Résolution des systèmes polynômiaux en utilisant les bases de Gröbner.

(algorithmes classiques)

Jean-Charles Faugère
INRIA (POLSYS) / UPMC / CNRS/ LIP6

1 Introduction

Le problème abordé dans ce cours est le problème fondamental de la résolution d'équations polynomiales par des méthodes de Calcul Formel. Il existe de nombreuses méthodes exactes permettant de résoudre ces systèmes (résultants, ensemble triangulaires, bases de bord, ...) ou semi-numériques (homotopies). Nous nous restreignons ici au calcul des *bases de Gröbner*. La notion de base de Gröbner a été introduite par Bruno Buchberger (Buchberger B., 1965; Buchberger B., 1970; Buchberger B., 1979; Buchberger, 1987) ; Buchberger a aussi proposé un algorithme permettant de calculer explicitement cet objet mathématique. Le principal objectif de ce document est de décrire les algorithmes classiques permettant de rendre le calcul des bases de Gröbner (algorithmes FGLM, F_4 et F_5) efficace en pratique. En particulier, ce document *ne* décrit *pas* les techniques plus récentes permettant de traiter les systèmes structurés: (comme par exemple les systèmes bilinéaires (Faugère *et al.* , 2011), les bases de Gröbner creuses (Faugère *et al.* , 2014a), les systèmes avec symétries (Faugère & Svartz, 2013), l'algorithme Sparse-FGLM (Faugère & Mou, 2011), ...) qui seront abordées pendant le cours. Pour traiter des applications, rendre les calculs de bases de Gröbner plus efficace est une nécessité. On verra aussi à travers quelques applications qu'il est souvent primordial d'adapter la mise en équation du problème pour tenir compte des spécificités des problèmes.

Le problème est de chercher les solutions d'un système d'*équations algébriques*:

$$\left\{\begin{array}{l} f_1(x_1,\ldots,x_n)=0 \\ \cdots \\ f_m(x_1,\ldots,x_n)=0 \end{array}\right. \tag{1}$$

où $f_1,\ldots,f_m$ sont des polynômes en les variables $x_1,\ldots,x_n$ et à coefficients dans un corps $\mathbb{K}$ dans lequel on sait calculer (par exemple $\mathbb{Q}$ ou $\mathbb{F}_p$).

1.1 Que veut dire résoudre un système algébrique ?

On sait que pour résoudre une équation algébrique en une variable, (le cas $m=n=1$ dans le système (1))

$$f(x)=0 \text{ où } f\in\mathbb{K}[x]$$

il est illusoire de chercher des formes closes des solutions.La question se pose alors de ce que veut dire résoudre une équation et donc a fortiori un système d'équations algébriques ? Considérons le système algébrique suivant:

$$C_3\left\{\begin{array}{l} x_1+x_2+x_3=0 \\ x_1x_2+x_2x_3+x_3x_1=0 \\ x_1x_2x_3=1 \end{array}\right. \tag{2}$$

Il est clair que ce système admet comme solutions complexes:

$$\left\{x_1=e^{\frac{2}{3}\imath\pi\,(k+2\,j)},x_2=e^{\frac{2}{3}\imath\pi\,(k+j)},x_3=e^{\frac{2}{3}\imath\pi\,k}\right\}\ \text{avec } k=0,1,2\ j=1,2 \tag{3}$$

De même que l'expression à l'aide de radicaux des solutions d'une équation algébrique de degré >5 n'est en général pas possible, l'expression des solutions d'un système algébrique ne peut se faire explicitement. L'exemple Cyclic 3 est donc atypique en ce sens. Dans la pratique on devra se contenter d'une expression *formelle* ou *approchée* des solutions; toutefois on s'attachera à garantir ou certifier tous les résultats. Pour l'exemple, on peut exprimer symboliquement toutes les solutions sous la forme:

$$=Sol_1\cup Sol_2\cup Sol_3 \tag{4}$$

où

$$Sol_1 \begin{cases} x_2^2 + x_2 + 1 = 0 \\ x_3 = 1 \\ x_1 = -x_2 - 1 \end{cases} \quad Sol_2 \begin{cases} x_3^2 + x_3 + 1 = 0 \\ x_2 = -x_3 - 1 \\ x_1 = 1 \end{cases} \quad Sol_3 \begin{cases} x_3^2 + x_3 + 1 = 0 \\ x_2 = 1 \\ x_1 = -x_3 - 1 \end{cases}$$

En d'autres termes, on *décompose* l'ensemble des solutions comme une *union* de trois sous-ensembles; chaque sous-ensemble de solutions est représenté formellement par une liste d'équations dont l'une d'entre elles est un polynôme en une variable. Chacun des systèmes Sol_i est aussi une base de Gröbner (pour l'ordre lexicographique). Ainsi pour retrouver, par exemple, la valeur numérique des solutions de Sol_1 il suffit de résoudre l'équation univariée $x_2^2 + x_2 + 1 = 0$ et de reporter dans les deux autres équations les valeurs trouvées. De façon intuitive, la solution formelle d'un système algébrique consiste donc à *réécrire* le système initial en un autre système équivalent plus simple, ou en une liste de systèmes algébriques plus simples dont la réunion des solutions est l'ensemble des solutions du système de départ. Plus exactement on cherche à se ramener au cas simple suivant:

$$\begin{cases} P_n(x_n) = 0, \\ x_{n-1} = P_{n-1}(x_n) \\ \cdots \\ x_1 = P_1(x_n) \end{cases}$$

Une base de Gröbner pour l'ordre lexicographique d'un système ayant autant d'équations que d'inconnues est le plus souvent de cette forme.

2 Bases de Gröbner et algorithme de Buchberger

2.1 Introduction

On définit un système d'équations polynomiales $f_1 = 0, \ldots, f_m = 0$ comme étant une liste de polynômes en plusieurs variables $x_1, \ldots, x_n$ avec des coefficients dans un corps $\mathbb{K}$ (les f_i sont des éléments de l'anneau $\mathbb{K}[x_1, \ldots, x_n]$). À un tel système on associe I, l'idéal engendré par $f_1, \ldots, f_m$; c'est le plus petit idéal contenant ces polynômes et c'est aussi l'ensemble des $\sum_{k=1}^{m} g_k \cdot f_k$ où les g_k sont dans $\mathbb{K}[x_1, \ldots, x_n]$. Comme f_k s'annule exactement aux points où tous les polynômes de I s'annulent, il est équivalent d'étudier le système d'équations ou l'idéal I.

Pour un ensemble d'équations linéaires on peut calculer un système triangulaire équivalent en "éliminant" tous les termes de têtes de chaque équation par application de l'algorithme d'élimination de Gauß. On peut appliquer une méthode similaire pour les polynômes. Pour cela, il est nécessaire de définir ce qu'est le terme de tête d'un polynôme, ou en d'autres mots, il faut se donner un *ordre* sur les monômes. La définition d'un ordre compatible avec la multiplication est donné en 4. Dans ce contexte, une base de Gröbner (plus exactement voir la définition 8) est un système de générateurs de l'idéal I ayant une structure triangulaire (lorsque l'ordre admissible est l'ordre lexicographique).

3 Idéaux. Variétés

3.1 Idéaux. Théorèmes de Hilbert.

Soit $\mathbb{K}$ un corps (ou un anneau euclidien). Parfois on se placera dans $\mathbb{L}$ un corps contenant $\mathbb{K}$. On note $\overline{\mathbb{K}}$ la clôture algébrique de $\mathbb{K}$ (ainsi par exemple si $\mathbb{K} = \mathbb{Q}$ alors la clôture algébrique est $\overline{\mathbb{Q}} \subset \mathbb{C}$). La notation $\mathbb{K}[x_1, \ldots, x_n]$ désigne l'ensemble des polynômes à plusieurs variables. Un polynôme p de $\mathbb{K}[x_1, \ldots, x_n]$ est une somme de monômes:

$$p = \sum_{\alpha \in \mathbf{N}^n} c_\alpha x^\alpha$$

dont presque tous les coefficients sont nuls; on utilise la notation $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$.

Nous supposons connu les notions d'idéaux et on pourra consulter (Cox *et al.* , 2007; Van der Waerden B.L., 1991) comme ouvrage de référence. Si F est un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$, alors $\mathrm{Id}(F) = \langle F \rangle$ désigne l'idéal engendré par F.

Si

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

est un système d'équations algébriques (on écrit aussi un tel système en omettant les égalités à zéro $S = (f_1, \ldots, f_m)$), on lui associe I l'idéal $\mathrm{Id}(f_1, \ldots, f_m) = \langle f_1, \ldots, f_m \rangle$ engendré par les équations de départ. Résoudre formellement un tel système c'est trouver un système de générateurs de l'idéal I *"plus simple"* que les équations de départ.

Le résultat suivant assure que tout idéal est engendré par un nombre fini d'éléments.

Théorème 1. *(Hilbert) Pour tout idéal I de $\mathbb{K}[x_1, \ldots, x_n]$, il existe un système fini de générateur $(g_1, \ldots, g_k)$ de polynômes tel que $I = \mathrm{Id}(g_1, \ldots, g_k)$.*

Théorème 2. *(Chaîne croissante d'idéaux) Si $(I_i)_{i \in \mathbb{N}}$ une famille d'idéaux tels que*

$$I_1 \subset I_2 \subset I_3 \subset \cdots$$

alors il existe un $N \in \mathbb{N}$ tel que

$$I_N = I_{N+1} = I_{N+2} = \cdots$$

3.2 Variété algébrique

Lorsqu'on cherche à *résoudre* un système algébrique on veut trouver les solutions du système algébrique; l'objet mathématique est alors la variété algébrique. Comme l'ensemble des solutions dépend de l'ensemble dans lequel on cherche les solutions:

Définition 1. *Soit $\mathbb{L}$ un corps contenant $\mathbb{K}$, alors la variété algébrique dans $\mathbb{L}$ associée à un idéal I est*

$$V_{\mathbb{L}}(I) = \{(a_1, \ldots, a_n) \in \mathbb{L}^n \text{ tel que } f(a_1, \ldots, a_n) = 0,\ \forall f \in I\}$$

(remarquer que cette définition dépend du corps dans lequel on se place).

Par exemple si $\mathbb{K} = \mathbb{Q}$ alors on peut chercher les solutions complexes du système $V_{\mathbb{C}}(I)$ ou uniquement les solutions réelles $V_{\mathbb{R}}(I) = V_{\mathbb{C}}(I) \cap \mathbb{R}^n$.

Remarque 1. *Si $S = (f_1, \ldots, f_m)$ est un système d'équations algébriques à résoudre, on peut donc lui associer deux objets mathématiques:*

- *l'idéal $I = \mathrm{Id}(f_1, \ldots, f_m)$.*
- *la variété algébrique correspondante $V_{\mathbb{L}}(I)$ qui est l'ensemble des zéros.*

Il faut observer que I contient "plus d'informations" que $V_{\mathbb{L}}(I)$: par exemple considérons les systèmes suivants (une variable et une équation):

$$x_1^2 = 0 \text{ et l'idéal associé } I_1 = \mathrm{Id}(x_1^2)$$
$$x_1 = 0 \text{ et l'idéal associé } I_2 = \mathrm{Id}(x_1).$$

Dans les deux cas la variété associée est identique $V_{\mathbb{C}}(I_1) = V_{\mathbb{C}}(I_2) = \{0\}$. Donc les idéaux I_1 et I_2 ont la même variété algébrique associée $\{0\}$. De façon intuitive, dans une variété algébrique on perd la notion de multiplicité.

Le théorème suivant dit du Nullstellensatz faible , permet de conclure qu'un système n'admet pas de solution si et seulement si l'idéal associé contient le polynôme constant 1.

Théorème 3. *(Hilbert Nullstellensatz faible) On suppose que $\mathbb{K}$ est algébriquement clos. Si $f_1, \cdots, f_m$ sont des polynômes de $\mathbb{K}[x_1, \ldots, x_n]$ et $I = \mathrm{Id}(f_1, \cdots, f_m)$. Alors $V_{\mathbb{K}}(I) = \varnothing$ implique que $I = \mathrm{Id}(1) = \mathbb{K}[x_1, \ldots, x_n]$.*

3.3 Correspondance entre les idéaux et les variétés

Jusqu'à présent nous avons cherché les solutions (des points de $\mathbb{K}^n$) d'un système d'équations polynomiales Réciproquement, à un ensemble de points on peut associer un idéal:

Définition 2. *Soit W un ensemble de $\mathbb{K}^n$. Alors*

$$I(W) = \{f \in \mathbb{K}[x_1, \ldots, x_n] \mid f(a_1, \ldots, a_n) = 0, \forall (a_1, \ldots, a_n) \in W\}$$

est un idéal.

Le théorème suivant dit du Nullstellensatz montre que si un polynôme f en plusieurs variable est nul sur tous les points d'une variété algébrique $V(I)$ n'implique pas l'appartenance de f à I en général cependant une certaine puissance de f est dans l'idéal:

Théorème 4. *(Hilbert Nullstellensatz) On suppose que $\mathbb{K}$ est algébriquement clos. Si $f, f_1, \cdots, f_m$ sont des polynômes de $\mathbb{K}[x_1, \ldots, x_n]$ alors $f \in I(V_{\mathbb{K}}(\mathrm{Id}(f_1, \cdots, f_m)))$ implique qu'il existe $k \in \mathbb{N}$ tel que $f^k \in \mathrm{Id}(f_1, \cdots, f_m)$.*

Proof. Voir la preuve dans théorème 2 (Cox *et al.* , 2007) page 180. □

Il est possible d'éliminer les multiplicités en calculant l'idéal radical:

Définition 3. *Si I est un idéal alors, l'ensemble suivant est le radical de I:*

$$\sqrt{I} = \left\{ f \in \mathbb{K}[x_1, \ldots, x_n] \mid \exists k \in \mathbb{N} \text{ tel que } f^k \in I \right\}$$

Le théorème est une simple transcription des théorèmes 1 et 3 :

Théorème 5. *1. Si $\mathbb{K}$ est algébriquement clos et si I est un idéal de $\mathbb{K}[x_1, \ldots, x_n]$ alors*

$$I(V_{\mathbb{K}}(I)) = \sqrt{I}$$

2. Si V est une variété algébrique sur $\mathbb{K}$ alors

$$V_{\mathbb{K}}(I(V)) = V$$

Une interprétation du théorème est qu'on peut transposer tout énoncé sur les variétés algébriques en un énoncé sur les idéaux radicaux (à condition que $\mathbb{K}$ soit algébriquement clos).

4 Réduction. Ordres monomiaux.

4.1 Ordres admissibles

Comme indiqué dans l'introduction du chapitre, afin de décrire un analogue de l'algorithme de Gauß nous devons définir un ordre sur les monômes d'un polynôme.

Dans la suite, $T(x_1, \ldots, x_n) = T$ est l'ensemble des termes que l'on peut former avec les variables $(x_1, \ldots, x_n)$. Par suite T est isomorphe a $\mathbb{N}^n$. Si $t = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in T$, on note $\deg(t)$ le *degré total* de t:

$$\deg(t) = \sum_{i=1}^{n} \alpha_i$$

Définition 4. *Soit $<$ un ordre total sur les exposants (donc dans $T \approx \mathbb{N}^n$); on dit que $<$ est* admissible *si l'une des conditions équivalentes est vérifiée:*

(i) $0 \leqslant \alpha$ *pour tout* $\alpha \in T$ *(autrement dit $<$ est un ordre total).*

(ii) $\alpha < \beta$ *implique* $\alpha + \gamma < \beta + \gamma$ *pour tout* $\gamma \in T$ *(c'est à dire que l'ordre monomial est compatible avec la multiplication).*

(iii) il n'y a pas de suite infinie $(\alpha_i)_{i \in \mathbb{N}}$ *strictement décroissante.*

Ordres usuels: lexicographique, DRL, ...

On définit ensuite quelques ordres admissibles qui seront les plus utilisés. L'ordre qui induit le plus de structure sur la base de Gröbner est l' ordre **lexicographique:**

$$x^\alpha = x^{(\alpha_1,\ldots,\alpha_n)} <_{\text{Lex}} x^\beta = x^{(\beta_1,\ldots,\beta_n)} \text{ s'il existe } i \text{ tel que } \begin{cases} \alpha_j = \beta_j & \text{pour} \quad j < i \\ \alpha_i < \beta_i \end{cases}$$

Ordre du degré lexicographique:

$$x^\alpha = x^{(\alpha_1,\ldots,\alpha_n)} <_{\text{Deg}} x^\beta = x^{(\beta_1,\ldots,\beta_n)} \text{ si } \begin{cases} \alpha_1 + \cdots + \alpha_n < \beta_1 + \cdots + \beta_n \\ \text{ou} \\ \alpha_1 + \cdots + \alpha_n = \beta_1 + \cdots + \beta_n \\ \text{et} \begin{cases} \alpha_j = \beta_j & \text{pour} \quad j < i \\ \alpha_i < \beta_i \end{cases} \end{cases}$$

L'ordre qui sera le plus efficace en pratique (voir l'article (D. & M., 1987) pour une justification) pour le calcul Refaire des bases de Gröbner sera l'**ordre du degré lexicographique inverse (DRL)** (F.S., 1916)

$$x^\alpha = x^{(\alpha_1,\ldots,\alpha_n)} <_{\text{DRL}} x^\beta = x^{(\beta_1,\ldots,\beta_n)} \text{ si } \begin{cases} \alpha_1+\cdots+\alpha_n < \beta_1+\cdots+\beta_n \\ \text{ou} \\ \alpha_1+\cdots+\alpha_n = \beta_1+\cdots+\beta_n \\ \text{et } \begin{cases} \alpha_j = \beta_j \quad \text{pour} \quad j > i \\ \alpha_i > \beta_i \end{cases} \end{cases}$$

Remarquer que l'inégalité est inversée $\alpha_i > \beta_i$: c'est donc l'ordre obtenu en filtrant par le degré puis en inversant l'ordre des variables et en prenant l'ordre lexicographique *opposé.* Pour le calcul des bases de Gröbner, l'ordre DRL est l'ordre le plus efficace en pratique ((D. & M., 1987)) dans la majorité des exemples (cependant pour certains exemples il peut être plus efficaces de calculer directement pour un ordre lexicographique).

Exemple 1. •

- *Pour l'ordre lexicographique tel que $z <_{Lex} y <_{Lex} x$ on a:*

$$x^3 > x^2y > x^2z > xy^2 > xyz > xz^2 > y^3 > y^2z > yz^2 > z^3$$

- *Pour l'ordre DRL tel que $z <_{DRL} y <_{DRL} x$ on a:*

$$x^3 > x^2y > xy^2 > y^3 > x^2z > xyz > y^2z > xz^2 > yz^2 > z^3$$

Ordre par blocs

On peut aussi fabriquer des ordres en découpant l'ensemble des variables $X = [x_1,\ldots,x_n]$ en $X_1 \cup X_1$ avec $X_1 = [x_1,\ldots,x_i]$ et $X_2 = [x_{i+1},\ldots,x_n]$; si on se donne deux ordres admissibles $<_1$ sur $\mathbb{N}^i$ et $<_2$ sur $\mathbb{N}^{n-i}$:

$$x^\alpha = x^{(\alpha_1,\ldots,\alpha_n)} <_{X_1,X_2} x^\beta = x^{(\beta_1,\ldots,\beta_n)} \text{ si } \begin{cases} x^{(\alpha_1,\ldots,\alpha_i)} <_1 x^{(\beta_1,\ldots,\beta_i)} \\ \text{ou} \\ \begin{cases} (\alpha_1,\ldots,\alpha_i) = (\beta_1,\ldots,\beta_i) \\ \text{et } x^{(\alpha_{i+1},\ldots,\alpha_n)} <_2 x^{(\beta_{i+1},\ldots,\beta_n)} \end{cases} \end{cases}$$

On peut aussi définir des ordres pondérés (voir un exemple, l'ordre $C_{a,b}$, dans la section **??**): par si $w = (w_1,\ldots,w_n)$ est vecteur d'entiers positifs alors on définit un degré total pondéré pour un terme $x^\alpha = x^{(\alpha_1,\ldots,\alpha_n)}$ par $\deg_w(x^\alpha) = \alpha_1w_1+\cdots+\alpha_nw_n$.

4.2 Terme de tête

Une fois l'ordre admissible fixé il est facile d'identifier dans un polynôme f son terme de tête (le plus grand terme du support de f).

Soit $f \in \mathbb{K}[x_1,\ldots,x_n]$, $f \neq 0$, tel que $f = \sum c(\alpha_1,\ldots,\alpha_n)x_1^{\alpha_1}\cdots x_n^{\alpha_n}$ (où $c(\alpha_1,\ldots,\alpha_n)$ sont des éléments de $\mathbb{K}$). On peut définir l'ensemble $M(f)$ des *monômes de f* comme

$$M(f) = \{c(\alpha_1,\ldots,\alpha_n)x_1^{\alpha_1}\cdots x_n^{\alpha_n} \mid c(\alpha_1,\ldots,\alpha_n) \neq 0\}$$

L'ensemble $T(f)$ des *termes de f* est:

$$T(f) = \{x_1^{\alpha_1}\cdots x_n^{\alpha_n} \mid c(\alpha_1,\ldots,\alpha_n) \neq 0\}$$

On définit $T_<(F)$ comme étant cet ensemble trié pour l'ordre admissible $<$:

$$T_<(F) = \text{Sort}(\{T(f) \mid f \in F\}, <)$$

Le *degré total* de $f \neq 0$ est défini par $\deg(f) = \max\{\deg(t) \mid t \in T(f)\}$.

On peut maintenant définir le *terme de tête* $\text{LT}_<(f)$, le *monôme de tête* $\text{LM}_<(f)$, et le *coefficient de tête* $\text{LC}_<(f)$ de f par rapport à $<$ de la façon suivante: $\text{LT}_<(f) = max(T(f))$, $\text{LM}_<(f) = \max(M(f))$, et $\text{LC}_<(f)$ comme étant le coefficient de $\text{LM}_<(f)$.

