MMA 数学特論 I

Algorithms for polynomial systems: elimination & Gröbner bases

多項式系のアルゴリズム: グレブナー基底 & 消去法

Lecture V: The Buchberger Algorithm

June, 3rd 2010.	Part I: S-polynomials
	Part II: The algorithm
June 10th.	Part III: Syzygies

Xavier Dahan

Part I: S-polynomials

Introduction

Gröbner bases **exist** $\rightarrow$ Dickson Lemma + Hilbert finite basis (Lect. IV)

Gröbner bases are **useful** $\rightarrow$ Ideal membership (Theo. 4), and several other applications (next lectures).

Part I: S-polynomials — Introduction

Gröbner bases **exist** $\rightarrow$ Dickson Lemma + Hilbert finite basis (Lect. IV)

Gröbner bases are **useful** $\rightarrow$ Ideal membership (Theo. 4), and several other applications (next lectures).

Let $F = \{f_1, f_2, \ldots, f_s\}$ polynomial system in $\Bbbk[X_1, \ldots, X_n]$, and let $I = \langle f_1, \ldots, f_s \rangle$ the ideal it generates.

Problem 1: Is F a Gröbner basis for I (w.r.t. to a monomial order $\prec$) ?

Problem 2: If not, how to compute a Gröbner basis for I, starting from F ?

Problem 3: Is it easy to compute a Gröbner basis ? (**efficiency**)

Part I: S-polynomials

Gröbner bases **exist** $\longrightarrow$ Dickson Lemma + Hilbert finite basis (Lect. IV)

Gröbner bases are **useful** $\longrightarrow$ Ideal membership (Theo. 4), and several other applications (next lectures).

Let $F = \{f_1, f_2, \ldots, f_s\}$ polynomial system in $\Bbbk[X_1, \ldots, X_n]$, and let $I = \langle f_1, \ldots, f_s \rangle$ the ideal it generates.

Problem 1: Is F a Gröbner basis for I (w.r.t. to a monomial order $\prec$) ?

Problem 2: If not, how to compute a Gröbner basis for I, starting from F ?

$\longrightarrow$ Answer: use "S-polynomials" and Buchberger's criterion.

Problem 3: Is it easy to compute a Gröbner basis ? (**efficiency**)

Part I: S-polynomials

Introduction

Gröbner bases **exist** $\longrightarrow$ Dickson Lemma + Hilbert finite basis (Lect. IV)

Gröbner bases are **useful** $\longrightarrow$ Ideal membership (Theo. 4), and several other applications (next lectures).

Let $F = \{f_1, f_2, \ldots, f_s\}$ polynomial system in $\mathbb{k}[X_1, \ldots, X_n]$, and let $I = \langle f_1, \ldots, f_s \rangle$ the ideal it generates.

Problem 1: Is F a Gröbner basis for I (w.r.t. to a monomial order $\prec$) ?

Problem 2: If not, how to compute a Gröbner basis for I, starting from F ?

$\longrightarrow$ Answer: use "S-polynomials" and Buchberger's criterion.

Problem 3: Is it easy to compute a Gröbner bais ? (**efficiency**)

$\longrightarrow$ Answer: Very hard. Many improvements possible $\longrightarrow$ still active research topic.

The problem

Let $F = \{f_1, \ldots, f_s\}$ be a finite set of polynomials, $\prec$ a monomial order.

If F is not a Gröbner basis for $I = \langle F \rangle$, then:

$$\exists f \in I, \text{ but } \textsc{lm}(f) \notin \langle \textsc{lm}(F) \rangle \ (\Leftrightarrow \textsc{lm}(f_i) \nmid \textsc{lm}(f), \ \forall\, i).$$

$\rightarrow$ $\textsc{lm}(F)$ is "too small" for being a Gröbner basis ($\Leftrightarrow \langle \textsc{lm}(F) \rangle \subsetneq \langle \textsc{lm}(I) \rangle$).

$\rightarrow$ *(graphic of the example on Slide 5, Lect. IV on the blackboard. . .)*

The problem

Let $F = \{f_1, \ldots, f_s\}$ be a finite set of polynomials, $\prec$ a monomial order.

If F is not a Gröbner basis for $I = \langle F \rangle$, then:

$$\exists f \in I, \text{ but } \textsc{lm}(f) \notin \langle \textsc{lm}(F) \rangle \; (\Leftrightarrow \textsc{lm}(f_i) \nmid \textsc{lm}(f), \; \forall\, i).$$

$\to$ $\textsc{lm}(F)$ is "too small" for being a Gröbner basis ($\Leftrightarrow \langle \textsc{lm}(F) \rangle \subsetneq \langle \textsc{lm}(I) \rangle$).

