On solving zero-dimensional and parametric systems

Fabrice Rouillier

Fabrice.Rouillier@inria.fr - http://fgbrs.lip6.fr/~rouillie

SALSA (INRIA) project and CALFOR (LIP6) team

Paris, France

General Objectives

$$\mathcal{E} = \{p_1, ..., p_r\}, \mathcal{F} = \{f_1, ..., f_l\}, \text{ with } p_i, f_i \in \mathbb{Q}[U, X]$$

$$U = U_1, ..., U_d \Rightarrow \textbf{parameters}$$

$$X = X_{d+1}, ..., X_n \Rightarrow \textbf{ indeterminates}$$

$$\mathcal{C} = \{x \in \mathbb{C}^n, p_1 = 0, ..., p_r = 0, f_1 \neq 0, ..., f_s \neq 0\}$$

$$\mathcal{S} = \{x \in \mathbb{R}^n, p_1 = 0, ..., p_r = 0, f_1 > 0, ..., f_s > 0\}$$

In part 1 : we suppose that $d = 0$ (no parameter) and that the ideal $\langle p_1, ..., p_r \rangle$ is zero-dimensional (finite number of complex zeroes).

In part 2 : we suppose that $d \neq 0$ and that the ideal $\langle p_1, ..., p_r \rangle_{U=u}$ is zero-dimensional for almost all $u \in C^d$ (general results will be described but only this case will be detailed).

General Objectives (2)

The goal is to propose mathematical objects and algorithms to SOLVE such systems. The end-user queries we are interested in are (see G.-M. Greuel's talk):

- Zero-dimensional systems :
  - count the real / complex roots ;
  - detect multiple points and compute multiplicities ;
  - provide an acurate and certified approximation of the roots ;
  - signs of polynomials at the real roots of a system ;
- Parametric systems :
  - count the real / complex roots wrt parameter's values ;
  - describe geometrically the solutions set ;
  - provide formal expressions of the roots ;
  - provide numerically stable solutions.

General Objectives (3)

Constraints :

- Exact/certified results : a real root is not a complex root with a small imaginary part ... A numerical approximation must be certified with respect to a precision given by the end-user, etc.

- Universal algorithms : being able to check the assumptions (for example zero-dimensional) and the mathematical arbitrary choices (so called "generic" choices).

- Efficiency : computation time (bit operations) but also memory consumming.

Part 1 : Zero-dimensional Systems

Univariate case

We assume that we know how to solve the univariate case (at least in the real case) :

Counting/isolating real roots : Descarte's based methods, Sturm sequences etc.

Evaluating polynomials (or their signs) at the roots of a univariate polynomial.

More details / examples will be given in workshop B2 (Feb 27 - March 3)

A reference book :

Algorithms in Real Algebraic Geometry - Basu Pollack and Roy (Springer)

Notations

- $\mathcal{E} = \{p_1, ...p_s\} \subset \mathbb{Q}[X_1, ..., X_n]$, $\langle \mathcal{E} \rangle$ is the ideal generated by $\mathcal{E}$;
- $\boldsymbol{V}(\mathcal{E}) \subset \mathbb{C}^n$ is the zero set of $\langle \mathcal{E} \rangle$ or equivalently the set of complex solutions of $\{x \in \mathbb{C}^n, p_1(x) = 0, ..., p_n(x) = 0\}$;
- $x \in \mathbb{C}^n$, x_i denotes its i-th coordinate.

Using Gröbner bases (see C. Traverso's talk)

Theorem 1. *Let* $G = \{g_1, ..., g_l\}$ *be a Gröbner basis for any ordering* $<$ *of* $\mathcal{E} = \{p_1, ..., p_s\} \in \mathbb{Q}[X_1, ..., X_n]^s$. *The following properties are equivalent:*

- *For all index* i, $i = 1...n$, *there exists a polynomial* $g_j \in G$ *and a positive integer* n_j *such that* $X_i^{n_j} = LM(g_j, <)$;
- *The system* $\{p_1 = 0, ..., p_s = 0\}$ *has a finite number of solutions in* $\mathbb{C}^n$.
- $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle G \rangle}$ *is a finite dimensional* $\mathbb{Q}$*-vector space*

Proof. According to Theorem 3, the 2nd and 3rd items are equivalent;

First item implies that $\forall i = 1...n$, $\exists m_i$ s.t. $1, ..., X_i^{m_i}$ are lineary dependent (modulo I) and thus implies 3rd item.

3rd item implies that $\forall i = 1...n, \exists m_i$ s.t. $1, ..., X_i^{m_i}$ are lineary dependent (modulo I) so that $\exists p_i \in \mathbb{Q}[X_i]$, $p_i \in \langle G \rangle$. Thus, normalForm$(p_i, G, < 0) = 0 \Rightarrow \exists g_i \in G$ such that LM$(g_i, <)$ divides LM$(p_i, <) = X_i^{\alpha m_i}, \alpha \in \mathbb{N}$. □

Corollary 2. $\mathcal{B} = \{t = X_1^{e_1} \cdot X_n^{e_n}\ ,\ (e_1, ..., e_n) \in \mathbb{N}^n |\ \text{normalForm}(t, G, <) = t\} = \{w_1, ..., w_D\}$ *is a basis of* $\mathbb{Q}[X_1, ..., X_n]/\langle \mathcal{E} \rangle$ *as a* $\mathbb{Q}$*-vector space;*

Counting the solutions (C. Traverso's talk)

Definition 3. *The multiplicity* $\mu(\alpha)$ *of* $\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)$ *is the dimension of the localization of* $\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}$ *at* α *(denoted by* $\left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$*).*

Theorem 4. $\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle} \cong \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$.

Proof.

- $\forall \alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle), \exists e_\alpha \in \frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}$ such that $\sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} e_\alpha = 1$, $e_\alpha\, e_{\alpha'} = 0$ if $\alpha \neq \alpha'$, $e_\alpha^2 = e_\alpha$. (consequence : $e_\alpha(\alpha) = 1$ and $e_\alpha(\beta) = 0$ if $\beta \neq \alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)$)
  - $s_\alpha = \prod_{\beta \neq \alpha} \frac{X_1 - \beta_1}{\alpha_1 - \beta_1}$: $\exists n_\alpha \in \mathbb{N}$ s.t. $t_\alpha = s_\alpha^{n_\alpha}$: $t_\alpha(\alpha) = 1$ and $t_\alpha t_\beta = 0$;
  - $\langle t_\alpha, \alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle) \rangle = \langle 1 \rangle \Rightarrow \exists r_\alpha,\ \sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} r_\alpha t_\alpha = 1$. Set $e_\alpha = r_\alpha t_\alpha$.
- $e_\alpha\left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right) \cong \left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$ □

Corollary 5. *The dimension of* $\frac{\mathbb{Q}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}$ *equals the number of complex zeroes of* $\langle \mathcal{E} \rangle$ *counted with multiplicities.*

Stickelberger's theorem (C. Traverso's talk)

Definition 6. *Let* $h \in \mathbb{C}[X_1, ..., X_n]$;

$$\begin{array}{rccc} m_h: & \frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle} & \longrightarrow & \frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle} \\ & u & \mapsto & \overline{h\,u} \end{array}$$

Theorem 7. *The eigenvalues of* m_h *are the* $h(\alpha)$, $\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)$ *with multiplicity* $\mu(\alpha)$.

