Lecture 2-13-1 - Polynomial systems, computer algebra and applications

Jean-Charles Faugère

Change the ordering (FGLM)
Algebraic Cryptanalysis
Linear Algebra
Characterizations of Gröbner Bases
F_4

2022 - 2023 – MPRI

FGLM: Change of the Ordering

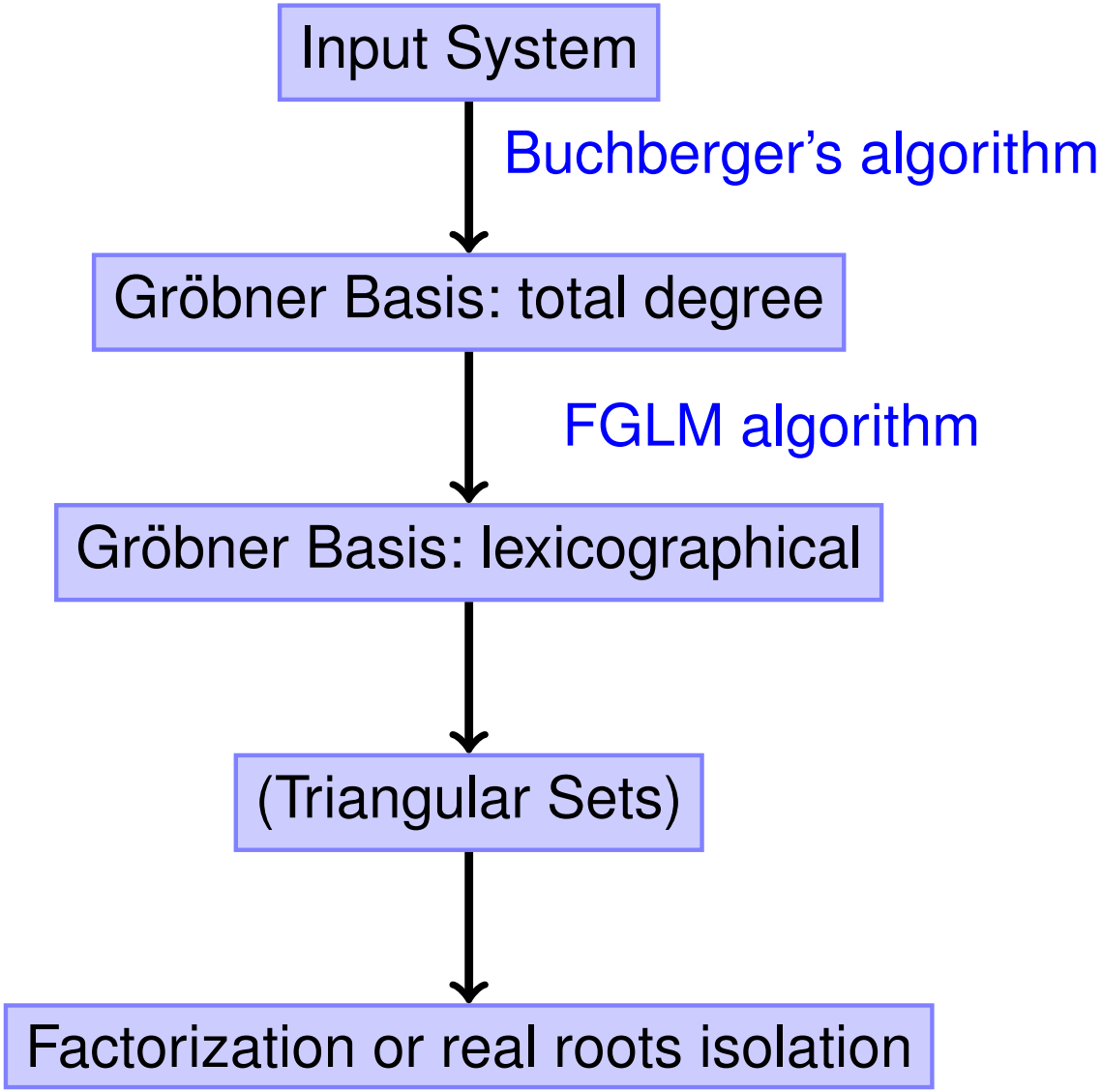
Improve the efficiency

The goal is to speed up the computation of Gröbner basis:

1. obtain some speedup for the direct computation Gröbner basis: Buchberger, Magma default (F_4) or any other algorithm.
2. compute a Gröbner basis in several steps: the lexicographical ordering is the most useful to compute the solutions but the computation of Gröbner basis for a total degree ordering is much faster.

The **FGLM** algorithm allows us to to use linear algebra to perform a change of ordering of a Gröbner basis of a zero dimensional ideal.

Need to use several algorithms



Input of the new algorithm

We assume that the ideal I is **zero-dimensional** and that we know a linear map:

$$\varphi_I : \left(\begin{array}{ccc} \mathbb{K}[x_1, \ldots, x_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]/I \\ p & \longmapsto & \overline{p} \end{array} \right)$$

which is a normalForm, that is to say satisfies the following conditions:

$$\varphi_I(p) = 0 \text{ if and only if } p \in I$$
$$\varphi_I(p \cdot q) = \varphi_I(p \cdot \varphi_I(q)) = \varphi_I(\varphi_I(p) \cdot \varphi_I(q))$$
$$\varphi \circ \varphi = \varphi$$

Input of the new algorithm

$$\varphi_I : \left(\begin{array}{ccc} \mathbb{K}[x_1, \ldots, x_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]/I \\ p & \longmapsto & \overline{p} \end{array} \right)$$

which is a normalForm, that is to say satisfies the following conditions:

$\varphi_I(p) = 0$ if and only if $p \in I$
$\varphi_I(p \cdot q) = \varphi_I(p \cdot \varphi_I(q)) = \varphi_I(\varphi_I(p) \cdot \varphi_I(q))$
$\varphi \circ \varphi = \varphi$

A natural choice is to take $\varphi_I(p) = \mathrm{NormalForm}(p, G, <)$. In that case, the kernel φ_I is exactly the ideal I :

$\ker(\varphi_I) = I$ (Buchberger's Theorem).

Input of the new algorithm

$$\varphi_I : \left(\begin{array}{ccc} \mathbb{K}[x_1, \ldots, x_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]/I \\ p & \longmapsto & \overline{p} \end{array} \right)$$

which is a normalForm, that is to say satisfies the following conditions:

$$\begin{array}{l} \varphi_I(p) = 0 \text{ if and only if } p \in I \\ \varphi_I(p \cdot q) = \varphi_I(p \cdot \varphi_I(q)) = \varphi_I(\varphi_I(p) \cdot \varphi_I(q)) \\ \varphi \circ \varphi = \varphi \end{array}$$

Example

$G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt then monomial ordering DRL with $x_2 > x_1$. Then

$$\varphi : p \longmapsto \text{NormalForm}(p, G_{<_{\text{DRL}}})$$

is a linear map and $\ker(\varphi) = I = \text{Id}(G_{<_{\text{DRL}}})$.

We say that p and q are congruent modulo I, written $p \equiv q$, iff $\varphi(p) = \varphi(q)$.

Proposition

Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be an ideal. Then the congruence modulo I is an equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$.

We say that p and q are congruent modulo I, written $p \equiv q$, iff $\varphi(p) = \varphi(q)$.

Proposition

Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be an ideal. Then the congruence modulo I is an equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$.

An equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$ partitions $\mathbb{K}[x_1, \ldots, x_n]$ into a collection of disjoint subsets called equivalence classes.
For any $p \in \mathbb{K}[x_1, \ldots, x_n]$, the class of p is the set

$$\overline{p} = \{q \in \mathbb{K}[x_1, \ldots, x_n] \mid \varphi(q) = \varphi(p)\}$$

Very often, in the following we will identify $\overline{p}$ and p !

We say that p and q are congruent modulo I, written $p \equiv q$, iff $\varphi(p) = \varphi(q)$.

Proposition

Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be an ideal. Then the congruence modulo I is an equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$.

An equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$ partitions $\mathbb{K}[x_1, \ldots, x_n]$ into a collection of disjoint subsets called equivalence classes.
For any $p \in \mathbb{K}[x_1, \ldots, x_n]$, the class of p is the set

$$\overline{p} = \{q \in \mathbb{K}[x_1, \ldots, x_n] \mid \varphi(q) = \varphi(p)\}$$

Very often, in the following we will identify $\overline{p}$ and p ! We can define the following ring operations:

$$\overline{p} + \overline{q} = \overline{p + q}$$
$$\overline{p} \times \overline{q} = \overline{p \times q}$$

We say that p and q are congruent modulo I, written $p \equiv q$, iff $\varphi(p) = \varphi(q)$.

Proposition

Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be an ideal. Then the congruence modulo I is an equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$.

An equivalence relation on $\mathbb{K}[x_1, \ldots, x_n]$ partitions $\mathbb{K}[x_1, \ldots, x_n]$ into a collection of disjoint subsets called equivalence classes.
For any $p \in \mathbb{K}[x_1, \ldots, x_n]$, the class of p is the set

$$\overline{p} = \{q \in \mathbb{K}[x_1, \ldots, x_n] \mid \varphi(q) = \varphi(p)\}$$

Very often, in the following we will identify $\overline{p}$ and p ! We can define the following ring operations:

$$\overline{p} + \overline{q} = \overline{p + q}$$
$$\overline{p} \times \overline{q} = \overline{p \times q}$$

This is also a Vector space : $\mathbb{K}[x_1, \ldots, x_n]/\ker(\varphi) = \mathbb{K}[x_1, \ldots, x_n]/I$

Basis of the vector space

Definition (staircase of an ideal)

Let I be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$ and $(G, <)$ a Gröbner basis of I, then we define the staircase of G by

$$\mathcal{E}(G) = \{t \in T \mid t \text{ is not reducible by} G\} \text{ sorted wrt } < .$$

We say that it is the canonical basis (wrt G) of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$.

Zero dimensional ideal

Recall that we have a criterion to determine when a system of polynomial equations over $\overline{\mathbb{K}}$ has only finitely many solutions:

Theorem

I ideal generated by $\langle f_1, \ldots, f_m \rangle$.
G Gröbner basis of I wrt $<$
The algebraic variety $V_{\overline{\mathbb{K}}}(I)$ is finite iff

$$\forall i \in \{1, \ldots, n\} \text{ there exists } g_i \in G \text{ such that } \mathrm{LT}(g_i) = x_i^{k_i}$$

or $\{t \in T \mid t$ is not reducible by $G\}$ is finite.

Example

$G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$.

Any polynomial

$$f = -x_1^2 - x_2^2 + 7\,x_1\,x_2$$

is first reduced wrt $G_{<_{\text{DRL}}}$

$$NF(f, G_{<_{\text{DRL}}}) \quad =$$

And so in the canonical basis $\mathcal{B}$ the polynomial f can be represented by the vector :

Example

$G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$.

$\mathcal{E}(G) = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\,x_2\}$

$\mathcal{B} = \{\overline{w_1}, \overline{w_2}, \overline{w_3}, \overline{w_4}\}$ is a basis of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$.

Any polynomial

$$f = -x_1^2 - x_2^2 + 7\,x_1\,x_2$$

is first reduced wrt $G_{<_{\text{DRL}}}$

$$\begin{aligned} NF(f, G_{<_{\text{DRL}}}) &= -3\,x_1 - 2\,x_2 + 7\,x_1\,x_2 \\ &= -3\overline{w_2} - 2\overline{w_3} + 7\overline{w_4} \end{aligned}$$

And so in the canonical basis $\mathcal{B}$ the polynomial f can be represented by the vector :

$$[0, -3, -2, 7]$$

Ideal of dimension 0

Definition (staircase of an ideal)

Let I be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$ and $(G, <)$ a Gröbner basis of I, then we define the staircase of G by

$$\mathcal{E}(G) = \{t \in T \mid t \text{ is not reducible by } G\} \text{ sorted wrt } <.$$

This a canonical basis (wrt G) of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$.

Ideal of dimension 0

Definition (staircase of an ideal)

Let I be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$ and $(G, <)$ a Gröbner basis of I, then we define the staircase of G by

$$\mathcal{E}(G) = \{t \in T \mid t \text{ is not reducible by } G\} \text{ sorted wrt } <.$$

This a canonical basis (wrt G) of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$.

Theorem (criterion to determine when a system of polynomial equations over $\overline{\mathbb{K}}$ has only finitely many solutions.)

The algebraic variety $V_{\overline{\mathbb{K}}}(\langle f_1, \ldots, f_m \rangle)$ is finite iff $D = \#\mathcal{E}(G) < \infty$. D is the number of solution (counting with multiplicities), in the algebraic closure, of the polynomial system $[f_1, \ldots, f_m]$.

Ideal of dimension 0

Definition (staircase of an ideal)

Let I be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$ and $(G, <)$ a Gröbner basis of I, then we define the staircase of G by

$$\mathcal{E}(G) = \{t \in T \mid t \text{ is not reducible by } G\} \text{ sorted wrt } <.$$

This a canonical basis (wrt G) of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$.

Theorem (criterion to determine when a system of polynomial equations over $\overline{\mathbb{K}}$ has only finitely many solutions.)

The algebraic variety $V_{\overline{\mathbb{K}}}(\langle f_1, \ldots, f_m \rangle)$ is finite iff $D = \#\mathcal{E}(G) < \infty$. D is the number of solution (counting with multiplicities), in the algebraic closure, of the polynomial system $[f_1, \ldots, f_m]$.

We denote by $D = \deg(I) = \#\mathcal{E}(G)$ the dimension of the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$ (this is also the degree of the ideal I).
We assume that:

$$\mathcal{E}(G) = \left\{w_1 = 1 < w_2 < \cdots < w_{\deg(I)}\right\}$$

What is the expected number of solutions ?

Theorem (Bezout's Bound)

Assume that $\mathbb{K}$ is algebraically closed. Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be a zero-dimensional ideal generated by $[f_1, \ldots, f_n]$ then

$$D = \deg(I) \leqslant \prod_{i=1}^{n} \deg(f_i)$$

For instance for n quadratic equations in n variables the bound is 2^n.

What is the expected number of solutions ?

Theorem (Bezout's Bound)

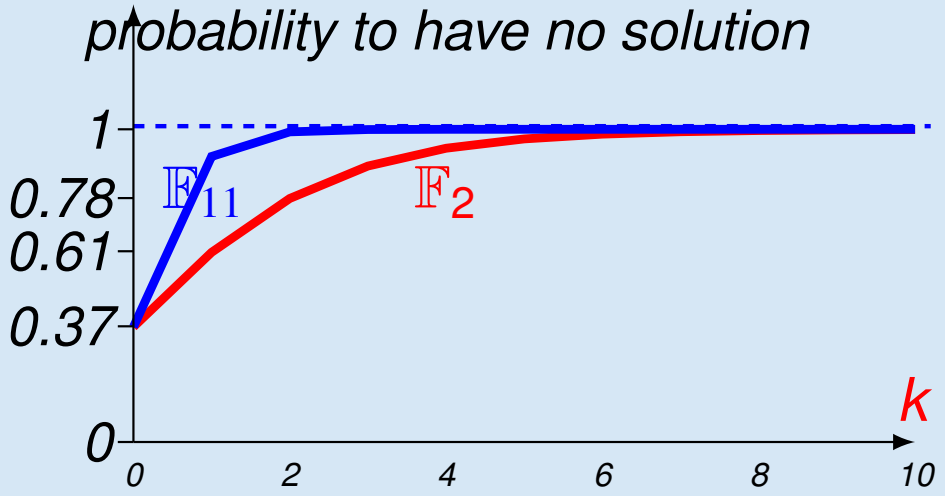
Assume that $\mathbb{K}$ is algebraically closed. Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be a zero-dimensional ideal generated by $[f_1, \ldots, f_n]$ then

$$D = \deg(I) \leqslant \prod_{i=1}^{n} \deg(f_i)$$

For instance for n quadratic equations in n variables the bound is 2^n.

Theorem (*Fusco and Bach*)

The probability that a random polynomial system of $n + k$ random equations of degree-d ($d \geqslant 2$) in n variables over $\mathbb{F}_p$, has no solution is $e^{-p^{-k}}$ (asymptotically)



Example

In $\mathbb{Q}[x_1, x_2]$ the list $G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$; we try to compute a lexicographical ($x_2 > x_1$) Gröbner basis of $Id(G)$.

staircase: $\mathcal{E}(G) = \{t \in T \mid t \text{ is not reducible by } G\}$

Example

In $\mathbb{Q}[x_1, x_2]$ the list $G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$; we try to compute a lexicographical ($x_2 > x_1$) Gröbner basis of $Id(G)$.

staircase: $\mathcal{E}(G) = \left\{t \in T \mid t \text{ is not reducible by } [x_1^2, x_2^2]\right\}$

Example

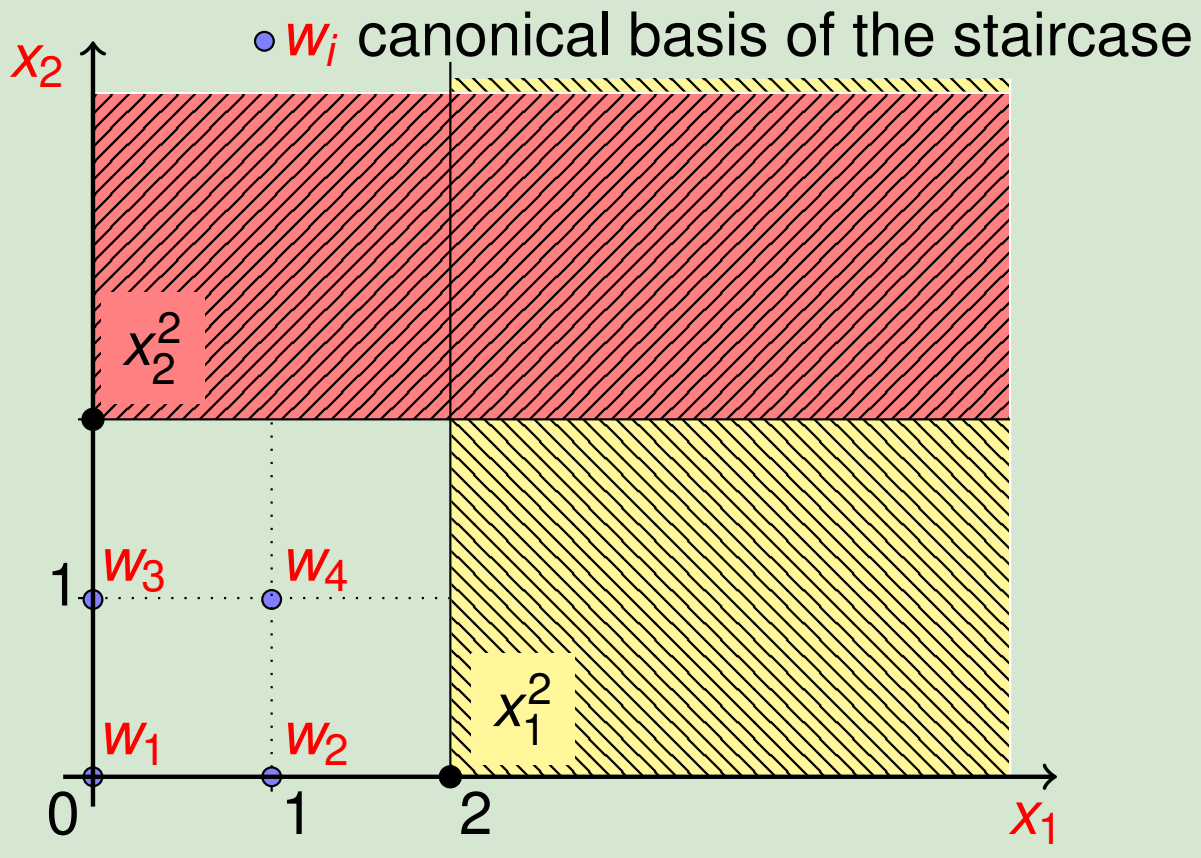
In $\mathbb{Q}[x_1, x_2]$ the list $G_{<_{\mathrm{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$; we try to compute a lexicographical ($x_2 > x_1$) Gröbner basis of $Id(G)$.

staircase: $\mathcal{E}(G) = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\,x_2\}$

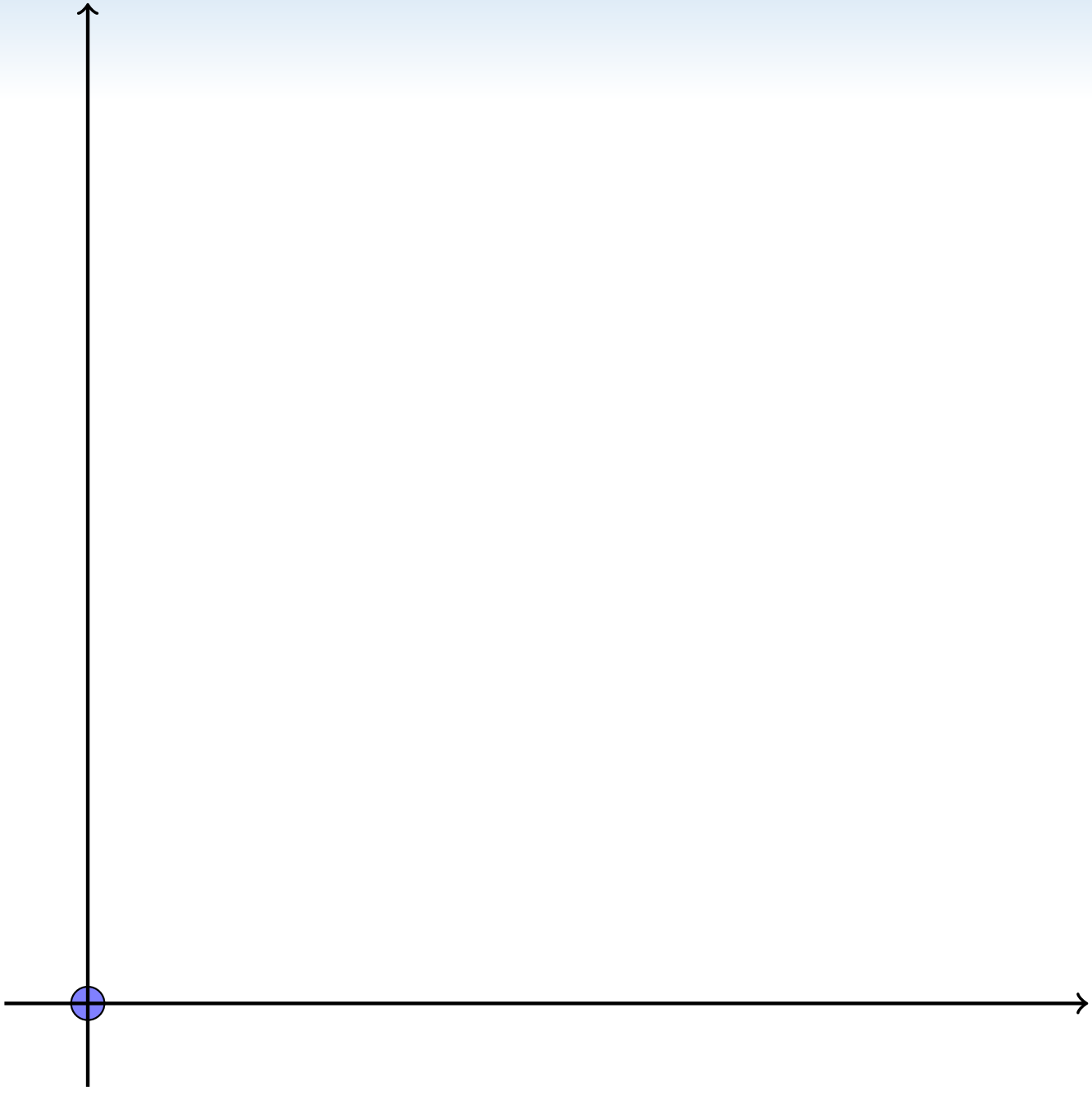
Example

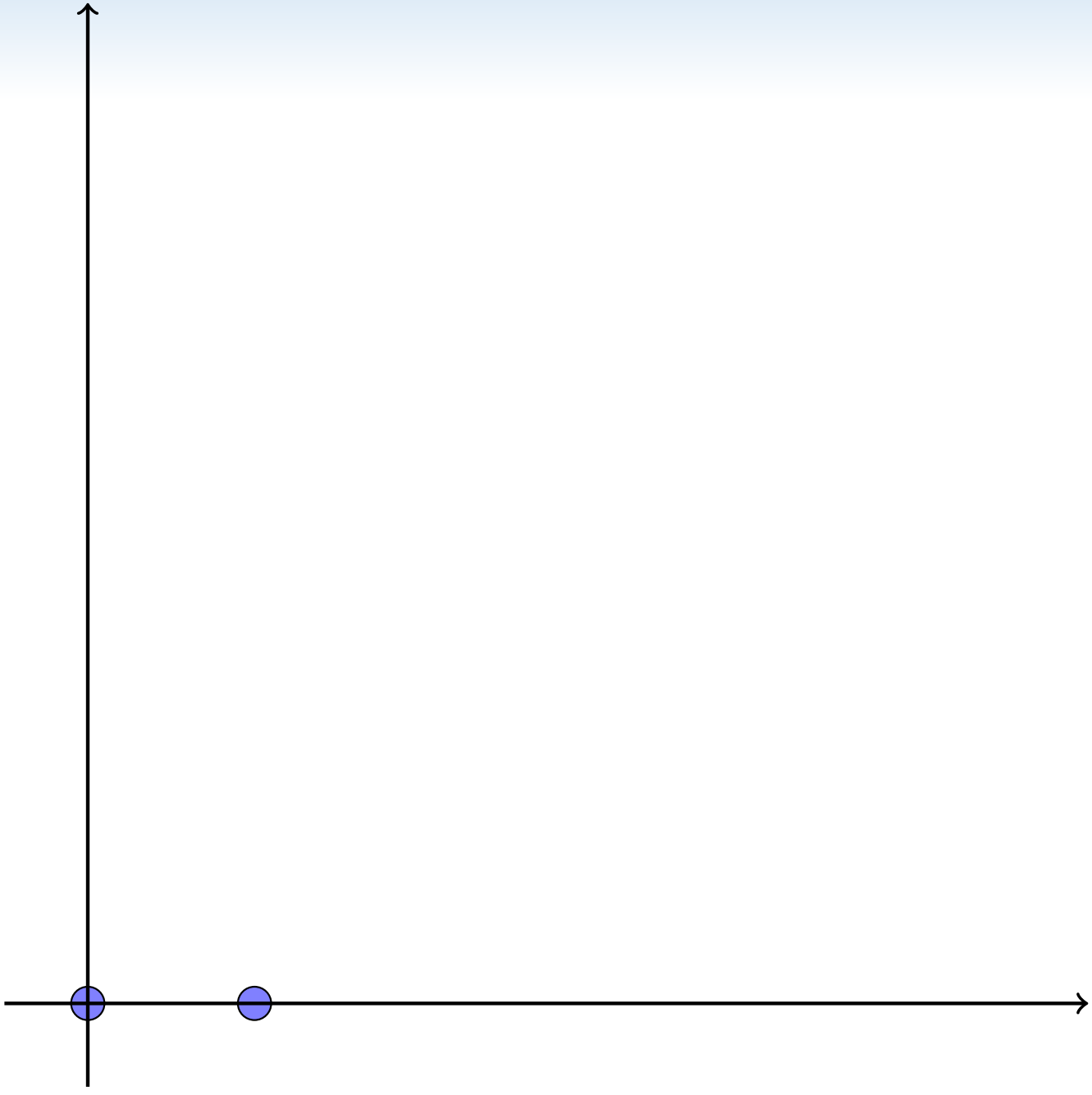
In $\mathbb{Q}[x_1, x_2]$ the list $G_{<_{\text{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt a DRL ordering such that $x_2 > x_1$; we try to compute a lexicographical ($x_2 > x_1$) Gröbner basis of $Id(G)$.

