CASC/SYNASC 2016

JHD

19–27 September 2016

Contents

Chapter 1

19 September 2016

1.1 Computing Sparse Representations of Systems of Polynomial Fractions — Lemaire

Motivation 1

Hence want a change of basis matrix N and a vector v to make $\overline{N} := CN + VP$ as sparse as possible.

Theorem 1 (Structure Theorem) *If F and $\overline{N}$ are row-equivalent to same matrix, and have same sparsity with rows sorted in order f decreasing sparsity, Then the matrices N_i and $\overline{N}_i$*

Conclusion: it's really a naming problem, but it's an important one. Note that output is not unique, and there's a choice here as well.

Q Might you lose, or gain, singularities?

A Possibly? (JHD didn't follow)

1.2 Arithmetic Computation of Polynomial Amoebas: Bogdanov

Let $p(x_1, \ldots, x_m) = \sum_{\alpha \in A} c_\alpha \mathbf{x}^{\mathbf{alpha}}$.

Theorem 2 (Fursbergetal2008) *Let $\{M\}$ denote the family of connected components of the amoeba complement ${}^cA_{p(x)}$. Then ...*

3D amoebas computing by fixing the absolute value of z and then taking the 2D amoeba as a section.

1.3 Sparse Gaussian Elimination mod p — an Update: Delaplace

Direct Gaussian elimination etc.: memory requirement uncertain. Generally faster when they terminate.

Iterative Wiedermann etc. — needs two vectors only.

See [**?**] and much work in the numerical world. [Davis2006Book]. Typical direct method is PLUQ, where P, Q are permutation matrices. Usually "right-looking", but also left-looking $\begin{pmatrix} L \setminus U & A \\ L & A \\ L & A \end{pmatrix}$

New algorithm GPLU, not used in Exact Linear Algebra. Works best when U is very sparse. Often outperforms right-looking, and occasionally vice versa.. Can we combine them?

So we have a heuristic [Faugere etc.] for finding the pivot. Each row is mapped to the column of its left-most coefficient., If several, use the sparsest. Move these rows to the top of the matrix, sorted by increasing position of leftmost coefficient. Then compute the Schur Complement.

$$PA = \begin{pmatrix} U_{00} & U_{01} \\ A_{10} & A_{11} \end{pmatrix} = \begin{pmatrix} Id & \\ L_{10} & L_{11} \end{pmatrix} \cdot \begin{pmatrix} U_{00} & U_{01} \\ & U_{11} \end{pmatrix}$$

On four examples from Dumas' collection, it always outperforms right-looking LinBox, and better or equal to GPLU. On (different!) examples, it always outperforms Weidemann.

Q Why p?

A Just to have small exact computations.

1.4 Lagrange: Stefănescu

Lagrange had two bounds. $P = a_o X^d + \cdots + a_m X^{d-m} - a_{m+1} X^{d-m-1} \pm \cdots$ when all $a_i \geq 0$. Then let $A := \max\{a_i : coeff(X^{d-i}) < 0\}$ then $1 + \left(\frac{A}{m}\right)^{??}$ is a bound.

Second bound "$R + \rho$": let $\{a_i : j \in J\}$ be the set of negative coefficients. Then the sum of the largest and second-largest in $\{\sqrt[j]{|a_j|} : j \in J\}$ is an upper bound for the positive real roots of F. There is a proof [Pury1842], [Westerfield1931]. [**?**].

Theorem 3 (Kiousteldis) *Let* $P(X) = X^d - b_1 X^{m_i} \ldots$

Theorem 4 (Stefănescu) *Let P be such that the number of sign variations in the coefficients is even. ....*

Note that there are now many ways of proving Lagrange, generally using Cauchy's method.

Theorem 5 ([?]) *Let σ be the unique positive root of $X_n - |a_1|X^{n-1} - \cdots - |a_n|$. Then any number greater than σ is a bound for the moduli of the roots of the original polynomial.*

Though Lagrange stated his results fro real roots, actually true for complex roots as well.

Theorem 6 (Fujiwara1962) *If $\sum \frac{1}{\lambda_i} = 1$*

Theorem 7 *Let the $i_1, \ldots$ be such that $|a_{i_1}|^{1/i_1} \geq |a_{i_2}|^{1/i_2} \geq \cdots$*

Theorem 8 (Hong) *$H(P) := 2 \max_{i:a_i<0} \min_{j>i;a_j>0} \left(\frac{|a_i|}{a_j}\right)^{1/(d_i-q_j)}$ is a bound*

Theorem 9 (MS2015)

$$|z| \leq \max\left\{C_j, \frac{R+\rho+\sqrt{R^2-2R\rho+5\rho^2}}{2}\right\}$$

Theorem 10 *Largest root of $H(x) = x^3 - 2\rho x^2 - (R^2-\rho^2)x + \rho(R^2-\rho^2)$ is a root bound.*

Special case when $R < \frac{3}{2}\rho$

Q–JHD What happens when $R = \rho$.

A Then we get $2R$, and this is what Theorem 10 collapses to.

1.5 A Symbolic Investigation of the influence of Aerodynamic Forces on Satellite equilibria: Gutnik

Cosmos-149 had $h_a \approx 297km$, $h_p \approx 248km$, $i = 48°$. PAMS launched from Space Shuttle in May 1996 has similar characteristics.

Let p, q, r be the projections of the satellite's angular velocity in Ox, Oy, Oz. A, B, C principal central moments of inertia. α, β, γ are pitch, yaw, roll. a_{ij} .... Assuming α etc. are constant, then we have equilibrium equations.

Uses FGLM(plex) to get 8 solutions in the general case, but there are situations with 4. Working pack to satellite equilibria, there are 24 in $h_1^{2/3} + h_2^{2/3} < v^{2/3}$, 20 in range $v^{2/3} < h_1^{2/3} + h_2^{2/3} < 3v^{2/3}$, then 16 or 12. We have looked at the bifurcation points.

Chapter 2

20 September 2016

2.1 Sommerse

Example: cyclic n-th roots.

Definition 1 *Newton polytope is the convex hull of the support of a polynomial.*

Definition 2 *The* normal cone *of a face F of a polytope P is the convex cone generated by all the facets ?normal to? F, and the* normal fan *of a polytope is the union of all the normal cones of all the faces.* Normal refinement *of two fans F_i is* $\{C_1 \cap C_1\}_{(C_1,C_2)\in F_1\times F_2}$

Given an input set of polytopes, take the common refinement of all their associated fans, considering only normal cones to faces of dimension ≥ 1. Pretropisms are the generating rays.

Example 1 *F_1 = square, F_2 = hexagon: naïvely, $4 \times 6 = 24$ intersections. Their algorithm, essentially walking round the two polygons, will only do $3 + 2 + 3 + 2 = 10$.*

Problem 1 *If we have more than two polytopes, we can get duplicate cones, and also cones contained in other cones. Solution is "on-the-fly" pruning.*

Prototype in SAGE/PPL (paper), but now a C++ version. Definition of pretropism took 58M intersections and 9 hours [SAGE], pruning took 40K intersections and 9.43 seconds for cyclic-8.

2.2 Computing all space curve solutions: Bliss

Goal is to use polyhedral methods to compute all space curves of polynomial systems. Compute - (truncated) tuple of power series satisfying equations. Assume the curve is in Noether position, i.e. specialising the first variable generically yields deg(curve) points. See [AdrovicverscheldeCASC2013].

Example 2 (Viviani's curve) *Intersection of sphere and cylinder.* $F := x^2 + y^2 + z^2 - 4$, $g := (x-1)^2 + y^2 - 1$.

Newton–Puiseux method.

1. Find exponent tuple λ using Newton polytope
2. Find coefficients $\mathbf{c}$ via vanishing equations
3. Repeat after substituting

In example, $\lambda = (2, 1, 0)$ is normal to an edge of each polytope (see previous talk).

Definition 3 *The* tropical prevariety *of a system F is the set of all rays $v \in \mathbf{R}^n$ such that $in_v(p)$ is not a monomial for every $p \in F$. Such a ray is called a* pretropism. *The* tropical variety *is the same for $p \in \langle F \rangle$.*

Theorem 11 *For n equations in $n+1$ unknowns with generic coefficients, the tropical variety is contained in the set of ray generators of the tropical prevariety, i.e. its 1-skeleton. See [HuberSturmfels1995], [Bernstein1975].*

But this is only generic. In general rays from the tropical variety can hide in the higher dimensional cones of the prevariety. Also has an example (not in paper) where the Noether position assumption fails.

So for each cone in the prevariety, check if its initial ideal contains a monomial, and if necessary introduce a witness polynomial to this fact. Uses Jensen's `Gfan`, but is expensive.

2.2.1 Polyhedral end games

This is an alternative. [Huberverschedle1998]. $f(x) = 0; tx_1 + (1-t)(x_1 - \gamma) = 0$ with $\gamma \in \mathbf{C} \setminus \{0\}$ and take the differences as $\log|x|$ runs along the solutions path.

2.3 Hahn

See "Computer Algebra in Theoretical Physics" — this CASC 2016 Invited talk. Shows a graph from hep-ph/9704332. Uses CA nontrivially to improve data from a mess to a reasonable error spread.

- How do we describe elementary particles
- How can we make and test theoretical predictions?

Theorists like $\mathcal{L}$: "the model". But experimentalists like $\mathcal{S}$, hence cross-sections, decay routes etc. $\mathcal{L}$ is usually a polynomial in quantum fields. In Field Theory, force is the exchange of a "force particle", which automatically fulfils SR. In a Lagrangian, bilinear terms describe propagation and multilinear terms describe interaction. $M = \langle out|\mathcal{S}|in\rangle$, and is $\sum$ Feynman diagrams. These are perturbation series in the coupling constant $\sqrt{\alpha}$. Feynman rules follow from the Lagrangian.

Standard model $\underbrace{SU(3)}_{StrongForce} \times \underbrace{SU(2)}_{WeakForce} \times \underbrace{U(1)}_{EM}$

Gravity $GL(4)$

So how do we unify?

Currently (unlike 1960s/70s), discoveries follow from theory: Top Quark, Higgs Boson. But where do all the model parameters come from? No gravity!

No loops is 10% accuracy, and every loop buys roughly $\times 10$ in accuracy. But only for 1 loop do we have a completely general process. We do see higher orders experimentally. Claims that there are indirectly visible effects, such as gluon fusion and flavour observables ΔM_S.

What to do about divergent Feynman integrals?

- Parametrize divergences (most popular; preserves symmetry)
- renormalise theory = subtract UV divergences
- Consider physical observables = cancel IR divergences

1999 Nobel Physics went to??/?? for showing the following process was an algorithm.

1. draw all possible types of diagrams with the given numbers of loops and external legs. Originally by hand, now computer-assisted.
2. Figure out which particles can run on each type of diagram (combinatorics +physics)
3. Translate diagrams into formulae by Feynman rules. This is a (unconventional!) database lookup.
4. Contract the indices, take traces etc. This is an algebraic simplification task
5. Convert into some numeric program
6. Run and publish results

Requires hybrid programming. Note that dimension $\mathcal{D}$ should be treated symbolically. Use packages `FeynArts` (Mathematica package) and `FormCalc` (Form).

The starting (0-leg) topologies are hard-coded, then add legs with some human intervention. It is important to do as much algebraic simplification as possible at the generic level (when there are fewer variables). Needs `NonCommutative` in Mathematica. `FeynArts` has various stock model files.

Need Form (successor to Schoonschip) During the execution of normal statements, terms are only generated, and special "dot" statement actually does the simplification. There's also a Form preprocessor, not unlike C, but more powerful `#do`, `#procedure` etc. There's a facility called "abbreviations" which compartmentalises the numerical computation. These can be expensive to compute, so it is key to separate them into different categories so that these are only evaluated when something changes.

2.3.1 Validating results

1. cancellation of divergences
2. Gauge invariance Compute in an arbitrary gauge and check the gauge parameters cancel.
3. special limits, e.g. high-energy limit for external weak bosons by Equivalence Theorem.
4. comparison with independent calculations. This is "gold-plated" but a lot of work.

