Computing with triangular families of polynomials, an overview

Éric Schost, ORCCA, UWO

Triangular representations

Triangular set: $\mathbf{n}$ polynomials in $\mathbf{n}$ unknowns over a field $\mathbb{K}$, of the form

$$\mathbf{T} \left| \begin{array}{l} T_n(X_1, \ldots, X_n) \\ \quad\vdots \\ T_2(X_1, X_2) \\ T_1(X_1), \end{array} \right.$$

with the conditions

- T_i is monic in X_i,
- (optional) the ideal $\langle \mathbf{T} \rangle$ generates a separable extension of $\mathbb{K}$.

  The system has no multiple root.

Note:

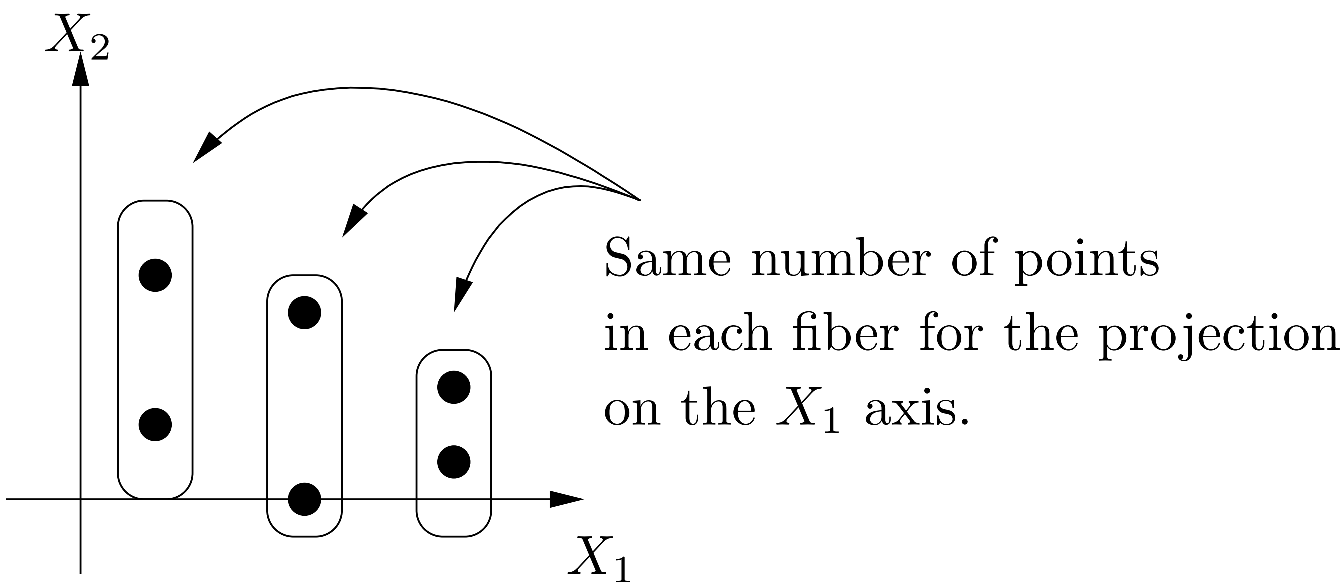
- particular case of a zero-dimensional Gröbner basis.
- we can hide parameters in the base field.