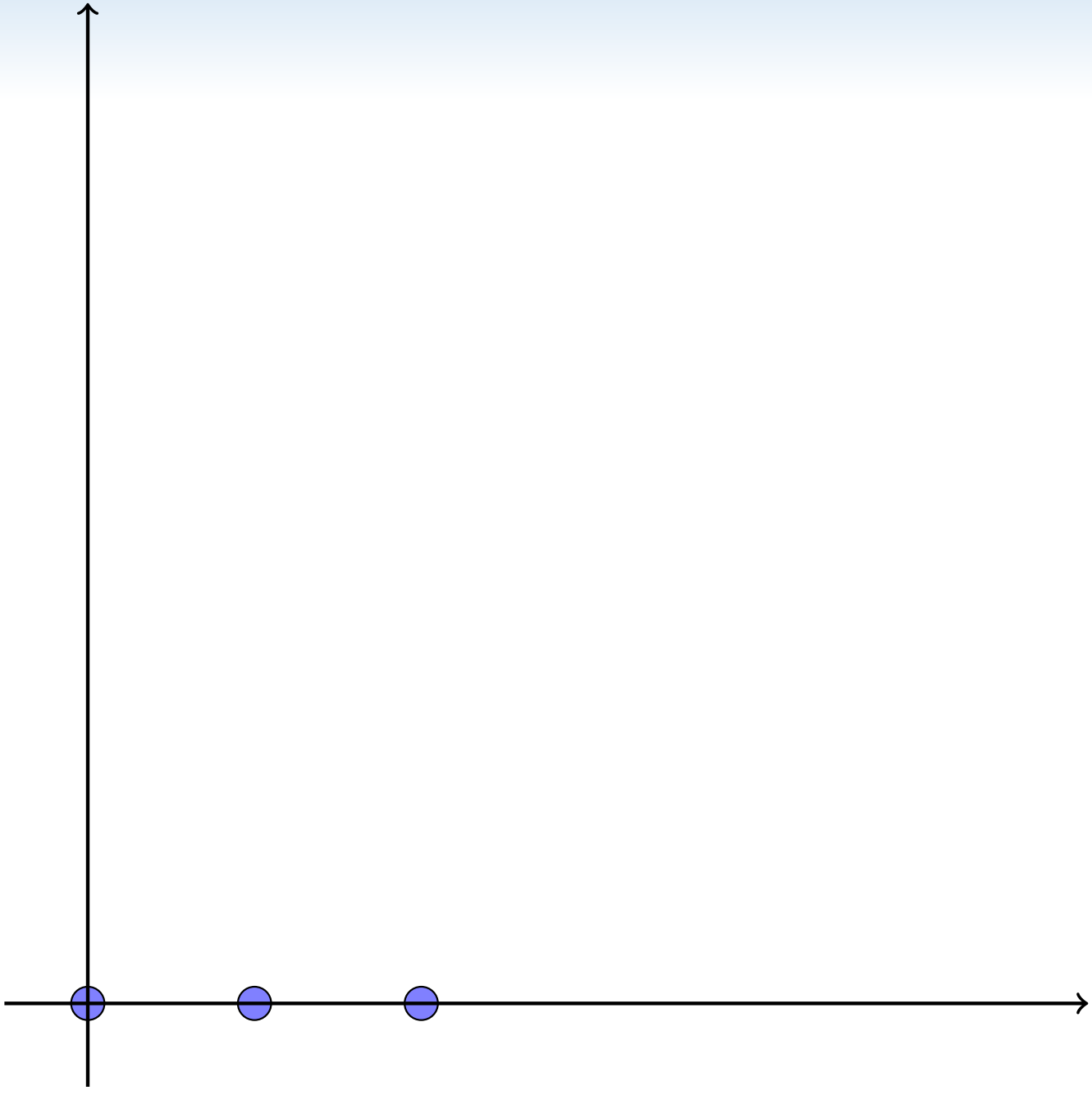
staircase: $\mathcal{E}(G) = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\,x_2\}$

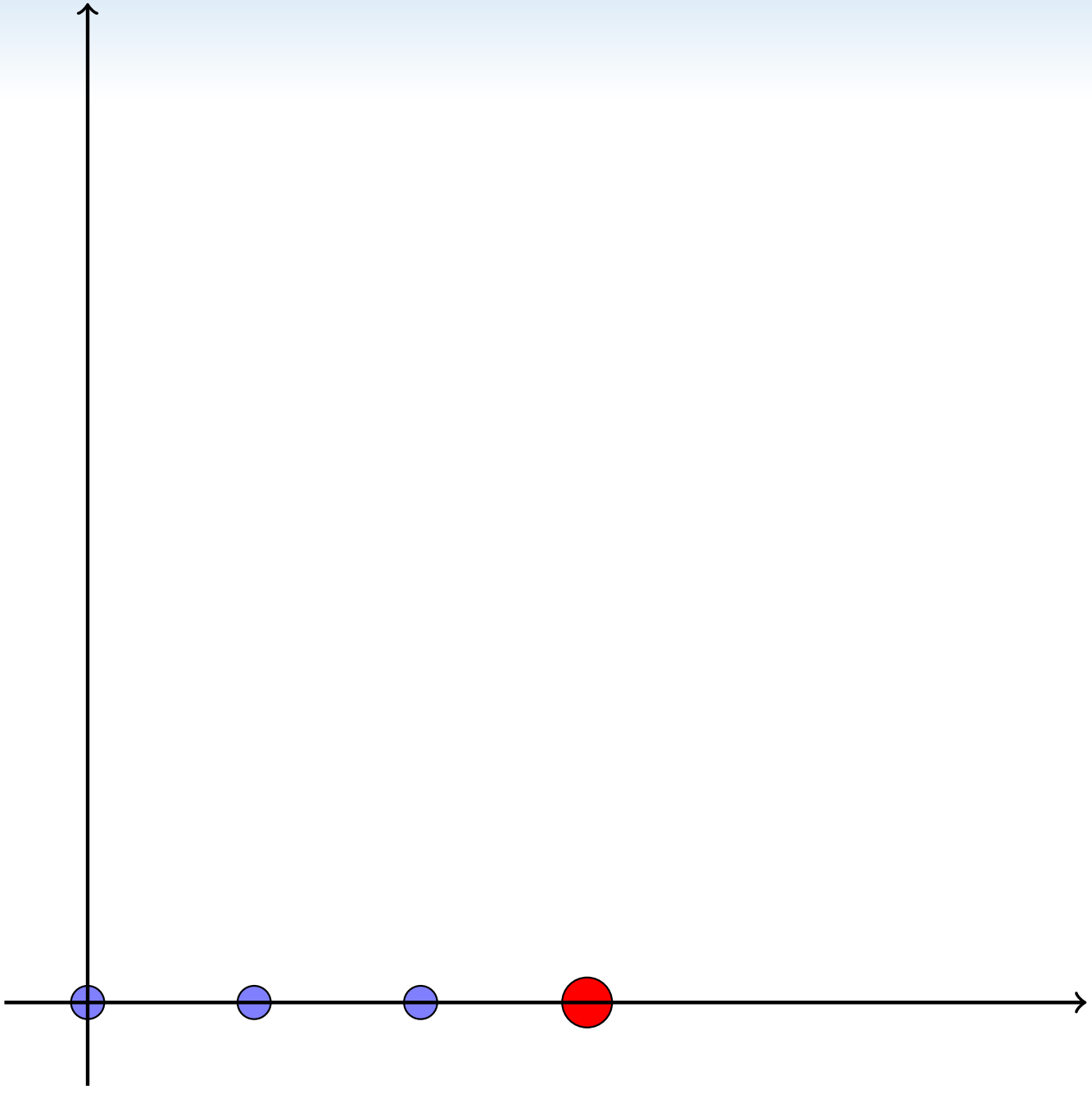


The system has 4 complex roots.

