Polynomial System Solving in the Real Case

(Using efficiently Gröbner bases for Real Solving)

F. Rouillier

`Fabrice.Rouillier@inria.fr`

SPACES Project - INRIA / University of Paris VI

Plan

Part 1 : General Introduction and motivations

Plan

Part 1 : General Introduction and motivations

- Solving ?
- A general (universal) method : the CAD;
- Powerfull (in practice) tools : Gröbner bases;

Plan

Part 1 : General Introduction and motivations
Part 2 : Zero-dimensional systems

Plan

Part 1 : General Introduction and motivations

Part 2 : Zero-dimensional systems

- Checking the hypothesis (Zero-dimensional);
- Switching to linear algebra;
- Counting Real Roots;
- Variable's elimination;
  - Lexicographic Gröbner bases;
  - Triangular sets;
  - Rational Univariate Representation;
- Isolating Real Roots of algebraic systems;
- Adding inequalities;

Plan

Part 1 : General Introduction and motivations
Part 2 : Zero-dimensional systems
Part 3 : Parametric Zero-dimensional Systems

Plan

Part 1 : General Introduction and motivations
Part 2 : Zero-dimensional systems
Part 3 : Parametric Zero-dimensional Systems

- Checking the hypothesis;
- Generic Solutions ?
  - Cool Solutions
  - Sympa Sotutions
- Adding Inequalities;
- Parameter's space decompositions
- Variety's decomposition w.r.t. inequalities;

Plan

Part 1 : General Introduction and motivations
Part 2 : Zero-dimensional systems
Part 3 : Parametric Zero-dimensional Systems
(small) Part 4 : Positive dimensional systems

- switch to the parametric case

References

The courses are self-contained modulo the following references :

- BW93 : T. Becker and V. Weispfenning. Gröbner bases, A Computational Approach to Commutative Algebra. Graduade Texts in Mathematics, 1993, Springer-Verlag.
- CLO92 :D. Cox and J. Little and D. O'Shea. Ideal Varieties and Algorithms, An introduciton to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics, 1992, Springer-Verlag.
- BPR03 :S. Basu and R. Pollack and M.F. Roy. Algorithms in Real Algebraic Geometry. Algorithms and Computations in Mathematics, 2003, Springer.
- Aub99 :P. Aubry. Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en

General Introduction and Motivations

Motivations

The goals

Motivations

The goals

- Solving systems of polynomial equalities and inequalities;

Motivations

The goals

- Solving systems of polynomial equalities and inequalities;
- Exact results : a real root is not a complex root with a small imaginary part, a cluster is not a singularity, etc.

Motivations

The goals

- Solving systems of polynomial equalities and inequalities;
- Exact results : a real root is not a complex root with a small imaginary part, a cluster is not a singularity, etc.
- Algorithms : always terminates (only a question of time or memory), checkable restrictions (ex. : zero-dimensional)

Motivations

The goals

- Solving systems of polynomial equalities and inequalities;
- Exact results : a real root is not a complex root with a small imaginary part, a cluster is not a singularity, etc.
- Algorithms : always terminates (only a question of time or memory), checkable restrictions (ex. : zero-dimensional)
- Software solutions and algorithms : can solve more than academic applications.

Computations - Algorithms

Computations - Algorithms

- Define a minimal set of usefull mathematical objects that can be computed efficiently.

Computations - Algorithms

- Define a minimal set of usefull mathematical objects that can be computed efficiently.
- Propose several computational solutions.

Computations - Algorithms

- Define a minimal set of usefull mathematical objects that can be computed efficiently.
- Propose several computational solutions.
- Adaptative strategies (depending on mathematical properties checked before or during the computations)

Computations - Algorithms

- Define a minimal set of usefull mathematical objects that can be computed efficiently.
- Propose several computational solutions.
- Adaptative strategies (depending on mathematical properties checked before or during the computations)

Build your own "Solve" function depending on the problem you want to solve

Solving "Semi-Algebraic" Systems

Academic challenge :
Deciding if a first order formula with equalities and inequalities is true or not.
Tarski-Seidenberg $\Rightarrow$ conjonctions and disjonctions of equalities and inequalities.
Applications' challenges : Many critical "sub"-problems

- equalities, inequalities in one variable : number of solutions solutions, numerical approximations, numerically stable solutions;
- zero-dimensional systems (with or without inequalities) : number of solutions solutions, numerical approximations, numerically stable solutions;
- systems with parameters : existence of solutions, properties' discussions w.r.t. parameter's values (ex : number of real roots);
- general positive dimensional sytems : existence of solutions, decomposition of the ambiant space in sign-invariant cells, etc.

Notations

$K \subset K'$ are ordered fields, R the real closure of K' and C the algebraic closure of R. In practice, we consider $K = \mathbb{Q}$, $R = \mathbb{R}$ and $C = \mathbb{C}$.

