GBRELA Workshop 2013

Jean-Charles Faugère

F_4 algorithm
F_5 algorithm

Hagenberg, Austria
September 03 – 06, 2013

informatics mathematics Inria
UPMC SORBONNE UNIVERSITÉS
cnrs dépasser les frontières
www.cnrs.fr

F_4

The F_4 algorithm

Definition

A critical pair of (f_i, f_j) is a member of

$$T \times T \times \mathbb{K}[x_1, \ldots, x_n] \times T \times \mathbb{K}[x_1, \ldots, x_n],$$
$$\mathrm{Pair}(f_i, f_j) := (\mathrm{lcm}_{ij}, t_i, f_i, t_j, f_j)$$

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:

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

Algorithm F_4 (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: a finite subset of $\mathbb{K}[x_1,\ldots,x_n]$.

$G := F$, $\tilde{F}_0^+ := F$, $d := 0$ and $P := \{\mathrm{Pair}(f,g) \mid (f,g) \in G^2$ 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)
$\widetilde{F}$:= 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* $\widetilde{F}$
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 $\mathrm{T}(F) \setminus Done$
  $Done := Done \cup \{m\}$
  if m top reducible 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 $\#\mathcal{S}el(l) = 1$ for all $l \neq \emptyset$ then the F_4 algorithm reduces to the Buchberger algorithm. In this case the function $\mathcal{S}el$ 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 $d + 1$.

Optimizations

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

Algorithm Buchberger Criteria - Implementation

$(G_{new}, P_{new}) := \textsc{Update}(G_{old}, P_{old}, h)$

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

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

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

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

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$(L_1, G, \emptyset)$:

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

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 $L_1 = \{(1, f_3), (b, f_4)\}$.

$\textsc{SymbolicPreprocessing}(L_1, G, \emptyset)$:

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

Example (Cyclic 4)

Monomial ordering is DRL and the normal strategy

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

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$(L_1, G, \emptyset)$:

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

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$(L_1, G, \emptyset)$:

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

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$(L_1, G, \emptyset)$:

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

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:

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

Example (Cyclic 4)

Gaussian reduction of A_1 is:

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

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 $bd\, f_4$ and also by $b\, f_5$!

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
$F_2 = \{f_2, cf_6\}$ $T(F_2) = \{\boxed{abc}, bc^2, \boxed{abd}, acd, bcd, cd^2\}$
abd is reducible by $bd\,f_4$ and also by $b\,f_5$!

We describe now Simplify :

Goal

replace any product $m \cdot f$ by a product $(u\,t) \cdot f'$ where (t, f') is a previously computed row and $u\,t$ divides the monomial m

Optimizations

In the first version of the algorithm: some rows of the matrix are never used (the rows in the matrix $\tilde{F}_d \backslash F_d^+$).
New version of the algorithm: we keep these rows

$$m \cdot f \in \text{Rows}(F) \longrightarrow m' \cdot f' \text{ with } m \geqslant m'$$
$$m \cdot f \in \text{Rows}(F) \longrightarrow x_k \cdot f'$$

SIMPLIFY tries to replace the product $m \cdot f$ by a product $(u\,t) \cdot f'$ where (t, f') is an already computed row in the gaussian reduction and $u\,t$ divides the monomial m; if we found such a better product then we call recursively the function SIMPLIFY:

Algorithm SIMPLIFY

Input: $\begin{cases} t \in T \text{ a monomial} \\ f \in \mathbb{K}[x_1, \ldots, x_n] \text{ a polynomial} \\ \mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}, \text{ where } F_k \subset \mathbb{K}[x_1, \ldots, x_n] \end{cases}$

Output: a product $m' \cdot f'$ equivalent to $t \cdot f$

for $u \in$ list of divisors of t **do**

 if $\exists j \ (1 \leqslant j < d)$ such that $(u \cdot f) \in F_j$ **then**

  $\tilde{F}_j$ is the Gaussian reduction of F_j wrt $<$

  there exists a unique $p \in \tilde{F}_j$ such that $\mathrm{LT}(p) = \mathrm{LT}(u \cdot f)$

  if $u \neq t$ **then**

   return SIMPLIFY$(\frac{t}{u}, p, \mathcal{F})$

  else

   return $1 \cdot p$

return $t \cdot f$

Algorithm SymbolicPreprocessing

Input: $\left\{ \begin{array}{l} L, G \text{ finite subsets of } \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}, \text{ where } F_k \\ \quad \text{a finite subset of } \mathbb{K}[x_1, \ldots, x_n] \end{array} \right.$

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

$F := L$

$Done := \mathrm{LT}(F)$

while $T(F) \neq Done$ **do**

  choose m an element of $T(F) \backslash Done$

  $Done := Done \cup \{m\}$

  if m top reducible modulo G **then**

    exists $g \in G$ and $m' \in T$ such that $m = m' \cdot \mathrm{LT}(g)$

    $F := F \cup \{\text{Simplify}(m', g, \mathcal{F})\}$

return F

In practice ...

Remark

In practice the result of Simplify is to return in 95% $x_i \cdot p$ where x_i is a variable
(and most often the product $x_n \cdot p$).
In some sense, these is somewhat similar to the FGLM algorithm where we use the multiplication matrices to compute normal forms.

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 $L_2 = \{(1, f_2), (c, f_6)\}$ 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 $bd\, f_4$:
SIMPLIFY: replace $bd\, f_4$ by $b\, f_5$, so that abd is reducible by $b\, f_5$!

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

$F_2 = \{f_2, cf_6, b\,f_5\} \quad T(F_2) = \{\boxed{abc}, bc^2, \boxed{abd}, acd, bcd, cd^2, b^2d, bd^2\}$

Example (Cyclic 4)

And so on . . .