Need resummations (`hbb`), approximations, K-factors, nontrivial renormalisation. Software design so far is mostly monolithic, controlled by "parameter cards",which are not easy to use beyond intended purpose. Hence we are reorganising the entire procedure. Implement calculations as several steps, need detailed logs and a `makefile` Current flow is seven steps (plus preparation).

2.3.2 Conclusions

Serious perturbative calculations can no longer be done by hand. Accuracy, #particles etc. Hence we need hybrid programming. More software engineering is a must.

Analogy to driving a car: you need to know where to go (hardest part), turn the car on etc., but don't need to know the detailed operation. However, you can only drive where there's a road! If not, you have to build one.

Q Limiting factors?

A Expressions in Form can be gigabytes, so this is a major factor. The Mathematica code is under development for years, and we have problems with the number of licences as we want to go parallel.

Q What CA tools.

A γ-integrals, but other than that largely just expression manipulation. Largely polynomials (polylogs, but (JHD thinks he meant) these are just treated polynomially.

Feynman diagrams are usually evaluated in momentum space. Momentum conservation fixes all momenta flowing through the tree parts of a Feynman diagram, but for each loop one momentum is unconstrained and has to be integrated over.

2.4 A Numerical Method for Computing Border Curves . . . : Chen

Applications in Robotics, stability analysis of biological systems and model predictive control. From the theoretical point of view, we see Real Numerical Algebraic Geometry, Parametric Polynomial Systems, and Homotopy continuation in $\mathbf{R}^n$. Note that Hoon

Hong (invited talk at MACIS 2013) suggested some possible ideas for doing QE by SN approach.

Assumptions

- $F(X_1, \ldots, X_m, U_1, U_2)$ is a square polynomial system
- ...

Definition 4 *The* border curve *B of F restricted to rectangle R is $\pi(V_{\mathbf{R}}(F, det(J_F))$.*

Given f, the BP/DV of f is defined by the discriminant

1. Let $F' := \{F, det(J_F)\}$, or a better formulation for numerical stability.
2. Trace. Need step size control [WeReidFend2015] and to detect jumping from components.

   Theorem 12 *Then the distance from B to the segment $\overline{u_0 u_1}$ is at most ....*

3.
4.
5.

Example 3 *Degree 24 polynomial with 301 terms. Numerically get a polyline between steps. Error 10^{-4} compared with upper bound 10^{-3}.*

Need the concept of being δ-connected, and a result saying that δ-connected implies connected.

Definition 5 *Consider R as a graph G on an $m \times n$ grid with distance δ. Let B be a polyline. A* connectivity graph *is ...*

Definition 6 *A* solution map *of F restricted to R is a quadruple (B, G, W, Z)*

B set of points approximating the border curve of F

Shows some experimental data: numeric is slower on small examples, but does large examples where symbolic (two methods: Border Polynomial and Discriminental Varieties — the two give mixed performances) fails. #points is $[10^3, 10^5]$. Need to relax assumptions and extend to more than two parameters. Also want efficient implementation when F' has special structures.

2.5 Sparse Interpolation in Hensel Lifting: Tuncer

Multivariate Hensel Lifting [**?**] as used in Maple: three main components.

1. Leading Coefficient Correction (we use Wang's basic idea here: therefore examples all monic)
2. factorization in $\mathbf{Z}[x]$
3. MDP — most expensive in practice.

Sparse — Zippel and then Kaltofen (HL): see paper. Note that the lifting to bivariate is done the usual way: sparsity intervenes later. Let I be the evaluation ideal.

Repeatedly solving $\sigma_2 f_2^{(i)} + \tau_2 g_2^{(i)} = e^{(i)}$. The solutions to these are the Taylor coefficients of the factors, but these are not necessarily sparse even if the answer is sparse.

Problem 2 (MDP) *Find* $\sigma, \tau \in \mathbf{Z}_{p^n}[x_1 \ldots, x_j]$ *such that* $\sigma u + \tau v = c \mod \langle I_j^{d+1}, p^n \rangle$

Wang solves this by lifting as in MHL, but when the evaluation points are non-zero we have a sparsity problem.

We have an interpolation method that fails if a GCD is non 1. Prob(this) is$\leq \deg(u)\deg(w)/p$ so choose a large p. We follow Zippel's sparse interpolation idea, and assume invariance of the skeleton (this is the skeleton of the cofactors as well: ?is this the new idea). Note that we lift (he claims new?) from $\mathbf{Z}_p[x_1, x_2]$ only, as we have an improved evaluation method, and the sparse interpolation linear systems will be smaller.

Example 4 ($n \times n$ cyclic matrices) $n = 11, 12, 13$ *Wang: 9/24/258 seconds, Bivariate sparse 0.5/7/20; SHL 0.35/4.3/14 seconds (for total process: SHL cost dominates for Wang, but not him).*

Example 5 ($n \times n$ Toeplitz) *factors are almost dense, but still see an improvement.*

Claims that Kaltofen only works when very sparse, but even then SHL still wins. Rigorous complexity submitted to JSC (35 pages).

Q–JHD You suggested large p — isn't the base factorisation expensive?

A No - we use 32-bit primes.

2.6 Resolving Decompositions: Albert

syzygy of U is the solution to a linear system over the ring $\mathcal{P} := k[x_1, \ldots, x_n]$. Generate a module $Syz(U)$, and we can iterate: hence a resolution. Minimal resolution is where differentials do not contain any non-zero constants. Bigraded Betti numbers $\beta_{i,j}$. Consider $\beta_{i,i+j}$ as a matrix. Width is projective dimension $pd(U)$ and height is Castelnuovo–Mumford regularity $reg(U)$.

Our goal is an axiomatic framework for generating special sets of the module U, which induces an explicit free resolution, computes Betti numbers etc. For a given $U \subset \mathcal{P}_d^m$, a resolving decomposition consists of

- $B = \{h - 1, \ldots, h_s\} \subset U$ finite
- *head module term* $hm(h)$ for every $h \in B$
-
- $U = \langle B \rangle$
- $h \in B \Rightarrow supp(h) \cap hm(B) = \{hm(h)\}$
- $hm(B) = \bigoplus_{h \in B} k[X_B(h)] \cdot hm(h)$ and $U = \bigoplus_{h \in B} k[X_b(h)] \cdot h$
- staircase property
-

Schreyer theorem for resolving decompositions. A unique standard representation of $x_k h_\alpha$ leads to a syzygy ....

This induced free resolution has in general greater length than the minimal; higher degree bound than minimal; $reg(U) \leq \deg(B)$. Now we need algebraic discrete Morse theory: given a nonminimal free resolution, encode information over nonvanishing entries of the differentials in a graph, find a Morse matching, and get a smaller resolution. Hence [Sköldberg2011] two-sided Koszul complex. This defines a resolution, but modules ave infinite rank. If U has head linear syzygies, there is a five-term exact sequence and we can split into critical and non-critical variables.

For us, resolving decompositions do have head linear syzygies.

Theorem 13 *Sköldberg resolutions are isomorphic to resolutions induced by resolving decomposition.*

To deduce graded Betti numbers, it suffices to know the shape of the free resolution and the "constant" part of the differentials. Sköldberg's theorem gets the constant parts without computing the whole differentials. The induced free resolution gives the shape. Hence we can get the Betti numbers. see 1511.03547.

Chapter 3

SC^2 Special Session: 20 September 2016

3.1 SMT for CA: Monniaux

SAT asks with a Boolean formula is satisfiable. Enumerating 2^n choices is possible. This was (at least one of) the first NP-complete problems, and hence we conjecture there is no polynomial-time algorithm. But this has enormous applications, e.g. in circuit design. In practice, we have programs that work well in practice. Note that for SAT we almost always want the assignment, and for UNSAT we would sometimes like a (short proof). The general input to SAT is CNF. But conversion to CNF naïvely, distributive laws, can cause exponential blow-up. There is a linear transformation [**?**] at the cost of adding new variables.

DPLL: each clause can act as a propagator: in we know the values of all but one, then either the clause is true or we can deduce the value of the last variable. Known as *unit propagation*. As in Sudoku guessing, when we can't propagate, we branch, and effectively make a guess. When we get a contradiction, we backtrack.

CDCL: when we get a contradiction, we track the assumptions that actually led to this decision (not the whole set of assumptions made), and then we know that this subset is infeasible, so we have learned the 'or' of the negations of these assumptions. This is a consequence of our original clauses, but may be helpful.

So pure DPLL is tree resolutions, and CDCL is DAG-resolution, and there is a proof that there are formula which have exponentially large tree resolutions, but polynomial DAG resolutions.

Two highly-optimised engines are MiniSAT and Glucose. Note that pre-processing is also a black art. See Knuth's work (under construction?).

In SAT, all these Boolean literals are independent. But in SMT, we wish to consider cases (conjunctions and disjunctions) where the literals come from some theory (integer arithmetic, real arithmetic).

DPLL(T) (should really be called CDCL(T)).

For LRA, we tend to combine inequalities via the simplex method, and see if we can deduce a contradiction (from a subset, which then forms a theory lemma).

For LIA, we can use LRA, and if this suggests $x = 4.3$, we consider $\leq 4 \vee x \geq 5$.

It is also possible to use Gomory cuts (strengthen the planes to exclude feasible but non-integer points).

Also uninterpreted functions, where we add $x = y \Rightarrow f(x) = f(y)$.

If we are reasoning about arrays, we can use *instantiation* as a technique.

Note that DPLL(T) never adds a new inequality etc. that was not in the statement.

Example 6 (Diamond example) $1 \leq i \leq n\big[x_i - t_i \leq 2$*;* $y_i - t_i \leq 3$*;* $(t_{i+1} \cdots \big]$ *will always need exponentially cases under DPLL(T).*

Abstract CDCL (ACDCL) assigns truth values to Booleans *and* intervals to reals. Then if $x \in [1, \infty)$ and $y \in [4, 10]$ then $z = x - y$ means $x \in [-9, \infty)$. Analyse contractions, and maybe learn clauses weaker than those in the original formula.

MCSAT. Assign to propositional atoms and numerical variables. When finding an impossibility when assigning to x_{n+1}, we derive a partial impossibility on $x_1, \ldots, x_n$ (partial projection). For NLSAT, this was extended by [**?**].

Not covered: MASXAT, QE, Craig Interpolation (which is a sort of generalisation of QE).

3.2 The Complexity of CAD w.r.t Polynomial Degree: England

Conventional wisdom says "polynomial systems are doubly exponential [in the number of variables]", be it GB or CAD.

GB: we have [**?**]. [**?**] showed this wasn't the case for 0-dimensional, and [**?**] in general showed it was doubly-exponential in the dimension.

What about CAD ([**?**])? Projection/Lifting, where projection operator guarantees that the sample point is "universal" in its cell in the decomposition. [**?**]. [**?**] $(2d)^{2^n-1}m^{2^n-1}2^{2^{n-1}-1}$ for the number of cells, which all experiments tend to agree is correlated with other measures. In practice, tough, we only need truth-invariance. bets technique is to project better. Define *equational constraint*, either explicit or implicit, as $f_1 f_2 = 0$ in $(f_1 = 0 \wedge g_1 < 0) \vee (f_2 = 0 \wedge g_2 < 0)$.

Example 7 (ISSAC 2015) base *1118205 cells*

[**?**] *11961, or 158475*

[**?**] *20179*

[**?**] *119 (or nearby) cells. Note that this was letting the set of projection polynomials vary. Also the invariance of the truth may not be simply in terms of the signs of the polynomials. Double exponent of m is now $n - \ell$, where ℓ is the number of primitive equational constraints in different variables. But there are still problems with degree.*

Why is degree too hard? Answer iterated resultants forces doubly-exponential. But the true answer might be the multivariate resultant [**?**]. Need to reformulate [**?**], as that does total degree as CAD is per-variable degree. Hence the degree in the projections

of the ECs are singly-exponential, and we don't care about the other projections. Two restrictions.