A semi-algebraic system will be denoted by

$$\mathcal{S} = \{E_1 = 0, \dots E_s = 0, F_1 > 0, \dots F_l > 0\},$$

where $E_i, F_i \in K[Y_1, \dots, Y_n]$

The main ideal of $K[Y_1, \dots Y_n]$ associated to $\mathcal{S}$:

$$I_K = < E_1, \dots E_s >$$

The main variety of $\mathcal{S}$: $V_C = \mathcal{V}(I_C) \subset C^n$. We define also $V_R = V_C \bigcap R^n$.

The Cylindrical Algebraic Decomposition (CAD)

Since the 70's, there exists a universal "black-box", the Cylindrical Algebraic Decomposition. Theoretically, it solves all the listed problems.

Description

- Input = a set of polynomials (F_i);
- Output = a partition of $\mathbb{R}^n$ such that $sign(F_i) = ct$

The CAD is defined recursively.

CAD - Projection Step

At level k, we have a set P_k of polynomial of $K[X_k, \ldots, X_n]$. We construct $P_{k+1} = Proj(P_k)$ as being the smallest set such that :

- If $p \in P_k$, $\deg_{X_k}(p) = d \geq 2$, $Proj(P_k)$ contains all the $sr_j(p, \frac{\partial p}{\partial X_k})$ (non-constant) for $j = 0, \ldots, d$.
- If $p \in P_k$, $q \in P_k$, $Proj(P_k)$ contains $sr_j(p, q)$ (non-constant) for $j = 0, \ldots, \min(\deg_{X_k}(p), \deg_{X_k}(q))$.
- If $p \in P_k$, $\deg_{X_k}(p) \geq 1$ and $lc_{X_k}(p)$ non constant, $Proj(P_k)$ contains $lc_{X_k}(p)$ and $Proj(P_k \setminus \{p\} \cup \{p - lc_{X_k}(p)\})$.
- If $p \in P_k$, $\deg_{X_k}(p) = 0$ and p non constant, $Proj(P_k)$, contains p.

CAD - lifting step

- (1) compute real roots of all polynomials of P_k and sort them;
- (2) take one point on each interval between roots of (1);
- (3) specialize X_k to (1) and (2) in $P_{k-1} \ldots P_1$.

Excepted for $k = n$ step (1) lead to isolate the real roots of polynomial with real algebraic numbers as coefficients which is, in practice, a difficult task.

Also, the basic CAD algorithm can be easily described and implemented using exclusively operations with univariate polynomials.

Computations and size of the output : $O(d^{O(2^n)})$.

The exponential behavior of the method is mainly due to the projection step and in particular the increase of polynomial degrees due to sub-resultant computations.

Why working on alternatives ?

- "Solving" first order formulas : known to be doubly exponential.
- One point / semi-algebraically connected component for an algebraic : known to be simply exponential (see M.F. Roy's lecture)
- etc.

Gröbner bases : some motivations

Simplification of polynomial systems : are two systems (ideals) "equivalent" ?

Zero-dimensional ideals :

- $C[Y_1, \ldots Y_n]/I_C = C \otimes_K K[Y_1, \ldots, Y_n]/I_K$ is a finite dimensional C-vector space;
- $K[Y_1, \ldots, Y_n] \bigcap K[Y_i] \neq \emptyset \ , \ \forall i = 1 \ldots n$
- $\ldots$

This requires to have a good (computable) representation of I and a function to (at least) decide if $p \in I$.

Gröbner bases : a minimal set of definitions

A Gröbner basis G of I w.r.t. any admissible monomial ordering $<$, is a set of generators of I such that $\exists$ a K-linear function (Normal Form) $NF_<(.,G) : K[Y_1,\dots Y_n] \longrightarrow K[Y_1,\dots Y_n]$ s.t.

$$NF_<(p,G) = 0 \Leftrightarrow p \in I$$

An admissible monomial ordering is a total well-ordering (compatible with the multiplication) on the monomials of $K[Y_1,\dots,Y_n]$.

$LM_<(p)$ (leading monomial) , $LC_<(p)$ (leading coefficient),
$LT_<(p) = LC_<(p)LM_<(p)$ (leading term).

The $NF_<$ function "generalizes" the Euclidian division for univariate polynomials.

Gröbner bases : caracterization and properties

A Gröbner basis can be computed adding to the set of generators polynomials in the form :

$$S(f_1, f_2) = \frac{LT_<(f_2)}{gcd(LM_<(f_1), LM_<(f_2))} f_1 - \frac{LT_<(f_1)}{gcd(LM_<(f_1), LM_<(f_2))} f_2$$

A set G is a Gröbner basis iff

$$NF_<(S(g_1, g_2), G) = 0 \ , \ \forall g_1, g_2 \in G$$

Monomial ideals :$< LT_<(I) >=< LT_<(G) >$
A reduced Gröbner basis G of I for $<$ is a Gröbner basis such that

$$NF_<(g - LT_<(g, G)) = g - LT_<(g, G) \ \forall g \in G$$

A (reduced) Gröbner basis is unique (for $<$).