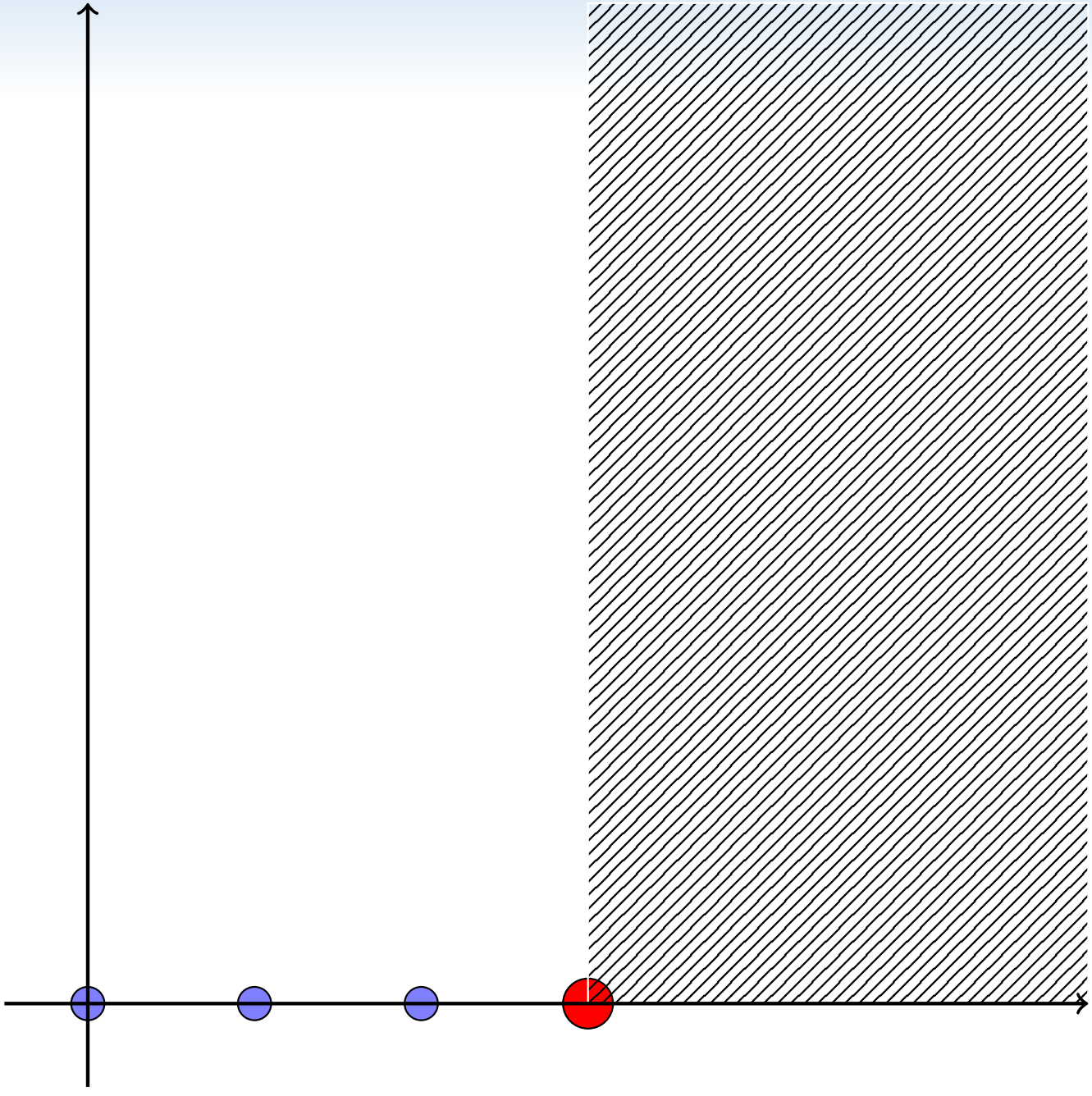




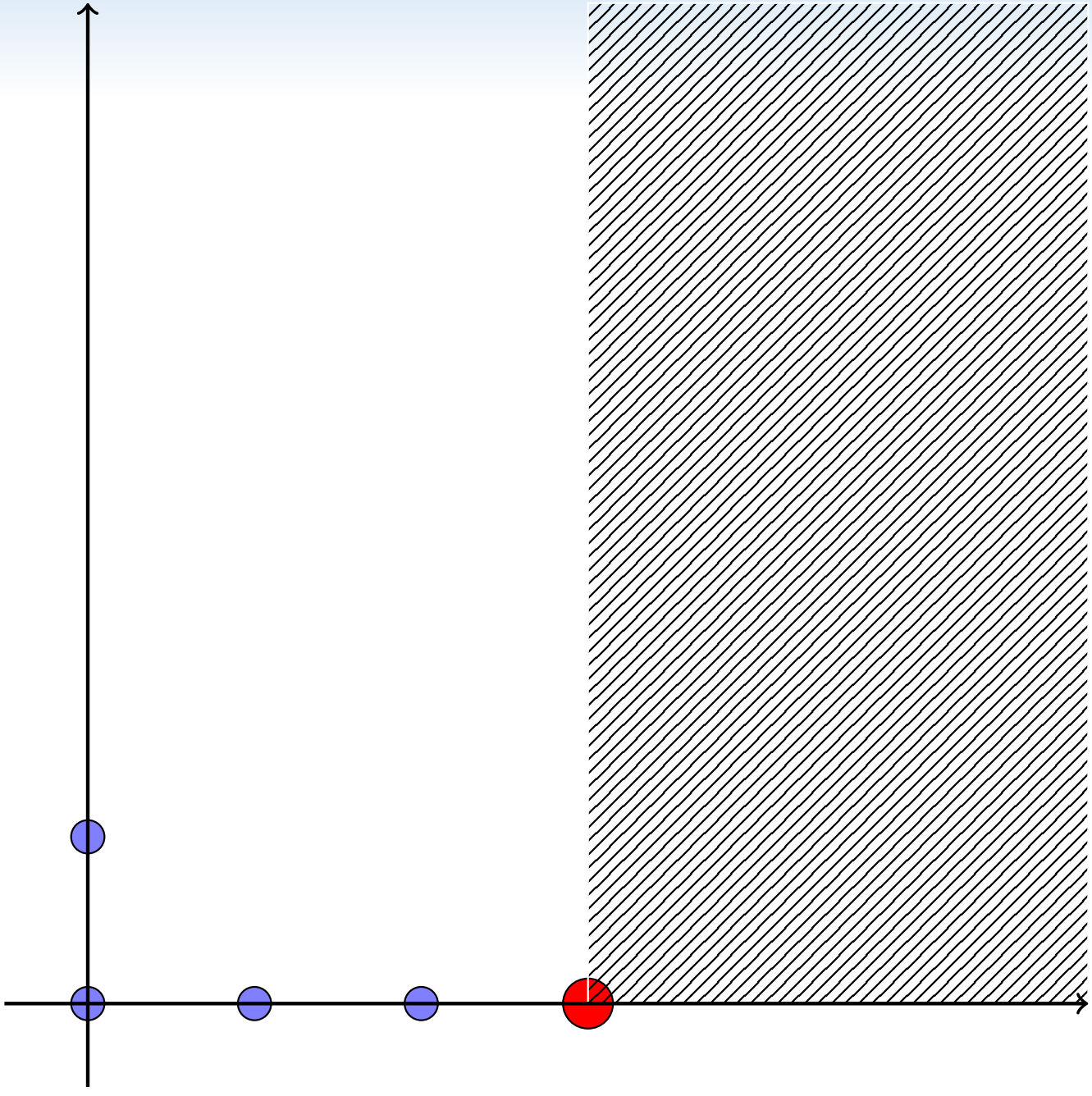




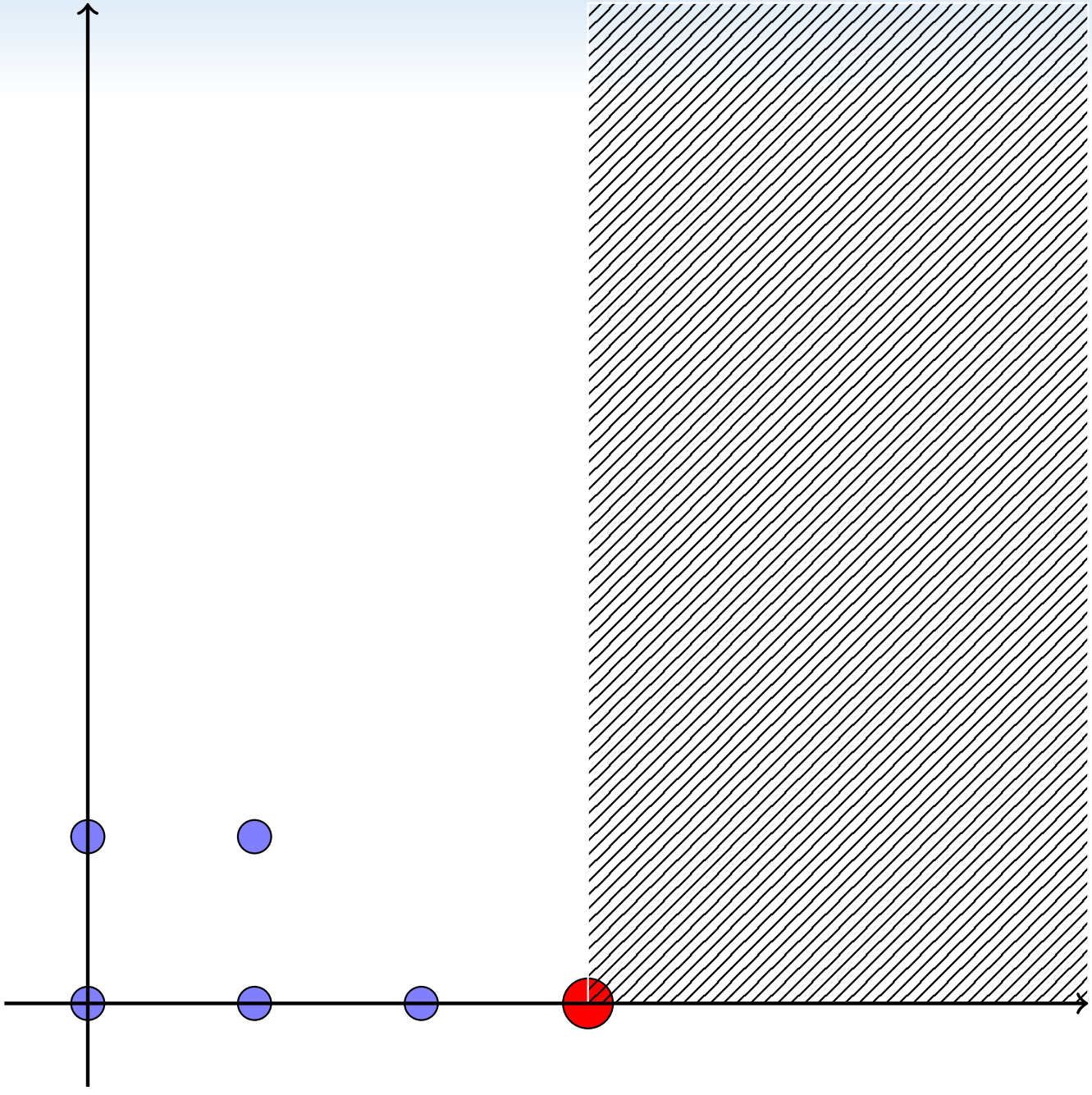
$g_1 = x^3 + \cdots$



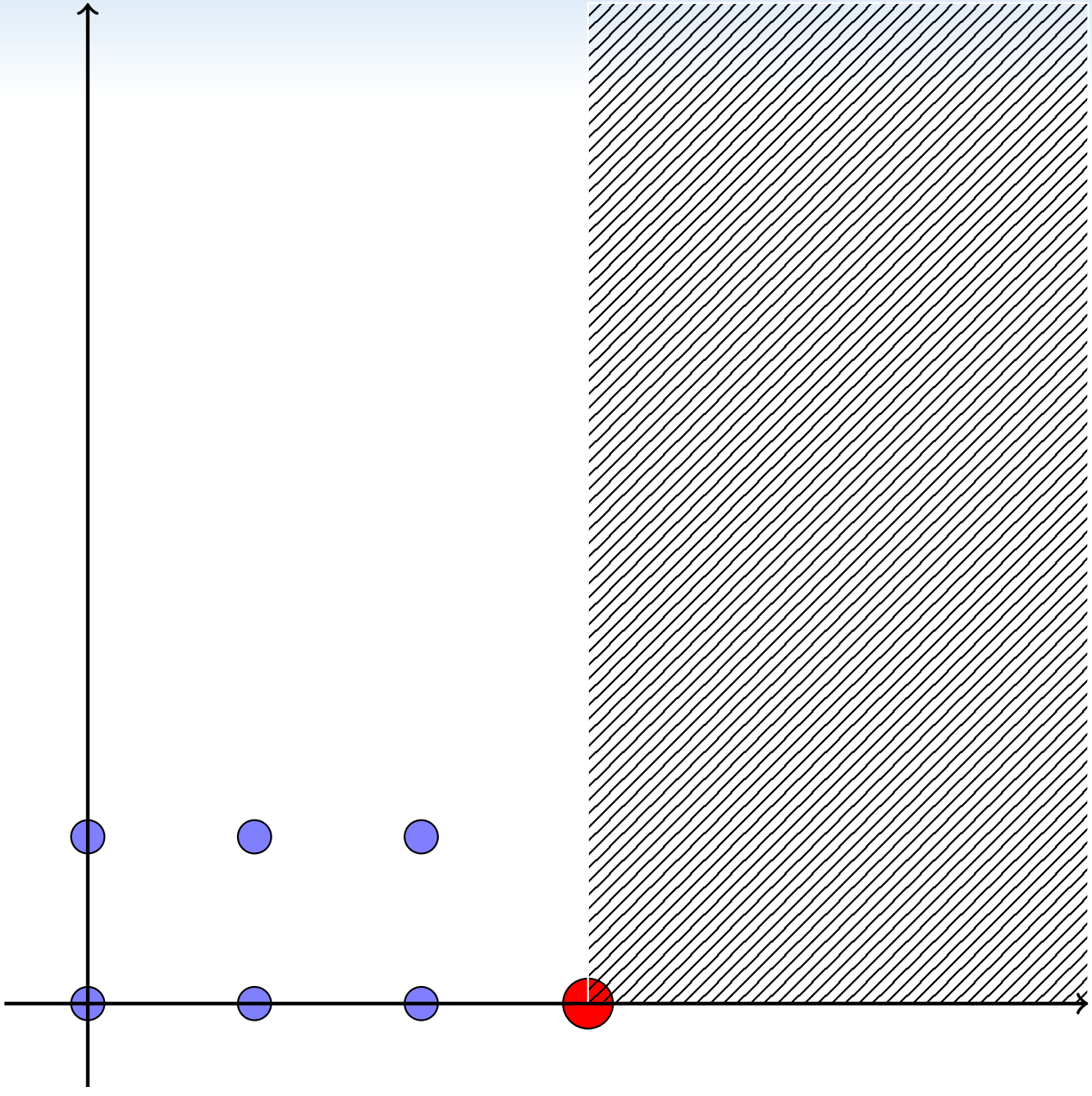
$g_1 = x^3 + \cdots$



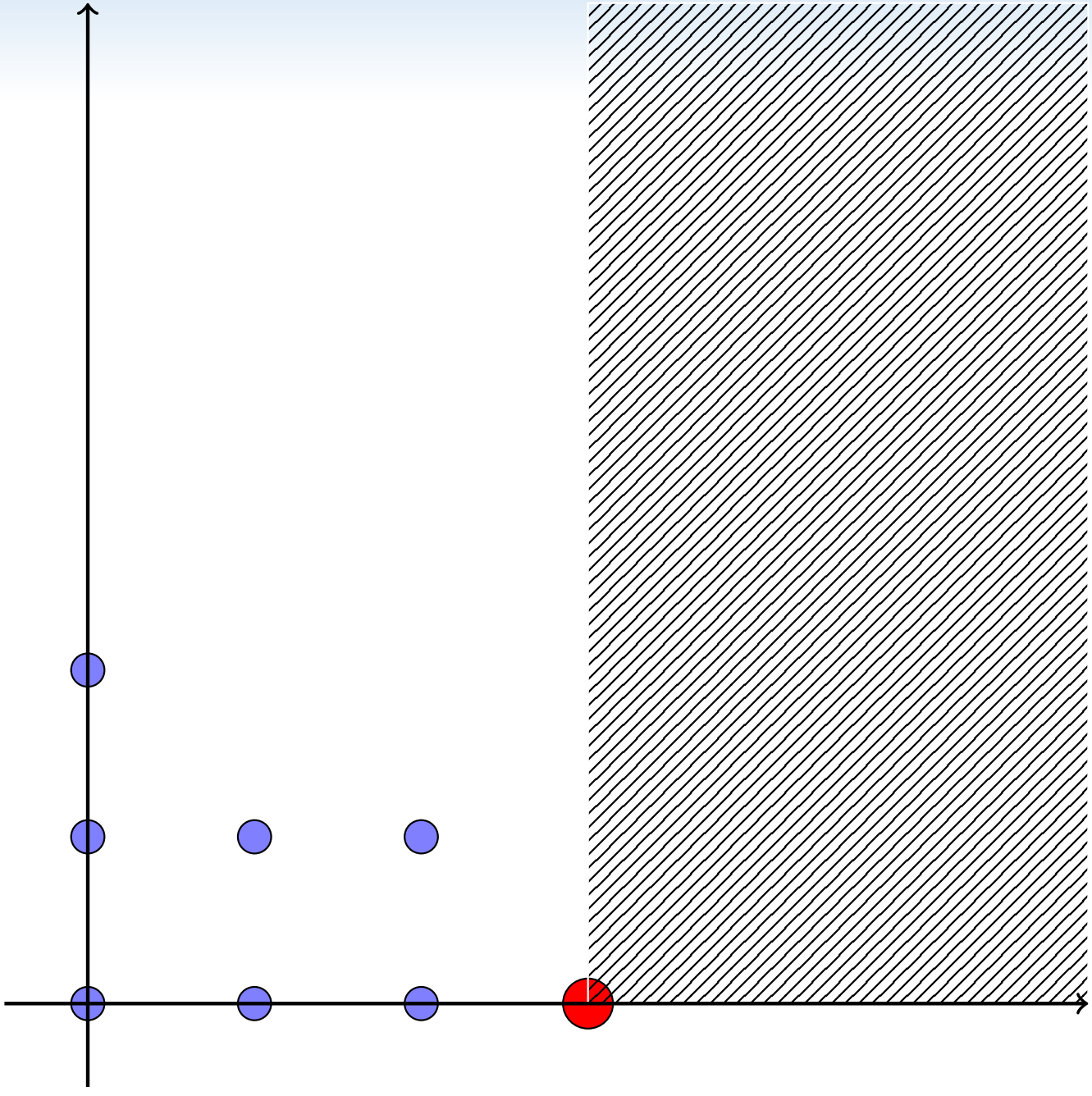
$g_1 = x^3 + \cdots$



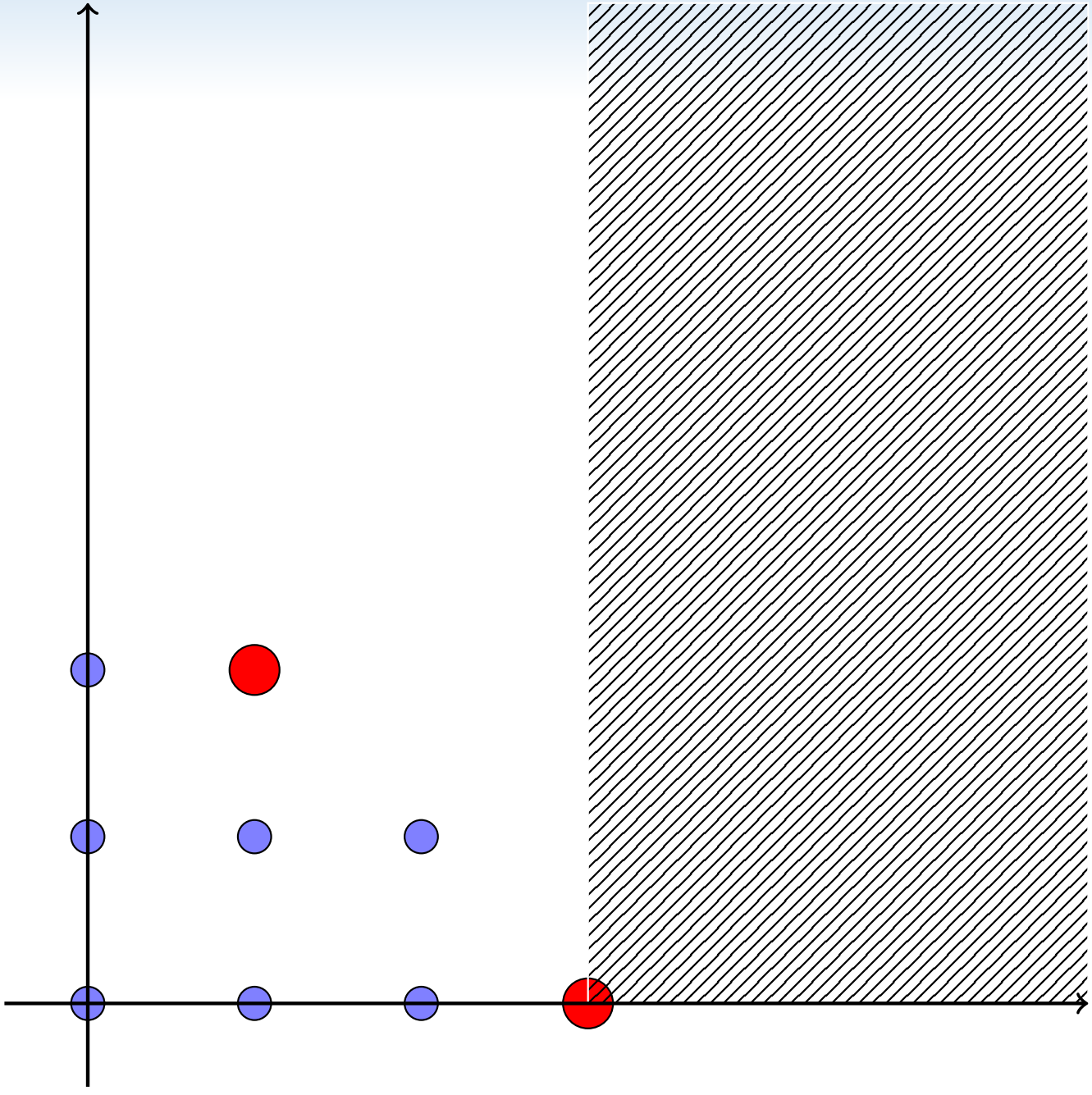
$g_1 = x^3 + \cdots$



$g_1 = x^3 + \cdots$

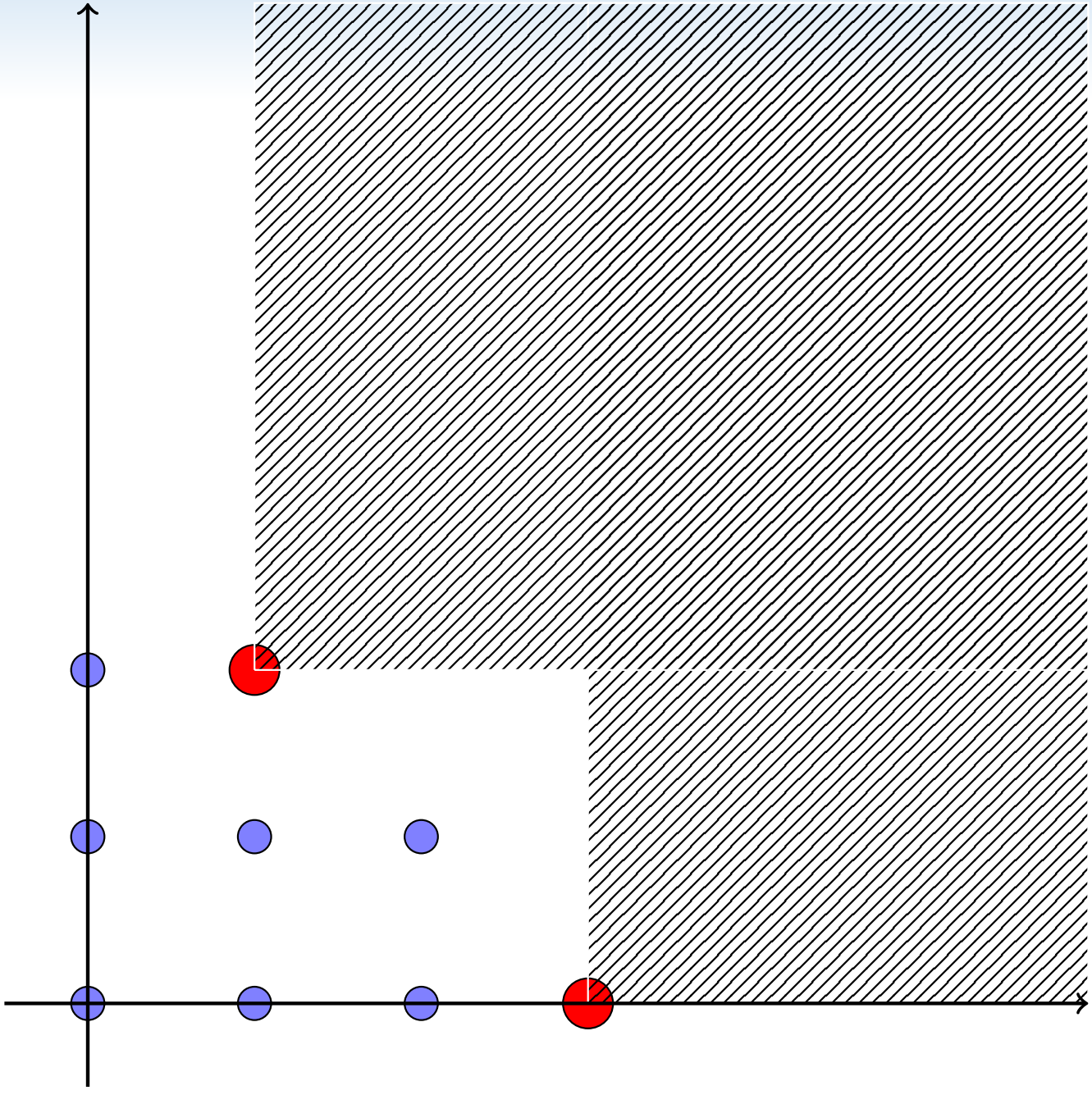


$g_1 = x^3 + \cdots$



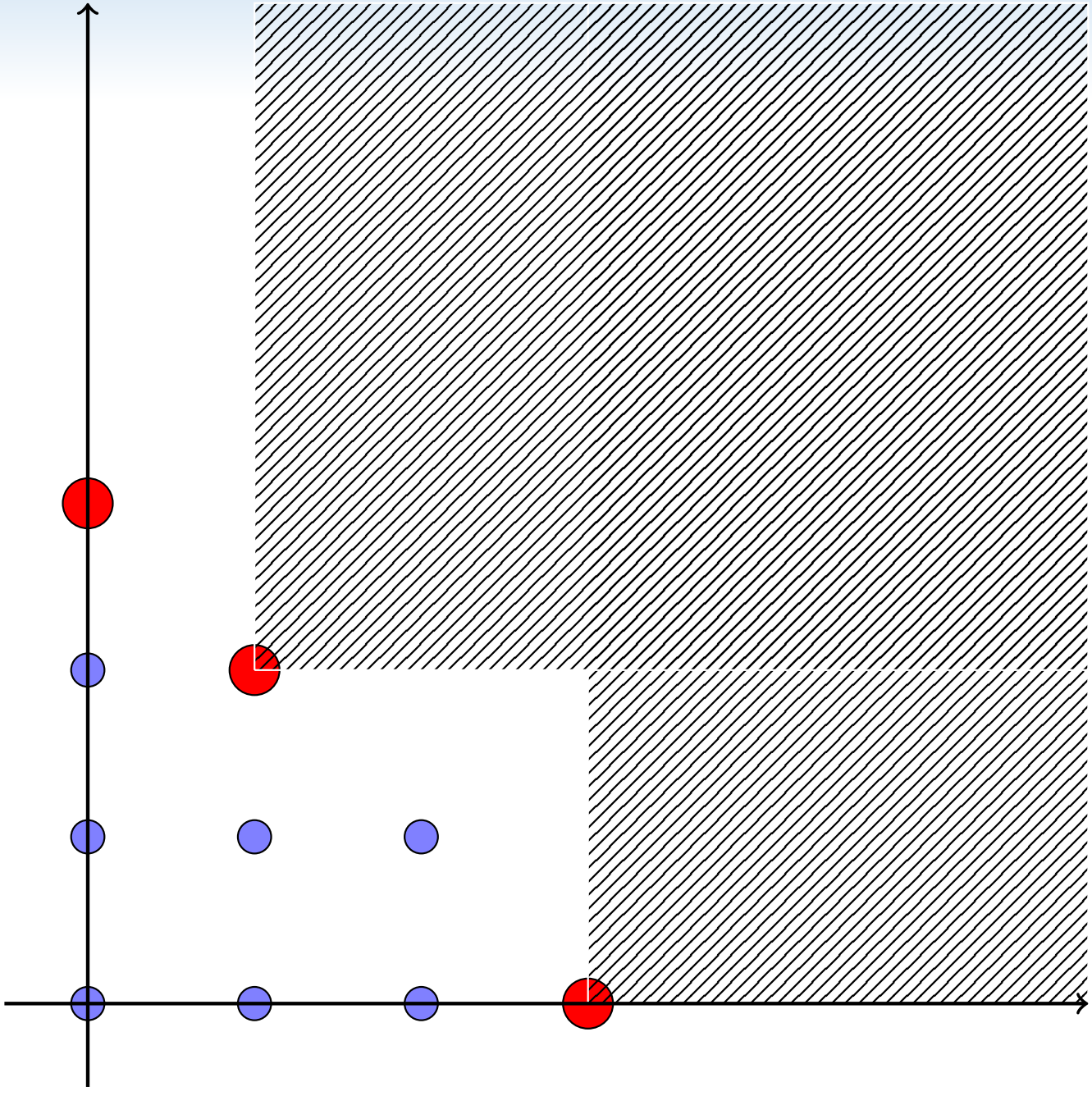
$$g_1 = x^3 + \cdots$$

$$g_2 = x\,y^2 + \cdots$$



$$g_1 = x^3 + \cdots$$

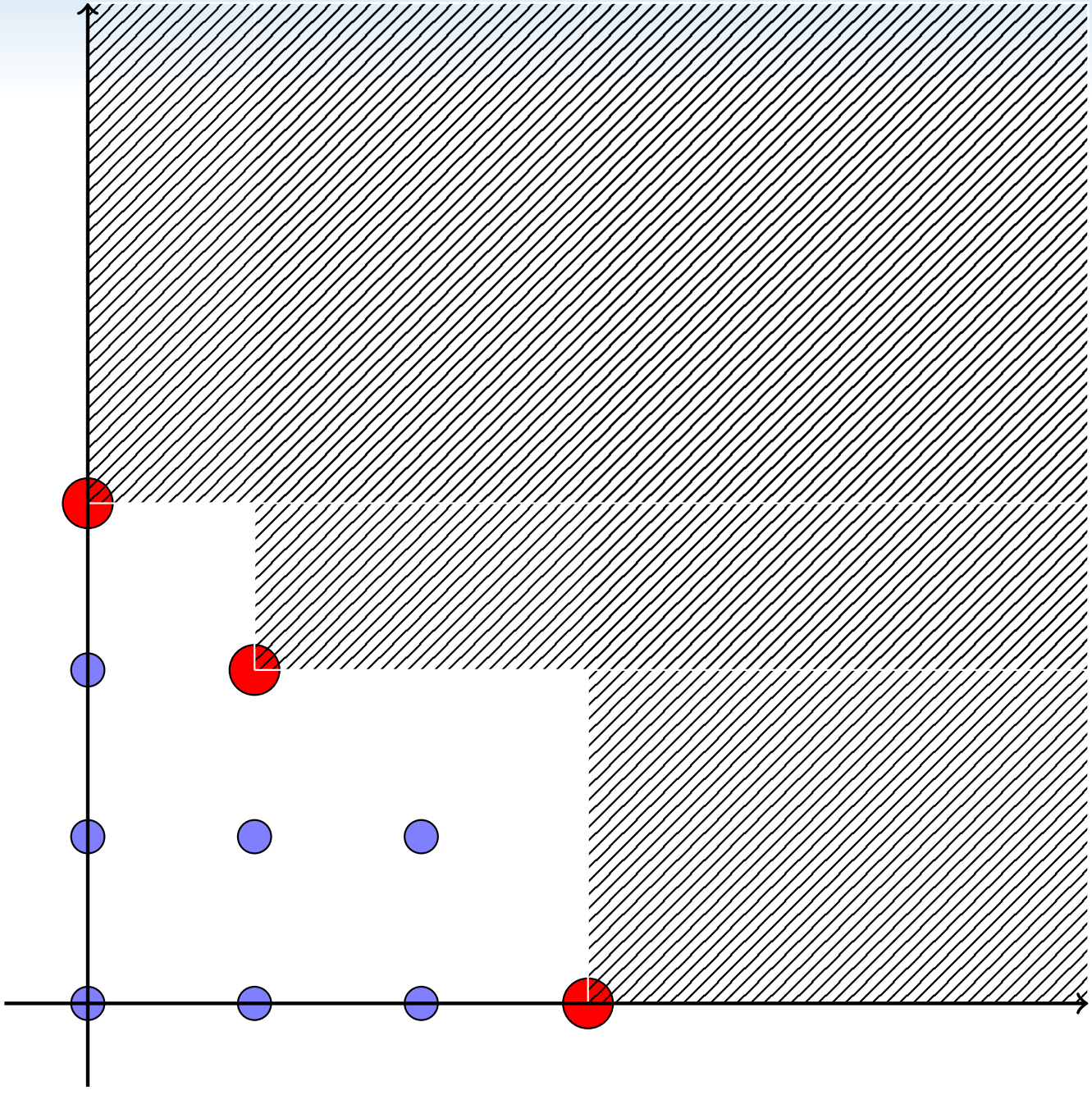
$$g_2 = x\,y^2 + \cdots$$



$$g_1 = x^3 + \cdots$$

$$g_2 = x\,y^2 + \cdots$$

$$g_3 = y^3 + \cdots$$



$$g_1 = x^3 + \cdots$$

$$g_2 = x\,y^2 + \cdots$$

$$g_3 = y^3 + \cdots$$

FGLM algorithm

FGLM*=*Faugère, Giani, Lazard, Mora, JSC, 1994

Algorithm (FGLM [6])

Input: $<_2$ a monomial ordering and NF a normal form.

Output: reduced Gröbner basis I wrt $<_2$

where $I = \{f \in \mathbb{K}[x_1, \ldots, x_n] \mid \mathrm{NF}(f) = 0\}$

$L := [\,]$ // list of next terms to study

$S := [\,]$ // the staircase wrt the new ordering $<_2$

$V := []$ // $V = \mathrm{NF}(S)$

$G := [\,]$, $t := 1$

infinite loop

...

Algorithm (FGLM [6])

...

infinite loop

$v := \mathrm{NF}(t)$ and $s := \#S$ is the number of elements in S.

if $v \in \mathrm{Vect}_{\mathbb{K}}(V)$ then

we can find (λ_i) s.t. $v = \sum_{i=1}^{s} \lambda_i \cdot V_i$

$$G := G \cup \left[t - \sum_{i=1}^{s} \lambda_i \cdot S_i \right]$$

else

$S := S \cup [t]$ and $V := V \cup [v]$

$L := \mathrm{Sort}\left(L \cup [x_i\, t \mid i = 1, \ldots, n], <_2\right)$

Sort L by increasing order (wrt $<_2$) and remove duplicates and multiple of $\mathrm{LT}(G)$

if $L = \varnothing$ then

return G

$t := \mathrm{first}(L)$ and remove t from L.

Compute normalForm using linear algebra

We are working in the $\mathbb{K}$-vector space $\mathbb{K}[x_1, \ldots, x_n]/I$ using the canonical basis wrt $<$ (old ordering: G is a Gröbner basis wrt $<$):

$$\mathcal{E}(G) = \left\{ w_1 = 1 < w_2 < \cdots < w_{\deg(I)} \right\}.$$

If $f \in \mathbb{K}[x_1, \ldots, x_n]$ we can compute $\mathrm{NF}(f, G) \in \mathbb{K}[x_1, \ldots, x_n]/I$ using the FULLREDUCTION algorithm but it is difficult to obtain a precise bound of complexity !

This can be done with linear algebra; more precisely only matrix vector products.

Border of an ideal

The staircase $\mathcal{E}(G)$ is stable under division:

Proposition

If $1 \neq e \in \mathcal{E}(G)$ then for all i such that x_i divides e we have $\frac{e}{x_i} \in \mathcal{E}(G)$.

