf{V}(p_{i,j}\right)$.. See [Wei92] for existence, various papers improving the output

1. Homogenise w..r.t. variables
2. Compute a disjoint reduced Comprehensive Gröbner System
3. Compute the P-representations
4. Add the segments with common lpp
5. Dehomogenise
6. For every segment, compute generic representation
7. If necessary, full representation

Example 8 (Steiner–Lehmus) *[MontesRecio2014]. Challenge is to distinguish internal bisectors from external.*

1.6 Maximum Likelihood Function for Parameter Estimation in Point Clouds by Gröbner bases

A laser system produces a vast point could.

Q Also used for reconstructing African temples. What's new here?

A Nonlinear (?).

Q Were these floating point Gröbner bases?

A In fact exact arithmetic. Mathematica.

1.7 What's new in CoCoA: Abbott

Also presented at ICMS 2006, 2010. Pascal first, then C. A closed-source package.

2000 New design

CoCoAlib Open source C++ Library.

CoCoA-5 specialised interactive server

CoCoAServer An OpenMath "server program".

Motto "No nasty surprises".

Lists `[...]` —shows "intuitive" examples. The input is readable.

Approximate arithmetic via TwinFloats — note that we assume the starting values are "exact". Example with TwinFloat reporting that more precision is needed.

Note that there's a natural route: write in CoCoA-

Q-JHD One of the aims of COBOL was that it should be readable by managers, and that didn't work. Will it work for you?

A Good point! Shows continued fractions. There's a built-in CoCoAlib iterator.

Q CoCoA, unlike Singular, is not strongly typed.

A This is almost a philosophical question. Comparing objects of different types is a good example — should you give an error, in which case looking up items in heterogenous lists doesn't really work. Therefore CoCoA doesn't.

1.8 Application of Groebner Bases to Nonlinear Mechanics Problems: Liu

1.8.1 Large Deflection Static Cable Analysis

Power lines, mooring systems, bridges. L is initial cable length, A is cable cross-section, Q is a constant force per unit of undeformed length, and perpendicular to the cable. q is self-weight per unit length: $q \ll Q$. E is Young's modulus, and T_0 is cable's pre-tension. Let u and v be horizontal and vertical displacement. T is the real tension. Get nonlinear differential equations.

1.8.2 Another cable problem

1.8.3 Geometrically nonlinear Plate Analysis

Equally, want the displacement, in-plane and transverse.

All these problems are examples of nonlinearity, which is key to many application areas. Various ways to convert differential equations to algebraic ones. Note that we still need numerics if polynomials have degree > 4. An example with Galerkin conversion has degree 25. It does produce subtly different results from the linearised analysis. The plate does give a third-degree equation, hence "analytic" solutions.

Q But this is only numerical — heated debate.

A Her trial functions show about a 1% difference from Ansys simulations.

1.9 Real Quantifier Elimination in the Regular-Chains Library: Moreno Maza

Define Quantifier Elimination. Unifying concept is that of "regular chain". A heavily-typed Maple library.

Showed an example which he claimed[2] to be [DH88], but said this eliminated to $(d-1=0) \vee (d+1=0)$, which he stated was false.

Define CAD. First idea was projection/lifting. Our new idea is based on RCs, but more precisely on "case discussion". Example where Quantifier Elimination can't be done directly, since we may need to express $RealRoot_2(f)$, which needs derivatives of f. Example [ST11], which we solve in 15 seconds.

Q That was just satisfiability.

A Yes, but many problems are not.

1.10 Skolemization Modulo Theories: Veanes

Skolemisation eliminates quantifiers in favour of function variables. preserves satisfiability. Original use of Symbolic Finite Transducer (SFT).

1.11 QBF

Quantified Boolean Formulas. Pspace-complete. many applications in model verification. Potentially exponentially more succinct than non-quantified Booleans. In practice we have a family of related QBFs to be solved: how can we use this? Use backtracking/resolution. Assume Boolean ϕ in CNF, quantified by blocks B_i of variables. No free variables. Showed a high-level DLL algorithm specialised to QPF. basic backtracking plus constraint learning (both constraints and cubes). There are issues with incremental learning: must discard things no longer true.

Doing incremental solving, we want to to re-use a maximal subset of the constraints learned when solving ϕ_i when solving ϕ_{i+1}. Solvers tend not to maintain the derivation. We add a fresh existential variables (selectors) to each clause, then these propagate to learned constraints. Don't seem to have a good way of propagating learned cubes.

Example 9 (Conformant planning) *Incremental approach solves more, in 14.5 rather than 24.4 seconds. However, we have examples where the converse behaviour is true, which is odd.*

`http://lonsing.github.io/depqbf`.

Q–JHD Is it just ϕ_i that feeds into ϕ_{i+1}?

[2] JHD: it's not. He seemed to have the Heinz construction, but didn't have the real/imaginary split.

A No, all learned information is available.

1.12 Quantifier Elimination for Modular Constraints; John

We have linear modular equalities (LME), inequations (LMD) and even inequalities (odd, since modular). Tends to work either by bit-level blasting, or going into integers.

$$\exists x((2x + z \neq 0) \wedge (2x + 3y = 4) \wedge (x_y \leq 3) \pmod 8$$

The moduli are powers of 2, so sometimes bits are irrelevant. Layer 3: can do a Fourier–Motzkin style Quantifier Elimination.

$$\exists x.((y \leq 4x) \wedge (4x < z))$$

which gives a FM case analysis. The number of cases is independent of the bit width, and really only depends on whether left/right side overflows.

If all else fails, do model enumeration, but we have never needed to.

Benchmarks from LinDD [Chakietal2008] regarded as 16-bit integers. Also VHDL examples. For VHDL, layer 3 was only used 0.5% of the time.

We are combining DD and SMT, which are very different algorithms, and this combination is more powerful than either. Future work; non-linear.

Q FM normally requires a variable that isn't multiplied. How do you avoid this?

A There are limitations, but we seem always to get round this.

Chapter 2

6 August 2014

2.1 Principles of Independence for Robust Geometric Software Learned by the Human Visual Computation: Sugihara

> We can avoid failure of geometric computation, even if we use imprecise computation, provided we respect independence.

Numerical computation normally gives **imprecise results**, but in geometry, this can cause inconsistency, and hence **no result**.

Example 10 (Voronoi diagram) *Incremental method, add one point at a time, adding perpendicular bisectors one at a time. This should give a closed curve of new lines, but if this fails,not just numerically but topologically, we have a problem.*

Example 11 *Various optical illusions are basically caused by failing to reconstruct 3 (z coordinates) from a 2D drawing. We have various equalities, and inequalities representing "F_1 hides F_2" etc. Then "picture represents polyhedron" $\Leftrightarrow$ "$A.w = 0, B.w > 0$ has solutions". But if he tries a truncated tetrahedron, the software fails as the three lines, which should be coincident in the point of intersection of the three planes, are not* **exactly** *coincident.*

The obvious solution is to introduce "fuzz", but then the Penrose triangle is accepted as valid as well.

There are basically three approaches to numerical error.

1. Three-valued logic [Milencovic1988,Guibas1989,Fortune1989][Hof89], but this has almost disappeared today.
2. Exact Computation (Mehlhorn/Pion/Yap). Many people pursue this.
3. Topology-based. [SugiharaIri1989,Sugihara2013]. Aim of today's talk

An optical illusion is an extreme response of the human visual system.

Example 12 (Slanted Line) *[Fraser1908] Black/white diagram looks like a spiral, but tracing in colours proves it's concentric circles.*

Example 13 (Slanted characters) *In Kanji: Meaningless sequence, but get the illusion of sloping lines even though the characters are in fact properly aligned.*

Example 14 *Ouchi illusion [Spillmannetal1986].*

Note that, in the human visual system, each neuron covers a small area of the retina. Each neuron detects one component (vertical or horizontal) of movement.

Example 15 (Escher's staircase) *Note that this is possible if one flight is sloping. But the human brain prefers aligned rectangles, so perceives an impossibility.*

Example 16 *Also a video (in fact five) with "impossible motion", since the bodies are in fact slanting, but we perceive verticals. Even having had the 3D structure rotated, and the illusion "explained', when it rotates back to original position, the illusion is restored.*

Also, human visual computation is local, and does not change as a result of global inconsistency.