Gröbner bases : definition of monomial orderings

The main used monomial orderings are :
Lexicographic orderings

$$Y_1^{\alpha_1} \cdot \ldots \cdot Y_n^{\alpha_n} <_{Lex} Y_1^{\beta_1} \cdot \ldots \cdot Y_n^{\beta_n} \Leftrightarrow \exists i_0 \leq n \; , \; \begin{cases} \alpha_i = \beta_i \; , \; \forall i = 1 \ldots i_0 - 1, \\ \alpha_{i_0} < \beta_{i_0} \end{cases}$$

Degree Reverse Lexicographic orderings

$$Y_1^{\alpha_1} \cdot \ldots \cdot Y_n^{\alpha_n} <_{DRL} Y_1^{\beta_1} \cdot \ldots \cdot Y_n^{\beta_N} \Leftrightarrow Y^{\left((\sum_k \beta_k), \beta_n, \ldots, \beta_1\right)} <_{Lex} Y^{\left((\sum_k \alpha_k), \alpha_n, \ldots, \alpha_1\right)}$$

Block Orderings
Let $<_1$ (resp. $<_2$) be an admissible ordering on $U = Y_1, \ldots, Y_d$ (resp. $X = Y_{d+1}, \ldots, Y_n$), we define $<$ on $[Y_1, \ldots, Y_n]$

$$U^m X^l < U^p X^q \Leftrightarrow ((X^l <_2 X^q) \; or \; (X^l = X^q \; and \; U^m <_1 U^p))$$

Gröbner basis : computations

- The computation time of a Gröbner basis depends on the used monomial ordering.
- In general, a lexicographic Gröbner basis is difficult to compute direclty;
- In general, Gröbner bases for a Degree orderings (including block orderings) a much more easy to compute than lexicographic Gröbner basis;
- In general, a Degree Reverse Lexicographic Gröbner basis is the fastest for computations;
- The variants of algorithms used for computing Gröbner basis differs by the criterion used to avoid unusefull computations (S-polynomials that reduces to 0), the strategies used for selecting critical pairs, and the internal representations;
- The initial version is due to Buchberger, the fastest one is due to J.C. Faugère : Algorithm F5 uses selection strategies such that no S-polynomials reduce to 0 during the computations.

Zero Dimensional Systems

Dimension 0 : check !

Let G a Gröbner basis of I for any admissible monomial ordering $<$.

Known result : $\sharp V_C < \infty \Leftrightarrow C[Y]/I_C$ is a finite dimensional C-vector space

($\Leftrightarrow K[Y]/I_K$ is a finite dimensional K-vector space $\Leftrightarrow I_K$ has dimension $0 \Leftrightarrow I_C$ has dimension 0)

Dimension 0 : check !

Let G a Gröbner basis of I for any admissible monomial ordering $<$.

Known result : $\sharp V_C < \infty \Leftrightarrow C[Y]/I_C$ is a finite dimensional C-vector space

($\Leftrightarrow K[Y]/I_K$ is a finite dimensional K-vector space $\Leftrightarrow I_K$ has dimension $0 \Leftrightarrow I_C$ has dimension 0)

I has dimension 0 iff $\forall i = 1 \ldots n, \exists g \in G, \exists n_i \in \mathbb{N}^* \ : \ LM_<(g) = Y^{n_i}$

Dimension 0 : check !

Let G a Gröbner basis of I for any admissible monomial ordering $<$.

Known result : $\sharp V_C < \infty \Leftrightarrow C[Y]/I_C$ is a finite dimensional C-vector space

($\Leftrightarrow K[Y]/I_K$ is a finite dimensional K-vector space $\Leftrightarrow I_K$ has dimension $0 \Leftrightarrow I_C$ has dimension 0)

I has dimension 0 iff $\forall i = 1 \ldots n, \exists g \in G, \exists n_i \in \mathbb{N}^* \ : \ LM_<(g) = Y^{n_i}$

$\Rightarrow$ Since $C[Y]/I_C$ is a finite dimensional C-vector space,
$\forall i = 1 \ldots n, \exists D_i \in \mathbb{N}\ ,\ 1, Y_i, \ldots, Y_i^{D_i}$ are C-lineary dependants in $C[Y]/I_C$. Also $\exists P_i \neq 0 \in C[Y_i] \bigcap I$. In particular $NF_<(P_i, G) = 0$.

Dimension 0 : check !

Let G a Gröbner basis of I for any admissible monomial ordering $<$.

Known result : $\sharp V_C < \infty \Leftrightarrow C[Y]/I_C$ is a finite dimensional C-vector space

($\Leftrightarrow K[Y]/I_K$ is a finite dimensional K-vector space $\Leftrightarrow I_K$ has dimension $0 \Leftrightarrow I_C$ has dimension 0)

I has dimension 0 iff $\forall i = 1 \ldots n, \exists g \in G, \exists n_i \in \mathbb{N}^* \ : \ LM_<(g) = Y^{n_i}$

$\Leftarrow$ If $\forall i = 1 \ldots n, \exists g \in G, n_i \in \mathbb{N}^* \ : \ LM_<(g) = Y^{n_i}$, then $p \in C[Y]/I_C$ is a linear combination of monomials in the form $Y_1^{m_1} \ldots Y_n^{m_n}$ with $m_i < n_i$ and so $C[Y]/I_C$ is a finite dimensional C-vector space.