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

$F_2 = [cf_5, df_7, bf_5, f_2, cf_6]$

$$A_2 = M(F_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Example (Cyclic 4)

Apply Gaussian reduction:

$$\tilde{A}_2 = \widetilde{M(F_2)} = \begin{bmatrix} & & & & 1 & 1 & 1 & & 1 & \\ & & & 1 & & & & 2 & & 1 \\ & & 1 & & & 1 & & -1 & & -1 \\ 1 & & & & & -1 & -1 & 1 & -1 & 1 \\ & 1 & & & & & 1 & -1 & & -1 \end{bmatrix}$$

Example (Cyclic 4)

$$\tilde{A}_2 = \widetilde{M(F_2)} = \begin{bmatrix} & & & & 1 & 1 & 1 & & 1 & \\ & & & 1 & & & & 2 & & 1 \\ & & 1 & & & 1 & & -1 & & -1 \\ 1 & & & & & -1 & -1 & 1 & -1 & 1 \\ & 1 & & & & & 1 & -1 & & -1 \end{bmatrix}$$

$\tilde{F}_2 = [f_9 = acd + bcd + c^2d + cd^2$,
$f_{10} = b^2d + 2\,bd^2 + d^3$,
$f_{11} = abd + bcd - bd^2 - d^3$,
$f_{12} = abc - bcd - c^2d + bd^2 - cd^2 + d^3$,
$f_{13} = bc^2 + c^2d - bd^2 - d^3]$ and

$$G = \{f_4, f_7, f_{13}\}.$$

Example (Cyclic 4)

For the next step we have

$$L_3 = \{(1, f_1), (bcd, f_4), (c^2, f_7), (b, f_{13})\}$$

and we recursively call Simplify:
$\textsc{Simplify}(bcd, f_4) = \textsc{Simplify}(cd, f_6) = \textsc{Simplify}(d, f_{12}) = (d, f_{12})$.

Example (Cyclic 4)

For the next step we have

$$L_3 = \{(1, f_1), (bcd, f_4), (c^2, f_7), (b, f_{13})\}$$

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}].$$

After few steps in SymbolicPreprocessing we found that

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}, df_{13}, df_{10}]$$

Example (Cyclic 4)

For the next step we have

$$L_3 = \{(1, f_1), (bcd, f_4), (c^2, f_7), (b, f_{13})\}$$

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}].$$

SymbolicPreprocessing

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}, df_{13}, df_{10}]$$

Doing some computations we found that the rank of $M(F_3)$ is only 5. This means that there is a useless reduction to zero !

Example (Cyclic 4)

For the next step we have

$$L_3 = \{(1, f_1), (bcd, f_4), (c^2, f_7), (b, f_{13})\}$$

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}].$$

SymbolicPreprocessing

$$F_3 = [f_1, df_{12}, c^2 f_7, bf_{13}, df_{13}, df_{10}]$$

$$\tilde{F}_3 = \begin{bmatrix} f_{15} = c^2 b^2 - c^2 d^2 + 2\, bd^3 + 2\, d^4, \\ f_{16} = abcd - 1, \\ f_{17} = -bcd^2 - c^2 d^2 + bd^3 - cd^3 + d^4 + 1, \\ f_{18} = c^2 bd + c^2 d^2 - bd^3 - d^4, \\ f_{19} = b^2 d^2 + 2\, bd^3 + d^4 \end{bmatrix}$$

Linear Algebra

To compute the Gaussian Elimination is the most costly (CPU/Memory):

Compress the storage of the matrices

More involved way to store the matrices $\searrow$ memory request:

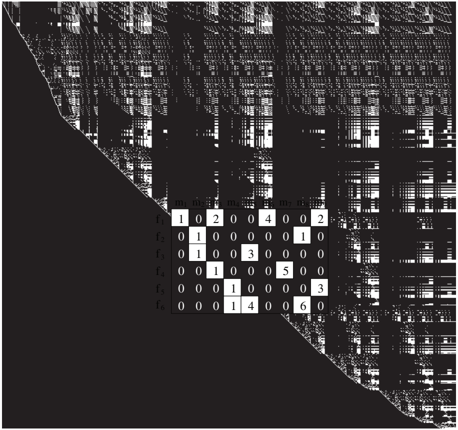
a matrix of dimension $5 \cdot 10^4 \times 5 \cdot 10^4$ with 10% non zero elements

if 1 byte is needed per coefficient

$\Rightarrow$ $\boxed{25 \cdot 10^7}$ bytes $\approx$ 238 MB to store the full matrix !

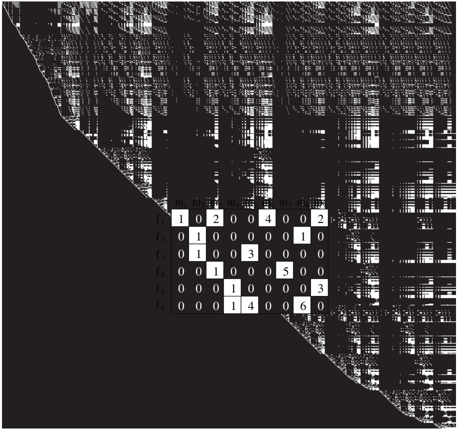
Shape of the generated matrices

Katsura 7 in $\mathbb{F}_{65521}$: 694×738 matrix of density 8%



Shape of the generated matrices

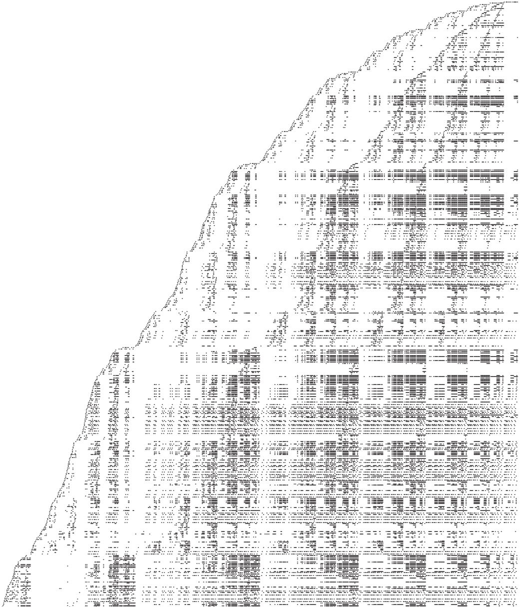
Katsura 7 in $\mathbb{F}_{65521}$: 694 $\times$ 738 matrix of density 8%



- sparse [0.1-25%],
- almost block triangular,
- can be huge (e.g. $1.6 \cdot 10^6$ columns for HFE Challenge 1).