It is inconsistency, not error, that causes the geometric failure. Hence we must divide our predicates into independent ones $P_1, \ldots, P_k$ and other (dependent) ones $P_{k+1}, \ldots$. Nice principle, but hard to implement. So our principle is to place higher priority on the preservation of topology. In Example 10, we will need to remove only a tree-structure from the old Voronoi diagram when adding a new point, and numerical calculations that contradict this should be ignored.

"Extract mutually independent predicates and use numerical computation for them only. Then we can achieve numerical robustness."

Q How do you determine independence?

A That's hard: involves knowing all geometric theorems. Hence topological consistency is easier.

Q Is the logical reasoning done by hand?

A Currently. I looked at the Voronoi diagram one, and it as computationally infeasible.

Q Is the "rectangular preference" nature or nurture?

A The psychological community is not convinced of this preference, actually. Small children are upset by these illusions, which implies something — what?.

2.2 Discourse-level Parallel Markup and Meaning Adoption in Flexiformal Theory Graphs: Kohlhase

Yesterday (Section 1.4): "documents are bad". Implemented in MMT, where the arrows in the diagram are actually "clickable". These assumes that all mathematics is formalisable.

1. printed
2. digitised
3. presentational (claims "this means it can be read to you by a screen-reader")
4. semantic (a minuscule fraction)

We believe that mathematics is formalisable, but don't do that in practice. Note that Flyspec (section 1.2) is maybe 3 years of mathematics and 20 of formalisation.

Claim that we need **parallel markup** — semantics and presentation,suitably cross-referenced. Multiple presentations supports multiple natural languages.

However, mathematical texts are more than formulae. Shows parallel markup of a definition (largely words) and formal mathematics. We need to take informality seriously in theory graphs. Therefore proposes an extension of MMT, "postulated views"[1], and this means that the "black box" informality becomes "grey box". Therefore OMDoc 2 is/will be OMDoc with MMT/opaque.

Q Use by people outside your group?

A Yes, we have a semantic atlas ....

2.3 Theorema 2.0: A System for Mathematical Theory Explanation: Windsteiger

Theorema aims to support *all* aspects of mathematical work. Hence my presentation here is a Theorema document. Claims that this should be the "pencil and paper of 21st century". Implemented in Mathematica, i.e. the programming language and the notebook front-end. Mathematica algorithms are *callable*, but are not automatically invoked. Have the notebook interface, but also the "Theorema commander" window

This differs from Section 1.2, in that we don't have a small kernel, rather a multi-method prover. The language is higher-order predicate language (currying etc.). Instead of a small trusted kernel, trust should come from readable proofs.

[1] JHD understands "postulated" to mean "user-claims that this is a correct view".

There could also be a program that inspects the proof object, but we have not done this.

The challenge is integrating proving and computing (and solving, but what does this mean?). Shows an example, and the fact that each Theorema knowledge item can be deactivated from a graphical window.

Example 17 *If $A \subseteq B$, then $A \setminus C = A \cap (B \setminus C)$. There's a formal translation, with $\subseteq$ all replaced by elementary membership statements. Shows both the notebook, and a tree representation of the proof, which are mutually clickable.*

Q Is Theorema multilingual?

A Theorema 2 also has German, and Theorema 1 also had Japanese.

Q Note the flattening in the example.

A-Buchberger Not necessary: we could have a set-theory prover working in terms of these definitions.

Q Do you need Mathematica?

A Yes, we need that, so Theorema itself is free, but you need Mathematica.

Q–JB SAGE implementation?

A No.

Q–MK There's a lot of state from the commander in the proof — is this recorded.

A Yes: it's in the proof object, and these can be re-loaded.

2.4 Complexity Analysis of the Bivariate Buchberger Algorithm in Theorema: Maleszky

This is based on [Buchberger1980], which uses the chain condition.

Theorem 1 *In a graded ordering, then* $\deg(G) \leq 2\deg(F)$*, where G is a Gröbner base of F in $K[x, y]$.*

1. Tuples, orders etc.
2. Rewriting lemmas, e.g. about
$$x + \min_A a = \min_A (a + x). \tag{2.1}$$
3. Lift these lemmas to inference rules. This is more efficient, but also produces more natural proofs. Also knowledge about quantifiers.
4. Introduce deg, lcm, chain criterion etc.

5. Rewrite lemmas for these, using the rules in set 3
6. prove main theorem.

Note that (2.1) becomes

$$\frac{(\Psi \vdash \Gamma)_{\min(\langle t|_{i=a,\ldots,b}\phi\rangle)+x\to\min(\langle t+x|_{i=a,\ldots,b}\phi\rangle)}}{(\Psi \vdash \Gamma)} \tag{2.2}$$

The proof: 292 formulae and 230 proofs, with 29 inference rules. 3400 Mathematica LoC. In fact more general than the original proof. Also simplified.

Q–Greuel I don't believe the theorem.

A Note it's only true in two variables.

Q What is a "specialised prover"? How do I trust this.

A Our provers are not provably correct. However, the inference rules are supported by lemmas.

Buchberger It would be nice to have a general proof that this lifting technique is correct. Example: in analysis texts, theorems are proved about lim applied to functions, them applied to formulae etc.

MK I think this is Wang's "apply theorem", and as such is in Isabelle.

2.5 Towards Formalising a key theorem from Auction Theory using Theorema: Windsteiger

Joint work with Kerber and Rowat (Birmingham): challenge posed by [Kerber2010 at MIPS]. The theorem isn't fully proved yet. Example is auctions for mobile spectrum, some of which have been unsuccessful (in terms of low income). Economists blame "poor auction design": auctions are designed on paper. We use "second price auctions of a single indivisible good".

Theorem 2 ([Vic61]) *In this setting, truth telling is a weakly dominant strategy, i.e. no bidder can do strictly better by bidding above or below her valuation of the good, whatever the others do.*

> A previous formalisation: " Vickrey's theorem is stated in two sentences and proved in six. We need eight problem-specific definitions."

Theorema needed additional lemmas, to reason about maxima.

Q–MK "Formalism is good" isn't true for humans.

A When I question the authors, I get "this is not well-written" etc.

All A debate about the rôle of Theorema — AK: "every paper followed by a Theorema notebook", WW: "No, every paper should **be** a Theorema notebook".

2.6 An algorithm for computing Tjurina stratifications of μ-constant deformations using algebraic local cohomology: tajima

Definition 1 *f is quasi-homogeneous of degree d, w is the w-weighted degree is d. f is semi-quasi-homogeneous of degree d, w if it if $f_0 + f_1$ where f_0 is quasi-homogeneous of degree d, w with an isolated singularity at the origin and f_1 has higher degree.*

τ is the Tjurina number, and μ the Morse number. $\tau = \mu \Leftrightarrow f$ is quasi-homogeneous, Since $\mu - \tau$ is a complex analytic invariant that measures the no-quasi-homogeneity of a hypersurface isolated singularity. So consider a parametrised polynomial, e.g. $x^3 + y^10 + ax^{\cdots} + by^{\cdots}$ with $w = (10, 3)$. Then $\mu = 18$ (JHD: why) and τ depends on whether a, b are zero.

Any element of the local cohomology can be represented as Čech cohomology. The first step (basis of vector space) is computable [**?**]. Then for each condition of the parameters we ... Comprehensive Gröbner basis.

2.7 An implementation method of Boolean Groebner bases and comprehensive Boolean Groebner bases on general computer algebra systems: Nagai

Note that Boolean Gröbner Bases are not the same thing as $\mathbf{F}_2[x_i]/(x_i^2 - x_i)$. Rather, they are over B. B is any commutative ring such that $\forall b \in B : b^2 = b, b + b = 0$. Besides F_2, we have $F_2 \times \mathbf{F}_2$, a family of sets with $\cdot = \cap$ and $+$ being "xor". $B[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n) = B(x_1, \ldots, x_n)$ is a "Boolean polynomial ring".

Theorem 3 (Stone) *$B \equiv \mathbf{F}_2^k$: the isomorphism is the compositum of many projection maps. Since we can compute in $\mathbf{F}_2[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$, this gives us one route.*

Using polybori, we can do this in Sage or Asir.

Buchberger I used to do Sudoku, but you've ruined it.

Q Compare SAGE/ASIR?

A Sage 10× faster (3.1: 39) for the $\mathbf{F}_2$ computation, but Sage spent a great deal of time (> 200 seconds) reconstructing over B. Maybe I need Sage help.