$\to$ *(graphic of the example on Slide 5, Lect. IV on the blackboard. . .)*

How to extend $\textsc{lm}(F)$? (Try to) find $f \in I$, such that $\textsc{lm}(f) \notin \langle \textsc{lm}(F) \rangle$.

$$\Longrightarrow f = \sum_{i=1}^{s} h_i f_i \;\text{ such that }\; \boxed{\textsc{lm}(f) = \textsc{lm}\Big(\sum_{i=1}^{s} f_i h_i\Big) \prec \max_{1 \le i \le s} \textsc{lm}(f_i h_i)} \; (\star)$$

Remember that. . . $\textsc{lm}_\prec(a_1 + a_2) \preccurlyeq \max\{\textsc{lm}_\prec(a_1), \textsc{lm}_\prec(a_2)\}$, with equality if $\textsc{lm}(a_1) \neq \textsc{lm}(a_2)$ . . . and that $\textsc{lm}(f) \prec \textsc{lm}(f_i) \Rightarrow \textsc{lm}(f_i) \nmid \textsc{lm}(f)$.

Conclusion: There is a term cancellation idenitity in $(\star)$.

S-polynomials

Definition 1 *Given two non-zero polynomials $f, g \in \Bbbk[X_1, \ldots, X_n]$, and a monomial order $\prec$, let $X^\alpha = \mathrm{LM}_\prec(f)$, and $X^\beta = \mathrm{LM}_\prec(g)$.*

The least common multiple *of X^α and X^β is X^γ, where $\gamma = (\max\{\alpha_1, \beta_1\}, \ldots, \max\{\alpha_n, \beta_n\})$, denoted* $\mathrm{LCM}(\mathrm{LM}_\prec(f), \mathrm{LM}_\prec(g)) = X^\gamma$.

The polynomial $S_\prec(f, g) := \frac{X^\gamma}{\mathrm{LT}_\prec(f)} f - \frac{X^\gamma}{\mathrm{LT}_\prec(g)} g$*, is called the* S-polynomial *of f and g (if it is clear what is $\prec$, we use simply $S(f, g)$ instead of $S_\prec(f, g)$).*

S-polynomials

Definition 1 *Given two non-zero polynomials $f, g \in \Bbbk[X_1, \ldots, X_n]$, and a monomial order $\prec$, let $X^\alpha = \mathrm{LM}_\prec(f)$, and $X^\beta = \mathrm{LM}_\prec(g)$.*

The least common multiple *of X^α and X^β is X^γ, where $\gamma = (\max\{\alpha_1, \beta_1\}, \ldots, \max\{\alpha_n, \beta_n\})$, denoted* $\mathrm{LCM}(\mathrm{LM}_\prec(f), \mathrm{LM}_\prec(g)) = X^\gamma$.

The polynomial $S_\prec(f, g) := \frac{X^\gamma}{\mathrm{LT}_\prec(f)} f - \frac{X^\gamma}{\mathrm{LT}_\prec(g)} g$ *, is called the* S-polynomial *of f and g (if it is clear what is $\prec$, we use simply $S(f, g)$ instead of $S_\prec(f, g)$).*

Comment: The S-polynomials control the "term cancellation identities":

Proposition 1 *Let $T = \sum_{i=1}^{s} c_i f_i$, with $c_i \in \Bbbk$, and $\delta = \mathsf{mdeg}_\prec(f_i)$ for all i. If* $\mathsf{mdeg}_\prec(T) \prec \delta$ *, then there exists $c_{j,k} \in \Bbbk$ such that*

$$T = \sum_{1 \leq j,k \leq s} c_{j,k} S_\prec(f_j, f_k).$$

Moreover $\mathsf{mdeg}_\prec(S_\prec(f_j, f_k)) \prec \delta$.