Si F est un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$ on peut étendre ces définitions: $\mathrm{LM}_<(F) = \{\mathrm{LM}_<(f) \mid f \in F\}$, $\mathrm{LT}_<(F) = \{\mathrm{LT}_<(f) \mid f \in F\}$ et $T(F) = \bigcup_{f \in F} T(f)$.

Définition 5. *On définit le p.p.c.m. (lcm en anglais) de deux termes par la formule:*

$$\mathrm{lcm}(x^\alpha, x^\beta) = x_1^{\max(\alpha_1, \beta_1)} \cdots x_n^{\max(\alpha_n, \beta_n)}$$

Par extension, si $<$ est un ordre admissible on définit $\mathrm{lcm}(f, g) = \mathrm{lcm}(\mathrm{LT}_<(f), \mathrm{LT}_<(g))$ où f et g sont des polynômes de $\mathbb{K}[x_1, \ldots, x_n]$.

Les ordres lexicographique et DRL vont induire des propriétés de structure différentes pour le calcul d'une base de Gröbner:

Proposition 6. *Soient f un polynôme de $\mathbb{K}[x_1, \ldots, x_n]$ et $<$ l'ordre lexicographique tel que $x_1 > \cdots > x_n$, alors si $\mathrm{LT}_{<_{Lex}}(f) \in \mathbb{K}[x_i, \ldots, x_n]$ on a $f \in \mathbb{K}[x_i, \ldots, x_n]$. En particulier si $\mathrm{LT}_{<_{Lex}}(f) \in \mathbb{K}[x_n]$ alors f est un polynôme en une variable (en x_n).*

Proposition 7. *Soit f un polynôme homogène de $\mathbb{K}[x_1, \ldots, x_n]$et $<$ l'ordre DRL tel que $x_1 > \cdots > x_n$. On a les propriétés suivantes:*

1. si x_n^k divise $\mathrm{LT}_<(f)$ pour un entier $k > 0$ alors x_n^k divise f.
2. pour tout entier $k \in \{1, \ldots, n\}$ si $\mathrm{LT}_<(f) = 0 \mod \{x_k, \ldots, x_n\}$ alors $f = 0 \mod \{x_k, \ldots, x_n\}$.

Remarque 2. *Comme le montre la proposition 7 la recherche d'éléments minimaux pour un ordre DRL conduira à l'obtention de polynômes de "bas degré" tandis qu'un ordre lexicographique favorise l'apparition de polynômes dépendant du plus petit nombre de variables.*

4.3 Réduction d'un polynôme

Dans l'anneau $\mathbb{K}[X]$ des polynômes en une variable tous les idéaux sont engendré un seul polynôme. En effet si $(f, g) \in \mathbb{K}[X]^2$ alors l'idéal $I = \mathrm{Id}(f, g)$ est engendré par le p.g.c.d. de f et g; c'est à dire $I = \mathrm{Id}(\gcd(f, g))$. Le calcul du p.g.c.d de deux polynômes par l'algorithme d'Euclide repose essentiellement sur l'opération de division euclidienne: pour tout $(f, g) \in \mathbb{K}[X]^2$ on peut trouver $(q, r) \in \mathbb{K}[X]^2$ tels que $f = q\,g + r$ avec $\deg(r) < \deg(g)$. Dans cette section nous allons introduire une généralisation de la division euclidienne pour les polynômes en plusieurs variables.

Plus exactement, nous allons introduire deux notions proches de réduction d'un polynôme f par rapport à un polynôme p; attention, toutefois, il faut distinguer la *notion mathématique* de réduction $f \xrightarrow{p} g$ (qu'on pourrait paraphraser par *f peut* se réduire en g modulo p) et la *définition algorithmique* (et donc *déterministe*) $g :=$ REDUCTION(f, p). Ces deux définitions sont ensuite étendues pour réduire un polynôme f par plusieurs polynômes: alors que mathématiquement on considère un sous ensemble fini P de $\mathbb{K}[x_1, \ldots, x_n]$, il est nécessaire d'ordonner les polynômes d'un point de vue algorithmique. Dans ce dernier cas, on utilisera la notion de *liste de polynômes* $F = [f_1, \ldots, f_m]$.

L'ordre admissible $<$ est supposé fixé.

Définition 6. *Soient $f, g, p \in \mathbb{K}[x_1, \ldots, x_n]$ tels que $p \neq 0$, et soit P un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$. Alors on dit que:*

- *f* se réduit en g modulo p *(notation $f \xrightarrow{p} g$), s'il existe $t \in T(f)$ tel que $\mathrm{LT}(p)$ divise t et $g = f - \frac{a}{\mathrm{LC}(p)} * \frac{t}{\mathrm{LT}(p)} * p$ où a est le coefficient de t dans f.*
- *f* se réduit en g modulo P *(notation $f \xrightarrow{P} g$), si $f \xrightarrow{p} g$ pour un certain $p \in P$.*
- *f* est réductible modulo p *s'il existe $g \in \mathbb{K}[x_1, \ldots, x_n]$ tel que $f \xrightarrow{g} g$.*

- f est réductible modulo P *s'il existe* $g \in \mathbb{K}[x_1, \ldots, x_n]$ *tel que* $f \xrightarrow{P} g$.
- f est top réductible modulo P *s'il existe* $g \in \mathbb{K}[x_1, \ldots, x_n]$ *tel que* $f \xrightarrow{P} g$ *et* $\mathrm{LT}(g) < \mathrm{LT}(f)$.
- $f \xrightarrow[P]{*} g$ *est la clôture réflexive transitive de* $\longrightarrow$.

Nous donnons maintenant une version algorithmique de la réduction d'un polynôme par rapport à une liste (triée) de polynômes:

Algorithme 1. REDUCTION

Input: $\begin{cases} f \text{ un polynôme} \\ F = [f_1, \ldots, f_m] \text{ une liste de polynômes} \\ < \text{ ordre admissible} \end{cases}$
Output: *un polynôme réduit.*
$f := p$
while $f \neq 0$ ***et*** f *est top réductible modulo* F ***do***
  $k := \min\{i \in \{1, \ldots, n\} \mid \mathrm{LT}(f_i) \text{ divise } \mathrm{LT}(p)\}$
  $f := f - \frac{\mathrm{LM}(f)}{\mathrm{LM}(f_k)} f_k$
return f

Proposition 8. *L'algorithme* REDUCTION *1 termine.*

On remarque que la réduction d'un polynôme par une liste de polynômes n'est pas unique et que le résultat dépend, *a priori*, de la façon d'ordonner les polynômes dans la liste F:

Exemple 2. *Voici un exemple de réduction dont le résultat change selon l'ordre des calculs:* $f = X^2 + X, f_1 = X^2 + 1, f_2 = X + 2$ *(l'ordre monomial est ici sans importance on peut considérer, par exemple, l'ordre lexicographique).*

- *Calculons* REDUCTION$(f, [f_1, f_2])$*:* f *est top-réductible modulo* $[f_1, f_2]$ *puisque* $\mathrm{LT}(f_1) = X^2 | \mathrm{LT}(f) = X^2$
  on calcule donc $f' := f - \frac{1}{1} f_1 = X - 1$*; de nouveau* f' *est top-réductible modulo* $[f_1, f_2]$ *puisque* $\mathrm{LT}(f_2) = X | \mathrm{LT}(f') = X$ *et*
  on calcule donc $f'' := f' - \frac{1}{1} f_2 = -3$*; l'algorithme termine car* $\mathrm{LT}\left(f''\right) = 1$ *n'est plus top-réductible.*
- *On calcule maintenant* REDUCTION$(f, [f_2, f_1])$*: cette fois* $\mathrm{LT}(f_2) = X | \mathrm{LT}(f) = X^2$ *et donc*
  on calcule donc $f' := f - \frac{X}{1} f_2 = X - 1 = X - 2X = -X$*; encore une fois* $\mathrm{LT}(f_2) = X | \mathrm{LT}(f') = X$ *et*
  on calcule donc $f'' := f' - \frac{-1}{1} f_2 = 2$*; l'algorithme termine car* $\mathrm{LT}\left(f''\right) = 1$ *n'est plus top-réductible.*
- *On a donc* REDUCTION$(f, [f_1, f_2]) = 3 \neq 2 =$REDUCTION$(f, [f_2, f_1])$
- *Avec la notion mathématique on a simultanément* $f \xrightarrow[[f_1, f_2]]{*} -3$ *et* $f \xrightarrow[[f_1, f_2]]{*} 2$.

Proposition 9. *1. Si* $r =$REDUCTION(p, F) *alors* $r - p \in \mathrm{Id}(F)$

2. Si $p \xrightarrow[F]{*} r$ *alors* $r - p \in \mathrm{Id}(F)$

Corollaire 1. *1. Si* $r =$REDUCTION(p, F) *alors il existe deux suites finies* $(g_n)_{n=0,\ldots,k}$ *et* $(m_n)_{n=0,\ldots,k}$ *des monômes telles que* $g_n \in F$ *et* $r - p = \sum_{i=1}^{k} m_i g_i$ *avec* $\mathrm{LT}(p) = \mathrm{LT}(m_1 g_1) > \mathrm{LT}(m_2 g_2) > \cdots > \mathrm{LT}(m_k g_k)$

2. Si $p \xrightarrow[F]{*} r$ *alors il existe deux suites finies* $(g_n)_{n=0,\ldots,k}$ *et* $(m_n)_{n=0,\ldots,k}$ *des monômes telles que* $g_n \in F$ *et* $r - p = \sum_{i=1}^{k} m_i g_i$ *avec* $\mathrm{LT}(p) \geqslant \mathrm{LT}(m_i g_i)$ *pour tout* $i \in \{1, \ldots, k\}$.

Proposition 10. *Si* $p =$REDUCTION(f, F) *et* $p \neq 0$ *alors* $\mathrm{LT}(p) \notin \mathrm{Id}(\mathrm{LT}(F))$.

Lorsqu'on considère deux polynômes f et g, en général on ne peut pas réduire f par g ou g par f. Par exemple considérons le cas

$$\begin{aligned} f_1 &= x^2y + x + 1 \\ f_2 &= xy^2 - 3 \end{aligned}$$

Dans ce cas il est nécessaire, pour "éliminer les termes de têtes" de multiplier f_1 et f_2 par des monômes:

$$y\, f_1 - x\, f_2 = y\,(x+1) - x\,(-3) = xy + 3x + y$$

La formule suivante du S-polynôme donne une définition précise pour cette opération dans le cas général, cette opération est une composante essentielle de l'algorithme de Buchberger:

Définition 7. *Le* S-polynôme *de f et g est défini par*

$$\mathrm{Spol}(f,g) = \mathrm{LC}(g)\frac{\mathrm{lcm}(f,g)}{\mathrm{LT}(f)}f - \mathrm{LC}(f)\frac{\mathrm{lcm}(f,g)}{\mathrm{LT}(g)}g$$

4.4 Réduction totale d'un polynôme

La fonction REDUCTION se contente de simplifier (réduire) le terme de tête d'un polynôme. La fonction REDUCTIONTOTALE permet de réduire tous les termes d'un polynôme (et donc d'obtenir une expression plus canonique):

Algorithme 2. REDUCTIONTOTALE

Input: $\begin{cases} f \text{ un polynôme} \\ F = [f_1, \ldots, f_m] \text{ une liste de polynômes} \\ < \text{ ordre admissible} \end{cases}$
Output: *un polynôme totalement réduit.*
$p := f$ *et* $p_0 := 0$
while $p \neq 0$ ***do***
  $p :=$REDUCTION(p, F)
  $p_0 := p_0 + \mathrm{LM}(p)$
  $p := p - \mathrm{LM}(p)$
return p_0

On peut facilement adapter les propositions liées à la fonction de réduction pour la fonction de réduction totale:

Proposition 11. *L'algorithme* REDUCTIONTOTALE *termine.*

Proposition 12. *1. Si* $p =$REDUCTIONTOTALE(f, F) *alors* $T(p) \cap \mathrm{Id}(\mathrm{LT}(F)) = \varnothing$

2. Si $r =$REDUCTIONTOTALE(f, F) *alors* $r - f \in \mathrm{Id}(F)$.

5 Bases de Gröbner

5.1 Définition d'une base de Gröbner

La définition suivante de base de Gröbner est purement mathématique et permet donc de définir indépendamment de tout algorithme une base "canonique" d'un idéal. Fort heureusement, il existe aussi plusieurs algorithmes permettant de calculer une base de Gröbner (voir la description de l'algorithme 3 de Buchberger plus bas).

Définition 8. *Soit I un idéal. Un sous ensemble fini $G = (g_1, \ldots, g_k)$ de $\mathbb{K}[x_1, \ldots, x_n]$ est dit une base de Gröbner de I pour l'ordre admissible $<$ si pour tout $f \in I$ il existe $1 \leqslant i \leqslant k$ tel que $\mathrm{LT}(g_i)$ divise $\mathrm{LT}(f)$.*

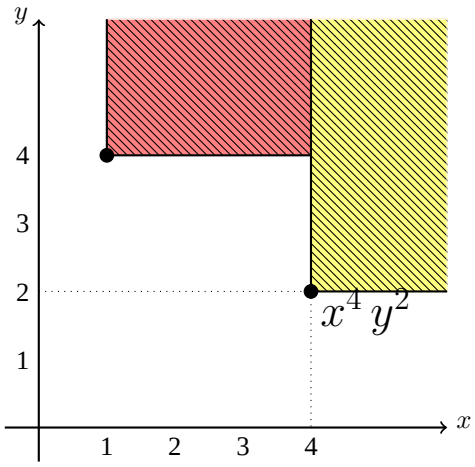


Figure 5.1: structure en escalier d'une base de Gröbner.

Afin de visualiser le concept de base de Gröbner, il est commode de reporter sur un dessin (voir figure 5.1) les points $\mathrm{LT}(f)$ pour $f \in I$; par exemple, en dimension $n = 2$, s'il existe un polynôme f_0 de l'idéal I tel que $\mathrm{LT}(f_0) = x^4y^2$ on reporte sur le dessin le point de coordonnées $(4, 2)$. On itère le processus pour tous les polynômes f de l'idéal I. Dans notre exemple on sait que $x^iy^j\ f_0 \in I$ pour tout $(i, j) \in \mathbb{N}^2$; ainsi $\mathrm{LT}\left(x^iy^j\ f_0\right) = x^iy^j\ \mathrm{LT}(f_0) = x^{4+i}y^{2+j}$ et on reporte sur le dessin tous les points $(4 + i, 2 + j)$ pour tout $0 \leqslant i, j$; c'est donc tout le quadrant supérieur au point $(4, 2)$ qui est ainsi hachuré sur la figure. Le dessin met en évidence la *structure d'escalier* : les points minimaux sont justement les éléments de la base de Gröbner.

Théorème 13. *On fixe un ordre admissible $<$. Tout idéal I possède une base de Gröbner G.*

Proof. On pourrait démontrer ce théorème par une preuve non constructive (basée sur le lemme de Dickson) mais nous allons donner un *algorithme* permettant de calculer une base de Gröbner à partir d'un système de générateurs. □

Théorème 14. *(Buchberger) Soient $G = [g_1, \ldots, g_k] \subset \mathbb{K}[x_1, \ldots, x_n]$ et $<$ un ordre admissible fixé. Les conditions suivantes sont équivalentes:*

(i) *G est une base de Gröbner de $\mathrm{Id}(g_1, \ldots, g_k)$ pour $<$.*

(ii) $\textsc{Reduction}(p, G) = 0$ *si et seulement si* $p \in \mathrm{Id}(G)$.

Théorème 15. *(Buchberger) Soit $G = [g_1, \ldots, g_k]$ une base de Gröbner de $\mathrm{Id}(g_1, \ldots, g_k)$ pour un ordre $<$ fixé. Alors* $\textsc{ReductionTotale}(p, G)$ *est unique quelque soit la stratégie de réduction (c'est à dire quelque soit la façon d'ordonner G).*

Définition 9. *Soit $G = [g_1, \ldots, g_k]$ une base de Gröbner d'un idéal I pour un ordre admissible $<$; pour tout polynôme f, on note* $\textsc{NormalForm}(f, G, <)$ *le résultat de la fonction* $\textsc{ReductionTotale}(f, G)$. *On notera aussi* $\mathrm{NF}(f, G, <)$ *cette forme normale.*

5.2 Algorithme de Buchberger

La forme la plus simple de l'algorithme de Buchberger est maintenant présentée; une notion fondamentale dans la description de l'algorithme est la notion de paire critique. *une paire critique* est simplement un couple de polynômes (f, g) pour lesquels on va calculer un S-polynôme; l'algorithme maintient une liste de tels couples (ou encore liste des paires critiques) qu'il faudra tous explorer. À noter que, sous cette forme, l'algorithme de Buchberger est totalement inefficace, mais on verra plus loin comment lui adjoindre des critères (les critères de Buchberger) permettant de limiter la liste des paires critiques et donc de le rendre plus efficace.

Algorithme 3. *(Buchberger)*

Input: $\begin{cases} F = [f_1, \ldots, f_s] \text{ une liste de polynômes} \\ < \text{un ordre admissible} \end{cases}$
Output: G un sous ensemble (fini) de $\mathbb{K}[x_1, \ldots, x_n]$.
$G := F$ et $m := s$
$P := \{(f_i, f_j) \mid 1 \leqslant i < j \leqslant m\}$ la liste des paires critiques
while $P \neq \varnothing$ ***do***
  Choisir et retirer de P une paire critique (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

Remarque 3. *Pour plus d'efficacité, dans l'algorithme de Buchberger, on peut utiliser la fonction* REDUCTIONTOTALE *à la place de l'appel à la fonction* REDUCTION. *Sous cette forme, l'algorithme possède plusieurs degrés de liberté: ainsi, dans la boucle principale, on peut choisir n'importe quelle paire critique; en pratique, cependant, il est nécessaire de ne pas choisir les paires critiques dans un ordre arbitraire (voir la section 6.2 sur les stratégies de calcul).*

Exemple 3. *On considère l'ordre lexicographique* $x > y$ *et la liste de polynômes* $F = \left[f_1 = x^2y - 1, f_2 = xy^2 - 3\right]$ *et on applique l'algorithme de Buchberger:*

Initialement $P = \{[f_1, f_2]\}$ *et* $G = [f_1, f_2]$

1. $P = \{[f_1, f_2]\}$ *on calcule* $f_3 := \mathrm{Spol}(f_1, f_2) = -y + 3\,x$ *puis* $f_3 :=$REDUCTION$(f_3, G) = 3\,x - y$
   On ajoute donc un nouvel élément dans G: $G = [f_1, f_2, f_3]$ *et* $P = \{[f_1, f_3], [f_2, f_3]\}$
2. $P = \{[f_1, f_3], [f_2, f_3]\}$ *on calcule* $f_4 := \mathrm{Spol}(f_1, f_3) = xy^2 - 3$ *puis* REDUCTION$(f_4, G) = 0$
3. $P = \{[f_2, f_3]\}$ *on calcule* $f_4 := \mathrm{Spol}(f_2, f_3) = -9 + y^3$ *puis* $f_4 :=$REDUCTION$(f_4, G) = y^3 - 9$
   On ajoute donc un nouvel élément dans G: $G = [f_1, f_2, f_3, f_4]$ *et* $P = \{[f_1, f_4], [f_2, f_4], [f_3, f_4]\}$
4. $P = \{[f_1, f_4], [f_2, f_4], [f_3, f_4]\}$ *on calcule* $f_5 := \mathrm{Spol}(f_1, f_4) = -y^2 + 9\,x^2$ *puis* REDUCTION$(f_5, G) = 0$
5. $P = \{[f_2, f_4], [f_3, f_4]\}$ *on calcule* $f_5 := \mathrm{Spol}(f_1, f_4) = -3\,y + 9\,x$ *puis* REDUCTION$(f_5, G) = 0$
6. $P = \{[f_3, f_4]\}$ *on calcule* $f_5 := \mathrm{Spol}(f_1, f_4) = -y^4 + 27\,x$ *puis* REDUCTION$(f_5, G) = 0$
7. $P = \varnothing$ *l'algorithme termine et retourne* $G = [x^2y - 1, xy^2 - 3, 3\,x - y, y^3 - 9]$.

Dans un premier temps on montre de façon non constructive la terminaison de l'algorithme de Buchberger; la preuve complète sera donnée dans la section suivante (théorème 21).

Théorème 16. *L'algorithme de Buchberger termine.*

Proof. On sait déjà que l'appel à la fonction REDUCTION se termine (voir proposition 8) et donc la preuve de la terminaison de l'algorithme de Buchberger se fait en remarquant que le seul moment où $\mathrm{Id}(G)$ change est lorsqu'on passe par la ligne $G := G \cup \{f_m\}$ dans l'algorithme de Buchberger; dans ce cas $\mathrm{Id}(G \cup \{f_m\}) \supseteq \mathrm{Id}(G)$ et, a fortiori, $\mathrm{Id}(\mathrm{LT}(G) \cup \{\mathrm{LT}(f_m)\}) \supset \mathrm{Id}(\mathrm{LT}(G))$.

Comme $f_m =$REDUCTION(f_m, G) on sait que f_m n'est pas réductible par G; autrement dit $\mathrm{LT}(f_m)$ n'est pas dans $\mathrm{Id}(\mathrm{LT}(G))$ (proposition 10) d'où l'inclusion stricte:

$$\mathrm{Id}\left(\mathrm{LT}(G \cup \{f_m\}\right) \supsetneq \mathrm{Id}(\mathrm{LT}(G)).$$

Par conséquent, si l'algorithme de Buchberger ne se terminait pas, on pourrait fabriquer une suite infinie strictement croissant d'idéaux monomiaux: $I_m = I_{m-1} + \mathrm{Id}\left(\mathrm{LT}\left(f_m\right)\right)$ pour $m > s$ et $I_s = \mathrm{Id}\left(F\right)$. D'après le théorème 2 on aurait une contradiction. □

Pour compléter la preuve complète de l'algorithme de Buchberger nous avons besoin d'un théorème caractérisant les bases de Gröbner qui fait l'objet du paragraphe suivant (voir théorème 21).