2.8 Mathematical hierarchies of Sudoku puzzles and its computation by Boolean Groebner bases: Inoue

Sudoku difficulty levels are heuristically-assigned. $X_{i,j}$ is an element in the Sudoku grid, and $B = P(\{1,\ldots,9\})$. Sudoku structure induces equations (row/column/block). let I be the ideal generated by these. Let $Sing$ denotes the subset of B^9

Definition 2 *$f(x) = x + \{s\}$ is called a* solution polynomial*, and $f(x) = \{s_1,\ldots,s_l\} + \{s_i\}$ $(1 \leq i \leq l)$, $f(x) = (1 + \{s\})x$ are called semi-solution polynomials. Any of $f(x) = a$, $f(x) = x$ or $f(x) = ax + b$ (with $|b| > 1$) are called contradiction polynomials.*

Definition 3 *I is* solvable *if $V_{Sing}(I) \neq \emptyset$, and* uniquely solvable *if this has cardinality 1.*

Define ψ_0 and ψ_0^* on ideals. If $\psi_0^*(I)$ is maximal, generated by 81 polynomials, we say that$I$ is *basic solvable*. For these we can classify by basic rank.

However, some polynomials are solvable but not basic solvable. Concept of basic refutable rank. If $\psi_\infty^* I)$ is maximal, we say that I is strategy solvable. Then the point at which $\psi_k^*(I)$ stabilises is the strategy rank.

We computed the ranks of 735 Sudoku's, classified 1–7 humanly. We see some correlation. The "World's Hardest" has s-rank ∞

Q–JHD Anything special about 9?

A No.

Q Does this imply that every human strategy will assign a finite rank?

A Yes

2.9 A method to determine if two parametric polynomial systems are equal

Problem 1 *Given an ideal I and a polynomial f, is $f \in I$. Solved by Gröbner bases.*

Problem 2 *Given a parametric ideal I and a parametric polynomial f, is $f \in I$ for any values of the parameters?*

Problem 3 *Given parametric ideals I, J $J = I$ for any values of the parameters?*

Let R be the parameter ring $k[u_1,\ldots]$Let E be a subset of R, and $\mathbf{V}(E)$ the common zeros in k^m.

Lemma 1 ([Sit2001]) *If I_1 and I_2 are quasi-algebraic sets, the $I_1 \cap I_2$ is quasi-algebraic. Quasi-algebraic is $\mathbf{V}(E) \setminus \mathbf{V}(h)$.*

Comprehensive Gröbner Bases. $\{(A_1, G_1) \ldots\}$. The A_i are algebraic sets. How do we compare these?

Example 18 *Same polynomials, Comprehensive Gröbner Bases by [SS01], [KSW10] are different*[2]*. Are they the same?*

Theorem 4 (Polynomial membership) *For each branch (A, G), let r be the pseudo-remainder of f by G. Then (*).*

Theorem 5 (Ideal membership) *For each branch (A, G_i), let r be the pseudo-remainder of f by G_i. Then (*).*

Theorem 6 (Ideal equality) *Let G_1 be a Comprehensive Gröbner Basis of I w.r.t. $<_1$ and G_2 be a Comprehensive Gröbner Basis of J w.r.t. $<_2$. For every pair of branches $(A, G \in G_1$, $(B, H) \in G_2$, let the r_i be the remainder of the generator of H w.r..t G and q_j vice versa. For each branch (A, G_i), let r be the pseudo-remainder of f by G_i. Then (*).*

JHD: (*) appears to be checking that the remainder (being nonzero) is incompatible with the A components.

2.10 Software for Quantifier Elimination in Propositional Logic

Assume a CNF $F(X, Y)$., and consider $\exists X : F(X, Y)$, which is $\equiv H(Y)$. This is relevant to reachability analysis.

The aim is to make the X-clauses redundant by adding resolvents: C is redundant if $\exists X F \equiv \exists X[F \setminus \{C\}]$. This is hard in general, but can approach via branching.

Use "atomic D-sequents".

Partial Quantifier Elimination. Given $\exists X(F \wedge G)$ find $H(Y)$ such that $H \wedge \exists X G \equiv \exists X(F \wedge G)$

Q How to find the maximal F?

A Describes various methods, but it looks heuristic.

2.11 Quantified Reasoning over the Reals: Gao

$\mathbf{R}, \geq, \mathcal{F}$. For example

$$\exists y \forall x_0 \forall t \forall x_t (x_0 = 0 \wedge x_t = f(x_0, u, t) \wedge x_t = 1) \rightarrow \left(\int_0^t c(x, u) ds < \int_0^t c(x, u') ds \right)$$

[2] JHD: the [KSW10] is not completely reduced, and if you do, you get the [SS01] basis.

is stability.

his is great but the problem is nonlinearity.

In this wild world we can't stick purely to symbolic methods

Quantifier Elimination is both more and less than is needed.

Assume $\mathcal{F}$ s type 2 Turing, i.e. allow real numbers represented as infinite sequences of rationals.

δ-weakening is replacing $>$ by $delta+$. So we want an algorithm that says: "ϕ is true" or "$\phi_d elta$ is false"

Theorem 7 *The δ-decision problem for a Σ_k-sentence is in [some complexity class].*

`dreal.cs.cmu.edu` — used by Toyota etc.

Chapter 3

7 August 2014

3.1 Chebfun as a software project: Trefethen

> I am a numerical analyst. For every one of you [symbolic folk] there are ten thousand of us. Floating-point arithmetic is fantastic.

Note that floating-point works by rounding, and this is also the principle of Chebfun. I care passionately about my ideas, and Chebfun: how do I ensure longevity?

3.1.1 Chebfun (demo)

In MatLab 2014a.

```
x=chebun('x',[-1,1]);
f=cos(100*x);
length(f);   % returns 149, so a
g=/(25*x^2);
length(g); % returns 181
h=f.*g;
length(h)   % less than 329, since Chebun rounds
```

Done by large amounts of Matlab overloading of built-in operators.

More recently, Chebfun does two-dimensions, using `chebfun2`.

3.1.2 Twenty Questions

I'm a newcomer(ish) to software engineering.

1. What do you do to make software last. In-house project; company; open source? I'm too old for (1), and the team didn't want (2) — might it have been different in the US, or in Korea? Hence number 3, but genuinely, in the sense that we are open to outside developers.

2. Which licence? Basically BSD or GPL. People told us "companies are scared of GPL", so we went to BSD (I forget which version!). But there are university bureaucrats, so we had to convince them that Chebfun was worthless.

3. Which programming language? We had five years of MatLab behind us. The overloading of MatLab was what inspired us. Look at the list of "most frequent 500 commands" in MatLab, and see which to overload. But there are concerns

   (a) Speed — good at what it's good at, but ....

   (b) Language – untyped, and non-matrix objects are curious.

   (c) Commercial software: most universities in the US have it, but less common (legally) elsewhere.

   Parts of "core" Chebfun (i.e. 2005) have been recoded in C/Octave/Python/....

4. How do make money out of Chebfun? We don't at the moment. One day we might found a consulting company.

5. How do we get funding? This is a challenge, especially in Britain. EPSRC CS (no money from Maths), MathWorks, and ERC grant (which, in principle, is funding algorithmic/mathematical research, not software development).

6. How do people get credit for work on software? I think this does count for me, as a "visible person": harder for youngsters, especially students/postdocs.

7. How do we learn about open source software? Recommended a book by Fagan.

8. Where do we host the code? Now GitHub — this has gone from 0 to 10^7 repositories in 6 years.

9. What's the coding and review process? We're in GitHub's pull request/ /review/merge process, which seems to work for us. Long discussions about spacing, Camel Case etc.

10. Testing? Introduced 'chebtest' in 2008. Rely (perhaps too much) on MatLab's platform independence.

11. Who responds to help? We have a tracker,and mail request. We haven't really addressed this.

12. How to make a good website? We have one bright guy!

13. Documentation? People do detailed comments, but I end up writing most of the

14. How do we get feedback from users? Badly! No registration process, and no database of users. Our knowledge is very anecdotal. Showed a Google Scholar list of citations of chebfun, with exponential growth (doubling every 2 years). Contrary to belief, this *doesn't count duplicates* — I

15. How should people cite Chebfun? We don't have a key paper. We now publish a new book every year, with three editors. Please cite this!

16. How does the team communicate? Oxford (most)/Delaware/Stellenbosch. Weekly meeting, with Google hangout[1] for the externals. Externals still complain.

17. How do we attract outside developers? We don't currently

18. Should we plan ahead? There's no requirements analysis etc. Most developments just "happened". We probably ought to get a bit better at it.