Representation of algebraic sets

Example: the family

$$\left| \begin{array}{l} T_2 = X_2^2 + (\cdots)\, X_2 + (\cdots) \\ T_1 = X_1^3 + \cdots \end{array} \right.$$

defines an equiprojectable variety (Aubry-Valibouze) of the form



Some examples

Using symmetries

Let $S(X)$ be a self-reciprocal polynomial of degree d, with $S(0) \neq 0$:

$$S(X) = X^d S\left(\frac{1}{X}\right).$$

Example: $\mathbf{1}X^6 - \mathbf{5}X^5 + \mathbf{6}X^4 - \mathbf{9}X^3 + \mathbf{6}X^2 - \mathbf{5}X + \mathbf{1}.$

Suppose we want to factor S. In order to make the factorization easier, we introduce $Y = X + \frac{1}{X}$. Then,

Using symmetries

Let $S(X)$ be a self-reciprocal polynomial of degree d, with $S(0) \neq 0$:

$$S(X) = X^d S\left(\frac{1}{X}\right).$$

Example: $\mathbf{1}X^6 - \mathbf{5}X^5 + \mathbf{6}X^4 - \mathbf{9}X^3 + \mathbf{6}X^2 - \mathbf{5}X + \mathbf{1}$.

Suppose we want to factor S. In order to make the factorization easier, we introduce $Y = X + \frac{1}{X}$. Then,

1. consider the equations

$$\begin{cases} Y - \left(X + \frac{1}{X}\right) \\ X^6 - 5X^5 + 6X^4 - 9X^3 + 6X^2 - 5X + 1 \end{cases}$$

Using symmetries

2. change the order of X and Y

$$\begin{cases} X^2 - YX + 1 \\ T: \quad Y^3 - 5Y^2 + 3Y + 1 \end{cases}$$

Using symmetries

2. change the order of X and Y

$$\begin{cases} X^2 - YX + 1 \\ T: \quad Y^3 - 5Y^2 + 3Y + 1 \end{cases}$$

3. factor T (or find a single factor)

$$\begin{cases} X^2 - YX + 1 \\ Y^2 - 4Y - 1 \end{cases}$$

Using symmetries

4. recover the factors of S by changing back the order

$$\begin{cases} Y - \left(X + \frac{1}{X}\right) \\ X^4 - 4X^3 + X^2 - 4X + 1. \end{cases}$$

Using symmetries

4. recover the factors of S by changing back the order

$$\begin{cases} Y - \left(X + \frac{1}{X}\right) \\ X^4 - 4X^3 + X^2 - 4X + 1. \end{cases}$$

Motivation: Point-counting in genus 2 (with P. Gaudry):

- one has to factor a polynomial over a large finite field;
- for help, one knows a polynomial $R(X)$ that plays the role of $X + \frac{1}{X}$.

Power series multiplication

The monomial ideal

$$\mathbf{T} = \left| \begin{array}{l} x_n^{d_n} \\ \vdots \\ x_2^{d_2} \\ x_1^{d_1} \end{array} \right.$$

is triangular.

There seems to be no straightforward algorithm to multiply modulo $\langle \mathbf{T} \rangle$ efficiently.

Polynomial multiplication

Reductions:

- to multiply polynomials of degree d, enough to multiply series mod X^d;
- we can take $d = 2^k$.

From uni- to multivariate (with $d = 8 = 2^3$)

- write $X = X_0$, and introduce X_1, X_2;
- use the equality between ideals

$$\left| \begin{array}{l} X_2 - X_0^4 \\ X_1 - X_0^2 \\ X_0^8 \end{array} \right. \quad = \quad \left| \begin{array}{l} X_0^2 - X_1 \\ X_1^2 - X_2 \\ X_2^2 \end{array} \right.$$

- change of basis is free via base-2 decomposition of indices.

Artin-Schreier (characteristic 2)

To handle degree-2 extensions

- of the form $\mathbf{F}[X]/\langle X^2 - X - \alpha \rangle$,
- where char($\mathbf{F}$)$= 2$.

Algorithms such as Couveignes' generate towers

$$\mathbf{T} = \left| \begin{array}{l} X_n^2 - X_n - \alpha_n(X_1, \ldots, X_{n-1}) \\ \qquad \vdots \\ X_2^2 - X_2 - \alpha_2(X_1) \\ X_1^2 - X_1 - \alpha_1 \end{array} \right.$$

and we have to compute modulo such $\mathbf{T}$'s.