Dimension 0 : check !

Let G a Gröbner basis of I for any admissible monomial ordering $<$.

Known result : $\sharp V_C < \infty \Leftrightarrow C[Y]/I_C$ is a finite dimensional C-vector space

($\Leftrightarrow K[Y]/I_K$ is a finite dimensional K-vector space $\Leftrightarrow I_K$ has dimension $0 \Leftrightarrow I_C$ has dimension 0)

I has dimension 0 iff $\forall i = 1 \ldots n, \exists g \in G, \exists n_i \in \mathbb{N}^* \; : \; LM_<(g) = Y^{n_i}$

If $\mathcal{S} \subset K[Y]$ then $G \in K[Y]$.

The dimension of the K-vector space (resp. C-vector space) $K[Y]/I_K$ (resp. $C[Y]/I_C$) is the number of complex zeroes of I_C counted with multiplicities.

Dimension 0 : computing $K[Y]/I_K$

A monomial basis of the K-vector space $K[Y]/I_K$ can be read on a Gröbner basis G of I_K (for any monomial ordering) :

$$\mathcal{B}_<(I_K) = \{m \in M[Y] \; : NF_<(m, G) = m\}$$

This is the set of all the possible monomials $m \in K[Y]$ that can not be reduced by $NF_<(., G)$, or equivalently such that $\nexists g \in G$ such that $LM_<(g)$ divides m.

Dimension 0 : multiplication maps

Let $h \in K[Y]$

$$\begin{array}{rccc} m_h : & C[Y]/I_C & \longrightarrow & C[Y]/I_C \\ & \overline{p} & \longmapsto & \overline{ph} \end{array}$$

(Stickelberger) The eigenvalues of m_h are exactly the $h(\alpha)$, $\alpha \in V_C$ with respective multiplicities the multiplicity of α (dimension of $(C[Y]/I_C)_\alpha$).

Suppose G is a Gröbner basis of I for $<$ and that $\mathcal{B}_<(G) = \{w_1, \ldots w_D\}$

If $NF_<(h, G) = \sum_{i=1}^{D} a_i w_i$ with $a_i \in K$ (uniquely defined if G is reduced), let denote $\overrightarrow{h} = [a_1, \ldots a_D]$, and by M_h the matrix of m_h with respect to $\mathcal{B}_<(G)$.

Then

$$M_h = [\overrightarrow{hw_1}, \ldots, \overrightarrow{hw_D}]^T$$

can explicitely computed.

Dimension 0 : applications of Stickelberger theorem

The eigenvalues of m_{Y_i} are exctly the i-th coordinates of all the points of V_C.

If I is radical and if $Y_1(\alpha) \neq Y_1(\beta) \forall \alpha \neq \beta \in V_C$, then a Gröbner basis for any lexicographic ordering such that $Y_1 < Y_i \; i = 1 \ldots n$ has always the following shape :

$$\begin{cases} f(Y_1) = 0 \\ Y_2 = f_2(Y_1) \\ \vdots \\ Y_n = f_n(Y_1) \end{cases}$$

When a Gröbner basis has this shape, the system is said to be in shape position.

Computing the complex/real roots of the system is now equivalent to solve $f(Y_1) = 0$

Dimension 0 : shape lemma

Suppose I radical.

Let $\mathcal{T} = \{Y_1 + iY_2, \ldots + i^{n-1}Y_n\ ,\ i = 1 \ldots nD(D-1)/2\}$. There exists $t \in \mathcal{T}$ s.t. $\alpha \neq \beta \in V_C \Rightarrow t(\alpha) \neq t(\beta)$.
Sickelberger $\Rightarrow f(T) = CharPol(m_t)$ is squarefree.

Also, the system can be re-written :

$$\begin{cases} f(T) = 0 \\ Y_2 = f_2(T) \\ \vdots \\ Y_n = f_n(T) \end{cases}$$

Computing the complex/real roots of the system is now equivalent to solve $f(T) = 0$

Dimension 0 : Hermite's quadratic form

For $h \in K[Y]$, let define :

$$\begin{array}{rccc} q_p : & K[Y]/I_K & \longrightarrow & K \\ & f & \longmapsto & Trace(m_{hp^2}) \end{array}$$

- $rank(q_p) = \sharp\{y \in V_C \ : \ p(y) \neq 0\}$
- $sig(q_p) = \sharp\{y \in V_R \ : \ p(y) > 0\} - \sharp\{y \in V_R \ : \ p(y) < 0\}$.

In particular, the rank (resp. signature) of q_1 give the number of distinct complex (resp. real) roots of $\mathcal{S}$.