19. What do we fight about? Relative priorities. NT regards backwards compatibility as vital, while the youngsters care more about speed and functionality.

20. What's our management structure? Most procedures aren't spelled out. Notionally we're a democracy, but (heavily) weighted. How to balance "hallway chats" with a distributed team?

Q–JAA Software has a value, but isn't counted. I made this mistake and I am now unemployed.

A These are serious issues.

3.2 Business meeting

3.2.1 This meeting

Attendees: Japan top (30), then USA (18) and Korea (13) and a total of 130. By self-declaration: 62 mathematics, 29 CS, (out of 100 online registrations). 92 staff and 19 students, 8 free students and 9 reduced-rate.

[1] Used to be Skype, but that was frustrating.

Table 3.1: ICMS 2014: Income

Registration	34,950,000
Hanyang	10,000,000
NIMS	20,000,000

3.2.2 Byelaws

The current byelaws were put up. The officers are currently self-sustaining, and there is no provision for elections.

MK It is useful to have a structured name for the conference, for people to contact. A domain is also useful, and the obvious `.info` is free.

Hong (many seconded) there should be a 2/3 requirement on the changes of byelaws. JHD suggested for for e-mail it should be "2/3 of those voting with a minimum of 30". It was modified to be "2/3 of those present at a Business Meeting, or 2/3 of those voting by e-mail with a minimum of 30". **Carried**.

Byelaws Carried.

3.2.3 Future

Meetings

Publications MJ mentioned various existing publications

1. One in optimisation
2. Macaulay journal, now Algebra and Geometry

NT ACM ToMS.

JHD LMS JCM.

* I don't put software in journals as they demand copyright — in debate this turned out to be ACM.

He thought that a list of "software-friendly" journals should be maintained on the ICMS website.

Borwein The "Brown" workshop on reliability [SBB+13]. We have an absence of standards, rather than an absence of journals.

MK moved a vote of thanks to the organisers. **Carried**.

3.3 BULL! Molecular Geometry Library: Deok–Soo Kim

We now know how to decode the human genome. But what does the work is the proteins. Their primary structure is given by amino-acid bonds, which is composed into secondary structure by hydrogen bonds, then ternary and quaternary structures. These are published in NIH's protein data bank. It is agreed that a protein's function depends on its structure, but there is less agreement on what "structure" means. For example, what is the surface, the voids etc.?

Obama launched the Advanced Manufacturing Initiative. Note in particular Metal-Organic Framework (MOF). Need to understand the boundary and the channel structure. Again, want to know if there are voids/tunnels. Applications in fuel cells, Lithium batteries etc.

Claims that Voronoi diagrams are key to this. This gives answers to convex hull shortest path, and many other problems. The dual is the Delaunay triangulation.

We need the "additively-weighted Voronoi Diagram", where we look at distance from boundary of circles/spheres, rather than fro points. No longer straight lines, but hyperboloids. We still get Voronoi cells. has a dual structure, known as "quasi-triangulation". Consider a ball of radius β, and ask where it can pass between circles/spheres, and remove the lines through which it can pass. This gives β-complex. Given the β-shape, it is easy to compute the offset shape.

The following situation can arise with respect to β-complexes. We have two triangles sharing two vertices, which means we do not have a simplicial complex. Showed an example in 2D, then one in 3D. Here we get two tetrahedra sharing three faces, hence one has positive volume and one negative. There is another anomaly, known as "dandling face/small world", but that is all the anomalies.

Definition 4 *A simplicial complex C has two conditions*

1. *Any face on an element in C is also in C;*
2. *two elements of C, if they intersect at all, intersect in one element of C.*

Definition 5 *A quasi-simplicial complex C has two conditions*

1. *Any face on an element in C is also in C;*
2. *two elements of C, if they intersect at all, intersect in one* or more *elements of C.*

How common are anomalies? In practice very rarely, either on real proteins or on "random" data. Have been working on "topologically-reliable" alternative algorithm based on ideas from Section 2.1.

I propose that we transform molecular problems into geometric problems, where we can look for geometric solutions. The inverse transform to recover the molecular solution is, he claimed, similar to the electronic structure calculations.

Pure code can be downloaded. Has APIs for VD, QT and BC, aimed at developers/mathematicians, and a molecular modeller API. Example: once the VD and QT are computed, can, for example, count boundary atoms etc.

Q–Buchberger What is "empty space"?

A The van der Waals' radius is a "one size fits all" approximation.

Q Why β-complex, not α?

A Time (this led to a debate, as elsewhere it was claimed that α was faster). Buchberger thoughts that this was really an issue of pre-processing.

3.4 Regular Chains: Moreno Maza

A Maple library, or via `www.regularchains.org`. These slides etc. are also there.

Point of view is recursive polynomials, and a triangular set has main variables disjoint. See[Rit32, Wu87]. The quasi-component is $W(T) - V(T) - V(h_T)$ and $sat(T) = \langle T \rangle : h_T^\infty$

Theorem 8 $\overline{W(T)} = \mathbf{V}(\mathrm{sat}(T))$. *Moreover, if* $\mathrm{sat}(T) \neq \langle 1 \rangle$, *then* $\mathrm{sat}(T)$ *is strongly equidimensional.*

Definition 6 (Kalkbrener) T *is a* regular chain *iff* T *is empty or* $\mathrm{lcoeff}(T)$ *is invertible w.r.t.* $\mathrm{red}(T)$ *and* $\mathrm{red}(T)$ *is itself a regular chain.*

Definition 7 $T_1, \ldots, T_e$ *is a* Kalkbrener triangular decomposition *of* $\mathbf{V}(F)$ *if* $\mathbf{V}(F) = \bigcup_i \mathbf{V}(\mathrm{sat}(T_i))$.

Definition 8 $T_1, \ldots, T_e$ *is a* Wu–Lazard triangular decomposition *of* $\mathbf{V}(F)$ *if* ...

Kalkbrener solving is about the "generic solution". To allow for other cases, we need `output=lazard`.

Can get sample points (possibly more than one per component). Other tools, such as ConstructibleSets, CAD and Quantifier Elimination.

3.4.1 Parameters

Definition 9 *A regular chain* specializes well *at* $u \in K^d$ *if* $T(u)$ *is a square-free regular chain and ... (no polynomial vanishes, ?or drops in leading variable).*

Definition 10 *A CTD of* (F) *is a finite partition* $\mathcal{C}$ *of the parameter space into constructible sets, with, above each* $C \in \mathcal{C}$, *a family* T_C *of squarefree triangular sets such that each squarefree chain* $T \in T_C$ *specialises well at any* $u \in C$ *and "the solutions are right".*

Chapter 4

8 August 2014

4.1 Numerical Algebraic Geometry: Theory and Practice: Sommese

States [BHSW13]: we had different views when we started Bertini than we have now. Hence bertini has evolved, and will have a further major rewrite into C++. An observation I made for General Motors in 1986 saved 4 hours out of an 8-hour job.

- Systems tend to be sparse
- Therefore they have fewer solutions than one "might expect"
- users typically only want real solutions, generally only finite ones.

Homotopy continuation is our main tool:

$$H(x,t) = (1-t)f(x) + tg(x) \tag{4.1}$$

when $t = 1$ means we have g (known), and $t = 0$ we have f (unknown).

Originally we didn't worry about path-crossing — "if we jump, so what".

1985 3 minutes/path: numerical analysis plus topology.

1991 8 seconds/path on IBM 3081: people started using algebraic geometry.

2006 many paths/seconds, and many cores — this is a very parallel problem.

[MorganSommese1989] if the parameter space is irreducible, solving the system at random points simplifies subsequent solves, often $\times 100$. We solved the nine-point problem

4.1.1 Positive-dimension

We want the positive-dimensional solutions f a polynomial system.

- Use the intersection of a component with generic linear space of complementary dimension (note that this works over the complexes) to get a witness set
- use continuation and deformation.
- At this point path crossing becomes a fundamental challenge, since we need to understand the origins of witness points.
- Multiple points are also a problem, as they nullify the Jacobian. Slice the null space — [VerscheldeleykinZhao] This deals with isolated multiple points: what about repeated components.

Endgames [MorganWamplerSommese] — we uniformize around a solution at $t = 0$, then use Cauchy Integral Theorem.

But there are applications (hyperbolic PDEs) where the singular solutions are the object, rather than a distraction. Without multiprecision, singular points can only be computed reliably for low multiplicities. Some people think "large means infinity". We transform infinity to a finite point, but solutions at infinity often have high multiplicity.