Implicitization

The field $\mathbb{K}$ may be a rational function field $\mathbb{L}(Y_1, \ldots, Y_r)$: leads to a representation of the generic solutions of systems of positive dimension.

Example:

$$
\begin{aligned}
x &- \frac{1}{3}\frac{pq^2+q^2+q+p^2q-6pq+p+p^2}{pq},\\
y &- -\frac{1}{9}\frac{(q-1+p)(-q-1+p)(-q+1+p)}{pq},\\
z &- \frac{1}{36}\frac{(p+1)^2(q+1)^2(q+2q^2+q^3+p+pq^3+2p^2-p^2q^2+p^3+qp^3)^2}{p^2q^2(p+q+1)^4}
\end{aligned}
$$

is triangular in $\mathbb{L}(p,q)[x,y,z]$, and represents the "generic points" of a variety defined over $\mathbb{L}$ by equations in $\mathbb{L}[p,q,x,y,z]$ (provided by I. Kogan.)

Implicitization

The field $\mathbb{K}$ may be a rational function field $\mathbb{L}(Y_1, \ldots, Y_r)$: leads to a representation of the generic solutions of systems of positive dimension.

Example:

$$q - c_0(x, y, z, p)$$

$$p^2 + b_1(x, y, z)p + b_0(x, y, z)$$

$$x^{12} + a_{11}(y, z)x^{11} + a_{10}(y, z)x^{10} + \cdots$$

is triangular in $\mathbb{L}(y, z)[x, p, q]$, and represents the "generic points" of the same variety.

Our questions

Previous work

1932 Ritt

1978 Wu

Characteristic sets (Chou, Gao, Wang, Hubert, . . .)

1987 Duval

Constructible sets (Gomez-Diaz, Dellière, . . .)

1991 Lazard

Dimension zero

1992 Lazard

1991 Kalkbrenner

Regular chains (Moreno Maza, Aubry, . . .)

Usually complex algorithms, especially in positive dimension.

Where the question stands

Already in dimension zero...

Compared to Gröbner bases?

- harder to compute than degree bases?
- faster arithmetic?

Compared to primitive element representations?

- more structure
- slower arithmetic?

Polynomial multiplication

Polynomial multiplication is a basic problem, with a variety of answers.

$$(F, G) \in \mathbb{K}[x] \mapsto FG.$$

- naive product, Karatsuba, Toom-Cook, FFT's,
- cost written $\mathsf{M}(d)$

Matrix multiplication

Another fundamental question is matrix multiplication:

$$(A, B) \in \mathscr{M}_n(\mathbb{K}) \ \mapsto \ AB.$$

An exponent of matrix multiplication is an ω such that matrix multiplication can be done in $O(n^\omega)$.

Known results

Using

- divide-and-conquer,
- Newton-Hensel lifting,
- baby steps / giant steps, ...

algorithms can be designed on top of polynomial and matrix multiplication, and their complexity can be expressed in terms of M and ω.

Goal: develop such a family of algorithms for triangular representations.

with Dahan, Jin, Li, Moreno Maza, Pascal, Wu, Xie

How:

- understand the hierarchy of problems;
- at the software level, concentrate the effort on a few key subroutines.

Dimension zero

Setting

Input data:

- $\mathbf{T}$ is a triangular set in $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \dots, X_n]$;
- the natural measure of complexity is $\delta_{\mathbf{T}} = d_1 \cdots d_n$, with $d_i = \deg(T_i, X_i)$.

Ideal target:

- algorithms for basic operations with $\mathbf{T}$ that would have cost $O\tilde{~}(\delta_{\mathbf{T}})$.

If this is too difficult...

- algorithms of complexity $O\tilde{~}(\mathsf{F}(n)\delta_{\mathbf{T}})$ or $O\tilde{~}(\mathsf{F}(n)\delta_{\mathbf{T}}^{\mathbf{k}})$.