Application : P separates V_C iff $degree(\overline{CharPol(m_p)}) = rank(q_1)$

Dimension 0 : the general case - Lex. G. Basis

The general shape of the Lexicographic Gröbner basis is the following :

$$
\begin{array}{l}
f_1(Y_1) \\
f_2(Y_1, Y_2) \\
\vdots \\
f_{k_2}(Y_1, Y_2) \\
f_{k_2+1}(Y_1, Y_2, Y_3) \\
\vdots \\
f_{k_{n-1}+1}(Y_1, \ldots, Y_n) \\
\vdots \\
f_{k_n}(Y_1, \ldots, Y_n)
\end{array}
$$

Dimension 0 : the general case - Lex. G. Basis

The general shape of the Lexicographic Gröbner basis is the following :

$$
\begin{array}{l}
f_1(Y_1) \\
f_2(Y_1, Y_2) \\
\vdots \\
f_{k_2}(Y_1, Y_2) \\
f_{k_2+1}(Y_1, Y_2, Y_3) \\
\vdots \\
f_{k_{n-1}+1}(Y_1, \dots, Y_n) \\
\vdots \\
f_{k_n}(Y_1, \dots, Y_n)
\end{array}
$$

Proof : since I_K has dimension 0, then $I_K \bigcap K[Y_i] \neq \emptyset \; \forall i = 1 \dots n$. If $p \in I_K \bigcap K[Y_i]$, then $NF_{<_{lex}}(p, G) = 0$ and in particular $\exists g \in G$ s.t. $LM_{<_{lex}}(g) = Y_i^{n_i}$, and consequently $g \in K[Y_1, \dots, Y_i]$.

Dimension 0 : the general case - Lex. G. Basis

The general shape of the Lexicographic Gröbner basis is the following :

$f_1(Y_1)$
$f_2(Y_1, Y_2)$
$\vdots$
$f_{k_2}(Y_1, Y_2)$
$f_{k_2+1}(Y_1, Y_2, Y_3)$
$\vdots$
$f_{k_{n-1}+1}(Y_1, \ldots, Y_n)$
$\vdots$
$f_{k_n}(Y_1, \ldots, Y_n)$

Proof : since I_K has dimension 0, then $I_K \bigcap K[Y_i] \neq \emptyset \; \forall i = 1 \ldots n$. If $p \in I_K \bigcap K[Y_i]$, then $NF_{<_{lex}}(p, G) = 0$ and in particular $\exists g \in G$ s.t. $LM_{<_{lex}}(g) = Y_i^{n_i}$, and consequently $g \in K[Y_1, \ldots, Y_i]$.

$G \bigcap K[X_1, \ldots, X_i]$ is a lex. G. Basis of $G \bigcap K[X_1, \ldots, X_i]$

Dimension 0 : the general case - Lex. G. Basis

The general shape of the Lexicographic Gröbner basis is the following :

$f_1(Y_1)$

$f_2(Y_1, Y_2)$

$\vdots$

$f_{k_2}(Y_1, Y_2)$

$f_{k_2+1}(Y_1, Y_2, Y_3)$

$\vdots$

$f_{k_{n-1}+1}(Y_1, \ldots, Y_n)$

$\vdots$

$f_{k_n}(Y_1, \ldots, Y_n)$

Proof : since I_K has dimension 0, then $I_K \bigcap K[Y_i] \neq \emptyset \; \forall i = 1 \ldots n$. If $p \in I_K \bigcap K[Y_i]$, then $NF_{<_{lex}}(p, G) = 0$ and in particular $\exists g \in G$ s.t. $LM_{<_{lex}}(g) = Y_i^{n_i}$, and consequently $g \in K[Y_1, \ldots, Y_i]$.

$G \bigcap K[X_1, \ldots, X_i]$ is a lex. G. Basis of $G \bigcap K[X_1, \ldots, X_i]$

Numerical "Solve" is difficult

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when

$\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \;:\; LM_{<2}(g) = Y_i^{n_i}$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \; : \; LM_{<2}(g) = Y_i^{n_i}$

---

compute $\overrightarrow{1}, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_1^d}$ and stop when a linear dependance is founded.

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \ : \ LM_{<2}(g) = Y_i^{n_i}$

$f_1(Y_1)$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \;:\; LM_{<2}(g) = Y_i^{n_i}$

$f_1(Y_1)$

follow with $\overrightarrow{Y_2}, \overrightarrow{Y_1 Y_2}, \ldots, \overrightarrow{Y_1^{d-1}} Y_2$ and stop when a linear dependance is founded.

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \ : \ LM_{<2}(g) = Y_i^{n_i}$

$f_1(Y_1)$

$f_2(Y_1, Y_2)$

$\vdots$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \ : \ LM_{<2}(g) = Y_i^{n_i}$

$f_1(Y_1)$

$f_2(Y_1, Y_2)$

$\vdots$

follow multiplying by Y_2 up to finding $g \in G_{<_2}$ such that $LM_{<_2}(g) = Y_2^{n_2}$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \; : \; LM_{<2}(g) = Y_i^{n_i}$

$$f_1(Y_1)$$
$$f_2(Y_1, Y_2)$$
$$\vdots$$
$$f_{k_2}(Y_1, Y_2)$$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \ldots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \ : \ LM_{<2}(g) = Y_i^{n_i}$

$f_1(Y_1)$

$f_2(Y_1, Y_2)$

$\vdots$

$f_{k_2}(Y_1, Y_2)$

Apply the same process iteratively with $Y_3, \ldots, Y_n$

Dimension 0 : FGLM Algorithm

Let G a G. Basis for any ordering $<_1$. One want to compute the G. Basis of $< G >$ for an ordering $<_2$.