We try to estimate the number of elements in G with respect to $\deg(I) = D$; to this end we define the border of the staircase:

Definition (Border of a Gröbner basis)

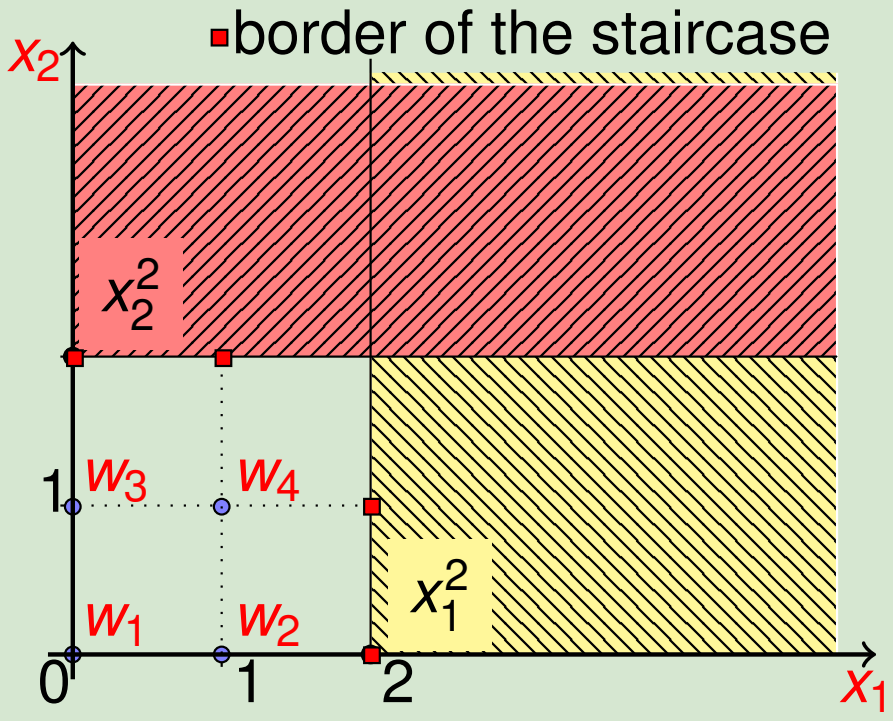
let $\mathcal{E}(G)$ be the canonical basis of $\mathbb{K}[x_1, \ldots, x_n] / I$, then the border of G is:

$$\mathcal{F}(G) = \{x_i e \mid e \in \mathcal{E}(G),\ 1 \leqslant i \leqslant n \text{ and } x_i\, e \notin \mathcal{E}(G)\}$$

Example

$G_{<_{\mathrm{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ is a Gröbner basis wrt the DRL ordering with $x_2 > x_1$.

border of G: $\mathcal{F}(G) = \{x_i e \mid e \in \mathcal{E}(G),\ 1 \leqslant i \leqslant n \text{ and } x_i\, e \notin \mathcal{E}(G)\}$



$$\mathcal{F}(G) = \left\{x_1^2,\ x_2^2,\ x_1^2 x_2,\ x_1 x_2^2\right\}$$

Proposition

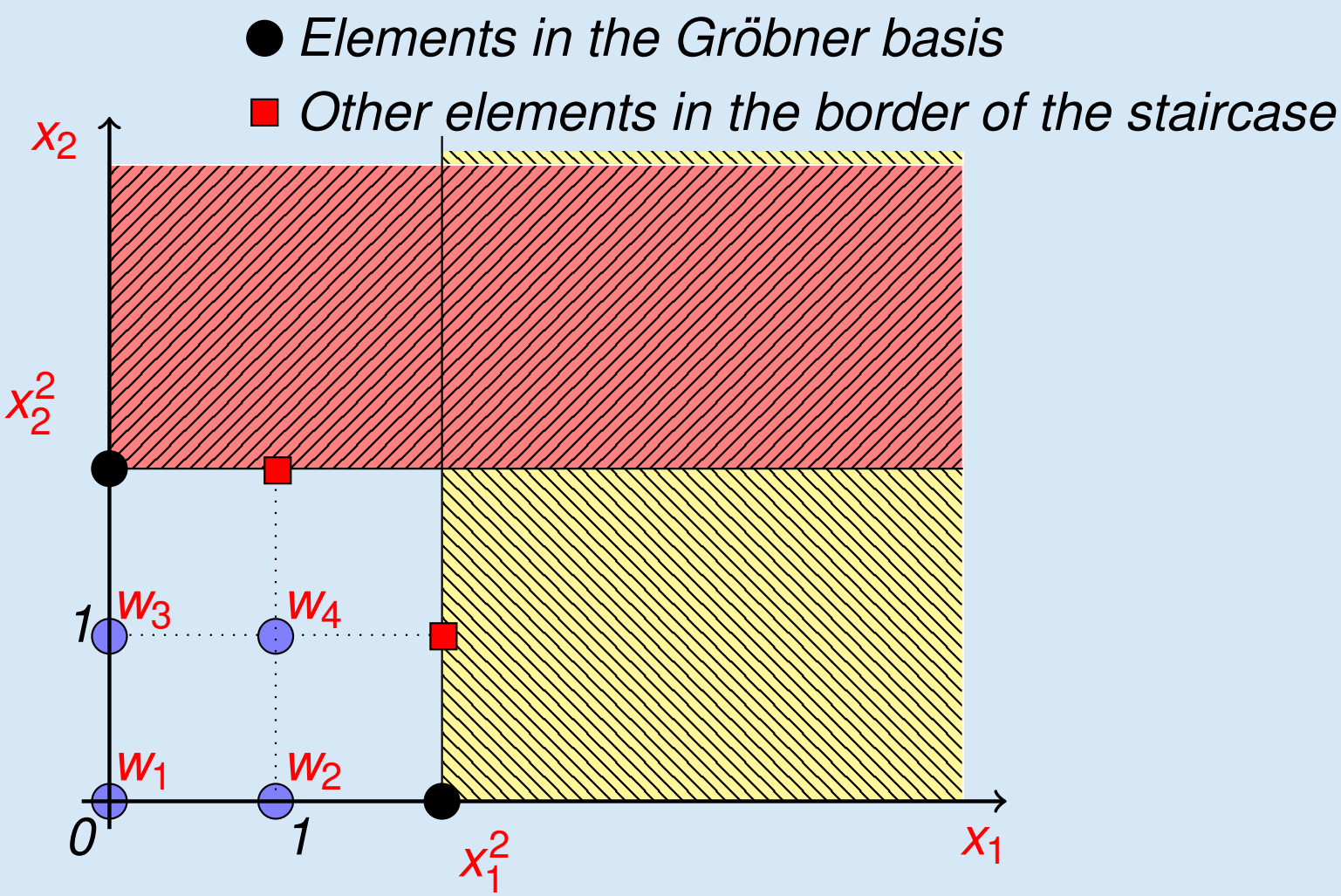
$\dim(I) = 0$, $(G, <)$ *a Gröbner basis, then*

$$\mathrm{LT}(G) \subset \mathcal{F}(G) \subset \mathrm{LT}(G) \cup \left\{ x_j t' \mid t' \in \mathcal{F}(G) \text{ and } 1 \leqslant j \leqslant n \right\}$$

Proposition

$\dim(I) = 0$, $(G, <)$ *a Gröbner basis, then*

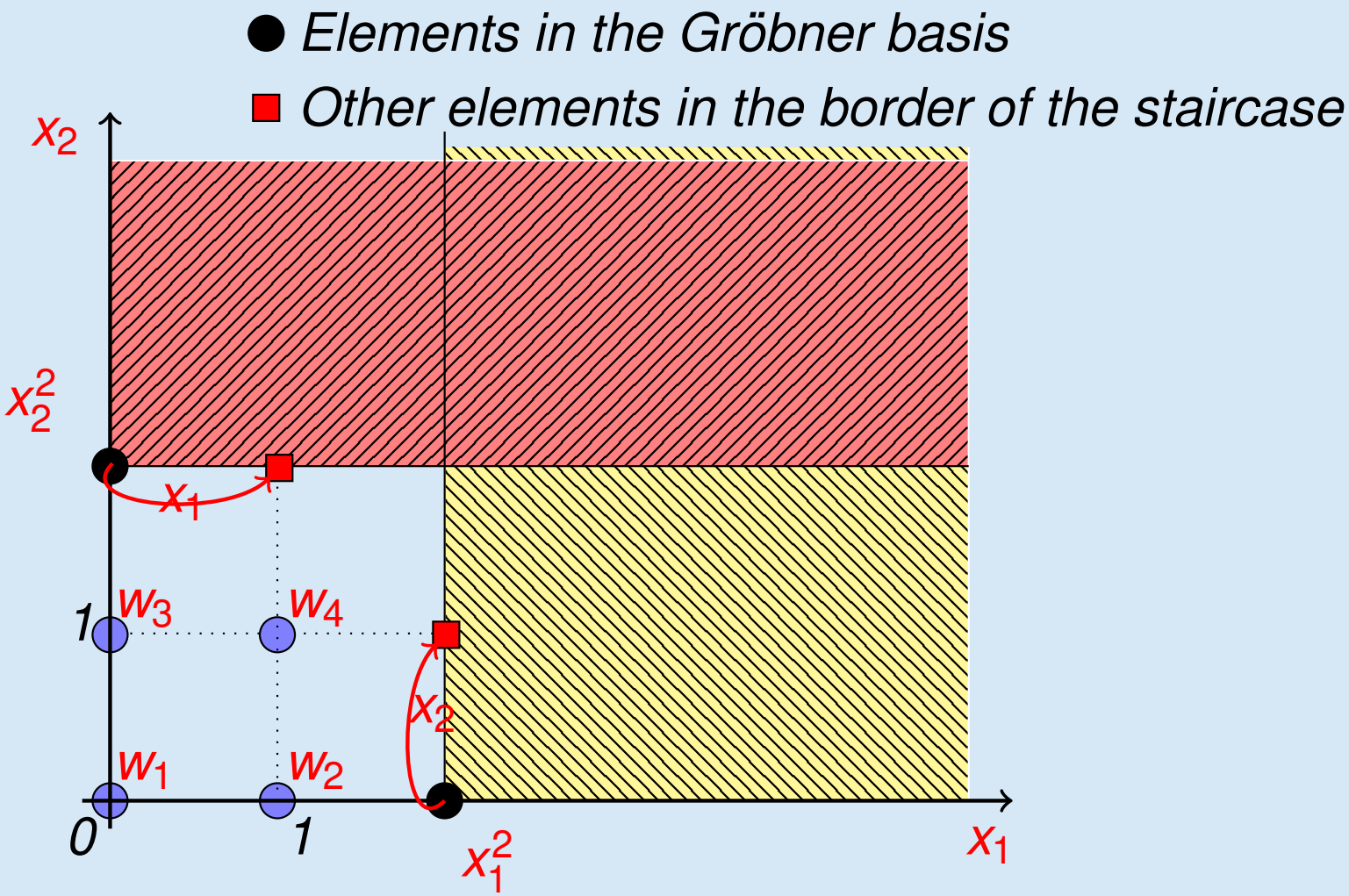
$$\mathrm{LT}(G) \subset \mathcal{F}(G) \subset \mathrm{LT}(G) \cup \left\{ x_j t' \mid t' \in \mathcal{F}(G) \text{ and } 1 \leqslant j \leqslant n \right\}$$



Proposition

$\dim(I) = 0$, $(G, <)$ *a Gröbner basis, then*

$$\mathrm{LT}(G) \subset \mathcal{F}(G) \subset \mathrm{LT}(G) \cup \left\{ x_j t' \mid t' \in \mathcal{F}(G) \text{ and } 1 \leqslant j \leqslant n \right\}$$

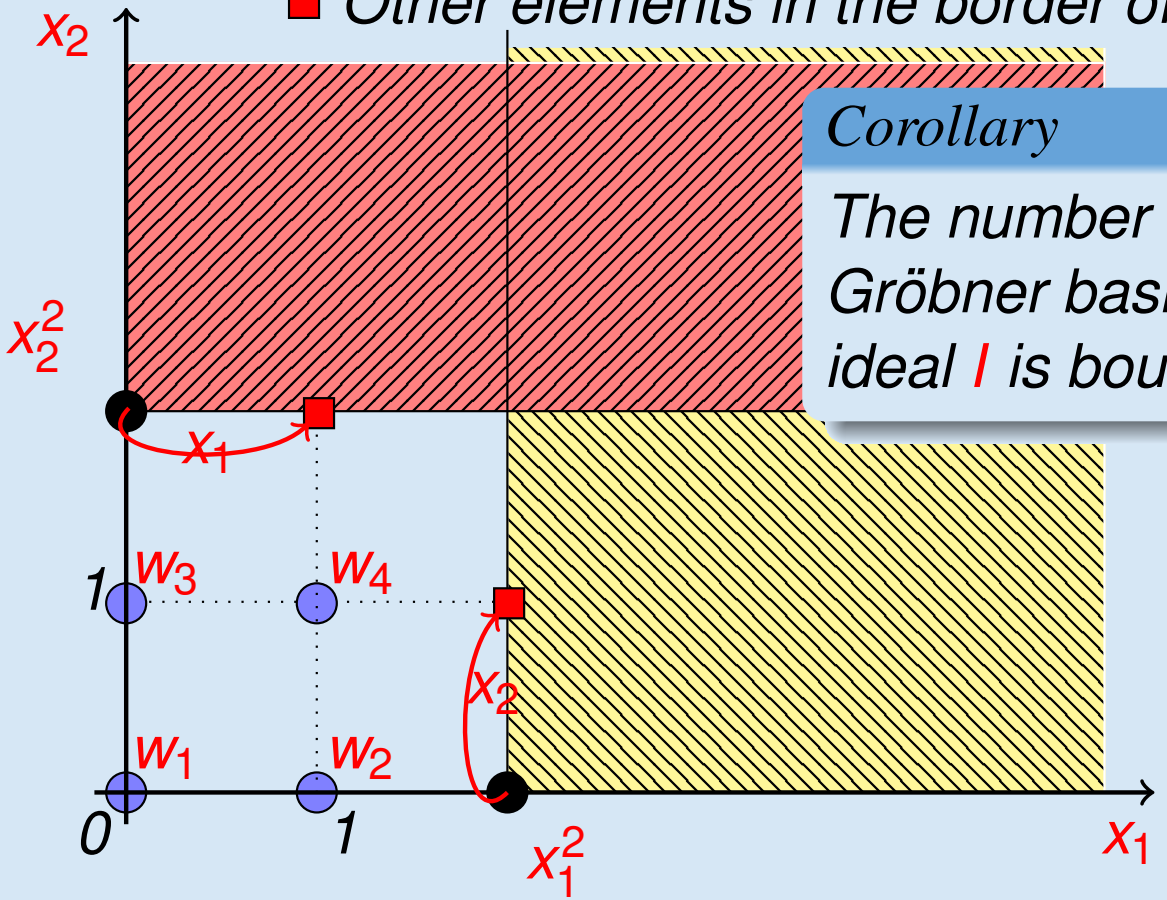


Proposition

$\dim(I) = 0$, $(G, <)$ *a Gröbner basis, then*

$$\mathrm{LT}(G) \subset \mathcal{F}(G) \subset \mathrm{LT}(G) \cup \left\{ x_j t' \mid t' \in \mathcal{F}(G) \text{ and } 1 \leqslant j \leqslant n \right\}$$

- *Elements in the Gröbner basis*
- *Other elements in the border of the staircase*



Corollary

The number of elements of a Gröbner basis of a zero-dimensional ideal I is bounded by $n \deg(I)$.

Computing normal Form using linear algebra

We take advantage of the structure of the vector space to compute in polynomial time the normal forms. When running the FGLM algorithm we have to compute $\varphi_I(t)$ in the following case:

Computing normal Form using linear algebra

We take advantage of the structure of the vector space to compute in polynomial time the normal forms. When running the FGLM algorithm we have to compute $\varphi_I(t)$ in the following case:

- if $t \in \mathcal{E}(G)$ we have $\varphi_I(t) = t$ hence no computation !
- if $t = LT(g)$ for some $g \in G$ (g monic) then $\varphi_I(t) = t - g$
- if $t \in \mathcal{F}(G)$ and $t \notin LT(G)$ then using the previous proposition we have $t = x_j\, t'$ and since $t' < t$ we have already computed $p = \varphi_I(t')$ and so we have to compute $\varphi_I(t) = \varphi_I(x_i \cdot p)$.

Computing normal Form using linear algebra

We take advantage of the structure of the vector space to compute in polynomial time the normal forms. When running the FGLM algorithm we have to compute $\varphi_I(t)$ in the following case:

- if $t \in \mathcal{E}(G)$ we have $\varphi_I(t) = t$ hence no computation !
- if $t = LT(g)$ for some $g \in G$ (g monic) then $\varphi_I(t) = t - g$
- if $t \in \mathcal{F}(G)$ and $t \notin LT(G)$ then using the previous proposition we have $t = x_j\, t'$ and since $t' < t$ we have already computed $p = \varphi_I(t')$ and so we have to compute $\varphi_I(t) = \varphi_I(x_i \cdot p)$.

Definition (Multiplication by one variable)

For all $1 \leqslant k \leqslant n$, consider the following linear map:

$$\phi_i : f \longmapsto \varphi_I(x_i\, f)$$

In the canonical basis the matrix representation of ϕ_i is the so called multiplication matrix M of size $\deg(I) \times \deg(I)$ such that :

$$M_{i,j}^{(k)} = \text{ the coefficient of } w_i \text{ in } \varphi_I(x_k\, w_j)$$

Example

$$G_{<_{\mathrm{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$$

$$\mathcal{E}(G) = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\,x_2\}$$

We can compute the matrix multiplication by x_1 and x_2:

$$M^{(1)} = \begin{array}{c|cccc} & x_1\,w_1 & x_1\,w_2 & x_1\,w_3 & x_1\,w_4 \\ w_1 & 0 & -1 & 0 & 3 \\ w_2 & 1 & 1 & 0 & 6 \\ w_3 & 0 & 3 & 0 & -4 \\ w_4 & 0 & 0 & 1 & 1 \end{array}$$

$$M^{(2)} = \begin{array}{c|cccc} & x_2\,w_1 & x_2\,w_2 & x_2\,w_3 & x_2\,w_4 \\ w_1 & 0 & 0 & 1 & -2 \\ w_2 & 0 & 0 & 2 & 3 \\ w_3 & 1 & 0 & -1 & 6 \\ w_4 & 0 & 1 & 0 & -1 \end{array}$$

Kronecker's delta: $\delta_{i,j}$ =1 if $i = j$ else 0 .

Algorithm Matrix multiplication

Initialisation of the matrices

$\mathcal{E}(G) = \{w_1 = 1 < w_2 < \cdots < w_{\deg(I)}\}$ canonical basis wrt G.

$N := []$ // an array of polynomial indexed by T

// and satisfying that for all $t \in T : N[t] = \mathrm{NF}(t, G, <)$

for i from 1 to $\deg(I)$ do

  $N[w_i] := w_i$

  for each k such that $w_i = x_k\, w_j$ do

    $M_{l,j}^{(k)} := \delta_{l,i}$ for all $l \in \{1, \ldots, n\}$

$F := [x_j\, w_i$ pour $j = 1, \ldots, n$, $i = 1, \ldots, \deg(I)]$

sort F wrt $<$, remove duplicates and the elements of $\mathcal{E}(G)$.

...

Algorithm (Matrix multiplication)

...

sort F wrt $<$, remove duplicates and the elements of $\mathcal{E}(G)$.

for t in F do

  if t is a strict multiple of some leading term of G then

    $t = x_j\, t'$ with $t' < t$

    We have already computed $N[t'] = \sum_{i=1}^{s} \mu_i w_i$ with $\mu_i \in K$ and $w_s < t'$

    $N[t] = \sum_{i=1}^{s} \mu_i \operatorname{Col}(M^{(j)}, i) = \sum_{i=1}^{\deg(l)} \lambda_i w_i$

    for each k such that $t = x_k\, w_l$ for some our un certain l do

      $M_{i,j}^{(k)} := \lambda_i$ for all $i \in \{1, \ldots, n\}$

  else

    there exists $g = t + \sum_{i=1}^{\deg(l)} \lambda_i w_i$ et $\lambda_i \in K \in G$ such that $t = \operatorname{LT}(g)$

    $N[t] := -\sum_{i=1}^{\deg(l)} \lambda_i\, w_i$

    for each k such that $t = x_k\, w_j$ for some j do

      $M_{i,j}^{(k)} := -\lambda_i$ for all $i \in \{1, \ldots, n\}$

return $M^{(k)}$ // matrix multiplication by x_k

Complexity: compute the matrices

Theorem

The previous algorithm computes the matrices $M^{(k)}$ and the complexity is bounded by $O(n \deg(I)^3)$.

Proof.

...

□

Matrix version of FGLM

We need to have a simple linear procedure to detect linear dependency of vectors in the $\mathbb{K}$-vectorial space $\mathbb{K}[x_1, \ldots, x_n]/I$.
The invertible matrix P represents a change of basis between the old basis

$$\mathcal{E}(G) = \left\{ w_1 = 1 < w_2 < \cdots < w_{\deg(I)} \right\}$$

and the new basis S. That is to say that if $S = \left[\varepsilon_1, \ldots, \varepsilon_{\deg(I)} \right]$ then at any step of the algorithm we have:

$$S = P \cdot \mathcal{E}(G)$$

At the beginning $S = [w_1]$, and we compute successively the vectors $v = \varphi(x_k\, w) = M^{(k)} \cdot w$ where v, w are the vectors wrt the basis $\mathcal{E}(G)$:

$$v = \sum_{i=1}^{\deg(I)} v_i\, \overline{w_i}$$

Matrix version of FGLM

At the beginning $S = [w_1]$, and we compute successively the vectors $v = \varphi(x_k\, w) = M^{(k)} \cdot w$ where v, w are the vectors wrt the basis $\mathcal{E}(G)$:

$$v = \sum_{i=1}^{\deg(I)} v_i\, \overline{w_i}$$

To test linear independence we can simply compute:

$$\lambda = P \cdot v = \sum_{i=1}^{\deg(I)} \lambda_i\, \overline{w_i}$$

1. if $\lambda_{\#S+1} = \cdots = \lambda_{\deg(I)} = 0$ then $v \in \mathrm{Vect}_{\mathbb{K}}(S)$.
2. if there exist $k > \#S$ such that $\lambda_k \neq 0$ then $\varepsilon_{\#S+1} := \lambda$ is an independent vector. We compute a matrix P' such that:

$$P' \cdot v = {}^{T}[0, \ldots, 0, 1, 0, \ldots, 0] = \varepsilon_{\#s+1}$$

Updating the Change of Basis Matrix

Algorithm (UPDATE Update the Change of Basis Marix)

Input: $s \in \mathbb{N}$, a vector λ and matrix P
Output: a new matrix P'
$k := \min \{j > s \text{ such that } \lambda_j \neq 0\}$
for j from 1 to $\deg(I)$ do
  $\alpha := \frac{P_{j,k}}{P_{k,k}}, P_{j,k} := P_{s+1,j}, P_{s+1,j} := \alpha$
  if $\alpha \neq 0$ then
    for i from 1 to $\deg(I)$ such that $i \neq s+1$ do
      $P_{i,j} := P_{i,j} - \alpha \lambda_i$
return P

Algorithm Matrix-FGLM

Input: $<$ new monomial ordering, $M^{(k)}$ multiplication matrices

$S := [1]$, $V := [w_1]$, $G := [\,]$, $t := (n, 1)$

$L := [(i, 1), i = 1, \ldots, (n-1)]$ // (i, j) equivalent to $x_i\, S[j]$

$P := I_{\deg(I)}$ change of basis matrix betwen the new basis S and $\mathcal{E}(G)$

infinite loop

$s := \#S$ number of elements in S.

$t = (k, l)$: we compute $v = M^{(k)} \cdot V_l$ and $\lambda = P \cdot v$

if $\lambda_{s+1} = \cdots = \lambda_{\deg(I)} = 0$ then

$$G := G \cup \left[x_k S_l - \sum_{i=1}^{s} \lambda_i \cdot S_i \right]$$

else

$P :=$ UPDATE(s, λ, P)

$S := S \cup [x_k S_l]$ and $V := V \cup [v]$

$L := \mathrm{Sort}(L \cup [(i, s) \mid i = 1, \ldots, n], <)$

Sort L wrt $<_2$ and remove duplicates and multiple of $\mathrm{LT}(G)$

if $L = \varnothing$ then return G

$t := \mathrm{first}(L)$ and removes t from L

Example

$n = 2$ and $G_{<_{\mathrm{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$

$$\mathcal{E}(G) = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\,x_2\}$$

Lexicographical ordering with $x_2 > x_1$.
Multiplication matrices by x_1 and x_2:

$$M^{(1)} = \begin{array}{c|cccc} & x_1\,w_1 & x_1\,w_2 & x_1\,w_3 & x_1\,w_4 \\ w_1 & 0 & -1 & 0 & 3 \\ w_2 & 1 & 1 & 0 & 6 \\ w_3 & 0 & 3 & 0 & -4 \\ w_4 & 0 & 0 & 1 & 1 \end{array}$$

$$M^{(2)} = \begin{array}{c|cccc} & x_2\,w_1 & x_2\,w_2 & x_2\,w_3 & x_2\,w_4 \\ w_1 & 0 & 0 & 1 & -2 \\ w_2 & 0 & 0 & 2 & 3 \\ w_3 & 1 & 0 & -1 & 6 \\ w_4 & 0 & 1 & 0 & -1 \end{array}$$

$L := [(2,1)], S := [1], V := [w_1], G := [\,], t := (1,1)$ represent the monomial $x_1 \cdot S_1 = x_1, P := I_4$

Example

Step 1: Since $t = (1, 1)$, we compute $v := M^{(1)} \cdot V_1 = M^{(1)} \cdot 1 = w_2$ and $\lambda = P \cdot v = w_2 =^T [0, 1, 0, 0]$
Hence $\lambda_2 \neq 0$, $S := [1, x_1]$, $V := [w_1, w_2]$ and the matrix P is left unchanged.
We update L: $L := [(1, 2), (2, 1), (2, 2)]$.

Example

Step 2: Since $t = (1, 2)$, we compute $v := M^{(1)} \cdot V_2 = M^{(1)} \cdot w_2 = {}^T[-1, 1, 3, 0]$
and $\lambda = P \cdot v = {}^T[-1, 1, 3, 0] := V_3$
since $\lambda_3 \neq 0$, $S := [1, x_1, x_1^2]$, $V := [w_1, w_2, {}^T[-1, 1, 3, 0]]$, then

$$P := \begin{bmatrix} 1 & 0 & 1/3 & 0 \\ 0 & 1 & -1/3 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We update $L : L := [(1, 3), (2, 1), (2, 2), (2, 3)]$.

Example

Step 3: $t = (1,3)$. We compute $v := M^{(1)} \cdot V_3 = M^{(1)} \cdot^T [-1,1,3,0] =^T [-1,0,3,3]$
and

$$\lambda = P \cdot v =^T [0,-1,1,3]$$

Since $\lambda_4 \neq 0$, $S := [1, x_1, x_1^2, x_1^3]$, $V := [w_1, w_2, V_3, ^T [-1,0,3,3]]$, and then

$$P := \begin{bmatrix} 1 & 0 & 1/3 & 0 \\ 0 & 1 & -1/3 & 1/3 \\ 0 & 0 & 1/3 & -1/3 \\ 0 & 0 & 0 & 1/3 \end{bmatrix}$$

We update $L := [(1,4),(2,1),(2,2),(2,3),(2,4)]$.

Example

Step 4: $t = (1, 4)$. We compute $v := M^{(1)} \cdot V_4 = M^{(1)} \cdot^T [-1, 0, 3, 3] =^T [9, 17, -12, 6]$
then $\lambda = P \cdot v =^T [5, 23, -6, 2]$
since $\lambda_5 = 0$, we find a polynomial $G := [x_1^4 - 2\,x_1^3 + 6\,x_1^2 - 23\,x_1 - 5]$
and now $L := [(2, 1), (2, 2), (2, 3), (2, 4)]$.