We use $O\tilde{~}(\ )$ to denote the omission of logarithmic factors.

Multiplication

Input: A and B in $\mathbb{K}[\mathbf{X}]$, reduced modulo $\langle \mathbf{T} \rangle$.

Output: AB modulo $\langle \mathbf{T} \rangle$.

Cost: $O\tilde{}(4^n \delta_{\mathbf{T}})$.

Algorithm: $(A, B) \;\mapsto\; C = AB \in \mathbb{K}[\mathbf{X}] \;\mapsto\; C \bmod \langle \mathbf{T} \rangle$.

The reduction modulo $\langle \mathbf{T} \rangle$ extends the algorithm for univariate Euclidean division (Cook, Sieveking, Kung).

- a direct recursive approach leads to $3^n \mathsf{M}(d_1) \cdots \mathsf{M}(d_n) \simeq O\tilde{}((3\mathsf{k})^n \delta_{\mathbf{T}})$.
- we use a mixed dense / recursive algorithm.
- practical algorithm.

What's missing: getting rid of the exponential factor 4^n.

Inversion

Input: $A \in \mathbb{K}[\mathbf{X}]$, reduced modulo $\langle \mathbf{T} \rangle$.

Output: A^{-1} modulo $\langle \mathbf{T} \rangle$ (supposing it exists), error otherwise.

Cost: $O\tilde{}(\mathsf{K}^n \delta_{\mathbf{T}})$.

Algorithm: a recursive Euclidean algorithm.

- if $\langle \mathbf{T} \rangle$ is maximal, no big problem (theoretically).
- else, leading terms can be zero-divisors; this induces splittings and requires an effective multiple reduction:

$$\mathbb{K}[\mathbf{X}]/\langle \mathbf{T} \rangle \to \mathbb{K}[\mathbf{X}]/\langle \mathbf{U}_1 \rangle \times \cdots \times \mathbb{K}[\mathbf{X}]/\langle \mathbf{U}_m \rangle.$$

- complex recursive algorithm, using fast algorithms for GCD, modular reduction, coprime factorization.

What's missing: a theoretically / practically good algorithm.

Change of order, dimension zero

Input: $\mathbf{T}$ and a target order on the variables.

Output: A triangular set $\mathbf{T}'$ for the target order such that $\langle \mathbf{T} \rangle = \langle \mathbf{T}' \rangle$ (if it exists).

Cost in two variables:

- $O(\delta_{\mathbf{T}}^{(\omega+1)/2})$, where ω is the exponent of linear algebra.
- $O\tilde{}(\delta_{\mathbf{T}})$ bit-cost over finite fields.

Algorithm.

- Leverrier-like algorithm, using trace computations (Rouillier, Diaz Toca-Gonzalez Vega),
- fast modular composition (Brent-Kung),
- over finite fields, Kedlaya-Umans.

What's missing: an algorithm in $O\tilde{}(\mathsf{K}^n \delta_{\mathbf{T}})$.

Positive dimension

Setting

Basic object:

- $V \subset \overline{\mathbb{K}}^n$ is a variety of dimension r and degree Δ, defined over $\mathbb{K}$.
- V irreducible (for simplicity).

Representations of V:

- Given an order on $\mathbf{X}$, there exists a partition of $\mathbf{X}$ into free variables $\mathbf{Y}$ and algebraic variables $\mathbf{Z}$, and $\mathbf{T}$ in $\mathbb{K}(\mathbf{Y})[\mathbf{Z}]$, such that
  - ⋆ $\mathbf{T}$ is a triangular set for the induced order on $\mathbf{Z}$,
  - ⋆ $\langle \mathbf{T} \rangle = I(V) \cdot \mathbb{K}(\mathbf{Y})[\mathbf{Z}]$.
- $\mathbf{T}$ describes the generic points of V.

Target:

- algorithms of complexity polynomial in Δ, when possible.