Proof.

- $e_\alpha(f - f(\alpha))(\beta) = 0, \forall \beta \in \boldsymbol{V}(\langle \mathcal{E} \rangle) \Rightarrow \exists n_\alpha \in \mathbb{N}, e_\alpha(f - f(\alpha))^{n_\alpha} \in \langle \mathcal{E} \rangle$
- $\Rightarrow$ the restriction of $m_{e_\alpha(f - f(\alpha))}$ to $\left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$ is thus nilpotent so that 0 is its unique eigenvalue (of multiplicity $\mu(\alpha) = \dim\left(\left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha\right)$).
- $\Rightarrow$ conclude using Theorem 4 : $e_\alpha\left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right) \cong \left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$ and $\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle} \cong \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \left(\frac{\mathbb{C}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}\right)_\alpha$. □

Application of Stickelberger's theorem

Corollary 8. *(Stickelberger)*

- $\mathrm{Trace}(m_h) = \sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \mu(\alpha) h(\alpha)$;
- $\mathrm{Charpol}(m_h, T) = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} (T - h(\alpha))^{\mu(\alpha)}$
- $\mathrm{Det}(m_h) = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} h(\alpha)^{\mu(\alpha)}$

Examples of use :

- $h = 1$: $\mathrm{Trace}(m_1) = \dim\left(\frac{\mathbb{C}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}\right)$
- $h = X_i$: $\mathrm{Charpol}(m_{X_i}, T) = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} (T - \alpha_i)^{\mu(\alpha)}$, and thus $\boldsymbol{V}(\mathrm{Charpol}(m_{X_i}, T)) = \{\alpha_i, \alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)\}$

C. Traverso's talk : A first objective is to COMPUTE explicitly the matrices $m_{X_i}, i = 1...n$.

Computing in the quotient algebra

(C. Traverso's talk)

For computing a matrix of m_{X_i}, we need :

- A (monomial) basis $\mathcal{B} = \{w_1, ..., w_D\}$ of $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$ (or equivalently of $\frac{\mathbb{C}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$) : can be deduced from a Gröbner basis for any ordering

- A way for computing the class $\overline{h}$ of $h \in \mathbb{Q}[X_1, ..., X_n]$ in $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$ as a vector wrt $\mathcal{B}$ (denoted by $\vec{h}^{\mathcal{B}}$ from now) : for example the "normalForm".

For example, the colums of the matrix of m_h wrt $\mathcal{B}$ are the coordinates of $\overrightarrow{h\,w_i}^{\mathcal{B}}$.

Counting distinct roots

The matrix of M_{X_i} wrt $\mathcal{B}$ has $\overrightarrow{X_i w_1}, \ldots, \overrightarrow{X_i w_D}$ as columns. If $\mathcal{E}$ has rational coefficients, then $G \subset \mathbb{Q}[X_1, \ldots, X_n]$ and M_{X_i} has also rational entries.

One can compute its characteristic polynomial and apply any univariate solver to isolate all the possible real i-th coordinates of the zeroes of $\langle \mathcal{E} \rangle$.

Definition 9. *(Separating element) Let* $t \in \mathbb{Q}[X_1, \ldots, X_n]$. t *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle)$ *if* $\forall (\alpha, \beta) \in \boldsymbol{V}(\langle \mathcal{E} \rangle)^2, \alpha \neq \beta \Rightarrow t(\alpha) \neq t(\beta)$.

Lemma 10. t *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle) \Rightarrow \sharp \boldsymbol{V}(\mathrm{charpol}(M_t))(\cap \mathbb{R}^n) = \# \boldsymbol{V}(\langle \mathcal{E} \rangle)(\cap \mathbb{R}^n)$

Lemma 11. *Almost all polynomials separate* $\boldsymbol{V}(\langle \mathcal{E} \rangle) \subsetneqq \mathbb{C}^n$.

$\Rightarrow$ Probabilistic algorithm for computing $\sharp \boldsymbol{V}(\langle \mathcal{E} \rangle)$ or $\sharp \boldsymbol{V}(\langle \mathcal{E} \rangle) \cap \mathbb{R}^n$

Lemma 12. $\mathrm{charpol}(M_t)$ *squarefree* $\Rightarrow t$ *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle)$.

$\Rightarrow$ Deterministic filter

A (naïve) deterministic algorithm

Lemma 13. *if* $d = \sharp \boldsymbol{V}(\langle \mathcal{E} \rangle) < D$, $\exists t \in \{ \sum_{i=1}^{n} j^{i-1} X_i, j = 0 \ldots n \frac{d\,(d-1)}{2} \}$ *such that* t *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle)$.

Proof. Let $(\alpha, \beta) \in \boldsymbol{V}(\langle \mathcal{E} \rangle)$, $\alpha \neq \beta$.

- $p_{\alpha,\beta}(T) = \sum_{i=1}^{n} T^{i-1}(\alpha_i - \beta_i) \neq 0$. Thus $\sharp \boldsymbol{V}(p_{\alpha,\beta}) < \infty$ and there exists at most n integers j such that $p_{\alpha,\beta}(j) = 0$.

- the number of distinct couples $(\alpha, \beta) \in \boldsymbol{V}(\langle \mathcal{E} \rangle)^2$ is $\frac{d(d-1)}{2}$, so there exists at most $n \frac{d\,(d-1)}{2}$ intergers j such that there exists $(\alpha, \beta) \in \boldsymbol{V}(\langle \mathcal{E} \rangle)^2, \alpha \neq \beta$ with $p_{\alpha,\beta}(j) = 0$ □

Lemma 14. t *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle)$ *iff*

$$\sharp \boldsymbol{V}(\mathrm{charpol}(m_t)) = \max\left(\sharp \boldsymbol{V}(\mathrm{charpol}(m_{\sum_{i=1}^{n} j^{i-1} X_i})), j = 0 \ldots n \frac{d\,(d-1)}{2}\right)$$

Hermite's quadratic form

Double goal : count the distinct complex/real roots - decrease the number of operations for searching a separating element.