5.3 Caractérisation d'une base de Gröbner

Le théorème de caractérisation suivant permet de prouver l'algorithme de Buchberger , la version forte de Buchberger avec critères et l'algorithme F_4; l'énoncé de ce théorème dépend de la notion de $t-$représentation où t est un terme: lorsqu'on considère un élément f d'un idéal $\mathrm{Id}(p_1,\ldots,p_k)$, il existe, souvent, une infinité d'écriture $f=\sum_{i=1}^{k} g_i\, p_i$ avec $g_i \in \mathbb{K}[x_1,\ldots,x_n]$; une $t-$représentation permet de "borner" une telle écriture en imposant $t \geqslant \mathrm{LT}(g_i\, p_i)$ pour tout $i \in \{1,\ldots,k\}$.

Définition 10. *Soient $P=[p_1,\ldots,p_k]$ un sous ensemble fini de $\mathbb{K}[x_1,\ldots,x_n]$, $f \in \mathbb{K}[x_1,\ldots,x_n]$, $f \neq 0$ et $t \in T$. S'il existe $(g_1,\ldots,g_k) \in \mathbb{K}[x_1,\ldots,x_n]^k$ tels que:*

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

*alors on dit que c'est une $t-$*représentation *de f par rapport à P si $t \geqslant \mathrm{LT}(g_i p_i)$ pour tout $1 \leqslant i \leqslant k$. On note $f=\mathcal{O}_P(t)$ cette propriété et on note $f=o_P(t)$ lorsqu'il existe $t' \in T$ tel que $t'<t$ et $f=\mathcal{O}_P(t')$.*

Exemple 4. *On considère l'exemple suivant dans $\mathbb{Q}[x,y,z]$ avec l'ordre DRL $z<y<z$ et les polynômes $f=x^2yz-xy^2z$, $f_1=xy^2+xyz$, $f_2=x^2y+xyz$ alors*

$$f=x\, f_1+y\, f_2$$

On a donc une écriture de $f \in \mathrm{Id}(f_1,f_2)$ sous la forme $f=g_1\, f_1+g_2\, f_2$ mais ce n'est pas une $\mathrm{LT}(f)-$représentation puisque $\mathrm{LT}(x\, f_1)=\mathrm{LT}(y\, f_2)=x^2y^2>\mathrm{LT}(f)$.

Proposition 17. *Si f,g sont des polynômes, t un terme, P un sous ensemble fini de polynômes alors*

$f=\mathcal{O}_P(t)$	$g=\mathcal{O}_P(t)$	*implique*	$f+g=\mathcal{O}_P(t)$
$f=o_P(t)$	$g=o_P(t)$	*implique*	$f+g=o_P(t)$
$f=\mathcal{O}_P(t)$	$u \in T$	*implique*	$uf=\mathcal{O}_P(ut)$
$f=o_P(t)$	$u \in T$	*implique*	$uf=o_P(ut)$

Le corollaire 1 nous permet de montrer:

Proposition 18.
1. *Si* $\textsc{Reduction}(p,P)=0$ *alors* $p=\mathcal{O}_P(\mathrm{LT}(p))$.
2. *Si* $p \xrightarrow[P]{*} 0$ *alors* $p=\mathcal{O}_P(\mathrm{LT}(p))$.

Le théorème suivant donne une caractérisation non algorithmique des bases de Gröbner (car elle implique un nombre de tests infini):

Théorème 19. *G est une base de Gröbner si et seulement si $\forall 0 \neq f \in \mathrm{Id}(G)$, $f=\mathcal{O}_G(\mathrm{LT}(f))$.*

Proof.
- Si on suppose que G est une base de Gröbner (constitué de polynômes unitaires), alors $\forall f \neq 0 \in Id(G)$, $\textsc{Reduction}(f,G)=0$ donc $f=\mathcal{O}_G(\mathrm{LT}(f))$ d'après la proposition 1.
- Réciproquement, soit f un élément quelconque de $\mathrm{Id}(G)$; par hypothèse on peut écrire f sous la forme $f=\sum_{i=1}^{k} h_i g_i$ avec $\mathrm{LT}(f) \geqslant \max_i \mathrm{LT}(h_i g_i)$. L'inégalité stricte est impossible donc il existe $i \in \{1,\ldots,k\}$ tel que $\mathrm{LT}(f)=\mathrm{LT}(h_i g_i)$. Donc f est top réductible par g_i donc par G et G est une base de Gröbner.

□

Le théorème permet de caractériser les bases de Gröbner; il peut servir à la fois dans la preuve de correction de l'algorithme de Buchberger mais aussi pour la preuve des algorithmes utilisant l'algèbre linéaire comme F_4 (théorème 44):

Théorème 20. *Soit $G \subset \mathbb{K}[x_1, \ldots, x_n]$ un ensemble fini de polynômes ne contenant pas zéro. On suppose que pour tout $(g_1, g_2) \in G^2$, $\mathrm{Spol}(g_1, g_2) = 0$ ou $\mathrm{Spol}(g_1, g_2) = o_G(\mathrm{lcm}(g_1, g_2))$. Alors G est une base de Gröbner de $\mathrm{Id}(G)$.*

Corollaire 2. *(Buchberger) Soit G un sous ensemble fini de polynômes. G est une base de Gröbner si et seulement si $\mathrm{Spol}(f, g) \xrightarrow[G]{*} 0$ pour tout $(f, g) \in G^2$ tels que $f \neq g$.*

Proof. Soient $(f, g) \in G^2$, $f \neq g$. On pose $t = \mathrm{LT}(\mathrm{Spol}(f, g) < \mathrm{lcm}(\mathrm{LT}(f), \mathrm{LT}(g))$. Si $\mathrm{Spol}(f, g) \xrightarrow[G]{*} 0$ alors d'après la proposition 1 $\mathrm{Spol}(f, g) = \mathcal{O}_G(\mathrm{LT}(\mathrm{Spol}(f, g))) = \mathcal{O}_G(t) = o_G(\mathrm{lcm}(f, g))$. □

Le corollaire suivant donne un moyen algorithme pour vérifier qu'une liste de polynômes est une base de Gröbner:

Corollaire 3. *Soit G un sous ensemble fini de polynômes. G est une base de Gröbner si et seulement si* REDUCTION$(\mathrm{Spol}(f, g), G) = 0$ *pour tout $(f, g) \in G^2$.*

Proof. Soient $(f, g) \in G^2$, $f \neq g$. On pose $t = \mathrm{LT}(\mathrm{Spol}(f, g) < \mathrm{lcm}(\mathrm{LT}(f), \mathrm{LT}(g))$. Si REDUCTION$(\mathrm{Spol}(f, g), G) = 0$ alors d'après la proposition 1 $\mathrm{Spol}(f, g) = \mathcal{O}_G(\mathrm{LT}(\mathrm{Spol}(f, g))) = \mathcal{O}_G(t) = o_G(\mathrm{lcm}(f, g))$. □

On peut maintenant terminer la preuve de correction de l'algorithme de Buchberger:

Théorème 21. *L'algorithme de Buchberger calcule une base de Gröbner de l'idéal engendré par $(f_1, \ldots, f_m)$.*

Proof. On note G_m la base G à l'étape m. $G_s = F$ et $G_k = G$ la base finale. On a $G_s \subset G_{s+1} \subset \cdots \subset G_k$. Pour tout $1 \leqslant i < j \leqslant k$ on a: (f_i, f_j) est une paire critique donc il existe une étape $m \leqslant k$ où cette paire a été considérée; deux cas se présenter:

ou REDUCTION$(\mathrm{Spol}(f_i, f_j), G_m) = 0$ donc $\mathrm{Spol}(f_i, f_j) \xrightarrow[G_k]{*} 0$

ou REDUCTION$(\mathrm{Spol}(f_i, f_j), G_m) = f_{m+1} \neq 0$ et donc $G_{m+1} = \{f_{m+1}\} \cup G_m$, par conséquent REDUCTION$(\mathrm{Spol}(f_i, f_j), G_{m+1}) = 0$ donc $\mathrm{Spol}(f_i, f_j) \xrightarrow[G_k]{*} 0$. La preuve est terminée grâce au corollaire 2. □

5.4 Base de Gröbner minimale et réduite. Unicité

À une normalisation près, on va montrer qu'une base de Gröbner pour un ordre fixé est unique.

Définition 11. *Une base de Gröbner G est minimale si pour tout $g \in G$*

(i) $\mathrm{LC}_<(g) = 1$

(ii) $\mathrm{LT}_<(g) \notin \mathrm{Id}(\mathrm{LT}_<(G \backslash \{g\}))$

L'algorithme suivant permet de rendre minimale une base de Gröbner:

Algorithme 4. MINGBASIS

***Input**: $F = [f_1, \ldots, f_m]$ une base de Gröbner*
***Output**: une base de Gröbner minimale.*
if** $F = \emptyset$ **then
  ***return** $\emptyset$*
p le plus grand élément de F pour $<$
$G :=$ minGBasis$(F \backslash \{p\})$
if** p n'est pas top réductible modulo G **then
  $G := G \cup \left\{ \frac{p}{\mathrm{LC}(p)} \right\}$
***return** G*

Proposition 22. *Si G est une base de Gröbner pour $<$ et G_1, G_2 des bases de Gröbner minimales de G pour $<$. Alors*

(i) $\mathrm{LT}(G_1) = \mathrm{LT}(G_2)$

(ii) cardinal G_1 = cardinal G_2

Une de Gröbner base minimale rendre une base de Gröbner vraiment unique il faut des hypothèses plus fortes que pour une base minimale:

Définition 12. *Une base de Gröbner G est réduite si pour tout $g \in G$*

Proposition 23. *(i)* $\mathrm{LC}(g) = 1$

(ii) $T(g) \cap \mathrm{Id}(\mathrm{LT}(G \backslash \{g\})) = \varnothing$

L'algorithme suivant permet de calculer une base réduite:

Algorithme 5. BASEREDUITE

***Input**: $F = (f_1, \ldots, f_m)$ une base de Gröbner*
***Output**: une base de Gröbner réduite.*
$G = minGBasis(F) = [g_1, \ldots, g_k]$
return $[\text{REDUCTIONTOTALE}(g_i, G \backslash \{g_i\})\ i = 1, \ldots, k]$

On peut facilement modifier l'algorithme de Buchberger pour qu'il retourne une base de Gröbner *réduite*:

Algorithme 6. *de Buchberger*

***Input**:* $\begin{cases} F = [f_1, \ldots, f_s] \text{ une liste de polynômes} \\ < \text{un ordre admissible} \end{cases}$
***Output**: G un sous ensemble (fini) de $\mathbb{K}[x_1, \ldots, x_n]$.*
$G := F$ *et* $m := s$
$P := \{(f_i, f_j) \mid 1 \leqslant i < j \leqslant m\}$ *la liste des paires critiques*
while $P \neq \varnothing$ ***do***
  Choisir et retirer de P une paire critique (f, g)
  $f_{m+1} :=$REDUCTION$(\mathrm{Spol}(f, g), G)$
  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 BASEREDUITE(G)

L'avantage de calculer une base de Gröbner complètement réduite est que le résultat est alors unique (et donc canonique) une fois l'ordre fixé:

Théorème 24. *Soient I un idéal et $<$ un ordre admissible fixé. Alors il existe une unique base de Gröbner complètement réduite de l'idéal I.*

5.5 Propriétés des bases de Gröbner. Élimination.

Une autre propriété fondamentale des bases lexicographiques est de pouvoir *éliminer* des variables:

Théorème 25 (Elimination Theorem)**.** *Soit I un idéal de $\mathbb{K}[x_1, \ldots, x_n]$, et $k \in \{1, \ldots, n\}$. Si G est une base de Gröbner pour l'ordre lexicographique ou un ordre par blocs de I, alors $G_k = G \cap \mathbb{K}[x_k, \ldots, x_n]$ est une base de Gröbner de $I_k = I \cap \mathbb{K}[x_k, \ldots, x_n]$.*

Proof. On fixe $k \in \{1, \ldots, n\}$. On veut montrer que G_k est une base de Gröbner de I_k: soit $f \in I_k$. A fortiori $f \in I$, donc il existe $g \in G$ tel que $\mathrm{LT}(g)$ divise $\mathrm{LT}(f) \in \mathbb{K}[x_k, \ldots, x_n]$ d'où $\mathrm{LT}(g) \in \mathbb{K}[x_k, \ldots, x_n]$; en appliquant la proposition 6 on obtient que $g \in \mathbb{K}[x_k, \ldots, x_n]$. Donc G_k est une base de Gröbner de I_k. □

Remarque 4. *En particulier, si on applique le théorème 25 avec $l = n$ on obtient que $G \cap \mathbb{K}[x_n]$ est une base de Gröbner de $I_n = I \cap \mathbb{K}[x_n]$ dans $\mathbb{K}[x_n]$; comme I_n est un idéal principal, il est donc engendré par un polynôme $P_n(x_n)$ (éventuellement nul).*

En pratique pour éliminer on préfère éviter l'usage de l'ordre lexicographique qui est en général très coûteux. Pour cette raison on définit la notion d'ordre d'élimination:

Définition 13. *Soit $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_s]$, un ordre $<$ est un ordre d'élimination par rapport à $y_1, \ldots, y_s$ si*

$$\forall f \in \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_s],\ \mathrm{LT}(f) \in f \in \mathbb{K}[y_1, \ldots, y_s] \text{ implique } f \in \mathbb{K}[y_1, \ldots, y_s]$$

Un exemple d'ordre d'élimination est l'ordre par bloc $<_{\mathrm{DRL,DRL}}$ où le premier groupe de variables est $[x_1, \ldots, x_n]$ et le deuxième groupe de variables est $[y_1, \ldots, y_s]$.

Théorème 26. *Si I est un idéal de $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_s]$, $<$ un ordre d'élimination par rapport à $y_1, \ldots, y_s$ et G une base de Gröbner de I pour cet ordre alors $G \cap \mathbb{K}[y_1, \ldots, y_s]$ est une base de Gröbner de $I \cap \mathbb{K}[y_1, \ldots, y_s]$.*

Base de Gröbner pour un ordre du degré

Bien que la forme d'une base de Gröbner pour un ordre du degré (par exemple l'ordre DRL) ne soit pas simple à décrire ces bases ont la propriété de contenir tous les polynômes de plus bas degré d'un idéal I. Plus exactement on peut calculer une base de $I_d = \{g \in I \mid \deg(g) = d\}$ lorsque d est le degré minimal d'un élément non nul de I:

Théorème 27. *Soit $I \subset \mathbb{K}[x_1, \ldots, x_n]$, $d = \min\{\deg(f) \mid 0 \neq f \in I\}$, et G une base de Gröbner pour l'ordre DRL de I. Alors:*

$$\mathrm{Vect}_{\mathbb{K}}\left(\{g \in G \mid \deg(g) = d\}\right) = \mathrm{Vect}_{\mathbb{K}}\left(I_d\right),$$

où $I_d = \{g \in I \mid \deg(g) = d\}$. Autrement dit, les éléments de plus bas degré de G forment une base (comme espace vectoriel) de I_d.

6 Stratégies - Polynômes homogènes

Dans l'algorithme de Buchberger il existe de nombreux choix (par le choix d'une paire critique) qui n'influent pas sur le résultat final mais qui ont une incidence très grande sur le temps de calcul, on appelle ces choix des *stratégies*; l'idée commune des stratégies est de se ramener plus ou moins au cas homogène; en effet pour les polynômes homogènes il existe une stratégie – la stratégie normale – qui consiste à traiter en priorité les paires critiques de plus bas degré; la motivation de cette stratégie provient du fait que pour les polynômes homogènes on peut définir une notion de base de Gröbner tronquée en degré d ayant les mêmes propriétés qu'une base de Gröbner lorsqu'on se limite au polynômes de degré d. Appliquer la stratégie normale revient donc à calculer successivement des bases de Gröbner tronquées en degré d, $d+1$, $d+2$, ...

6.1 Base de Gröbner tronquée

Définition 14. *Un polynôme f est homogène si pour tout $t \in T(f)$ on a $\deg(t) = deg(f)$.*

Supposons que tous les polynômes de F sont homogènes. Pour tout $(f, g) \in F^2$, les polynômes $h = \mathrm{Spol}(f, g) = m\, f - m^{'}\, g$ et $f \to f' = f - m\, p$, où $\mathrm{LT}(m\, p) \in T(f)$, sont encore homogènes. La définition suivante est donc bien fondée:

Définition 15. *Si $f_1, \ldots, f_m$ sont des polynômes homogènes et $<$ un ordre admissible, on note G_d le résultat de l'algorithme de Buchberger tronqué au degré d appliqué à la liste de polynômes $[f_1, \ldots, f_m]$ (c'est à dire qu'on élimine toutes les paires critiques dont le degré total (voir la définition 29) est $> d$); on appelle G_d une d-base de Gröbner de $I = \mathrm{Id}\left(f_1, \ldots, f_m\right)$ pour l'ordre $<$.*

Le théorème suivant ((Lazard D., 1983), (Becker T. and Weispfenning V., 1993) p. 471) permet de donner une structure à cette liste de polynômes lorsqu'ils sont homogènes:

Théorème 28. *Pour des polynômes homogènes $f_1, \ldots, f_m$, G_d est une base de Gröbner "jusqu'au degré d" pour un ordre $<$; les propriétés suivantes sont vérifiées:*

- $\xrightarrow{*}$ *ne dépend pas de l'ordre des calculs pour les polynômes* f *tels que* $\deg(f) \leqslant d$.
- *Pour tout* $p \in I$ *tel que* $\deg(p) \leqslant d$ *on a* $p \xrightarrow{*} 0$
- $\mathrm{Spol}(f,g) \xrightarrow{*} 0$ *pour* (f,g) *dans* G_d^2 *tels que* $\deg(\mathrm{lcm}(\mathrm{LT}(f),\mathrm{LT}(g))) \leqslant d$
  On a inclusion des bases de Gröbner, et il existe d_∞ *tel que*

$$G_2 \subset G_3 \subset \cdots \subset G_{d_\infty} = G_{d_\infty+1} = G$$

  où G *est la base de Gröbner* $f_1, \ldots, f_m$,

6.2 Stratégie normale. Polynômes homogènes.

6.2.1 Stratégie sur le choix des paires critiques et la réduction des polynômes

Le choix d'une paire critique est le principal problème de choix dans l'algorithme de Buchberger.La stratégie la plus naturelle pour sélectionner les paires critiques est de considérer prioritairement les paires de bas degré:

Algorithme 7. SELECTION *(Stratégie Normale)*

Input*:* $P \neq 0$ *une liste de paires critiques*
Output*: sélection d'une paire dans* P
$P = [(f_i, g_i),\ i = 1, \cdots, k]$
$t_0 = \min_<\{\mathrm{lcm}(\mathrm{LT}(f_i), \mathrm{LT}(g_i)),\ i = 1, \cdots, k\}$
$\exists i_0$ *tel que* $\mathrm{lcm}(\mathrm{LT}(f_{i_0}), \mathrm{LT}(g_{i_0})) = t_0$
return (f_{i_0}, g_{i_0})

6.2.2 Stratégie pour la réduction d'un polynôme.

Pour des polynômes homogènes, en suivant la stratégie normale, on obtient le diagramme suivant (voir figure 6.2.2) donnant le degré maximal du monôme $\mathrm{lcm}(\mathrm{LT}(f), \mathrm{LT}(g))$ en fonction de l'étape de l'algorithme. La stratégie est d'un certain point de vue optimale puisqu'on va successivement calculer bases de Gröbner tronquée $G_d \subset G_{d+1} \subset \cdots$ (en particulier les polynômes calculés $\neq 0$ sont dans la base finale).

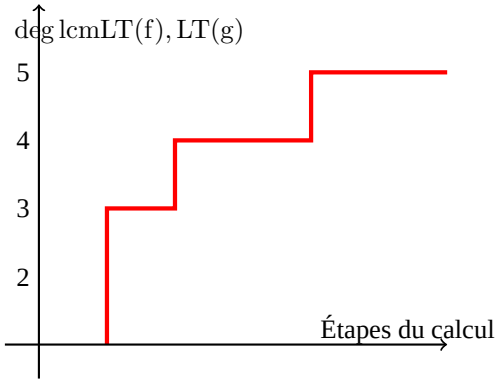


Fig 6.2.2: comportement avec des polynômes homogènes.

6.3 Cas affine - Homogénéisation

En revanche, si on applique cette même stratégie à des polynômes affines, on observe des chutes de degré (voir la figure 6.3). Ceci induit souvent un comportement chaotique de l'algorithme et, par exemple, une croissance exagérée des coefficients.