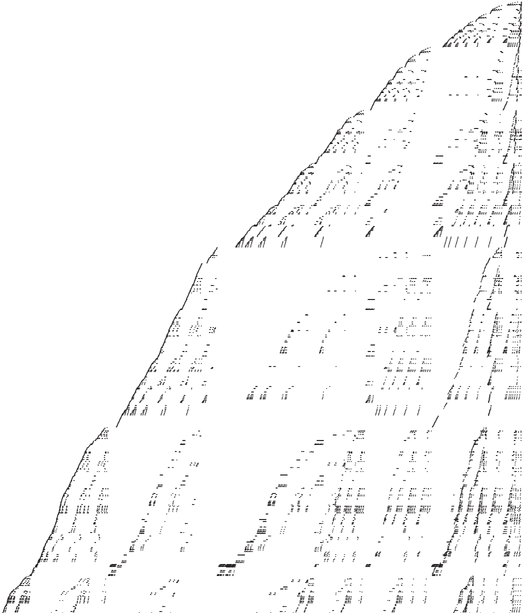
Example of matrix

generated by F_4*: Cyclic* 7



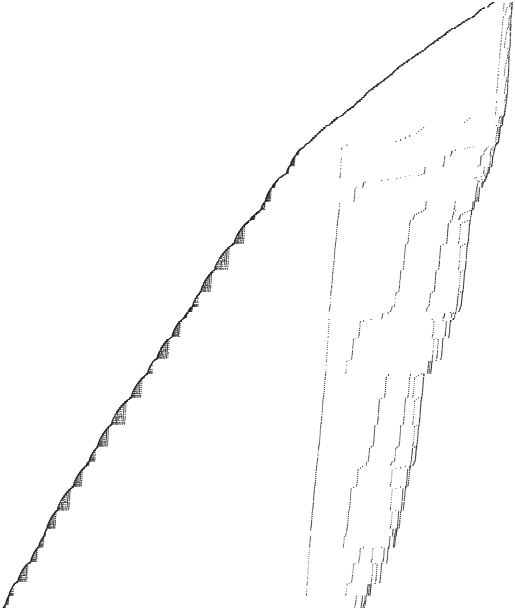
Example of matrix

generated by F_4: Filter banks F855



Example of matrix

generated by F_4



Compress the matrices

To compute the Gaussian Elimination is the most costly (CPU/Memory):
Implementations: we avoid to duplicate the coefficients:
most of the rows are multiplications of the **same** polynomial f by several monomials $\longrightarrow$ we have to consider only the position of the non zero elements in the matrix:
$\longrightarrow$ This is equivalent to compress a sequence of 1 and 0 (bitmap).

Compress the matrices

(i) Compression bitmap: denote by

$$j_1, j_2, j_{3,} \ldots$$

the position of the non zero elements in the matrix, then

$$\sum_k 2^{j_k - 1}$$

is the corresponding bitmap.
This is efficient but the reduction factor is not big (constant factor).

Compress the matrices

(ii) Another idea is to consider the differences (Lempel-Ziv coding):

$$\boxed{j_1}\boxed{j_2 - j_1}\boxed{j_3 - j_2}\cdots$$

when the difference $j_k - j_{k-1}$ is small (< 128), $\longrightarrow$ we can use one byte to store the result.
This method is more efficient wrt the memory usage and only slightly slower (10%).

F_5

$$F_5$$

Algorithms: for *computing* Gröbner bases.

- Buchberger (1965,1979,1985)
- F_4 using linear algebra (1999) (strategies)
- F_5 no reduction to zero (2002)
  - Today $\longrightarrow$ simple matrix F_5 algorithm

F_5 *algorithm*

- Goal: avoid (useless) reduction to 0
- Incremental algorithm

$$(f_m) + G_{\text{prev}}$$

- We have to explain: new F_5 criterion

F_5 *the idea I*

We consider the following example: (b is a parameter):

$$\mathcal{S}_b \begin{cases} f_3 = x^2 + 18\,xy + 19\,y^2 + 8\,xz + 5\,yz + 7\,z^2 \\ f_2 = 3\,x^2 + (7 + \;b)\,x\,y + 22\,x\,z + 11\,yz + 22\,z^2 + 8\,y^2 \\ f_1 = 6\,x^2 + 12\,xy + 4\,y^2 + 14\,xz + 9\,yz + 7\,z^2 \end{cases}$$

For now we assume that $b = 0$
With Buchberger $x > y > z$:

- 5 useless reductions
- 5 useful pairs

F_5 *the idea II*

We proceed degree by degree.

$$A_2 = \begin{array}{c} \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{cccccc} x^2 & x\,y & y^2 & x\,z & y\,z & z^2 \\ \hline 1 & 18 & 19 & 8 & 5 & 7 \\ 3 & 7 & 8 & 22 & 11 & 22 \\ 6 & 12 & 4 & 14 & 9 & 7 \end{array}$$

$$\widetilde{A_2} = \begin{array}{c} \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{cccccc} x^2 & x\,y & y^2 & x\,z & y\,z & z^2 \\ \hline 1 & 18 & 19 & 8 & 5 & 7 \\ & 1 & 3 & 2 & 4 & -1 \\ & & 1 & -11 & -3 & -5 \end{array}$$

“new” polynomials $f_4 = xy + 4\,yz + 2\,xz + 3\,y^2 - z^2$ and
$f_5 = y^2 - 11\,xz - 3\,yz - 5\,z^2$