Degree bounds

Question: what are the degrees that can appear in $\mathbf{T}$?

Answer: the degrees in the algebraic variables are $\leq \Delta$; the degrees in the free variables are $\leq 2\Delta^2$.

Remark: instead of $\mathbf{T}$, one can use $\mathbf{U}$, with

$$U_i = \frac{\partial T_1}{\partial X_1} \cdots \frac{\partial T_{i-1}}{\partial X_{i-1}}\, T_i \mod \langle T_1, \dots, T_{i-1} \rangle.$$

Then all degrees in U_i are $\leq 2\Delta$.

What's missing: bit-size, when $\mathbb{K} = \mathbb{Q}$.

Lifting techniques

Input:

- a system $\mathbf{F} = (F_1, \ldots, F_{n-r})$ such that $V \subset V(\mathbf{F})$ and $V \not\subset V(\mathrm{Jac}(\mathbf{F}, \mathbf{Z}))$.
- the specialization $\mathbf{T}(0, \mathbf{Z})$ (assumed to be lucky).

Output: $\mathbf{T}$.

Algorithm: compute expansions of $\mathbf{T}$ modulo $\langle \mathbf{Y} \rangle^{2^i}$, and recover the coefficients in $\mathbb{K}(\mathbf{Y})$ by multivariate rational function reconstruction.

Complexity: combines the costs of most previous subroutines, not polynomial in Δ.

Remark: also works to lift $\mathbb{Z}/p\mathbb{Z} \to \mathbb{Q}$.

Change of order, positive dimension

Input: A triangular set $\mathbf{T}$ in $\mathbb{K}(\mathbf{Y})[\mathbf{Z}]$ and a target order on $\mathbf{X}$.

Output: The specialization $\mathbf{T}'(0, \mathbf{Z}')$.

Algorithm:

- finding the free / algebraic variables for the target order;
- going from $\mathbf{T}(0, \mathbf{Z})$ to $\mathbf{T}'(0, \mathbf{Z}')$ changing one variable at the time.

Complexity: polynomial in Δ. Combines the costs of most previous subroutines.

Summary

As of now: the previous algorithms are sufficient to treat several applications.

- implementations of low-level algorithms in C.
- lifting and change of order in positive dimension available in Maple (RegularChains: Lemaire, Moreno Maza, Xie).

Ongoing work: solving general systems, by incremental intersection.

- previous algorithms
- (sub)resultant techniques

Complexity of multiplication

In one variable

Quotient and remainder: Euclidean division

$$T, B \mapsto B \bmod T,$$

with $\deg T = d$ and $\deg B \leq 2d$ costs $2\mathsf{M}(d) + O(d)$ base ring operations, assuming precomputations (Cook, Sieveking, Kung).

The trick: power series division at ∞.

Modular multiplication: $3\mathsf{M}(d) + O(d)$

In several variables

Let now $\mathbf{T}$ be a triangular set of multi-degree $(d_1, \ldots, d_n)$. Plain recursive:

$$3^n \mathsf{M}(d_1) \cdots \mathsf{M}(d_n).$$

For $\mathsf{M}(d) = \mathsf{k}\, d \log(d) \log\log(d)$, this is essentially $(3\mathsf{k})^n \delta_{\mathbf{T}}$.

Remark: this is polynomial in $\delta_{\mathbf{T}}$ (because one can assume all $d_i \geq 2$, so $\delta_{\mathbf{T}} \geq 2^n$).