Pour remédier à ce problème, on ramène le cas des polynômes affines au cas des polynômes homogènes par homogénéisation :

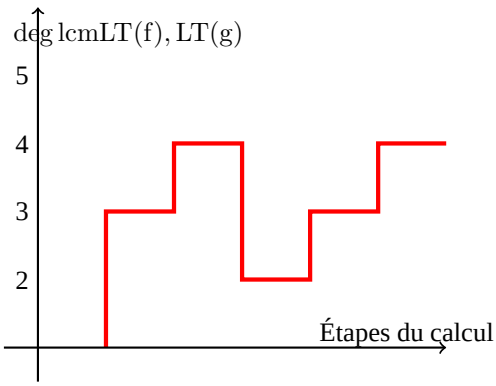


Fig 6.3: comportement en affine

Définition 16. *On définit l'opération d'homogénéisation des polynômes:*

$$\begin{array}{rcl} \mathbb{K}[x_1,\cdots,x_n] & \to & \mathbb{K}[x_1,\cdots,x_n,h] \\ f & \mapsto & f^h = h^{\deg(f)} \cdot f(\frac{x_1}{h},\cdots,\frac{x_n}{h}) \end{array}$$

On note f_i^H la partie homogène de plus haut degré de f_i:

$$f_i^H = f^h(x_1,\cdots,x_n,0)$$

Réciproquement on peut déshomogénéiser:

$$\begin{array}{rcl} \mathbb{K}[x_1,\cdots,x_n,h] & \to & \mathbb{K}[x_1,\cdots,x_n,h] \\ f & \mapsto & f_h = f(x_1,\cdots,x_n,1) \end{array}$$

Théorème 29. *Soit $F \subset \mathbb{K}[x_1,\cdots,x_n]$ un ensemble fini, on considère F^h les polynômes homogènes obtenus à partir de F par homogénéisation. On peut ensuite calculer une base de Gröbner, G, de F^h en utilisant la stratégie normale. Enfin, par déshomogénéisation, on obtient une liste de polynômes $G' = G_h$. Alors G' est une base de Gröbner (non réduite) de F.*

Résumons la stratégie sur un diagramme:

$$\begin{array}{ccc} & \text{homogénéisation} & \\ F & \longrightarrow & F^h \\ \updownarrow\ ? & & \downarrow \\ G_h & \longleftarrow & G := \texttt{Groebner}(F^h) \\ & \text{déshomogénéisation} & \end{array}$$

Remarque 5. *La base de Gröbner ainsi obtenue n'est pas réduite. On peut dire que cette différence $G_h \backslash G$ constitue des solutions parasites (ou encore "solutions à l'infini") correspondant à $h = 0$.*

7 Que peut on lire sur une base de Gröbner ?

7.1 Nombre fini de solutions

Si $(f_1,\ldots,f_m)$ est un système d'équations, on lui associe I l'idéal $\mathrm{Id}(f_1,\ldots,f_m)$ et on calcule une base de Gröbner réduite G de I pour un ordre admissible $<$ fixé (mais pas nécessairement l'ordre lexicographique).

On peut détecter immédiatement si le système admet des solutions:

Proposition 30. *Le système admet des solutions dans la clôture algébrique de $\mathbb{K}$ si et seulement si G n'est pas $\{1\}$.*

il est très facile de détecter si l'ensemble des solutions est fini.

Lemme 1. *On suppose que $\mathbb{L}$ est algébriquement clos. On note $\pi_i : \begin{pmatrix} \mathbb{L}^n & \longrightarrow & \mathbb{L} \\ (x_1,\ldots,x_n) & \mapsto & x_i \end{pmatrix}$ est la ième projection et $\pi_i(V_{\mathbb{L}}(I))$ est fini alors il existe un polynôme en une variable $p \in \mathbb{K}[X]$ tel que $p(x_i) \in I$.*

Le corollaire suivant est immédiat d'après la définition d'une base de Gröbner et permet de lire directement la propriété sur n'importe quelle base de Gröbner:

Corollaire 4. *Si $\pi_i(V_{\mathbb{L}}(I))$ est fini et G une base de Gröbner de I alors il existe $p \in G$ tel que $\mathrm{LT}(p) = x_i^k$ pour un certain k.*

La réciproque est fausse en général sauf si on suppose que l'escalier est fini (c'est à dire sur le dessin que les points qui se trouvent sous l'escalier sont en nombre fini). La preuve du lemme suivant est constructive et permet de construire un polynôme univarié:

Théorème 31. *Si pour tout $i \in \{1, \ldots, n\}$, $\mathrm{LT}(p_i) = x_i^{k_i}$ où $p_i \in G$ alors $V_{\mathbb{L}}(I)$ est fini. Autrement dit, le système admet un nombre fini de solutions.*

Théorème 32. *Soit D le nombre de monômes sous l'escalier. Si D est fini, D est le nombre de racines dans la clôture algébrique comptée avec multiplicité.*

7.2 Fonction et série de Hilbert. Dimension. Degré.

La fonction de Hilbert d'un idéal I regroupe des propriétés combinatoires et géométriques associées à cet idéal. Cette fonction est une donnée intrinsèque de l'idéal, et ne dépend pas, en particulier, du système de générateurs choisi. Nous rappelons la définition et les propriétés essentielles de la fonction de Hilbert (pour plus de détails voir (Cox *et al.* , 2007)). La fonction est un outil essentiel pour estimer la complexité des calculs et en particulier la taille des matrices générées par les algorithmes de type F_4 et F_5.

Pour $d \in \mathbb{N}$ l'ensemble $\mathbb{K}[x_1, \ldots, x_n]_d = \{f \in \mathbb{K}[x_1, \ldots, x_n] \mid \deg(f) = d\}$ est un $\mathbb{K}$ espace vectoriel de dimension $\binom{n+d-1}{d}$. Si I est un idéal, alors $I_d = I \cap \mathbb{K}[x_1, \ldots, x_n]_d$ est aussi un $\mathbb{K}$ espace vectoriel.

Définition 17. *La fonction de Hilbert d'un idéal homogène $I = \mathrm{Id}\,(f_1, \ldots, f_m)$ en degré d est définie par*

$$\mathrm{HF}_I(d) = \mathrm{HF}(d) = \dim(\mathbb{K}[x_1, \ldots, x_n]/I)_d = \dim(\mathbb{K}[x_1, \ldots, x_n]_d) - \dim(I_d)$$

Théorème 33. *(Hilbert) À partir d'un certain degré d_0 il existe un polynôme P tel que*

$$\mathrm{HF}_I(d) = P(d) \text{ pour } d \geqslant d_0$$

d_0 est appelé la régularité de Hilbert, ou indice de régularité; on le note noté $H(I)$.
Le degré de P est la dimension de l'idéal et noté $\dim(I)$.

Définition 18. *La série de Hilbert est la série génératrice de HF_I:*

$$\mathrm{HS}_I(z) = \sum_{d \geqslant 0} \mathrm{HF}_I(d)\, z^d$$

d'après le théorème 33 c'est une fraction rationnelle, qui peut s'écrire

$$\frac{N(z)}{(1-z)^d} \text{ avec } N(1) \neq 0$$

où d est la dimension *de I et $N(1)$ est le* degré de l'idéal I *(noté aussi $\deg(I)$).*

Remarque 6. *Lorsqu'on voudra parler de la dimension d'un $\mathbb{K}$ espace vectoriel E on utilisera la notation $\dim_{\mathbb{K}}(E)$ afin de distinguer la dimension d'un idéal $\dim(I)$.*

7.3 Calcul de la dimension d'un idéal

À partir de la fonction de Hilbert d'un idéal I on peut calculer la *dimension* et le *degré* de l'idéal. Intuitivement la dimension d'un idéal premier ou d'une variété algébrique est le nombre de "paramètres libres". Pour un idéal I quelconque, la dimension de I est le maximum des dimensions dans une décomposition en idéaux premiers. De façon pratique:

Proposition 34. *Si I est un idéal et G une base de Gröbner, alors $\dim(I) = \dim(\mathrm{LT}(G)) = \dim(\sqrt{\mathrm{LT}(G)})$.*

Comme de plus la dimension d'un idéal est égale a la dimension de l'idéal radical on peut se contenter de calculer $\dim(\mathrm{flat}(\mathrm{LT}(G)))$ où $\mathrm{flat}(x_1^{\alpha_1}\cdots x_n^{\alpha_n}) = x_1^{\beta_1}\cdots x_n^{\beta_n}$ où $\beta_i = 1$ si $\alpha_i \neq 0$ et $\beta_i = 0$ si $\alpha_i = 0$.

L'exemple suivant est un exemple classique dans la résolution des équations algébriques:

$$\text{(Cyclic } n) \; C_n \begin{cases} C_{n,0} = 0 \\ \cdots \\ C_{n,k} = 0 \\ \cdots \\ C_{n,n-2} = 0 \\ C_{n,n-1} = n \end{cases} \quad \text{avec } C_{n,k} = \sum_{i=0}^{n-1} \prod_{j=0}^{k} x_{(i+j) \bmod n}$$

Exemple: Cyclic 4 Soit G_4 la base de Gröbner de C_4 pour l'ordre DRL sur $[a, b, c, d]$.

$$\mathrm{LT}(G) = [c^2d^4, c^3d^2, bd^4, bcd^2, bc^2, b^2, a]$$

on calcule le radical:

$$\mathrm{Id}(\sqrt{\mathrm{LT}(G)}) = \mathrm{Id}(cd, cd, bd, bcd, bc, b, a) = \mathrm{Id}(cd, a, b) = \mathrm{Id}(d, a, b) \cap \mathrm{Id}(c, a, b)$$

et le système est de dimension 1.

7.4 Suites régulières. Degré de régularité

Considérons un système algébrique $F = [f_1, \ldots, f_m]$ auquel on applique un algorithme de calcul de base de Gröbner comme F_4 (voir la section 9) qui utilise l'algèbre linéaire pour effectuer les calculs: dire qu'une matrice générée par cet algorithme n'est pas de rang plein est équivalent à dire que les lignes de cette matrice ne sont pas indépendantes. Comme chaque ligne de la matrice est un produit $t \times f$ où t est un terme et $f \in F$, la dépendance linéaire s'exprime sous la forme $\sum_{f\in F, t\in T} \lambda_{t,f}\ t\, f = 0$ ou en regroupant les termes:

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

où les g_i sont des polynômes de $\mathbb{K}[x_1, \ldots, x_n]$. On dit encore que $(g_1, \ldots, g_m)$ est une syzygie. On peut aussi écrire la relation (5) sous la forme:

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

autrement dit on a un *diviseur de zéro* (*si* $g_1 \neq 0$).

Pour un système linéaire on dit que le système est non singulier s'il n'existe pas de combinaison linéaire non nulle

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

mais pour les systèmes algébriques il n'est pas possible d'imposer la non existence de relations (5) non nulles: en effet il existe toujours des relations

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

qui sont des syzygies triviales; ainsi il est naturel de dire définir qu'un système est régulier s'il n'existe pas de relation de type (5) hormis les relations triviales (voir aussi la définition 19 de la section 2):

Définition 19. *Définition géométrique: le système* $(f_1, \ldots, f_m)$ *de polynômes* homogènes *est régulier si pour tout* $i \in \{1, \ldots, m\}$*, la dimension de* $\langle f_1, \ldots, f_i \rangle$ *est* $n - i$*. On dit encore que la suite* $(f_1, \ldots, f_m)$ *est régulière.*

Définition algébrique: le système $(f_1, \ldots, f_m)$ *de polynômes* homogènes *est régulier si pour tout* $i = 1, \ldots, m$ *et* g *tel que*

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

alors g *est aussi dans* $\langle f_1, \ldots, f_{i-1} \rangle$*.*

Le système $(f_1, \ldots, f_m)$ *de polynômes (pas nécessairement homogènes) est régulier si le système* $(f_1^h, \ldots, f_m^h)$ *l'est (* f_i^h *est la partie homogène de plus haut degré de* f_i*).*

Remarque 7. *On peut aussi caractériser les suites régulières en disant qu'il n'existe pas de relations algébriques non nulles de la forme:*

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

hormis les relations induites par les relations triviales $f_i f_j = f_j f_i$*.*

Remarque 8. *En utilisant la définition géométrique, il est facile de voir qu'il n'existe pas de système régulier lorsque* $m > n$*; la définition 40 est adaptée au cas des systèmes sur-déterminés (la semi-régularité) de la section 14.*

Pour une suite régulière on peut calculer explicite ment la série de Hilbert, la dimension de la variété algébrique associée qui est nécessairement $n - m$. Les suites régulières sont également caractérisées par l'absence de diviseurs de zéro : une suite est régulière si le i-ème polynôme f n'est pas diviseur de zéro dans $\mathbb{K}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_m \rangle$. Les propriétés suivantes permettent de caractériser les suites régulières ((Cox *et al.* , 1998; Lang, 2002; Fröberg, 1997)) et leur comportement par rapport à un calcul de base de Gröbner est parfaitement connu:

Théorème 35. *1.* $(f_1, \ldots, f_m)$ *est régulier si et seulement si sa série de Hilbert (voir définition 18) est égale à:*

$$\mathrm{HS}_{\langle f_1, \ldots, f_m \rangle}(z) = \sum_{d \geqslant 0} \mathrm{HF}_{\langle f_1, \ldots, f_m \rangle}(d)\, z^d = \frac{\prod_{i=1}^{m} (1 - z^{d_i})}{(1 - z)^n}$$

2. après un changement linéaire générique des variables le degré des polynômes d'une base de Gröbner pour un ordre DRL est borné par le l'index de régularité:

$$\textit{Borne de Macaulay: } \sum_{i=1}^{m} (d_i - 1) + 1$$

Proof. Pour la preuve de (i) voir (Lang, 2002, Theorem 6.6 p. 436); ou (Fröberg, 1997, p. 137). La preuve de l'assertion (ii) se trouve dans (Lazard D., 1983; Giusti, 1994). □

Remarque 9. *L'algorithme* F_5 *11 permettra de faire un calcul de base de Gröbner sans calcul inutile lorsque le système algébrique de départ est régulier.. Les suites régulières seront au coeur de la section 14.*

8 Algorithme FGLM

L'algorithme FGLM (le nom est la réunion des initiales des auteurs de l'algorithme) a été publié dans (Faugère, J.C., Gianni, P., Lazard, D. and Mora T., 1993). Des versions plus efficaces (Faugère & Mou, 2011; Faugère *et al.* , 2014b) ont été publiées récemment.

8.1 Introduction.

L'algorithme FGLM permet de ramener à un calcul d'algèbre linéaire l'opération de changer l'ordre d'une base de Gröbner d'un idéal zéro dimensionnel. Un avantage théorique de cet algorithme est de pouvoir dériver une estimation très précise de la complexité de cet algorithme; de plus, cela permet d'améliorer la complexité du calcul d'une base de Gröbner pour n'importe quel ordre: par exemple en utilisant les résultats de (Caniglia L. and Galligo A. and Heintz J., 1988; Caniglia L. and Galligo A. and Heintz J., 1991) on peut montrer que pour l'ordre DRL, la complexité du calcul d'une base de Gröbner d'un système algébrique d'un système de n équations en n variables est borné par $C_1\, d^{C_2\, n^2}$ (où C_1, C_2 sont des constantes). Avec l'hypothèse supplémentaire que le système admet un nombre fini de solutions à l'infini cette complexité est majorée par $C_1\, d^{C_2\, n}$ (voir (Lazard D., 1983)). En pratique, le calcul des bases de Gröbner est souvent plus rapide mais on constate que c'est souvent avec l'ordre DRL que le calcul est le plus rapide. Pour l'ordre lexicographique la combinaison de l'algorithme FGLM et de ces résultats permettent de garder la même complexité: par exemple ceci permet d'améliorer la borne $C_1\, d^{C_2\, n^3}$ ((Caniglia L. and Galligo A. and Heintz J., 1988)) en $C_1\, d^{C_2\, n^2}$. La section 14 contient d'autres résultats de complexité dans le cas où le système initial est régulier ou semi-régulier.

Pour appliquer l'algorithme FGLM il suffit d'une base de Gröbner calculée pour un ordre admissible; à partir de cette base on calcule des matrices de multiplications par les variables (voir la section 8.4 8 et l'algorithme 8). À noter que ces matrices peuvent être utilisées pour calculer numériquement les racines d'un système algébrique ((Auzinger and Stetter H., 1998; Möller H.M., 1993)): les valeurs propres des matrices de multiplications donnent les (projections) des solutions du système. Symboliquement, le plus petit polynôme de la base de Gröbner pour l'ordre lexicographique est aussi le polynôme minimal de la matrice de multiplication par la plus petite variable.

L'algorithme FGLM est implanté dans tous les systèmes de Calcul Formel (Mathematica, Maple, Magma, Singular, ...) et l'expérience a montré l'efficacité de cet algorithme par rapport à un calcul direct.

Pour simplifier l'écriture des algorithmes on écarte également le cas trivial où $I = \mathbb{K}[x_1, \ldots, x_n]$.

8.2 Espace vectoriel quotient. Idée de l'algorithme.

On considère uniquement un idéal I zéro dimensionnel (voir le théorème 31 pour un moyen algorithmique de vérifier qu'un idéal est zéro dimensionnel).

On suppose qu'on connaît une base de Gröbner G de l'idéal I pour un certain ordre (par exemple l'ordre DRL); alors l'application

$$\varphi : p \longmapsto \textsc{NormalForm}(p, G)$$

est linéaire et son noyau est $\ker(\varphi) = I = \mathrm{Id}(G)$.

On peut définir une relation d'équivalence $p \equiv q$ si et seulement $\varphi(p) = \varphi(q)$.

On définit ensuite la classe d'équivalence d'un polynôme p par $\overline{p} = \{q \in \mathbb{K}[x_1, \ldots, x_n] \mid \varphi(q) = \varphi(p)\}$. L'ensemble de ces classe d'équivalences constitue l'idéal quotient $E = \mathbb{K}[x_1, \ldots, x_n]/I = \{\overline{p} \mid p \in \mathbb{K}[x_1, \ldots, x_n]\}$. Une conséquence du théorème 31 est que cet espace vectoriel est un espace vectoriel de dimension finie $D = \deg(I)$.

Pour donner une idée de l'algorithme supposons qu'on cherche à calculer un polynôme en une variable, disons x_i, dans l'idéal: on considère alors les éléments suivants de $\mathbb{E}$:

$$\overline{1}, \overline{x_i}, \overline{x_i^2}, \cdots, \overline{x_i^D}$$

Comme E est de dimension D on sait que ces vecteurs ne sont pas linéairement indépendants: il existe $(\lambda_i)_{i=0,\ldots,D}$ de $\mathbb{K}$ non tous nuls tels que:

$$\lambda_0 \overline{1} + \lambda_1 \overline{x_i}, \overline{x_i^2} + \cdots + \lambda_D \overline{x_i^D} = 0$$

autrement dit on a trouvé le polynôme $P_i(x_i) = \sum_{j=0}^{D} \lambda_j x_i^j$ tel que $\overline{P_i} \equiv 0$ c'est à dire $P_i \in I$.

Pour l'évaluation de la complexité d'un tel algorithme la principale difficulté est de compter le nombre d'opérations pour le calcul de

$$\varphi(1), \varphi(x_i), \varphi\left(x_i^2\right), \cdots, \varphi\left(x_i^D\right)$$

En fait on remarque qu'on peut toujours écrire le calcul de la forme normale x_i^k comme: $\varphi\left(x_i^k\right) = \varphi\left(x_i\, \varphi\left(x_i^{k-1}\right)\right)$; on peut calculer incrémentalement ces formes en se contant des opérations élémentaires suivantes:

$$p \longmapsto \psi(p) = \varphi(x_i\, p)$$

8.3 Escalier. Frontière d'un idéal.

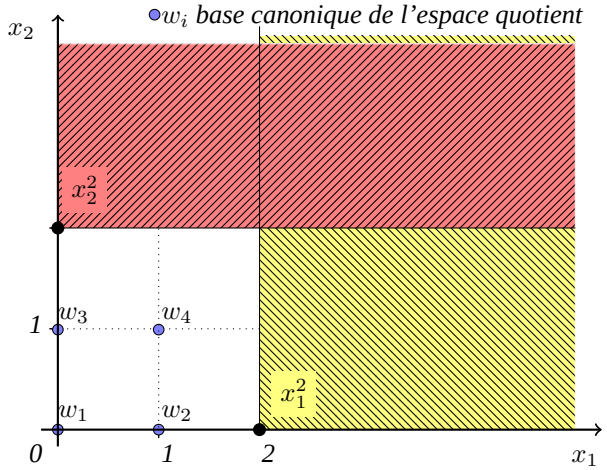
Définition 20. *Étant donné un idéal de dimension zéro I dans $\mathbb{K}[x_1, \ldots, x_n]$ et $(G, <)$ une base de Gröbner de I, on considère l'ensemble*

$$\mathcal{E}_<(G) = \{t \in T \mid t \text{ n'est pas réductible par } G\} \text{ trié pour l'ordre } <.$$

On dit que $\mathcal{E}(G)$, l'escalier de I, est une base canonique par rapport à G du $\mathbb{K}$-espace vectoriel $\mathbb{K}[x_1, \ldots, x_n]/I$.On note $\deg(I)$ la dimension du $\mathbb{K}$-espace vectoriel $\mathbb{K}[x_1, \ldots, x_n]/I$: c'est donc le degré de l'idéal I (voir définition 18 p. 18) et le cardinal de la base canonique $\mathcal{E}(G)$. On note

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

Exemple 5. $G_{<_{DRL}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ *base de Gröbner pour l'ordre DRL avec $x_2 > x_1$.*
escalier $\mathcal{E}_<(G) = \{t \in T \mid t \text{ n'est pas réductible par } G\} = \{w_1 = 1, w_2 = x_1, w_3 = x_2, w_4 = x_1\, x_2\}$



$\{\overline{w_1}, \overline{w_2}, \overline{w_3}, \overline{w_4}\}$ *est une base de l'espace vectoriel quotient $\mathbb{K}[x_1, \ldots, x_n]/I$.*
Dans cette base le polynôme $-x_1^2 - x_2^2 + 7\,x_1\,x_2 \longrightarrow -3\,x_1 - 2\,x_2 + 7\,x_1\,x_2$ est le vecteur $[0, -3, -2, 7]$.

