Polynomial Systems Solving by Fast Linear Algebra

J.-C. Faugère[1] P. Gaudry[2] L. Huot[1] G. Renault[1]

1: POLSYS Project INRIA Paris-Rocquencourt ; UPMC, Univ Paris 06, LIP6 ; CNRS, UMR 7606, LIP6
2: CARAMEL Project INRIA Grand-Est; Université de Lorraine; CNRS, UMR 7503; LORIA

PolSys Seminar, April 25th

Context

Motivation

Many applications: coding theory, cryptography, computational game theory, optimization *etc*

What means solving?

Depends on the context.

- find one solution;
- enumerate all the solutions in some field;
- find a certified approximation of the real solutions;
- ...

Problem: univariate polynomial representation (PoSSo)

Let $S = \{f_1, \ldots, f_s\} \subset \mathbb{K}[x_1, \ldots, x_n]$ be a set of polynomial equations with a **finite number of solutions** which are all **simple**. Find a univariate polynomial representation of the solutions of S *i.e.* $h_1, \ldots, h_n \in \mathbb{K}[x_n]$ such that $\{x_1 - h_1 = \cdots = x_{n-1} - h_{n-1} = h_n = 0\}$ has the same solutions as S.

State of the art

Let D be the number of solutions of S.

Particular case

$\mathbb{K}$: field of characteristic zero. Sub-cubic algorithms to:

- approximate all the real roots: $\widetilde{O}(12^n D^2)$ (Mourrain, Pan 1998);
- compute a rational parametrization knowing the multiplicative structure of the quotient ring: $\widetilde{O}(n2^n D^{\frac{5}{2}})$ (Bostan, Salvy, Schost 2003).

If the ideal is in *Shape Position*, the **univariate polynomial representation is given by the LEX Gröbner basis**.

General case

In the **best case**, the complexity of computing a **LEX Gröbner basis** is bounded by $O(nD^3)$.

Our aim

Providing the **first algorithm** with sub-cubic complexity to compute a univariate polynomial representation of the solutions.

PoSSo and Gröbner basis

In our context PoSSo $\equiv$ computing a LEX Gröbner basis.

Usual algorithm to compute a LEX Gröbner basis

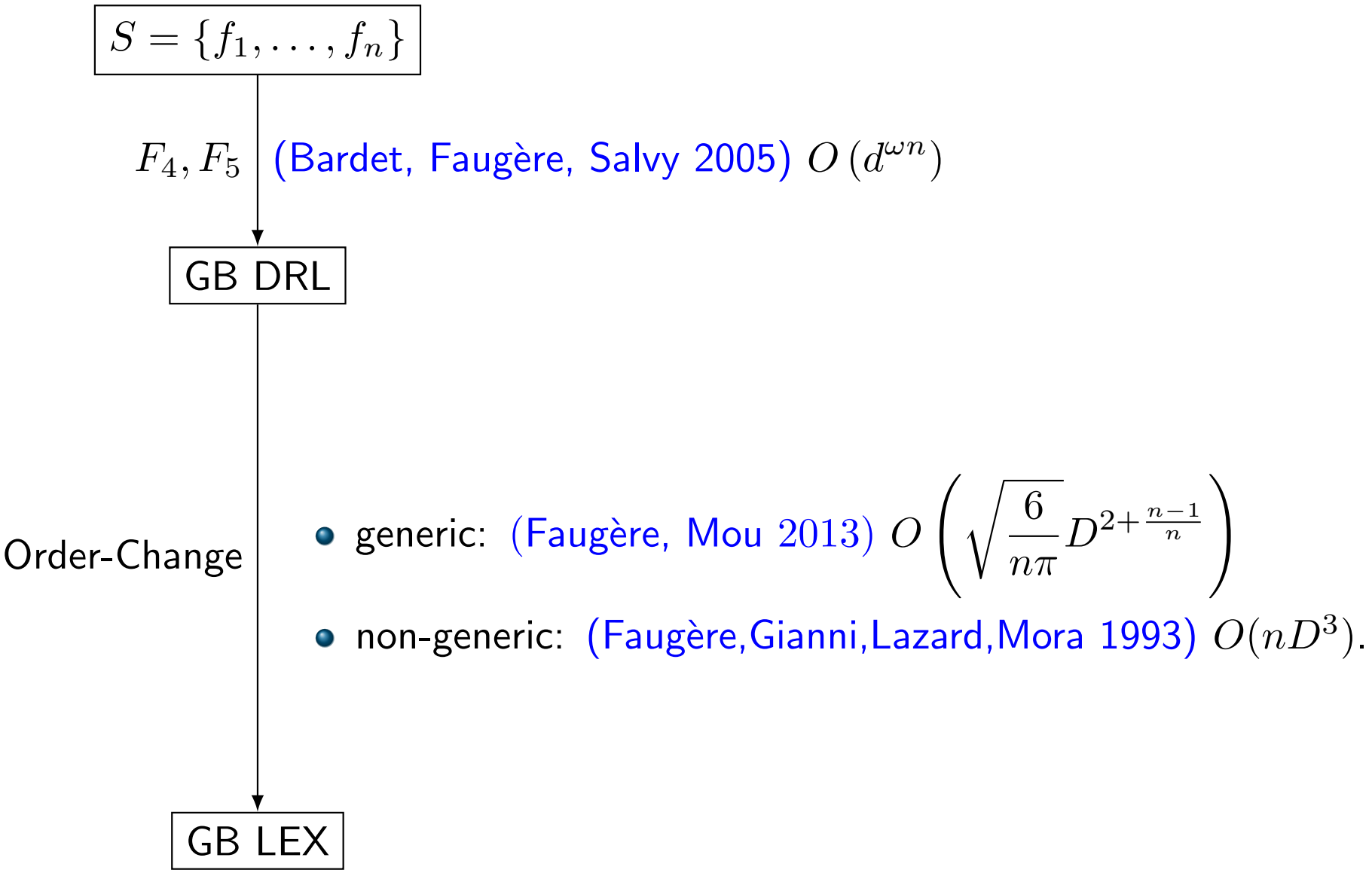
Input: A polynomial system $S \subset \mathbb{K}[x_1, \ldots, x_n]$.
Output: The LEX Gröbner basis of $\langle S \rangle$.

1. Compute the DRL Gröbner basis of $\langle S \rangle$;
2. From the DRL Gröbner basis, by using a change of ordering algorithm, compute the LEX Gröbner basis of $\langle S \rangle$.

- Gröbner basis algorithms:
  - Historical: (Buchberger 1965) Buchberger's algorithm;
  - Efficient: (Faugère 1999/2002) F_4 and F_5.
- Change of ordering algorithm:
  - (Faugère, Gianni, Lazard, Mora 1993) FGLM;
  - (Faugère, Mou 2011/2013) Sparse FGLM.