Definition 15. *(Hermite's quadratic form). For* $f \in \mathbb{Q}[X_1, ..., X_n]$ *:*

$$\begin{array}{rccc} q_f: & \frac{\mathbb{Q}[X_1,...,X_n]}{\langle \mathcal{E} \rangle} & \longrightarrow & \mathbb{Q} \\ & \vec{h} & \mapsto & \mathrm{Trace}(m_{fh^2}) \end{array}$$

Theorem 16. *For* $f \in \mathbb{Q}[X_1, ..., X_n]$,

- $\mathrm{rank}(q_f) = \#\{x \in \boldsymbol{V}(\langle \mathcal{E} \rangle), f(x) \neq 0\}$
- $\mathrm{signature}(q_f) = \#\{x \in \boldsymbol{V}(\langle \mathcal{E} \rangle) \cap \mathbb{R}^n, f(x) > 0\} - \#\{x \in \boldsymbol{V}(\langle \mathcal{E} \rangle) \cap \mathbb{R}^n, f(x) < 0\}$

Corollary 17. *Taking* $f = 1$ *:*

- $\mathrm{rank}(q_1) = \#\boldsymbol{V}(\langle \mathcal{E} \rangle)$
- $\mathrm{signature}(q_1) = \#(\boldsymbol{V}(\langle \mathcal{E} \rangle) \cap \mathbb{R}^n)$

Hermite's quadratic form

Set $d = \#\boldsymbol{V}(\langle\mathcal{E}\rangle)$, t a separating element of $\boldsymbol{V}(\langle\mathcal{E}\rangle)$.

- $\exists w'_d, ..., w'_D$ st $\{w'_i\}_{i=1}^D = \{1, t, ..., t^{d-1}, w'_d, ..., w'_D\}$ is a basis of $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle\mathcal{E}\rangle}$;
- $q_f(\vec{h}) = \sum_{\alpha\in\boldsymbol{V}(\langle\mathcal{E}\rangle)} \mu(\alpha) f(\alpha) \Big(\sum_{i=1}^D h_i w'_i(\alpha) \Big)^2$ for $\vec{h} = \sum_{i=1}^D h_i w_i$;
- $q_f(\vec{h}) = (h_1, ..., h_D) \Gamma^t \begin{pmatrix} \mu(\alpha_1) f_1 & 0 & ... & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & ... & ... & \mu(\alpha_d) f_d \end{pmatrix} \Gamma \begin{pmatrix} h_1 \\ \vdots \\ \vdots \\ h_D \end{pmatrix}$

  with $\Gamma = \begin{pmatrix} 1 & t(\alpha_1) & ... & t(\alpha_1)^{d-1} & w_d(\alpha_1) & ... & w_D(\alpha_1) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & t(\alpha_d) & ... & t(\alpha_d)^{d-1} & w_d(\alpha_d) & ... & w_D(\alpha_d) \end{pmatrix}$
- $q_f(\vec{h}) = \sum_{\alpha\in\boldsymbol{V}(\langle\mathcal{E}\rangle)} \mu(\alpha) f(\alpha) (l_\alpha(h))^2$
- if $\alpha \neq \overline{\alpha}$, $\mu(\alpha) f(\alpha) (l_\alpha(h))^2 + \mu(\overline{\alpha}) f(\overline{\alpha}) (l_{\bar{\alpha}}(h))^2 = l_{1,\alpha}(\vec{h})^2 - l_{2,\alpha}(\vec{h})^2$, $l_{1,\alpha}, l_{2,\alpha}$ being linar forms with real coefficients.
- $l_\alpha, l_{1,\alpha}, l_{2,\alpha}$ linearly indep $\Rightarrow$ $\vec{h} \mapsto \sum_{\alpha\in\boldsymbol{V}(\langle\mathcal{E}\rangle)\cap\mathbb{R}^n} \mu(\alpha) f(\alpha) (l(\alpha, h))^2$ has the same signature as q_f.

Variable's elimination

An easy example :

- compute G a Gröbner basis of $\langle \mathcal{E} \rangle$;
- compute $D = \dim_{\mathbb{Q}} \frac{\mathbb{Q}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$ and $\{w_1, ..., w_D\}$ from G;
- compute $\vec{1}$, $\overrightarrow{X_1}$, ..., $\overrightarrow{X_1^{D-1}}$ (normalForm) and check they are lineary independant;
- if so, solve the linear system $\left[\vec{1}, \overrightarrow{X_1}, ..., \overrightarrow{X_1^{D-1}}\right] \beta = \left[\overrightarrow{X_1^D}, \overrightarrow{X_2}, ..., \overrightarrow{X_n}\right]$

  and get a system equivalent to $\mathcal{E}$: $\begin{cases} X_1^D - \sum_{i=0}^{D-1} \beta_{1,i+1} X_1^i = 0 \\ X_2 - \sum_{i=0}^{D-1} \beta_{2,i+1} X_1^i = 0 \\ \vdots \\ X_n - \sum_{i=0}^{D-1} \beta_{n,i+1} X_1^i = 0 \end{cases}$
- **ELSE ??**

Lexicographic Gröbner bases

By change of ordering (FGLM algorithm - see C. Traverso's talk)

Suppose that $\frac{\mathbb{Q}[X_1,...,X_n]}{\langle \mathcal{E} \rangle}$ is computed (Gröbner basis G wrt any monomial ordering $<$ and a monomial basis $\mathcal{B} = \{w_1,...,w_D\}$).

Strategy : compute the image $\vec{m}^{\mathcal{B}}$of monomials m in increasing order wrt the lexicographic ordering until getting a set $D = \#\mathcal{B}$ $\mathbb{Q}$-linearly independant vectors $\{\overrightarrow{w'_1},...,\overrightarrow{w'_D}\}$ ($w'_1 = 1$ and $w'_{i-1} <_{\text{lex}} w'_i, i = 2...D$).

Solve $\left[\overrightarrow{w'_1},...\overrightarrow{w'_D}\right]\beta = \left[\overrightarrow{X_i w'_j}\right]_{i=1...n, j=1...D}$, set $g_{i,j} = X_i w'_j - \sum_{k=1}^{D} \beta_{i,j,k} w'_k$

Theorem 18. *$G' = \{g_{i,j}, i = 1...n, j = 1...D\}$ is a Gröbner basis wrt $<_{\text{lex}}$ of $\langle \mathcal{E} \rangle$.*

Proof. $\forall p \in \langle \mathcal{E} \rangle, \exists g \in G'$ s.t. $\text{LM}(g, <_{\text{lex}})$ divides $\text{LM}(p, <_{\text{lex}})$ since other else $p = \sum_{i=1}^{D} a_i w'_i$ (and thus do not belongs to $\langle \mathcal{E} \rangle$) $\square$

Remarks on lexicographic Gröbner bases

General shape in the zero-dimensional case for $<_{\text{lex}}$:

$$\begin{cases} f_1(X_1) \\ f_2(X_1, X_2) \\ \vdots \\ f_{k_3-1}(X_1, X_2) \\ f_{k_3}(X_1, X_2, X_3) \\ \vdots \\ f_{k_n}(X_1, ..., X_n) \\ \vdots \\ f_{k_n+1}(X_1, ..., X_n) \end{cases}$$

Case of ideals in "Shape position ideals for $<_{\text{lex}}$" (degree$(f_1) = D$) :

$$\begin{cases} f_1(X_1) \\ X_2 - f_2(X_1, X_2) \\ \vdots \\ X_n - f_n(X_1, ..., X_n) \end{cases}$$

This occurs for example when X_1 is separating and $\langle \mathcal{E} \rangle = \sqrt{\langle \mathcal{E} \rangle}$.