Degree 3 (first try)

$$f_3 = x^2 + 18\,xy + 19\,y^2 + 8\,xz + 5\,yz + 7\,z^2$$
$$f_2 = 3\,x^2 + 7x\,y + 22\,x\,z + 11\,yz + 22\,z^2 + 8\,y^2$$
$$f_1 = 6\,x^2 + 12\,xy + 4\,y^2 + 14\,xz + 9\,yz + 7\,z^2$$
$$f_4 = xy + 4\,yz + 2\,xz + 3\,y^2 - z^2$$
$$f_5 = y^2 - 11\,xz - 3\,yz - 5\,z^2$$

and

$$f_2 \longrightarrow f_4$$
$$f_1 \longrightarrow f_5$$

Degree 3 (first try)

$$A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & \dots \\ 0 & 1 & 18 & 19 & 0 & \dots \\ 1 & 18 & 19 & 0 & 8 & \dots \\ 0 & 0 & 0 & 0 & 3 & \dots \\ 0 & 3 & 7 & 8 & 0 & \dots \\ 3 & 7 & 8 & 0 & 22 & \dots \\ 0 & 0 & 0 & 0 & 6 & \dots \\ 0 & 6 & 12 & 4 & 0 & \dots \\ 6 & 12 & 4 & 0 & 14 & \dots \end{pmatrix} \end{array}$$

Degree 3 (first try)

$$A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & \textcircled{1} & \dots \\ 0 & 1 & 18 & 19 & 0 & \dots \\ 1 & 18 & 19 & 0 & 8 & \dots \\ 0 & 0 & 0 & 0 & 3 & \dots \\ 0 & 3 & 7 & 8 & 0 & \dots \\ 3 & 7 & 8 & 0 & 22 & \dots \\ 0 & 0 & 0 & 0 & 6 & \dots \\ 0 & 6 & 12 & 4 & 0 & \dots \\ 6 & 12 & 4 & 0 & 14 & \dots \end{pmatrix} \end{array}$$

Degree 3 (first try)

$$A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\ \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & \textcircled{1} & \dots \\ 0 & 1 & 18 & 19 & 0 & \dots \\ 1 & 18 & 19 & 0 & 8 & \dots \\ 0 & 0 & 0 & 0 & \textcircled{3} & \dots \\ 0 & 3 & 7 & 8 & 0 & \dots \\ 3 & 7 & 8 & 0 & 22 & \dots \\ 0 & 0 & 0 & 0 & 6 & \dots \\ 0 & 6 & 12 & 4 & 0 & \dots \\ 6 & 12 & 4 & 0 & 14 & \dots \end{array}\right) \end{array}$$

Degree 3 (first try)

$$
A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array}
\begin{array}{c} \begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\
\left(\begin{array}{cccccc}
0 & 0 & 0 & 0 & \textcircled{1} & \dots \\
0 & 1 & 18 & 19 & 0 & \dots \\
1 & 18 & 19 & 0 & 8 & \dots \\
0 & 0 & 0 & 0 & \textcircled{3} & \dots \\
0 & 3 & 7 & 8 & 0 & \dots \\
3 & 7 & 8 & 0 & 22 & \dots \\
0 & 0 & 0 & 0 & 6 & \dots \\
0 & 6 & 12 & 4 & 0 & \dots \\
6 & 12 & 4 & 0 & 14 & \dots
\end{array}\right) \end{array}
$$

Degree 3 (first try)

$$
A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array}
\begin{array}{cccccc}
x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \\
\hline
0 & 0 & 0 & 0 & \textcircled{1} & \dots \\
0 & 1 & 18 & 19 & 0 & \dots \\
1 & 18 & 19 & 0 & 8 & \dots \\
0 & 0 & 0 & 0 & \textcircled{3} & \dots \\
0 & 3 & 7 & 8 & 0 & \dots \\
3 & 7 & 8 & 0 & 22 & \dots \\
0 & 0 & 0 & 0 & \textcircled{6} & \dots \\
0 & 6 & 12 & 4 & 0 & \dots \\
6 & 12 & 4 & 0 & 14 & \dots
\end{array}
$$

Degree 3 (first try)

$$
A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array}
\begin{array}{c}
\begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\
\left(\begin{array}{cccccc}
0 & 0 & 0 & 0 & \textcircled{1} & \dots \\
0 & 1 & 18 & 19 & 0 & \dots \\
1 & 18 & 19 & 0 & 8 & \dots \\
0 & 0 & 0 & 0 & \textcircled{3} & \dots \\
0 & 3 & 7 & 8 & 0 & \dots \\
3 & 7 & 8 & 0 & 22 & \dots \\
0 & 0 & 0 & 0 & \textcircled{6} & \dots \\
0 & 6 & 12 & 4 & 0 & \dots \\
6 & 12 & 4 & 0 & 14 & \dots
\end{array}\right)
\end{array}
$$

Degree 3 (first try)

Already
Done !

$f_2 \longrightarrow f_4$

$f_1 \longrightarrow f_5$

$A_3 :=$

	x^3	x^2y	xy^2	y^3	x^2z	$\dots$
zf_3	0	0	0	0	1	$\dots$
yf_3	0	1	18	19	0	$\dots$
xf_3	1	18	19	0	8	$\dots$
zf_2	0	0	0	0	3	$\dots$
yf_2	0	3	7	8	0	$\dots$
xf_2	3	7	8	0	22	$\dots$
zf_1	0	0	0	0	6	$\dots$
yf_1	0	6	12	4	0	$\dots$
xf_1	6	12	4	0	14	$\dots$

Degree 3

$$
A_3 := \begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_4 \\ yf_4 \\ xf_4 \\ zf_5 \\ yf_5 \\ xf_5 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & xyz & y^2z & xz^2 & yz^2 & z^3 \end{array} \\
\left(\begin{array}{cccccccccc}
 & & & & 1 & 18 & 19 & 8 & 5 & 7 \\
 & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\
1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\
 & & & & & 1 & 3 & 2 & 4 & 22 \\
 & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\
 & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 & 0 \\
 & & & & & & 1 & 12 & 20 & 18 \\
 & & & 1 & 0 & 12 & 20 & 0 & 18 & 0 \\
 & & 1 & 0 & 12 & 20 & 0 & 18 & 0 & 0
\end{array}\right)
\end{array}
$$