L'escalier $\mathcal{E}_<(G)$ est clos pour la division:

Proposition 36. *Si $1 \neq e \in \mathcal{E}_<(G)$ alors pour tout i tel que x_i divise e on a $\frac{e}{x_i} \in \mathcal{E}_<(G)$.*

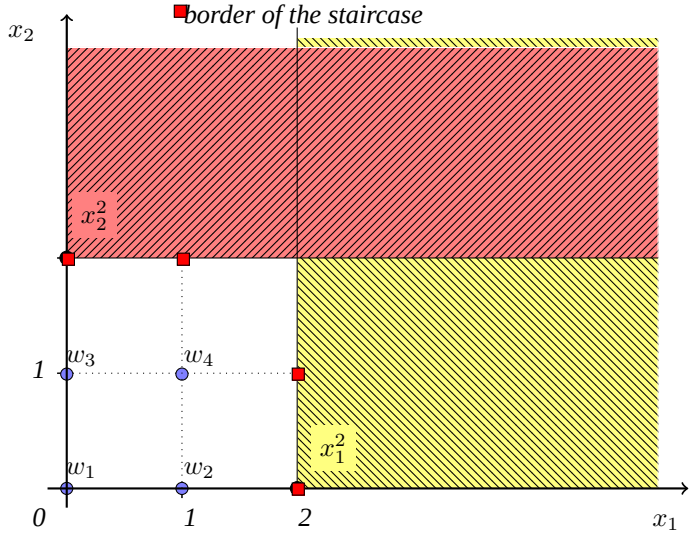
Comme expliqué dans la section 8.2 on cherche à calculer incrémentalement les calculs de formes normales en se restreignant à des opérations élémentaires du type $\varphi(x_i p)$ où $e \in E$; on peut donc supposer que $p = \sum_{i=1}^{\deg(I)} \lambda_i w_i$ et on donc calculer des formes normales de la forme $\varphi(x_i e)$ où $e \in \mathcal{E}(G)$. Le cas où $x_i e \in \mathcal{E}(G)$ est trivial car $\varphi(x_i e) = x_i e$ sans calcul. Reste le cas où $x_i e \notin \mathcal{E}_<(G)$. Pour estimer le nombre de tels éléments on introduit la notion de frontière, c'est à dire les points qui sont à distances 1 de l'escalier:

Définition 21. *Soit $\mathcal{E}_<(G)$ la base canonique de $\mathbb{K}[x_1,\ldots,x_n]/I$, alors*

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

est la frontière de G.

Exemple 6. $G_{<_{DRL}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ *base de Gröbner pour l'ordre DRL avec $x_2 > x_1$.*
frontière $\mathcal{F}_<(G) = \{x_i e \mid e \in \mathcal{E}(G),\ 1 \leqslant i \leqslant n \text{ et } x_i\, e \notin \mathcal{E}_<(G)\}$



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

On peut caractériser les éléments de $\mathcal{F}_<(G)$:

Proposition 37. *Si I est un idéal de dimension zéro, $(G,<)$ une base de Gröbner réduite par rapport à un ordre admissible $<$, alors pour tout élément $t \in \mathcal{F}(G)$ alors*

1. *ou il existe $g \in G$ tel que $t = \mathrm{LT}(g)$*
2. *ou il existe j et $t' \in \mathcal{F}_<(G)$ tel que $t = x_j t'$.*

Proof. On fixe $t \in \mathcal{F}_<(G)$ et considère l'ensemble $A_t = \{1 \leqslant j \leqslant n$ tel que x_j divise t et $\frac{t}{x_j} \notin \mathcal{E}(G)\}$.

1. Si A_t est vide. Comme $t \notin \mathcal{E}_<(G)$ il existe $g \in G$ tel que $\mathrm{LT}(g)$ divise t: soit $u = \frac{t}{\mathrm{LT}(g)}$. S'il existait x_j divisant u alors en posant $v = \frac{u}{x_j}$ on aurait $\frac{t}{x_j} = \mathrm{LT}(g)\, v \notin \mathcal{E}(G)$ et donc $j \in A_t$ ce qui est absurde. Donc $u = 1$ et $\mathrm{LT}(g) = t$.
2. Si A_t n'est pas vide il existe donc j tel que x_j divise t et $t' = \frac{t}{x_j} \notin \mathcal{E}_<(G)$. Comme $t \in \mathcal{F}_<(G)$ il existe $e \in \mathcal{E}_<(G)$ tel que: $t = x_i\, e$. Comme $x_i\, e = t = x_j\, t'$, l'égalité $i = j$ est impossible car cela impliquerait $t' = e \in \mathcal{E}_<(G)$. Donc $i \neq j$ et x_j divise e; de plus $e' = e/x_j$ (propriété de stabilité) est dans $\mathcal{E}_<(G)$; ainsi $t' = x_i \cdot e' \in \mathcal{F}_<(G)$ et t est de la forme $t = x_j\ t'$.

□

Corollaire 5. *Le nombre de générateurs d'une base de Gröbner réduite d'un idéal zéro dimensionnel I est inférieur à $n \deg(I)$.*

Proof. Si $g \in G$, alors $\mathrm{LT}(g) \notin \mathcal{E}_<(G)$ et pour tout k tel que x_k divise $\mathrm{LT}(g)$ on a $\frac{\mathrm{LT}(g)}{x_k} \in \mathcal{E}(G)$; donc $\mathrm{LT}(g) \in \mathcal{F}(G)$. La borne découle du fait $\mathrm{LT}(G) \subseteq \mathcal{F}(G)$. □

Remarque 10. *On verra une meilleure borne (théorème 9) lorsque le système est générique (plus exactement en position de Noether simultanée) et que $\deg(I)$ est égale à la borne de Bezout.*

8.4 Construction des matrices de multiplication

Dans la suite on travaille dans l'espace vectoriel $\mathbb{K}[x_1, \ldots, x_n]/I$ et on considère la base canonique pour l'ordre $<$ associée à G:

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

Pour trouver l'expression d'un polynôme f dans cette base on doit calculer $\mathrm{NF}(f, G)$ en utilisant la procédure 2: cependant il est difficile d'obtenir une borne de complexité précise en théorie ou en pratique. C'est la raison pour laquelle on va ramener ce calcul un produit matrice vecteur dont il sera facile de borner la complexité.

Dans la suite on suppose qu'on connaît une fonction linéaire

$$\varphi_I : \begin{pmatrix} \mathbb{K}[x_1, \ldots, x_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]/I \\ p & \longmapsto & \overline{p} \end{pmatrix}$$

qui soit une forme normale, c'est à dire qui vérifie les propriétés:

$$\varphi_I(p) = 0 \text{ si et seulement si } p \in I$$
$$\varphi_I(p \cdot q) = \varphi_I(p \cdot \varphi_I(q)) = \varphi_I(\varphi_I(p) \cdot \varphi_I(q))$$

Un moyen pour se donner une telle forme normale est de calculer une base de Gröbner G de I pour l'ordre admissible $<$ et ensuite de prendre $\varphi_I(p) = \mathrm{NF}(p, G, <)$. Dans ce cas on sait que le noyau de φ_I est égal à I: $\ker(\varphi_I) = I$. Une solution alternative est d'utiliser les formes normales généralisées. Dans la suite on note $\varphi_I(p) = \mathrm{NF}(p)$ cette fonction de forme normale.

Pour optimiser le calcul on va exploiter la structure de $\mathcal{F}_<(G)$ donnée par la proposition 37: on va calculer uniquement des formes normales de la forme $\varphi_I(x_i \cdot p)$ où p est déjà réduit. On considère donc les applications linéaires de multiplication par une variable :

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

Définition 22. *On définit les matrices de multiplication par une variable: pour tout* $1 \leqslant k \leqslant n$ *on définit la matrice* $M^{(k)}$ *de taille* $\deg(I) \times \deg(I)$ *telle que:*

$$M^{(k)}_{i,j} = \textit{ le coefficient } w_i \textit{ dans } x_k\, w_j$$

Afin de décrire l'algorithme permettant de calculer les matrices de multiplication par une variable, on utilise les notations: $\delta_{i,j}$ est le symbole de Kronecker et vaut 1 si $i = j$ et 0 sinon; si M est une matrice alors $\mathrm{Col}(M, j)$ désigne la $j$ème colonne de M.

Algorithme 8. *Matrices de multiplications*

Input: G *une base de Gröbner réduite pour l'ordre* $<$
$\mathcal{E}(G) = \{w_1 = 1 < w_2 < \cdots < w_{\deg(I)}\}$ *base canonique pour* G.
$N := []$ *// une table de polynômes indexée par* T
// et vérifiant pour tout $t \in T : N[t] = \mathrm{NF}(t, G, <)$
for i ***from*** 1 ***to*** $\deg(I)$ ***do***
  $N[w_i] := w_i$
  for each k *tel que* $w_i = x_k\, w_j$ ***do***
    $M^{(k)}_{l,j} := \delta_{l,i}$ *pour tout* $l \in \{1, \ldots, n\}$
$F := [x_j\, w_i$ *pour* $j = 1, \ldots, n$, $i = 1, \ldots, \deg(I)]$
trier F *pour* $<$, *éliminer les doublons et les éléments de* $\mathcal{E}(G)$*:*
for t ***in*** F ***do***
  if t *est un multiple strict d'un terme de tête de* G
    $t = x_j\, t'$ *avec* $t' < t$
    On a déjà calculé $N[t'] = \sum_{i=1}^{s} \mu_i w_i$ *avec* $\mu_i \in \mathbb{K}$ *et* $w_s < t'$
    $N[t] = \sum_{i=1}^{s} \mu_i \,\mathrm{Col}(M^{(j)}, i) = \sum_{i=1}^{\deg(I)} \lambda_i w_i$
    for each k *tel que* $t = x_k\, w_l$ *pour un certain* l ***do***
      $M^{(k)}_{i,j} := \lambda_i$ *pour tout* $i \in \{1, \ldots, n\}$
  else
    il existe $g = t + \sum_{i=1}^{\deg(I)} \lambda_i w_i$ *et* $\lambda_i \in \mathbb{K} \in G$ *tel que* $t = \mathrm{LT}(g)$
    $N[t] := -\sum_{i=1}^{\deg(I)} \lambda_i\, w_i$
    for each k *tel que* $t = x_k\, w_j$ *pour un certain* j ***do***
      $M^{(k)}_{i,j} := -\lambda_i$ *pour tout* $i \in \{1, \ldots, n\}$
return $M^{(k)}$ *// matrice de multiplication par* x_k

Théorème 38. *L'algorithme 8 calcule les matrices* $M^{(k)}$ *et la complexité arithmétique est bornée par* $O(n\, \deg(I)^3)$.

Proof. Pour montrer la correction de l'algorithme il suffit de montrer que dans l'expression $N[t] = \sum_{i=1}^{s} \mu_i \,\mathrm{Col}(M^{(j)}, i)$ la colonne i de la matrice $M^{(j)}$ a déjà été calculée auparavant. Comme $N[t'] = \sum_{i=1}^{s} \mu_i w_i$ avec $w_s < t'$ on a $\mathrm{NF}(t) = \mathrm{NF}(x_j\, t') = \sum_{i=1}^{s} \mu_i\ \mathrm{NF}(x_j\, w_i)$, comme l'ordre est admissible $x_j w_i < x_j w_s < x_j\, t' = t$. Donc tous les termes $x_j\, w_i$ ont été traités et la colonne i de la matrice $M^{(j)}$ a donc été remplie.

Dans l'algorithme il est clair que F est la frontière $\mathcal{F}(G)$ et donc le nombre d'itérations dans la boucle principale est bornée par $n\, D\,(I)$; de plus le calcul de $\sum_{i=1}^{s} \mu_i \,\mathrm{Col}(M^{(j)}, i)$ nécessite au plus $s\ \deg(I) \leqslant \deg(I)^2$ opérations. □

8.5 Description de l'algorithme FGLM

Dans l'algorithme suivant si S est une liste alors:

- $\#S$ désigne le nombre d'éléments de S
- $\mathrm{first}(S)$ est le premier élément de la liste ou $\varnothing$ si S est vide.

Algorithme 9. *FGLM*

Input: $<_2$ *un ordre admissible et* NF *une forme normale.*
Output: *base de Gröbner réduite de l'idéal* I *pour* $<_2$
où $I = \{f \in \mathbb{K}[x_1, \ldots, x_n] \mid \mathrm{NF}(f) = 0\}$
$L := [\,]$ *// liste des prochains termes à étudier*
$S := [\,]$ *// l'escalier pour le nouvel ordre* $<_2$
$V := []$ *//* $V = \mathrm{NF}(S)$
$G := [\,], t := 1$
infinite loop
$v := \mathrm{NF}(t)$ *et* $s := \#S$ *le nombre d'éléments de* S.
if $v \in \mathrm{Vect}_{\mathbb{K}}(V)$ ***then***
on peut trouver (λ_i) *t.q.* $v = \sum_{i=1}^{s} \lambda_i \cdot V_i$
$G := G \cup \left[v - \sum_{i=1}^{s} \lambda_i \cdot S_i \right]$
else
$S := S \cup [t]$ *et* $V := V \cup [v]$
$L := \mathrm{Sort}(L \cup [x_i\, t \mid i = 1, \ldots, n], <_2)$
Éliminer de L *les doublons et les multiples de* $\mathrm{LT}(G)$
if $L = \varnothing$ *then*
return G
$t := \mathrm{first}(L)$ *et supprime* t *de* L.

Le théorème 39 prouve que cet algorithme se termine et retourne le bon résultat.

8.6 Version matricielle de FGLM

Afin de rendre explicite la détection de la dépendance linéaire des vecteurs du $\mathbb{K}$-espace vectoriel $\mathbb{K}[x_1, \ldots, x_n]/I$ on introduit la matrice de passage P entre l'ancienne base $\mathcal{E}(G) = \left\{w_1 = 1 < w_2 < \cdots < w_{\deg(I)}\right\}$ et la nouvelle base S en cours de construction. Si on note $S = \left[\varepsilon_1, \ldots, \varepsilon_{\deg(I)}\right]$ cette base alors à tout moment de l'algorithme

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

Initialement $S = [w_1]$, et on construit incrémentalement des vecteurs $v = \varphi(x_k\, w) = M^{(k)} \cdot w$ où v, w sont des vecteurs exprimés dans la base $\mathcal{E}(G)$:

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

Pour tester l'indépendance linéaire il suffit de calculer:

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

1. si $\lambda_{s+1} = \cdots = \lambda_{\deg(I)} = 0$ où s est le nombre d'éléments de S alors le vecteur v appartient au $\mathbb{K}$-espace vectoriel engendré par S.

2. s'il existe $k > s$ tel que $\lambda_k \neq 0$ alors $\varepsilon_{s+1} = \lambda$ est un vecteur linéairement indépendant On calcule une nouvelle matrice P' tel que:
$$P' \cdot v =^T [0, \ldots, 0, 1, 0, \ldots, 0] = \varepsilon_{s+1} \tag{9}$$

   La procédure UPDATE met à jour la matrice P pour que l'équation (9) soit vérifiée:

Algorithme 10. UPDATE *Mise à jour matrice de passage P*

Input: *$s \in \mathbb{N}$, le vecteur λ, la matrice P*
Output*: la matrice P mise à jour*
$k := \min\{j > s$ *tel que* $\lambda_j \neq 0$ *}*
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*** $\deg(I)$ *tel que* $i \neq s+1$ ***do***
      $P_{i,j} := P_{i,j} - \alpha\,\lambda_i$
return P

On peut donc maintenant décrire explicitement l'algorithme FGLM:

Algorithme 11. *Version matricielle de FGLM*

Input*: $<$ un ordre, $M^{(k)}$ les matrices de multiplications, φ forme normale*
Output*: base de Gröbner réduite pour l'ordre $<$ de* $\ker(\varphi)$.
$S := [1]$ *// l'escalier pour le nouvel ordre $<$.*
$V := [w_1]$ *// $V = \mathrm{NF}(S)$*
$L := [(i,1), i = 1, \ldots, (n-1)]$ *// liste de paires (k,l) correspondant à $x_i \cdot S_l$*
$G := [\,], t := (n,1)$
$P := I_{\deg(I)}$ *matrice de passage entre la nouvelle base S et $\mathcal{E}(G)$*
infinite loop
 $s := \#S$ *le nombre d'éléments de S.*
 $t = (k,l)$*: on calcule $v = M^{(k)} \cdot V_l$ puis $\lambda = P \cdot v$*
 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)
  $S := S \cup [x_k S_l]$ *et* $V := V \cup [v]$
  $L := \mathrm{Sort}(L \cup [(i,s) \mid i = 1, \ldots, n], <)$
  Éliminer de L les doublons et les multiples de $\mathrm{LT}(G)$
  if $L = \varnothing$ then
   return G
  $t := \mathrm{first}(L)$ *et supprime t de L.*

8.7 Exemple pas à pas

On se place dans $\mathbb{Q}[x_1, x_2]$ et soit $G_{<_{\mathrm{DRL}}} = \left[x_1^2 - 3\,x_2 - x_1 + 1, x_2^2 - 2\,x_1 + x_2 - 1\right]$ qui est une base de Gröbner pour l'ordre DRL avec $x_2 > x_1$ et on chercher à calculer la base pour l'ordre lexicographique avec $x_2 > x_1$.

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

On obtient les matrices de multiplication par x_1 et par x_2:

$$M^{(1)} = \begin{array}{c|cccc} & x_1\,w_1 & x_1\,w_2 & x_1\,w_3 & x_1\,w_4 \\ w_1 & 0 & -1 & 0 & 3 \\ w_2 & 1 & 1 & 0 & 6 \\ w_3 & 0 & 3 & 0 & -4 \\ w_4 & 0 & 0 & 1 & 1 \end{array}, M^{(2)} = \begin{array}{c|cccc} & x_2\,w_1 & x_2\,w_2 & x_2\,w_3 & x_2\,w_4 \\ w_1 & 0 & 0 & 1 & -2 \\ w_2 & 0 & 0 & 2 & 3 \\ w_3 & 1 & 0 & -1 & 6 \\ w_4 & 0 & 1 & 0 & -1 \end{array}$$

On commence par $L := [(2,1)], S := [1], V := [w_1], G := [\,], t := (1,1)$ correspondant au monôme $1 \cdot S_1 = 1, P := I_4$

Étape 1: On calcule $v := M^{(1)} \cdot V_1 = M^{(1)} \cdot 1 = w_2$ puis $\lambda = P \cdot v = w_2$
comme $\lambda_2 \neq 0$, $S := [1, x_1]$, $V := [w_1, w_2]$ et la matrice P reste inchangée, puis $L := [(1,2),(2,1),(2,2)]$.
Étape 2: $t = (1,2)$. On calcule $v := M^{(1)} \cdot V_2 = M^{(1)} \cdot w_2 =^T [-1,1,3,0]$ puis $\lambda = P \cdot v =^T [-1,1,3,0]$
comme $\lambda_3 \neq 0$, $S := [1, x_1, x_1^2]$, $V := [w_1, w_2, ^T[-1,1,3,0]]$, puis $P := \begin{bmatrix} 1 & 0 & 1/3 & 0 \\ 0 & 1 & -1/3 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $L := [(1,3),(2,1),(2,2),(2,3)]$.
Étape 3: $t = (1,3)$. On calcule $v := M^{(1)} \cdot V_3 = M^{(1)} \cdot^T [-1,1,3,0] =^T [-1,0,3,3]$ puis $\lambda = P \cdot v =^T [0,-1,1,3]$
comme $\lambda_4 \neq 0$, $S := [1, x_1, x_1^2, x_1^3]$, $V := [w_1, w_2, V_3, ^T[-1,0,3,3]]$, puis $P := \begin{bmatrix} 1 & 0 & 1/3 & 0 \\ 0 & 1 & -1/3 & 1/3 \\ 0 & 0 & 1/3 & -1/3 \\ 0 & 0 & 0 & 1/3 \end{bmatrix}$, $L := [(1,4),(2,1),(2,2),(2,3),(2,4)]$.
Étape 4: $t = (1,4)$. On calcule $v := M^{(1)} \cdot V_4 = M^{(1)} \cdot^T [-1,0,3,3] =^T [9,17,-12,6]$ puis $\lambda = P \cdot v =^T [5,23,-6,2]$
comme $\lambda_5 = 0$, $G := [x_1^4 - 2\,x_1^3 + 6\,x_1^2 - 23\,x_1 - 5]$, puis $L := [(2,1),(2,2),(2,3),(2,4)]$.
Étape 5: $t = (x_2, 1)$. On calcule $v := M^{(2)} \cdot V_1 = M^{(2)} \cdot w_1 = w_3$ puis $\lambda = P \cdot w_3 =^T \left[\frac{1}{3}, \frac{-1}{3}, \frac{1}{3}, 0\right]$
comme $\lambda_5 = 0$, $G := [x_1^4 - 2\,x_1^3 + 6\,x_1^2 - 23\,x_1 - 5, x_2 - \frac{1}{3}x_1^2 + \frac{1}{3}x_1 - \frac{1}{3}]$, puis en éliminant les multiples de $\mathrm{LT}(G_2) = x_2$, on obtient $L := []$ et l'algorithme FGLM se termine.

8.8 Preuve de l'algorithme

Théorème 39. *Les algorithmes 9 et 11 se terminent et calculent une base de Gröbner Le nombre d'opérations dans* $\mathbb{K}$ *de l'algorithme 11 est borné par* $O(n \deg(I)^3)$.

Proof. On fait la preuve de l'algorithme 11, la preuve de l'algorithme non matriciel étant similaire.

L'entrée de l'algorithme étant la forme normale φ, on note I le noyau de cette forme normale; par hypothèse I est un idéal et soit $G' = [g'_1, g'_2, \ldots]$ la base de Gröbner réduite de I pour l'ordre $<$ (on suppose que $\mathrm{LT}_<(g'_1) < \mathrm{LT}_<(g'_2) < \cdots$) et $\mathcal{E}_<(G') = \{w'_1 = 1 < w'_2 < \cdots < w'_D\}$ l'escalier de G' pour l'ordre $<$ avec $D = \deg(I)$.

On note $G_i = [g_1, g_2, \ldots, g_i]$ (respectivement S_i) la valeur de la variable G (resp. S) lorsqu'on ajoute g_i dans G (resp. lorsque $S := S \cup [x_k S_l]$) dans l'algorithme 11. Pour tout i, tous les éléments de G_i sont dans l'idéal I: en effet lorsqu'on rajoute le polynôme $p = x_k S_l - \sum_{i=1}^{\#S} \lambda_i \cdot S_i$ on a par construction $\varphi(p) = 0$ et donc $p \in I = \ker(\varphi)$. De plus, il est clair que les éléments de S_i sont linéairement indépendants modulo I : comme l'espace vectoriel $\mathbb{K}[x_1, \ldots, x_n]/I$ est de dimension finie ceci implique la terminaison de l'algorithme. Soit s le nombre d'éléments de $G = G_s$ lorsque l'algorithme se termine.