The basic principle is simple : considere all the possible monomials in increasing order for $<_2$ as vectors w.r.t $\mathcal{B}_{<_1}(G_{<1})$, detect the linear combinations (polynomials of the new G. Basis : $G_{<_2}$), stop when $\forall i = 1 \dots n \exists n_i \in \mathbb{N}^* \exists g \in G_{<_2} \; : \; LM_{<2}(g) = Y_i^{n_i}$

$$
\begin{array}{l}
f_1(Y_1) \\
f_2(Y_1, Y_2) \\
\vdots \\
f_{k_2}(Y_1, Y_2) \\
f_{k_2+1}(Y_1, Y_2, Y_3) \\
\vdots \\
f_{k_n}(Y_1, \dots, Y_n)
\end{array}
$$

Apply the same process iteratively with $Y_3, \dots, Y_n$

Dimension 0 : the general case - RUR

Let $t \in \mathcal{T}$ s.t. $\alpha \neq \beta \in V_C \Rightarrow t(\alpha) \neq t(\beta)$.
Let $g_t(T) = CharPol(m_t) = \prod_{\alpha \in V_C} (T - t(\alpha))^{\mu(\alpha)}$.

We denote by $\overline{f}$ the square-free part of $f \in K[T]$ and by $H_i(f)$ the i-th Horner's polynomial associated to f : $H_i(f)(T) = \sum_{j=0}^{i} a_{i-j} T^i$ if $f = \sum_{j=0}^{D} a_i T^i$.

For $p \in K[Y]$, if $d = degree(\overline{f})$ and $g_{t,p}(T) = \sum_{i=0}^{d-1} Trace(m_{pt^i}) H_{d-i-1}(g_t)(T)$, then $p(\alpha) = \frac{g_{t,p}(t(\alpha))}{g_{t,1}(t(\alpha))}$.

"Proof" : since $Trace(m_p) = \sum_{\alpha \in V_C} \mu(\alpha) p(\alpha)$, then

$$g_{t,p}(T) = \sum_{\alpha \in V_C} \mu(\alpha) p(\alpha) \prod_{\beta \in V_C, \beta \neq \alpha} (T - t(\beta))$$

Dimension 0 : the general case - RUR

Let $t \in \mathcal{T}$ s.t. $\alpha \neq \beta \in V_C \Rightarrow t(\alpha) \neq t(\beta)$.
Let $g_t(T) = CharPol(m_t) = \prod_{\alpha \in V_C} (T - t(\alpha))^{\mu(\alpha)}$.

We denote by $\overline{f}$ the square-free part of $f \in K[T]$ and by $H_i(f)$ the i-th Horner's polynomial associated to f : $H_i(f)(T) = \sum_{j=0}^{i} a_{i-j} T^i$ if $f = \sum_{j=0}^{D} a_i T^i$.

For $p \in K[Y]$, if $d = degree(\overline{f})$ and
$g_{t,p}(T) = \sum_{i=0}^{d-1} Trace(m_{pt^i}) H_{d-i-1}(g_t)(T)$, then $p(\alpha) = \frac{g_{t,p}(t(\alpha))}{g_{t,1}(t(\alpha))}$.

A one-to-one correspondance :

$$\begin{array}{ccc} \mathcal{V}(I_K) & \longrightarrow & \mathcal{V}(g_t) \\ (\alpha_1, \ldots, \alpha_n) & \longrightarrow & t(\alpha_1, \ldots, \alpha_n) \\ \left(\frac{g_{t,Y_1}(\beta)}{g_{t,1}(\beta)}, \ldots, \frac{g_{t,Y_n}(\beta)}{g_{t,1}(\beta)} \right) & \longleftarrow & \beta \end{array}$$

Dimension 0 : the Rational Univariate Representation

$\{g_t, g_{t,1}, g_{t,Y_1}, \ldots, g_{t,Y_n}\}$ is the Rational Univariate Representation of V_C associated to t.

Note that $g_{t,1} = \overline{g_t'}$ s.t. g_t and $g_{t,1}$ are coprime.

Solving the system through the RUR means :

- solving the univariate polynomial g_t
- evaluating/studying the rational funtions $g_{t,Y_i}/g_{t,1}$ at the roots of g_t.

Since the RUR has coefficients in K, it preserves the real roots.

By construction, it “preserves” the multiplicities. In particular, a squarefree decomposition of g_t would decompose the zeroes w.r.t. the multiplicities.