Degree 3

$$
A_3 := \begin{array}{c} \\ xf_3 \\ yf_3 \\ yf_4 \\ xf_2 \\ zf_3 \\ zf_4 \\ zf_5 \\ yf_5 \\ xf_5 \end{array}
\begin{array}{c} \begin{array}{cccccccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & xyz & y^2z & xz^2 & yz^2 & z^3 \end{array} \\
\begin{pmatrix}
1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\
 & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\
 & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\
 & & & 1 & 0 & 0 & 8 & 1 & 18 & 15 \\
 & & & & 1 & 18 & 19 & 8 & 5 & 7 \\
 & & & & & 1 & 3 & 2 & 4 & 22 \\
 & & & & & & 1 & 12 & 20 & 18 \\
 & & & & & & & 1 & 11 & 13 \\
 & & & & & & & & 1 & 18
\end{pmatrix} \end{array}
$$

Degree 3

Summary: we have constructed 3 new polynomials

$$
\begin{aligned}
f_6 &= y^3 + 8\,y^2 z + xz^2 + 18\,yz^2 + 15\,z^3 \\
f_7 &= xz^2 + 11\,yz^2 + 13\,z^3 \\
f_8 &= yz^2 + 18\,z^3
\end{aligned}
$$

And we have the linear equivalences:

$$
\begin{gathered}
x\,f_2 \leftrightarrow x\,f_4 \leftrightarrow f_6 \\
y\,f_1 \leftrightarrow f_7 \\
x\,f_1 \leftrightarrow f_8
\end{gathered}
$$

Degree 4

The matrix whose rows are

$$x^2 f_i, x\,y f_i, y^2 f_i, x\,z f_i, y\,z f_i, z^2 f_i,\ \ i = 1, 2, 3$$

is not full rank !

Why ? (1)

$6 \times 3 =$ 18 rows
$x^4, x^3\,y, \ldots, y\,z^3, z^4$ 15 columns

Why ? (1)

$6 \times 3 =$ 18 rows

$x^4, x^3 y, \ldots, y z^3, z^4$ 15 columns

Simple linear algebra theorem: 3 useless row (but which ones ?)

Trivial relations

$$f_2\, f_3 - f_3\, f_2 = 0$$

can be rewritten

$$\begin{aligned}&3\,x^2\,f_3 + (7 + \ b)\,xy\,f_3 + 8\,y^2\,f_3 + 22\,xz\,f_3\\&+11\,yz\,f_3 + 22\,z^2\,f_3 - \boxed{x^2\,f_2} - 18\,xy\,f_2 - 19\,y^2\,f_2\\&-8\,xz\,f_2 - 5\,yz\,f_2 - 7\,z^2\,f_2 = 0\end{aligned}$$

We can remove the row $x^2 f_2$

same way $f_1 f_3 - f_3 f_1 = 0 \longrightarrow$ remove $x^2 f_1$

but $f_1 f_2 - f_2 f_1 = 0 \longrightarrow$ remove $x^2 f_1$! ???

Combining trivial relations

$$
\begin{aligned}
0 &= (f_2 f_1 - f_1 f_2) - 3(f_3 f_1 - f_1 f_3) \\
0 &= (f_2 - 3f_3) f_1 - f_1 f_2 + 3 f_1 f_3 \\
0 &= f_4 f_1 - f_1 f_2 + 3 f_1 f_3 \\
0 &= \left((1 - b)xy + 4\,yz + 2\,xz + 3\,y^2 - z^2\right) f_1 \\
&\quad -(6x^2 + \cdots) f_2 + 3(6x^2 + \cdots) f_3
\end{aligned}
$$

- if $b \neq 1$ remove $x\,y\ f_1$
- if $b = 1$ remove $y\,z\ f_1$

Need “some” computation

Degree 4 I

$$y^2 f_1, x\,z f_1, y\,z f_1, z^2 f_1, x\,y f_2, y^2 f_2, x\,z f_2,$$
$$y\,z f_2, z^2 f_2, x^2 f_3, x\,y f_3, y^2 f_3, x\,z f_3, y\,z f_3, z^2 f_3$$

In order to use previous computations (degree 2 and 3):

$$\begin{array}{cc} x f_2 \to f_6 & f_2 \to f_4 \\ x f_1 \to f_8 & y f_1 \to f_7 \\ f_1 \to f_5 & \end{array}$$

$$y f_7, z f_8, z f_7, z^2 f_5, y f_6, y^2 f_4, z f_6, y\,z f_4,$$
$$z^2 f_4, x^2 f_3, x\,y f_3, y^2 f_3, x\,z f_3, y\,z f_3, z^2 f_3,$$

Degree 4 II

$$A_4 := \begin{bmatrix}
1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\
 & & 1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 \\
 & & & 1 & 3 & 0 & 0 & 2 & 4 & 0 & 0 & 22 & 0 & 0 & 0 \\
 & & & & 1 & 0 & 0 & 0 & 8 & 0 & 1 & 18 & 0 & 15 & 0 \\
 & & & & & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\
 & & & & & & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\
 & & & & & & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\
 & & & & & & & & 1 & 0 & 0 & 8 & 1 & 18 & 15 \\
 & & & & & & & & & 1 & 18 & 19 & 8 & 5 & 7 \\
 & & & & & & & & & & 1 & 11 & 0 & 13 & 0 \\
 & & & & & & & & & & & 1 & 12 & 20 & 18 \\
 & & & & & & & & & & & & 1 & 11 & 13 \\
 & & & & & & & & & & & & & 1 & 18 \\
 & & & & & & & & & & 1 & 3 & 2 & 4 & 22
\end{bmatrix}$$