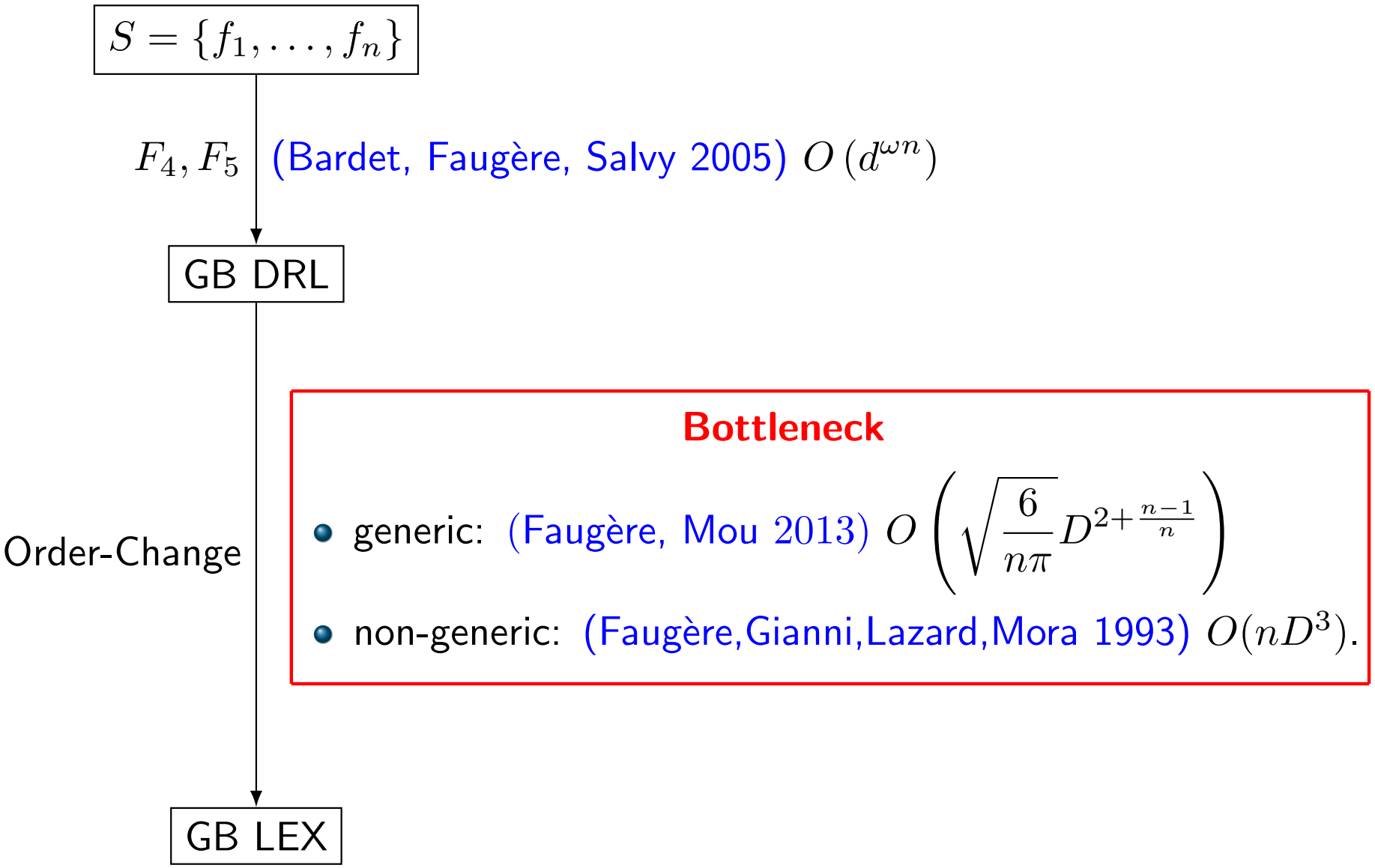
Gröbner basis and Complexity – State of the art

$(f_1, \ldots, f_n)$ regular sequence with $\deg(f_i) \leq d$.



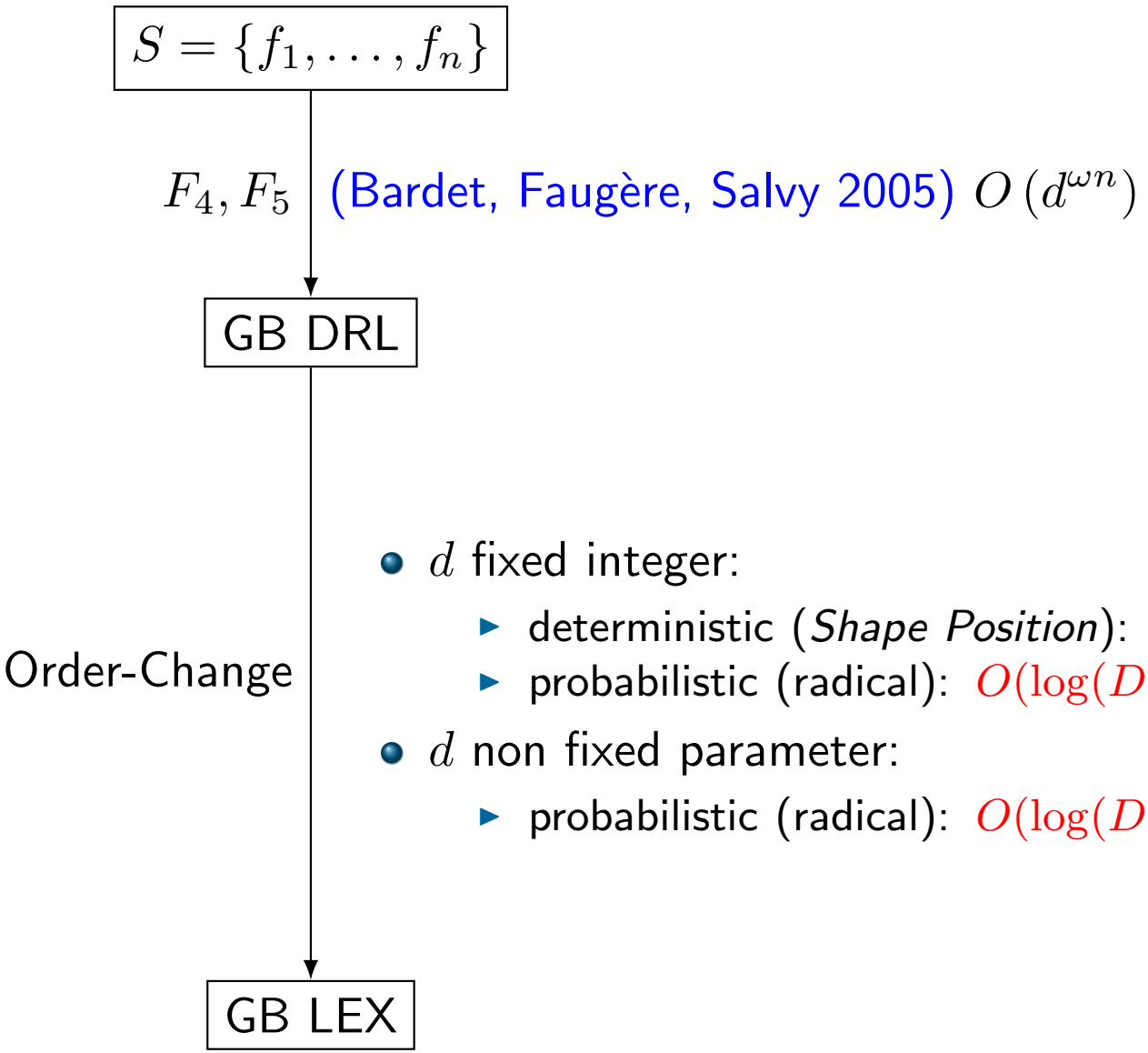
Gröbner basis and Complexity – State of the art

$(f_1, \ldots, f_n)$ regular sequence with $\deg(f_i) \leq d$.



Gröbner basis and Complexity – Contributions

$(f_1, \ldots, f_n)$ regular sequence with $\deg(f_i) \leq d$.