Triangular sets (lexicographic Gröbner bases in the zero-dimensional case) :

$$\begin{cases} f_1(X_I) = 0 \\ X_2^{n_2} + f_2(X_1) = 0 \\ \vdots \\ X_n^{n_n} + f_n(X_1, ..., X_{n-1}) = 0 \end{cases}$$

The case of radical ideals

Main remark : if $t \in \mathbb{Q}[X_1, ..., X_n]$ separates $\boldsymbol{V}(\langle \mathcal{E} \rangle)$ and if $\langle \mathcal{E} \rangle = \sqrt{\langle \mathcal{E} \rangle}$, then assuming that T is a new variable, $\langle \mathcal{E} \cup \{T - t\} \rangle$ is in shape position for $<_{\text{lex}}$ with $T <_{\text{lex}} X_1 <_{\text{lex}} ... <_{\text{lex}} X_n$.

Proof : $\text{charpol}(m_t) = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} (T - t(\alpha)) = \text{minpol}(m_t)$ is squarefree.

An algorithm :

- compute G a Gröbner basis of $\langle \mathcal{E} \rangle$ for $<_{\text{DRL}}$, $\mathcal{B} = \{w_1, ..., w_D\}$ a basis of $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$ and $d = \#\boldsymbol{V}(\langle \mathcal{E} \rangle)$ (Hermite's quadratic form);
- compute $p_t = \text{minpol}(m_t)$ for $t \in \{ \sum_{i=1}^{n} j^{i-1} X_i,\ j = 0...n\ \frac{D(D-1)}{2} \}$ until $\text{degree}(\text{squarefree}(p_t)) = d$;
- if $\text{degree}(p_t) = D$ solve $\left[\vec{1}, \vec{t}, ..., \overrightarrow{t^{D-1}}\right] \beta = \left[\overrightarrow{t^D}, \overrightarrow{X_1}, ..., \overrightarrow{X_n}\right]$ and return

  $\mathcal{E}' = \{T_1^D - \sum_{i=0}^{D-1} \beta_{1,i+1} T^i, X_j - \sum_{i=0}^{D-1} \beta_{2,i+1} T^i, j = 1...n\}$

  $\left(x = (x_0, x_1, ..., x_n) \in \boldsymbol{V}(\langle \mathcal{E}' \rangle) \Leftrightarrow x = (x_1, ..., x_n) \in \boldsymbol{V}(\langle \mathcal{E} \rangle)\right)$
- **ELSE ?**

Remarks

There exists algorithms to compute the radical of an ideal but they are inefficient for large problems. They require the computation of several Gröbner bases ;

For some problems, one would also like to compute the multiplicities of the solutions ...

Note that the above algorithm may compute $O(n\ D^2)$ minimal polynomials before concluding with a failure !

At this stage : we do not know how to reduce the problem (efficiently) to an easy univariate one when the ideal is not in "shape position";

Numerical instability : the polynomials in a lexicographic Gröbner basis have HUGE coefficients (excepted the first one) : this often forbids to plug numerical approximations of the roots of the first polynomial into the basis to get an accurate approximation of the coordinates for large examples.

The Rational Univariate Representation (RUR)

Definition 19. *For* $t \in \mathbb{Q}[X_1, ..., X_n]$*:*

- $f_t = \sum_{i=0}^{D} a_i T^{D-i} = \text{charpol}(m_t)$, $\tilde{f_t}$ *square-free part.*
- $\forall v \in \mathbb{Q}[X_1, ..., X_n]$, $g_{t,v}(T) = \sum_{i=0}^{d-1} \text{Trace}(m_{vt^i}) H_{d-i-1}(\tilde{f_t}, T)$, *where*

$$d = \text{degree}(\tilde{f_t}) \text{ and } H_j(\tilde{f_t}, T) = \sum_{i=0}^{j} a_i T^{j-i}$$

Theorem 20. *If* t *separates* $\boldsymbol{V}(\langle \mathcal{E} \rangle)$*, then*

$$\begin{array}{lcl} \boldsymbol{V}(\langle \mathcal{E} \rangle)(\cap \mathbb{R}^n) & \approx & \boldsymbol{V}(f_t)(\cap \mathbb{R}) \\ \alpha = (\alpha_1, ..., \alpha_n) & \rightarrow & t(\alpha) \\ \mathcal{R}_t(\beta) = (\frac{g_{t,X_1}(\beta)}{g_{t,1}(\beta)}, ..., \frac{g_{t,X_n}(\beta)}{g_{t,1}(\beta)}) & \leftarrow & \beta \end{array}$$

and :

- $f_t \in \mathbb{Q}[T]$ *and* $\forall v \in \mathbb{Q}[X_1, ..., X_n]$, $g_{t,v} \in \mathbb{Q}[T]$*;*
- $\mu(\alpha) = \mu(t(\alpha)) = \mu(\mathcal{R}_t(t(\alpha))) = \frac{\widetilde{(f_t')}(t(\alpha))}{\widetilde{(f_t)}'(t(\alpha))}$

This generalizes the "elimination process" to the case of non radical ideals

The RUR : Proof

By construction, $f_t \in \mathbb{Q}[T]$ and $\forall v \in \mathbb{Q}[X_1, ..., X_n]$, $g_{t,v} \in \mathbb{Q}[T]$;

Thus $\boldsymbol{V}(\langle \mathcal{E} \rangle) \approx \boldsymbol{V}(f_t) \Rightarrow \boldsymbol{V}(\langle \mathcal{E} \rangle)(\cap \mathbb{R}^n) \approx \boldsymbol{V}(f_t)(\cap \mathbb{R}^n)$.

By Stickelberger's theorem, $f_t(T) = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} (T - t(\alpha))^{\mu(\alpha)}$.