Degree 4 III

$$A_4 := \left[\begin{array}{cccccccccc|ccccc}
1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\
 & & 1 & 18 & 19 & 0 & 0 & 8 & 5 & 0 & 0 & 7 & 0 & 0 & 0 \\
 & & & 1 & 3 & 0 & 0 & 2 & 4 & 0 & 0 & 22 & 0 & 0 & 0 \\
 & & & & 1 & 0 & 0 & 0 & 8 & 0 & 1 & 18 & 0 & 15 & 0 \\
 & & & & & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\
 & & & & & & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\
 & & & & & & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\
 & & & & & & & & 1 & 0 & 0 & 8 & 1 & 18 & 15 \\
 & & & & & & & & & 1 & 18 & 19 & 8 & 5 & 7 \\
\cline{11-15}
 & & & & & & & & & & 1 & 11 & 0 & 13 & 0 \\
 & & & & & & & & & & & 1 & 12 & 20 & 18 \\
 & & & & & & & & & & & & 1 & 11 & 13 \\
 & & & & & & & & & & & & & 1 & 18 \\
 & & & & & & & & & & 1 & 3 & 2 & 4 & 22
\end{array}\right]$$

We need to consider only a small sub matrix:

$$A'_4 := \begin{array}{c} \\ yf_7 \\ z^2 f_5 \\ zf_7 \\ zf_8 \\ z^2 f_4 \end{array} \begin{array}{c} \begin{array}{ccccc} xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 \end{array} \\ \begin{pmatrix} 1 & 11 & 0 & 13 & 0 \\ & 1 & 12 & 20 & 18 \\ & & 1 & 11 & 13 \\ & & & 1 & 18 \\ 1 & 3 & 2 & 4 & 22 \end{pmatrix} \end{array}$$

F5 Criterion : analysis

Example: compute a Gröbner basis of $[f_1, f_2, f_3]$

Any combination of the trivial relations $f_i f_j = f_j f_i$ can always be written:

$$u(f_2 f_1 - f_1 f_2) + v(f_3 f_1 - f_1 f_3) + w(f_2 f_3 - f_3 f_2) = 0$$

where u, v, w are arbitrary polynomials.

F5 Criterion : analysis

Example: compute a Gröbner basis of $[f_1, f_2, f_3]$
Any combination of the trivial relations $f_i f_j = f_j f_i$ can always be written:

$$u(f_2 f_1 - f_1 f_2) + v(f_3 f_1 - f_1 f_3) + w(f_2 f_3 - f_3 f_2) = 0$$

where u, v, w are arbitrary polynomials.

$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 + u\, f_2 f_1 - u\, f_1 f_2 + v\, f_3 f_1 - w f_3 f_2 = 0$$

$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 \longrightarrow 0$$

F5 Criterion : analysis

Example: compute a Gröbner basis of $[f_1, f_2, f_3]$

Any combination of the trivial relations $f_i f_j = f_j f_i$ can always be written:

$$u(f_2 f_1 - f_1 f_2) + v(f_3 f_1 - f_1 f_3) + w(f_2 f_3 - f_3 f_2) = 0$$

where u, v, w are arbitrary polynomials.

$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 + u\, f_2 f_1 - u\, f_1 f_2 + v\, f_3 f_1 - w f_3 f_2 = 0$$

$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 \longrightarrow 0$$

(trivial) relation $h\, f_3 + \cdots = 0 \leftrightarrow \; h \in \mathrm{Id}(f_1, f_2)$

F5 Criterion : analysis

Example: compute a Gröbner basis of $[f_1, f_2, f_3]$
Any combination of the trivial relations $f_i f_j = f_j f_i$ can always be written:

$$u(f_2 f_1 - f_1 f_2) + v(f_3 f_1 - f_1 f_3) + w(f_2 f_3 - f_3 f_2) = 0$$

where u, v, w are arbitrary polynomials.

$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 + u\, f_2 f_1 - u\, f_1 f_2 + v\, f_3 f_1 - w f_3 f_2 = 0$$
$$\boxed{(w\, f_2 - v\, f_1)}\, f_3 \longrightarrow 0$$

(trivial) relation $h\, f_3 + \cdots = 0 \leftrightarrow \; h \in \mathrm{Id}(f_1, f_2)$

F_5 Criterion: compute a Gröbner basis $G_2$of $Id(f_1, f_2)$.

Remove row $t\, f_3$ iff t reducible by $\mathrm{LT}(G_2)$

Keep row $t\, f_3$ iff t not reducible by $\mathrm{LT}(G_2)$

F_5 *algorithm*

- Incremental algorithm

$$(f_3) + G_{\text{prev}}$$

- Incremental degree by degree

Special/Simpler version of F_5 for dense/generic quadratic polynomials.
The maximal degree D is a *parameter* of the algorithm.

$$
\begin{array}{l}
 \\
u_1 f_1 \\
\vdots \\
u_{r_1} f_1 \\
\vdots \\
v_{r_{k-1}} f_{k-1} \\
w_1 f_k \\
w_2 f_k
\end{array}
\quad
\begin{array}{cccccc}
m_1 & m_2 & m_3 & m_4 & m_5 & \ldots
\end{array}
\atop
\begin{pmatrix}
1 & x & x & x & x & \ldots \\
0 & \ddots & x & x & x & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
0 & 0 & 0 & 1 & x & \ldots \\
0 & 0 & 0 & 0 & 1 & \ldots
\end{pmatrix}
$$

F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d + 1)$

Already computed
Groebner $(\langle f_1, \ldots, f_k \rangle), d)$

Matrix in degree d

$$
\begin{array}{c}
 \\ u_1 f_1 \\ \vdots \\ u_{r_1} f_1 \\ \vdots \\ v_{r_{k-1}} f_{k-1} \\ w_1 f_k \\ w_2 f_k
\end{array}
\begin{array}{c}
\begin{array}{cccccc} m_1 & m_2 & m_3 & m_4 & m_5 & \ldots \end{array} \\
\left(\begin{array}{cccccc}
1 & x & x & x & x & \ldots \\
 & \ddots & & & & \\
0 & & x & x & x & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
0 & 0 & 0 & 1 & x & \ldots \\
0 & 0 & 0 & 0 & 1 & \ldots
\end{array}\right)
\end{array}
$$