Contributions

Shape Position case

An ideal I is said to be in *Shape Position* if its LEX Gröbner basis if of the form:

$$\{x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), h_n(x_n)\} \ .$$

Main results

Let $S = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ with $\deg(f_i) \leq d$. If $(f_1, \ldots, f_n)$ is a **regular sequence** then

- if $\mathbf{d}$ is a **fixed integer** and $\langle S \rangle$ is in *Shape Position* then there exists a deterministic algorithm solving the PoSSo problem in $\widetilde{O}(d^{\omega n} + D^{\omega})$;

Contributions

Shape Position case

An ideal I is said to be in *Shape Position* if its LEX Gröbner basis if of the form:

$$\{x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), h_n(x_n)\} \ .$$

Main results

Let $S = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ with $\deg(f_i) \leq d$. If $(f_1, \ldots, f_n)$ is a **regular sequence** then

- if $\mathbf{d}$ is a **fixed integer** and $\langle S \rangle$ is in *Shape Position* then there exists a deterministic algorithm solving the PoSSo problem in $\widetilde{O}(d^{\omega n} + D^{\omega})$;
- if $\mathbf{d}$ is a **fixed or non fixed parameter** then there exists a Las Vegas algorithm solving the PoSSo problem in $\widetilde{O}(d^{\omega n} + D^{\omega})$.

Contributions

Shape Position case

An ideal I is said to be in *Shape Position* if its LEX Gröbner basis if of the form:

$$\{x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), h_n(x_n)\} \ .$$

Main results

Let $S = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ with $\deg(f_i) \leq d$. If $(f_1, \ldots, f_n)$ is a **regular sequence** then

- if $\mathbf{d}$ is a **fixed integer** and $\langle S \rangle$ is in *Shape Position* then there exists a deterministic algorithm solving the PoSSo problem in $\widetilde{O}(d^{\omega n} + D^{\omega})$;
- if $\mathbf{d}$ is a **fixed or non fixed parameter** then there exists a Las Vegas algorithm solving the PoSSo problem in $\widetilde{O}(d^{\omega n} + D^{\omega})$.

$$\text{If the Bézout's bound is reached } \widetilde{O}(d^{\omega n} + D^{\omega}) = \widetilde{O}(D^{\omega})$$

where $2 \leq \omega < 2.3727$ is the linear algebra constant.

Change of ordering algorithm

Given $\mathcal{G}_{\text{drl}}$ be the DRL Gröbner basis of an ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$.

I has a **finite number of solutions** $D \Rightarrow V = \mathbb{K}[x_1, \ldots, x_n]/\langle \mathcal{G}_{\text{drl}} \rangle$ is a $\mathbb{K}$-vector space of dimension D.

Let $B = \{1 = \epsilon_1 < \cdots < \epsilon_D\}$ be the canonical basis of V.

Algorithm

1. Compute the multiplicative structure of V *i.e.* the multiplication matrices $T_1, \ldots, T_n$ with

$$T_i = \begin{array}{c} \begin{matrix} \text{NF}_{\text{drl}}(\epsilon_1 x_i) & \cdots & \text{NF}_{\text{drl}}(\epsilon_D x_i) \end{matrix} \\ \begin{pmatrix} \star & \cdots & \star \\ \vdots & \ddots & \vdots \\ \star & \cdots & \star \end{pmatrix} \end{array} \begin{matrix} \epsilon_1 \\ \vdots \\ \epsilon_D \end{matrix}$$

2. From this multiplicative structure, recover the LEX Gröbner basis.

Key ideas

(Sparse) FGLM

- Multiplication matrices
  nD normal forms $\equiv$ dependent matrix-vector products

  $O(nD^3)$ arithmetic operations

- LEX Gröbner basis
  $2D$ matrix-vector products $T^j\mathbf{r}$ for $j = 0, \ldots, 2D-1$

  $O(D^3)$ arithmetic operations

This work

Key ideas

(Sparse) FGLM

- Multiplication matrices
  nD normal forms $\equiv$ dependent matrix-vector products
  $O(nD^3)$ arithmetic operations

- LEX Gröbner basis
  $2D$ matrix-vector products $T^j\mathbf{r}$ for $j = 0, \ldots, 2D-1$
  $O(D^3)$ arithmetic operations

This work

- Multiplication matrices
  $\log_2(D)$ row echelon form
  Fast matrix multiplication
  $\widetilde{O}(D^\omega)$ arithmetic operations

- LEX Gröbner basis
  $2\log_2(D)$ matrix products
  Fast matrix multiplication
  $\widetilde{O}(D^\omega)$ arithmetic operations

Fast Univariate Polynomial Representation

Univariate polynomial representation

Hypothesis

I is in *Shape Position i.e.*

$$\mathcal{G}_{\text{lex}} = \{x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), h_n(x_n)\}$$

where $\deg(h_n) = D$ and $\deg(h_i) = D - 1$ for $i = 1, \ldots, n-1$.

Problem

Given $\mathcal{G}_{\text{drl}}$ and the multiplication matrices $T_1, \ldots, T_n$ compute the polynomials h_n and $h_i = \sum_{k=0}^{D-1} c_{i,k} x_n^k$ for $i = 1, \ldots, n-1$.

Sparse change of ordering: Part 1

Faugère, Mou 2011 and 2013

1. Find the univariate polynomial $h_n(x_n)$
   Let $\mathbf{r}$ be a random vector.
   1. Construct $S = [\langle \mathbf{r}, T_n^i \mathbf{1} \rangle \ : \ i = 0, \ldots, 2D-1]$.
      Note that $\langle \mathbf{r}, T_n^i \mathbf{1} \rangle = \langle (T_n^t)^i \mathbf{r}, \mathbf{1} \rangle$
      $\rightsquigarrow O(ND)$
   2. Berlekamp-Massey $\Rightarrow$ minimal polynomial of $S = \mu$.
      If $\deg(\mu) = D$ then $h_n = \mu$
      $\rightsquigarrow O(\log_2(D)^2 D)$
2. Recover the coefficients $c_{i,k}$ by solving structured linear systems (Hankel matrices).
   $\rightsquigarrow O(n \log_2(D)^2 D)$

Sparse change of ordering

Lemma

From $\mathcal{G}_{\text{drl}}$, $T_i\mathbf{1}$ can be computed without T_i.

Theorem (Faugère and Mou)

Given T_n and $\mathcal{G}_{\text{drl}}$ of an ideal in *Shape Position*, the LEX Gröbner basis can be computed in

- $O(D(N + n\log_2(D)^2))$ (**probabilistic**)
- $O(D(N + D(n + \log_2(D)\log_2(\log_2(D)))))$ (**deterministic**)

arithmetic operations where N is the number of non zero entries in T_n.

Open issue

If T_n is assumed to be dense $\Rightarrow$ $O(D^3)$ arithmetic operations.
How to compute efficiently $T^j\mathbf{r}$ for $j = 0, \ldots, 2D-1$ with $T = T_n^t$?

Computing $T^j\mathbf{r}$ for $j = 0, \ldots, 2D-1$

1. Compute $T^2, T^4, \ldots, T^{2^{\lceil \log_2(D) \rceil}}$ with $\lceil \log_2(D) \rceil$ multiplication matrices.

$$O(\log_2(D) D^\omega) \text{ arithmetic operations}.$$

2. (Keller-Gehrig) Compute $\lceil \log_2(D) \rceil$ multiplication matrices of the form

$$\begin{aligned}
T^2(T\mathbf{r} \mid \mathbf{r}) &= (T^3\mathbf{r} \mid T^2\mathbf{r}) \\
T^4(T^3\mathbf{r} \mid T^2\mathbf{r} \mid T\mathbf{r} \mid \mathbf{r}) &= (T^7\mathbf{r} \mid T^6\mathbf{r} \mid T^5\mathbf{r} \mid T^4\mathbf{r}) \\
&\vdots \\
T^{2^{\lceil \log_2(D) \rceil}}(T^{2^{\lceil \log_2(D) \rceil - 1}}\mathbf{r} \mid \ldots \mid \mathbf{r}) &= (T^{2D-1}\mathbf{r} \mid T^{2D-2}\mathbf{r} \mid \ldots \mid T^{2^{\lceil \log_2(D) \rceil}}\mathbf{r}).
\end{aligned}$$

$$O(\log_2(D) D^\omega) \text{ arithmetic operations}.$$

Fast univariate polynomial representation

Theorem

Given T_n and $\mathcal{G}_{\mathsf{drl}}$ of an ideal in *Shape Position*, the LEX Gröbner basis can be computed in

- $O(\log_2(D)(D^\omega + n\log_2(D)D))$ (**probabilistic**);
- $O(\log_2(D)D^\omega + D^2(n + \log_2(D)\log_2(\log_2(D))))$ (**deterministic**).