Example

Step 5: $t = (2, 1)$. We compute $v := M^{(2)} \cdot V_1 = M^{(2)} \cdot w_1 = w_3$ and $\lambda = P \cdot w_3 = {}^T\left[\frac{1}{3}, \frac{-1}{3}, \frac{1}{3}, 0\right]$

since $\lambda_5 = 0$, $G := [x_1^4 - 2\,x_1^3 + 6\,x_1^2 - 23\,x_1 - 5, x_2 - \frac{1}{3}x_1^2 + \frac{1}{3}x_1 - \frac{1}{3}]$, by removing multiples of $\mathrm{LT}(G_2) = x_2$, we obtain $L := []$ and the FGLM algorithm stops.

Theorem

The complexity (number of operations in $\mathbb{K}$) of the FGLM and the matrix FGLM algorithms is bounded by $O(n \deg(I)^3)$. Moreover, the result is a Gröbner basis.

Proof.

...

□

Theorem

The complexity (number of operations in $\mathbb{K}$) of the FGLM and the matrix FGLM algorithms is bounded by $O(n \deg(I)^3)$. Moreover, the result is a Gröbner basis.

Proof.

...

□

Remark

Let V be the algebraic variety associated to the ideal $\langle f_1, \ldots, f_m \rangle$ in n variables. When the system has a finite number of solutions:

1. We can compute $P(x_n)$ the smallest polynomial in a lexicographical Gröbner basis as the minimal polynomial P of the matrix $M^{(n)}$.
2. The projection of V on $\mathbb{K}[x_n]$ is thus the eigenvectors of the matrices $M^{(n)}$.

Application of polynomial system solving: Algebraic Crypto

- Evaluate the security of existing cryptosystems.

  - Investigating the security of extensively used cryptographic standards – such as AES, SHA, RSA and new post-quantum crypto systems – against the most powerful attacks is a **permanent concern**.

Application of polynomial system solving: Algebraic Crypto

- Evaluate the security of existing cryptosystems.

  • Investigating the security of extensively used cryptographic standards – such as AES, SHA, RSA and new post-quantum crypto systems – against the most powerful attacks is a **permanent concern**.

  • *Any progress* in the cryptanalysis of such standards could have a huge impact, from a scientific and also economical point of view.

Application of polynomial system solving: Algebraic Crypto

- Evaluate the security of existing cryptosystems.

  • Investigating the security of extensively used cryptographic standards – such as AES, SHA, RSA and new post-quantum crypto systems – against the most powerful attacks is a **permanent concern**.

  • *Any progress* in the cryptanalysis of such standards could have a huge impact, from a scientific and also economical point of view.

  • *General* methods have been proposed: linear cryptanalysis, differential cryptanalysis, . . .

Application of polynomial system solving: Algebraic Crypto

- Evaluate the security of existing cryptosystems.

  • Investigating the security of extensively used cryptographic standards – such as AES, SHA, RSA and new post-quantum crypto systems – against the most powerful attacks is a **permanent concern**.

  • *Any progress* in the cryptanalysis of such standards could have a huge impact, from a scientific and also economical point of view.

  • *General* methods have been proposed: linear cryptanalysis, differential cryptanalysis, . . .

  ☞ describe another *general method* : *Algebraic Cryptanalysis* .

Algebraic cryptanalysis

In this talk $\longrightarrow$ another general method: **Algebraic Cryptanalysis**

Principle

- Model a cryptosystem as a set of algebraic equations
- Try to solve this system (or estimate the difficulty of solving it)

Algebraic Cryptanalysis: model

Very simple idea:

1 Model a cryptosystem as a set of algebraic of equations:

$$\mathcal{S} \begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \cdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

where all the $f_i \in \mathbb{K}[x_1, \ldots, x_n]$ and $\mathbb{K}$ is a finite field (for instance $\mathbb{K} = \mathbb{F}_2$).

Algebraic Cryptanalysis: solving

2 Evaluate the difficulty of **solving** the corresponding algebraic system $\mathcal{S}$.

$$V_{\mathbb{K}} = \left\{ z = (z_1, \ldots, z_n) \in \mathbb{K}^n \mid f_1(z) = \cdots = f_m(z) = 0 \right\}$$

$\longrightarrow$ General Method: new criteria to evaluate the security.

Approach

Difficulties

- Model a cryptosystem as a set of algebraic of equations
  "universal" approach
  (PoSSo is NP-Hard))
  several models are possible !!!
- Solving
  - Minimize the number of variables/degree
  - Maximize the number of equations

Tool

Gröbner Bases computations:
Algorithms + Complexity

Specificity

Solving algebraic systems :

- Huge systems
- Sparse/structured systems
- Try to predict accurately the complexity of computing Gröbner basis for particular instances.

Crypto example: AES

A Zero-Dimensional Gröbner Basis for AES-128
Buchmann, J. and Pyshkin, A. and Weinmann, R.-P.

Abstract. We demonstrate an efficient method for computing a Gröbner basis of a zero-dimensional ideal describing the key-recovery problem from a single plaintext/ciphertext pair for the full AES-128. This Gröbner basis is relative to a degree-lexicographical order. We investigate whether the existence of this Gröbner basis has any security implications for the AES.

Crypto example: AES

A Zero-Dimensional Gröbner Basis for AES-128
Buchmann, J. and Pyshkin, A. and Weinmann, R.-P.

Abstract. We demonstrate an efficient method for computing a Gröbner basis of a zero-dimensional ideal describing the key-recovery problem from a single plaintext/ciphertext pair for the full AES-128. This Gröbner basis is relative to a degree-lexicographical order. We investigate whether the existence of this Gröbner basis has any security implications for the AES.

3.2 The S-Box

The S-Box used in Rijndael can be interpolated as a sparse polynomial over F:

$$\sigma : F \to F, \quad x \mapsto \begin{array}{l} \texttt{05}x^{254} + \texttt{09}x^{253} + \texttt{F9}x^{251} + \texttt{25}x^{247} + \texttt{F4}x^{239} + \\ \texttt{B5}x^{223} + \texttt{B9}x^{191} + \texttt{8F}x^{127} + \texttt{63} \end{array} \tag{2}$$

Crypto example: AES

3.2 The S-Box

The S-Box used in Rijndael can be interpolated as a sparse polynomial over F:

$$\sigma : F \to F, \quad x \mapsto \begin{array}{l} \mathtt{05}x^{254} + \mathtt{09}x^{253} + \mathtt{F9}x^{251} + \mathtt{25}x^{247} + \mathtt{F4}x^{239} + \\ \mathtt{B5}x^{223} + \mathtt{B9}x^{191} + \mathtt{8F}x^{127} + \mathtt{63} \end{array} \tag{2}$$

3.5 Choosing a Suitable Variable Order

The plaintext and ciphertext polynomials simply are of the form

$$x_{i,0} + p_i \qquad p_i \in F, 0 \leq i \leq 15 \tag{13}$$

respectively

$$x_{i,0} + c_i \qquad c_i \in F, 0 \leq i \leq 15. \tag{14}$$

Let $\mathcal{A}$ be the union of the left-hand side of equations (9), (10) and (12) for all rounds $1 \leq j \leq 10$ as well as the plaintext and ciphertext polynomials. Ordering the variables as follows makes all head terms pairwise prime:

1. plaintext variables: $x_{0,0} < \ldots < x_{15,0}$
2. ciphertext variables: $x_{0,10} < \ldots < x_{15,10}$
3. key variables of all rounds in natural order: $k_{0,0} < k_{1,0} < \ldots < k_{15,10}$
4. intermediate state variables in their natural order

Crypto example: AES

3.5 Choosing a Suitable Variable Order

The plaintext and ciphertext polynomials simply are of the form

$$x_{i,0} + p_i \qquad p_i \in F, 0 \leq i \leq 15 \tag{13}$$

respectively

$$x_{i,0} + c_i \qquad c_i \in F, 0 \leq i \leq 15. \tag{14}$$

Let $\mathcal{A}$ be the union of the left-hand side of equations (9), (10) and (12) for all rounds $1 \leq j \leq 10$ as well as the plaintext and ciphertext polynomials. Ordering the variables as follows makes all head terms pairwise prime:

1. plaintext variables: $x_{0,0} < \ldots < x_{15,0}$
2. ciphertext variables: $x_{0,10} < \ldots < x_{15,10}$
3. key variables of all rounds in natural order: $k_{0,0} < k_{1,0} < \ldots < k_{15,10}$
4. intermediate state variables in their natural order

The degree lexicographical term order with the above variable order will be in the following be referred to as $<_{\mathcal{A}}$. By Theorem 1, the set of polynomials $\mathcal{A}$ is a Gröbner basis relative to this term order! Moreover, checking Lemma 1 we verify that this ideal is zero-dimensional.

Crypto example: AES

This result is sufficient to give a bound on the complexity of the Gröbner basis conversion using FGLM. The following theorem is a slightly rephrased version of Theorem 5.1 in [12]:

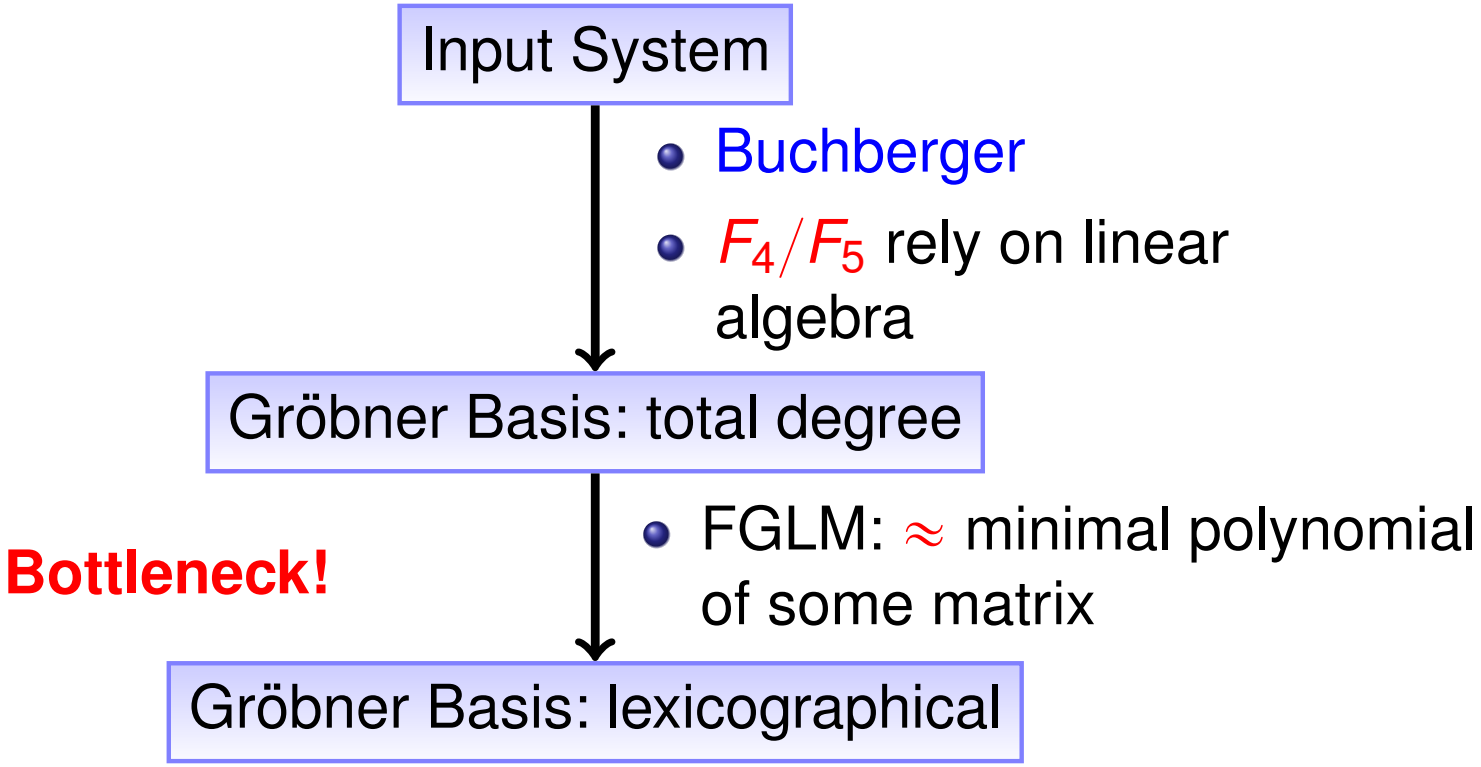
Theorem 2. *Let F be a finite field and $R = F[x_1, \ldots, x_n]$. Furthermore $G_1 \subset R$ is the Gröbner basis relative to a term order $<_1$ of an ideal I, and $D = dim(R/I)$. We can then convert G_1 into a Gröbner basis G_2 relative to a term order $<_2$ in $O(nD^3)$ field operations.*

$$dim(R/\mathcal{A}) = 254^{200} \approx 2^{1598}$$

Fast FGLM: High Performance Algorithm and Implementation

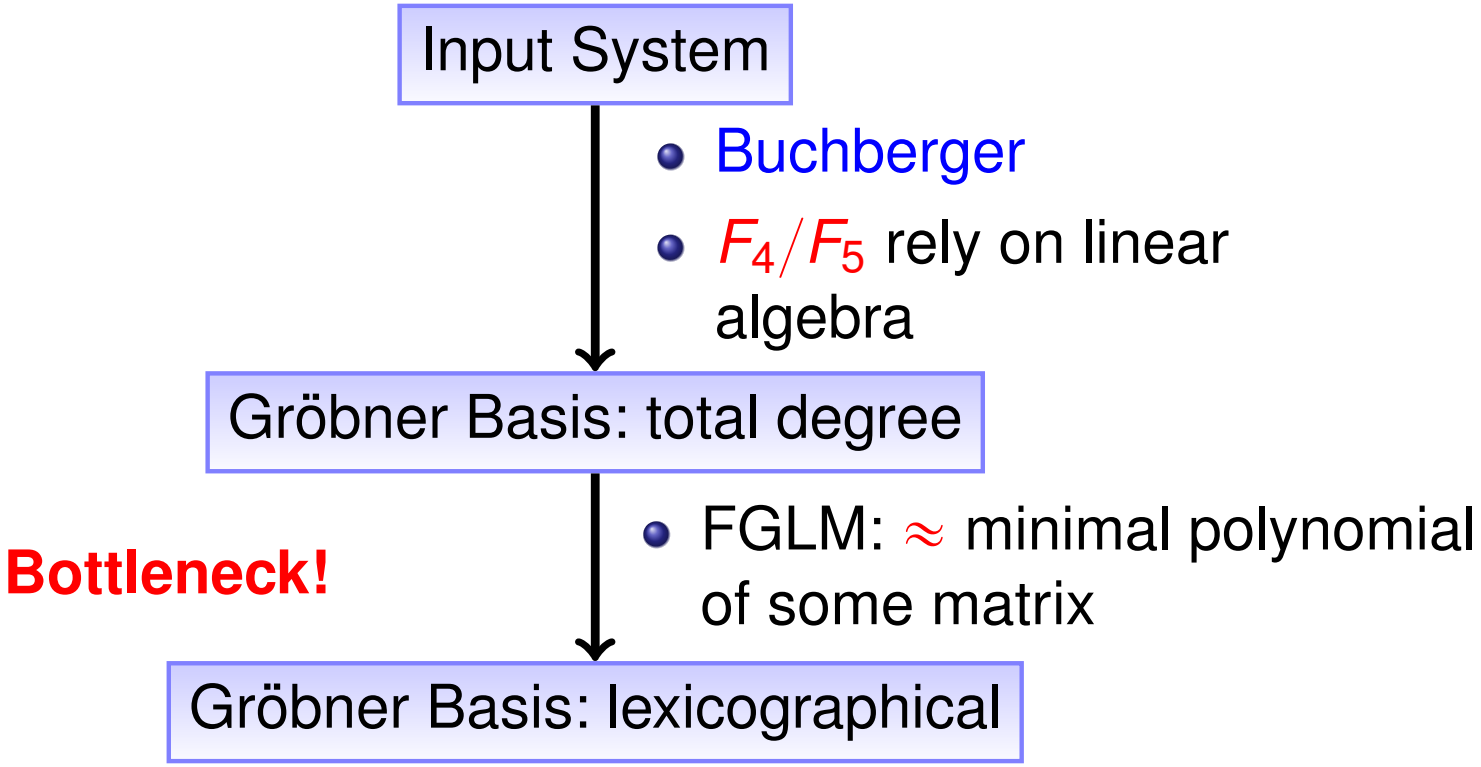
Fast FGLM - Problem

with C. Mou



Fast FGLM - Problem

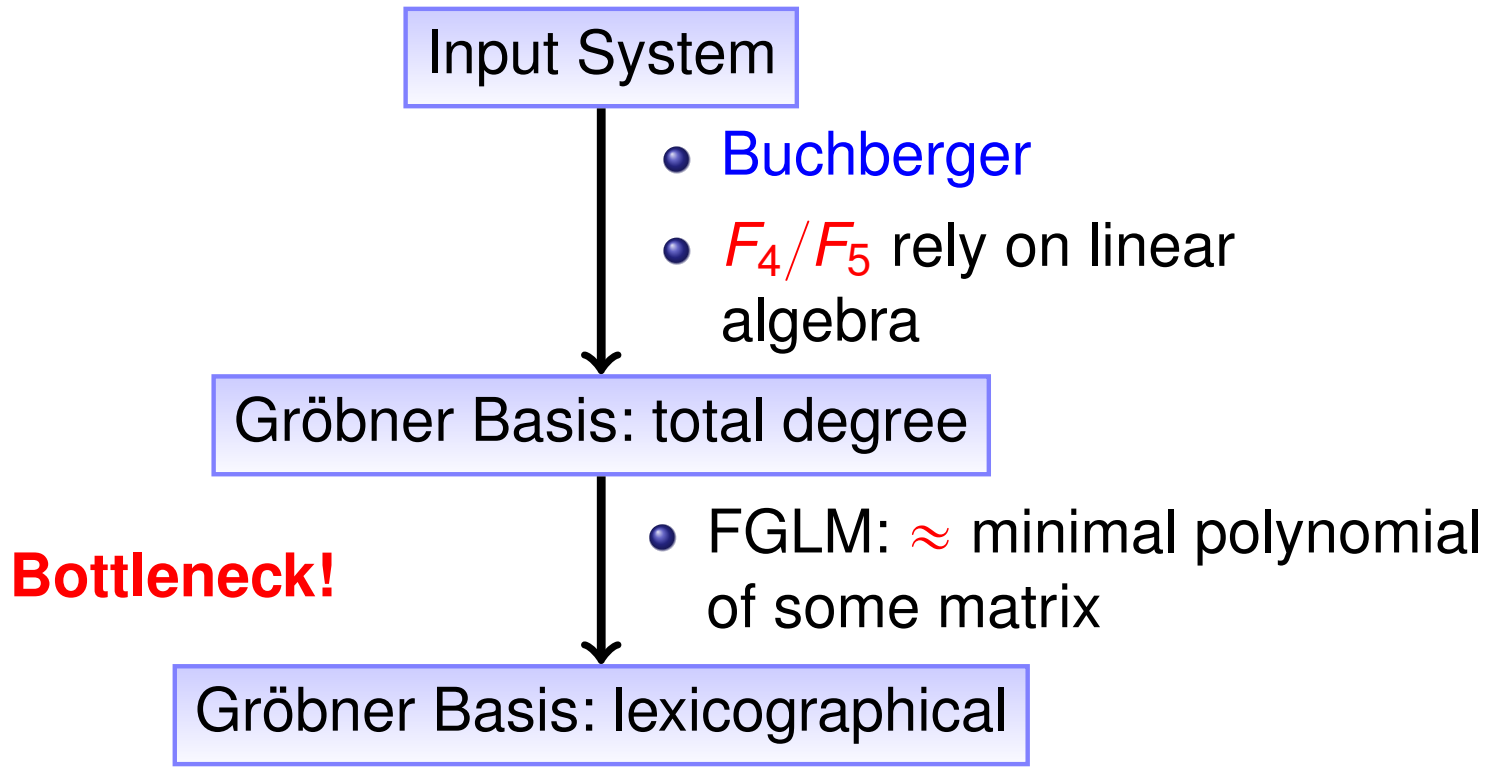
with C. Mou



Magma	MinRank(9,7,4)	MinRank(9,8,5)	Random(14, 2)	Random(15, 2)
D	4116	14112	2^{14}	2^{15}
Step 1	208.1s	3343.5s	7832.4s	74862.9s
Step 2	1360.4s	$>$1 day	84374.6s	$>$15 days

Fast FGLM - Problem

with C. Mou



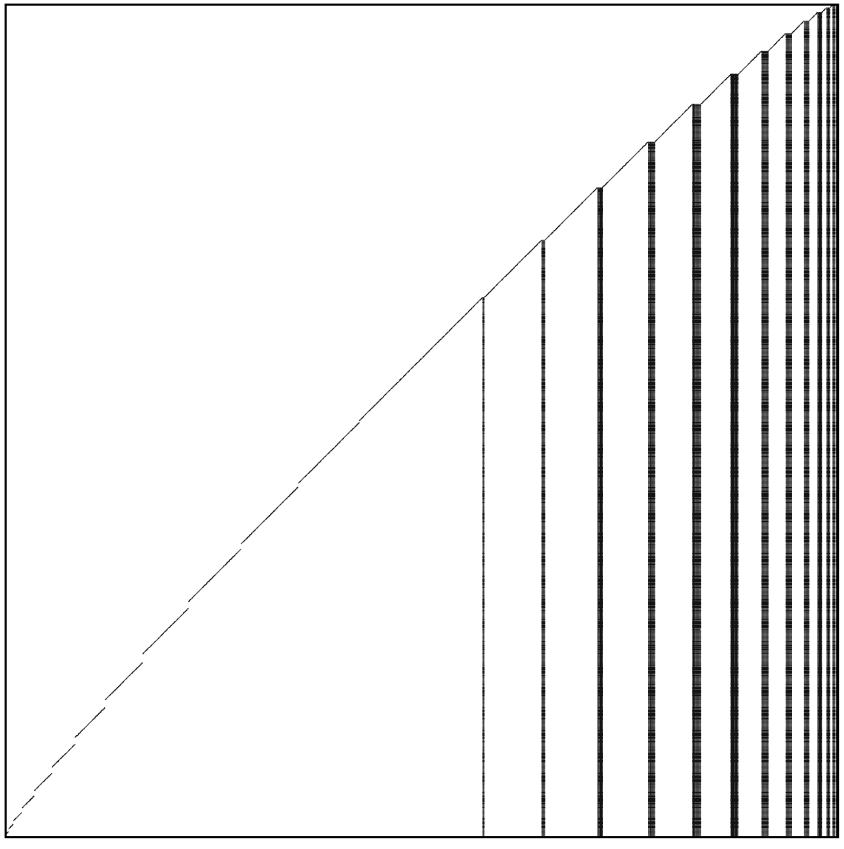
Magma	MinRank(9,7,4)	MinRank(9,8,5)	Random(14, 2)	Random(15, 2
D	4116	14112	2^{14}	2^{15}
Step 1	208.1s	3343.5s	7832.4s	74862.9s
Step 2	1360.4s	$>$1 day	84374.6s	$>$15 days

Goal: a faster algorithm for the change of ordering

Key observation 1

with C. Mou

$T_1, \ldots, T_n$ are sparse, especially T_1.