- Constraints in successive main variables — probably only necessary for the analysis.
- Primitivity. Necessary because of [**?**].

Q–EA But we know that the single-block QE problem is singly-exponential: what can you say about this?

A Not a great deal: we have taken account of the logical structure, but not of the quantifier structure.

Q What about Gröbner complexity you are using?

A In practice, if you can do the CAD, the GB is trivial.

3.3 MathCheck 2: Bright

A SAT+CAS verifier for Combinatorial Conjectures. In particular, we are interested in conjectures in combinatorial design theory about the existence of Hadamard and Williamson matrices.

1. Williamson matrices of order 35 do not exist. Proved by Dokovic, but wanted an independent verification
2. Can show exist $n < 35$ (even orders mostly unstudied).
3. Over 160 new Hadamard matrices not previously in MAGMA's library.

Definition 7 *Hadamard matrix has* ± 1 *entries and any two rows are orthogonal.*

Order must be a multiple of 4, but do they always exist? Easy to encode ± 1 as T/F, multiplication is easy, but the cardinality constraint is harder: need to construct an n-but binary adder to check the output is $n/2$. Got up to $n = 20$.

Definition 8 *Williamson matrix. Needs to be symmetric and circulant, so in fact an* $n\times$ *n matrix is defined by* $\lceil \frac{n+1}{2} \rceil$ *elements. Hence really consider the Williamson sequence. Also need*

$$PAF_A(s) + PAF_B(s) + PAF_C(s) + PAF_D(s) = 0 \forall s$$

where PAF is a sum of autocorrelations.

Definition 9 *The* m*-compression of a sequence of length* dm *is a square of length* d *whose* j*th entry is* $\sum +i = 0^{m-1} a_{j+dk}$.

Theorem 14 (Dokovic–Kotsireas) *Williamson sequences satisfy* $PAF_A(s)+PAF_B(s)+PAF_C(s)+PAF_D(s)=4n$*, and this is true even if we compress.*

We therefore use CAS to solve $w^2+x^2+y^2+z^2=4n$. We also partition the search space (parallelism!), which we do by Power Spectral Density.

Theorem 15 (Dokovic–Kotsireas) *Williamson sequences satisfy* $PSD_A(s)+PSD_B(s)+PSD_C(s)+PSD_D(s)=4n$*, and this is true even if we compress.*

With 7-compression, there are 119 5-sequence which are legitimate. But also the instances are very similar, so we can look fro an UNSAT core, and hope it can apply to other cases.

Q Generalisation of Strassen's matrix formula, e.g. $n=3$ where 23 is least known, but unproven?

A Heard of it, but not done it yet.

3.4 Generalised Branch+Bound and SAM modulo NIA: Kremer

Software is SMTRAT. Fundamental branch-and-bound, dto decide if to use a **R**-solver. If UNSAT, then UNSAT, if SAT $\in$ **Z** then solved. Otherwise return a lemma which excludes (at least this) non-integral solution, but no integral solutions.

3.4.1 NRA Strategy

- Divide search space into finitely many satsfiability-equivalent regions. Simple CAD-like scheme. How do we generate 1D sample points?
- Go from $k-1$ to k
- How to ensure completeness.

heuristics (soe worked for us). Select integes form themiddle of an interal.

Linearisation as n CVC4

Interval Constraint Propagation iSAT3, raSAT.

Bitblastig AProVe, CVC4, Z3. Seems to win currently.

RAT$_{\mathbf{Z}}$ is our solution. But now bitblast for ≤ 16.

SAT does as many as Z3, in one-tenth the time. Still room for improvement for UNSAT.

3.5 Erascu

Computing $\sqrt{x}$: While width (I)$> \epsilon$; refine(I, x).

See [**?**]: this is actually non-trivial. Standard map is Secant-Newton: $I := \left[L + \frac{x-L^2}{L+U}, U + \frac{x-U^2}{2U}\right]$. Generalise to $L/U := L/U + \frac{x-quadratic}{linear}$. needs correctness, quadratic convergence $\forall x \exists c \forall L, UU' - L' \leq c(U-L)^2$. Then minimize $E := \sup_{L,U,x} \frac{U'-L'}{U-L}$. This can't be done numerically as we have quantified constraints.

1. subdivide QE problem: in particlar eliminated y from correctness by observing the function is convex, so end-points suffice. Also needed results about moving monotone functions just to endpoints.

2. use Mathematica/QEPCAD etc

3. Some manual work.

Result is that $E \geq \frac{1}{4}$, equality at $p = (1, 0, 0, 1, 1)$, $q = (\dots)$. In fact only U changes.

Note that convex/cave and monotonicity are easy for lienars, but even quadratics produces a case distinction, and I can't go further.

Chapter 4

21 September 2016

4.1 Enhancing extended Hensel by GB: Sasaki

Trying to lift $F \in \mathbf{Q}[x, \mathbf{u}]$ $F = f_n(u)x^n + \cdots$, the singular case is $f_n(0) = 0$ or $F(0) = 0$, when standard Hensel doesn't work. Factorization: G= Generalised HC; W=Wang; E= EHC [SasakiInada]. [InadaSIGSAM2005] timings. Also series expansion. Let N_i be the loer sides of the Newton polynomial of F (x-degreee versus total degree in u_i. Starting at x-degree n_i to n_{i-1}. Let $\overline{F}_{N_i}$ be sum of terms on N_i, and ? $= \overline{F}_{N_i}/x^{n_i}$ — net polynomial.

We have a choice, between max-EC (lift one fatcor along N_i, and min-EHC (split N_i according to all the points lying on/above (?) it. Formally EHC i sthe same as GHC [sketch of two-gactor algorithm]: solve $\delta F^{(k)} = \delta H^{(k)} G_0 + \delta G^{(k)} H_0$. [**?**] $A_l G_0 + B_l H_0 = x^l$ by Euclidean. But we have denomnators.

Two ideas

1. Use GB instead of Moses–Yunfunctions Conider $GB(\langle G_0, H_0 \rangle) =: \Gamma = \{\hat{G}_1, \ldots, \hat{G}_m\}$. Syzygies $\hat{G}_j = a_j G_0 + b_j H_0$. Note that we need the degree conditions $\deg(\delta G^{(k)}) < \deg(G_0)$ etc. Then claims that $\hat{G}_i$ gives the denominator D: $D\delta R^{(k)} = SG_0 + TH_0$ with D, S, T all polynomial.

2. Symbolise the denominators. Introduce a system variable `%D[i]` for the denominator. Also `%T` (?). Then can compute with these symbolically.

So during the GB we only compute "procedural" syzygies, then afte rthe Gb computation we convert these to real syzygies. We give a symbolic number to each polynomial computed/reduced.

Claims a time saving $\times 100$, but old is Mathematica program (not native?), and new is GCAL.

4.2 Abramov

K a field of characteristic 0 with derivation $\partial = \prime$. Conisider $Mat_n(K[\partial])$. "ord" = differnetial order. [Miyake1980] shows an invertible (unimodular) operator in L. So what

is the complexity, eitehr arithmetic or just number of differntiations (differential complexity). So this is like comparing sorting on #comparisons and #swaps. Let $F_X(n,d)$ be the full (arithmetic) complexity of algorithm X.

The frontal matrix is obtained by expressing as a polynomial in ∂ then taking ?the first non-zero row in each?. In his example, this matrix was not invertible. Two algorithms, EG and RR.

EG-reduction: Given L of full rank, EG constructs embracing operator $\tilde{L}$ such that $odr(\tilde{L}) \leq orl(L)$, the leading matrix of $\tilde{L}$ is invertible,a nd $L\tilde{L} = QL$ where ??. Claims $F_{EG}(n,d) = \Theta(n^3d^2)$, also RR.

Proposition 1 $\tilde{T}_{\Delta EG}(n,d) = \Theta(n^2d^2)$, $T_{\Delta EG}(n,d) = \Theta(n^3d^2)$,.

Each of ΔEG, ΔRR wil give a function whose zeros are a superset of the singular points. At least in complexity theory terms, ΔEG seems better.

For tetsing unomodular (only), then the number of differnetiations is exactly nd nd theleading matrixis invertible n K afte rthese differentiations. Sucj testing has complexity $\Theta(n^3d^2)$. Full complexity of ΔRR in $\Theta(n^4d^2 + n^3d^3)$. This uses the estimate $ord(L^{-1}) \leq (n-1)d$ [**?**]— this bound is tight (SA's work). In particular for $n = 2$ $ord(L^{-1}) = ord(L)$.

Problem 3 *Can we invert in $O(n^ad^b)$ with $a < 3$ — analogue of [**?**]? Thinks he can prove that multiplication is reducible to inversion, but other way?*

4.3 Incompleteness, Undecidability and Automated Proofs: Calude/Thompson

These have been used as arguments against automating mathematics. proof assistants have revived the interest in formal profs, and diminished these arguments.

Theorem 16 (Gödel) *No consistent systems of axioms whose theoreems can be effctively listed is capable of proving all true relations between natural numbers (arithmetic).*

Theorem 17 (Turing) *No Turing program can (correctly) solve every instance of the halting problem.*

Claims that both of these depend on the fixed system. In particular Turing says nothing about quantum computers (??).

Let $N(P,v)$ mean that P with never halts on input n. Suppose that certain strings of symbols have been singked out as proofs of patricular statements of the form $N(P,v)$. Two requirements.

Soundness If there is a proof of $N(P,v)$ then P never halts on v.

Completeness If P never halts on v, then there is a proof of $N(P,v)$.

Impossible.

Hilbert: "mathematics is only a language and a series of games, but not an arbitrary game with arbitrary rules". Hence Hilbert's programme. [Hilbert1900] cites "axiom of solvability" [there is a problem, we will solve it]. Emil du Bois-Reymond had claimed "ignoramus et ignorabimus', but Holbert claimed (at same conference as Gödel presented both completeness of propositional calculus and incompleteness, both ignored at the time, except by von Neumann) "we must know and we will know".

In fact, the undecidable outnumber the decidable, but this is an asymptotic result, and seems not to affect us in practice. The author believes this is due to the fact that we only care about tehorems with short statements. [Gödel1951] subjective mathematics is the body of all humanly demonstrable or knowable mathematically true statements.

Theorem 18 (MartinLöf[1]) *There are no propositions which can neither be known to be true not be known to be false.*

Many formal proofs. Includes Gödel's proof of the existence of God. Apparently [BenzmullerPaleo2016a–IJCAI] two months ago it was shown that Gödel's original system is inconsistent. The soutionsof Erdős Boolean Pythagorean triples program announced in May 2016 is 200TB. How to understand?

Note various proof systems: ALF (Aceremann is not primitive recursive), HOL4, Matita and Isabelle. In Isabelle, we prove equivaknce by simulation: Turing ⇒ abacus machines ⇒ partial recursive functions ⇒ Turing machines. How do we formalise halting within Isabelle? Note that we must be careful not to run programs. Use Hoare triples `{P}p{Q}`. But claims this means that a program p with a tape satisfying P will after some m steps halt with that table satisfying Q. Then we assume H exists for `halts M n`. Let contra(m) be ∞ if $M(M)$ halts (i.e. H(M,M)=0) and 1 otherwise. Then contra(contra) is ∞ if contra(contra) halts, and 1 otherwise. Isabelle models partial recursive functions via a dataype with `eval` (syntactic definitions) and `terminate` (inductively defined predicate). Then isf signaturen) is 1 if $n \geq 1$ else 0 if $n = 0$. Let $g(x, n) = F(x) \times signature(n)$. `eval(g(x,0))=0`, but `terminate`$(g(x, 0))$ is undefeind as F is unknown.

f(n)= 1 iff RH is true is recursive (since constant!) but not computable (currently!). Isabelle will confirm termination!