Open issue

Compute T_n with less than $O(nD^3)$ arithmetic operations.

Multiplication Matrices

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

$$T_i = \begin{array}{c} \begin{matrix} \mathsf{NF}_{\mathsf{drl}}(\epsilon_1 x_i) & \cdots & \mathsf{NF}_{\mathsf{drl}}(\epsilon_D x_i) \end{matrix} \\ \begin{pmatrix} \star & \cdots & \star \\ \vdots & \ddots & \vdots \\ \star & \cdots & \star \end{pmatrix} \end{array} \begin{matrix} \epsilon_1 \\ \vdots \\ \epsilon_D \end{matrix}$$

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Proposition (Faugère, Gianni, Lazard, Mora)

Let $t = \epsilon_i x_j$; t can be

(I) either in B *i.e.* $t = \epsilon_k$ for some $k > i$;

(II) or in $E(I)$ *i.e.* $t = \mathsf{LT}_{\mathsf{drl}}(g)$ with $g \in \mathcal{G}_{\mathsf{drl}}$;

(III) or in $F \setminus E(I)$ *i.e.* $t = x_k t'$ with $t' \in F$.

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Proposition (Faugère, Gianni, Lazard, Mora)

Let $t = \epsilon_i x_j$; t can be

(I) either in B *i.e.* $t = \epsilon_k$ for some $k > i$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t$

(II) or in $E(I)$ *i.e.* $t = \mathsf{LT}_{\mathsf{drl}}(g)$ with $g \in \mathcal{G}_{\mathsf{drl}}$;

(III) or in $F \setminus E(I)$ *i.e.* $t = x_k t'$ with $t' \in F$.

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Proposition (Faugère, Gianni, Lazard, Mora)

Let $t = \epsilon_i x_j$; t can be

(I) either in B *i.e.* $t = \epsilon_k$ for some $k > i$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t$

(II) or in $E(I)$ *i.e.* $t = \mathsf{LT}_{\mathsf{drl}}(g)$ with $g \in \mathcal{G}_{\mathsf{drl}}$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t - g$

(III) or in $F \setminus E(I)$ *i.e.* $t = x_k t'$ with $t' \in F$.

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Proposition (Faugère, Gianni, Lazard, Mora)

Let $t = \epsilon_i x_j$; t can be

(I) either in B *i.e.* $t = \epsilon_k$ for some $k > i$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t$

(II) or in $E(I)$ *i.e.* $t = \mathsf{LT}_{\mathsf{drl}}(g)$ with $g \in \mathcal{G}_{\mathsf{drl}}$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t - g$

(III) or in $F \setminus E(I)$ *i.e.* $t = x_k t'$ with $t' \in F$.
$\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = \sum_{\ell=1}^{D} \alpha_\ell \mathsf{NF}_{\mathsf{drl}}(x_k \epsilon_\ell)$ where $\mathsf{NF}_{\mathsf{drl}}(t') = \sum_{i=\ell}^{D} \alpha_\ell \epsilon_\ell$

Computing $T_1, \ldots, T_n$: the original algorithm

Computing $T_1, \ldots, T_n$ requires to compute $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_j)$ for $i = 1, \ldots, D$ and $j = 1, \ldots, n$.

Let denote,

- $E(I) = \{\mathsf{LT}_{\mathsf{drl}}(g) \mid g \in \mathcal{G}_{\mathsf{drl}}\}$;
- $F = \{\epsilon_i x_j \mid i = 1, \ldots, D \text{ and } j = 1, \ldots, n\} \setminus B$.

Proposition (Faugère, Gianni, Lazard, Mora)

Let $t = \epsilon_i x_j$; t can be

(I) either in B *i.e.* $t = \epsilon_k$ for some $k > i$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t$

(II) or in $E(I)$ *i.e.* $t = \mathsf{LT}_{\mathsf{drl}}(g)$ with $g \in \mathcal{G}_{\mathsf{drl}}$; $\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = t - g$

(III) or in $F \setminus E(I)$ *i.e.* $t = x_k t'$ with $t' \in F$.
$\Rightarrow \mathsf{NF}_{\mathsf{drl}}(t) = \sum_{\ell=1}^{D} \alpha_\ell \mathsf{NF}_{\mathsf{drl}}(x_k \epsilon_\ell)$ where $\mathsf{NF}_{\mathsf{drl}}(t') = \sum_{i=\ell}^{D} \alpha_\ell \epsilon_\ell$

Complexity

$\#F \leq nD \Rightarrow$ at most nD matrix-vector products.

$O(nD^3)$ arithmetic operations

Computing $T_1, \ldots, T_n$ using fast linear algebra

The normal forms of all the monomials of same degree can be computed simultaneously.

$M =$

	$t \in F_d$				$t \in F_{d-1} \cup \cdots \cup F_{d_{\min}}$			$t \in B$		
f_{i+s_d}	1	$\star$	$\cdots$	$\star$	$\star$	$\cdots$	$\star$	$\star$	$\cdots$	$\star$
f_{i+s_d-1}	0	1	$\cdots$	$\star$	$\star$	$\cdots$	$\star$	$\star$	$\cdots$	$\star$
$\vdots$	$\vdots$	**T**	$\ddots$	$\vdots$	$\vdots$	**B**	$\vdots$	$\vdots$	**C**	$\vdots$
f_i	0	0	$\cdots$	1	$\star$	$\cdots$	$\star$	$\star$	$\cdots$	$\star$
$\widetilde{f}_{i-1}$	0	0	$\cdots$	0	1	$\cdots$	0	$\star$	$\cdots$	$\star$
$\vdots$	$\vdots$			$\vdots$	**Id**	$\ddots$		$\vdots$	**D**	$\vdots$
$\widetilde{f}_1$	0	0	$\cdots$	0	0	$\cdots$	1	$\star$	$\cdots$	$\star$

where $F_d = \{t \in F \mid \deg(t) = d\} = \{t_i < \cdots < t_{i+s_d}\}$

- $\widetilde{f}_i = t_i - \mathsf{NF}_{\mathsf{drl}}(t_i)$
- If $t_i \in E(I)$ then $f_i = \widetilde{f}_i = g$ with $g \in \mathcal{G}_{\mathsf{drl}}$ st $\mathsf{LT}_{\mathsf{drl}}(g) = t_i$.
- Else $t_i \in F \setminus E(I) \Rightarrow t_i = x_k t_j$ and $f_i = x_k \widetilde{f}_j = t_i + \sum_{i=1}^{D} \alpha_j x_k \epsilon_j$.

Computing $T_1, \ldots, T_n$ using fast linear algebra

M: matrix of size at most $(nD \times (n+1)D)$.
Computing the row echelon form of M can be done in $O(n^\omega D^\omega)$ arithmetic operations.

Theorem

Given $\mathcal{G}_{\text{drl}}$, computing all the multiplication matrices $T_1, \ldots, T_n$ can be done in

$$O(d_{\max} n^\omega D^\omega) \text{ arithmetic operations}$$

where $d_{\max} = \max\{\deg(t) \mid t \in F\} = \max\{\deg(g) \mid g \in \mathcal{G}_{\text{drl}}\}$.