Proof: *(On the balckboard. . .)* $\square$

Main theorem: Buchberger's criterion

The previous Proposition 1 is important for the following criterion (Theo. 1). Before, a definition...Let $G = \{g_1, \dots, g_s\}$ be a polynomial system and $\prec$ a monomial order:

Definition 2 *A polynomial f is said to* reduce to 0 modulo G, *denoted* $f \rightarrow_G 0$ *if there exists $a_1, \dots, a_s \in \Bbbk[X_1, \dots, X_n]$ such that:*

$$f = a_1 g_1 + \cdots + a_s g_s, \ \textit{with}\ \mathrm{LM}(a_i g_i) \preccurlyeq \mathrm{LM}(f)\ \textit{if}\ a_i \neq 0.$$

Remark: If there exists (at least) one permutation $\sigma \in \mathfrak{S}_s$, such that:

$$\mathrm{NF}_{\prec}(f, [g_{\sigma(1)}, g_{\sigma(2)}, \dots, g_{\sigma(s)}]) = 0,$$

then $f \rightarrow_G 0$, but the reciprocal is not true in general.

Theorem 1 *G is a Gröbner basis of $\langle G \rangle$, iff for all pairs $i \neq j$,* $S(g_i, g_j) \rightarrow_G 0$.

Proof: *(On the blackboard...)*

Is a polynomial system a Gröbner basis ?

This is the problem 1 of Introduction.

The Buchberger criterion (Theorem 1), implies this algorithm to decide if a polynomial system F is a Gröbner basis or not.

`# Inputs:` A polynomial system $F = \{f_1, \ldots, f_s\}$. A monomial order $\prec$.
`# Output:` true if F is a Gröbner basis for $\langle F \rangle$, false else.

`for` $p, q \in F, p \neq q$ `do`
  `if` $\mathrm{NF}_{\prec}(S_{\prec}(p, q), F) \neq 0$ `then return` false ; `end if`
`end for`
`return` true

Remark: It is just to show the power of S-polynomials. Else, it is very inefficient in practice, and not very useful.

Part II: The Algorithm **Version 1**

Version 1: very naive and slow.

`# Inputs:` Non-zero polynomial system $F = \{f_1, \ldots, f_s\}$. A monomial order $\prec$.

`# Output:` A Gröbner basis $G = \{g_1, \ldots, g_t\}$ for $\langle F \rangle$, w.r.t. $\prec$.

1: $G \leftarrow F$
2: `do`$\{$ $G' \leftarrow G$
3:   `for` $p, q \in G', p \neq q$ `do`
4:     $S \leftarrow \mathrm{NF}(S(p,q), G')$ // computed for any sequence order of G'
5:     `if` $S \neq 0$ `then` $G \leftarrow G \cup \{S\}$; `end if`
6:   `end for`
7: $\}$ `until` $(G = G')$ // repeat from Step 2
8: `return` G

Correctness - Termination

Correctness: Claim 1: we always have $F \subset G \subset I$ *(proof on the blackboard...)*

So, if $\langle F \rangle = I$, then $\langle G \rangle = I$.

Claim 2: When $G = G'$ ($\Leftrightarrow$ exit the `do/until` loop $\Leftrightarrow$ end of algorithm), we have $S = \text{NF}(S(p,q), G') = 0$ for all $p \neq q$ in G. By Buchberger's criterion (Theo. 1), G is a Gröbner basis.

Termination: If $\text{LM}(G') = \text{LM}(G)$ then $G = G'$.
We have $\langle \text{LM}(G') \rangle \subset \langle \text{LM}(G) \rangle$, so the sequence $\{\text{LM}(G')\}$ verifies the "ascending chain condition" (Definition 4, Lect. IV), in $\mathbb{k}[X_1, \ldots, X_n]$.
Because it is Nœtherian (Lect. IV, Theo. 3), after a finite number of steps, we have $\text{LM}(G) = \text{LM}(G')$.

Efficiency: detect useless S-polynomial

! Computing a division (or normal form) can be slow: the size of the numbers can grow a lot.

!! If $S(p, q)$ reduces to 0 modulo G, then nothing happens in the algorithm !

$\rightarrow$ computing the division of $S(p, q)$ that gives a 0 remainder is useless .

$\Rightarrow$ Need to decrease as much as possible the number of divisions of S-polynomials computed at Step 4 of the Algo. version 1 (Slide 7)

Efficiency: detect useless S-polynomial

! Computing a division (or normal form) can be slow: the size of the numbers can grow a lot.

!! If $S(p,q)$ reduces to 0 modulo G, then nothing happens in the algorithm !

$\rightarrow$ computing the division of $S(p,q)$ that gives a 0 remainder is useless .

$\Rightarrow$ Need to decrease as much as possible the number of divisions of S-polynomials computed at Step 4 of the Algo. version 1 (Slide 7)

Unnecessary pairs (1): Since $S(p,q) = -S(q,p)$: pair (p,q) already tested $\Rightarrow$ need not consider the pair (q,p) (see definition of set B at Step 1, next slide).