4.4 Setup of order conditions for splitting methods: Hofstätter

These apply to evolution equations:

$$\partial_t u = H(u) = A(u) + B(u)[+C(u) + \cdots]$$

where $\partial_t u = A(u)$ and $\partial_t u = B(u)$ are easily solved as $u(t) = E_A(t, u_0)$ etc. Example: Schrödinger.

Lie–Trotter $E_H(h, u) \approx E_B(h, E_a(h, u))$

Strang $E_H(h,u) \approx E_A(\frac{h}{2}, E_B(h, E_A(\frac{h}{2}, u)))$

Higher order $E_H(h,u) \approx S(h,u) = E_B(b_s h, \cdots) \circ E_A(a_s h, \cdots) \circ \cdots E_A(a_1 h, \cdots)$: what a_i, b_i? What ...

Taylor expansion of the local error gives multivariate polynomial equations in the a_i, b_i: order equations. Need to setup up (our question) and solve (challenging). We had an implementationin maple in the paper, but have since replaced it by a Julia implementation.

In the linear case, if A and b commuted, Lie–Trotter would be exact, so we get an expansion in $[A,B]$: in fact $-\frac{1}{2}[A,B](u) + O(h^3)$. In the nonlinear case, same with $[A,B] = A'B - B'A$. For Strang, we get $\frac{h^3}{6}\frac{d^3}{dt^3}L(0,u) + O(h^4)$, where the leasing term is a linear combination of $[A,[A,B]]$ and $[[A,B],B]$. If (4.4) has more terms, we get a mess of commutators. Formal considerations of order conditions only needs the linear case.

Ends up with questions of indepentent words in commutators.

So choose s (and number of terms, e.g. $A, B.C$). Then for $q = 1 \ldots, p$, in the expression

$$\frac{d^q}{dt^q}L(0) = \sum_{|k|=q} Binomial(q,k) \prod_{j=s\ldots 1} \sum_{l=0}^{k_j} \cdots$$

Search for Lydon–Shirshov words [Duval1988]. Note there are lots of these, 9 for $q = 6$ for AB, but ABC is much worse (115?).

For a given order s this is a large under-determined system, which gives a problem of finding an "optimal" solution.

4.5 Improved Computation of Involutive Bases; Seiler

Implementation done by Iranian co-authors. The basic algorithms for Gröbner or involutive algorithms are simple but highly inefficient. Needs many optimisations, especially avoiding reductions to zero. Gröbner [**?**], [Traverso1996], F4, F5. Involutive much less progress [ZharkovBlinkov1996], [GerdtBlinkov1998] had TQ-algorithm. [Gerdt2005] Buchberger criteria but less important.

A Pommaret basis reflects many homological properies of the ideal. Many invariants just drop out without further computation.

Problem 4 (δ-regularity) *finite Pommaret bases only exist in generic coordinates.*

Of course this doesn't worry theorists.

Our process is to make a sequence of "elementary moves", which is a deterministic apporach without parameters.

1. Compute Janet basis
2. while not Pommaret

3. Do elementary move

4. Recompute Janet basis (*)

So we need Janet bases (many). But note that Buchberger eats GB's asily, while TQ destroys the Janet basis before rebuilding it, so (*) is bad news. We are asking for a Janet basis of the same ideal in different coordinates. Need ideas from [**?**], which we reformulate as signature-based.

1. Initialise S as empty set

2. Always consider a tuple (p, me_i), where me_i is the signature

3. If the signature is divisible by a previous one, we can discard.

4. When computing a new polynomial, update S

5. Also allows for Buchberger gcd citerion as a signature test

So our idea is to initialise S by previous one, since same ideal. Needs to show that thisoptimisation only eliminates correct elements — see paper.

Compares with a "level playing field" implementation of F5 in Maple. Once 10% slower, but up to $\times 110$ faster. Note that [**?**] is less prescriptive, so we can, and do, use Schreyer ordering, and this might be the real reaosn for the speedup.

For Hilbert-driven, consider the homogeneous case only (simlicity). $HF_I(s) = \dim_K(P_s/I_s)$. $IHF_F(\dim_k(P_s/(\langle F\rangle_{L,\prec})_s)$ — note that this depends on the basis F.

1. Assume a priori HF_I

2. set T contains current basis, Q elements stillto be treated

3. If $p \in Q$ next tuple to be considered and $d = \deg(Poly(p))$, then check whether HF=IHF: if so discard.

In many cases this gives us a further $\times 2$. it detects many more reduction to 0. Occasionally slower, possibly due to computing HF.

4.6 Tash...

$P = k[x_1, \ldots, x_n]$, $I = \langle f_1, \ldots, f_s\rangle$. Ordered by $\prec$. G is a GB iff $LT_\prec(\langle G\rangle) = \langle LT_\prec(G)\rangle$.

We are interested in dynamic GB. Note that the shortest GB (depending on ordering) is unpredictable. So when a new polynomial is to be ordered, we changethe ordering (not changing previous LTs). [Caboara1993], [CaboaraPerry2014]. An ordering is defined by a real vector v. Write $\prec_1 \sim \prec_2$ if $LT_{\prec_1}(f) = LT_{\prec -2}(f)\ forall f \in F$. Notethat smaller Hilbert functions give smaller ideals, so use the HF [CaboaraPerry2014] to choose the new order.

We have an improved version o fthe theorem of [CaboaraPerry2014], use SAGE for polyhedron computations, and eliminating extra inequalities. Maple's polynomials. GebauerMoller updating and sugar strategy for pair selection.

Generally slightly better than [CaboaraPerry2014]: one case significantly word. But always better than DRL ordeing.

4.7 Symbolic Algorithms for geneating irreducible rotational-vibrational bases of point groups

Rayleigh expansion of the nuclear surface: 2D Manifold $\{q_1 = \theta, q_2 = \phi\}$ Basis Y_{lm}.

Example 8 (((3)+5+7)D Harmonic Oscillator) *We need an intrinsic frame G denotes a Lie group of transformations on the configuraton space $X = \{\xi = (\xi_1, \ldots)\}$. We have to trabnfer from the lab (?coordinates?) to the intrinsic coordinates. Pauili prescription quantisation of the intrinsic classic Hamiltonian.*

DVR algorithm expresses as Y functions, but in fact takes months to compute by Monte Carlo methods, once the representations are computed (Reduce, Maple).

4.8 Characteristic Polynomials of Structured Matrices: Law

$C(\lambda, x, y) = \det(A(x, y) - \lambda I_n)$. Examples came from Kauers, but not sure why. Entries are monomials in x, y, and in fact constant y exponent per column. Magma uses Bareiss. Maple uses Berkowitz [**?**]. There's also Hessenberg Algorithm. Decompose matrix so zero below immediate sub-diagonal, then polynomial drops out.

We reduce mod p, and replace x, y by integers, then use Hessenberg, and then lots of interpolation. Hence need a lot of bounds. Use $\min(H_r, H_c)$ for degrees.

But for these cases $c_i(x, y) = x^{f_i} y^{g_i} (x^2 - 1)^{h_i} \overline{c}_i$. Furthermore the f_i, x exponents in $\overline{c}_i$ are always even, so in fact we can interpolate in x^2. All this makes a major change.

Note, however, that these are only known factors in the answer, hence we need to evaluate, then take out the known factor, and then interpolate and restore the known symbolic factor. Also need to undo the $x^2 \mapsto x$ trick. For $n = 16$ we see $22\times$ improvement in number of points to interpolate.

Chinese Remainder Early Termination: easier to spot with symmetric representation.

On 64×64 seeing 21.52 seconds versus Maple 2.86 hours and Magma 15.1. Also very parallel as the Hessenberg can be done in parallel.

4.9 SN for BVPs: Gusev

Kantorovich expansion. FEM becomes BVP for a system of N ODEs, and hence a generalized eigenvalue problem $A + R(E) - EB)\Phi^h = 0$. Claims to reduce an $NL \times NL$ problem to two $N \times N$ problems. His methods get results matching to within 10^{-10} the original FEM. Used Maple/Fortran.

4.10 Multiple Eigenevalues of a Mathix dpending on a Parameter: Kalinina

Given square omplex matrices A, B: find all λ such that $A+\lambda B$ has multiple eigenvalues. Need Kronecker product $\otimes$ of two matrices. A, B have a common eigenvalue iff $C := \otimes I_k + I_k \otimes B$ has eigenvalue 0. In fact eigenvalues of C are pairwise differences of eigenvalues of A, B.

Q Can we fix the matrices to choose the multiplicities of eigenvalue?.

A Yes: see perturbation theory.

4.11 Kinematic Cosserat Equations: Lyakhov

Studying nearly 1D structures. Ther eis the special Cosserat theory of rods, but it is more convenient to use Darboux κ and ω vetors.

$$\begin{aligned} \kappa_t &= \omega_s - \omega \times \kappa \\ \nu_t &= v_s + \kappa \times v - \omega \times \nu \\ &\dots \end{aligned}$$

> If you cannot solve a nonlinear differnetial equation, search for a group. [Ibragimov]

Solves 75% of Kamke. Generating the determining equations for a one-parameter group is fully algorithmic, see any CAS. The convert to involutive system 9also algorothmic), solve the PDE (not algorithmic, but easy), then .... For exmaple, we have 138 PDEs and this took 80 minutes to compute.

So we have a semi-analytic solution which solves the stiffness problem and reduces numeric instabilities.

4.12 Business Meeting

37 submissions, 30 accepted. 112 reviews and 34 external reviewers. Top submissions Russia (20 authors), Canada (13), France (12). UK (2) was JHD/ME.

Changbo Chen proposed Academy of Mathematical and System Sciences in Beijing, 18–22 September 2017, more specifically KLMM. Many hotels near the site: 40–100 euros depending on *.

François Lemaire presented a proposal for Lille in 2018. Campus is 10 minutes from downtown Lille. Airports are Paris or Brussels.

Chapter 5

23 September 2016

5.1

Preconditioners, multigrid and Krylov spaces are all used, but independently. We propose CLR = Collocation and Least Residues. Us eNewton linearisation of Navier-Stokes in local coordinates.

$$A_{i,j} \cdot X_{i,j}^{s+1} = f_{i,j}^{s,s+1},$$

where the $A_{i,j}$ were derived with Mathematica and `FullSimplify` to reduce the length. The condition number is $\kappa(A) = \sqrt{||A_1|| \cdot ||A_i^{-1}||}$ where $A_1 = A \cdot A^T$. Tried to minimise κ by choosing ξ, η, but after 90 minutes this failed. Instead tried over a coarse grid D_1, then use a new D_2 with half the step-size but the same number of grid points. A graph shows that κ is very flat around the minimum.

10 Krylov residuals seems optimal. 20 very similar, but 1,2 very much worse (JHD: why not show, say, 5). Overall, best acceleration factor (230) was with $k = 9$ and an multigrid K of 5. For a different problem 9/4 was the optimum (162). but 8/4 and 10/4 were very similar.

Q–Gerdt Might this extend to 3D?

A Yes, but the ?? only needs to be done once. The structure is independent of the number of space dimensions.

5.2 Quadric arrangements: Pluta

Rigid motions are the key: $x \mapsto x.R + t$ where R is rotation matrix and t is translation. We want to use Cayley's transform: $R = (I_A)(I + A)^{-1}$. This is continuous,but I am a digital geometer: what happens in $\mathbf{Z}^3$? [Ngoetal,IEEETRansImageproc2014] in $\mathbf{Z}^2$. Note that such a transform may lose properties like connectivity. What hapepns in $\mathbf{Z}^3$? In pafrtcicular, what happens to $3 \times 3 \times 3$ cube under rigid motions. Our spandard image patch in $\mathbf{Z}^3$ is seven points: $(0, 0, 0)$ and each coordinate (but only one) can be ± 1.