Regular System

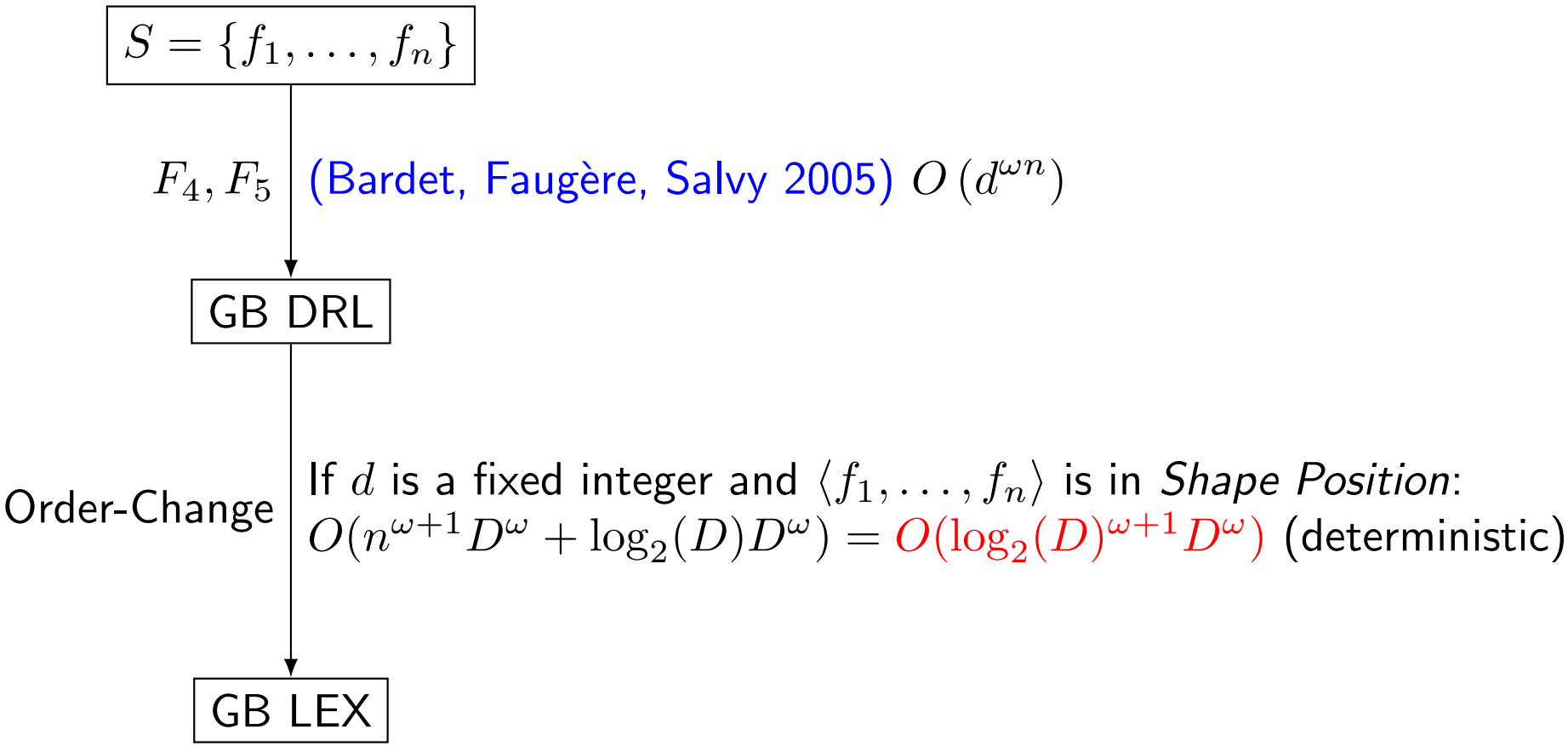
Let $S = \{f_1, \ldots, f_n\}$ with $\deg(f_i) \leq d$ and $(f_1, \ldots, f_n)$ is a regular sequence.

- Macaulay's bound $\Rightarrow d_{\max} \leq n(d-1)+1$;
- Bézout's bound $\Rightarrow D \leq d^n$.

If d is a fixed integer then computing $T_1, \ldots, T_n$ given $\mathcal{G}_{\text{drl}}$ can be done in $O(n^{\omega+1} D^\omega) = O(\log_2(D)^{\omega+1} D^\omega)$ arithmetic operations.

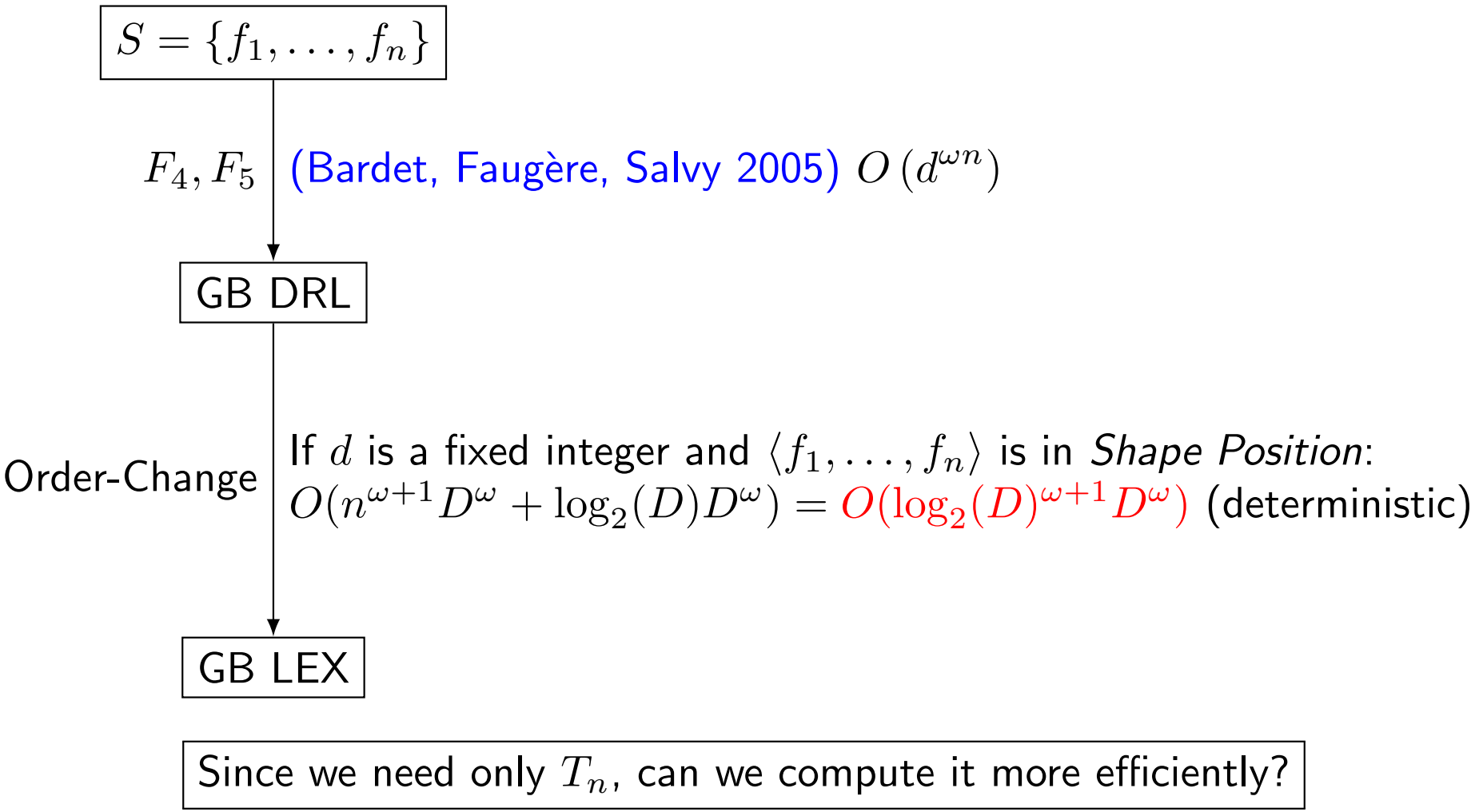
Gröbner basis and Complexity – Contributions

$(f_1, \ldots, f_n)$ regular sequence with $\deg(f_i) \leq d$.