Thus $\widetilde{f_t(T)} = \prod_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} (T - t(\alpha))$

Consider $g_{v,t} = \sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \mu(\alpha) v(\alpha) \Big(\prod_{\beta \in \boldsymbol{V}(\langle \mathcal{E} \rangle), \beta \neq \alpha} (T - t(\beta)) \Big)$

if t separates $\boldsymbol{V}(\langle \mathcal{E} \rangle)$

- $g_{t,v}(t(\alpha)) = \mu(\alpha) v(\alpha) \Big(\prod_{\beta \in \boldsymbol{V}(\langle \mathcal{E} \rangle), \beta \neq \alpha} (t(\alpha) - t(\beta)) \Big)$
- $\widetilde{(f_t(T)')} = g_{v,1}(T) = \sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \mu(\alpha) \Big(\prod_{\beta \in \boldsymbol{V}(\langle \mathcal{E} \rangle), \beta \neq \alpha} (T - t(\beta)) \Big)$
- $(\widetilde{f_t(T)})' = \sum_{\alpha \in \boldsymbol{V}(\langle \mathcal{E} \rangle)} \Big(\prod_{\beta \in \boldsymbol{V}(\langle \mathcal{E} \rangle), \beta \neq \alpha} (T - t(\beta)) \Big)$

so that $v(\alpha) = \frac{g_{t,v}(t(\alpha))}{g_{t,1}(t(\alpha))}$ and $\mu(\alpha) = \frac{\widetilde{(f_t')}(t(\alpha))}{g_{t,1}(t(\alpha))} = \frac{\widetilde{(f_t')}(t(\alpha))}{\widetilde{(f_t)}'(t(\alpha))}$

The RUR : Proof

To prove the theorem, we finally need to show :

$$\boldsymbol{g_{v,t}(T) = \sum_{\alpha \in V(\langle \mathcal{E} \rangle)} \mu(\alpha) v(\alpha) \Big(\prod_{\beta \in V(\langle \mathcal{E} \rangle), \beta \neq \alpha} (T - t(\beta)) \Big)}$$

$$\boldsymbol{= \sum_{i=0}^{d-1} \mathrm{trace}(m_{vt^i}) H_{d-i-1}(\tilde{f}_t, T)}$$

with $\tilde{f}_t = \sum_{i=0}^{D} a_i T^{D-i} = \prod_{\alpha \in V(\langle \mathcal{E} \rangle)} (T - t(\alpha))$

and $H_j(\tilde{f}_t, T) = \sum_{i=0}^{j} a_i T^{j-i}$

$$\frac{g_{v,t}(T)}{\tilde{f}_t(T)} = \sum_{\alpha \in V(\langle \mathcal{E} \rangle)} \frac{\mu(\alpha) v(\alpha)}{T - t(\alpha)} = \sum_{i \geq 0} \frac{\sum_{\alpha \in V(\langle \mathcal{E} \rangle)} \mu(\alpha) v(\alpha) t(\alpha)^i}{T^{i+1}} = \sum_{i \geq 0} \frac{\mathrm{trace}(m_{vt^i})}{T^{i+1}}$$

$$g_{v,t}(T) = \sum_{i=0}^{d-1} \sum_{j=0}^{d-i-1} \mathrm{trace}(m_{vt^i}) a_j T^{d-i-j-1} =$$

$$\sum_{i=0}^{d-1} \mathrm{trace}(m_{vt^i}) H_{d-i-1}(\tilde{f}_t, T)$$

The RUR : a naïve algorithm

Algorithm :

- compute G a Gröbner basis of $\langle \mathcal{E} \rangle$ for $<_{\mathrm{DRL}}$, $\mathcal{B} = \{w_1, ..., w_D\}$ a basis of $\frac{\mathbb{Q}[X_1, ..., X_n]}{\langle \mathcal{E} \rangle}$ and $d = \#\boldsymbol{V}(\langle \mathcal{E} \rangle)$ (Hermite's quadratic form);
- compute $f_t = \mathrm{charpol}(m_t)$ for $t \in \{ \sum_{i=1}^{n} j^{i-1} X_i,\ j = 0...n\ \frac{D(D-1)}{2}\}$ until $\mathrm{degree}(\tilde{f_t}) = d$;
- compute $\mathrm{Trace}(m_{X_i t^j})$, $i = 1...n, j = 1...d$; and deduce;
- return $g_{t,v}(T) = \sum_{i=0}^{d-1} \mathrm{trace}(m_{v t^i}) H_{d-i-1}(\tilde{f_t}, T)$ for $v = 1, X_1, ..., X_n$

Remarks :

- $\mathrm{Trace}(m_{X_i t^j}) \Rightarrow \mathrm{normalForm}(X_i t^j w_k), i = 1...n, j = 1...D, k = 1...D$
  $\Rightarrow O(n\, D^2\, N), N \gg D = \mathrm{cost\ of\ normalForm}$
- Hermite's quadratic form $\mathrm{Trace}(w_i w_j) \Rightarrow \mathrm{normalForm}(w_i w_j w_k), k = 1...D$
  $\Rightarrow O(D^3\, N),\ N \gg D = \mathrm{cost\ of\ normalForm}$
- $\mathrm{charpol}(m_t) \Rightarrow O(D^4)$ (with a naïve algorithm)
- Theoretical complexity : $\gg O(n\ D^6)$ arithm. operations, "Practical" complexity : $\gg O(D^4 + n\, D^3)$ arithm. operations

The RUR : remarks

For computing RUR, we only need

- a monomial basis $\mathcal{B} = \{w_1, ..., w_D\}$ of $\frac{\mathbb{Q}[X_1, ..., X_n]}{I}$; Suppose that $w_1 = 1$ and $\forall i > 1, \exists k, w_i = X_k w'_j$
- the matrices M_{X_i} of m_{X_i} wrt $\mathcal{B}$;

This input can be provided by Gröbner bases but also by some new alternatives (B. Mourrain and P. Trebuchet's work).

From now we suppose we only have these data as input and that the rational numbers in the M_{X_i} are of binary size t.

The goal is now to design an efficient algorithm (binary complexity) for computing the RUR.

Objectives

RUR = separating element t AND a RUR-Candidate $\{f_t(T),\ g_{t,1}(T), g_{t,X_1}(T), ..., g_{t,X_n}(T)\}$: **if t separates $V(I)$,**

$$\begin{array}{ccc} V(\langle\mathcal{E}\rangle)(\cap\mathbb{R}) & \approx & V(f_t)(\cap\mathbb{R}) \\ \alpha=(\alpha_1,...,\alpha_n) & \rightarrow & t(\alpha) \\ (X_1(\alpha)=\frac{g_{t,X_1}(t(\alpha))}{g_{t,1}(t(\alpha))},...,X_n(\alpha)=\frac{g_{t,X_n}(t(\alpha))}{g_{t,1}(t(\alpha))}) & \leftarrow & t(\alpha) \end{array}$$

preserving **multiplicities and real roots.**

Many variants for computing a RUR-Candidate - few solutions for computing a RUR. This difference is critical for "decision" algorithms.

Can be computed as a lexicographic Gröbner basis when I is radical and X_1 separates $V(\langle\mathcal{E}\rangle)$(Faugère, Yokoyama, ... multi-mod. or p-adic methods).