(a, b, c, t_1, t_2, t_3) where $-\frac{1}{2} \leq t_i \leq \frac{1}{2}$ (w.l.o.g.). We have $O(r^4)$ hypersurfaces, but this means $O(r^{24})$ arrangements. Tried with CAD, but impossible, hence a dimension reduction: uncoupled the rigid motion. With the Caylet transformatoin we get polynomials of degree 2. This image patch gives 441 quadrics, but relaly 81 distinct. Similar to [MourrainTecourtTeillaud] ti get one sample point in each 3D-connected component. Differenc eis that we are using non-generic directions. Consider detection of events as a plane moves through the quadric.

User Maple 2015, grid framework for parellelism, and critical points as implemented in RAGLIB. 40core machine with ∼250GB RAM.

Q Different resolutions in depth versus image directions?

A would be harder!

5.3 Hofstätter

Lie–Trotter splitting as before.

Variation of constants gives a linear initial value problem $\partial_t X(t,u) = F'(E_F(t,u)) \cdot X(t,u) + R(t,u); X(0,u) = X_0(u)$.

$$X(t,u) = \partial_2 E_F(t,u) \cdot \left(X_0(u) + \int_0^t \partial_2 E(F(-\tau, E_F(\tau,u)) \cdot R(\tau,u) d\tau \right)$$

is the soluton.

Use our tool to check that X, R do indeed satisfy this. Then estimate X from this integral representation using the fact that $\int_0^t (\ldots)$ is asymptotlically smaller as $t \to 0$ than the integrand.

$$F(E_F(t,u)) = \partial_2 E_F(t,u) \cdot F(u)$$

is the fundamential identity. Repeated differentiation of this gives many replacement rules.

Claim defect $S(t,u)$ has a representation as $\tilde{S}^{(1)}(t, E_A(t,u))$ (Julia notebook) and $\tilde{S}(t,u)$ satisfies a differential equation as in variation of constants formula (Julia). This gives $D(t,u) = O(t)$ and $L(t,u) = O(t^2)$.

Demonstrates Julia notebook. The output is mathematical — $\mathcal{E}_A(t,u)$ for example.

Same approach for a more subtle error shows $O(t^3)$. [AuzingeretalJComp.ApplMath2015].

Q Can you use the cancellation law for your splitting for your approach?

A If we had one, but there isn't a generic one?

Q In your first talk you mentioned Maple/Julia: compare?

A We had very good experiences with Julia, as it is compiled.

5.4 Qualitative Analysis . . . Kowalewki top

Use $Ox_1y_1z_1$ as the interial ocordinate system, and $Oxyz$ the coordinates relative ot the bodt. γ_i copenents of first force field on $Oxyz$, δ_i second. 9 equations of motion. We want stationary sets for these, which are extremum cases for the first integrals. Several $\partial K \partial v$ for various variables v. In order to obtain invariant manifolds, we use Gröbner bases to eliminate in these. The equations of the maniforld themselves are quite simple, $\gamma_2\delta_3 - \gamma_3\delta - 2 = \gamma_3\delta - 1 - \gamma_1\delta - 3 = 0$, for example, but the vector field equations are much larger.

Can also obtain higher-level, even up to fourth, invariant manifolds.

In order to obtain in the phase space of the initial equations the equations of IM corresponding to a second-level on a first-level we just add the equations to the set, and so on. In each case, there's a suitable integral which takes on a stationary value on the IM. The IMs of lesser dimension are submanifolds of the higher dimension ones.

There is a question of stability, but the quadratic form of partial derivatives of the integral is positive definite.

Considering two IMs, each have a vector field $\dot{p} = 0$, and each corresponds to curve over which the family of solutions $p = p_0$ exist. The integral $V_4 := \cdots$ is an invariant. The equations are compatible with suitable $v \neq 0$, but stability depends on the sign of some of these (generally x_0)

Chapter 6

24 September 2016 Invited talks

6.1 Bridging Two Communities to Solve Real Problems: Chris Brown

two communities: not so much dealing with differnet objects as dealing with things in different ways. I'm dealing with an overlap area: real polynomial constraints. I am taking the area of CAD, and seing where areas from SAT-SOlving can help.

6.1.1 CAD

The logic viewpoint is a Tarski formula, the geometric view is the semi-algebraic object. Satisfiability of Tarski formula is non-emptiness of the semi-algebraic set. Then there's existential quantifier elimination, which is projection. Full QE is a combiation of projections and closures.

Hence CAD. A cell is a generalisation of a box: in $\mathbf{R}^1$ it's the same, in $\mathbf{R}^2$ the bounds are no longer constants, but depend on the $\mathbf{R}^1$ variable, and so on. Cylindricity says that projections are equal or disjoint. We can construct CADs for any Tarski formula. Start with the polynomials, and these sign-invariant regions, but not cylindrically. Add projection polynomials to solve this problem. Then do, recursively, univariate root isolation to produce sample points. Projection (according to variable order) and complementation are trivial. Any subset of the cells can be represented by a Tarski formula.

Example 9 *3 variables of degree 4. Shows the* $\mathbf{R}^2$ *decomposition: a great deal of cells.*

There's a lot of projection work to be done before any sampling. No use was made of logical structure.

6.1.2 SAT: DPLL and NLSAT

DPLL: simple example is Boolean SAT. Arbitrary decisions, plus propagation and backtracking.

polynomial constraint SMT is where (some of) the Booleans have meanings (in $\mathbf{R}$). Keymaera had 4000 real polynomial constriant problems, also metiTarski. CAD does really poorly on these. `cbrt-problem-2-weak-chuck-0073.smt2` the problem is trivially

false. This actually happens a lot— $> 25\%$. `vsl-alert.proof-node1357.smt2` $> 90\%$ in Keymaera.

Example 10 (NLSAT [?]) $(\underbrace{x < -\frac{3}{2}}_{l_1} \vee x \geq -\frac{3}{4}) \wedge \underbrace{x^2 + y^2 + z^2 < 1}_{l_3} \wedge (z + x > 3 \vee z - x - 2y < -1$. *Choose* l_1 *true, so* $x = -2, y = 0$ *then get UNSAT for* l_3. *Have to roll back* y, x, *and have to add* $\underbrace{x \geq -1}_{l_6}$ *and then this rools back* l_1. *So assume* $\overline{l_1}$, *hence* l_2 *and so on.*

6.1.3 SC^2

Generalise from a false point to a false region. He produces a decomposition, which is *not* a CAD. Call it a Non-uniform CAD. Still allows complementation, but not directly projection.

This process *is* incremental, so can "get lucky". Makes strong use of logical structure.

Example 11 (Zankl matrix 2-all-9) *Z3 solves in 190 seconds, making 8 variables 0. He can't solve this with NuCAD, but can if variables are set to 0.*

6.2 Abraham

SAT: competitions since 2002, and the 2016 has 6 tracks all with a standard format etc.

DPLL solving: conceptually enumeration, but using propagation to reduce the search.

Theory solvers, and how do they produce explanations of unsatisfiability. Various theories, UIF, various arithmetics $\mathbf{R}/\mathbf{Z}$, and linear or not. Note that nonlinear inters are undecidable in theory, but we can consider bounded ones.

These solvers need to be SMT-compliant: work incementally, generate lemmas explaining inconsistences and br able to backtrack. There are also software packaging issues: not thread-safe, not available as libraries/subroutines. Henc eour SMT-RAT library of procedures to plug into SMT solvers.

Chapter 7

SC^2: 24 September 2016

7.1 EnglandDavenport

See `http://staff.bath.ac.uk/masjhd/Slides/SCSC-2016-1.pdf`.

7.2 Computing Boolean Border Bases: Messeng & Horacek

Definition 10 *$\partial O = (x_1 O \cup x_2 O \cup \cdots) \setminus O$ for an order ideal O (closed under multiplication). An* O–border prebasis *has every element $b - \sum t_i u_i$ with $b \in \partial O$ and $u_i \in O$.* O-boder basis *if also ...*

How to compute?

1. $U = \langle Supp(f_i) \rangle$
2. V an LT-intereduced basis of $\langle f_i \rangle$
3. stabulisation
4. $O := U \setminus \cdots$
5.

7.2.1 Boolean Polynomials

Let S^n be the square-free terms. Note the importance of Boolean polynomials in pratice, e.g. AES equations. Hence

$$\partial O = ((\partial O)^{sf} \cup x_1 O \cup \cdots) \setminus O$$

Hence the stabilisation phase becomes slightly more complicated. Whenever we get a new polynomial we "send it to the SAT-solver".

7.2.2 SAT

"Let W' be ..." is the key step. The set of polynomials in V can be regarded as a matrix over $\mathbf{F}_2$ — sparse or dense.

$$V_i \subseteq V_{i-1} \cup B_{i-2} \cup W_{i-2}^{+} = V_{i-1}^{(+)}.$$

A SAT solver eats CNF, and DPLL, while BBA is linear algebara elimination. BBA produces many new polynomials eatc ime. [Courtoisetal2007,JovanonicKreuzer2010]. Want polynomials we smallsupport, so to produce short CNFs as well. Note that $l_1 \equiv x_i + 1$ and $\overline{l_i} \equiv x_i$ so negative literals are good.

MJB Modern SAT solvers aren't DPLL, and haven't been fro 15 years.

Vijay Perhaps you should lift the concepts, rather then convert the polynomials.

7.3 CoCoA

7.3.1 Bigatti

Basis commands to define a ring, and then an ideal. Note the twin float representation which lets us represent rationals. Hence "retry with higher precision" is a valid answer.

7.3.2 Abbott

CoCoAlib is the C++ library underpinning (most of — some packages are still to translate) CoCoA. The CoCoALib software is designed to be easy touse correctly.

Vijay How to you achieve "no surprises"? Might machine-generated tests be better?

JAA We've use random numbers, but this isn't perfect, e.g. factoring

7.4 CEGAR: Griggio

Mostly Ahmed Irfan's work — still work in progress. Do SMT in the sersice of verification. We have formal models of transition systems. Linear case is well-supported. Interpolants, unsatisfiable cores etc. However, there are nonlinear ocnstraints 9and even non-poynomialones) epsecially in cyber-physical. Noet that most of the model way still be linear. We will abstract away NRA via UIF: regard nonlinear multiplication as `fmul(x,y)`. If UNSAT, then certainly unsatisfiable. If we get a model μ, check that $\mu[\texttt{fmul(x,y)}] = \mu[x] \cdot \mu[y]$. If not, we re-approximate the problem. CEGAR = CounterExample-Guded Abstraction Refinement.

Naïve refinement, e.g. $\mu[\texttt{fmul(3,5)}] = 10$ gets corrected by forcing `fmul(3,5)`=15. Only on epoint. Could add two lines, which is better, but still doesn't solve. Instead we add the tangent plkanes to `fmul` at (x, y). This gives us four inequalities in the four quadrants, and is strictly stronger than the two previous.

Howver, we can keep adding lower (say) bounds, and get a Zeno-like problem.. Instead, keep a frontier, and generate a second tangent plane that generates an upper bound tangent plane covering the same region. This solves Zeno for bounded-regions.

Note that there's symmetry, rathe rthan adding a lemma for `fmul(x,y)` in terms of $x$and y, we can add a lemma for y interms of x and `fmul(x,y)` etc. We can also move away from messy rationals, we can look at $(\lfloor \mu(x) \rfloor, \mu(y))$ rather than $(\mu(x), \mu(y))$, which is still exact at $(\mu(x), \mu(y))$. Also cheap lemmas: monotonicity, commutativity etc.

Not complete. how about $\delta - satisfiability$, which is appropriate in cyber-physical systems. On the UNSAT QF_NRA, we had a Python implementation. 2500 are UNSAT just with `fmul`. After a time (1600), this beats `dreal`, and CVC4 at about 500. Still well behind Z3.

Ongoing work:

- integration in our tools
- deta-completeness
- pre-processing
- beyond polynomials.

MJB CVC4 doesn't really support non-linear (hence this is really a problem with the benchmarks). We also find CBMT to be better than CEGAR.

A Thanks.

Q ICP?

A That gives us nasty staircase effects.

JAA More than two multiplicands?