Hence we need multiprecision

- To ensure the endgame
- Evaluation; $p(z) = z^{10} - 28z^9 + 1$ has a root $a = 27.999999999$, but $p(a) \approx -2$ by naïve evaluation. Clever evaluation is something we can't do in higher dimensions.
- In the 9-point problem, out of 1433690 paths, 1184 used higher precision and dropped back to double, but 680 used at least 96-bit precision.

Early versions of Bertini had many hard-coded limits, which we gradually had to remove. We've needed to add support for sparse linear algebra. Also need to inter-operate with computer algebra [BDH$^+$14].

Q How do you know you're following a good path?

A We track condition numbers.

Q–Hong Can you do this over the complexes without $\mathbf{R}^{2n}$?

A [He] is adding extra variables but staying over the reals.

Q How do you guarantee global correctness?

A Algebraic geometry guarantees continuity in the parameters.

Q–Trefethen Random numbers are a pain: we use a fixed "arbitrary" number.

A We have a seed, and it's always printed out.

4.2 An Introduction to Software in Numerical Algebraic Geometry: Hauenstein

Alt's problem — design a mechanism passing through 9 points, and there are 1442 such mechanisms. Most of these are complex, and many of the real ones have degeneracies, but we'll ignore that for now. Bertini: how many conics intersect 8 lines — answer 92. Can one do this over the reals?

HOM4PS-3.0 Section 4.4.

NAG4M2

Mostly interested in solutions over $\mathbf{C}$.

Example of solving an easy system, then moving it. The homotopy should exploit what structure there is in the target system. Having chosen a source g, multiplies it by random $\gamma \in C$ to "avoid singularities".

What about positive dimension? If we have irreducible components

Algebra Ideals (prime)

Geometry witness sets. Claims that these facilitate membership testing, sampling, intersection, projection and real points.

How do we split components? In particular, now do we partition the witness points? On an irreducible component, the smooth points are connected. [SommeseVerscheldeWampler2002]. So as we move our linear slice, these points should interchange within components. In practice, random loops work well.

The centroid of the points of a connected component must move on a line if we take parallel hyperplanes.

Q–NT How much of this is automated?

A Bertini allows a lot of user control: choose, or write, your own homotopy. Monodromy and trace test are totally automated.

4.3 Paramotopy: software for parameter homotopies: Bates

Can solve each of M problem separately, using P paths, total cost $P \cdot M$.

1. Solve the system for fixed (random) values of parameters P, L of which are successful.
2. Track each successful for the M values.

Cost $P + L \cdot M$.

NB This won't always work, so may need to redo step 1.

Problem with standard NAG[1] software, which has parallelization within runs, not between runs. Failure mitigation was not automated, and the startup costs weren't automated.

Therefore `www.paramopy.com`. Written C++/Boost/ Bertini (as a subroutine). Very interactive. Asks what to do about failed points, for example.

Gunawardena lab observed multi-stability in biochemical systems. But Michaelis–Menton very rarely exhibits this sort of multi-stability. Whether there is multi-stability depends on the reaction rates. Running experiments to determine rates is expensive, so let's do homotopy instead. Reduces 13×13 system to 2×2 over 8-dimensional parameter states. Brake looked at 13×13 system and found 2 out of 200,000 points of multistability in parameter space. Therefore lots of questions about the points of multistability in parameter space. Do some Monte-Carlo sampling. Then used VEGAS adaptive sampling. There seems to be one large component (17,000 samples) and a large number of small ones (≤ 20). We can show it's not convex.

Q What do failures tell you?

A Discriminant locus.

Q What does this 8-D space look like?

A I can tell you some properties,but I can't describe it!

4.4 Hom4PS 3.0

Note that $t = 1$ is the target for me, not $= 0$. [HuberSturmfels1994]. assume two trinomials in two variables, with generic coefficients. Multiply t^{ω_i} into each monomial. Problem with $t = 0$. Do a change of variables: $x_i = y_i t^{\alpha_i}$.

Efficiency? Parallelism? Scale? hows cyclic–15, with almost perfect scaling up to 64 cores. Based on version 2,but written in C++., supports multi-precision f.p.

Q Licence?

A None yet, I'll be showing the team yesterday's talk (Section **??**)

4.5 Bertini_Real: software for real algebraic sets

It's compiled command-line software using Bertini. Produces a cellular decomposition of real algebraic components. Using MatLab for symbolic/visual computation. Can decompose higher-level objects. Suppose $F : \mathbf{C}^N \to \mathbf{C}^n$.

[1] In this context, "Numerical Algebraic Geometry", not "Numerical Algorithms Group".

1. Find the critical points. Since f is one-dimensional, consider [f,linear]. Then use regeneration.
2. Intersect with sphere with (large) sphere, to determine components that go to infinity.
3. Slice (through critical points, and mid-slices as well).
4. Connect the dots
5. Merge (dots from above that we no longer need topologically, having connected components)
6. Refine to get decent-looking curves.

For a surface decomposition.

1. Find the critical curves. This is the most challenging part. End up with Jacobians of parametrised versions of Jacobians [JHD couldn't take down the formulae].
2. Decompose singular curves
3. Intersect with sphere with (large) sphere, to determine components that go to infinity.
4. Slice
5. Connect the dots (track the midpoints of the midslices.
6. Refine to get decent-looking images.

How to make a 3D print:

Run Bertini

Run Bertini_real

Refine

Make `.stl` via a MatLab tool.

Thicken firstly via MatLab, then Python's blender

Print

Q Can you tell a collection of curves from a genuine surface?

A Good question!

Q What happens if you move to the radical zero-set?

A Discussion abut symbolic/numeric pro/con.

Q Running time? Longest step?

A Example — 20 seconds, which some is calling MatLab. Critical points of critical curves is by far the longest.

4.6 Quantifier Elimination Software based on comprehensive Gröbner systems

Quantifier Elimination: $\exists x \in \mathbf{R}(x^2 + ax + b = 0$ is $a^2 - 4b \geq 0$. Note over $\mathbf{C}$ it's `true`. Over $\mathbf{C}$ we have GCD-C-QE, Regular Chains and CGS-C-QE. Also over $\mathbf{R}$. [Wei92, SS01, KSW10, Nab07], [Nabeshima2012a]. We use the last.

Variables $x_1, \ldots, x_n$, parameters $a_1, \ldots, a_m$. Let m_h be a linear map $A \to A$ and B a bilinear form $A \times A \to \mathbf{R}$. Let $\chi)Y)$ be the characteristic polynomial of N. C_+ number of sign changes of $\chi(Y)$, and C_- of $\chi(-Y)$. We introduce Hybrid-C-QE.

For inequalities $q_i \neq 0$, introduce new z_i and $1 - z_i q_i$. Use [Kal97] to say $G(\overline{a})$ is a Gröbner basis when $\overline{a} \in S$. This works over $\mathbf{C}$. JHD:only for a conjunction of equalities and inequations.

Over R, where we have $r_i > 0$, we add z_i and $1 - z_i^2 r_i = 0$, similarly for $\geq$. Again only for a conjunction of ground terms.

Has implemented the complex case in in RisaAsir, the real case in Maple.

Claims that for

$$\exists x, y, z(xy + axz + yz = 1) \wedge \ldots \wedge \ldots$$

no existing package terminates in a day.

However, we do not have simplification over $\mathbf{R}$ yet.

Q–JHD Can you deal with more complex formulae.

A "step by step" (needs discussing)

Q What about using the Gröbner cover?

A Discuss.

Q Suppose the ideal is not zero-dimensional?

A We transform it [Possibly making variables into parameters]

4.7 Iwane

Fujitsu: trying to solve optimisation problems. Want the output to be a feasible region, or true/false if all variables are quantified. We have

- General Quantifier Elimination By CAD: $O(2^{2^n}$. [Col75, CH91]
- Virtual Term Substitution for linear/quadratic formulae: $O(2^k)$
- Quantifier Elimination by SH sequences.