Gröbner basis and Complexity – Contributions

$(f_1, \ldots, f_n)$ regular sequence with $\deg(f_i) \leq d$.



Construction of T_n: the generic case

To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Construction of T_n: the generic case

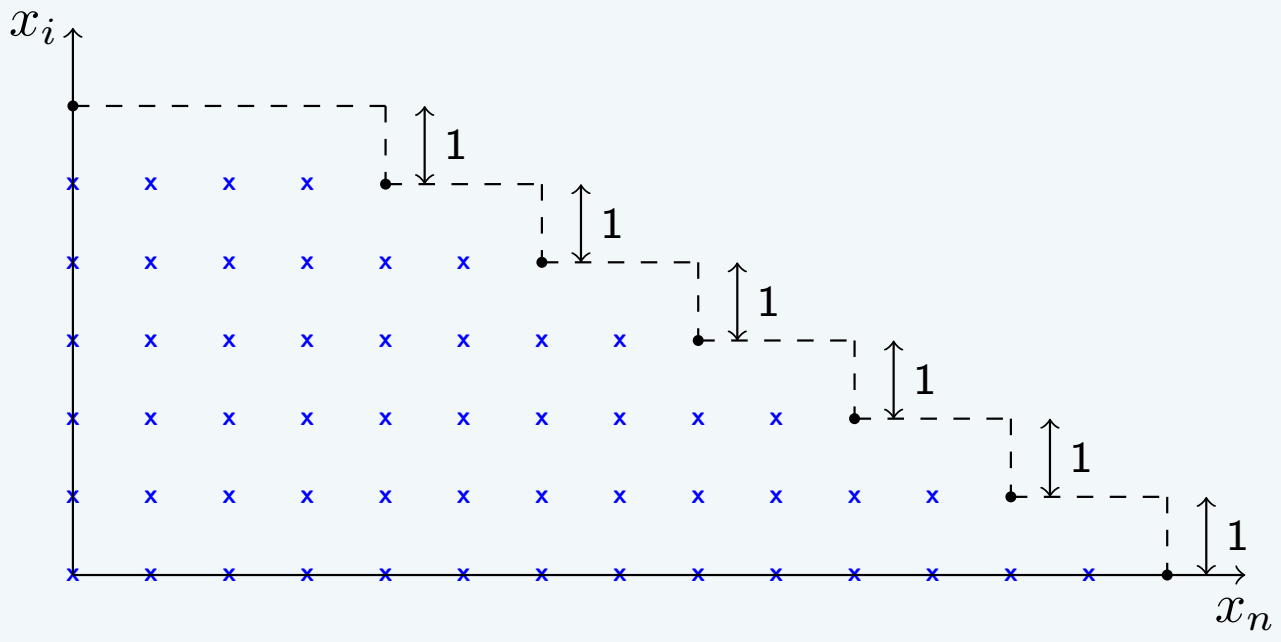
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the generic case

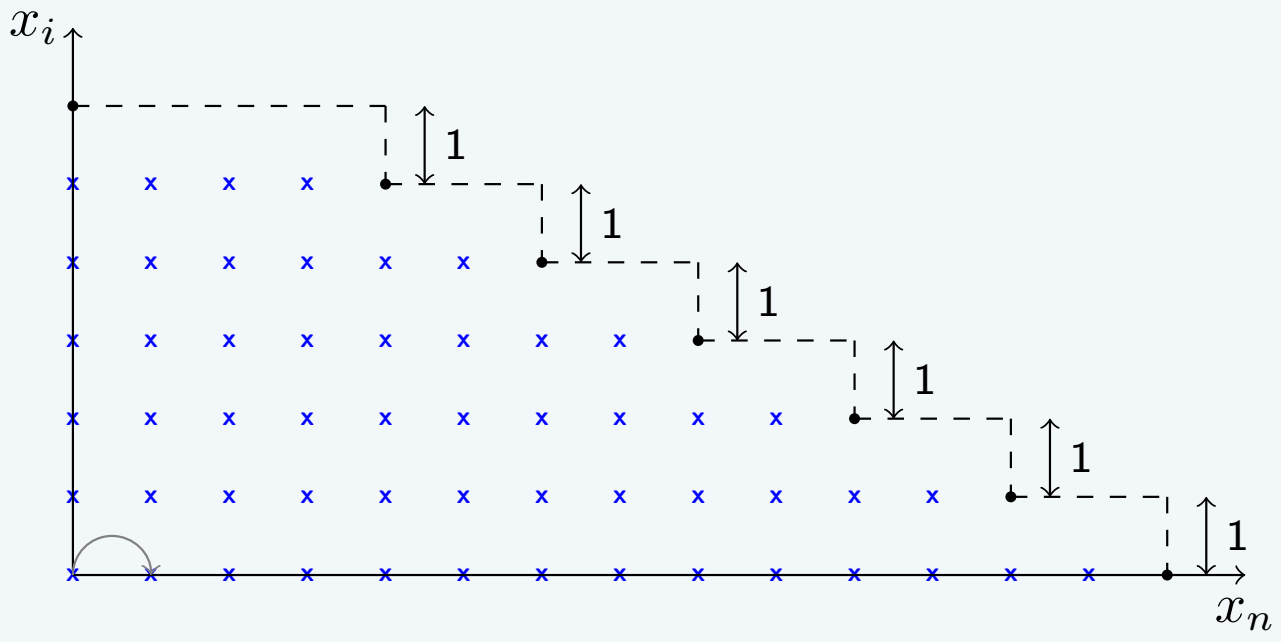
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the generic case

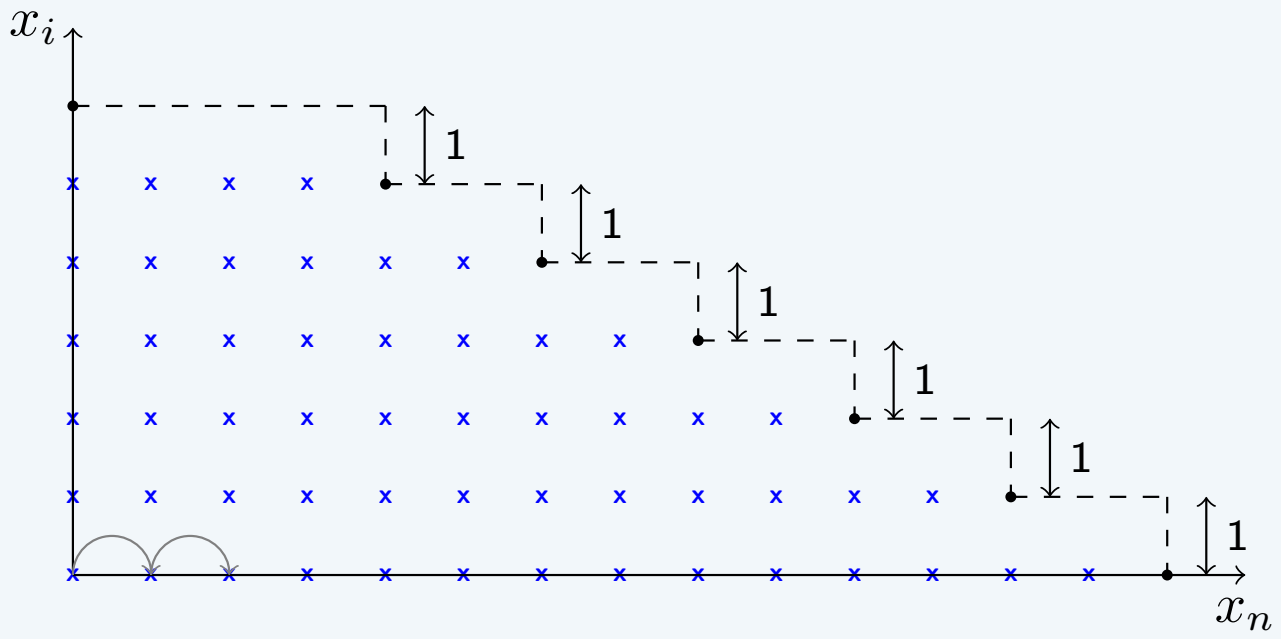
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the generic case