A No, but we did treat squaring as a special case. It might make sense, though, expecially if `x*y*x` occured a lot.

7.5 MathCheck2; Vijay Ganesh

Symbolic Formal manipulation. Complete in some domains.

Search method Complete only for finite domains. Effective when combined with learning.

DPLL

propagate

Detect conflict Key improvement is to detect a "root cause" of the conflict (CDCL).

Decide (branch)

Backtracking (implicit).

An SMT Theory solver adds more power to conflict analysis.

7.5.1 Bright

Hadamrd matrices ±1 entries, any pair of rows orthogonal.

Heuristics?

A Not yet. Looekd briefly.

Q–ME SMT-compliant?

A essentially start anew SAT each time.

7.6 Accurate Deadcode Detection: Neubauer

Project with Oldenburg, Frieburg, BTC and Sick AG. Dead code is progrmaming errors, or automaic code geneation, compiler optimisations, or using a general code in specific applicatuons. Note that dead code hurts test coverage. Also, some standards require dead code removal. Pur application is BTC's SimuLink generated code.

C → BTC Toolchain → SMI2SAT → HYS → iSAT3 → scripts, and back to BTC toolchain.

Adds `LINE_Y_REACHED` variables. Look for one coverage goal [at a time], unwrap loops, and convert to SSA format. Two techniques can make this complete Craig interpolation and .... isat3 includes non-linea nd even transcedental functions. It is not a lazy model, but rathe tightly integrated, with diretc reasoning about theory atoms. See FMCAD 16. IEE754, bitwis einteger Used time-limit 60 seconds, and BTC's live code (8778) each is a one-coverage task. ISAT3 solves 8430, 8099 for CBMC [bit-blasting]. Amost equal for reachable, but ISAT3 got 200more unreachables. Also more "bounded unreacahables" (unrolled to depth 51 and proved nothing). Once we get over 3 seconds, ISAt3 is better.

Sick's code is very interrupt-driven, and we can't do that. Need a new technique: semantics-preserving restriction of interrupt points. Should also add elementary functions. Should also look at symbolic computation.

Q f.p.

A use a new ICP contraction in each case. We use machine floats with outer rounding for the real domain, and the target float.

7.7 Satisfaction meets Practice and Confidence: Bienmüller+Teige

TS BTC was not a "tick on the website" company.

TB But we also want to gain form the project

As a tool vendor, we have to provide reliability. 99% automotive: ISO 26262: heavily in MatLab's dspace toolchain. Model classes are iportant, especially IEEE 754 (customers are moving this way, but not there yet: we have *some* time). Also interested in proof certificatoins.

Fixed-point approximation for real numbers. Model checkers reduce this well to Boolean Satisfiability. But as costs for FPUs come down, customers are moving this way. Bitblasting is expensive in time and space.

Example 12 (exp) *700 lines of code, and cmbc time was 5 minutes. With experiment SMT in isat3 it was* < 1 *sec, but* $\mathbf{R}$ *not* $\mathbf{R}_{IEEE}$.

ISO 26262 requires "tool qualification", depending on Safety Integrity Level. Showed a TUV Certificate for BTC `Embeddedtester`, for "Back-to-Back-testing". We now want our Formal Verification to reach same level. But there may be objections from certification authority: "this will reduce testing".

False Negatives are not a real problem, but a False Positive is a disaster. A bug in a tool might force a manufacturer to reverify everything. What can we do to achieve $\sim$ 100% confidence in our tools. We are not necessarily asking for perfection, but improvement. It can't rely on manual intervention.We would like "proof certificates" on the input code, not just the inout to the model checker, which has had various pre-processing etc. Certification technology shoul dnot increase verification time "too much".

"Can't youjust apply a second model checker"? Well, we already have a multi-tool solution. also requires extra computation resources at the users.

`http://techon.nikkeibp.co.jp/atclen/news_en/...` as a quote from Toyota Prius, and one from MAN.

We have no experience at BTC about symbolic computation. Can you tell us how this technology could help us.

PF Benchmarks? Can you provide some to us?

A We could do so: it would require obfuscation.

TS Daimler gave an obfuscated version of a complete configuration of a model car to community.

A Precisely.

TS If I gave you perfect $\mathbf{R}$ code, would this help?

A No, I really need $\mathbf{R}_{IEEE}$.

TT We are currently verifying production code, and at that point it is $\mathbf{R}_{IEEE}$. It comes froma Simulink model, and this is also $\mathbf{R}_{IEEE}$.

Vijay What sort are the f.p. errors?

7.8 JHD

See `http://staff.bath.ac.uk/masjhd/Slides/SCSC-2016-2.pdf`.

7.9 Brain

There's a coevolution between CAV and SMT. Hence types like "bit vector" and "arrays". The SMT side can't really do quantifiers, so CAV doesn't. This also explains the feedback loop in benchmarks. At least in SMTLib, problems are generated by computers, not mathematicians. **Challenge: relevance**.

Verification really generates verb+if...then...else+. We once saw `X?Y:Z` nested 700-fold deep.

The theory of bit vectors has over 30 operators, `bvsod` etc. **Opportunity: various subalgebras** such as ARX, max-plus. We have reduced IEEE to 12-pages, but it's not the Numerical Analysts' standard model, but the SMTLIBmodel which[1] has $v : \mathbf{R}_{e,s} \to \mathbf{R}^*$ such that $f \oplus_r g$ is $\texttt{round}(, rv(f)+v(g)$. This has been cross-tested against hardware, and is finding bugs in the hardware. **Opportunity mixed real/float** based on $\mathbf{R}*$.

Expression simplification is SMT's dirty secret.

$I(x)$ Initial state.

P safety property

FRStep(S $S \cup \{t|T(s,t) \wedge s \in S\}$

FReach $LFDP(FRStep, [I(x)])$ but there's a lot of issue son fixed points.

Safety $FReach \subset P(x)$

Approximate upper/lower bounds for pre-/post-image. **Opportunity: better sets than boxes**.

This talk contains more questions than answers.

[1] Slight simplification to deal with signed zeros, so round has a hidden parameter.

Chapter 8

25 September2016

8.1 Mechanically certifying formula-based Noetherian Induction Reasoning: Stratulat

$$0 + y = y \tag{8.1}$$
$$s(u) + v = s(u + v) \tag{8.2}$$

Hence $0 + 0 = 0$ by first. Then $s(0) + 0 = s(0)$. But what about $z + 0$? This seems to end in a loop unless we assume Peano induction:

$$\frac{\phi(0) \qquad \phi(x) \Rightarrow \phi(s(x))}{\forall z \quad \phi(x)}. \tag{8.3}$$

Noetherian inductoin: let $(\mathcal{E}, <)$ be a well-founded poset.

$$\frac{\forall m \in \mathcal{E}(\forall k \in \mathcal{E}, k < m \Rightarrow \phi(k)) \Rightarrow \phi(m)}{\forall p \in \mathcal{E}, \phi(p)}$$

Peano induction, structural induction [Burstall1969]; cover-set induction [Zhnagetal1988].

We therefore have induction shemas etc., but it's hard to deal with mutually-recursive predicates.

Formula-based induction.

$$\frac{\forall \gamma \in \mathcal{E}(\forall \delta \in \mathcal{E}, \delta <_f \gamma \Rightarrow \delta) \Rightarrow \gamma}{\forall p \in \mathcal{E}, p}$$

Inductionless induction, term-rewritinginduction [Reddy1990], cyclic induction [Stratulat2012a].

Theorem 19 *The term-based induction principle can be presrepsented as formual-based induction.*

Theorem 20 *Formula-based induction induction principle can be represented as term-based induction whern $\mathcal{E}$ is of the form $\{\phi(t_1), \ldots, \phi(t_n)\}$*

The general case (without this restriction) is only conjectured.

Infinite descent is the contrapositive of Noetherian deduction.

$$\frac{\forall m \in \mathcal{E}, \neg\phi(m) \Rightarrow (\exists k, k < m \neg\phi(k)}{\forall p \in \mathcal{E}, p}.$$

8.1.1 Mechanical

CIC-based and integrates Netherian induction. Proof certificatoin by Curry–Howard correspondence. Various formal proof developments (CompCert project, Odd-order theorem [**?**]).

So to formalise formula-based induction, we most formalise

1. Induction ordering and formula weights
2. formula-based induction principle
3. inference steps from the formula-based proof.

To formalise the weights, we use COCCINELLE [Contejeanetal2012a]. Slide of code for COCCINELLE. Then (8.3) looks like this.

```
Fixpoint plus (x y:nat): nat :=
match x with
| O => y
| S(x') => S(plus x' y)
end.
```

Formally, syntactically represent each conjecture ϕ as a weight w_ϕ; variables are shared by anonymous functions, and $\mathcal{E}'$ will consist of anonymous functions. Extend COCCINELLE with a dual computable function for `less`.

- Inference rules: transitions between states
- Derivation of E^0
- ??

Concrete inference system I_{imp}: axioms are oriented into rewrite rules: GeNNat (G), SimpEq (S) and Equational (E). So to prove (8.3) in Coq, we weight the symbols (e.g. plus is 7), express (8.3) in terms of these rules, and get Coq to check.

An alternative toimplicit induction is cyclic inference, whihc has fewer constraints.

This gives us the SPIKE theorem prover Integrates an automatic form of cyclic induction reasoning. Specification by conditoinal rewrite rules.

Examples, including validation of a conformity algorithm for ABR telecomms protocol. There is a queston of Coq certification of Spike implicit induction proofs. 40% of ABR lemmas were automatically proved. However, there are limits: the conditional rewriting can only have true/false, and limited arithmetic. here is a Coq tactic Spike which solves the translation problem at the specification level. There is more work to do in this area. See [**?**].

8.2 machine learning to decide when to precondition CAD with GB: England

ZH's PhD work. Traditionally build a CAD by P/L. Note that a sign-invariant CAD is often overkill, aim for truth-invariant CAD. If we have equational constraints,we can replace by a Gröbner base [**?**], and use this to simplify [**?**]. [**?**] had one example that timed out when GBs were used. By [**?**] GBs were much faster, but some examples still slowed down. So can we use Machine Learning to decide when?

Various heuristics have been used [**?**] for variable ordering, using NLSAT dataset. For this question, every example was sped up, unlike the [**?**] examples. So generated random examples in Maple. In 300 seocnds, 1062 problems fininshed. 85% benefitted from GB preconditioning (using RCCCAD). We were measuring performance by #cells, which is hardware-independent. Use SVM-lite, with 80:20 training:testing (maintaining 75% ratio). 28 features, of which 11 turned out to be useful.

- Needed to use features from after the GB to improve over blind choice.
- Did better than any particular heuristic, but still not perfect.
- Would like to run on Todai robot problem set, when it's available.
- How to combine this question with other questons like variable ordering?

Q–Lichtblau Interaction with ordering?

A For this, we fixed $x > y > z$, but some problems come with fixed ordering, some don't.

Q-Brown Were you reducing the inequalities as well?

A Not here: there's a lot more to explore, but ZH needed to finish her thesis.

8.3 Parallel Integer Polynomial Multiplication: Chen

Reduce from $\mathbf{Z}[z]$ to $\mathbf{F}_p[x, y]$ and use 2D FFTs. Choose $2^l > ||f||_\infty + ||g||_\infty + \max(m, n) + 1$, and efectively replace 2^l by y. Other options include Toom-Cook k-way, with static parallelism $\times 7$ for $k = 4$ and $\times 13$ for $k = 8$.

Assume $K = \Theta(d)$ and $M = \Theta(\log d)$: use [**?**] to manage these assumptions. $O(dN \log(d^2) \dots)$, which is better than SS. For really large examples, beats both Flint and Maple's standard. This algorithm is best for very large problems on lots of cores: on few cores use Toom-Cook.

8.4 Polynomial GCD by Syzygies; Duarte & Lichtblau

Multivariate Polynomial GCDs: methods include PRS, GB (which essentially compute LCM, so slow), Interpolation (problems with small fields).