Unnecessary pairs (2): If $S(p,q) \rightarrow_{G'} 0$, then $S(p,q) \rightarrow_{G' \cup \{S(a,b)\}} 0$ for any S-polynomial of $a, b \in G'$.

$\rightarrow$ Hence, the pair (p,q) needs not to be kept in the set B of all indices of pairs to be tested (see Step 10, next slide).

Buchberger algorithm: Version 2

`# Inputs:` A polynomial system $F = \{f_1, \dots, f_s\}$
`# Output:` A Gröbner basis $G = \{g_1, \dots, g_t\}$ for $I = \langle F \rangle$.

$G \leftarrow F; t \leftarrow s$
$B \leftarrow \{(i,j)\,,\ 1 \leq i < j \leq s\}$ // indices of pairs f_i, f_j to be tested
`while` $B \neq \emptyset$ `do`
  `for` $(i,j) \in B$ `do`
    $S \leftarrow \mathrm{NF}(S(f_j, f_i), G)$
    `if` $S \neq 0$ `then` // the S-pol. has not a 0 remainder
      $t \leftarrow t+1\ ;\ f_t \leftarrow S$
      $G \leftarrow G \cup \{f_t\}$ // then we add this remainder to G...
      $B \leftarrow B \cup \{(i,t)\,,\ 1 \leq i \leq t-1\}$ // and the new indices.
    `else` $B \leftarrow B - \{(i,j)\}$; `end if` // else the pair of index i, j...
`end for ; end while` ...will allways reduced to 0
`return` G

Another criterion to detect useless pairs

This Proposition 2 permits to detect some pairs of polynomials p, q such that $S(p, q)$ will reduce to 0 modulo G.

$\rightarrow$ permits to avoid useless computations (see Slide 14).

Proposition 2 *Let G be finite set of polynomials. For a pair $f, g \in G$ and a monomial order $\prec$, if* $\text{LCM}(\text{LM}_\prec(f), \text{LM}_\prec(g)) = \text{LM}_\prec(f)\text{LM}_\prec(g)$*, then* $S_\prec(f, g) \rightarrow_G 0$.

PROOF: *(On the blackboard…)*

Application: This criterion is easy to check. (comparing to do a division).

Buchberger: Version 2.1

`# Inputs:` A polynomial system $F = \{f_1, \ldots, f_s\}$

`# Output:` A Gröbner basis $G = \{g_1, \ldots, g_t\}$ for $I = \langle F \rangle$.

$G \leftarrow F;\ t \leftarrow s$

$B \leftarrow \{(i,j)\ ,\ 1 \le i < j \le s\}$ // indices of pairs f_i, f_j to be tested

while $B \neq \emptyset$ **do**

  for $(i,j) \in B$ **do**

    if $\mathrm{LCM}(\mathrm{LM}(f_i), \mathrm{LM}(f_j)) \neq \mathrm{LM}(f_i)\mathrm{LM}(f_j)$ **then**

      $S \leftarrow \mathrm{NF}(S(f_j, f_i), G)$

      if $S \neq 0$ **then** // the S-pol. has not a 0 remainder

        $t \leftarrow t+1\ ;\ f_t \leftarrow S$

        $G \leftarrow G \cup \{f_t\}$ // then we add this remainder to G...

        $B \leftarrow B \cup \{(i,t)\ ,\ 1 \le i \le t-1\}$ // and the the new indices

      else $B \leftarrow B - \{(i,j)\}$; **end if** // else the pair of index i, j...

    end if

end for ; **end while** ...will allways reduced to 0

return G

Part III: Syzygies

Module over a ring

Let R be a commutative ring with 1_R for unit element, with addition $+$ and multiplication $\cdot$.

An abelian group $(M, +)$ is an **R-module** if, there is a map:
$R \times M \to M,\ (r, m) \mapsto rm$, such that:

- $1_R m = m$
- $(r \cdot r')m = r(r'm) = r(r'm)$
- $(r + r')m = rm + r'm$
- $r(m + m') = rm + rm'$

Facts: If R is a field then R-modules are the vector spaces over R.

If R is not a field, then a module M **has no base** in general.

An R-module M is finitely generated if there exists some elements $m_1, \ldots, m_s$ in M such that $\forall m \in M$, $\exists r_1, \ldots, r_s$ elements in R with: $m = r_1 m1 + \cdots + r_s m_s$.

Examples: Let $I \subset R$ be an ideal. The quotient ring R/I is an R-module...