Supposons que $\mathcal{E}_<(G') \neq S_s$ on note $i := \min\{j \mid S_j \neq \{w'_1, \ldots, w'_i\}\}$ (ce nombre existe car $S_1 = \{1\} = \{w'_1\}$ donc $i > 1$). On note $S_i = \{w_1 = 1 < w_2 < \cdots < w_i\}$; par hypothèse $w_i \neq w'_i$. Il y a deux cas

1. $w_i > w'_i$: pour tout x_j divisant w'_i on sait que $\frac{w'_i}{x_j} \in \mathcal{E}_<(G')$ donc il existe $k_j < i$ tel que $\frac{w'_i}{x_j} = w'_{k_j} = w_{k_j}$. On a donc traité le monôme $t = w_{k_j}$ dans une étape précédente et on a ajouté dans L tous les multiples $x_l w_{k_j}$ donc en particulier w'_i. Comme $[\varphi(w'_1), \ldots, \varphi(w'_i)]$ sont linéairement indépendant et que $w'_i < w_i$ on a donc ajouté w_i dans S. Absurdité.

2. $w_i < w'_i$: on a donc $w_i \notin \mathcal{E}_<(G')$ et donc $\left[\varphi(w'_1), \ldots, \varphi\left(w'_{i-1}\right), \varphi(w_i)\right]$ ne sont pas linéairement indépendants. Donc, dans l'algorithme FGLM lorsqu'on traite le monôme w_i on trouve une combinaison linéaire $\varphi(w_i) = \lambda_1 \varphi(w'_1) + \cdots + \lambda_{i-1} \varphi\left(w'_{i-1}\right)$ et on ajoute le polynôme $g = w_i - \lambda_1 w'_1 - \cdots - \lambda_{i-1} w'_{i-1}$ dans G et w_i n'est jamais dans S.

Comme dans les deux cas on obtient une contradiction donc $\mathcal{E}_<(G') = S_s$. Maintenant pour tout polynôme $g \in G'$ (unitaire) et pour tout j tel que $x_j | \mathrm{LT}(g)$ on sait que $\frac{\mathrm{LT}(g)}{x_j} \in \mathcal{E}_<(G')$ donc il existe k_j tel que $\frac{\mathrm{LT}(g)}{x_j} = w'_{k_j}$ et donc

a ajouté $x_j\, w'_{k_j} = \mathrm{LT}(g)$ dans L. De plus pour tout $t \in T\,(g - \mathrm{LM}(g))$ on a $t \in \mathcal{E}_<(G')$ et donc $t = w_{j_t}$ pour un certain j_t. En notant $l = \max\{k_j\}$ on a donc $j_t < l$ puis $\{w'_1, \ldots, w'_l\}$ sont linéairement indépendant mais Comme $\{\mathrm{LT}(g)\} \cup \{w'_1, \ldots, w'_l\}$ sont linéairement dépendant: l'algorithme FGLM trouve la relation de dépendance linéaire $\varphi\,(\mathrm{LT}(g)) = \lambda_1 \varphi\,(w'_1) + \cdots + \lambda_l \varphi\,(w'_l)$ et ajoute à G le polynôme $\mathrm{LT}(g) - \lambda_1 w'_1 - \cdots - \lambda_{i-1} w'_{i-1} = g$. Par conséquent $G_s = G'$.

Il est clair que la complexité de l'algorithme 10 est bornée par $O(n\ \deg(I)^2)$; de plus dans l'algorithme principal FGLM on augmente L seulement lorsqu'on détecte un vecteur linéairement indépendant; par conséquent la taille de L et le nombre d'itérations de l'algorithme est borné par $n\,D$. Les autres opérations arithmétiques sont des produits $M \cdot v$ dont la complexité est bornée par $O(D^2)$. Par conséquent la complexité globale est $O\left(n\,D^3\right)$. □

8.9 Complexité de FGLM.

Dans cette section on résume les résultats de complexité obtenus dans cette section. Les résultats sont complètement différents si les opérations arithmétiques se font en temps constant (corps fini de petite taille) ou si on prend en compte la croissance des coefficients. Dans ce dernier il est toujours préférable de calculer une forme RUR ou RR qui permet de limiter la croissance des coefficients dans les calculs et dans le résultat (voir l'annexe **??** et (Rouillier, 1999)). On se réfère à l'article (Faugère, J.C., Gianni, P., Lazard, D. and Mora T., 1993) pour une estimation de la complexité booléenne de l'algorithme FGLM.

Corollaire 6. *Soit I un idéal zéro dimensionnel et $(G_1, <_1)$, une base de Gröbner réduite par rapport à un ordre admissible $<_1$. Étant donné un ordre admissible $<_2$, il est possible de calculer la base de Gröbner du même idéal I par rapport à l'ordre $<_2$ en $O(n\ \deg(I)^3)$ opérations arithmétiques.*

Proof. En utilisant l'algorithme 8 on calcule en $O(n\ \deg(I)^3)$ opérations les matrices de multiplications. On peut ensuite appliquer l'algorithme 11 pour effectuer le changement d'ordre. □

En combinant ce résultat avec des résultats de complexité pour le calcul d'une base de Gröbner pour un ordre DRL on obtient le résultat général:

Théorème 40. *Soit I un idéal de dimension zéro engendré par des polynômes en n variables de degré au plus d. Si les générateurs de I ont un nombre fini de solutions à l'infini alors on peut calculer la base de Gröbner de I pour n'importe quel ordre en temps polynomial en d^n. Si l'ensemble des zéros à l'infini est infini, alors la complexité est polynomiale en d^{n^2}.*

Proof. Si le nombre de solutions à l'infini est fini alors (Lazard D., 1983) montre que le calcul de la base de Gröbner par l'algorithme de Buchberger se fait en temps polynomial en d^n; le même résultat est obtenu avec les algorithmes F_4 ou F_5 (voir la section 14). On applique ensuite l'algorithme FGLM pour calculer la base pour n'importe quel ordre. Le résultat résulte de la borne $O(n\ \deg(I)^3)$ (du corollaire 6) et de la borne de Bezout $\deg(I) \leqslant d^n$. Si on prend en compte la croissance des coefficients voir la preuve dans (Faugère, J.C., Gianni, P., Lazard, D. and Mora T., 1993)). □

9 Algorithme F_4

9.1 Introduction

Si on se contente d'appliquer les algorithmes (Buchberger B., 1965; Buchberger B., 1970; Buchberger B., 1985) déjà décrit dans la section 2 sur une liste de problèmes typiques provenant d'applications diverses on constate que même les meilleures implantations sont incapables de calculer les bases de Gröbner pour des calculs de taille moyenne. Dans cette section, on va décrire un autre algorithme (dont le nom est F_4) pour calculer efficacement des bases de Gröbner et principalement des bases pour un ordre du degré (DRL). Il sera donc nécessaire en pratique d'appliquer un autre algorithme d'élimination ou de changement d'ordre sur le résultat fournit par F_4. On verra cependant des exemples où le calcul pour un ordre d'élimination est aussi très efficace. En simplifiant, on pourrait suggérer deux amé liorations de l'algorithme de Buchberger: comme 90% du temps est passé à réduire des paires critiques vers zéro il serait utile de posséder des critères plus fort que les critères de Buchberger (Buchberger B., 1979) pour éliminer *toutes* les paires critiques inutiles (de façon théorique il existe un tel critère mais son application en pratique est très coûteuse). Cet aspect fondamental est l'objet de la section suivante (chap.11). La seconde amélioration concerne les stratégies: durant un calcul de base de Gröbner on est confronté à plusieurs choix: choisir une paire critique dans une liste de paires critiques, choisir un réducteur dans une liste. On sait que quelque soit les choix le résultat final de l'algorithme ne change pas (théorème de Buchberger), en revanche les calculs peuvent devenir impossible en cas de mauvais choix. Même si diverses stratégies ont été proposées ((Giovini A. and Mora T. and Niesi G. and Robbiano L. and Traverso C., 1991) ou même (Gerdt V.P., 1995)), les heuristiques sur lesquelles elles reposent ne sont pas entièrement expliquées et sont plus ou moins efficaces sur divers exemples; il est difficile d'en désigner une qui soit optimale dans tous les cas. L'objectif de cette section est de présenter un algorithme plus puissant de réduction. Dans ce but on va chercher à réduire *simultanément* plusieurs polynômes par une liste de polynômes en utilisant des techniques d'algèbre linéaire (la réduction de Gauß principalement).

L'algorithme F_4 de base est présenté dans la section suivante. Cette section est divisée en plusieurs parties: premièrement nous établissons le lien entre algèbre linéaire (matrices) et l'algèbre des polynômes et la méthode de Macaulay(Macaulay, 1916) et lien entre et le calcul des bases de Gröbner est explicité. Ensuite nous présentons dans la sous-section 10.1 une version simplifiée de l'algorithme qui n'utilise que partiellement le résultat de la réduction de Gauß. Une version améliorée de l'algorithme incluant les critères de Buchberger est donnée dans 10.2. Nous terminons cette section dans 10.4 en donnant des indications sur le choix d'une bonne stratégie de sélection des paquets de paires critique. Cet algorithme a été implanté dans un logiciel écrit par l'auteur nommé FGb (Fast Gb) accessible via le logiciel généraliste Maple et plus récemment dans d'autres logiciels dont en particulier Maple. Le nom de cet algorithme est simplement l'algorithme numéro 4. Dans le reste du document, F_4 désigne cet algorithme.

9.2 Bases de Gröbner et Macaulay

9.3 Représentation matricielle des polynômes.

Définition 23. *Si $F = [f_1, \ldots, f_m]$ est un vecteur de m polynômes et $<$ un ordre admissible, $\mathrm{T}_<(F) = [t_1, \ldots, t_l]$ les termes du support de F triés pour l'ordre $<$ (voir la définition dans la section 4.2 p. 6). Alors une représentation matricielle $M_{T_<(F)}(F)$ de F est une matrice:*

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

dont le coefficient d'indice (i, j) est le coefficient du terme t_j dans f_i . De plus, $M_{T_<(F)}(F)$ vérifie l'équation:

$$F = M_{T_<(F)}(F) \cdot T_<(F)$$

Pour alléger les notations on note $M(F) = M_{T_<(F)}(F)$.

Définition 24. *Réciproquement si M est une matrice de taille $l \times m$ à coefficients dans $\mathbb{K}$ et $X = [t_1, \ldots, t_m]$ un vecteur de termes alors la représentation polynomiale de M par rapport à X est le vecteur de l polynômes déterminé par l'équation:*

$$F = M \cdot X$$

À partir de polynômes on peut toujours construire une matrice, généralisation naturelle de la matrice de Sylvester, qui consiste à multiplier tous les polynômes par tous les termes possibles:

Définition 25. *Matrice de Macaulay ((Macaulay, 1916)). Soient $F = [f_1, \ldots, f_m]$ un vecteur de m polynômes et d un entier positif alors la matrice de Macaulay en degré d de F, notée $\mathcal{M}_d^{\text{acaulay}}(F)$, est la représentation matricielle de*

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

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

L'opération de base pour les matrices est le calcul d'une forme échelon; ce sera l'opération principale et bien souvent la plus coûteuse.

Définition 26. *Si $M(F)$ est la représentation matricielle d'un vecteur de polynômes F on note $\widetilde{M(F)}$ le résultat d'une élimination de Gauß de la matrice $M(F)$ (sans faire de pivot sur les colonnes).*

Pour simplifier l'exposition de l'algorithme on définit également l'élimination de Gauß d'un vecteur de polynômes:

Définition 27. *Soit F un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$ et $<$ un ordre admissible. À partir de F on construit sa représentation matricielle, $M(F)$, puis on calcule sa forme échelon, $\widetilde{M(F)}$, enfin on considère la représentation polynomiale, notée $\tilde{F}$ de cette dernière matrice; on dira que $\tilde{F}$ est la forme échelonnée de F (ou élimination de Gauß) par rapport à $<$.*

9.4 Méthode de Macaulay

L'utilisation de l'algèbre linéaire pour résoudre un système algébrique remonte à Macaulay (voir (Macaulay, 1916)); Macaulay généralise la matrice de Sylvester (J., 1853) (c'est la matrice utilisé pour le résultant de deux polynômes univariés) aux polynômes à plusieurs variables.

Le lien entre le calcul d'une base de d-Gröbner (voir définition 15 p.15) et l'algèbre linéaire découle du fait qu'on peut calculer une base de Gröbner à partir de la matrice de Macaulay suivant le théorème de D. Lazard:

Théorème 41. *(Lazard (Lazard D., 1983; Lazard D., 1981)) Si $F = \{f_1, \ldots, f_m\}$ est un ensemble de polynômes homogènes alors la représentation polynomiale de $\widetilde{\mathcal{M}_d^{\text{acaulay}}}(F)$ est une d-base de Gröbner (non réduite) de F.*

Par conséquent si d est *assez grand* alors la matrice de Macaulay correspondante contient la base de Gröbner complète; en particulier:

Théorème 42. *(borne de Macaulay). Soit $F = \{f_1, \ldots, f_m\}$ un ensemble de polynômes homogènes réguliers. On définit:*

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

alors $\widetilde{\mathcal{M}_D^{\text{acaulay}}}(F)$ est une base de Gröbner (non réduite) de F.

Remarque 11. *D'un point de vue pratique l'utilisation de la matrice de Macaulay est désastreuse; en effet les lignes de cette matrice ne sont pas indépendantes et donc appliquer une élimination de Gauß est inutilement coûteux. Le but de cette section et de la section suivant (sect. 11) est de construire des matrices définissant le même espace vectoriel mais dont la taille sera beaucoup plus petite. De plus dans la méthode de Macaulay il faut une borne D sur le degré maximal des calculs ou une hypothèse sur le système initial. La présente section permet de réduire considérablement les tailles de ces matrices; la section suivante (sect. 11) construit des matrices dont la taille est optimale dans le sens où les matrices seront de rang pleins. Toutefois il est important de noter que des matrices de plus petites tailles n'impliquent pas* automatiquement *des calculs plus rapides.*

Remarque 12. *En cryptographie, l'algorithme XL (Courtois* et al. *, 2000) a été créé pour chercher* une *solution d'un système algébrique dans un corps fini. En résumé, le principe est de construire la matrice de Macaulay sans un certain degré puis de trianguler la matrice pour faire apparaître un polynôme univarié dont il est facile d'extraire une solution dans le corps fini. Cet algorithme peut toujours être simulé par un calcul de base de Gröbner (voir (Ars* et al. *, 2004; G., 2005)).*

10 L'algorithme F_4

10.1 Description de l'algorithme F_4

On sait que durant l'exécution de l'algorithme de Buchberger, on dispose de plusieurs degré de liberté:

- choisir une paire critique dans une liste de paires critique.
- choisir un réducteur dans une liste de réducteurs lorsqu'on réduit un polynôme par une liste de polynômes.

Bruno Buchberger (Buchberger B., 1965) a prouvé que ces choix ne sont pas importants en regard de la preuve de l'algorithme, en revanche ces choix influent de manière *cruciale* sur le temps de calcul. De plus les meilleures stratégies ((Giovini A. and Mora T. and Niesi G. and Robbiano L. and Traverso C., 1991)) se basent uniquement sur le terme de têtes des polynômes pour faire un choix. Si on prend le cas extrême où tous les polynômes ont le même terme de tête, toutes les paires critiques sont égales et il n'est pas possible de faire un choix. Nous résolvons ce problème d'une manière simple et un peu surprenante: *on ne fait pas de choix.* Plus exactement au lieu de choisir *une paire critique* à chaque étape, on considère un *sous-ensemble* de paires critiques que l'on va traiter *simultanément*. À partir de ce choix on construit une matrice la plus creuse possible mais contenant toutes les réductions *a priori*. Ensuite il faut appliquer une réduction de Gauß sur cette matrice. Par conséquent, on reporte les choix nécessaires (choix de pivot par exemple) dans la seconde phase de l'algorithme qui est la phase d'algèbre linéaire de l'algorithme.

Nous modifions légèrement la définition usuelle de paire critique:

Définition 28. *Une paire critique de deux polynômes (f_i, f_j) est un élément de $T^2 \times \mathbb{K}[x_1, \ldots, x_n] \times T \times \mathbb{K}[x_1, \ldots, x_n]$,*

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

tel que

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

On définit les deux opérateurs de projections:

Définition 29. *On dira que le degré d'une paire critique $p_{i,j} = \mathrm{Pair}(f_i, f_j)$, $\deg(p_{i,j})$, est $\deg(lcm_{i,j})$. De plus on définit les opérateurs:*

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

Nous avons maintenant les outils pour présenter une version simplifiée de l'algorithme. Toutes les matrices apparaissant dans les algorithmes suivants sont la représentation matricielle d'une liste de polynômes tel que décrit dans la définition 23.

Algorithme 12. *Algorithme F_4 (version simplifiée)*

Input: $\begin{cases} F \text{ un sous-ensemble fini de } \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{S}el \text{ une fonction } List(Pairs) \to List(Pairs) \\ \quad \text{tel que } \mathcal{S}el(l) \neq \varnothing \text{ si } l \neq \varnothing \end{cases}$
***Output**: un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$.*
$G := F$, $\tilde{F_0^+} := F$, $d := 0$ *et* $P := \left\{\mathrm{Pair}(f, g) \mid (f, g) \in G^2 \text{ avec } f \neq g\right\}$
while $P \neq \varnothing$ ***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

Nous devons maintenant étendre la définition de la réduction d'un polynôme modulo un sous-ensemble de $\mathbb{K}[x_1, \ldots, x_n]$, à la réduction d'un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$ modulo un autre sous-ensemble de $\mathbb{K}[x_1, \ldots, x_n]$:

Algorithme 13. REDUCTION
Input: *L, G sous-ensembles finis de $\mathbb{K}[x_1, \ldots, x_n]$*
***Output**: un sous-ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$ (éventuellement vide).*
F :=SYMBOLICPREPROCESSING(L, G)
$\widetilde{F}$:= *Réduction de Gauß de F par rapport à $<$*
$\tilde{F^+} := \left\{f \in \tilde{F} \mid \mathrm{LT}(f) \notin \mathrm{LT}(F)\right\}$ // partie "utile" de $\widetilde{F}$
return F^+

Nous décrivons maintenant la fonction principale de l'algorithme, c'est à dire la construction de F et donc de la matrice $M(F)$. Elle ajoute, en fonction des données L et G, tous les polynômes nécessaires pour que la réduction de Gauß contienne les réductions modulo G de l'ensemble L. Dans la mesure ou aucune opération arithmétique n'est utilisée, c'est vraiment un pré-traitement *symbolique*.

Algorithme 14. SYMBOLICPREPROCESSING
Input: *L, G sous-ensembles finis de $\mathbb{K}[x_1, \ldots, x_n]$*
***Output**: un sous-ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$*
$F := L$
$Done := \mathrm{LT}(F)$
while $\mathrm{T}(F) \neq Done$ ***do***
  choisir m un élément de $\mathrm{T}(F) \backslash Done$
  $Done := Done \cup \{m\}$
  if *m top réductible modulo G* ***then***
    il existe $g \in G$ et un $m' \in T$tel que $m = m' \cdot \mathrm{LT}(g)$
    $F := F \cup \{m' \cdot g\}$
return F

Remarque 13. *La procédure* SYMBOLICPREPROCESSING *est très efficace puisque sa complexité est proportionnelle en la taille de sa sortie si $\#G$ est plus petite que le taille finale de $T(F)$ ce qui est souvent le cas en pratique. Comme l'ordre monomial n'intervient pas on peut aussi envisager une implantation en parallèle.*

Nous renvoyons à (**?**) pour des preuves complètes de terminaison et de correction de l'algorithme et nous contentons d'en esquisser les grandes lignes:

Lemme 2. *Pour tout polynôme $p \in L$,on a $p \xrightarrow{G \cup \tilde{F^+}} 0$*

Théorème 43. *L'algorithme F_4 calcule une base de Gröbner G dans $\mathbb{K}[x_1, \ldots, x_n]$ telle que $F \subseteq G$ et $Id(G) = Id(F)$.*

Proof. Terminaison: Par l'absurde. Supposons que la boucle **while** ne se termine pas. On en déduit qu'il existe une suite croissante (d_i) d'entiers naturels tels que $\tilde{F^+_{d_i}} \neq \varnothing$ pour tout i. Fixons $q_i \in \tilde{F^+_{d_i}}$ (et donc q_i peut être n'importe quel élément dans $\tilde{F^+_{d_i}}$). On définit U_i comme étant l'idéal $U_{i-1} + \mathrm{Id}(\mathrm{LT}(q_i))$ pour $i > 1$ et $U_0 = \mathrm{Id}(0)$. La ligne

$$\tilde{F^+} := \left\{ f \in \tilde{F} \mid \mathrm{LT}(f) \notin \mathrm{LT}(F) \right\}$$

implique que $U_{i-1} \subsetneq U_i$. Ceci contredit le fait que $\mathbb{K}[x_1, \ldots, x_n]$ soit noethérien.

validité: On a $G = \cup_{d \geqslant 0} \tilde{F^+_d}$. On montre que les quantités suivantes sont des invariants de la boucle **while**: G est un sous-ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$ tel que $F \subset G \subset \mathrm{Id}(F)$, et $spol(g_1, g_2) \xrightarrow{*} 0$ pour tout $(g_1, g_2) \in G^2$ tel que $(g_1, g_2) \notin P$. Pour cette dernière assertion, si $(g_1, g_2) \notin P$, ceci implique que $Pair(g_1, g_2) = (lcm_{1,2}, t_1, g_1, t_2, g_2)$ a été sélectionné lors d'une précédente étape (disons à l'étape d) par la fonction de sélection $\mathcal{S}el$. Par conséquent $t_1 \cdot g_1$ et $t_2 \cdot g_2$ sont dans L_d, donc $spol(g_1, g_2) = t_1 \cdot g_1 - t_2 \cdot g_2 \xrightarrow{*} 0$ d'après le lemme 2. □

Remarque 14. *Si $\#\mathcal{S}el(l) = 1$ pour tout $l \neq \varnothing$ alors l'algorithme F_4 est exactement l'algorithme de Buchberger. Dans ce cas la fonction $\mathcal{S}el$ correspond à la stratégie de sélection des paires critiques de l'algorithme de Buchberger.*

Remarque 15. *Dans la preuve de terminaison on peut se demander pourquoi on ne considère qu'un seul élément de $\tilde{F^+_d}$ et pas $\tilde{F^+_d}$ en entier. En considérons l'exemple suivant:*

pour l'ordre lexicographique tel que $x > y > z$, soient $F = [f_1 = xy^2 + 1, f_2 = xz^2 + 1, f_3 = y^3 + y^2]$ et $\mathcal{S}el =$ la fonction identité. On trouve successivement $P_1 = \{Pair(f_1, f_2), Pair(f_2, f_3), Pair(f_1, f_3)\}$ puis $\tilde{F^+_1} = \{y^2 - z^2, y + 1\}$ et donc que $Id(LT(\tilde{F^+_1})) = \{y\}$. Par suite, et contrairement à l'algorithme de Buchberger, il n'est pas vrai qu'après chaque opération

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

on ait $Id(LT(G')) \supsetneq Id(LT(G))$.