Syzygy of $(f_1 \ldots, f_m)$ is $(s_1, \ldots, s_m)$ such that $\sum f_i s_i = 0$. Knowing the GCD helps find the syzygies: reverse engineer is. If (p, q) is minmalsyzygy of (P, Q) then $Q/p = -P/q = \gcd(P, Q)$.

1. Let $M = \begin{pmatrix} P & 1 & 0 \\ Q & 1 & 0 \end{pmatrix}$. The rows generate a submodule of $K[x_i]^3$.
2. use a Position Ove rTerm ordering on the columns and drl on x_i.
3. Compute a module GB of the submodule geberated by the rows of M.
4. a Linear combinator of the rows of M is a syzygy, and POT/drl proves its minimal.

Over **Q** MMa used sparse interpolation: 1.7 seconds versus our 20. $\mathbf{F}_3$: 25 versus 9.7. $\mathbf{Z}_5$ 172 versus 20, $\mathbf{Z}_{19}$ 408 versus 20, but for large primes the MMa wins again. Note that there is no unlucky prime concept here.

Q Algebraic number fields?

A Should work if you regarded the algebraic numbers as polynomials with relations.

AB This is related to the CoCoA method.

8.5 Effective nodeterministic PD test for unidiaginal integral matrices: Mroz

Positive definite matrices: many characterisations. $Q_A(x) > 0$ for evey nonzero $\mathbf{x}$; All leading principal minors of $A^T + A$ are positive, etc.

Work with unidiagonal $A \in M_n(\mathbf{Z})$. Pessimistic $O(n^4)$, but good case is positive definite. A side effect is that we discover the Dynkin type of A is positive definite. Better versins are $O(n^3)$. Start with $\Delta(A)$ as a bigraph (in fact bi-multigraph).

Theorem 21 *Let Δ be a connected loop-free bigraph with $n \geq 1$ bertices. Then is Δ is positive every sqeuence of inflations is bounded by $\beta(n)$, $n^2 - 2n$ for $n > 8$.*

Versus Maple 15, our C++ shows a 4–6 speedup: not breathtaking but not bad.

Q Any analysis of the randomised graph process: Markov process for example?

A Not formal.

8.6 Split Type Problems in Nonlnear Analysis: Ansari

How do we find a point in the intersection $\bigcap C_i$, when very little is known about the C_i. Essentially the general feasiblity problem.

$C \subseteq H_1$, $Q \subseteq H_2$ nonempty closed convex sets. $A : H_1 \to H_2$ bounded linear operator. Find x^* such that $x* \in C$ and $A(x^*) \in Q$.. This is a split feasibility problem. Writing it as $x \in C \cap A^{-1}(Q)$ appears to reduce it to a simple feasibility probelm.

8.7 The quest for Symmetry: Heule

Cross-reference to Section 7.8. For me, breaking is the hard point. SAT-solving struggles in the presence of symmetries.

Does there exist a graph of six vertives which doesn't have q clique or co-clique of size 3 (Ramsey 3). This is OK, but even Ramsey 4 is very hard. Note there are eight graphs on three vertices, in four 4 classes (0/1/2/3 edges). These can be eliminated by $(ab \vee \overline{ac} \wedge (\ldots)$. Existing symmetry-breaking methods constrain the adjacency matrix. Current such techiques are perfect for $n \leq 4$, but by $n = 10$ has 12–15 per isomorphism class. Given $n = 3$, Tseytin encoding gives 13 clauses (for each) but it can be reduced to 2. Use Nauty's canonical sets and a SAT-minimiser.

Alternativerly, let $F_{k,m}$ express the existence of a perfect isolator. $F_{5,12}$ was 102 seconds, but $F_{5,11}$=UNSAT took an hour. But $k = 6$ was unsolved after 24 hours.

In practice new techniques scale well, growing slowly (quadratically).

Chapter 9

26 September2016

9.1 Time Tracks and Time Segments. Rethinking the Way to Look at Texts: Dan Cristea

At a time, we remember what happened at another time. Our work is framed in Temporal Logic. TimeML related dates to DCT (Document Creation Time). Claims (Webster)

1941 Time line (JHD: OED 1907: communicated to speaker)

1951 Story line (similar)

A "Time Yard" is a diagrammatic reprsentation of thecharacters' destinites. Made up of Time Tracks (TTs) make up of Time segments (TS). Then a book is a sequence of TS. One TS can only be narrated by one relator. A time segment can be part of more that one time track, when it relates events about more than one person etc.

9.2 Lexicalisation of DBPedia

Various differently phrased sentences meaning the same thing, e.g. "died of cancer", or "worked in Analytic Number Theory". From DBPedia, but also other sources. Specific goal is "ontology lexicalisation", a relatively new concept.. There is averb+lemon+ standard for sharing lexical information on the semantic web. Want to exploit the relation betwene Wikipedia and DBPedia. Two approaches in the literature:

M-ATOLL sytart from a property and retrieve triples from DBpedia;

Learning Start from a multilingual corpus.

We are following the first. Starting from triples in DBpedia, we can look in Wikipedia: Full matching, partial matching or "co-reference matching" [pronoun following is needed to complete the triple]. The unspervised learning by spectral clustering.

9.3 Extracting gamers opinions from reviews: Sirbu

Note that marketing budgets can be >> production costs for games, and reviews really matter. PCA detected 8 components explaining 51% of the variance (JHD? of what, probably scores). Neutral reviews were the hardest to predict.

9.4 Comparing different term weighting schemas for Topic Modeling: Truic̆(a)

Do I get a better distribution if I change the weighting. Also runtime questions. Evaluation via a 'purity' score.

Use 18464 conference artucle, and 18814 articles form "20 Newsgroup". Clean text expanded contractions, removed stop words and punctuation. Then lemma text did rather more. LDA and NMF (Nonnegative Matrix factorisation). Clusering via K-means. Scikit-learn Python package. Lots of statistics: geneal concluson is that the best weighting scheme depends on the analysis method, and also on large/small vocabulary. Lemma text doesn't really change the scores, but improves the running times as the matrix is smaller.

9.5 Software defect prediction: Marian

Trying to identify those parts where problems are likely to occur, hence directing effort. This is a classification problem, so most ML approaches use supervised learning. The data are unbalanced, and the entities are often similar. Out method is based on fuzzy decision trees (new for this problem).

Training data comes from a softwraesystems, and are high-dimensional data. Based on literature review [RHTv13] we have three metrics (not explained). Two fuzzy functions for each attribute:inspired by [FBF15]. Because of the unbalanced, accuracy seems high (0.80–0.88) but AUC is lower. Our fuzzy DTs do best at 0.735.

9.6 ML for Bioarchaeology

Propose to use SOM, compared with RBFN.

1. 200 male + 200 female skeletons form Pretoria Bone and Raymond A, Dart collections. 10 radius + 9 ulna.

2. 297 from Robert J. Terry —cranial data.

Aim was to identify sex: Accuracy seemed 0.8.

9.7 HCI inserious gaming for clinical puroposes:

Kinect etc. can detect input from body gestures. Kinect 2 detects better for pose, especially with lower performance people (under 0.9). But much nore time consuming. Hence doesn't necessarily won for real-time systems

9.8 Malware classification based on dynamic behavior

Advantage is high proactivity, but some actions can't be undone. Hence dynamic analysis on a virtual machine with a controlled environment with no provate data. So the "unknown file queue" gets sent to queue of virtual machines, and the logs from this sent to a post-processor (feature extraction) and hecne to a ML classifier. URL accessed, items from the path (now close to root), and we count how many occur in clean/malware.

Shows a 'learning curve' which lets one spot under-training.

9.9 Reflection on Geometric Exercises in Origami: Ida

I have bene working in Origami for 15 years, so what have I learned. Earliest record is a folding frog in circa 1174. He has a reproduction of the page, but cannot actually read it.

Play Origami Forgs, cranes, etc., see dramas of saikakui Ohara (Japan's Shakespeare, c. 1682)

Protocol Origami The rule book was Teijoi Ise 1764.

Matematical Origami Geometrical Exercises in paper folding by T Sudara Row. Huzita–Justin "elementary fold principle" 1986. Tools: a ruler with unit. reflection by foldng, transfer of line segements by ruler, marking of points is implciitly allowed. 1989 was first Origami conference. Fold lines are easy,but how do you determine. Note that operation 6, "slide P along L until X", is legal. Allows

1. angle trisection
2. cube doubling
3. regular heptagon

Shows some of his construction: fewer intermediate points than [**?**]. Need to show that sides are equal and angles are right angles. Assumes non-collinearity, then does a w.l.o.g. (see section 7.8) coordinatisation.

9.10 GDML: Watt

What is a Library? A building in Alexandria? A collection of books and journals? A collection of subscriptions and databases? Look at what Google has done, with Streetview, or Google books. EuDML shows a quite significnat collection of journal backfiles.

But maybe we should more ambitious. A digital map is more powerful that a paper one: one can query it. Can we do the same? Of course, people ask why mathematics? Permanence is one — look at dates of references. It is (rather) precise. One could ask "is this result known" (sometimes one hopes for yes, and sometimes for no!).

Shannon tells us about information, but this is not the same as knowledge. Hence we start talking about Kolmogorov complexity.

IMU WG contains librarians, but I come from computer algebra. In CA, "knowledge" tends to be hard-coded. So in Maple $0 \cdots x \Rightarrow x$ is hard-wired.

There's also the world of formal mathematics. Note that [**?**] and its editorial disclaimer changed the shape/rôle of refereeing. But there's the de Bruijn factor: < 10 but $>> 1$ currently.

We probably want subject-based simplification rules, e.g. $1+x+x^2$ for Taylor series, x^2+x+1 for ideal theory, etc.

Note that collecting the PDFs is largely done.Making sense of them is a different issue. Don't forget the importance of bibliographic data: helps to groups mathematics into "schools". Note the number of hand-curated databases, OEIC, DLMF etc. Bizarrely, "encyclopedia of triangle centres". Libraries in proof system, which are unfortunately not ineroperable.

A rather different area is courseware, e.g. Waterloo and MIT, Coursera, Khan Academy.

1. Assemble physical documents (not perfectly done, and some are hard)
2. Capture page sets (born digital or scanned), but needs symbol reconstruction etc.
3. Sieze the metadata (not trivial, in fact a whole industry)
4. Semantic capture — comes in various levels, and can be context-sensitive

Good TEX versus bas TEX: parsing most TEX will fail.

Rate of growth of languages and, worse, libraries. All sorts of language issues: scoping etc. See [**?**] to see how even "equation" is actually undecidable. Claims that handwriting disambiguation is *not* that much harder than the other parts of the understanding scenario. Note that we need to be able to talk about false things. Standards in convergence: MathML and OpenMath.

Challenge 1 *To extract and mechanize the world's mathematical knowledge.*

9.11 On complexity of the detection problem for bounded length polymorphic viruses: Lita

Encoding QSAT formuale xa represent $x1$ and $xbbbb$ is $\overline{x_3}$, with concatenation for $\wedge$ and $\vee$ (but clauses is $(\cdots)$. Therefore our grammar can be hard (PSPACE claimed).

[Cohen1985] showed virus detection is undecidable; [Spinellis2003] showed bounded-length viruses for which exist detection is NP-complete. The viruses produce solve the satisfiability problem.

9.12 Partial finitely generated bi-ideals: Bets

Partial words appear in DNA clmputing, data communication etc.

15 uears ago Blanchet-Sfafri et al. worked in this are. DNA has some structure, as do b-ideals so can we unify?

Definition 11 *Aninfinite word is periodic if it is u^ω. A seuqnece of finite word sis a bi-idela if ...*

Let $v_0 = u_0; v_{i+1} = v_u u_{i+1} v_i$. The limit o fthis sequence is called a bi-ideal, and the u_i are a basis. An inifite word is called *recurrent* if each of its factors occurs infinity often

Proposition 2 *An infinite word is recurrent iff it is a bi-ideal.*

A bi-odea; s f.g. if the basis sequence is periodic. Write$\langle u_1, u_1, \ldots, u_m \rangle$ in this case.