SyNRAC — our code: Symbolic–Numeric toolbox for Real Algebraic Constraints. It is a Maple toolbox written in Maple and C. Applications such as parametric optimisation

Problem 4 *Minimise $f(x)$ subject to $C(x)$. $\exists x(y = f(x) \wedge C(x))$ gives the feasible region for y.*

Problem 5 *Minimise $f(x, \theta)$ subject to $C(x, \theta)$. $\exists x(y = f(x, \theta) \wedge C(x, \theta))$ gives the feasible region for y.*

Used this for shape design of air-bearing surface of a HDD. need to make the flight height as close as possible while roll etc. are within bounds. Took 10days to two weeks down to one day.

used dynamic evaluation (only requires square-freeness) and avoids expensive factorisation. Also use validated numbers. By bounded CAD we can avoid needless constructions.

[AnaiHara1999] looks at $\forall x(x \geq 0 \rightarrow f(x) > 0)$. Note that we can compute the (parametric) SH sequence once, independent of where we're evaluating. Major savings.

4.8 Software using the Gröbner cover for geometrical local computation and classification: Montes

The Singular `grobcov` library. Standard Dynamic Geometry Systems(DGS), such as Geogebra, aren't able to determine the locus algebraically. . The Gröbner cover is the equivalent of a reduced Gröbner basis for a parametric ideal. For a fixed term order, it is a canonical object. See [Wib07, MW10]. We also use [KSW10].

$F = \{f_1, \ldots, f_s\} \subset \mathbf{Q}[u][x]$. We get a unique canonical partition of $\mathbf{C}^m$ into locally closed sets (segments), with associated generalized reduced Gröbner bases. The P-representation of S is

$$Prep(S) = \{\{p_i, \{p_{i,j}\}\}$$

where $\forall j, p_i \subset p_{i,j}$. The B_i are given by monic I-regular functions over S_i. an I-regular function allows a generic and a full representation. The generic representation is given by a single polynomial that specialises well ...

4.8.1 Locus

[Abanadesetal]. We consider a *tracer point* $P(u_1, u_2)$, whose coordinates are deemed to be parameters, and we want the points of the parameter space where there are solutions.

Let π_1 and π_2 be the projections into variable and parameter space. Normal locus points are the points n parameter space which correspond to 1 (or a finite number) of solutions ($u \in \mathbf{C}^2 : \dim(\pi_2(\mathbf{V}(F)\ldots) = 0)$, whereas non-normal is those with infinitely many. Hence, given a Gröbner cover, we can split the components according to this.

Example 19 (Pascal Limaçon) $F = \{y_1^2 + y_2^2 = 4, (2 - y_1)u_1 + y_1(_2{-}2) = 0,)u_1 - y_1)^2_(u_2 - y_2)^2 = 1\}$.

Example 20 (Offset of a circle) *Consider the locus of an offset 1 of a circle of radius 1. Note we have the outer circle, and also the accumulation point in the centre.*

Example 21 (Steiner–Lehmus) *Groebner Cover identifies various multiple points, which are self-crossing points of the locus, which need to be "smoothed out" into the drawn locus.*

Q Why Gröbner cover, not Comprehensive Gröbner Basis?

A It's canonical.

4.9 Using Maple's RegularChains Library to automatically classify plane geometric loci: Botana

Application is dynamic geometry. Shows Limaçon of Pascal in GeoGebra. We propose a taxonomy of the spaces. I use `crc:=ComplexRootClassification` from`RegularChains`, then

```
map(x->Info(x[1],R),x[2]],crc)
```

Note that Gröbner Cover doesn't solve the sliding ladder problem, but RegularChains can.

Note that GeoGebra is free, and can use Singular for free via Sage, so RegularChains's dependence on Maple is a financial problem for us in Spain.

Q-speaker Why do have superfluous conditions?

A-Moreno Maza Should be using Real....

4.10 Solving Parametric Polynomial Systems via `RealComprehensiveTriangularize`: Moreno Maza

As well as `Info`, note `Display`. Shows an example of looking for bistable states of a chemical system, by asking `RealComprehensiveTriangularize` for the parameter space over which there are two solutions (two calls: once to build the structure and one to extract). `RealComprehensiveTriangularize` provides a case discussion of the number/dimension of solutions, depending on parameter values. To specialise well, the initials must not vanish, and remain invertible.

But in fact we want the chain to be squarefree (DSCTD), and the specialisation to remain squarefree. Then above each cell of a DSCTD.

- there are no solutions

- There are a finite number of solutions, continuous functions of the parameters.
- Infinitely many, of constant dimension.

Algorithm

1. Compute a DSCTD
2. Refine each cell into connected semi-algebraic sets via CAD
3. Count the solutions.

Example 22 (Mad Cow disease) S_1 *describes all equilibria*

... *describes the case of two stable equilibria*

...

Q I don't understand what the output is?

A The first list is a description of the cells in parameter space, the second list is a (matching) description of the solutions.

4.11 A Package for Parametric Matrix Computation: Thornton

Based on RegularChains. Note that, say, rank is not a continuous function of the parameters. Consider JCF of $\begin{pmatrix} \alpha & 0 & 1 \\ 0 & \alpha - 3 & 0 \\ 0 & 0 & -2 \end{pmatrix}$. Both systems Maple/Mathematica give 3 1-blocks. Sage refuses.

1. Define the polynomial system
2. Call triangularize
3. Do various set differences

Example of a previous paper,which had misquoted conditions for rank degeneracy.

4.11.1 Zigzag form

See [Sto98]. Define a Frobenius companion matrix: Let B_b be a matrix with b in $(1,1)$, 0 elsewhere. A Zigzag form is block diagonal apart from B_{b_i} Then we modify [Sto98] to split whenever ranks are not constant.

Aim to produce `ParametricMatrixTools`. We want to minimise unnecessary splitting.

Q Gaussian elimination/triangular form?

A Yes, same technique as rank.

Chapter 5

9 August 2014

5.1 Computer Discovery and Visual Theorems in Mathematics: J. Borwein

Started to decode Ramanujan's notebooks. Wanted to Now at Computationally-Assisted Research in Mathematics and its Applications: CARMA. This talk is an extract from a three-hour presentation.

Shows 100 billions digits of π, with two bits meaning left/right/up/down (it needed this many to get return to origin!): 108GB, hosted on a commercial service as their first science[1]. Using techniques borrowed from meteorologists to store/visualise data.

> The Computer as Collaborator (alternative title of this talk)

As a collaborator, it should know something I don't,but I should also get along with it.

> The object of mathematical rigour is to sanction and legitimize the conquests of intuition, and there was never any other object for it: Hadamard

Also Littlewood on pictures.

5.1.1 Visual Theorems

De Morgan's quote on [futility of] perspective drawings. See ICMI Study 19 [HdV08]. Shows visualisation of matrices: Hilbert and random. Also demonstrations of values of numerical Hilbert matrix inverses, and where the errors occur.

My level of understanding of [CLO92] was much enhanced by running the examples in real time (Maple) while reading it, and that was 20 years ago on a laptop. Note also PSLQ [FBA99][2].

[1] Also appeared in Wired UK, August 2014

[2] Ferguson also sculpted, in bronze, the Borwein award for the CMS.

5.1.2 Case Studies

Proteins NMR can discover a subset of inter-atomic distances without damage. Throw away all distances over 6A. This becomes a low-rank Euclidean distance matrix computation. For 1PTQ, get a perfect result after 5,000 steps. 1POA fits to -49dB, but looks awful: power of visualisation. Compares (visually) Douglas–Rachford reflection (good) with von Neumann's alternating projection. Why: AP is very good elsewhere (Hubble).

100 digit NT's SIAM News 2002 challenge. Every (human) solver started in one of Maple, Mathematica, MatLab. One problem: maximise

$$I(\alpha) = \int_0^2 (2 + \sin(10\alpha))\, x^\alpha \sin\left(\frac{\alpha}{2-x}\right) dx$$

for $\alpha \in [0, 5]$. Note that $I(\alpha)$ is a Meijer G-function, and Maple/Mathematica know this.

5.1.3 Random walks

[Pearson, nature 1905]. [Rayleigh, next issue] had a large n estimate of the radial density. Venn[1888] drew π (decimal digits, but 0–7 meaning N,NE,E,...).

Has a visualisation of random short walks.

$$W_2 = \int_0^1 \int_0^1 \left| e^{2\pi i x} + e^{2\pi i y} \right| dxdy$$

which both Maple and Mathematica think is zero.

W_3 has a closed form; $\frac{13\sqrt[3]{4}+\cdots}{\cdots} + \frac{\cdots}{\cdots}$. W_4 in terms f G-functions.

$\frac{3624360069}{\cdots}$ etc. have enormous periods [Masaglia2010].