Syzygy (1/3)

Given an R-module M, the *first syzygy module* or the *syzygies* of M on a set of generators $(m_1, \ldots, m_s)$ is the kernel the following presentation of M:

$$\begin{array}{ccc} R^s & \xrightarrow{\times(m_1,\ldots,m_s)} & M \to 0, \\ (r_1, \ldots, r_s) & \longmapsto & r_1 m_1 + \cdots + r_s m_s. \end{array}$$

then $Syz(m_1, \ldots, m_s) := \{(r_1, \ldots, r_s) \in R^s \mid \sum_i a_i m_i = 0\}$, so that $M \simeq R^s / Syz(m_1, \ldots, m_s)$.

Definition 3 *Let $F = (f_1, \ldots, f_s)$ a family of s polynomials. We simply denoted by $Syz(F)$ the* syzygies on the leading terms *of F:*

$$Syz(\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)) := \{(h_1, \ldots, h_s) \in \Bbbk[X_1, \ldots, X_n]^s \mid \sum_i h_i \mathrm{LT}(f_i) = 0\}.$$

Syzygy (2/3)

Homogeneous syzygy in $Syz(F)$ of (multi)degree $\alpha \in \mathbb{N}^n$:

$$(c_1 X^{\alpha(1)}, \ldots, c_s X^{\alpha(s)}), \text{ where } c_i \neq 0 \Rightarrow X^{\alpha(i)}\mathrm{LM}(f_i) = X^{\alpha}.$$

Lemma 1 *Every syzygy of $Syz(F)$ can be written uniquely as a linear combination over $\Bbbk$ of homogeneous syzygies.*

Proof: *(On the blackboard...)*

Proposition 3 *Let $F = (f_1, \ldots, f_s)$ be a family of polynomials, and $Syz(F)$ be the syzygy module on the leading terms of F. For $1 \leq i < j \leq s$, consider the pair f_i, f_j of F, and let $X^\gamma := \mathrm{LCM}(\mathrm{LM}(f_i), \mathrm{LM}(f_j))$. Define* $\mathbf{e}_1 = (1, 0, \ldots, 0)$, $\mathbf{e}_2 = (0, 1, 0, \ldots)$, $\ldots$, $\mathbf{e}_r = (\ldots, 0, 1)$ *and*

$$S_{ij} := \frac{X^\gamma}{\mathrm{LT}(f_i)}\mathbf{e}_i - \frac{X^\gamma}{\mathrm{LT}(f_j)}\mathbf{e}_j \in (\Bbbk[X_1, \ldots, X_n])^r,$$

The syzygies $\{S_{ij}\}_{1 \leq i,j \leq s}$ generate $Syz(F)$ as a $\Bbbk[X_1, \ldots, X_n]$-module.

Syzygy (3/3)

PROOF:First we must check that S_{ij} are effectively syzygies on the leading terms of F (easy).

Next, we must show that each syzygy $S \in Syz(F)$ can be written:

$$S = \sum_{i<j} p_{ij} S_{ij}, \quad p_{ij} \in \mathbb{k}[X_1, \ldots, X_n]$$

By Lemma 1 of the previous slide, we can assume that S is homogeneous of (multi)degree α. A syzygy $S \in Syz(F)$ *must* have at least two non-zero components, say $c_i X^{\alpha(i)}$ and $c_j X^{\alpha(j)}$ with $i < j$. By definition, we have $X^{\alpha(i)}\mathrm{LM}(f_i) = X^{\alpha(j)}\mathrm{LM}(f_j) = X^\alpha$, so $X^\gamma | X^\alpha$.

Claim: $S - c_i \mathrm{LC}(f_i) X^{\alpha-\gamma} S_{ij}$ has its i-th component equal to zero, so has more zero components than S. By repeating this, we obtain that S is a $\mathbb{k}[X_1, \ldots, X_n]$-linear combination of the S_{ij}, as required. $\square$

The syzygy criterion

It gives another refinement of the Buchberger criterion that precises Theorem 1.

Theorem 2 *Let* $G = \{g_1, \ldots, g_s\}$ *be a family of polynomials, and* $Syz(G)$ *the Syzygy module on the leading terms of* G*. Let* $\mathcal{S}$ *be a homogeneous generating set of* $Syz(G)$*. We have:*

$$G \text{ is a Gröbner basis iff for all } S \in \mathcal{S},\ S \cdot G = \sum_{i=1}^{s} h_i g_i \rightarrow_G 0 .$$

Proof: *(On the blackboard…)*

Remark: If we choose $\mathcal{S} = \{S_{ij},\ i < j\}$, as indicated in Proposition 3, then $S_{ij} \cdot G = S(g_i, g_j)$. Hence, Theorem 1 is a special case of the above one.