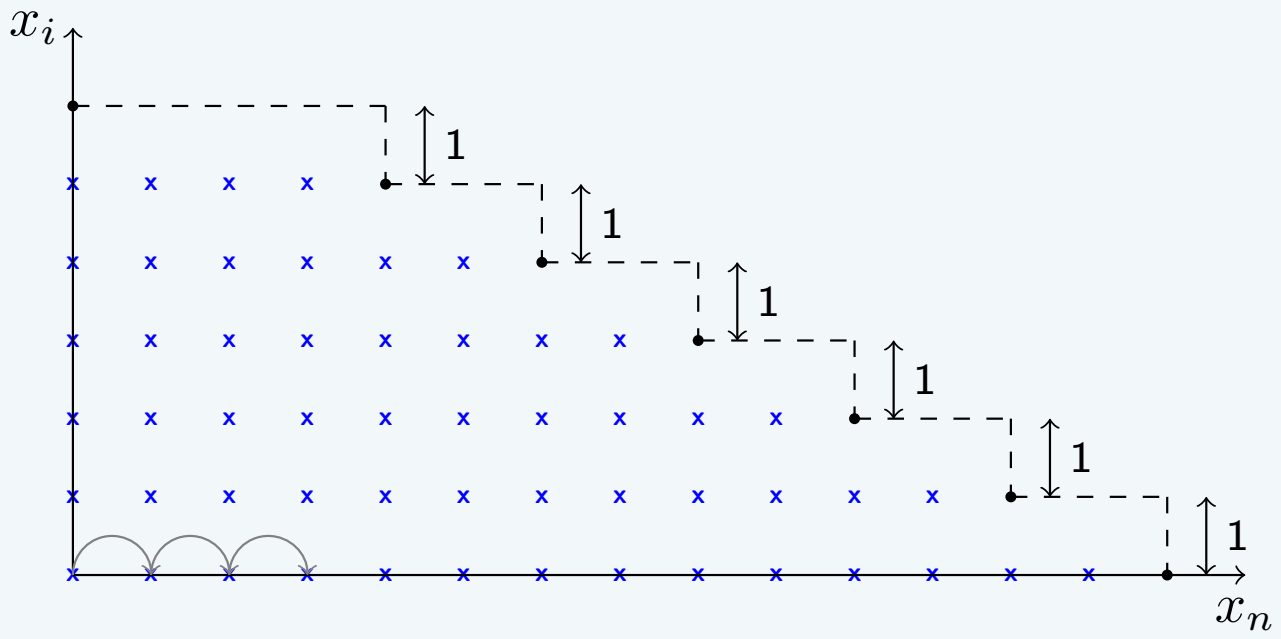
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the generic case

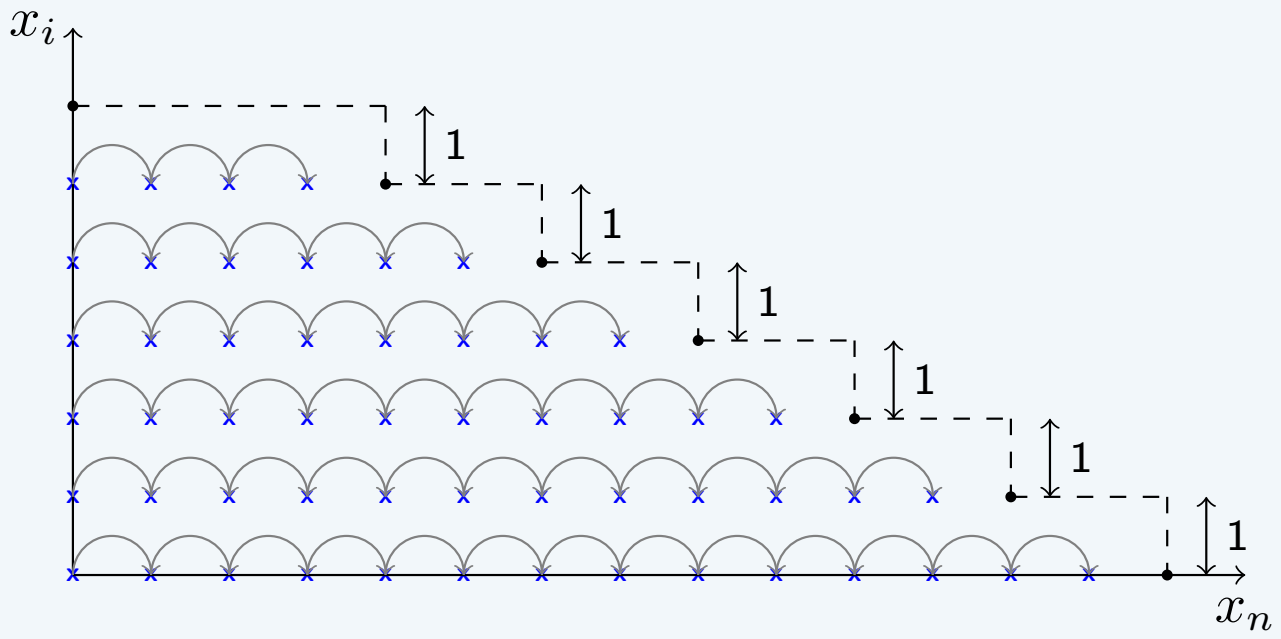
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the generic case