Various results on normality and nonnormality of Stoneham numbers. e.g $\alpha_{2,3}$ is normal base 2 but not base 6. Visualisations are very powerful. Note that the Champernowne number C_{10} is normal,but looks totally patterned.

5.2 Cylindrical Algebraic Decompositions in the RegularChains Library: Moreno Maza

Definition of CAD. [Col75]. New scheme via complex space (`CylindricalDecompose`) [CMMXY09]. Shows case discussion in the case of $\exists x : ax^2 + bc + c = 0$.

Note `CylindricalAlgebraicDecompose(...,output=cadcell)`, and `output=rootof`. Improvements via `optimiztion=EC` or `TTICAD`.

We have now solved the Joukowski problem [DBEW12]. Uses `precondition=true` and `partial=true`. 55 seconds and 21 seconds.

5.3 Choosing a variable ordering for truth-table invariant cylindrical algebraic decomposition by incremental triangular decomposition: Davenport

See `http://staff.bath.ac.uk/masjhd/Slides/JHD-ICMS2014-post.pdf`.

Q–Hong What's the difference between TTI and equational constraints?

A (a) more local; (b) applies even when there's no global equation constraint.

5.4 Using `RegularChains` to do Projecting/Lifting CAD; England

Available from Bath, and aim to integrate into `regularchains.org`. Used as a research tool, let to TTICAD in PL, which led to TTICAD for RC. Using many routines from the `RegularChains` library.

One difficulty is that `RegularChains` assumes that,in addition to delineability, the polynomials *separate* above each cell: use `Triangularize` here, and Rong Xiao's adaptation of Musser's squarefree algorithm.

Sign-Invariant CADs with Collins or McCallum

Order-Invariant

Brown's "minimal delineating polynomials" (not actually in QEPCAD)

Equational Constraints as in [McC99], but with improved lifting as well.

Truth Table-Invariant As in [BDE+13]. Note that their applications of TTICAD which aren't just truth invariance: simplification, motion-planning. See also BranchCuts in `FunctionAdvisor` [ECTB+13].

Layered sub-CAD either directly or incremental. In particular full-dimensional never involves algebraic numbers.

Variety sub-CAD Smaller output, and potential time savings.

Combinations of all the above.

Q-Moreno Maza You said you don't need cylindricity for the branch cut application? What about `RealTriangularize`

A Possibly, but does that handle TTI-options

Moreno Maza Not yet: we should talk.

5.5 Hierarchical Comprehensive Triangular Decomposition: Tang

Note that Gauss triangularises a linear system. What about nonlinear? In general, will work, and a smooth change of coefficients doesn't change things. But there are special cases. Existing CTD

1. Full solution in all variables/parameters
2. Partition parameter space into constructible sets
3. Find the TD over each (it wasn't clear to JHD how much re-computation was required).

So we introduce a new algorithm.

1. Compute a TD in variable space only, tracking side-conditions.
2. Add one side-condition, and its parameter as a variable.
3. repeat as necessary.

In practice this is 1.3-2 times slower on one set, but 4–5 times faster on another set. For hard examples, when CTD doesn't terminate, we at least get the generic case.

Q-Hong Can you re-use the computations from one TD to the next?

A In some examples we can.

5.6 Doing Algebraic Geometry with the `RegularChains` library: Moreno Maza

Our driving application will be intersection multiplicity. Let $f_1, \ldots, f_n \in k[x_1, \ldots, x_n]$ such that $\mathbf{V}$ is zero-dimensional. Want to compute the intersection multiplicity $I(p; f_1, \ldots, f_n)$ for $p \in \mathbf{V}(f_1, \ldots, f_n)$. In the projective plane, this specifies the weights for Bezout's theorem. It is not natively computable by Maple. Singular and Magma only do it for points in k.

5.6.1 Two plane curves

Definition 11 *The* intersection multiplicity *of* p *in* $\mathbf{V}(f, g)$ *is*

$$I(p, f, g) = \dim_{\overline{k}}(\mathcal{O}_{\mathbf{A}^2,p}/\langle f, g\rangle)$$

1. We set $I = \infty$ of there's a common factor
2. $I(p; f, g) = 0$ iff

3. $I(p; f, g)$ is invariant under affine change of coordinates

4. $I(p; f, g) = I(p; g, f)$

5. Multiplicative if factorisations of f and g.

Gives an algorithm for this due to Fulton, which doesn't work in more than 2D, since $k[x_1, \ldots, x_{n-1}]$ is no longer a PID, also assumes p defined over base field. We aim to relax the second definition, and give a recursive algorithm in n.

We need the multivariate/regular chains version of the D5 principle.

Theorem 9 *If $\langle T \rangle$ is maximal, then $I(p, f_1, f_2)$ is the same at any point $o \in \mathbf{V}(T)$.*

New command `TriangularizeWithMultiplicity`. Demonstrated on an example (working modulo 101 for simplicity).

Theorem 10 *Assume that $h_n = \mathbf{V}(f_n)$ is non-singular at p. Let v_n be its tangent hyperplane. assume it meets each component transversally. Then $I(p; f_1, \ldots, f_{n-1}, f_n) = I(p; f_1, \ldots, f_{n-1}, h_n)$, and therefore reduced to $n-1$ dimensions.*

Need the non-trivial limit points, via Puiseux series of regular chains.

5.7 Computing Moore-Penrose inverse: Zhang

Differential Operators can be multiplied. Hence we need Ore polynomials, with $\delta(a+b) = \delta(a) + \delta(b)$ and $\delta(ab) = \sigma(a)\delta(b) = \delta(a)b$. Maple's `Ore` package.

Want to work in $R[x; \sigma, \delta]^{n \times m}$ of matrices over an Ore ring. Note the quaternions $\mathbf{H}$.

We can have quaternion polynomials, where x commutes with the quaternions. $A^\ddagger$ is the Moore–Penrose inverse of $A \in \mathbf{H}[]^{m \times n}$ if it solves $AXA = A; XAX = X, (AX)^* = AX, (XA)^* = XA$.

Lemma 2 *If $A \in \mathbf{H}[x]^{m \times n}$, then the eigenvalues of AA^* are real.*

5.8 Multivariate Birkhoff Polynomial Interpolation

The orders of the derivative conditions at some nodes are discontinuous. Need the "lower set" of a staircase.

Q-Moreno Maza Real/complex?

A Use `Triangularize`, and then `RealRootClassification` for the reals.

5.9 An Improvement of the Rosenfeld–Gröbner Algorithm: Hashemi

Hakim Pmar Khayyám Nieshapuri wrote *Demonstration of Problems of Algebra* (1070) which described how to solve the cubic, and the triangular array of binomial coefficients..

Aim: to bring algebraic solutions to PDEs. See [Rit50]. Define $\prec_{lex}$ and $\prec_{drl}$ as orders.

Definition 12 *For polynomials G is a* Gröbner Basis *if* $\mathrm{lm}(G) = \mathrm{lm}((G))$.

Define differential ring (possibly partial). Differential ideal. For $\theta = \delta_1^{m_1} \cdots \delta_m^{m_m}$ write $\mathrm{ord}(\theta) = \sum m_i$. Polynomial differential ring $K\{u_1, ldots, u_n\}$

Definition 13 $>$ *is a* ranking *if* $\forall \delta \in \Theta$, $\forall v, w \in \Theta U$*:* $\delta v > v$ *and* $v > w \Rightarrow \delta v > \delta w$.

Define leader, and separant of p $= \partial p \partial \,\mathrm{leader}(p)$.

Consider two differential polynomial p_1, p_2 with leaders $\mathrm{leader}(p_i) = \theta_i u_i$. Then

$$\delta(p_1, p_2) = \begin{cases} s_{p_2} \frac{\theta_{1,2}}{\theta_1} \cdots & ?? \\ 0 & otherwise \end{cases}$$

Hence Rosenfeld–Gröbner,which constructs Δ-polynomials, and reduces them differentially as well as algebraically w.r.t. the current set. **But** we need to discuss the nullity of initials. (More like Comprehensive Gröbner Basis).

See [CarraFerroOllivier1987], Mansfield's work, [Boulier1991]. Boulier proved this for ordinary differential rings, and we have this extension.