A major problem is to check that a RUR-Candidate is a RUR or equivalently that t separates $V(\langle\mathcal{E}\rangle)$ for non radical ideals.

From "few" traces

Theorem 21. *Given* $q_1[1] = [\mathrm{Trace}(m_{w_1}), ..., \mathrm{Trace}(m_{w_D})]$ *(first line of Hermite's quadratic form), a* ***RUR-Candidate*** *can be computed in* $O(D^3 + n\,D^2)$ *arithmetic operations.*

Proof. See algorithms below □

The first polynomial

According to Stickelberger's theorem, $\text{Trace}(m_{t^i}) = \sum_{i=1}^{D} \alpha^i$ is the i-th Newton's sum of $f_t = \text{charpol}(m_t) = \sum_{i=0}^{D} a_i T^{D-i}$. One can thus deduce the coefficients of f_t from the scalars $\text{Trace}(m_{t^i}), i = 0...D$ by solving the triangular linear system : $\left\{ (D-i)\, a_i = \sum_{j=0}^{i-1} a_{i-j} \text{Trace}(m_{t^j}) \right\}_{i=0...D}$

Algorithm charpol :

Input $= \boldsymbol{q_1[1] = [\text{Trace}(m_{w_i})]_{i=1...D}}$, $+ \boldsymbol{\mathcal{B}} + \boldsymbol{M_{X_i}, i = 1...n}$

$\vec{y} = [1, 0, ..., 0]$; $M_t = \sum_{i=1}^{D} t_i M_{X_i}$;

For $i = 0...D$ do

- $\text{Trace}(m_{t^i}) = \vec{y} \cdot q_1[1]$
- $\vec{y} = M_t \cdot \vec{y}$ /* at the i-th step $\vec{y} = \vec{t^i}$

Solve $(D-i)\, a_i = \sum_{j=0}^{i-1} a_{i-j} \text{Trace}(m_{t^j})$

$O(D^3)$ operations (note this is a general algorithm for computing the characteristic polynomial of any element in $\frac{\mathbb{Q}[X_1, ..., X_n]}{I}$.

The coordinates

Algorithm coordinates :

Input = data from algorithm charpol + f_t

$\tilde{f_t} = \sum_{i=0}^{j} a_i T^{d-i}$ the squarefree part of f_t

For $i = 0 \ldots d$ $\overrightarrow{H_i(\tilde{f_t}, t)} = \sum_{j=0}^{i} a_j \underbrace{\overrightarrow{t^{i-j}}}_{\text{already computed}}$

For $v \in \{1, X_1, \ldots, X_n\}$

- $[\mathrm{Trace}(m_{vw_1}), \ldots, \mathrm{Trace}(m_{vw_D})] = M_v \cdot q_1[1]^t$
- For $i = \ldots d-1$

  $\mathrm{Trace}(m_{vH_i(t)}) = \overrightarrow{H_i(t)} \cdot \ [\mathrm{Trace}(m_{vw_1}), \ldots, \mathrm{Trace}(m_{vw_D})]^t$

$g_{t,v}(T) = \sum_{i=0}^{d-1} \mathrm{Trace}(m_{vH_i(\tilde{f_t},t)}) T^{d-i-1} = \sum_{i=0}^{d-1} \mathrm{Trace}(m_{X_j t^i}) H_{d-i-1}(\tilde{f_t}, T)$

$O(n\, D^2)$ arithmetic operations

Computing Hermite's quadratic form

$q_1 = [\mathrm{Trace}(m_{w_i w_j})]_{i=1\ldots D}^{j=1\ldots D}$

Remark :

- $\mathrm{Trace}(w_i w_j) = \overrightarrow{w_i w_j} \cdot q_1[1]$;

Proposition 22. *Hermite's quadratic form can be computed from* $q_1[1]$ *performing* $O(\#\mathcal{T}D)$ *arithmetic operations where* $\mathcal{T} = \{w_i w_j,\ i = 1\ldots D,\ j = 1\ldots D\}$.

RUR from "few" traces

Algorithm RUR-Classic :

Input $= q_1[1] = [\mathbf{Trace}(m_{w_i})]_{i=1...D}, + \mathcal{B} + M_{X_i}, i = 1...n$

- $d = \#V(\langle \mathcal{E} \rangle)$ (Hermite's quadratic form); $O(D^3)$
- compute $f_t = \text{charpol}(m_t)$ for $t \in \{ \sum_{i=1}^{n} j^{i-1} X_i,\ j = 0...n \frac{D(D-1)}{2} \}$ until $\text{degree}(\tilde{f_t}) = d$; $O(D^3)$ – practical / $O(n\, D^5)$ – theoretical
- compute $g_{t,v},\ v = 1, X_1, ..., X_n$ $O(n\, D^2)$
- return $f_t, g_{t,v}, v = 1, X_1, ..., X_n$

The theoretical complexity is a worst case : all the possible separating elements excepted the last one are not separating.

In practice, I haven't any example with more than two tries ...

RUR from a multiplication table

Definition 23. $\mathcal{T} = \{\overrightarrow{w_j w_j}, i = 1...D, j = 1...D\}$

Remark : $\#\mathcal{T} < D^2$

Examples :

- from a lexicographic G. basis in "Shape position" : $\#\mathcal{T} = O(D^2)$
- If $D = \delta^n$ (Bezout), $\#\mathcal{T} = O(2^n D^2)$ (ex. when $\#G = n$)

Computing $\mathcal{T}$

Computing $\overrightarrow{w_i w_j}$: $\exists i', k, w_i w_j = X_k w_{i'} w_j \Rightarrow \overrightarrow{w_i w_j} = M_{X_k} \overrightarrow{w_{i'} w_j}$

Requires $O(\#\mathcal{T} D^2)$ arithmetic operations

Computing q_1 from $\mathcal{T}$: for $i = 1...D$, $\mathrm{Trace}(m_{w_i}) = \sum_{i=1}^{D} \overrightarrow{w_i w_j}[j]$ $(O(D^2))$

Theorem 24. *Computing a RUR from $\mathcal{T}$ requires $O(\#\mathcal{T} D^2 + D^3 + n\ D^2)$ arithmetic operations.*

Complexity : remarks

Remark :

- up to now we have counted $O(1)$ each arithmetical operation; This is fine for data of fixed sizes ($\mathbb{Z}/p\mathbb{Z}$, floating point numbers, etc.) but not for integers, rationals, polynomials, series, etc.