F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d+1)$

Matrix in degree d

$$
\begin{array}{l}
\\
u_1 f_1 \\
\vdots \\
u_{r_1} f_1 \\
\vdots \\
v_{r_{k-1}} f_{k-1} \\
w_1 f_k \\
w_2 f_k
\end{array}
\quad
\begin{array}{c}
\begin{array}{cccccc} m_1 & m_2 & m_3 & m_4 & m_5 & \ldots \end{array} \\
\begin{pmatrix}
1 & x & x & x & x & \ldots \\
0 & \ddots & x & x & x & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
0 & 0 & 0 & 1 & x & \ldots \\
0 & 0 & 0 & 0 & 1 & \ldots
\end{pmatrix}
\end{array}
$$

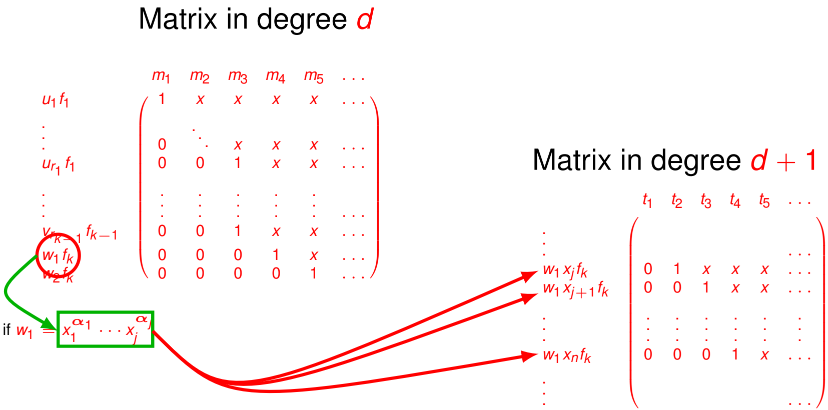
F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d+1)$

Matrix in degree d

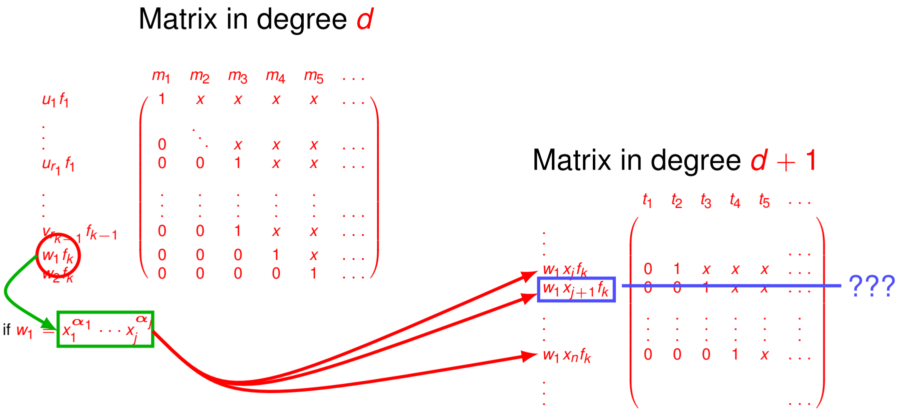
$$
\begin{array}{l}
u_1 f_1 \\
\vdots \\
u_{r_1} f_1 \\
\vdots \\
v_{r_{k-1}} f_{k-1} \\
w_1 f_k \\
w_2 f_k
\end{array}
\quad
\begin{array}{c}
\begin{array}{cccccc} m_1 & m_2 & m_3 & m_4 & m_5 & \ldots \end{array} \\
\left(\begin{array}{cccccc}
1 & x & x & x & x & \ldots \\
 & \ddots & & & & \\
0 & & x & x & x & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\
0 & 0 & 1 & x & x & \ldots \\
0 & 0 & 0 & 1 & x & \ldots \\
0 & 0 & 0 & 0 & 1 & \ldots
\end{array}\right)
\end{array}
$$

if $w_1 = x_1^{\alpha_1} \ldots x_j^{\alpha_j}$

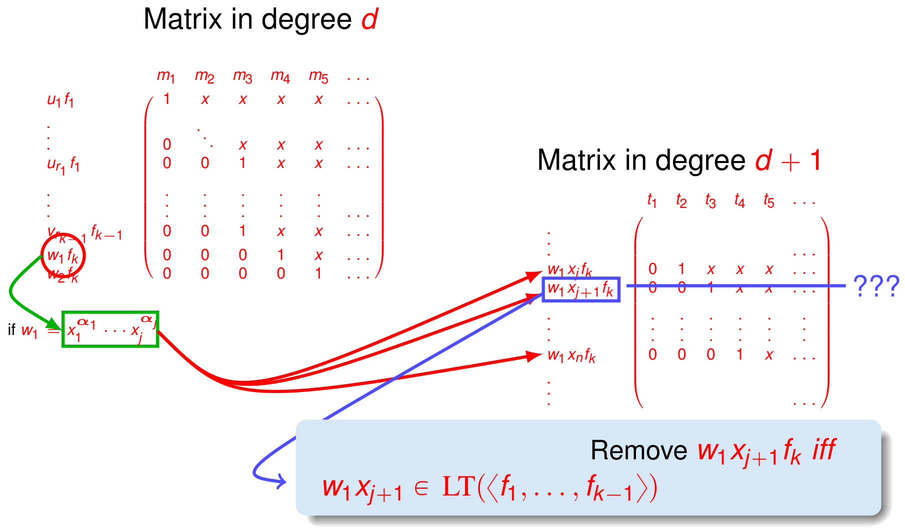
F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d + 1)$