Remark : this costly computation can be avoid since

$$\frac{\overline{g_t}'}{g_{t,1}}(t(\alpha)) = \mu(\alpha)$$

RUR : a naive algorithm

- (1) compute $d = rank(q_1)$
- (2) find $t \in \mathcal{T} = \{Y_1 + iY_2, \ldots + i^{n-1}Y_n \ , \ i = 1 \ldots nd(d-1)/2\}$ such that $degree(\overline{PolChar(m_t)}) = d$
- (3) compute the $Trace(m_{X_j t^i})$ for $i = 1 \ldots d$ and $j = 1 \ldots n$
- construct the RUR

In practice, one guess a separating t modulo p (steps (1) and (2)), and check after the full computation that the computed set is a RUR :

- $\{g_t, g_{t,1}, g_{t,Y_1}, \ldots, g_{t,Y_n}\}$ is a RUR iff $g_t(t) \in I_K$ and $h_j = g_{t,1}(t)Y_j - g_{t,Y_j} \in \sqrt{I_K}$.
- $h_j \in \sqrt{I_K}$ iff $rank(q_{h_j}) = 0$ iff $Trace(m_{h_j w_i}) = 0 \ , \ \forall i = 1 \ldots D$.

Another trick is that $Trace(m_{t^i})$ is exactly the i-th Newton sum of g_t (Stickelberger) : all the polynomials of the RUR can be easily computed once knowing the $Trace(m_{Y_j t^i})$

Dimension 0 : back to the shape lemma

When I is radical and Y_1 is separating V_C, one can compute the RUR associated with Y_1, and we have an "equivalent" system :

$$\begin{array}{l} g_{Y_1}(Y_1) \\ g_{Y_1,1}(Y_1)Y_2 - g_{Y_1,Y_2}(Y_1) \\ \vdots \\ g_{Y_1,1}(Y_1)Y_n - g_{Y_n,Y_2}(Y_1) \end{array}$$

One can deduce a lexicographic Gröbner basis from a RUR

Dimension 0 : back to the shape lemma

When I is radical and Y_1 is separating V_C, one can compute the RUR associated with Y_1, and we have an “equivalent” system :

$$
\begin{array}{l}
g_{Y_1}(Y_1) \\
Y_2 - g_{Y_1,1}(Y_1)^{-1} g_{Y_1,Y_2}(Y_1) \ mod \ g_{Y_1}(Y_1) \\
\vdots \\
Y_n - g_{Y_1,1}(Y_1)^{-1} g_{Y_n,Y_2}(Y_1) \ mod \ g_{Y_1}(Y_1)
\end{array}
$$

This computation induces, in general, a growth of coefficients such that the coefficients of the RUR are smaller than those of the lexicographic Gröbner basis

Triangular sets

A triangular set is a set of polynomials with the following shape :

$$\left\{\begin{array}{l} t_1(X_1) \\ t_2(X_1, X_2) \\ \vdots \\ t_n(X_1, \ldots, X_n) \end{array}\right.$$

(the t_i may be identically zero).

Triangular sets : basic definitions

For $p \in K[X_1, \ldots, X_n] \setminus K$, we denote by $\mathrm{mvar}(p)$ (and we call *main variable* of p) the greatest variable appearing in p w.r.t. a fixed lexicographic ordering.

Notations :

- h_i the leading coefficient of t_i (when $t_i \neq 0$ is seen as a univariate polynomial in its main variable), and $h = \Pi_{i=1, t_i \neq 0}^{n} h_i$.
- $\mathrm{sat}(T) =< T >: h^\infty = \{p \in K[X_1, \ldots, X_n] \mid \exists m \in \mathbb{N},\ h^m p \in \langle T \rangle\}$;
- $\overline{\mathcal{V}(T) \setminus \mathcal{V}(h)} = \mathcal{V}(sat(T))$ (elementary property of localization).

A triangular set $T = (t_1, \ldots, t_n) \subset K[X_1, \ldots, X_n]$ is said to be *regular* if $\forall i \in \{1, \ldots, n\}$, such that $t_i \neq 0$, the initial h_i does not belong to any associated prime ideal of $\mathrm{sat}(t_1, \ldots, t_{i-1}) \bigcap K[X_1, \ldots, X_{i-1}]$.

Triangular sets : representation of a variety

One may naturally "compute"

$$\overline{V(< T >) \setminus V(h)} = V(sat(< T >))$$

but the full study of $V(< T >)$ requires additional computations.

If T is regular, then $sat(T)$ is equidimensional (elementary property of localization).

It is always possible to represent an algebraic variety as the union of varieties defined as zeroes of regular triangular sets

$$V_C = \bigcup_i V(\mathrm{sat}(T_i))$$

but this do not give a straightforward representation (need to compute $\mathrm{sat}(T_i)$).