For integers :

- The addition is linear in the size of its operands;
- The naïve multiplication is quadratic in the size of its operands; (fast fft-based versions are "close" to be linear (forget the log) for very large integers.
- Data are growing at each operation : $a+b$ has size $O(\max(\log_2(a), \log_2(b))+1)$, $a\,b$ has size $O(\log(a)+\log(b))$

Funny example : "fast" exponentiation may not be so fast ... ($m(a,\,b)$= binary cost of the mult. of and integer of size a by an integer of size b)

$X^{2^n} = x \cdot .... \cdot x$: arithmetic cost $O(2^n)$ - binary cost $O\Big(\sum_{i=1}^{2^n} m(i,1)\Big)$

$X^{2^n} = \big(...(x^2)^2...\big)^2$ arithmetic cost $O(n)$ - binary cost $O\big(\sum_{i=1}^{n} m(2^i,2^i)\big)$

Complexity : choosing the right model

We are working with rationals !

- The (bound on the) growth of coefficients in an addition is greater than in a multiplication !

An example : $[a_1, ..., a_D] \cdot [b_1, ..., b_D]$ a_i, b_i of size t

- with integers, result of size $O(2\,t + D)$
- with rationals, result of size $O(2\,D\,t)$

k iterations $v = M \cdot v$, v (resp. M) a vector (resp. a matrix) of dimension D with entries of size t :

- with integers, result with scalars of size $O(k(\,t + D))$
- with rationals, result of size $O(D^k\,t)$

A more precise model : $\frac{1}{u}[a_1, ..., a_D] \cdot \frac{1}{v}[b_1, ..., b_D]$ u, v, a_i, b_i of size t $\Rightarrow$ result of size $O(2\,t + D)$

For our problem we can assume that the scalars are integers !

Complexity : choosing the right model

Suppose that t is the binary size of the integers in $M_{X_i} = \frac{1}{d_{X_i}} M'_{X_i}$

The multiplication table : at most $\delta = \max(\deg(w_i w_j))$ iterations $v = M \cdot v$.

Growth of coefficients : $t \longrightarrow \delta\,(t + D)$

Bound on the binary cost : $O(\#\mathcal{T} D^2 \delta (t + D))$

Hermite's quadratic form :

- $\mathrm{Trace}(w_i w_j) = \overrightarrow{w_i w_j} \cdot q_1[1]$; size $O(\delta\,(t + D))$ bin. cost : $O(\#\mathcal{T} D \delta\,(t + D))$

Reduction of Hermite's quadratic form : growth of coefficients $t' \longrightarrow D\ t'$ (Rouillier's fraction-free algorithm) binary cost : $O(D^4 \delta\,(t + D))$

First polynomial of the RUR : it is the characteristic polynomial of M_t thus its coefficients are of size $O(D\,t)$ (see G. Villard's survey).

The intermediate iteration : $t^{i+1} = M_t\, t^i$ induces a growth $t \longrightarrow D(t + D)$

(thus the resolution of the triangular system do not induce any growth)

The binary cost is bounded by $O(D^4(t + D))$.

Coordinates : no significant growth (compared with the computation of the first polynomial) $\Rightarrow O(n\, D^3(t + D))$

Complexity : First remarks

The computation of the multiplication table is not so costly compared to the rest since the data are not growing too much.

The reduction of Hermite's quadratic form is the main operation.

Algorithm RUR-Classic:

Input $= q_1[1] = [\mathbf{Trace}(m_{w_i})]_{i=1...D}, + \mathcal{B} + M_{X_i}, i = 1...n$

- $d = \#V(\langle \mathcal{E} \rangle)$ (Hermite's quadratic form); $O(D^3)$
- compute $f_t = \text{charpol}(m_t)$ for $t \in \{ \sum_{i=1}^{n} j^{i-1} X_i,\ j = 0...n \ \frac{D(D-1)}{2} \}$ until $\text{degree}(\tilde{f_t}) = d$; $O(D^3) - \text{practical} \ / O(n\, D^5) - \text{theoretical}$
- compute $g_{t,v},\ v = 1, X_1, ..., X_n$ $O(n\, D^2)$
- return $f_t, g_{t,v}, v = 1, X_1, ..., X_n$

We need to change again our strategy !

Computing faster

The strategy :

- compute d and t using modular arithmetic (modulo a prime number). The result will be correct excepted for a finite number of primes.
- check that the RUR-Candidate is a RUR after the computation to get a deterministic algorithm.

Advantages :

- fast computations (fixed precision) for "predicting" t
- if the prime numbers are big enough : very few are "bad";
- take $t = X_i$ when possible (smaller results due to the sparsity of M_{X_i})
- t may be given by the user.

Checking a RUR-Candidate

There exists a filter : checking that f_t is squarefree

For the general case :

Proposition 25. *A RUR-Candidate* $\mathcal{R}_t(\langle\mathcal{E}\rangle) = \{f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n}\}$ *is a RUR iff* $\mathrm{Trace}(m_{p_i w_j}) = 0, \forall i = 1\ldots n, j = 1\ldots D$ *with* $p_i = X_i g_{t,1}(t) - g_{t,X_i}(t)$*;*

Proof. $\mathcal{R}_t(\langle\mathcal{E}\rangle)$ is a RUR iff $p_i(\alpha) = 0$, $\forall i = 1\ldots n$, $\forall \alpha \in \boldsymbol{V}(\langle\mathcal{E}\rangle)$ since $\#\boldsymbol{V}(f_t) \leq \boldsymbol{V}(\langle\mathcal{E}\rangle)$.

Hermite's quadratic form must be of rank 0:

$$\mathrm{rank}(q_{p_i}) = \mathrm{rank}(\left[\mathrm{Trace}(m_{p_i w_j w_k})\right]_{j=1\ldots D}^{k=1\ldots D}) = \#\{\alpha \in \boldsymbol{V}(\langle\mathcal{E}\rangle), p_i(\alpha) \neq 0\}$$

$\Leftrightarrow$ its first line is null $\Leftrightarrow \mathrm{Trace}(m_{p_i w_j}) = 0, j = 1\ldots D$ $\square$

Checking a RUR-Candidate

Corollary 26. *A RUR-Candidate* $\mathcal{R}_t(\langle\mathcal{E}\rangle) = \{f_t,\ g_{t,1},\ g_{t,X_1},\ \ldots,\ g_{t,X_n}\}$ *is a RUR iff* $q_1\vec{p_i} = 0, \forall i = 1\ldots n,\ j = 1\ldots D$ *with* $p_i = X_i g_{t,1}(t) - g_{t,X_i}(t)$*;*

Proposition 27. *One can check that a RUR-Candidate is a RUR in* $O(D^3 + n\ D^2)$ *arithmetic operations (if the RUR-Candidate has been computed using RUR-Classic).*