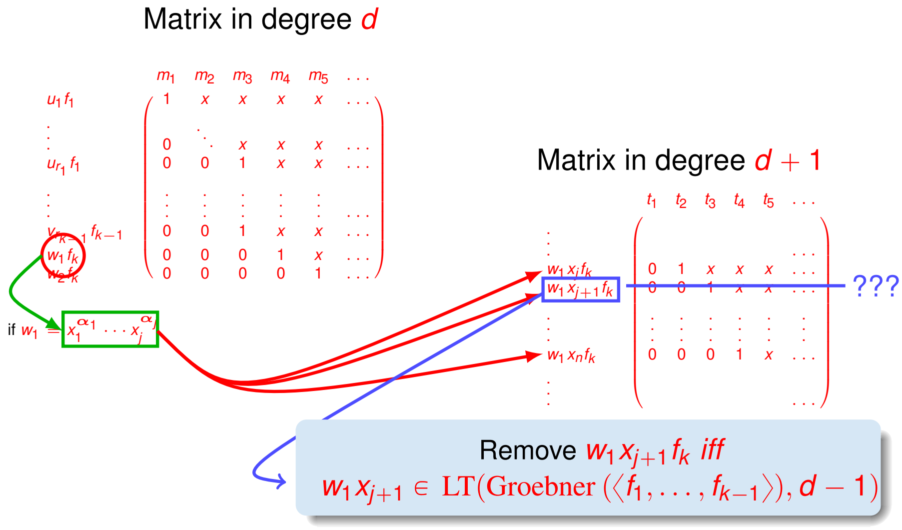
F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d+1)$



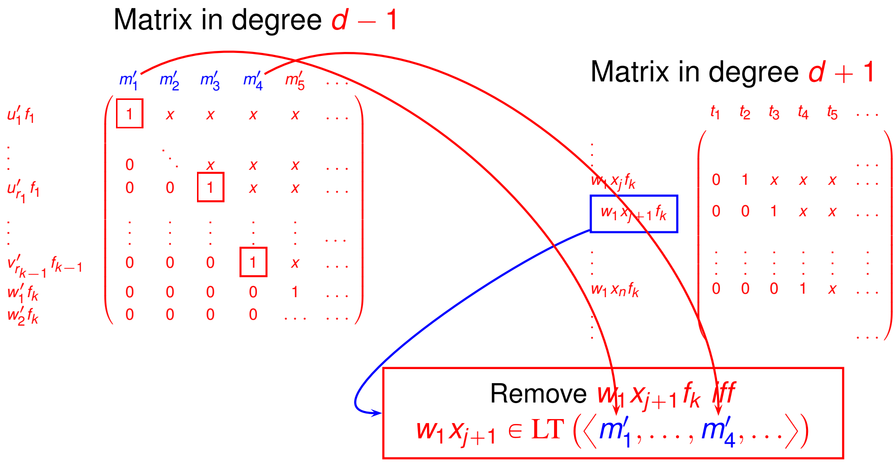
F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d+1)$



F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d+1)$



(Final) F5: compute Groebner $(\langle f_1, \ldots, f_k \rangle), d + 1)$



matrix F_5 algorithm

Algorithm matrix F_5 algorithm

Input: $\begin{cases} \text{coefficient field } \mathbb{K} \neq \mathbb{F}_2 \\ F = [f_1, \ldots, f_m] \text{ polynomials; total degree } d_1 \leqslant \cdots \leqslant d_m, \\ \text{integer } D > 0 \end{cases}$

Output: a D-Gröbner basis of F wrt an admissible ordering $<$.

$M^{(*)}([]) := \emptyset$, $\widetilde{M^{(*)}}([]) := \emptyset$

for d **from** d_1 **to** D **do** *// Degree loop*

  for i **from** 1 **to** m **do** *// Equation loop*

    // Build a new matrix $M^{(d)}([f_1, \ldots, f_i])$:

    if $d = d_i$ **then**

$$M^{(d)}([f_1, \ldots, f_i]) := \begin{array}{c|c|c} & \widetilde{M^{(d)}}([f_1, \ldots, f_{i-1}]) & \\ f_i & \cdots & f_i \end{array}$$

    else

$$M^{(d)}([f_1, \ldots, f_i]) := \widetilde{M^{(d)}}([f_1, \ldots, f_{i-1}])$$

      ...

Algorithm matrix F_5 algorithm

else

$M^{(d)}([f_1,\ldots,f_i]) := \widetilde{M^{(d)}}([f_1,\ldots,f_{i-1}])$

// $J_{\text{Criterion}}$

$J_{\text{Criterion}} := \mathrm{Id}\left(\mathrm{LT}\left(\widetilde{M^{(d-d_i)}}([f_1,\ldots,f_{i-1}])\right)\right)$

for each row f whose label is $t\,f_i$ in $\widetilde{M^{(d-1)}}([f_1,\ldots,f_i])$ **do**

Let k the greatest integer s.t. x_k divides t

for j **from** k **to** n **do**

if $t\,x_j \notin J_{\text{Criterion}}$ **then**

$$M^{(d)}([f_1,\ldots,f_i]) := \begin{array}{c|c|c} & \widetilde{M^{(d)}}([f_1,\ldots,f_i]) & \\ tx_jf_i & \cdots & x_jf \end{array}$$

Compute $\widetilde{M^{(d)}}([f_1,\ldots,f_i])$ Gaussian reduction

Keep the same order for the labels).

return Polynomial representation of $\widetilde{M^{(D)}}([f_1,\ldots,f_m])$

Properties of F_5

There is a full version of the algorithm F_5 : D the maximal degree is no more a parameter

Theorem

If $F = [f_1, \ldots, f_m]$ *is a regular sequence, then all the matrices generated by the algorithm have full rank.*

- Easy to adapt for special cases $\mathbb{F}_2$ (*new trivial relation:* $f_i^2 = f_i$).
- We can swap the two loops: degree first and the equation by equation
- matrix F_5 is very easy to implement: for instance HFE Challenge 1 broken