T_1 for Random(3, 10): 1000×1000, 6.86%

$$T_i \times v = NormalForm(x_i \times v)$$

	DLP Edwards	Cyclic10	MinRank (9,9,6)
D	4096	34940	41580
Sparsity	3.4%	1.0%	16%

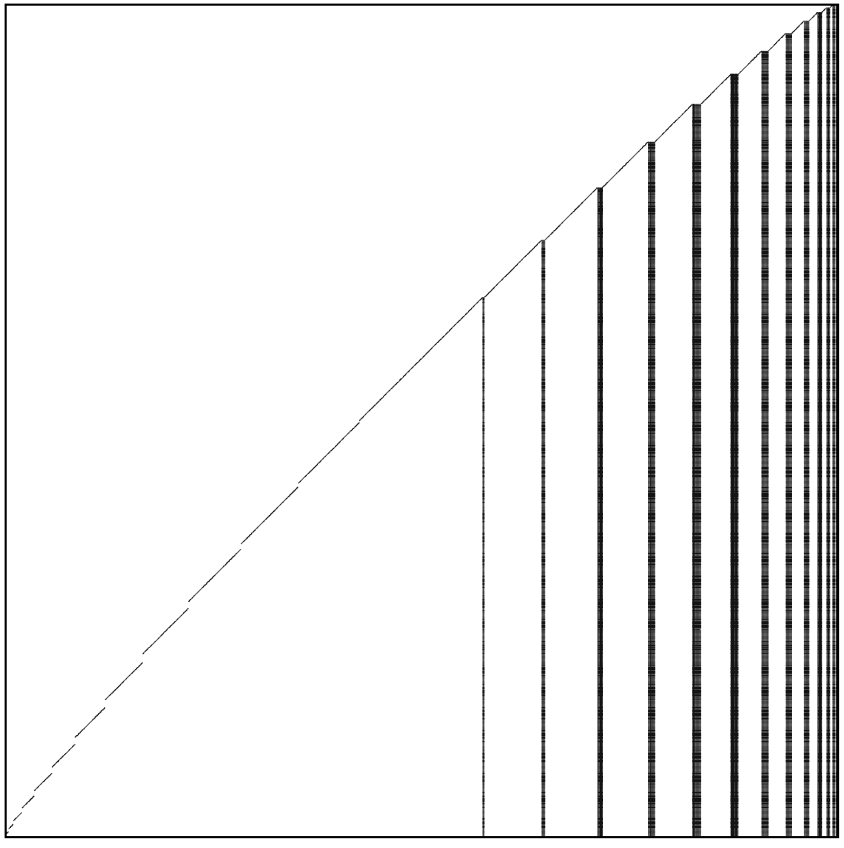
	Random(3, 14)	Random(3, 40)
D	2744	64000
Sparsity	4.2%	1.6%

Key observation 1

with C. Mou

$T_1, \ldots, T_n$ are sparse, especially T_1.

$$T_i \times v = NormalForm(x_i \times v)$$



T_1 for Random(3, 10): 1000×1000, 6.86%

	DLP Edwards	Cyclic10	MinRank (9,9,6)
D	4096	34940	41580
Sparsity	3.4%	1.0%	16%

	Random(3, 14)	Random(3, 40)
D	2744	64000
Sparsity	4.2%	1.6%

Theorem (Faugère,Mou)

n is fixed. For generic polynomial systems of degree d:

$$\%\ \text{of nonzero entries} \underset{d\to\infty}{\sim} \sqrt{\frac{6}{\pi}}\ \frac{1}{d\, n^{\frac{1}{2}}}$$

Key observation 2

The cost of a matrix/vector multiplication $T \times v$ is $\boxed{\#T \ll D^2}$

Any polynomial $\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \boldsymbol{x}^{\boldsymbol{s}}$ in the Gröbner basis is a minimal relation:

$$\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} T_1^{s_1} \cdots T_n^{s_n} \mathbf{1} = 0.$$

Define a n-dimensional mapping $E : \mathbb{Z}_{\geqslant 0}^n \longrightarrow \mathbb{K}$ as

$$\Psi : (s_1, \ldots, s_n) \longmapsto \langle T_1^{s_1} \cdots T_n^{s_n} \mathbf{1}, \boldsymbol{r} \rangle \qquad \boldsymbol{r} \text{ random vector.}$$

Key observation 2

The cost of a matrix/vector multiplication $T \times v$ is $\boxed{\#T \ll D^2}$

Any polynomial $\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \boldsymbol{x}^{\boldsymbol{s}}$ in the Gröbner basis is a minimal relation:

$$\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} T_1^{s_1} \cdots T_n^{s_n} \mathbf{1} = 0.$$

Define a n-dimensional mapping $E : \mathbb{Z}_{\geqslant 0}^n \longrightarrow \mathbb{K}$ as

$$\Psi : (s_1, \ldots, s_n) \longmapsto \langle T_1^{s_1} \cdots T_n^{s_n} \mathbf{1}, \boldsymbol{r} \rangle \qquad \boldsymbol{r} \text{ random vector.}$$

Find **minimal recurrence relation** of Ψ $\rightsquigarrow$ can be found using BMS (Berlekamp-Massey-Sakata from Coding Theory)

Multi-dimensional generalization of Berlekamp–Massey algorithm

[Sakata 1988 & 1990; Saints and Heegard 2002]

Key observation 2

The cost of a matrix/vector multiplication $T \times v$ is $\boxed{\#T \ll D^2}$

Any polynomial $\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \boldsymbol{x}^{\boldsymbol{s}}$ in the Gröbner basis is a minimal relation:

$$\sum_{\boldsymbol{s}} c_{\boldsymbol{s}} T_1^{s_1} \cdots T_n^{s_n} \mathbf{1} = 0.$$

Define a n-dimensional mapping $E : \mathbb{Z}_{\geqslant 0}^n \longrightarrow \mathbb{K}$ as

$$\Psi : (s_1, \ldots, s_n) \longmapsto \langle T_1^{s_1} \cdots T_n^{s_n} \mathbf{1}, \boldsymbol{r} \rangle \qquad \boldsymbol{r} \text{ random vector.}$$

Find **minimal recurrence relation** of Ψ $\rightsquigarrow$ can be found using BMS (Berlekamp-Massey-Sakata from Coding Theory)

Multi-dimensional generalization of Berlekamp–Massey algorithm

[Sakata 1988 & 1990; Saints and Heegard 2002]

Recent results:

- *Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences*, [Berthomieu, F. 2015]
- *A Polynomial-Division-based Algorithm for Computing Linear Recurrence Relations*, [Berthomieu, F. 2018]

Easy case: Shape position case

Assume that I is in shape position:

Shape position *[Becker, Mora, Marinari, and Traverso 1994]*

Ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ is in shape position if its Gröbner basis w.r.t. LEX ($x_1 < \cdots < x_n$) is of the form

$$[f_1(x_1), x_2 - f_2(x_1), \ldots, x_n - f_n(x_1)].$$

Easy case: Shape position case

Shape position *[Becker, Mora, Marinari, and Traverso 1994]*

Ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ is in shape position if its Gröbner basis w.r.t. LEX ($x_1 < \cdots < x_n$) is of the form

$$[f_1(x_1), x_2 - f_2(x_1), \ldots, x_n - f_n(x_1)].$$

Recover f_1: Wiedemann algorithm

Construct $s = \left[\langle \boldsymbol{r}, T_1^i \mathbf{1} \rangle : \ i = 0, \ldots, 2D - 1\right]$, with $\boldsymbol{r}$ a random vector

$\Downarrow$

Compute $\tilde{f}_1$ from s via Berlekamp–Massey algorithm

$\Downarrow$

Check $\deg(\tilde{f}_1) = D$

Easy case: Shape position case

Shape position *[Becker, Mora, Marinari, and Traverso 1994]*

Ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ is in shape position if its Gröbner basis w.r.t. LEX ($x_1 < \cdots < x_n$) is of the form

$$[f_1(x_1), x_2 - f_2(x_1), \ldots, x_n - f_n(x_1)].$$

Recover f_1: Wiedemann algorithm

Construct $s = \left[\langle \boldsymbol{r}, T_1^i \mathbf{1} \rangle : \ i = 0, \ldots, 2D - 1\right]$, with $\boldsymbol{r}$ a random vector

$$\Downarrow$$

Compute $\tilde{f}_1$ from s via Berlekamp–Massey algorithm

$$\Downarrow$$

Check $\deg(\tilde{f}_1) = D \rightsquigarrow$ shape position

Shape position case

Recover $f_2, \ldots, f_n$: constructing linear equations

$$NormalForm(x_i - \sum_{k=0}^{D-1} c_{i,k} x_1^k) = 0$$

Shape position case

Recover $f_2, \ldots, f_n$: constructing linear equations

$$NormalForm(x_i - \sum_{k=0}^{D-1} c_{i,k} x_1^k) = 0$$

$$\Downarrow$$

$$T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^k \mathbf{1}$$

Shape position case

Recover $f_2, \ldots, f_n$: constructing linear equations

$$NormalForm(x_i - \sum_{k=0}^{D-1} c_{i,k} x_1^k) = 0$$

$$\Downarrow$$

$$T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^k \mathbf{1}$$

$$\Downarrow$$

$$T_1^j T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^j T_1^k \mathbf{1}$$

Shape position case

Recover $f_2, \ldots, f_n$: constructing linear equations

$$NormalForm(x_i - \sum_{k=0}^{D-1} c_{i,k} x_1^k) = 0$$

$$\Downarrow$$

$$T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^k \mathbf{1}$$

$$\Downarrow$$

$$T_1^j T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^j T_1^k \mathbf{1}$$

$$\Downarrow$$

$$\langle \boldsymbol{r}, T_1^j T_i \mathbf{1} \rangle = \sum_{k=0}^{D-1} c_{i,k} \cdot \langle \boldsymbol{r}, T_1^{k+j} \mathbf{1} \rangle, \quad j = 0, \ldots, D-1$$

Shape position case

Recover $f_2, \ldots, f_n$: constructing linear equations

$$NormalForm(x_i - \sum_{k=0}^{D-1} c_{i,k} x_1^k) = 0$$

$$\Downarrow$$

$$T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^k \mathbf{1}$$

$$\Downarrow$$

$$T_1^j T_i \mathbf{1} = \sum_{k=0}^{D-1} c_{i,k} \cdot T_1^j T_1^k \mathbf{1}$$

$$\Downarrow$$

$$\langle \boldsymbol{r}, T_1^j T_i \mathbf{1} \rangle = \sum_{k=0}^{D-1} c_{i,k} \cdot \langle \boldsymbol{r}, T_1^{k+j} \mathbf{1} \rangle, \quad j = 0, \ldots, D-1$$

$$\Downarrow$$

$$\langle T'^j \boldsymbol{r}, T_i \mathbf{1} \rangle = \sum_{k=0}^{D-1} c_{i,k} \cdot \langle T'^{k+j} \boldsymbol{r}, \mathbf{1} \rangle, \quad j = 0, \ldots, D-1$$

where $T' = T_1^t$ is the transpose matrix

We compute only **one time** the sequence of vectors

$\boldsymbol{v_0} = \boldsymbol{r}, \boldsymbol{v_1} = T'\boldsymbol{r}, \boldsymbol{v_2} = T'^2\boldsymbol{r}, \ldots, \boldsymbol{v_{2D-1}} = T'^{2D-1}\boldsymbol{r}$ using

$$\boxed{\boldsymbol{v_{i+1}} = T' \times \boldsymbol{v_i}}$$

so that $s = \left[\langle \boldsymbol{r}, T_1^i \mathbf{1} \rangle = \langle T'^i \boldsymbol{r}, \mathbf{1} \rangle = \langle \boldsymbol{v_i}, \mathbf{1} \rangle \ : \ i = 0, \ldots, 2D-1\right]$

Shape position case

Solve: $\boxed{H\,c_i = b}$ with $c_i = {}^t\,[c_{i,0}, \ldots, c_{i,D-1}]$

$$H = \begin{bmatrix} \langle \mathbf{v_0}, \mathbf{1} \rangle & \langle \mathbf{v_1}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle \\ \langle \mathbf{v_1}, \mathbf{1} \rangle & \langle \mathbf{v_2}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_D}, \mathbf{1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle & \langle \mathbf{v_D}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{2D-2}}, \mathbf{1} \rangle \end{bmatrix}$$

Matrix H is a Hankel matrix:

Shape position case

Solve: $\boxed{H\,c_i = b}$ with $c_i = {}^t\,[c_{i,0}, \ldots, c_{i,D-1}]$

$$H = \begin{bmatrix} \langle \mathbf{v_0}, \mathbf{1} \rangle & \langle \mathbf{v_1}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle \\ \langle \mathbf{v_1}, \mathbf{1} \rangle & \langle \mathbf{v_2}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_D}, \mathbf{1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle & \langle \mathbf{v_D}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{2D-2}}, \mathbf{1} \rangle \end{bmatrix}$$

Matrix H is a Hankel matrix:

- Its construction is free
- It is invertible: relationship between linear recurring sequences and Hankel matrices [Jonckheere and Ma 1989]
- Solving efficiently $H\,x = b$: complexity $O(D \log^2(D))$ [Brent, Gustavson, and Yun 1980].

Shape position case

Solve: $\boxed{H\,c_i = b}$ with $c_i = {}^t\,[c_{i,0}, \ldots, c_{i,D-1}]$

$$H = \begin{bmatrix} \langle \mathbf{v_0}, \mathbf{1} \rangle & \langle \mathbf{v_1}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle \\ \langle \mathbf{v_1}, \mathbf{1} \rangle & \langle \mathbf{v_2}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_D}, \mathbf{1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{v_{D-1}}, \mathbf{1} \rangle & \langle \mathbf{v_D}, \mathbf{1} \rangle & \cdots & \langle \mathbf{v_{2D-2}}, \mathbf{1} \rangle \end{bmatrix}$$

Matrix H is a Hankel matrix:

- Its construction is free
- It is invertible: relationship between linear recurring sequences and Hankel matrices [Jonckheere and Ma 1989]
- Solving efficiently $H\,x = b$: complexity $O(D \log^2(D))$ [Brent, Gustavson, and Yun 1980].

Construction of $\langle (T_1^t)^j \mathbf{r}, T_i \mathbf{1} \rangle$ is also free: v is also free.

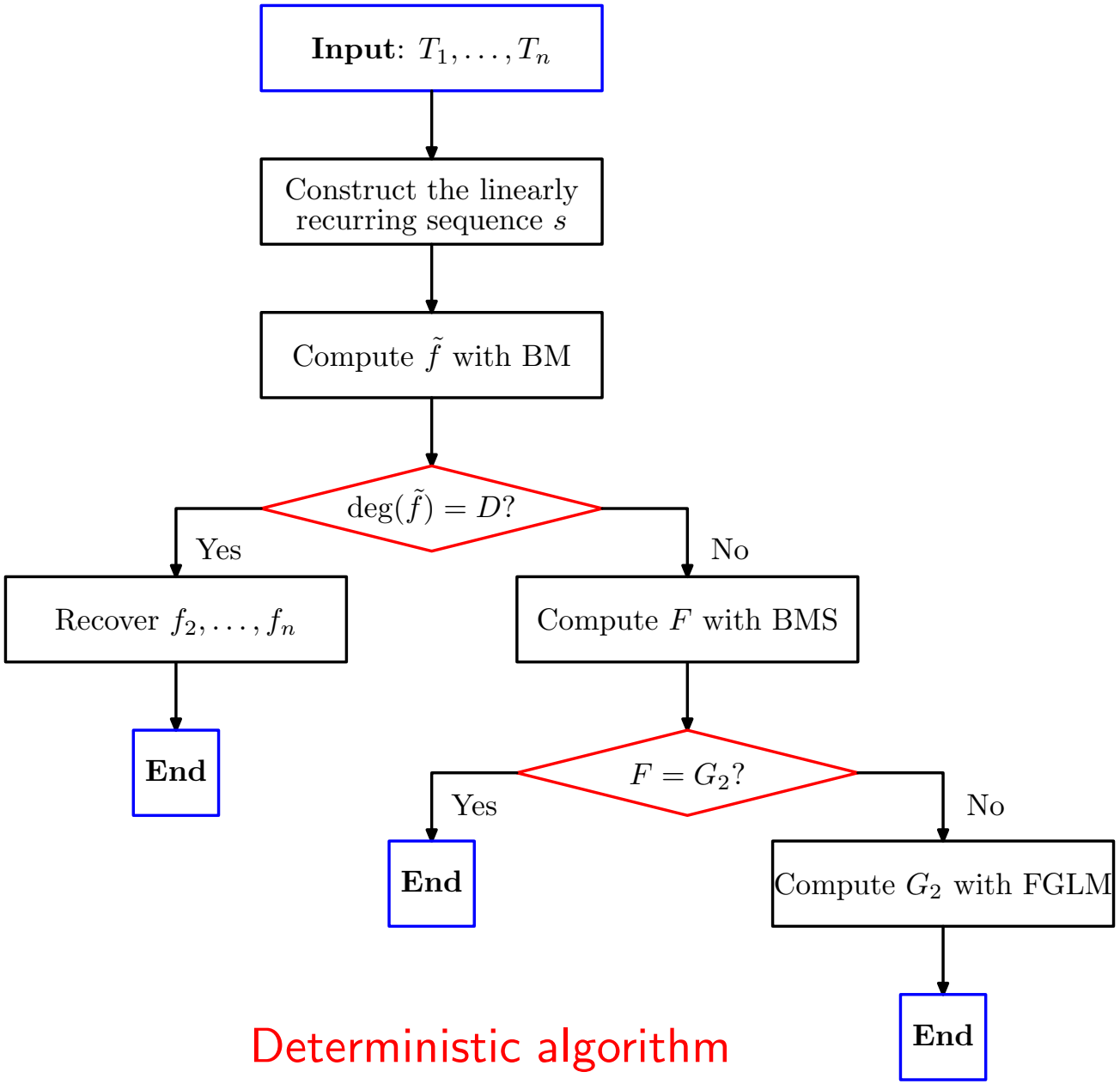
Shape position case

Total complexity for ideals in shape position

$O(D(N_1 + n\log^2(D)))$: $N_1 = \#T' = \#T_1$ the number of nonzero entries in T_1

- compared with $O(nD^3)$ for FGLM
- computing the minimal polynomial of T_1.

General Algorithm



Deterministic algorithm

Efficient Implementation

- Preliminary implementation of the BMS-based method for the general case in Magma.
- Shape position case: first has been implemented in C over fields of characteristic 0 and finite fields.
- We report also a new SSE/multicore implementation.

SSE 4.1 dotproduct fast implementation

16 bits implementation - Intel

Main operation: $\mathbf{y} := T_1\mathbf{x}$

The matrix vector product is equivalent to compute several dot products:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{D} x_i y_i \mod p$$

- lazy reduction: we compute the modulo p only at the end
- using 128 bits registers `xmm`0-15 that is to say 8 16bits words.
- Using SSE instructions we can perform 8 16-bits multiplications simultaneously !
- unrolling the loop we perform **32** multiplications in **one** loop.

Multi-core implementation

Two parallel versions:

- Using Openmp
- Using pthreads

☞ have to rewrite the generation of the matrix T_1 !

Comparing original C-code (Issac 2011) and the new code:

	D	%	Magma	Singular	**New**	**New+SSE**
Katsura 12	4096	21.2%	1408s	2623.5s	18.1s	0.73s

Multi-core implementation

Two parallel versions:

- Using Openmp
- Using pthreads

☞ have to rewrite the generation of the matrix T_1 !

Comparing original C-code (Issac 2011) and the new code:

	D	%	Magma	Singular	**New**	**New+SSE**
Katsura 12	4096	21.2%	1408s	2623.5s	18.1s	0.73s
Random(n=3,d=19)	6859	3.50%	1084s	8248s	15.3s	0.74s

Linear Algebra
Idea of the Algorithms

Solve the following systems:

$$\mathcal{S}_1 \begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

and

$$\mathcal{S}_2 \begin{cases} -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Solve a linear system:

$$\begin{cases} 123 - 7\,e_1 + 22\,e_2 - 94\,e_3 = 0 \\ 11 - 62\,e_2 - 73\,e_3 = 0 \\ -4 - 5\,e_1 + 31\,e_2 + 40\,e_3 = 0 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Solve a linear system:

$$\begin{cases} 123 - 7\,e_1 + 22\,e_2 - 94\,e_3 = 0 \\ 11 - 62\,e_2 - 73\,e_3 = 0 \\ -4 - 5\,e_1 + 31\,e_2 + 40\,e_3 = 0 \end{cases}$$

Recover the solutions:

$$e_1 = 1, e_2 = -1, e_3 = 1$$
$$x = -y = \pm 1$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6 + 3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6 + 3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

$$\begin{bmatrix} 0 & -20 & 0 & -96 & -15 & 0 & 0 & 0 & -59 & 72 \\ 0 & 132 & 0 & 92 & -90 & 0 & 0 & 0 & 43 & -91 \\ 0 & 5 & 0 & 13 & 11 & 0 & 0 & 0 & 12 & -17 \\ 0 & 0 & -20 & 0 & 0 & 0 & 72 & -96 & -15 & -59 \\ 0 & 0 & 132 & 0 & 0 & 0 & -91 & 92 & -90 & 43 \\ 0 & 0 & 5 & 0 & 0 & 0 & -17 & 13 & 11 & 12 \\ -20 & -96 & 0 & -15 & 0 & 72 & 0 & -59 & 0 & 0 \\ 132 & 92 & 0 & -90 & 0 & -91 & 0 & 43 & 0 & 0 \\ 5 & 13 & 0 & 11 & 0 & -17 & 0 & 12 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ e_2 \\ e_3 \\ \vdots \\ e_{10} \end{bmatrix} = 0$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6+3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

$$\begin{bmatrix} 0 & -20 & 0 & -96 & -15 & 0 & 0 & 0 & -59 & 72 \\ 0 & 132 & 0 & 92 & -90 & 0 & 0 & 0 & 43 & -91 \\ 0 & 5 & 0 & 13 & 11 & 0 & 0 & 0 & 12 & -17 \\ 0 & 0 & -20 & 0 & 0 & 0 & 72 & -96 & -15 & -59 \\ 0 & 0 & 132 & 0 & 0 & 0 & -91 & 92 & -90 & 43 \\ 0 & 0 & 5 & 0 & 0 & 0 & -17 & 13 & 11 & 12 \\ -20 & -96 & 0 & -15 & 0 & 72 & 0 & -59 & 0 & 0 \\ 132 & 92 & 0 & -90 & 0 & -91 & 0 & 43 & 0 & 0 \\ 5 & 13 & 0 & 11 & 0 & -17 & 0 & 12 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ e_2 \\ e_3 \\ \vdots \\ e_{10} \end{bmatrix} = 0$$

Solve : $e_2 = -e_3 = e_4 = \cdots = 1$ and recover $x = -y = 1$

Linear Algebra
Idea of the Algorithms

Solve the following systems:

$$\mathcal{S}_1 \begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

and

$$\mathcal{S}_2 \begin{cases} -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Solve a linear system:

$$\begin{cases} 123 - 7\,e_1 + 22\,e_2 - 94\,e_3 = 0 \\ 11 - 62\,e_2 - 73\,e_3 = 0 \\ -4 - 5\,e_1 + 31\,e_2 + 40\,e_3 = 0 \end{cases}$$

Computing Gröbner Bases: example

$$\begin{cases} 123 - 7\,x^2 + 22\,xy - 94\,y^2 = 0 \\ 11 - 62\,xy - 73\,y^2 = 0 \\ -4 - 5\,x^2 + 31\,xy + 40\,y^2 = 0 \end{cases}$$

We linearize the problem:

$$x^2 = e_1, y^2 = e_3, xy = e_2 \text{ (forget that } e_1 e_3 = e_2^2)$$

Solve a linear system:

$$\begin{cases} 123 - 7\,e_1 + 22\,e_2 - 94\,e_3 = 0 \\ 11 - 62\,e_2 - 73\,e_3 = 0 \\ -4 - 5\,e_1 + 31\,e_2 + 40\,e_3 = 0 \end{cases}$$

Recover the solutions:

$$e_1 = 1, e_2 = -1, e_3 = 1$$
$$x = -y = \pm 1$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6 + 3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6 + 3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

$$\begin{bmatrix} 0 & -20 & 0 & -96 & -15 & 0 & 0 & 0 & -59 & 72 \\ 0 & 132 & 0 & 92 & -90 & 0 & 0 & 0 & 43 & -91 \\ 0 & 5 & 0 & 13 & 11 & 0 & 0 & 0 & 12 & -17 \\ 0 & 0 & -20 & 0 & 0 & 0 & 72 & -96 & -15 & -59 \\ 0 & 0 & 132 & 0 & 0 & 0 & -91 & 92 & -90 & 43 \\ 0 & 0 & 5 & 0 & 0 & 0 & -17 & 13 & 11 & 12 \\ -20 & -96 & 0 & -15 & 0 & 72 & 0 & -59 & 0 & 0 \\ 132 & 92 & 0 & -90 & 0 & -91 & 0 & 43 & 0 & 0 \\ 5 & 13 & 0 & 11 & 0 & -17 & 0 & 12 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ e_2 \\ e_3 \\ \vdots \\ e_{10} \end{bmatrix} = 0$$

Computing Gröbner Bases: example (2)

$$\begin{cases} f_1 = -20 - 15\,x^2 - 59\,xy - 96\,x + 72\,y^2 \\ f_2 = 132 - 90\,x^2 + 43\,xy + 92\,x - 91\,y^2 \\ f_3 = 5 + 11\,x^2 + 12\,xy + 13\,x - 17\,y^2 \end{cases}$$

Not enough equations ! Cannot Linearize !