Theorem 11 (Generalisation of Buchberger's gcd criterion) *Suppose* $\mathrm{leader}(p) = \theta u$ *and* $\mathrm{leader}(q) = \phi u$ *and* $\mathrm{lcm}(\theta u, \phi u) = \theta \phi u$*. Then* $\Delta(p, q)$ *reduces to zero.*

5.10 Bertini for Macaulay2: Rodriguez

`needsPackage(Bertini)` is all that is needed. Shows `bertiniZeroDimSolve`, which also has condition numbers etc., and lots of bertini-internals. Also `bertini PosDimSolve`, which gives the components of the "numerical variety".

Also `bertiniParameterHomotopy`: see Section 4.3.

Q–Hong What was the hardest part?

A Interfacing all the types of numbers.

Q This was all Macaulay calling Bertini: reverse direction?

A That was the demo I didn't get to!

5.11 Using Monodromy to avoid high precision: Niemerg

"Consider the Latin[3] definition of monodromy": running round once. Multi-precision involves moving from hardware to software.

Define a path segment P to be a triple $(p, t_{...}, t_{...})$. Consider two containers P_{forward} and P_{backward}, with these rules.

1. Always finish P_{backward} first
2. Always use the same loop after you've made one trip.

This gives us a heuristic process

1. Create your favourite start system and populate P_{forward} with the start solutions.
2. perform normal tracking.
3. If you hit a trigger (condition number etc.) then construct an *ad hoc* monodromy loop.
4. Follow the rules.

But we can hit triggers as we move along a path, and forward and backward tracking on the same path can have different size loops. JHD: seems to be "work in progress", and there was a lot of possible directions based on a complicated diagram.

Q Does this pay off?

A Good question. It does in our examples.

5.12 Software for MKM: Ion

MSC [Mathematics Subject Classification] and the underlying concept has a long history: Leibniz, Dewey, LoC, etc. MSC 2010 exists: 5606 leaf nodes. MathSciNet indexes 3M publication, by 720K authors. This is now an RDF linked dataset using SKOS (Subject Knowledge Organisation System) as the vocabulary. LoC uses SKOS, with 250K leaves. This allows multilingual support with Unicode. This has three kinds of links: "see also", "see mainly" and "for XX see".

There is a SPARQL interface.

[3] JHD did a little research, which he discussed afterwards with the speaker. The etymology is clearly Greek, but in fact, according to the Oxford English Dictionary, the work itself was coined by Cauchy. They quote [Cau47, p.352], which does indeed look like a coinage: "Nous les appellerons *monodromes*" [original emphasis] and "le mot monodrome paraît bien exprimer assez bien la propriété de la fonction que l'on considère, puisque celles-ce varie par degrés insensibles, en acquerant à chaque instant une *valeur unique*, tandis que le point mobile A correspondant l'affixe z *court* ça et là, ...".

Q–Hong What is a subject?

A Shows 11G as an example. This is a list of topics, others might have methods, or problems.

Bibliography

[BDE+13] R.J. Bradford, J.H. Davenport, M. England, S. McCallum, and D.J. Wilson. Cylindrical Algebraic Decompositions for Boolean Combinations. In *Proceedings ISSAC 2013*, pages 125–132, 2013.

[BDH+14] D.J. Bates, W. Decker, J.D. Hauenstein, C. Peterson, G. Pfister, F.O. Schreyer, A.J. Sommese, and C.W. Wampler. Comparison of probabilistic algorithms for analyzing the components of an affine algebraic variety. *Applied Mathematics and Computation*, 231:619–633, 2014.

[BHSW13] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Numerically Solving Polynomial Systems with Bertini. *SIAM Press*, 2013.

[Cau47] A.-L. Cauchy. *Exercices d'Analyse et de Physique Mathématique IV*. Bachelier, Paris, 1847.

[CH91] G.E. Collins and H. Hong. Partial Cylindrical Algebraic Decomposition for Quantifier Elimination. *J. Symbolic Comp.*, 12:299–328, 1991.

[CLO92] D. Cox, J. Little, and D. O'Shea. Ideals, Varieties and Algorithms: An Introduction of Computational Algebraic Geometry and Commutative Algebra. *Springer-Verlag*, 1992.

[CMMXY09] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing Cylindrical Algebraic Decomposition via Triangular Decomposition. In J. May, editor, *Proceedings ISSAC 2009*, pages 95–102, 2009.

[Col75] G.E. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

[DBEW12] J.H. Davenport, R. Bradford, M. England, and D. Wilson. Program Verification in the presence of complex numbers, functions

with branch cuts etc. In *Proceedings SYNASC 2012*, pages 83–88, 2012.

[DH88] J.H. Davenport and J. Heintz. Real Quantifier Elimination is Doubly Exponential. *J. Symbolic Comp.*, 5:29–35, 1988.

[ECTB+13] M. England, E. Cheb-Terrab, R. Bradford, J.H. Davenport, and D.J. Wilson. Branch Cuts in Maple 17. `http://arxiv.org/abs/1308.6523`, 2013.

[EH13] M. Erașcu and H. Hong. The Secant-Newton Map is Optimal Among Contracting Quadratic Maps for Square Root Computation. *J. Reliable Computing*, 18:73–81, 2013.

[FBA99] H.R.P. Ferguson, D.H. Bailey, and S. Arno. Analysis of PSLQ, an Integer Relation Finding Algorithm. *Math. Comp.*, 68:351–369, 1999.

[Gon12] Gonthier,G. *et al.* The formalization of the Odd Order theorem has been completed on September 20th, 2012. `http://www.msr-inria.inria.fr/Projects/math-components/feit-thompson`, 2012.

[Hal12] T.C. Hales. Dense sphere packings: a blueprint for formal proofs. *Cambridge University Press*, 2012.

[HdV08] G. Hanna, , and M. de Villiers. ICMI study 19: Proof and proving in mathematics education. *ZDM Mathematics Education*, 40:329–336, 2008.

[Hof89] C.M. Hoffmann. The Problems of Accuracy and Robustness in Geometric Computation. *IEEE Computer 3*, 22:31–39, 1989.

[Kal97] M. Kalkbrener. On the stability of Gröbner bases under specializations. *J. Symbolic Comp.*, 24:51–58, 1997.

[KSW10] D. Kapur, Y. Sun, and D. Wang. A New Algorithm for Computing Comprehensive Gröbner Systems. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 29–36, 2010.

[McC99] S. McCallum. On Projection in CAD-Based Quantifier Elimination with Equational Constraints. In S. Dooley, editor, *Proceedings ISSAC '99*, pages 145–149, 1999.

[MW10] A. Montes and M. Wibmer. Gröbner bases for polynomial systems with parameters. *J. Symbolic Comp.*, 45:1391–1425, 2010.

[Nab07] K. Nabeshima. A speed-up of the algorithm for computing comprehensive Gröbner systems. In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 299–306, 2007.

[OLBC10] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, and C.W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, 2010.

[Rab13] F. Rabe. The MMT API: A Generic MKM System. In J. Carette *et al.*, editor, *Proceedings CICM 2013*, pages 339–343, 2013.

[Rit32] J.F. Ritt. *Differential Equations from an Algebraic Standpoint*. Volume 14. American Mathematical Society, 1932.

[Rit50] J.F. Ritt. *Differential Algebra*. Colloquium Proceedings vol. XXXIII. American Mathematical Society, 1950.

[SBB+13] V. Stodden, D.H. Bailey, J. Borwein, R.J. LeVeque, W. Rider, and W. Stein. Setting the Default to Reproducible: Reproducibility in Computational and Experimental Mathematics. `http://icerm.brown.edu/html/programs/topical/tw12_5_rcem/icerm_report.pdf`, 2013.

[SS01] Y. Sato and A. Suzuki. Discrete Comprehensive Gröbner Bases. In B. Mourrain, editor, *Proceedings ISSAC 2001*, pages 292–296, 2001.

[ST11] Thomas Sturm and Ashish Tiwari. Verification and synthesis using real quantifier elimination. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, pages 329–336. ACM, 2011.

[Sto98] A. Storjohann. A $O(n^3)$ Algorithm for the Frobenius Normal Form. In O.Gloor, editor, *Proceedings ISSAC '98*, pages 101–104, 1998.

[Vic61] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16:8–37, 1961.

[Wei92] V. Weispfenning. Comprehensive Gröbner Bases. *J. Symbolic Comp.*, 14:1–29, 1992.

[Wib07] M. Wibmer. Gröbner bases for families of affine or projective schemes. *J. Symbolic Comp.*, 42:803–834, 2007.

[Wu87] W.-T. Wu. A Zero Structure Theorem for Polynomial Equations Solving. *MM Research Preprints*, 1:2–12, 1987.