Practically ? The advantage of using this criterion is the possiblity to take a *smaller* generating set for $Syz(G)$ than the $\{S_{ij}\}$.

$\rightarrow$ then we can avoid more useless pairs than the criterion of Proposition 2.

Choosing a smaller generating set

1) Start form $\{S_{ij},\ i<j\}$ for a generating set of $Syz(G)$.

2) Suppose we have constructed a (smaller generating set) $\mathcal{S} \subset Syz(G)$.

Proposition 4 *If* $\text{LM}(g_\ell) | \text{LCM}(\text{LM}(g_i), \text{LM}(g_j))$ *and* $S_{i\ell}, S_{j\ell} \in \mathcal{S}$, *then* $\mathcal{S} - \{S_{ij}\}$ *is a (smaller) basis of* $Syz(G)$.

PROOF:Suppose $i<j<\ell$, and let $X^{\gamma_{i\ell}} := \text{LCM}(\text{LM}(g_i), \text{LM}(g_\ell))$ (and also let $X^{\gamma_{j\ell}}$, $X^{\gamma_{ij}}$ for the corresponding LCM). By assumption, both $X^{\gamma_{j\ell}}$ and $X^{\gamma_{i\ell}}$ divides $X^{\gamma_{ij}}$.

$$S_{ij} = \frac{X^{\gamma_{ij}}}{X^{\gamma_{i\ell}}} S_{i\ell} - \frac{X^{\gamma_{ij}}}{X^{\gamma_{j\ell}}} S_{j\ell}$$

so S_{ij} is generated by $S_{i\ell}$ and $S_{j\ell}$ and can be removed from $\mathcal{S}$. $\square$

Aim: We want to reduce the number of *pairs* to test. Let $[i,j] = (i,j)$ if $i<j$ and $[i,j] = (j,i)$ if $i>j$. Let $B \subset \{(i,j),\ 1 \le i < j \le s\}$, such that $\{S_{ab},\ (a,b) \in B\}$ generate $Syz(G)$.

Buchberger algorithm: Version 3

Define the boolean *Criterion*(f_i, f_j, B) as **true** if $[i, \ell]$ and $[j, \ell]$ are not in B, and if $\text{LM}(f_\ell)|\text{LCM}(\text{LM}(f_i), \text{LM}(f_j))$ and **false** else.

$G \leftarrow F$; $B \leftarrow \{(i,j)\,,\ 1 \leq i < j \leq s\}$; $t \leftarrow s$
while $B \neq \emptyset$ **do**
  for $(i,j) \in B$ **do**
    if $\text{LCM}(\text{LM}(f_i), \text{LM}(f_j)) \neq \text{LM}(f_i)\text{LM}(f_j)$ **and** !*Criterion*(f_i, f_j, B) **then**
      $S \leftarrow \text{NF}(S(f_j, f_i), G)$
      if $S \neq 0$ **then**
        $t \leftarrow t + 1$; $f_t \leftarrow S$
        $G \leftarrow G \cup \{f_t\}$
        $B \leftarrow B \cup \{(i,t)\,,\ 1 \leq i \leq t-1\}$
      end if ; **end if**
  $B \leftarrow B - \{(i,j)\}$
end for ; **end while** ; **return** G

Conclusion: Remarks about efficiency

…still a lot of research to compute Gröbner bases quickly…

(Buchberger, 1985), (Gebauer-Möller, 1988) $\rightarrow$ "Normal strategy" for choosing pairs to reduce and good reductors (will give a zero quickly).

(Giovanni, Mora *et al.*, 1991) "Sugar" and "Double sugar" strategy, refinement and heuristics.

J.-C. Faugère. *A new efficient algorithm for computing Gröbner bases (F_4).* J. Pure Appl. Algebra, pp:75–83, (1999, updated 2002).

Gröbner bases for grevlex are usually faster to compute
(Bayer-Stillman, 1987) $\rightarrow$ monomial order conversion algorithm (to compute a lex GB, first compute a grevlex one and *convert it* into a lex).

(Faugère, Gianni *et al.*, 1993), FGLM, change of order by linear algebra,
(Collart, Kalkbrener *et al.*, 1993 97), "Gröbner walk" on different orders.