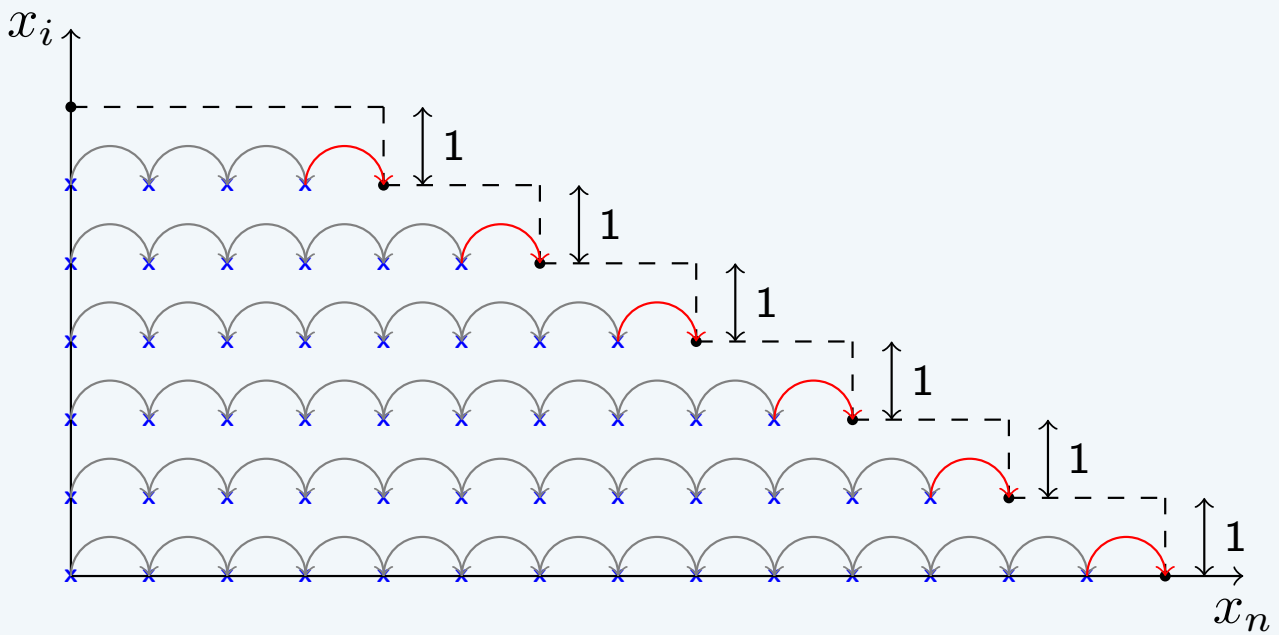
To compute T_n we only need $\mathsf{NF}_{\mathsf{drl}}(\epsilon_i x_n)$ for $i = 1, \ldots, D$.

Proposition

For generic ideals, $\epsilon_i x_n \in B \cup E(I)$ for $i = 1, \ldots, D$.

Moreno-Socias

For any instantiation of $\deg_{x_j}$ for $j \in \{1, \ldots, n-1\} \setminus \{i\}$



Construction of T_n: the non-generic case

Galligo, Bayer and Stillman, Pardue

Let I be an homogeneous ideal. Let p denotes the characteristic of $\mathbb{K}$. If $p = 0$ or if p is sufficiently large then there exists a Zariski open subset $U \subset \mathsf{GL}(\mathbb{K}, n)$ such that $\forall g \in U$, $g \cdot \mathcal{I}$ has the structure of generic ideals.

Can be generalized to affine ideals.

Theorem

Let p denotes the characteristic of $\mathbb{K}$. If $p = 0$ or if p is sufficiently large and $\mathbb{K}$ too then **no arithmetic operations** are required to compute T_n of $g \cdot \mathcal{I}$ where g is randomly chosen in $\mathsf{GL}(\mathbb{K}, n)$.

New algorithm for PoSSo

Shape Lemma

Let I be a **radical** ideal. There exists a Zariski open subset $U' \subset \mathrm{GL}(\mathbb{K}, n)$ such that for all $g \in U'$, $g \cdot I$ is in *Shape Position*.

Another algorithm for PoSSo.

Input : A polynomial system $S = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ generating a radical ideal.

Output: g in $\mathrm{GL}(\mathbb{K}, n)$ and the LEX Gröbner basis of $\langle g \cdot S \rangle$ or *fail*.

Choose randomly g in $\mathrm{GL}(\mathbb{K}, n)$;

Compute $\mathcal{G}_{\mathsf{drl}}$ the DRL Gröbner basis of $g \cdot S$;

if T_n *can be read from* $\mathcal{G}_{drl}$ **then**

  Extract T_n from $\mathcal{G}_{\mathsf{drl}}$;

  if $\langle g \cdot S \rangle$ *is in Shape Position* **then**

    From T_n and $\mathcal{G}_{\mathsf{drl}}$ compute $\mathcal{G}_{\mathsf{lex}}$;

    return g and $\mathcal{G}_{\mathsf{lex}}$;

return *fail*;

New algorithm for PoSSo: Probability and Complexity

Probability of success

Let p be the characteristic of $\mathbb{K}$ and $\mathbb{P}$ the probability of success.

- If $p = 0$, then $\mathbb{P} = 1$;
- If $p > \sum_{i=1}^{n}(\deg(f_i) - 1) + 1$, then
  $\mathbb{P} \geq 1 - \frac{1}{\#\mathbb{K}} \left(\frac{D(D-1)}{2} + (\sum_{i=1}^{n}(\deg(f_i) - 1) + 1)\, nD \right)$.

New algorithm for PoSSo: Probability and Complexity

Probability of success

Let p be the characteristic of $\mathbb{K}$ and $\mathbb{P}$ the probability of success.

- If $p = 0$, then $\mathbb{P} = 1$;
- If $p > \sum_{i=1}^{n}(\deg(f_i) - 1) + 1$, then
  $\mathbb{P} \geq 1 - \frac{1}{\#\mathbb{K}}\left(\frac{D(D-1)}{2} + \left(\sum_{i=1}^{n}(\deg(f_i) - 1) + 1\right) nD\right)$.

Complexity

Let d such that $\deg(f_i) \leq d$.

- Compute $\mathcal{G}_{\text{drl}}$ of $\langle g \cdot S \rangle$: $O(d^{\omega n})$ arithmetic operations;
- Compute T_n: no arithmetic operation;
- Compute $\mathcal{G}_{\text{lex}}$ given T_n: $O(\log_2(D)(D^{\omega} + n\log_2(D)D))$ arithmetic operations.

Total: $O(d^{\omega n} + \log_2(D)D^{\omega})$ arithmetic operations.

Experiments

$\mathcal{I} = \langle f_1, \ldots, f_n \rangle$ where $f_i = x_i^2 + \sum_{k=1}^{n} \left(c_k x_k + \sum_{\substack{j>i \\ j \neq k}} c_{k,j} x_k x_j \right)$ with $c_k, c_{k,j} \in \mathbb{K}$.

$$\rightsquigarrow \mathcal{G}_{\text{drl}} = \{f_1, \ldots, f_n\}$$

Normal forms to compute : $x_n^2 \cdot m$ for all monomial m linear in each the $n-1$ first variables $\Rightarrow 2^{n-1} - 1$.

n	Algorithm	D	DRL	T_n	Univariate polynomial representation	Total
9	$F_5 \rightarrow$ FGLM	512	0.00s	255NF 12.81s	0.29s	13.10s
9	This work	512	0.00s	0.00s	0.01s	**0.01s**
11	$F_5 \rightarrow$ FGLM	2^{11}	0.00s	1023NF 7520.89s	$>$ 2 days	$>$ 2 days
11	This work	2^{11}	5.02s	0.15s	0.13s	**5.30s**
13	$F_5 \rightarrow$ FGLM	2^{13}	0.00s	4095NF $>$ 2 days		$>$ 2 days
13	This work	2^{13}	156.67s	2.36s	25.80s	**184.83s**

To summarize

$S = \{f_1, \ldots, f_n\}$ with $\deg(f_i) \leq d$.

New algorithms

- PoSSo
- Change of ordering

	deterministic	probabilistic
◇ Multiplication matrices	$O(dn^{\omega+1}D^{\omega})$	free
◇ Univariate polynomial representation	$O(\log_2(D)D^{\omega})$	$O(\log_2(D)D^{\omega})$

New complexity for PoSSo (simple roots)

- d **fixed** integer:
  - deterministic: (*Shape Position*) $O(d^{\omega n} + \log_2(D)^{\omega+1}D^{\omega})$ arithmetic operations;
  - probabilistic: $O(d^{\omega n} + \log_2(D)D^{\omega})$ arithmetic operations;
- d **non fixed** parameter:
  - probabilistic: $O(d^{\omega n} + \log_2(D)D^{\omega})$ arithmetic operations.