10.2 Ajout des critères de Buchberger dans F_4

Afin d'obtenir une version vraiment efficace de l'algorithme il est indispensable de lui adjoindre des critères pour éviter des calculs inutiles. Dans cette section on intègre a F_4 les critères de Buchberger; une autre possibilité est d'utiliser les critères F_5 de la section11. Dans la description suivante on utilise l'implantation standard de ces critères ((Gebauer & Möller, 1988)):

Algorithme 15. *Critères de Buchberger*

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

Input: $\begin{cases} \text{un sous-ensemble fini } G_{old} \text{ de } \mathbb{K}[x_1, \ldots, x_n] \\ \text{un ensemble fini } P_{old} \text{ de paires critiques de } \mathbb{K}[x_1, \ldots, x_n] \\ 0 \neq h \in \mathbb{K}[x_1, \ldots, x_n] \end{cases}$

Output: *un sous-ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$ et une liste de paires critiques correspondant à la mise à jour de la liste des paires critiques.*

10.3 Version optimisée de l'algorithme F_4

Dans la version initiale de l'algorithme nous avons utilisé seulement *certaines* lignes de la matrice mise sous forme échelonnée (les lignes correspondant à F^+_d). Dans la nouvelle version de l'algorithme on va garder ces lignes inutilisées en essayant de remplacer certains produits $m \cdot f$ apparaissant dans les lignes de la matrice F par un produit

équivalent $m' \cdot f'$ avec $m \geqslant m'$ (en pratique on ne conserve que des produits de la forme $x_k \cdot f'$. Une nouvelle variable $\mathcal{F}$ est introduite dans l'algorithme: $\mathcal{F}$ est un vecteur contenant toutes les formes échelons qui ont été calculées:

Algorithme 16. *Algorithme F_4 (version améliorée)*

Input: $\begin{cases} F \text{ un sous-ensemble fini de } \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{S}el \text{ une fonction } \mathrm{List}(Pairs) \to \mathrm{List}(Pairs) \\ \quad \text{tel que } \mathcal{S}el(l) \neq \emptyset \text{ si } l \neq \emptyset \end{cases}$
***Output**: un sous ensemble fini de $\mathbb{K}[x_1, \ldots, x_n]$.*
$G := \emptyset$ *et* $P := \emptyset$ *et* $d := 0$
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

La nouvelle fonction de réduction est identique à la précédente à l'exception d'un nouvel argument en entrée; en sortie on retourne d'une part les nouveaux éléments de la base de Gröbner mais aussi la matrice complète mise sous forme échelon:

Algorithme 17. REDUCTION

Input: $\begin{cases} L, G \text{ sous-ensemble finis de } \mathbb{K}[x_1, \ldots, x_n] \\ \mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}, \text{ où } F_k \\ \quad \text{est un sous ensemble fini de } \mathbb{K}[x_1, \ldots, x_n] \end{cases}$
***Output**: deux sous-ensembles fini de $\mathbb{K}[x_1, \ldots, x_n]$.*
$F :=$SYMBOLICPREPROCESSING$(L, G, \mathcal{F})$
$\tilde{F} :=$ *Réduction de Gauß de F par rapport à* $<$
$\tilde{F^+} := \left\{ f \in \tilde{F} \mid \mathrm{LT}(f) \notin \mathrm{LT}(F) \right\}$
return $(\mathcal{F}, \tilde{F^+})$

La variante principale avec la version simplifiée de l'algorithme se trouve dans la fonction SYMBOLICPREPROCESSING: c'est ici qu'on opère la substitution d'un produit $m \cdot f$ par un "meilleur" produit $m' \cdot f'$:

Algorithme 18. SYMBOLICPREPROCESSING

Input: L, G *sous-ensemble finis de* $\mathbb{K}[x_1, \ldots, x_n]$; $\mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}$, *où* F_k *est un sous ensemble fini de* $\mathbb{K}[x_1, \ldots, x_n]$

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

$F := L$
$Done := \mathrm{LT}(F)$
while $T(F) \neq Done$ ***do***
  choisir m *un élément de* $T(F) \backslash Done$
  $Done := Done \cup \{m\}$
  if m *top réductible modulo* G ***then***
    il existe $g \in G$ *et un* $m' \in T$ *tel que* $m = m' \cdot \mathrm{LT}(g)$
    $F := F \cup \{\text{SIMPLIFY}(m', g, \mathcal{F})\}$
return F

La fonction SIMPLIFY cherche à remplacer le produit $m \cdot f$ par un produit $(u\,t) \cdot f'$ où (t, f') est une étiquette de ligne dont on a déjà calculé la forme échelon et $u\,t$ divise le terme m'; si on trouve effectivement un meilleur produit on ré-appelle récursivement la fonction SIMPLIFY (voir l'exemple dans la section 10.5 pour un exemple d'appel récursif):

Algorithme 19. SIMPLIFY

Input: $t \in T$ *un terme*; $f \in \mathbb{K}[x_1, \ldots, x_n]$ *un polynôme*; $\mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}$, *où* F_k *est un sous ensemble fini de* $\mathbb{K}[x_1, \ldots, x_n]$

Output: *un élément de la forme* $m' \cdot f'$
for $u \in$ *liste des diviseurs de* t ***do***
 if $\exists j$ $(1 \leqslant j < d)$ *tel que* $(u \cdot f) \in F_j$ ***then***
    $\tilde{F}_j$ *est la forme échelonnée de* F_j *par rapport à* $<$
    il existe un et un seul $p \in \tilde{F}_j^+$ *tel que* $\mathrm{LT}(p) = \mathrm{LT}(u \cdot f)$
    if $u \neq t$ ***then***
      return $\text{SIMPLIFY}(\frac{t}{u}, p, \mathcal{F})$
    else
      return $1 \cdot p$
return $t \cdot f$

Remarque 16. *L'expérience montre que l'effet de Simplify est de retourner, dans* 95% *des cas, un produit* $x_i \cdot p$ *où* x_i *est une variable (et même le plus souvent le produit* $x_n \cdot p$ *par la dernière variable). Cette technique est un peu similaire à l'algorithme FGLM (voir la section 8 page 21) où on utilise des matrices de multiplication pour calculer des formes normales.*

Encore une fois on se réfère à l'article original ((**?**)) pour la preuve complète de l'algorithme modifié; le théorème suivant permet de se ramener au théorème (20 p. 13) caractérisant les bases de Gröbner:

Théorème 44. *Soit* F *un sous-ensemble fini de* $\mathbb{K}[x_1, \ldots, x_n]$, $\mathcal{F} = (F_k)_{k=1,\ldots,(d-1)}$, *où* F_k *est un sous-ensemble fini de* $\mathbb{K}[x_1, \ldots, x_n]$, $Pair(g_1, g_2) = (lcm_{1,2}, t_1, g_1, t_2, g_2)$ *avec* $lcm_{1,2}, t_1, t_2 \in T$ *tel que les conditions suivantes soient vérifiées:*

(i) $(\tilde{F}_k)_+ \subset G$ *pour* $k = 1, \ldots, (d-1)$

(ii) $f_i =$SIMPLIFY$(t_i, g_i, \mathcal{F})$ *pour* $i = 1, 2$.

(iii) $\mathrm{spol}(f_1, f_2) = o_F(\mathrm{lcm}(\mathrm{LT}(f_1), \mathrm{LT}(f_2)))$ *(voir section 10 page 12).*

Alors $\mathrm{spol}(g_1, g_2) = o_F(lcm_{1,2})$.

Proof. Soit (t'_i, g'_i) le résultat de SIMPLIFY$(t_i, g_i, \mathcal{F})$. D'après le lemme **??** on a $\mathrm{LT}(t'_1 g'_1) = \mathrm{LT}(t_1 g_1) = lcm_{1,2} = \mathrm{LT}(t_2 g_2) = \mathrm{LT}(t'_2 g'_2)$ de telle sorte que (en supposant tous les polynômes unitaires):

$$\begin{aligned} \mathrm{spol}(g'_1, g'_2) &= t'_1 g'_1 - t'_2 g'_2 \\ &= (t'_1 g'_1 - t_1 g_1) + (t_1 g_1 - t_2 g_2) + (t_2 g_2 - t'_2 g'_2) \\ &= r + \mathrm{spol}(f_1, f_2) + r' \end{aligned}$$

avec $r, r' \in \mathrm{Id}(\tilde{\mathcal{F}^+} \cup \mathcal{F}) \subset Id(G)$ tel que $\max(\mathrm{LT}(r), \mathrm{LT}(r')) < lcm_{1,2}$. Ainsi $\mathrm{spol}(g_1, g_2) = \mathcal{O}_F(t')$ pour $t' = \max(\mathrm{LT}(r), \mathrm{LT}(r'), t) < lcm_{1,2}$. Le résultat découle du théorème 20. □

10.4 Fonction de sélection

Le choix d'une bonne fonction $\mathcal{S}el$, c'est à dire une stratégie de calcul, est très important pour les performances de l'algorithme.

Calculer une base de Gröbner pour un ordre du degré est souvent une partie très difficile dans la chaîne des calculs aboutissant à la résolution complète d'un système algébrique. Une explication est que l'entrée de l'algorithme est simplement un sous-ensemble de $\mathbb{K}[x_1, \ldots, x_n]$ *sans* structure mathématique. On peut se donner une certaine structure dès le début de l'algorithme en utilisant le concept de d-bases de Gröbner (voir 15 p. 15). C'est à dire que la fonction de sélection choisit *toutes* les paires critiques de degré minimale:

Algorithme 20. $\mathcal{S}el$

Input: *P une liste de paires critiques*
Output*: une liste de paires critiques.*
$d := \min\{\deg(\mathrm{lcm}(p)) \mid p \in P\}$
$\tilde{F} :=$ *Réduction de Gauß de F par rapport à $<$*
$P_d := \{p \in P \mid \deg(\mathrm{lcm}(p)) = d\}$
return P_d

On appelle cette stratégies la *stratégie normale pour l'algorithme* F_4. Ainsi si les polynômes de départ son homogènes, on obtient, en degré d, une d base de Gröbner; $\mathcal{S}el$ sélectionne donc à l'étape suivante toutes les paires qui sont nécessaires pour calculer une base de Gröbner jusqu'en degré $d+1$.

Une variante serait de changer $\deg(\mathrm{lcm}(p))$ par $\deg_{Sugar}(\mathrm{lcm}(p))$ où $\deg_{Sugar}$ représente le "sucre" (voir l'article (Giovini A. and Mora T. and Niesi G. and Robbiano L. and Traverso C., 1991)).

10.5 Exemple détaillé étape par étape

L'exemple suivant est donné à titre d'illustration; il est malheureusement impossible de montrer le gain en efficacité sur un exemple de petite taille. On considère le problème cyclic 4. On prend comme ordre admissible l'ordre degree reverse lexicographical ordering (DRL) et la stratégie normale (voir aussi section 10.4)

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

Au départ $G = \{f_4\}$ et $P_1 = \{\mathrm{Pair}(f_3, f_4)\}$ de telle sorte que $L_1 = \{(1, f_3), (b, f_4)\}$.

On rentre dans la fonction SYMBOLICPREPROCESSING$(L_1, G, \varnothing)$; $F_1 = L_1$, $Done = \mathrm{LT}(F_1) = \{ab\}$ et $T(F_1) = \{ad, ab, b^2, bc, bd, cd\}$, on choisit un élément dans $T(F_1) \backslash Done$, par exemple ad, mais ad est top réductible par G; ainsi $Done = \{ab, ad\}$, $F_1 = F_1 \cup \{df_4\}$ et $T(F_1) = T(F_1) \cup \{d^2\}$.

Comme les autres éléments de $T(F_1)$ ne sont pas top réductible par G, SYMBOLICPREPROCESSING retourne

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

La représentation matricielle de F est:

$$A_1 = M(F_1) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

et la réduction de Gauß de A_1 est:

$$\tilde{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

par conséquent $\tilde{F}_1 = [f_5 = ad + bd + cd + d^2, f_6 = ab + bc - bd - d^2, f_7 = b^2 + 2\,bd + d^2]$ et comme $ab, ad \in \mathrm{LT}(F_1)$ on a $\tilde{F}_{1+} = [f_7]$ et maintenant $G = \{f_4, f_7\}$.

Lors de la prochaine étape on doit considérer $P_2 = \{\mathrm{Pair}(f_2, f_4)\}$, ainsi $L_2 = \{(1, f_2), (bc, f_4)\}$ et $\mathcal{F} = \{F_1\}$.

Dans SYMBOLICPREPROCESSING on cherche à simplifier les produits $1 \cdot f_2$ et $bc \cdot f_4$ avec $\mathcal{F}$. On voit que $bf_4 \in F_1$ et que f_6 est l'unique polynôme dans $\tilde{F}_1$ tel que $\mathrm{LT}(f_6) = \mathrm{LT}(bf_4) = ab$, ainsi $\textsc{Simplify}(bc, f_4, \mathcal{F}) = c \cdot f_6$. Maintenant $F_2 = [f_2, cf_6]$ et $T(F_2) = \{abc, bc^2, abd, acd, bcd, cd^2\}$. On choisit abd qui est réductible par bdf_4 mais encore une fois on peut remplacer ce produit par $b \cdot f_5$. Après quelques itérations de l'algorithme on trouve que

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

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

$$\tilde{A}_2 = \widetilde{M(F_2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \end{bmatrix}$$

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

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

Lors de la prochaine étape on a

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

et on appelle récursivement la fonction Simplify: $\textsc{Simplify}(bcd, f_4) = \textsc{Simplify}(cd, f_6) = \textsc{Simplify}(d, f_{12}) = (d, f_{12})$. On obtient

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

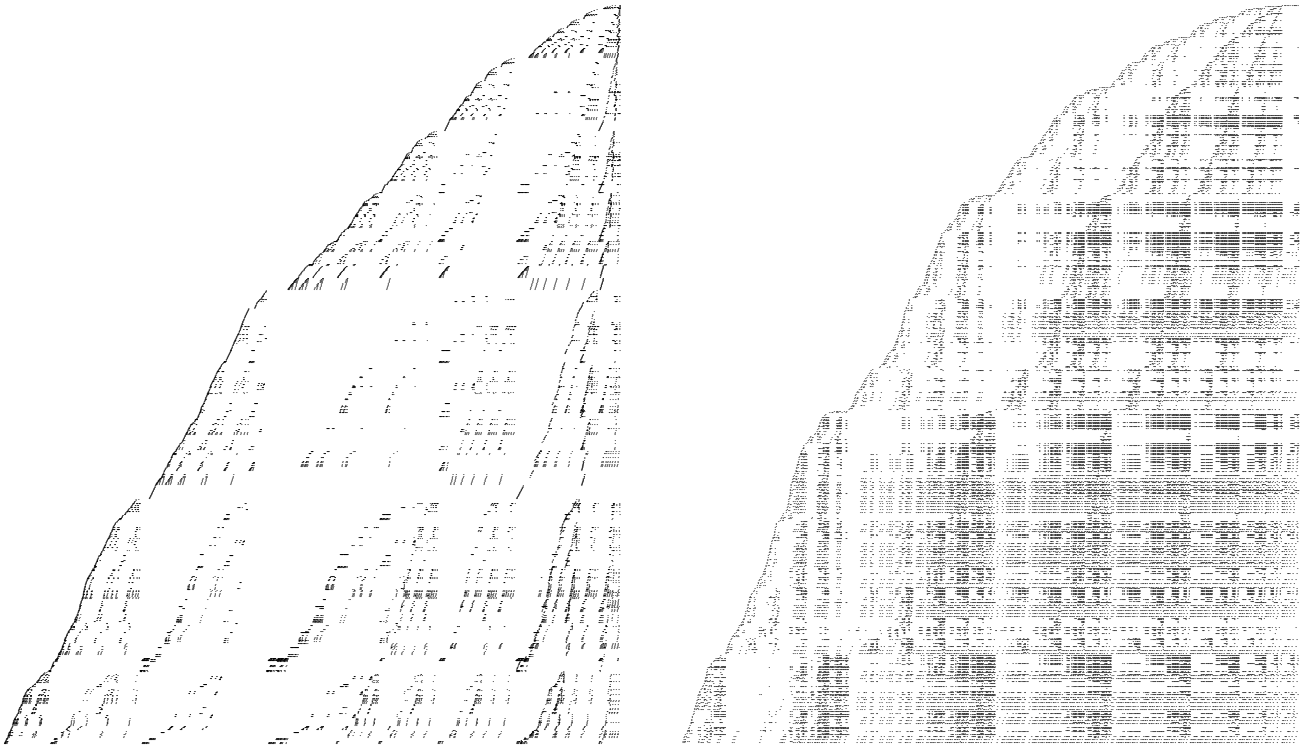
On remarque que $c^2 f_7$ ne peut être simplifié. Après quelques étapes dans SYMBOLICPREPROCESSING on trouve

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

puis $\tilde{F_3} = [f_{15} = c^2b^2 - c^2d^2 + 2\,bd^3 + 2\,d^4,\ f_{16} = abcd - 1,\ f_{17} = -bcd^2 - c^2d^2 + bd^3 - cd^3 + d^4 + 1,\ f_{18} = c^2bd + c^2d^2 - bd^3 - d^4,\ f_{19} = b^2d^2 + 2\,bd^3 + d^4]$.

Un rapide calcul montre que le rang de F_3 est seulement 5. Cela signifie qu'il y a une réduction à zéro qui n'a pas été évitée.

10.6 Optimisation de l'algèbre linéaire



L'étape de mise sous forme échelon des matrices est l'étape la plus coûteuse en temps d'exécution et en espace mémoire.

10.6.1 Algèbre linéaire dédiée

Afin de tenir compte de la structure quasi-triangulaire des matrices générées par l'algorithme F_4 nous avons proposé, avec S. Lacharte (Faugère, Jean-Charles and Lachartre, Sylvain, 2010) et C. Eder, des algorithmes dédiés. Une librairie GBLA[1], sous license LGPL, est également disponible et permet de réduire ces matrices sur des machines avec plusieurs processeurs.

[1] http://www-polsys.lip6.fr/~jcf/Software/GBLA/index.html

10.6.2 Compression des matrices

Lorsque les matrices sont grosses il est nécessaire d'adopter un schéma de stockage des données plus compliqué afin de réduire l'occupation mémoire: si on considère une matrice $5.10^4 \times 5.10^4$ avec 10% d'éléments non nuls (c'est exactement le cas pour le benchmark Cyclic 9 par exemple); même si on alloue un seul octet pour chaque coefficient (ce qui est plutôt optimiste si on considère que les coefficients ont souvent plusieurs centaines de chiffres) il faut $250 * 10^6$ octets pour stocker la matrice. Dans notre implantation on évite de dupliquer les coefficients (en effet la plupart des lignes sont des multiplication d'un même polynôme f par plusieurs monômes); ainsi on a juste a considérer les positions des éléments non nuls des éléments de la matrice: c'est une succession de 1 et 0 qu'il faut compresser (c'est donc une bitmap). Nous avons testé plusieurs méthodes:

(i) Pas de compression: inefficace à la fois d'un point de vue temps de calcul et d'un point de vue mémoire.

(ii) Compression bitmap: si on note par

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

les positions des éléments non nuls dans une ligne de la matrice, alors $\sum_k 2^{j_k - 1}$ est la forme bitmap. Cette méthode est efficace mais ne compresse par beaucoup la matrice (et seulement d'un facteur constant).

(iii) Une autre technique consiste a considérer les différences entre les positions non nulles:

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

quand les la différence $j_k - j_{k-1}$ est petite (< 128), et ceci arrive souvent, on peut la stocker sur un octet. Cette méthode est plus efficace en terme d'occupation mémoire que la précédente et seulement un peu plus lente (10% plus lente).

(iv) On peut aussi appliquer une technique à la gzip sur la représentation précédente: par on peut identifier des *patterns* de r blocs de 1 consécutifs (c'est à dire $j_i = j_{i+1} - 1 = j_{i+2} - 2 = \cdots$. Cette méthode est plus complexe à mettre en oeuvre mais elle est beaucoup plus efficace.