Triangular sets in the zero-dimensional case : lextriangular

Start from a Lexicographic Gröbner basis :

$$
\begin{array}{l}
f_1(Y_1) \\
f_2(Y_1, Y_2) \\
\vdots \\
f_{k_2}(Y_1, Y_2) \\
f_{k_2+1}(Y_1, Y_2, Y_3) \\
\vdots \\
f_{k_{n-1}}(Y_1, \ldots, Y_n) \\
f_{k_{n-1}+1}(Y_1, \ldots, Y_n) \\
\vdots \\
f_{k_n}(Y_1, \ldots, Y_n)
\end{array}
$$

Triangular sets in the zero-dimensional case : lextriangular

Start from a Lexicographic Gröbner basis :

$f_1(Y_1)$
$f_2(Y_1, Y_2)$
$\vdots$
$f_{k_2}(Y_1, Y_2)$
$f_{k_2+1}(Y_1, Y_2, Y_3)$
$\vdots$
$f_{k_{n-1}}(Y_1, \ldots, Y_n)$
$f_{k_{n-1}+1}(Y_1, \ldots, Y_n)$
$\vdots$
$f_{k_n}(Y_1, \ldots, Y_n)$

The triangular set extracted from the Lex. G. basis is not necessarily regular.

- if $< LC(f_2, Y_2), f_1 > \neq < 1 >$, split into two systems : $< G, LC(f_2, Y_2) >$, and $G : LC(f_2, Y_2)$ and follow with the same strategy.
- otherelse, do the same with f_{k_2+1} and Y_3.

and so on ...

Triangular sets in the zero-dimensional case : lextriangular

Start from a Lexicographic Gröbner basis :

$f_1(Y_1)$
$f_2(Y_1, Y_2)$
$\vdots$
$f_{k_2}(Y_1, Y_2)$
$f_{k_2+1}(Y_1, Y_2, Y_3)$
$\vdots$
$f_{k_{n-1}}(Y_1, \ldots, Y_n)$
$f_{k_{n-1}+1}(Y_1, \ldots, Y_n)$
$\vdots$
$f_{k_n}(Y_1, \ldots, Y_n)$

The triangular set extracted from the Lex. G. basis is not necessarily regular.

- if $< LC(f_2, Y_2), f_1 > \neq < 1 >$, split into two systems : $< G, LC(f_2, Y_2) >$, and $G : LC(f_2, Y_2)$ and follow with the same strategy.
- otherelse, do the same with f_{k_2+1} and Y_3.

and so on ...

Due to the choice of the polynomials, the computed T_i are lexicographic Gröbner basis. In particular, $< G_i >= sat(T_i)$.

Triangular sets in the zero-dimensional case : lextriangular

Start from a Lexicographic Gröbner basis :

$f_1(Y_1)$
$f_2(Y_1, Y_2)$
$\vdots$
$f_{k_2}(Y_1, Y_2)$
$f_{k_2+1}(Y_1, Y_2, Y_3)$
$\vdots$
$f_{k_{n-1}}(Y_1, \ldots, Y_n)$
$f_{k_{n-1}+1}(Y_1, \ldots, Y_n)$
$\vdots$
$f_{k_n}(Y_1, \ldots, Y_n)$

The triangular set extracted from the Lex. G. basis is not necessarily regular.

- if $< LC(f_2, Y_2), f_1 > \neq < 1 >$, split into two systems : $< G, LC(f_2, Y_2) >$, and $G : LC(f_2, Y_2)$ and follow with the same strategy.
- otherelse, do the same with f_{k_2+1} and Y_3.

and so on ...

Due to the choice of the polynomials, the computed T_i are lexicographic Gröbner basis. In particular, $< G_i > = sat(T_i)$.
Since $sat(T)$ is equidimensional when T is regular, this method can easily be generalized to the positive dimensional case (Safey El Din's thesis).

Adding Inequalities

Solutions of a zero dimensional system where $F_j > 0$?

For each F_j, compute

$$g_{t,F_j}(T) = \sum_{i=0}^{d-1} Trace(m_{F_j t^i}) H_{d-i-1}(g_t)(T)$$

Then $F_j(\alpha) = \frac{g_{t,F_j}(t(\alpha))}{g_{t,1}(t(\alpha))}$

Also, it is sufficient to compute the sign of $g_{t,F_j} g_{t,1}$ at the real roots of g_t.

Computational Strategies

Computational strategies and tricks will be studied in the practical session.

Examples of Software that can be used :

- Maple 8 user interface
- Gb (J.C. Faugère) [external] - Gröbner basis computations
- RS (F. Rouillier) [external] - RUR - Real Roots of zero-dimensional systems and univariate polynomials

Available at `http;//spaces.lip6.fr`

MuPAD versions in progress.

Algorithms performances

Empirical measures of performances :

- A means at least "average" compared with Gb implementation of algorithm F4 (Faugère) for computing DRL Gröbner bases;
- B means "slower" but may be reasonable;
- C means "very slow";

- Buchberger's Algorithm for DRL G. Basis (Gb) :C;
- F4 Algorithm for Lex G. Basis (Gb) :C;
- FGLM on a DRL G. Basis (Gb) :B for low degree and small coefficients, otherelse C in shape lemma case, maybe B for some non shape lemma cases.
- RUR on any G. Basis (RS) : A in shape lemma case for reasonable degrees, B in non shape lemma case for reasonable degrees, C for high degrees;
- Lextriangular (Gb) : A
- Real Root isolation (RS) :A.

Examples

Fabrice : Start your Maple session !