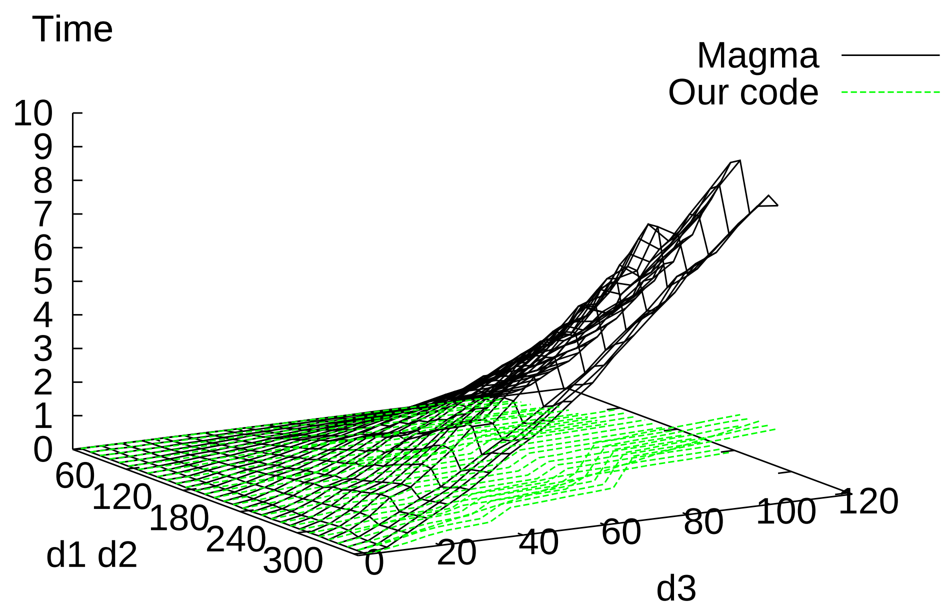
All purpose algorithm: $O\tilde{~}(4^n \delta_{\mathbf{T}})$.

- mixed dense / recursive algorithm, relying on polynomial multiplication.
- we start by expanding the product and reducing it.
- so no way to get better than $2^n \delta_{\mathbf{T}}$.

Practical aspects

The algorithm was implemented by Xin Li:

- C code.
- univariate FFT multiplication and Kronecker substitution.



Beating the 4^n factor

Interpolation techniques (cf. FFT multiplication, Pan).

- When all T_i factor into linear terms, $V(\mathbf{T})$ is a union of $\mathbb{K}$-rational points that form an equiprojectable set.
- Fast univariate evaluation and interpolation has complexity $O(\mathsf{M}(d)\log(d))$.
- Using this, one can multiply polynomials modulo $\langle \mathbf{T} \rangle$ in time

$$O\Big(\delta_{\mathbf{T}} \sum \frac{\mathsf{M}(d_i)\log(d_i)}{d_i}\Big) \subset O\tilde{}(\delta_{\mathbf{T}})$$

Homotopy techniques

When the roots are not in $\mathbb{K}$:

- set up a homotopy with a system that splits

$$V_i = \varepsilon U_i + (1-\varepsilon) T_i$$

- $AB \bmod \langle \mathbf{T} \rangle = \mathtt{subs}(\varepsilon = 1, AB \bmod \langle \mathbf{V} \rangle)$.
- the V_i have roots in $\mathbb{K}[[\varepsilon]]$ (Hensel lemma).

Using the evaluation / interpolation algorithm over $\mathbb{K}[[\varepsilon]]$.

- complexity $O\tilde{}(\delta_{\mathbf{T}} r_{\mathbf{T}})$
- $r_{\mathbf{T}}$ is the needed precision in ε

Homotopy techniques

Nice monomial supports induce low precisions.

Univariate polynomials

- $T_i = T_i(X_i)$
- $r_{\mathbf{T}} = O(\sum d_i)$

Polynomial multiplication

- $T_i = X_i^2 - X_{i-1}$
- $r_{\mathbf{T}} = O(n) \implies \mathsf{M}(d) = O(d \log(d)^2 \log(\log(d))^2 \cdots)$

Artin-Schreier (over $\mathbf{F}_2$)

- $T_i = X_i^2 + X_i + \alpha_i(X_1, \ldots, X_{i-1})$
- $r_{\mathbf{T}} = O(1.5^n)$.