We generate "new" equations: $x\,f_1, x\,f_2, x\,f_3, y\,f_1, y\,f_2, y\,f_3$

We obtain $6+3$ equations and 9 variables:

$$e_2 = x, e_3 = y, e_4 = x^2, e_5 = x^3, e_6 = y^2, e_7 = y^3, e_8 = xy, e_9 = x^2y, e_{10} = y^2x$$

$$\begin{bmatrix} 0 & -20 & 0 & -96 & -15 & 0 & 0 & 0 & -59 & 72 \\ 0 & 132 & 0 & 92 & -90 & 0 & 0 & 0 & 43 & -91 \\ 0 & 5 & 0 & 13 & 11 & 0 & 0 & 0 & 12 & -17 \\ 0 & 0 & -20 & 0 & 0 & 0 & 72 & -96 & -15 & -59 \\ 0 & 0 & 132 & 0 & 0 & 0 & -91 & 92 & -90 & 43 \\ 0 & 0 & 5 & 0 & 0 & 0 & -17 & 13 & 11 & 12 \\ -20 & -96 & 0 & -15 & 0 & 72 & 0 & -59 & 0 & 0 \\ 132 & 92 & 0 & -90 & 0 & -91 & 0 & 43 & 0 & 0 \\ 5 & 13 & 0 & 11 & 0 & -17 & 0 & 12 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ e_2 \\ e_3 \\ \vdots \\ e_{10} \end{bmatrix} = 0$$

Solve : $e_2 = -e_3 = e_4 = \cdots = 1$ and recover $x = -y = 1$

F_4

F_4

Matrix representation of polynomials

Definition

If $F = [f_1, \ldots, f_m]$ is a vector of m polynomials and $<$ an admissible ordering, $\mathrm{T}_<(F) = [t_1, \ldots, t_l]$ the monomials in the support of F sorted for $<$. The matrix representation of $M_{T_<(F)}(F)$ wrt F is:

$$M(F) = \begin{array}{c|ccc} & t_1 & t_2 & t_3 \\ f_1 & \cdots & \cdots & \cdots \\ f_2 & \cdots & \cdots & \cdots \\ f_3 & \cdots & \cdots & \cdots \end{array}$$

$$M(F)_{f_i,t_j} = \mathrm{coeff}\left(f_i, t_j\right)$$

Moreover, $M(F)$ satisfies the following equation:

$$F = M(F) \cdot T_<(F)$$

Polynomial representation of a matrix

Definition

If M is a matrix of size $l \times m$ with coefficients in $\mathbb{K}$ and $X = [t_1, \ldots, t_m]$ is a vector of terms, then the polynomial representation of M wrt X is the vector of l polynomials given by:

$$F = M \cdot X$$

Example (Cyclic 4 Problem)

The monomial ordering is DRL

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

The matrix representation of $F_1 = [f_3, bf_4, df_4]$ is:

$$A_1 = M(F_1) = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ f_3 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ bf_4 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}$$

Example (Cyclic 4 Problem)

The monomial ordering is DRL

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

The matrix representation of $F_1 = [f_3, bf_4, df_4]$ is:

$$A_1 = M(F_1) = \begin{array}{c|ccccccc|} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & 1 & & 1 & \\ bf_4 & 1 & 1 & 1 & & 1 & & \end{array}$$

Macaulay matrix

Definition (Macaulay matrix [8])

Let $F = [f_1, \ldots, f_m]$ a vector of m polynomials and d a non negative integer then the Macaulay matrix in degree d of F $\mathcal{M}_d^{\text{acaulay}}(F)$, is the matrix representation of

$$F^{(d)} = \left[t_j \cdot f_i \mid 1 \leqslant i \leqslant m \text{ and } t_j \in T \text{ with } \deg(t_j) \leqslant d - \deg(f_i) \right]$$

$$\mathcal{M}_d^{\text{acaulay}}(F) = M(F^{(d)}) = \begin{array}{c|ccc} & m_1 & m_2 & m_3 \\ t_1\, f_1 & \cdots & \cdots & \cdots \\ t_2\, f_2 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{array}$$

Echelon form of a matrix

The basis operation is to compute a row echelon form of matrix; this will be the most costly operation.

Definition

If $M(F)$ is the matrix representation of a vector of polynomials F we denote by $\widetilde{M(F)}$ the Gaussian elimination of $M(F)$ (without pivoting the columns of the matrix).

We extend this definition to polynomials:

Definition

Let $F \subset \mathbb{K}[x_1, \ldots, x_n]$ and $<$ a monomial ordering. We denote by $\tilde{F}$ the polynomial representation of $\widetilde{M(F)}$. We say that $\tilde{F}$ is the echelon form of F (or a Gaussiant elimination) wrt $<$.

Example

The matrix representation of $F_1 = [f_3, bf_4, df_4]$ is:

$$A_1 = M(F_1) = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & 1 & & 1 & \\ bf_4 & 1 & 1 & 1 & & 1 & & \end{array}$$

Example

The matrix representation of $F_1 = [f_3, bf_4, df_4]$ is:

$$\tilde{A}_1 = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & & -1 & & -1 \\ bf_4 & & 1 & & & 2 & & 1 \end{array}$$

Example

The polynomial representation of:

$$\tilde{A}_1 = \begin{array}{c} \\ df_4 \\ f_3 \\ bf_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} ab & b^2 & bc & ad & bd & cd & d^2 \end{array} \\ \left| \begin{array}{ccccccc} & & & 1 & 1 & 1 & 1 \\ 1 & & 1 & & -1 & & -1 \\ & 1 & & & 2 & & 1 \end{array} \right| \end{array}$$

$$\tilde{F}_1 = \begin{bmatrix} f_5 = ad + bd + cd + d^2, \\ f_6 = ab + bc - bd - d^2, \\ f_7 = b^2 + 2\,bd + d^2 \end{bmatrix}$$

Macaulay method

The idea of using linear algebra to solve polynomial systems date back to Macaulay.
Macaulay matrix is a generalization of the Sylvester matrix [7] (the matrix involved in the computation of the resultant of 2 polynomials).

The link between the computation of a truncated d-Gröbner basis is given by the following theorem of Lazard:

Theorem (Lazard)

If $F = \{f_1, \ldots, f_m\}$ is a set of homogeneous polynomials then $\mathcal{M}_d^{\widetilde{\text{acaulay}}}(F)$ is a (non reducible) d-Gröbner basis of F.
If $F = \{f_1, \ldots, f_m\}$ is a set of polynomials, then there exists $d > 0$ such that $\mathcal{M}_d^{\widetilde{\text{acaulay}}}(F)$ is a Gröbner basis of F.

Macaulay bound

Theorem (Macaulay bound)

Let $F = \{f_1, \ldots, f_m\}$ is a set of homogeneous polynomials which is a regular sequence. We define

$$D = 1 + \sum_{i=1}^{m} (deg(f_i) - 1)$$

then $\widetilde{\mathcal{M}_D^{\text{acaulay}}(F)}$ is a (non reduced) Gröbner basis of F.

Regular sequence I

We consider the Macaulay matrix of $F = [f_1, \ldots, f_m]$.
If the Macaulay matrix is singular $\longleftrightarrow$ the rows of the matrix are not independent.
Moreover, each row of the matrix is a product $t \times f$ where t is a term and $f \in F$; the linear dependence can be expressed by $\sum_{f \in F, t \in T} \lambda_{t,f}\ t\, f = 0$ or equivalently by grouping terms:

$$\sum_{i=1}^{m} g_i f_i = 0 \tag{1}$$

where g_i are polynomials in $\mathbb{K}[x_1, \ldots, x_n]$. We say that $(g_1, \ldots, g_m)$ is a syzygy. The relation (5) can be rewritten:

$$g_1 f_1 = 0 \text{ modulo } \mathrm{Id}(f_2, \ldots, f_m) \tag{2}$$

in other words it is a **zero divisor** $(\textit{if}\ g_1 \neq 0)$.

Regular sequence II

A linear system is non singular if one cannot find a non zero linear combination:

$$\sum_{i=1}^{m} \lambda_i f_i = 0 \text{ with } \lambda_i \in \mathbb{K} \tag{3}$$

For algebraic systems: it is not possible to avoid non zero relations (5) :

$$f_i f_j - f_j f_i = 0 \tag{4}$$

We say that it is a trivial syzygy.

Regular sequence I

We consider the Macaulay matrix of $F = [f_1, \ldots, f_m]$.
If the Macaulay matrix is singular $\longleftrightarrow$ lthe rows of the matrix are not independent.
Moreover, each row of the matrix is a product $t \times f$ where t is a term and $f \in F$; the linear dependence can be expressed by $\sum_{f \in F, t \in T} \lambda_{t,f}\ t\, f = 0$ or equivalently by grouping terms:

$$\sum_{i=1}^{m} g_i f_i = 0 \tag{5}$$

where g_i are polynomials in $\mathbb{K}[x_1, \ldots, x_n]$. We say that $(g_1, \ldots, g_m)$ is a syzygy. The relation (5) can be rewritten:

$$g_1 f_1 = 0 \text{ modulo } \mathrm{Id}(f_2, \ldots, f_m) \tag{6}$$

in other words it is a **zero divisor** $(\textit{if}\ g_1 \neq 0)$.

Regular sequence II

A linear system is non singular if one cannot find a non zero linear combination:

$$\sum_{i=1}^{m} \lambda_i f_i = 0 \text{ with } \lambda_i \in \mathbb{K} \tag{7}$$

For algebraic systems: it is not possible to avoid non zero relations (5) :

$$f_i f_j - f_j f_i = 0 \tag{8}$$

We say that it is a trivial syzygy.

Regular sequence III

Definition (Regular Sequence

Algebraic definition: the system $(f_1, \ldots, f_m)$ of homogeneous polynomials is **regular** if for all $i = 1, \ldots, m$ and g such that

$$g \cdot f_i \in \langle f_1, \ldots, f_{i-1} \rangle$$

then g is also in $\langle f_1, \ldots, f_{i-1} \rangle$.

Geometric definition: the system$(f_1, \ldots, f_m)$ of homogeneous polynomials is **regular** if for all $i \in \{1, \ldots, m\}$, the dimension of $\langle f_1, \ldots, f_i \rangle$ is $n - i$.
We say that the sequence $(f_1, \ldots, f_m)$ is regular.
The sequence $(f_1, \ldots, f_m)$ of affine polynomials is regular if the sequence $(f_1^h, \ldots, f_m^h)$ is regular (f_i^h is the highest homogeneous par of f_i).

Regular sequence IV

Remark

Another characterization of regular sequences: there is no relation

$$\sum_i g_i \cdot f_i = 0 \text{ with } g_i \in \mathbb{K}[x_1, \ldots, x_n]$$

except the relations induced by the trivial syzygies $f_i f_j = f_j f_i$.

Remark

From the geometric definition: there is no regular sequence when $m > n$.

Characterizations of Gröbner Bases

Characterizations of Gröbner Bases

Useful characterizations of Gröbner bases.

Definition (t-representation)

Let $P = [p_1, \ldots, p_k]$ be a finite subset of $\mathbb{K}[x_1, \ldots, x_n]$, $0 \neq f \in \mathbb{K}[x_1, \ldots, x_n]$, and $t \in T$. Assume that there exists $(g_1, \ldots, g_k) \in \mathbb{K}[x_1, \ldots, x_n]^k$ such that:

$$f = \sum_{i=1}^{k} g_i \, p_i$$

We say that it is a $t-$*representation* of f wrt P if $t \geqslant \mathrm{LT}(g_i p_i)$ for all $1 \leqslant i \leqslant k$. We denote by $f = O_P(t)$ this property.
We note $f = o_P(t)$ when there exists $t' \in T$ such that $t' < t$ and $f = O_P(t')$.

Characterizations of Gröbner Bases

Proposition

If f, g are polynomials and , t is a term, P a finite subset of polynomials, then

$$\begin{array}{llll} f = O_P(t) & g = O_P(t) & \text{implies} & f + g = O_P(t) \\ f = o_P(t) & g = o_P(t) & \text{implies} & f + g = o_P(t) \\ f = O_P(t) & u \in T & \text{implies} & u\ f = O_P(u\,t) \\ f = o_P(t) & u \in T & \text{implies} & u\,f = o_P(u\,t) \end{array}$$

Proposition (R)

If $\textsc{Reduction}(p, P) = 0$ *or* $p \xrightarrow[P]{*} 0$ *then* $p = O_P(\mathrm{LT}(p))$.

Proof.

Easy exercise. □

Characterizations of Gröbner Bases

When $f = O_G(\mathrm{LT}(f))$ we say that f has a standard representation wrg G.

Theorem

G is a Gröbner basis if and only if $\forall 0 \neq f \in \mathrm{Id}(G)$, $f = O_G(\mathrm{LT}(f))$.

Proof.

...

□

and what happen when

$$f \neq O_G(\mathrm{LT}(f)) \ ?$$

Sum of polynomials

$$f \in I = \mathrm{Id}\,(f_1, \cdots, f_m)$$

By definition:

$$f = g_1 f_1 + \cdots + g_m f_m$$

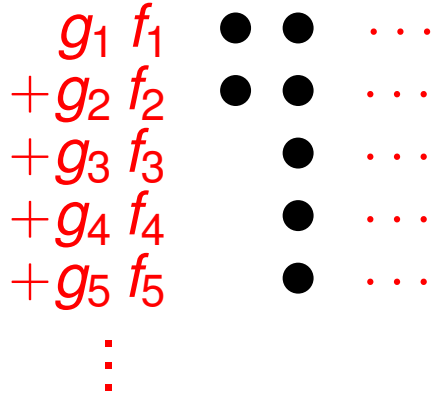
Sum of polynomials

$$f \in I = \mathrm{Id}\,(f_1, \cdots, f_m)$$

By definition:

$$f = g_1 f_1 + \cdots + g_m f_m$$

Not a unique representation !



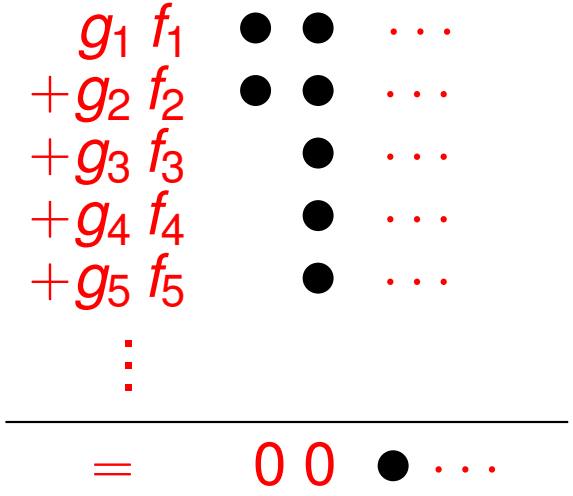
Sum of polynomials

$$f \in I = \mathrm{Id}\,(f_1, \cdots, f_m)$$

By definition:

$$f = g_1 f_1 + \cdots + g_m f_m$$

Not a unique representation !



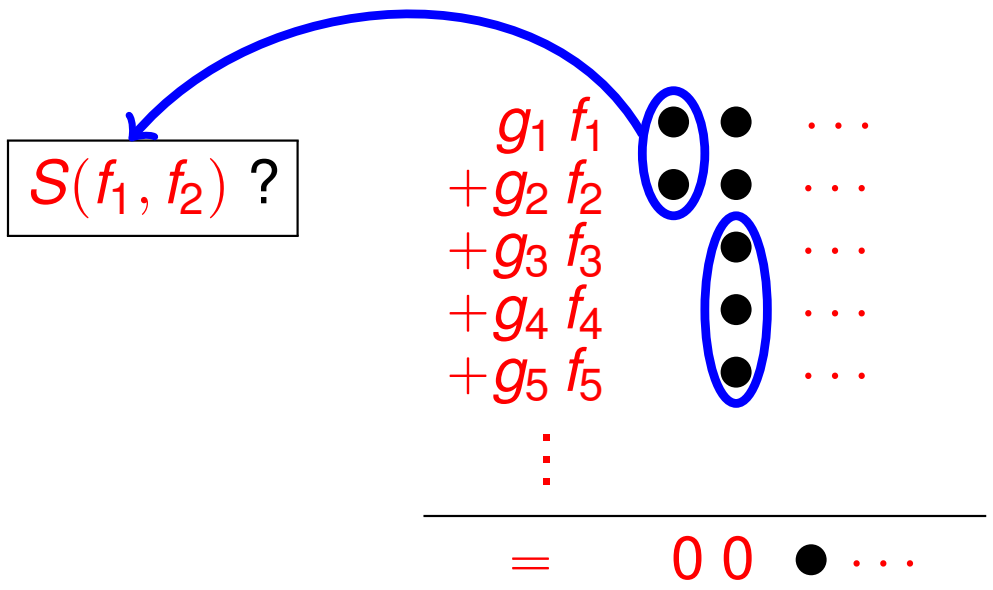
Sum of polynomials

$$f \in I = \mathrm{Id}\,(f_1, \cdots, f_m)$$

By definition:

$$f = g_1 f_1 + \cdots + g_m f_m$$

Not a unique representation !



Characterizations of Gröbner Bases

Theorem

G is a Gröbner basis if and only if $\forall 0 \neq f \in \mathrm{Id}(G)$, $f = O_G(\mathrm{LT}(f))$.

Theorem

Let G be a finite subset of polynomials. If for all g_1, g_2 in G, we have $\mathrm{Spol}(g_1, g_2) = 0$ or $\mathrm{Spol}(g_1, g_2) = o_G(\mathrm{lcm}(g_1, g_2))$, then G is a Gröbner basis.

Proof.

We need to proof a lemma first . . . □

Proof of the theorem: lemma

Lemma

Let $f_1, \ldots, f_k$ *be nonzero polynomials in* $\mathbb{K}[x_1, \ldots, x_n]$ *and* $t \in T$. *Consider* $f = O_P(t) = \Sigma_{i=1}^{k} c_i \mathbf{x}^{\alpha_i} f_i$, *where* $c_i \in \mathbb{K}^*$ *such that*

$$t = \mathbf{x}^{\alpha_1} LT(f_1) = \cdots = \mathbf{x}^{\alpha_k} LT(f_k).$$

If $LT(f) < t$, *then* $k > 1$ *and* f *can be rewritten:*

$$f = \sum_{i=1}^{k-1} b_i \frac{t}{\tau_i} \operatorname{Spol}(f_i, f_{i+1}) \text{ with } b_i \in \mathbb{K}, \tag{9}$$

where $\tau_i = lcm(f_i, f_{i+1})$. *Furthermore*

$$\frac{t}{\tau_i} LT(\operatorname{Spol}(f_i, f_{i+1})) < t, \text{ for all } i = 1, \ldots, k-1.$$

Characterizations of Gröbner Bases

Corollary (Buchberger)

Let G be a finite subset of polynomials . G is a Gröbner basis if and only if $\mathrm{Spol}(f,g) \xrightarrow{*} 0$ for all $(f,g) \in G^2$.

Corollary (Buchberger)

Let G be a finite subset of polynomials . G is a Gröbner basis if and only if REDUCTION$(\mathrm{Spol}(f,g), G) = 0$ *for all $(f,g) \in G^2$.*

Proof.

Let $(f,g) \in G^2$, $f \neq g$. Put $t = \mathrm{LT}(\mathrm{Spol}(f,g)) < \mathrm{lcm}(f,g)$
If REDUCTION$(\mathrm{Spol}(f,g), G) = 0$ then from proposition (R) :
$\mathrm{Spol}(f,g) = O_G(\mathrm{LT}(\mathrm{Spol}(f,g))) = O_G(t) = o_G(\mathrm{lcm}(f,g))$and we can apply the theorem. □

Buchberger Algorithm

Very simple version of the **Buchberger** algorithm:

Algorithm (Buchberger)

Input: $\begin{cases} F = [f_1, \ldots, f_s] \text{ a list of polynomials} \\ < \text{ admissible ordering} \end{cases}$

Output: G a finite subset of $\mathbb{K}[x_1, \ldots, x_n]$.

$G := F$ and $m := s$

$P := \{(f_i, f_j) \mid 1 \leqslant i < j \leqslant m\}$ *the list of critical pairs*

while $P \neq \emptyset$ **do**

  Select and remove from P a critical pair (f, g)

  $f_{m+1} := \mathrm{Spol}(f, g)$

  $f_{m+1} :=$ REDUCTION(f_{m+1}, G)

  if $f_{m+1} \neq 0$ **then**

    $m := m + 1$

    $P := P \cup \{(f_i, f_m) \mid 1 \leqslant i < m\}$

    $G := G \cup \{f_m\}$

return G

F_4

F_4

The F_4 algorithm

Definition

A critical pairs of (f_i, f_j) is a member of $T^2 \times \mathbb{K}[x_1, \ldots, x_n] \times T \times \mathbb{K}[x_1, \ldots, x_n]$,

$$\mathrm{Pair}(f_i, f_j) := (\mathrm{lcm}_{ij}, t_i, f_i, t_j, f_j)$$

such that

$$\mathrm{lcm}(\mathrm{Pair}(f_i, f_j)) = \mathrm{lcm}_{ij} = \mathrm{LT}(t_i f_i) = \mathrm{LT}(t_j f_j) = \mathrm{lcm}(f_i, f_j)$$

Definition

We define the degree of the critical pair $p_{i,j} = \mathrm{Pair}(f_i, f_j)$, $\deg(p_{i,j})$, to be $\deg(lcm_{i,j})$. We define the following operators:

$$\mathrm{Left}(p_{i,j}) := t_i \cdot f_i \;\; et \;\; \mathrm{Right}(p_{i,j}) := t_j \cdot f_j$$

Algorithm F_4 [5] (simplified version)

Input: $\begin{cases} F \text{ is a finite subset of } \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{S}el \text{ is a function } List(Pairs) \to List(Pairs) \\ \qquad \text{such that } \mathcal{S}el(l) \neq \emptyset \text{ if } l \neq \emptyset \end{cases}$

Output: un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$.

$G := F$, $\tilde{F_0^+} := F$, $d := 0$ and $P := \{\mathrm{Pair}(f, g) \mid (f, g) \in G^2 \text{ with } f \neq g\}$

while $P \neq \emptyset$ **do**

$d := d + 1$

$P_d := \mathcal{S}el(P)$

$P := P \backslash P_d$

$L_d := \mathrm{Left}(P_d) \cup \mathrm{Right}(P_d)$

$\tilde{F_d^+} :=$ REDUCTION(L_d, G)

for $h \in \tilde{F_d^+}$ **do**

$P := P \cup \{\mathrm{Pair}(h, g) \mid g \in G\}$

$G := G \cup \{h\}$

return G

We can now extend the definition of reduction of a polynomial modulo a subset of$\mathbb{K}[x_1, \ldots, x_n]$, to the reduction of a subset of $\mathbb{K}[x_1, \ldots, x_n]$ modulo another subset of $\mathbb{K}[x_1, \ldots, x_n]$:

Algorithm REDUCTION

Input: L, G finite subsets of $\mathbb{K}[x_1, \ldots, x_n]$
Output: a finite subset of $\mathbb{K}[x_1, \ldots, x_n]$ (could be empty).
$F :=$SYMBOLICPREPROCESSING(L, G)
$\tilde{F} :=$ Gaussian reduction of F wrt $<$
$\tilde{F}^+ := \left\{ f \in \tilde{F} \mid \mathrm{LT}(f) \notin \mathrm{LT}(F) \right\}$ // *the "useful" part of* $\tilde{F}$
return $\tilde{F}^+$

No arithmetic operation is used: it is a symbolic preprocessing.

Algorithm SymbolicPreprocessing

Input: L, G finite subsets of $\mathbb{K}[x_1, \ldots, x_n]$
Output: a finite subset of $\mathbb{K}[x_1, \ldots, x_n]$
$F := L$
$Done := \mathrm{LT}(F)$
while $\mathrm{T}(F) \neq Done$ **do**
    choose m an element of $\mathrm{T}(F) \backslash Done$
    $Done := Done \cup \{m\}$
    if m top réductible modulo G **then**
        exists $g \in G$ and $m' \in T$ such that $m = m' \cdot \mathrm{LT}(g)$
        $F := F \cup \{m' \cdot g\}$
return F

The SymbolicPreprocessing function is very efficient: its complexity is proportional to the size of the output (if $\#G$ is smaller than the final size of $T(F)$) [parallel implementation].

Lemma (1)

For all polynomials $p \in L_d$,we have $p \xrightarrow{G \cup \tilde{F^+}} 0$

Theorem

The F_4 algorithm computes a Gröbner basis of G in $\mathbb{K}[x_1, \ldots, x_n]$ such that $F \subseteq G$ and $Id(G) = Id(F)$.

Proof.

. . . □

Remark

If $\#Sel(l) = 1$ for all $l \neq \emptyset$ then the F_4 algorithm reduces to the Buchberger algorithm. In this case the function Sel is the equivalent of the selection strategy for the Buchberger algorithm.

Selection function

Algorithm Selection

Input: P a list of critical pairs
Output: a list of critical pairs.
$d := \min\{\deg(\mathrm{lcm}(p)) \mid p \in P\}$
$P_d := \{p \in P \mid \deg(\mathrm{lcm}(p)) = d\}$
return P_d

We call this strategy *the normal strategy for* F_4.
Hence, if the input polynomials are homogeneous, we obtain in degree d, a d Gröbner basis; $\mathcal{S}el$ selects, in the next step, all the critical pairs which are needed to compute the Gröbner basis in degree $d+1$.

Optimisations

- including Buchberger Criteria (or F_5 criterion).
- reuse **all** the rows in the reduced matrices.

Algorithm Buchberger Criteria - Implementation

$(G_{new}, P_{new}) :=$ UPDATE(G_{old}, P_{old}, h)

Input:
- a finite subset G_{old} of $\mathbb{K}[x_1, \ldots, x_n]$
- a finite subset P_{old} of critical pairs in $\mathbb{K}[x_1, \ldots, x_n]$
- $0 \neq h \in \mathbb{K}[x_1, \ldots, x_n]$

Output: a finite subset in $\mathbb{K}[x_1, \ldots, x_n]$ an updated list of critical pairs.

Algorithm F_4 algorithm (with Criteria)

Input: $\begin{cases} F \subset \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{S}el \text{ a function } \mathrm{List}(Pairs) \to \mathrm{List}(Pairs) \end{cases}$

Output: a finite subset of $\mathbb{K}[x_1, \ldots, x_n]$.