Proposition 3 (Lorencs2012) *If x is a bi-ideal geneated by (u_n), then it is also geneated bu $(u_i u_{i+1})$.*

Theorem 22 *If x is generated by (u_n). Defined ...*

Theorem 23 *There is only one way to fill the finite number of holes for a given finitely generated bi-ideal.*

Example 13 (counter) *We have a bi-ideal with 0 on every odd position. But ...*

9.13 Combinatorics of hybrid sets: Watt

Gneerlisation inclusion-exclusion — "selling before buying". n-fold piecewise functions with k of them, can give n^k combinations, especially if the bouding points are symbolic. Hybrid sets, multisets with negative counts, solve this. Also makes sense of "generalised intervals".

Instead of $\cup$ etc. use $\oplus$ etc., then you get a decent algebraic structure (Watt did define it). If H is a hybrid set, S_H^+ as those element with positive weight. $\sum_{u \in U} |H(u)|$ is weight, $\sum_{u \in U} H(u)$ is cardinality (so cardinality $0 \neq$ empty). However, subsets are harder to define.

G is a *natural subset* of H if $S_G \subset S_H$ and $G(u)$ is closer/as close to 0 as $H(u)$. Counting patritions is unsolved, even for multisets.

9.14 Various enhancements of extended Hensel construction for sparse multivariate polynomials: Sasaki

EH is $f_n(u) = 0$ or $F(xmu0 = 0$, where we are looking at $F \in \mathbf{Q}[x, u]$. G is our previous Generalised, W=Wang, E=now. Polynoial with terms like $x(y^k + z^k)$ with

$k \in 10, 20, .., 50$. Large savings for large k. Consider Newton polynomial: do we separare (minimal) or not (maximal) on a given line?

As CASC, compared Mathematics with his GAL method.

Let $\Gamma_{i,j}$ be Gröbner basis of G_i, G_j. $\Gamma_{i,j} \cap \mathbf{Q}[u] = \{\hat{G}_{i,j}\}$ only. Three ideas

1. $\delta F^{(k)}$: regard δP is it is were 0. Essentially the best simplification policy is to unify to a common denominator
2. A divide and conquer method: always very useful
3. Sometimes good, sometimes not

But showed sometimes a $\times 500$ improvement over CASC for ideas 1–3 combined.

Chapter 10

27 September2016

10.1 HPC for Environmental Simulations: Mundani

Showed KAUST's 13k×3k wall-sized screen for visualising simulations. Six-stage process from modelling to embedding: will focus on 2 (numerical modelling) to 4.

- Better hardware — work harder;
- Better algorithms etc. — work smarter;
- parallelism — get some help

10.1.1 Modelling

"Spacetrees" are the generalisation of quadtrees/octrees. Much better than uniform subdivision: typically reduces $O(N^2)$ to $O(N^2)$ for 3D problems. BREP model of a power station has 12M faces. Did a complete model of an operation theatre $6.3 \times 6.25 \times 3.50m^3$: need to take heat away from the hot lamps, but circulation must not drag pollutants to wards patient. Very complex geometry inside a populated operating theatre. Also BMW HQ — a large building in shape of a 4-cylinder engine. Multiscale model: building in its surrounds (flood risk) right down to rooms. Keep the octree in memory, but the leaves only contain pointers into the full model on disc.

Flood prediction requires coupling the 3-D model of Munich with a 1-D model of the sewerage system. Curious boundary conditions at the manholes.

10.1.2 Foundations

Cache is not just "nice fast memory", it is fundamental for performance. Hierarchy: Program/process/block/instrction/sub-instruction. Importance of instruction pipelining. Questions of multiple I-units — typically OpenMP. Vector units have limited applications, but useful for much of what we do.

Note also the Sandybridge etc. NUMA problem. Showed a graph of various compilers and -O0 gives 0.8 Gflop consistently. Everything -O3+ achieves 1 Gflop for very large

loop sizes. But 2–2.2 Gflops when in L3 cache, Intel (but not GCC) gets up to 7.5 Gflop when fits in L1. Note that therefore we need to optimise on the local machine first.

Looks at Top 500, especially Top 10. He's quite positive about #1.

10.1.3 Multigrid solvers

These are fundamental. Imagine a system of 10^9 unknowns. Gauss implies 10^{27} operations, 10^{10} seconds $\approx$ 300 years on 100Pflop machine. Hence iterative solvers. CG and multigrid are harder to parallelise, but State-of-the-Art.

Let w_k be the kth eigenvector of the iteration method, then the error was $\sum c_i w_i$,and the nth error is $\sum c_k \lambda_k^n w_k$. Eigenvalues close to 0 are high frequency parts of the error, so it is the low frequency parts of the error that survive. But these are also visible on a coarser grid. Hence

1. "fine grid to kill h.f. terms", $\nu_1 \in \{1, 2, 3\}$ times
2. aggregation: "coarse grid to kill l.f. terms", $\nu_2 \in \{1, 2, 3\}$ times
3. "map back (prolongation) and refine". $\nu_3 \in \{1, 2, 3\}$ times
4. And possibly repeat.

Of course we are not limited to two levels. Shows a 1/16/64 model. But each cell also has a halo of "ghost cells". Get vertical communication of aggregation/prolongation, and horizontal communication of update of ghost layers.

Might use a space-filling curve to map the cells to a curve (?why). Lebesgue's curve has a very simple inversion. Hence if we map blocks that are neighbours on the SFC to the same processor, we minimise inter-node communication. For am example showed some solutions on SuperMUC. One example (20K cores for 20 minutes) used 2500kWh — household for a year.

The Munich sewerage example was inspired by what happened in a shopping ,all in Pasing 2011: shops ruined the evening before the opening, even though the rainstorm wasn't nearby.

10.1.4 Private Conversation

JHD asked if he had ever modelled the "urban heat island" effect. he hadn't directly, but the BMW simulation was somewhat similar. It resembled a multigrid, but one started with the coarse grid, to get a solution for natural air flow, solar gain etc. Then on a finer grid, one modelled (?a sample of) rooms, and fitted them with curtains/blinds/air conditioning etc. Then back to the coarse grid to see how this changed the overall properties, then if necessary fine grid again. There's nothing really published on this, but he'll send JHD some working papers.

10.2 : Cristea

Look at Google books as an example of digital conservation versus copyright. The problemis not so much physical depreciation of digital objects as moral depreciation (of the curators).

Protage FP7 project

DARIAH a network of institutions, directed at the humanities. Standards, good practices and Text Encoding Initiative (TEI). Concept of "digital surrogates" — summaries

Google books 130M titles. 20% public domain. 10–15% are in print. 25M scanned books. "Our goal is to improve access to books, not to replace them".

Europeana hriacc+others. 2000 institutions cooperating 10M+ digital obects

-e-read]my COST project.

10.2.1 Technical narrative

[Eleni Gallotou, Using digital corpora for preserving and processing cultural heritage texts: a case study, Literary Review 2014] — mostly old books stored in monasteries etc. Note Greek in diachronicity. Typically manuscripts of 13-19 century, printed books.

- Manual classifiation
- Digitisation (camera+tripd), 113k pages so far. But direct OCR is very poor, hence direct search of keywords in page images (word spotting), using a Stuttgart Finite Stage Transnducer tool to build a morphological generator for the early stages of modern Greek.
- Access (?)

My project looked at versions of the Quo Vadis corpus. Looks an anaprohic and non-anaphoric relationships. Vast range of anaphoric references in a complex social story.

Also "Mapping Books" looks at Orhan Pamuk. A "Mapped Book" has connections betweenthe books and the (real or imaginary) world. In particular, cna we connect youngsters more to books this way. No *a priori* knowledge – system reads the book, identifying relationships, and using geography texts.

Can use this to build communities of readers.

Projects COROLA to curate and serve the [Romanian] language (1945–). I was amazed how many owrds I don't know. Knowing 25% of the words in the [Romanian] dictionary makes me a very cultivated guy. Note that 6–7 [Romanian] libraries have to keep copies of books, but this law only dals with apper copies of paper books. Metadata requirements.

POS tagging, Noun Phrase chunking, Syntactic parsing. DRUKOLA joint DE/RO project, funded by Alexander von Humbolt Foundation. based on massive German curpos.

Q what about incorrect usage in your corpus

A The language is the language as it is: new orthography or old orthography etc. We need to be sensitive to changes in the language.

10.3

Need < 100ms resposne time.

Our training algorithm is a One Side Class (OSC) perceptron. Therefore 0 FP on training set.

Domain : IP address (bad), whitelisted (Google) etc.

Extension executable etc.

...

Classic approach: trained on 10 hours, then run on 1000. OK forfirst 200 hours, then suddenly detection drops, presumably by a new spam campaign. Average rate 68% detection. FP 0.35%

Hence train two OScs: one positive and one negative. They only disagree on 0.64%. These are passed (offline) to an external system for classification, and to update the training data. 82% detection rate. FP 0.75%.

10.4 Identifying DGA-based botnets using network anomaly dtection

Domain geneating Algorithms is the key to controlling botnets. The bot generates domain names based on a seed, makes queries and tries to connect to a successful IP to recieve instructions. hence access to raw network data might be useful. They use a variety of TLDs. Bot's query at intervals: mean between 0.0348 seonds (patriot) and 1.7131 (cryptolocker). So how does one tell if a domain name is "random" – we used a 10K word English dictionary and look for "too many" unknown triples. Analysing time frequency is also useful.

$$p(|X - \overline{X}) > k\sigma) \leq \frac{1}{k^2}$$

Detects five big families, and, with a significant level of 5%, produces no FPs on a 150-host network. Note, however, that the router capturing the DNS data has to be before a DNS server.

10.5 Irrelevance in incomplete fuzzy arithmetic: Franzoi

A fuzzy n-tuple $X_1 m \ldots, X_n$ is defined by its distribution function $f_X(x) = f(x) : \mathbf{R}^n \to [0,1]$. Montecatini lemma impies that in the joint distribution approach, the results on exact numbers are the usual ones.

A T-norm is a commutative monotime function $[0,1] \times [0,1] \to [0,1]$ which is 1-absorbing: $\forall u : uT1 = 1Tu = 1$.

But do we need this? A *join* is a function such that ..., weaker than a T-norm. And this can be defined on incomplete fuzzy quantities as well.

10.6 Levenberg-Marquardt learning algorithms

Quaternion feed-forward neural networks. We know what happens for $\mathbf{R}$ and $\mathbf{C}$ (good results). Quaternion-based NNs have increaing interest over the last few years, with many applications. Need **HR**-calculus. Used in signal processing applicatiob (well-known benchmark). LM had a precision gain of 8.94, versus Gradient Descent 4.51. (this metric is logarithmic). For 3D Lorenz problem, had 31.45 versus 7.56.

10.7 Behavioural Trading Systems for Stock Markets: Tirea

Aims: predict trend direction, prices. Multi-agent system. Example: BMW stock for 20 years, Elliott Wave Evaluation.

Looked at Fibonacci and GANN approach. Compute sqrt of opening price: choose two inetgers above and below. square of first is centre of Gann square. Consier a rotation of 45 degreees, quare root of the number +0.125, compute the square root of the obtained value — left grid to the centre one. Continue until GANN squate of 9 is found. Hence deduce resistance points and support points. At 99.95% of ?? buy, at 100.05% of ?? sell.

We create dan indenendent system capable of making deicsion. Wanting an interface to this multi-agent system. Android API for Jade agenda

10.8 Parallel Heuristics for Equation Preconditining

Aimed at reducing the average bandwidth f sparse matrices.

$$mbw(A) = \frac{1}{k} \sum_{a_{ij} \neq 0} \cdots .$$

From the matrix point of view, we are permuting rows/columns, whereas from the graph point of view, ... Two algorithms. One offers $\times 3$ and two $\times 7.5$ in ?execution time of matrix operations?.