Algorithm CheckRUR-Classic

- $\overrightarrow{H_0} = a_0[1, 0, \ldots, 0]$; for $i = 1\ldots D$ $\overrightarrow{H_i(t)} = M_t\overrightarrow{H_{i-1}(t)} + a_i\overrightarrow{H_{i-1}(t)}$;
- if $q_1\,\overrightarrow{H_D(t)} \neq [0, \ldots, 0]$ then return(FALSE)
- for $i = 1\ldots n$ $\overrightarrow{g_{t,X_i}(t)} = \sum_{i=1}^{D} \underbrace{\mathrm{Trace}(X_i t^j)}_{\text{already computed}}\ \overrightarrow{H_{D-i-1}}(t)$
- $\overrightarrow{g_{t,1}(t)} = \sum_{i=1}^{D} \mathrm{Trace}(t^j)\overrightarrow{H_{D-i-1}(t)}$
- for $i = 1\ldots n$ if $q_1(\overrightarrow{g_{t,X_i}(t)} + M_{X_i}\overrightarrow{g_{t,1}(t)}) \neq [0, \ldots, 0]$ then return(FALSE)
- return(TRUE)

$O(D^3 + n\ D^2)$ arithmetic operations - no significant growth of coefficients (compared to the other sub-algorithms)

RUR vs Lexicographic Gröbner bases in the shape position case

When X_1 separates $\boldsymbol{V}(\langle\mathcal{E}\rangle) + \langle\mathcal{E}\rangle$ is radical

RUR	**Lex. G.B.**
$\begin{cases} f(X_1)=0 \\ X_2=\frac{h_2(X_1)}{f'(X_1)} \\ \vdots \\ X_n=\frac{h_n(X_1)}{f'(X_1)} \end{cases}$	$\begin{cases} f(X_1)=0 \\ X_2=f_2(X_1) \\ \vdots \\ X_n=f_n(X_1) \end{cases}$

(same number of arithmetic operations in the "shape position" case)

Proposition : $f'(X_1)\,X_i - h_i(X_1) = X_i - f_i(X_1) \bmod I$

Equivalently : $f_i = f'^{-1}(X_1)h_n(X_1) \bmod I$

Remarks :

all the coefficients of the RUR have the "same" size $(O(D\,t))$

In a Lex. G. B. the coefficients of f appear to be "small" $(O(D\ t))$ compared to those of f_i.

Adding inequations/inequalities

$$\mathcal{E} = \{p_1, ..., p_r\}, \mathcal{F} = \{f_1, ..., f_l\}, \text{ with } p_i, f_i \in \mathbb{Q}[X_1, ..., X_n]$$

$$\mathcal{C} = \{x \in \mathbb{C}^n, p_1 = 0, ..., p_r = 0, f_1 \neq 0, ..., f_s \neq 0\}$$

$$\mathcal{S} = \{x \in \mathbb{R}^n, p_1 = 0, ..., p_r = 0, f_1 > 0, ..., f_s > 0\}$$

A straightforward method :

Compute $f_i(\frac{g_{t,X_1}}{g_{t,1}}, ..., \frac{g_{t,X_n}}{g_{t,1}}), i = 1...s$ and study the signs of these univariate polynomials at the roots of f_t.

Drawback : terrible computations (substitutions+reduction modulo f_t)

A less straightforward method :

$\mathcal{E}' = \{p_1, ..., p_r, T_1 - f_1, ..., T_s - f_s\} \subset \mathbb{Q}[X_1, ..., X_n]$

Compute the RUR of $\langle \mathcal{E}' \rangle$: $\{f_t,\ g_{t,1},\ g_{t,X_1},\ ...,\ g_{t,X_n},\ g_{t,T_1},\ ...,\ g_{t,T_s}\}$ and study the signs of $g_{t,1}, g_{t,T_1}, ..., g_{t,T_s}$ at the roots of f_t (univariate problem).

Drawback : can not study $f_i, i = 1...s$ without re-computing an ideal

Adding inequations/inequalities

A solution : "simulating" Gröbner bases

Definition 28. *Given two monomial orderings* $<_U$ *(w.r.t. the variables* $U_1, ..., U_d$*) and* $<_X$ *(w.r.t. the variables* X_{d+1}, ..., X_n*) one can define a "block" ordering* $<_{U,X}$ *:* $m <_{U,X} m'$ *if and only if*

$m_{|_{U_1=1,\ldots,U_d=1}} <_X m'_{|_{U_1=1,\ldots,U_d=1}}$ *or*

$(m_{|_{U_1=1,\ldots,U_d=1}} = m'_{|_{U_1=1,\ldots,U_d=1}}$ *and* $m_{|_{X_{d+1}=1,\ldots,X_n=1}} <_U m'_{|_{X_{d+1}=1,\ldots,X_n=1}})$.

Lemma 29. *If* G *is a Gröbner basis of* $\langle \mathcal{E} \rangle = \langle p_1, ..., p_r, T_1 - f_1, ..., T_s - f_s \rangle$ *for* $<_X$, *then* $G \cup \{T_i - \text{normalform}(f_i), i = 1...s\}$ *is a Gröbner basis of* $\langle \mathcal{E}' \rangle = \langle p_1, ..., p_r, T_1 - f_1, ..., T_s - f_s \rangle$ *for* $<_{T,X}$ *where* T *is any admissible monomial ordering wrt* $T = [T_1, ..., T_n]$;

Remark : $\frac{\mathbb{Q}[X_1,\ldots,X_n]}{\langle \mathcal{E} \rangle} \cong \frac{\mathbb{Q}[X_1,\ldots,X_n,T_1,\ldots,T_n]}{\langle \mathcal{E}' \rangle}$ so that the only extraneous computations needed to extend the RUR are the construction of the M_{T_i} and the computations of the Trace$(m_{T_i t^j})$.

Complexity : $O(\underbrace{D^3 + n\,D^2}_{\text{RUR}} + s\,D^2)$

Software

http://fgbrs.lip6.fr/Software

SALSA library containing

- FGb (F4 algorithm implemented by J.C. Faugère) - Dynamic library written in C
- RS (RUR + real root isolation by F. Rouillier) - Dynamic library written in C
- An interface with Maple Software (version >9.5)
- ... (see next lecture).

Will be linked with Maple 11 (official distribution)

Conclusion

We have seen how to "solve" in a certified way any kind of zero-dimensional system : counting complex/real roots, certified isolation (by the way or interval arithmetic + symbolic resolution of univariate polynomials), computation of the multiplicities, certified evaluation of polynomials at the roots of a zero-dimensional system (easy to certify the sign for example).

Recent progress have been made using "baby step/giant step technics" (Rouillier 2005) decreasing the number of bit operations required to compute a RUR-Candidate and using specific multiplication tables (only partially stored) decreasing the memory consummed.

The RUR can be used jointly with splitting technics like triangular sets.

Systems with 100/200 complex roots are "easy" in general (seconds)

Systems with up to 1000 complex solutions are (in general) reachable by the current versions of the software (minutes/hours)

Systems with $>$10000 solutions have been solved by the beta versions.

Important remark : only few solvers are able to provide the required univariate "black-boxes" able to "solve" such large polynomials (first polynomial of a RUR)...