$G := \emptyset$ and$P := \emptyset$ and $d := 0$

while $F \neq \emptyset$ **do**

$f := \mathrm{first}(F)$; $F := F \backslash \{f\}$

$(G, P) :=$UPDATE(G, P, f)

while $P \neq \emptyset$ **do**

$d := d + 1$

$P_d := \mathcal{S}el(P)$; $P := P \backslash P_d$

$L_d := \mathrm{Left}(P_d) \cup \mathrm{Right}(P_d)$

$(\tilde{F_d^+}, F_d) :=$REDUCTION$(L_d, G, (F_i)_{d=1,\ldots,(d-1)})$

for $h \in \tilde{F_d^+}$ **do**

$P := P \cup \{\mathrm{Pair}(h, g) \mid g \in G\}$

$(G, P) :=$UPDATE(G, P, h)

return G

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{l} f_1 = {x_1}^2 + 66\,x_1\,x_2 + 4\,x_1\,x_3 + 25\,{x_2}^2 + 41\,x_2\,x_3 + 54\,{x_3}^2 + 42\,x_1 \\ \quad +87\,x_2 + 22\,x_3 + 86, \\ f_2 = {x_1}^2 + 22\,x_1\,x_2 + 38\,x_1\,x_3 + 9\,{x_2}^2 + 53\,x_2\,x_3 + 6\,{x_3}^2 + 92\,x_1 \\ \quad +61\,x_2 + 74\,x_3 + 49, \\ f_3 = {x_1}^2 + 13\,x_1\,x_2 + 86\,x_1\,x_3 + 29\,{x_2}^2 + 11\,x_2\,x_3 + 81\,{x_3}^2 + 98\,x_1 \\ \quad +67\,x_2 + 7\,x_3 + 40 \end{array} \right.$$

At the beginning $G = \{f_1\}$ and $P_1 = \{\mathrm{Pair}(f_2, f_3), \mathrm{Pair}(f_1, f_2)\}$ such that $L_1 = \{(1, f_3), (1, f_2), (1, f_1)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, f_2, f_1\}$ $\quad T(F_1) = \{\boxed{x_1^2}, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1, x_2, x_3, 1\}$

$\boxed{x_1^2}$ is already done. All the other monomials are not reducible.

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

At the beginning $G = \{f_1\}$ and $P_2 = \{\mathrm{Pair}(f_2, f_3), \mathrm{Pair}(f_1, f_2)\}$ such that $L_2 = \{(1, f_3), (1, f_2), (1, f_1)\}$.

SymbolicPreprocessing$(L_2, G, \varnothing)$:

$$F_2 = \{f_3, f_2, f_1\} \quad T(F_2) = \{\boxed{x_1^2}, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1, x_2, x_3, 1\}$$

$\boxed{x_1^2}$ is already done. All the other monomials are not reducible.

Matrix representation of $F_1 = [f_3, f_2, f_1]$ is:

$$A_1 = M(F_1) = \begin{array}{c|cccccc|c} & x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 & \cdots \\ f_3 & 1 & 13 & 29 & 86 & 11 & 81 & \cdots \\ f_2 & 1 & 22 & 26 & 38 & 53 & 6 & \cdots \\ f_1 & 1 & 66 & 25 & 4 & 41 & 54 & \cdots \end{array}$$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

At the beginning $G = \{f_1\}$ and $P_2 = \{\mathrm{Pair}(f_2, f_3), \mathrm{Pair}(f_1, f_2)\}$ such that $L_2 = \{(1, f_3), (1, f_2), (1, f_1)\}$.

SymbolicPreprocessing$(L_2, G, \varnothing)$:

$$F_2 = \{f_3, f_2, f_1\} \quad T(F_2) = \{\boxed{x_1^2}, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1, x_2, x_3, 1\}$$

$\boxed{x_1^2}$ is already done. All the other monomials are not reducible.

$$\widetilde{A_2} = \begin{array}{c|cccccc|c} & x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 & \cdots \\ f_6 & 0 & 0 & 1 & 28 & 19 & 79 & \cdots \\ f_5 & 0 & 1 & 0 & 12 & 2 & 5 & \cdots \\ f_1 & 1 & 66 & 4 & 25 & 41 & 54 & \cdots \end{array}$$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

At the beginning $G = \{f_1\}$ and $P_2 = \{\mathrm{Pair}(f_2, f_3), \mathrm{Pair}(f_1, f_2)\}$ such that $L_2 = \{(1, f_3), (1, f_2), (1, f_1)\}$.

SymbolicPreprocessing$(L_2, G, \varnothing)$:

$$F_2 = \{f_3, f_2, f_1\} \quad T(F_2) = \{\boxed{x_1^2}, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1, x_2, x_3, 1\}$$

$\boxed{x_1^2}$ is already done. All the other monomials are not reducible.

$$\widetilde{A_2} = \begin{array}{c|cccccc|c} & x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 & \cdots \\ f_6 & 0 & 0 & 1 & 28 & 19 & 79 & \cdots \\ f_5 & 0 & 1 & 0 & 12 & 2 & 5 & \cdots \\ f_1 & 1 & 66 & 4 & 25 & 41 & 54 & \cdots \end{array}$$

Polynomial representation of $\widetilde{A_2}$:

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

At the beginning $G = \{f_1\}$ and $P_2 = \{\mathrm{Pair}(f_2, f_3), \mathrm{Pair}(f_1, f_2)\}$ such that $L_2 = \{(1, f_3), (1, f_2), (1, f_1)\}$.

SymbolicPreprocessing$(L_2, G, \varnothing)$:

$F_2 = \{f_3, f_2, f_1\} \quad T(F_2) = \{\boxed{x_1^2}, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1, x_2, x_3, 1\}$

$\boxed{x_1^2}$ is already done. All the other monomials are not reducible.

$$\widetilde{A_2} = \begin{array}{c|cccccc|c} & x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 & \cdots \\ f_6 & 0 & 0 & 1 & 28 & 19 & 79 & \cdots \\ f_5 & 0 & 1 & 0 & 12 & 2 & 5 & \cdots \\ f_1 & 1 & 66 & 4 & 25 & 41 & 54 & \cdots \end{array}$$

Polynomial representation of $\widetilde{A_2}$:

$f_5 = x_1\,x_2 + 12\,x_1\,x_3 + 2\,x_2\,x_3 + 55\,{x_3}^2 + 66\,x_1 + 88\,x_2 + 60\,x_3 + 92,$
$f_6 = {x_2}^2 + 28\,x_1\,x_3 + 19\,x_2\,x_3 + 79\,{x_3}^2 + 30\,x_1 + 50\,x_2 + 59\,x_3 + 46$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

In degree 3: $P_3 = \{\mathrm{Pair}(f_1, f_5), \mathrm{Pair}(f_5, f_6)\}$ such that
$L_3 = \{(x_2, f_1), (x_1, f_5), (x_2, f_5), (x_1, f_6)\}$.

SymbolicPreprocessing$(L_3, G, \varnothing)$:
$F_3 = \{x_2 f_1, x_1 f_5, x_2 f_5, x_1 f_6\}$
$T(F_3) = \{\boxed{x_1^2 x_2}, \boxed{x_1 x_2^2}, x_2^3, x_1 x_2 x_3, x_1 x_3^2, x_1 x_3, \ldots\}$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

In degree 3: $P_3 = \{\mathrm{Pair}(f_1, f_5), \mathrm{Pair}(f_5, f_6)\}$ such that
$L_3 = \{(x_2, f_1), (x_1, f_5), (x_2, f_5), (x_1, f_6)\}$.

SymbolicPreprocessing$(L_3, G, \varnothing)$:
$F_3 = \{x_2 f_1, x_1 f_5, x_2 f_5, x_1 f_6\}$
$T(F_3) = \{\boxed{x_1^2 x_2}, \boxed{x_1 x_2^2}, x_2^3, x_1 x_2 x_3, x_1 x_3^2, x_1 x_3, \ldots\}$
x_2^3 is divisible by $x_2 f_6 \longrightarrow F_3 = F_3 \cup \{x_2 f_6\}$
$x_1 x_2 x_3$ is divisible by $x_3 f_5 \longrightarrow F_3 = F_3 \cup \{x_3 f_5\}$
$\ldots$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

In degree 3: $P_3 = \{\mathrm{Pair}(f_1, f_5), \mathrm{Pair}(f_5, f_6)\}$ such that
$L_3 = \{(x_2, f_1), (x_1, f_5), (x_2, f_5), (x_1, f_6)\}$.

SymbolicPreprocessing$(L_3, G, \varnothing)$:
$F_3 = \{x_2 f_1, x_1 f_5, x_2 f_5, x_1 f_6\}$
$T(F_3) = \{\boxed{x_1^2 x_2}, \boxed{x_1 x_2^2}, x_2^3, x_1 x_2 x_3, x_1 x_3^2, x_1 x_3, \ldots\}$
x_2^3 is divisible by $x_2 f_6 \longrightarrow F_3 = F_3 \cup \{x_2 f_6\}$
$x_1 x_2 x_3$ is divisible by $x_3 f_5 \longrightarrow F_3 = F_3 \cup \{x_3 f_5\}$
$\ldots$
$F_3 = [x_2\, f_1, x_1\, f_5, x_2\, f_5, x_1\, f_6, x_2\, f_6, x_3\, f_5, x_3\, f_6, f_5, f_6, x_3\, f_1, f_1]$

F4: step by step

Example (3 quadratic equation in $\mathbb{F}_{101}$)

x_2^3 is divisible by $x_2 f_6 \longrightarrow F_3 = F_3 \cup \{x_2 f_6\}$
$x_1 x_2 x_3$ is divisible by $x_3 f_5 \longrightarrow F_3 = F_3 \cup \{x_3 f_5\}$
...

$$F_3 = [x_2\, f_1, x_1\, f_5, x_2\, f_5, x_1\, f_6, x_2\, f_6, x_3\, f_5, x_3\, f_6, f_5, f_6, x_3\, f_1, f_1]$$

$$A_3 = \begin{array}{r|cccccccccccccccc}
f_6 & & & & & & & & & & & 1 & 28 & 19 & 79 & 30 & 50 \\
f_5 & & & & & & & & & & 1 & 0 & 12 & 2 & 55 & 66 & 88 \\
. & & & & & & & & & 1 & 66 & 25 & 4 & 41 & 54 & 42 & 87 \\
. & & & & & 1 & 28 & 19 & 79 & 0 & 0 & 0 & 30 & 50 & 59 & 0 & 0 \\
. & & & & 1 & 0 & 12 & 2 & 55 & 0 & 0 & 0 & 66 & 88 & 60 & 0 & 0 \\
. & & & 1 & 66 & 25 & 4 & 41 & 54 & 0 & 0 & 0 & 42 & 87 & 22 & 0 & 0 \\
. & & 1 & 0 & 28 & 19 & 0 & 79 & 0 & 0 & 30 & 50 & 0 & 59 & 0 & 0 & 46 \\
x_2 f_5 & 1 & 0 & 0 & 12 & 2 & 0 & 55 & 0 & 0 & 66 & 88 & 0 & 60 & 0 & 0 & 92 \\
f_{10} & 1 & 0 & 28 & 19 & 0 & 79 & 0 & 0 & 30 & 50 & 0 & 59 & 0 & 0 & 46 & 0 \\
x_2 f_1 & 1 & 66 & 25 & 0 & 4 & 41 & 0 & 54 & 0 & 0 & 42 & 87 & 0 & 22 & 0 & 0 & 86 \\
f_8 & 1 & 0 & 0 & 12 & 2 & 0 & 55 & 0 & 0 & 66 & 88 & 0 & 60 & 0 & 0 & 92 & 0
\end{array}$$

$$f_{10} = x_1\, {x_3}^2 + 23\, {x_3}^3 + 77\, x_1\, x_3 + 66\, x_2\, x_3 + 84\, {x_3}^2 + 48\, x_1 + 38\, x_2 + 44\, x_3 + 68$$
$$f_8 = x_2\, {x_3}^2 + 98\, {x_3}^3 + 60\, x_1\, x_3 + 34\, x_2\, x_3 + 85\, {x_3}^2 + 65\, x_1 + 9\, x_2 + 74\, x_3 + 28$$

F4: step by step

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \emptyset)$:

$F_1 = \{f_3, b\, f_4\}$ $\quad T(F_1) = \{\boxed{ab}, ad, b^2, bc, bd, cd\}$

$\boxed{ab}$ is already done.

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, b\, f_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, b^2, bc, bd, cd\}$

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, b\, f_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, b^2, bc, bd, cd\}$

ad is top reducible by $f_4 \in G$!

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, b\,f_4, df_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, b^2, bc, bd, cd, d^2\}$

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, b\,f_4, df_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, \boxed{b^2}, bc, bd, cd, d^2\}$

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \emptyset)$:

$F_1 = \{f_3, b\,f_4, df_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, \boxed{b^2}, bc, bd, cd, d^2\}$

b^2 is not reducible by G

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing$(L_1, G, \varnothing)$:

$F_1 = \{f_3, b\,f_4, df_4\}$ $\quad T(F_1) = \{\boxed{ab}, \boxed{ad}, \boxed{b^2}, \boxed{bc}, \boxed{bd}, \boxed{cd}, \boxed{d^2}\}$

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

$$F = \left[\begin{array}{c} f_1 = abcd - 1, f_2 = abc + abd + acd + bcd, \\ f_3 = ab + bc + ad + cd, f_4 = a + b + c + d \end{array} \right]$$

At the beginning $G = \{f_4\}$ and $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ such that $L_1 = \{(1, f_3), (b, f_4)\}$.

SymbolicPreprocessing $(L_1, G, \varnothing)$ returns

$$F_1 = [f_3, bf_4, df_4].$$

Example (Cyclic 4)

Matrix representation of $F_1 = [f_3, bf_4, df_4]$ is:

$$A_1 = M(F_1) = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & 1 & & 1 & \\ bf_4 & 1 & 1 & 1 & & 1 & & \end{array}$$

Example (Cyclic 4)

Gaussian reduction of A_1 is:

$$\widetilde{A_1} = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & & -1 & & -1 \\ bf_4 & & 1 & & & 2 & & 1 \end{array}$$

Example (Cyclic 4)

$$\widetilde{A_1} = \begin{array}{c|ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & & & & 1 & 1 & 1 & 1 \\ f_3 & 1 & & 1 & & -1 & & -1 \\ bf_4 & & 1 & & & 2 & & 1 \end{array}$$

$$\tilde{F}_1 = \begin{bmatrix} f_5 = ad + bd + cd + d^2, \\ f_6 = ab + bc - bd - d^2, \\ f_7 = b^2 + 2\,bd + d^2 \end{bmatrix}$$

Example (Cyclic 4)

$\tilde{F_1} = [f_5 = ad + bd + cd + d^2,$
$f_6 = ab + bc - bd - d^2,$
$f_7 = b^2 + 2\,bd + d^2]$

and since $ab, ad \in \mathrm{LT}(F_1)$ we have
$\tilde{F_1}_+ = [f_7]$
and now $G = \{f_4, f_7\}$.

Example (Cyclic 4)

For the next step we have to consider $P_2 = \{\mathrm{Pair}(f_2, f_4)\}$
hence $L_2 = \{(1, f_2), (bc, f_4)\}$ and $\mathcal{F} = \{F_1\}$.

Example (Cyclic 4)

$L_2 = \{(1, f_2), (bc, f_4)\}$ et $\mathcal{F} = \{F_1\}$.
In SymbolicPreprocessing we can try to simplify the products $1 \cdot f_2$ and $bc \cdot f_4$ using the previous computations:

For instance $\mathrm{LT}(bc\, f_4) = abc = \mathrm{LT}(c\, f_6)$ and so instead of $bc \cdot f_4$ we can consider $c \cdot f_6$.

Example (Cyclic 4)

For the next step we have to consider $P_2 = \{\mathrm{Pair}(f_2, f_4)\}$
hence $L_2 = \{(1, f_2), (bc, f_4)\}$ and $\mathcal{F} = \{F_1\}$.
SymbolicPreprocessing

$F_2 = \{f_2, c\,f_6\}$ $\quad T(F_2) = \{\boxed{abc}, bc^2, abd, acd, bcd, cd^2\}$

Example (Cyclic 4)

For the next step we have to consider $P_2 = \{\mathrm{Pair}(f_2, f_4)\}$
hence $L_2 = \{(1, f_2), (bc, f_4)\}$ and $\mathcal{F} = \{F_1\}$.
SymbolicPreprocessing

$F_2 = \{f_2, cf_6\}$ $\quad T(F_2) = \{\boxed{abc}, bc^2, \boxed{abd}, acd, bcd, cd^2\}$

Example (Cyclic 4)

$\tilde{F_1} = [f_5 = ad + bd + cd + d^2, f_6 = ab + bc - bd - d^2, f_7 = b^2 + 2\,bd + d^2]$

For the next step we have to consider $P_2 = \{\mathrm{Pair}(f_2, f_4)\}$
hence $L_2 = \{(1, f_2), (bc, f_4)\}$ and $\mathcal{F} = \{F_1\}$.
SymbolicPreprocessing

$F_2 = \{f_2, cf_6\} \quad T(F_2) = \{\boxed{abc}, bc^2, \boxed{abd}, acd, bcd, cd^2\}$

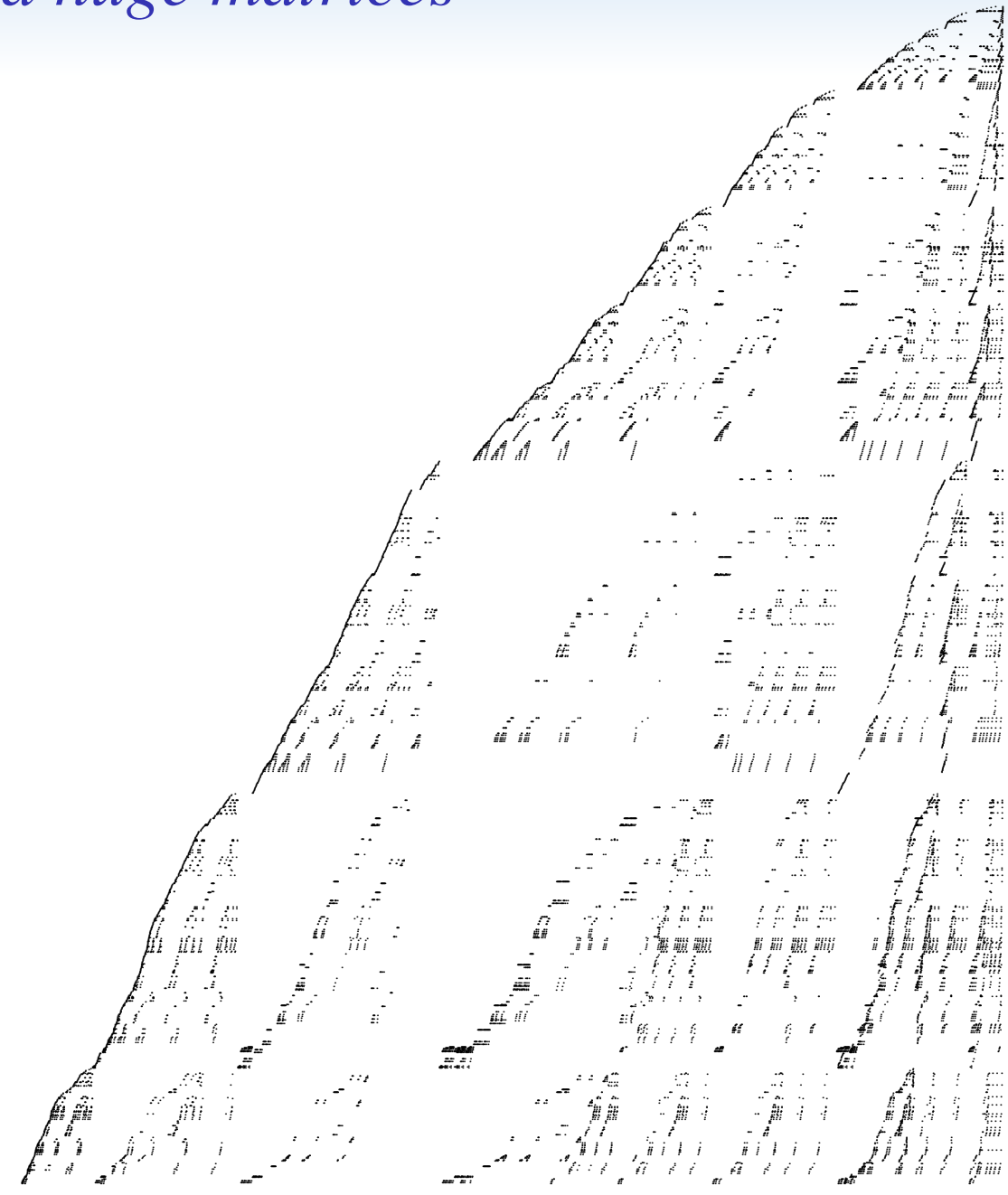
abd is reducible by $bd\,f_4$ but also by $b\,f_5$!

Optimisations

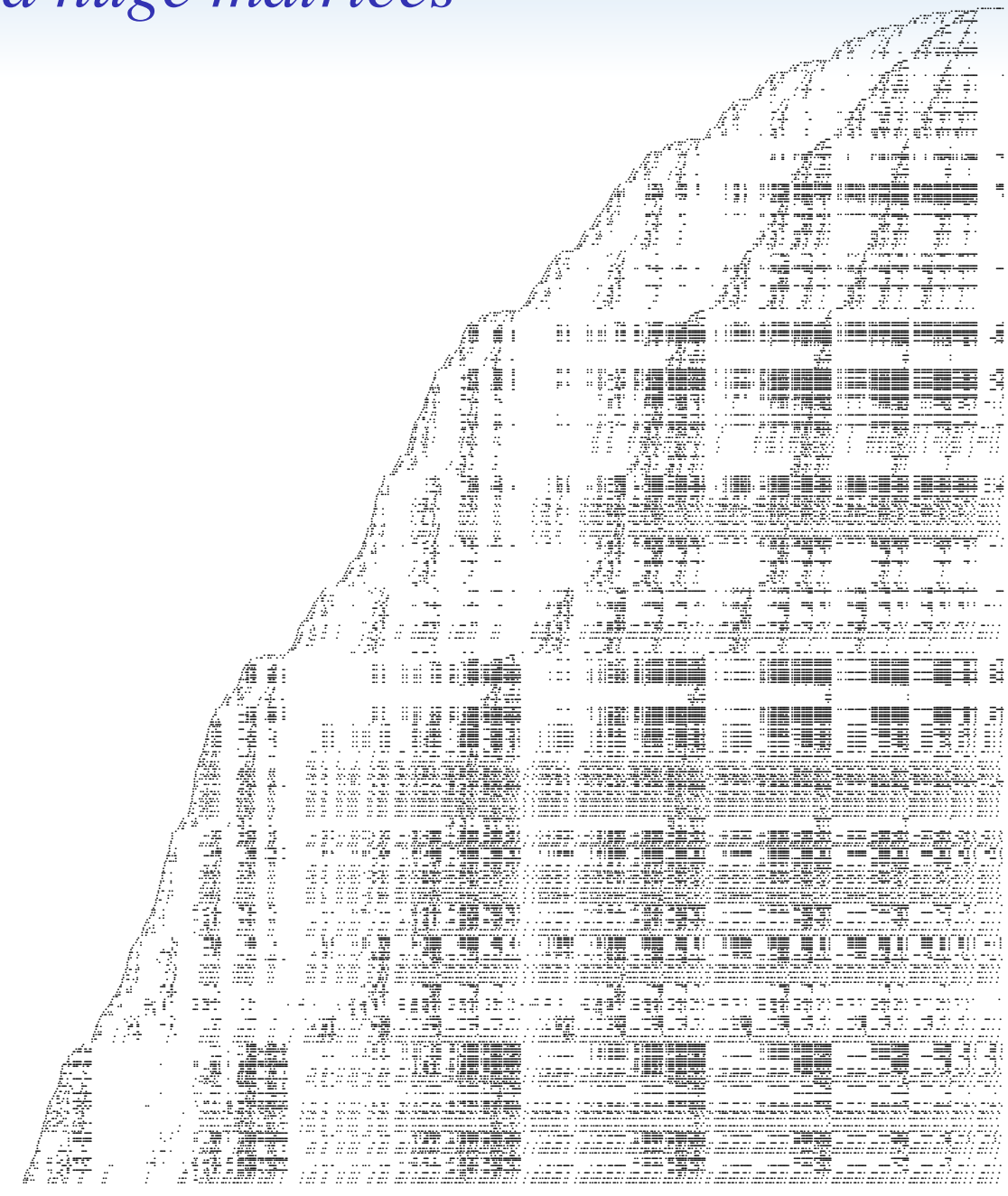
- including Buchberger Criteria (or F_5 criterion).

- reuse **all** the rows in the reduced matrices.
- Improve the linear algebra step (dedicated algorithms, matrix compression, . . .)

F_4 generated huge matrices



F_4 generated huge matrices



Références I


B. Buchberger.
An algorithm for finding the basis elements in the residue class ring modulo a zero dimensional polynomial ideal.
Journal of Symbolic Computation, 41(3-4):475–511, 3 2006.


Buchberger B.
Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.
PhD thesis, Innsbruck, 1965.


Buchberger B.
An Algorithmical Criterion for the Solvability of Algebraic Systems.
Aequationes Mathematicae, 4(3):374–383, 1970.
(German).


Cox D., Little J., and O'Shea D.
Ideals, Varieties and Algorithms.
Springer Verlag, New York, 1992.

Références II

J.-C. Faugère.
A new efficient algorithm for computing Gröbner bases (F4).
Journal of Pure and Applied Algebra, 139(1–3):61–88, June 1999.

Faugère, J.C., Gianni, P., Lazard, D. and Mora T.
Efficient Computation of Zero-Dimensional Gröbner Basis by Change of Ordering.
Journal of Symbolic Computation, 16(4):329–344, October 1993.

Sylvester J.
On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of sturm's functions, and that of the greatest algebraical common measure.
Philosophical Trans., 143:407–548, 1853.

Références III


F.S. Macaulay.
The algebraic theory of modular systems., volume xxxi of *Cambridge Mathematical Library.*
Cambridge University Press, 1916.