11 Critère F_5 et algorithme F_5.

11.1 Introduction

Une première méthode pour améliorer l'efficacité de l'algorithme de Buchberger (Buchberger B., 1965; Buchberger B., 1979; Buchberger B., 1985) a déjà été décrite dans la section 9; le but de cette section est de décrire un nouveau critère pour remplacer les critères de Buchberger et éliminer *complètement* toutes les réductions inutiles lors d'un calcul de base de Gröbner (ou de manière équivalente générer des matrices de rang plein pour un algorithme matriciel de type F_4). En effet, les critères de Buchberger sont très efficaces puisqu'ils permettent d'éviter beaucoup de calculs inutiles; cependant, pour un pourcentage élevé d'exemples, après application des critères, 90% des paires critiques se réduisent à zéro après application des critères de Buchberger. Il faut préciser aussi qu'il n'est pas toujours possible d'éliminer *toutes* les réductions inutiles puisque ce n'est déjà pas vrai pour un système linéaire d'équations singulier (c'est à dire dont le rang n'est pas maximal). Il faut aussi indiquer qu'on cherche à obtenir ce résultat sans sacrifier l'efficacité de l'algorithme; ainsi des techniques calculant en même temps que la base de Gröbner le premier module des syzigies (Mora, T. and Möller, H.M. and Traverso, C., 1992) sont à exclure. On verra que l'algorithme F_5 (Faugère, 2002) permet, en pratique, de gagner un ordre de grandeur par rapport à l'algorithme F_4 sur une large classe d'exemples; ceci est également illustré dans plusieurs applications. On se reporte à l'article original (Faugère, 2002) pour une comparaison avec des méthodes connexes:

- les critères de Buchberger (Buchberger B., 1979; Buchberger B., 1985) et l'implantation de ces critères (Gebauer & Moller, 1986).
- les "staggered linear bases" (Gebauer & Moller, 1986)
- le calcul simultané de la base de Gröbner et du module des syzygies (Mora, T. and Möller, H.M. and Traverso, C., 1992).
- Le concept des bases involutives (V.P. Gerdt and Yu.A.Blinkov, 1998) peut être vu comme un moyen d'interdire certaines réductions et donc d'éviter des calculs. L'article (J. & R., 2002) présente un analogue du deuxième critère de Buchberger pour le calcul des bases involutives.

Notons toutefois qu'aucune de ces méthodes ne permet de répondre à l'ensemble des spécifications: élimination totale des calculs inutiles et efficacité; sselon les auteurs de l'article (Mora, T. and Möller, H.M. and Traverso, C., 1992): *"many useless pairs are discovered, but it involves a lot of extra computation, so the execution time is increased"*. Depuis la parution de l'article original, beaucoup de chercheurs ont étudié le critère et les notions théoriques associées. Ceci a donné lieu a de nouvelles idées pour optimiser les algorithmes ainsi que de nouvelles variantes de l'approche par signature (Eder, C., 2008; Eder, C. & Perry, J., 2010; Arri, A. & Perry, J., 2011). Beaucoup de nouvelles variantes de F_5ont ainsi été introduites comme par exemple , G2V (Gao, S. & Volny IV, F., 2010) resp. GVW (Gao, S. & Wang, D., 2010; Volny, F., 2011; Gao, S. & Wang, D., 2011; Gao, S. & Wang, D., 2013) ou SB (Roune, B. H. & Stillman, M., 2012a; Roune, B. H. & Stillman, M., 2012b). Il faut aussi mentionner l'existence de plusieurs articles essayant de classifier les différentes variantes d'algorithmes fondés sur l'utilisation des signatures (Huang, L., 2010; Eder, C. & Perry, J., 2011; Sun, Y. & Wang, D. K., 2011; Pan, S. & Wang, B., 2012; Pan, S. & Wang, B., 2013; Eder, C. & Roune, B. H., 2013). En conséquence, la littérature sur le sujet est tellement vaste qu'il est impossible de la résumer dans ce document. Avec Christian Eder (Eder & Faugère, 2014), nous avons donc proposé de classifier tous ces articles (voir figure 11.1 p. 42) dans le but d'identifier les différences et les similarités entre ces algorithmes.

La stratégie de l'algorithme F_5 est de calculer degré par degré *et* équation par équation les d-base de Gröbner des systèmes (f_m), (f_{m-1}, f_m), $\ldots$, $(f_1, \ldots, f_m)$. On montre (voir corollaire 7) que si le système initial est régulier alors il n'y a pas de réduction à zéro. De plus, en pratique, pour la plupart des systèmes il n'y a pas ou très peu de réduction à zéro. Même si la complexité du pire cas (doublement exponentiel) n'est pas amélioré on constate expérimentalement que pour une large classe de systèmes l'algorithme F_5 est plus rapide que toutes les autres implantation (en particulier l'algorithme F_4). L'idée de l'algorithme est d'abord détaillée pas à pas sur un exemple (11.2) puis une version matricielle (F_5 matriciel) est ensuite décrite: en effet, cette version est, d'une part, facilement implantable et d'autres le programme est déjà très efficace pour traiter des exemples denses (par exemple c'est cette version qui a

été utilisée pour résoudre le challenge HFE 1). De plus, cette version permet des extensions multiples comme le cas creux (Faugère *et al.* , 2014a) et le cas des bases SAGBI (Faugère & Rahmany, 2009).

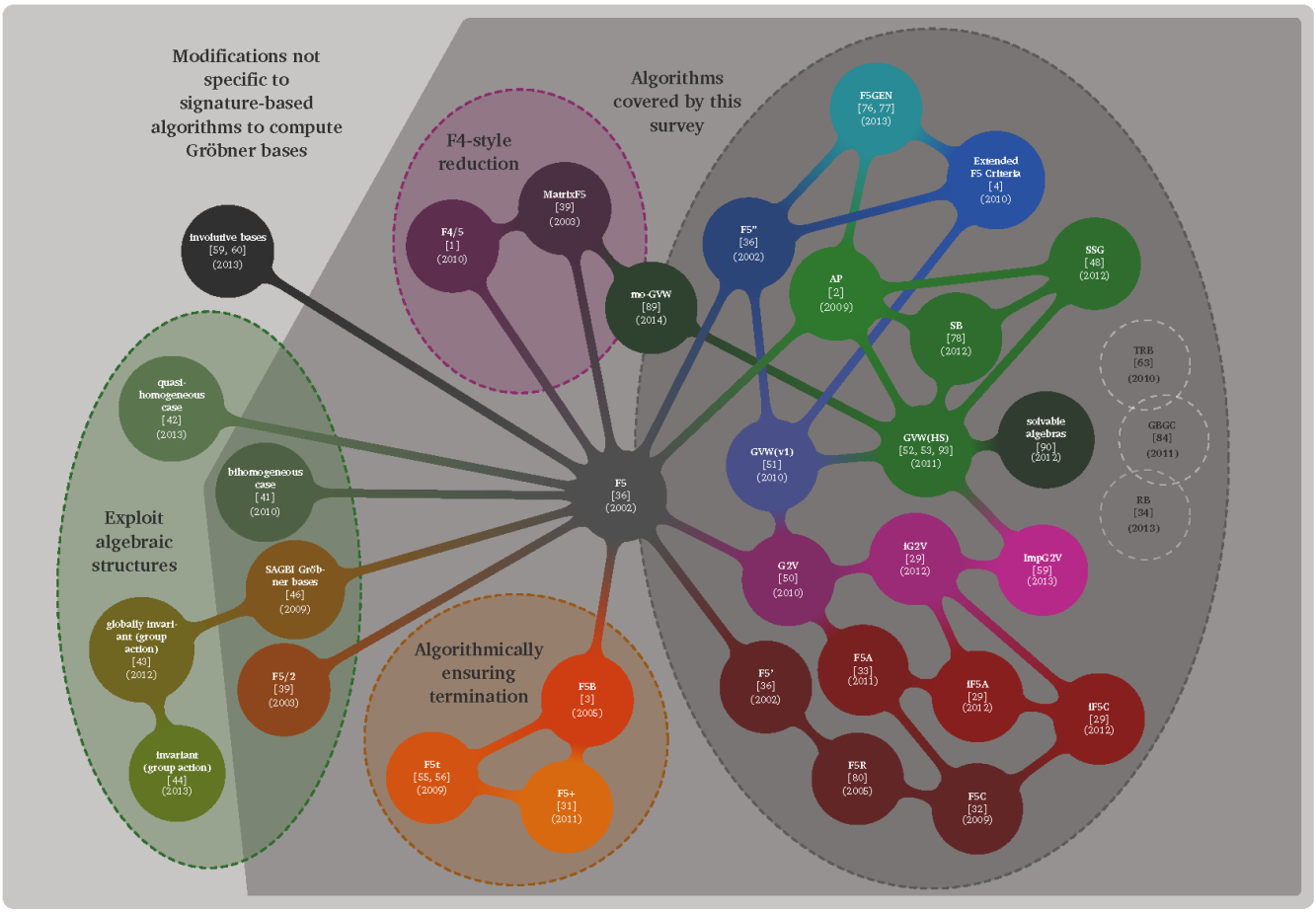


Figure 1: Dépendances entre divers algorithmes fondés sur les signatures (Janvier 2015)

11.2 L'idée préliminaire à l'algorithme F_5.

Afin d'introduire l'idée on considère un système algébrique de degré 2 en 3 variables x, y, z dépendant d'un paramètre b qui prendra la valeur 0 ou 1:

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

On veut calculer la base de Gröbner de f_1, f_2, f_3 modulo 23 pour un ordre du degré tel que $x > y > z$. Si on applique l'algorithme de Buchberger (avec les critères de Buchberger) il y a 5 paires critiques utiles et 5 paires inutiles (qui se réduisent à 0). Dans un premier temps on suppose $b = 0$. Pour calculer la base de Gröbner on procède degré

par degré. En degré 2 on construit directement la représentation matricielle (voir section 9) de $[f_1, f_2, f_3]$:

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

ce qui donne après mise sous forme triangulaire de la matrice A_2 (contrairement à la section 9 on calcule des mises sous forme triangulaire de matrice et pas la forme échelonnée complète, la raison en sera explicité plus bas):

$$B_2 = \begin{array}{c|cccccc} & x^2 & x\,y & y^2 & x\,z & y\,z & z^2 \\ f_3 & 1 & 18 & 19 & 8 & 5 & 7 \\ f_2 & & 1 & 3 & 2 & 4 & -1 \\ f_1 & & & 1 & -11 & -3 & -5 \end{array}$$

(afin de mieux faire ressortir la structure des matrices on remplace un 0 par un espace). Donc on a construit deux "nouveaux" polynômes dans l'idéal $f_4 = xy + 4\,yz + 2\,xz + 3\,y^2 - z^2$ et $f_5 = y^2 - 11\,xz - 3\,yz - 5\,z^2$.
En degré 3 on pourrait construire la représentation matricielle de

$$[x\,f_1, y\,f_1, z\,f_1, x\,f_2, y\,f_2, z\,f_2, x\,f_3, y\,f_3, z\,f_3]$$

et obtenir la matrice suivante:

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

Pour trianguler la matrice on peut réduire les lignes xf_2 et xf_1 par la ligne xf_3. Mais ce serait une perte de temps puisque ceci a déjà été réalisé à l'étape précédente: par exemple $f_4 = -f_2 + 3f_3$, et donc $xf_4 = -xf_2 + 3xf_3$. C'est une idée importante de l'algorithme de Buchberger: on utilise le plus possible ce qui a déjà été calculé en degré plus petit. Il est clair qu'on ne doit pas mettre f_1 et f_4 *simultanément* dans la même matrice car ces deux polynômes ne sont pas linéairement indépendants. Par conséquent on remplace, dans A_3, f_2 (resp. f_1) par f_4 (resp. f_5) et on calcule la représentation matricielle de $[x\,f_5, y\,f_5, z\,f_5, x\,f_4, y\,f_4, z\,f_4, x\,f_3, y\,f_3, z\,f_3]$:

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

Après mise sous forme triangulaire:

$$\widetilde{A_3'} = \begin{array}{c|cccccccccc} & x^3 & x^2y & xy^2 & y^3 & x^2z & xyz & y^2z & xz^2 & yz^2 & z^3 \\ xf_3 & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\ yf_3 & & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\ yf_2 & & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\ \mathbf{xf_2} & & & & \mathbf{1} & 0 & 0 & 8 & 1 & 18 & 15 \\ zf_3 & & & & & 1 & 18 & 19 & 8 & 5 & 7 \\ zf_2 & & & & & & 1 & 3 & 2 & 4 & 22 \\ zf_1 & & & & & & & 1 & 12 & 20 & 18 \\ \mathbf{yf_1} & & & & & & & & \mathbf{1} & 11 & 13 \\ \mathbf{xf_1} & & & & & & & & & \mathbf{1} & 18 \end{array}$$

Donc on construit 3 nouveaux polynômes (ligne dont l'index est en fonte grasse) comme par exemple $f_6 = y^3 + 8\,y^2z + xz^2 + 18\,yz^2 + 15\,z^3$; il faut noter que ce polynôme provient, en fait, de la ligne étiquetée par $x\,f_4$ elle même équivalente à $x\,f_2$; plus exactement on sait qu'on peut trouver un polynôme $g_{6,3}$ de degré 1 tel que

$$f_6 = (\alpha_6\,x + \cdots)\ f_2 + g_{6,3}\,f_3 \text{ avec } \alpha_6 \in \mathbb{K}$$

et de manière similaire il existe $(g_{8,2}, g_{8,3}, g_{7,2}, g_{7,3}) \in \mathbb{K}[x_1, \ldots, x_n]^4$ tels que:

$$f_7 = \left(\alpha_7 \boxed{\text{y}} + \cdots\right)\ f_1 + g_{7,2}\,f_2 + g_{7,3}\,f_3 \text{ avec } \alpha_7 \in \mathbb{K} \tag{10}$$

$$f_8 = \left(\alpha_8 \boxed{\text{x}} + \cdots\right)\ f_1 + g_{8,2}\,f_2 + g_{8,3}\,f_3 \text{ avec } \alpha_8 \in \mathbb{K} \tag{11}$$

Pour la suite de l'algorithme on peut noter également que si on remplace f_8 par une combinaison linéaire faisant intervenir les lignes f_7, f_8, f_6 alors l'écriture (11) est préservée: si $f_8^{'} = \beta_8\,f_8 + \beta_7\,f_7 + \beta_6\,f_6$ (avec $\beta_8 \neq 0$) alors

$$f_8^{'} = \left(\alpha_8^{'} \boxed{\text{x}} + \cdots\right)\ f_1 + g_{8,2}^{'}\,f_2 + g_{8,3}^{'}\,f_3 \text{ avec } \alpha_8^{'} \in \mathbb{K} \text{ et } \left(g_{8,2}^{'}, g_{8,3}^{'}\right) \in \mathbb{K}[x_1, \ldots, x_n]^2$$

En revanche ceci n'est pas vrai si on modifie la ligne f_7 par $f_7^{'} = \gamma_8\,f_8 + \gamma_7\,f_7 + \gamma_6\,f_6$ la structure (10) n'est pas conservée:

$$f_7^{'} = \left(\alpha_7^{'} \boxed{\text{x}} + \cdots\right)\ f_1 + g_{7,2}^{'}\,f_2 + g_{7,3}^{'}\,f_3 \text{ avec } \alpha_7^{'} \in \mathbb{K} \text{ et } \left(g_{7,2}^{'}, g_{7,3}^{'}\right) \in \mathbb{K}[x_1, \ldots, x_n]^2$$

Dans l'algorithme on pourra donc réduire la ligne f_8 par f_7 mais pas l'inverse; autrement dit on pourra réduire la ligne f_8 d'étiquette $x\,f_1$ par une ligne dont l'étiquette est strictement plus petite. Ceci explique pourquoi on se contente d'une forme triangulaire et pas d'une forme échelonnée.

En degré 4 un nouveau phénomène apparaît: la représentation matricielle de:

$$\left[x^2 f_i, x\,y f_i, y^2 f_i, x\,z f_i, y\,z f_i, z^2 f_i,\ i = 1, 2, 3\right]$$

n'est pas de rang plein ! (ceci correspond à 3 paires critiques inutiles dans l'algorithme de Buchberger). La raison en est que partant de la relation triviale $f_2 f_3 - f_3 f_2 = 0$ on peut la réécrire:

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

Cette écriture permet d'expliciter la dépendance linéaire entre les lignes et même, plus précisément, quelle ligne on peut retirer de la matrice: ici on enlève $x^2 f_2$ de A_4. En utilisant la deuxième relation triviale $f_1 f_3 - f_3 f_1 = 0$ on peut retirer de la même façon la ligne $x^2 f_1$ de la matrice A_4. Comme il existe une dernière relation triviale $f_1 f_2 = f_2 f_1$ on sait qu'il existe une autre ligne inutile dans la matrice A_4 mais comme on a déjà retiré $x^2 f_1$ quelle autre ligne peu

on retirer ? peut on retirer $x\,y\,f_1$? Pour cela il faut combiner les relations triviales entre elles (et dans ce calcul on reprend les polynômes originaux de $\mathcal{S}_b$ avec $b \in \{0,1\}$):

$$\begin{array}{rcl} 0 & = & (f_2 f_1 - f_1 f_2) - 3(f_3 f_1 - f_1 f_3) \\ 0 & = & (f_2 - 3f_3) f_1 - f_1 f_2 + 3 f_1 f_3 \\ 0 & = & f_4 f_1 - f_1 f_2 + 3 f_1 f_3 \\ 0 & = & \Big((1-b) \boxed{x\,y} + 4\,yz + 2\,xz + 3\,y^2 - z^2 \Big)\ f_1 \\ & & -(6x^2 + \cdots)\, f_2 + 3(6x^2 + \cdots)\, f_3 \end{array}$$

On en déduit qu'on peut retirer de la matrice A_4 la ligne $x\,y\,f_1$lorsque que $b \neq 0$ et la ligne $y\,z\,f_1$ lorsque $b = 0$. Par conséquent des calculs sont nécessaire pour savoir quelle est la ligne à éliminer (car cela dépend de la *valeur* de b). Plus généralement, si on combine les trois relations triviales $f_i f_j = f_j f_i$ on obtient:

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

où u, v, w sont des polynômes quelconques. En regroupant les termes:

$$(uf_2 + vf_3) f_1 - u f_1 f_2 - v f_1 f_3 + w f_2 f_3 - w f_3 f_2 = 0$$

Ainsi toutes les relations triviales $h f_1$ sont telles que h est dans l'idéal engendré par f_2 et f_3. Si on calcule une base de Gröbner G_{prev} de l'idéal engendré (f_2, f_3) on élimine les lignes $m\,f_1$ où m est un monôme divisible par le terme de tête d'un élément de G_{prev}. Plus généralement, si on a calculé une base de Gröbner de $(f_2, \ldots, f_m)$, lors du calcul de la base de Gröbner de $(f_1) + G_{\text{prev}}$ on va construire des matrices dont les étiquettes des lignes sont de la forme $m\,f_1$ avec m un monôme irréductible par rapport à G_{prev}.

Pour terminer l'exemple (on suppose à nouveau $b = 0$) et afin d'utiliser les calculs en degré 2 et 3 on applique les règles de réécriture (dans cet ordre):

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

Maintenant les lignes de la matrice A_4 sont donc

$$\left[y f_7, z f_8, z f_7, z^2 f_5, y f_6, y^2 f_4, z f_6, y\,z f_4, z^2 f_4, x^2 f_3, x\,y f_3, y^2 f_3, x\,z f_3, y\,z f_3, z^2 f_3 \right]$$

et la matrice est

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

Ainsi A_4 est pratiquement triangulaire si on excepte un bloc 5×5 :

$$A_4' = \begin{array}{c|ccccc|} & xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 \\ z^2f_4 & 1 & 3 & 2 & 4 & 22 \\ z^2f_5 & & 1 & 12 & 20 & 18 \\ zf_7 & & & 1 & 11 & 13 \\ zf_8 & & & & 1 & 18 \\ yf_7 & 1 & 11 & & 13 & \\ \end{array} \tag{12}$$

La réduction de la matrice donne le nouveau polynôme $f_9 = z^4$ et comme il comme correspond à la ligne d'étiquette $y\, f_7 \approx y^2\, f_1$ on sait qu'on peut trouver une écriture de f_9 de la forme:

$$f_9 = \left(\alpha_9\, y^2 + \cdots\right)\, f_1 + (\cdots)\, f_2 + (\cdots)\, f_3.$$

Dans cet exemple le calcul se fait sans aucune réduction à zéro. Il en ressort également que dans l'algorithme il est nécessaire de conserver l'étiquette originelle de chaque ligne pour chaque polynôme calculé: se sera la "signature". Par exemple la signature de f_6 sera $x f_2$.

12 Algorithme F_5 matriciel

Le but de cette section est la description de la version matricielle de l'algorithme F_5 qui sera adaptée à une analyse de complexité et efficace pour des systèmes denses. Une différence fondamentale avec la version originale de l'algorithme F_5 (Faugère, 2002) est que, d'une part le degré maximal atteint par les calculs doit être donné en entré de l'algorithme (c'est le paramètre D dans la description de l'algorithme 21); d'autre part on ne tient pas compte du caractère éventuellement creux des polynômes. L'algorithme qui en résulte est particulièrement simple à décrire et à implanter.

12.1 Représentation matricielle avec étiquette

Comme cela a été illustré dans la section 11.2, il nécessaire d'adjoindre à chaque polynôme, et donc à chaque ligne de matrice, une *étiquette;* pour cette raison il nous faut étendre la définition de la représentation matricielle 23:

Définition 30. *Si $F = [f_1, \ldots, f_m]$ est un vecteur de m polynômes, $e = [e_1, \ldots, e_m]$ un vecteur de m étiquettes et $<$ un ordre admissible, $\mathrm{T}_<(F) = [t_1, \ldots, t_l]$ les termes du support de F triés pour l'ordre $<$. Alors une représentation matricielle $M_{e,T_<(F)}(F)$ de F est une matrice dont les lignes sont indexées par $[e_1, \ldots, e_m]$:*

$$M_{e,T_<(F)}(F) = \begin{array}{c|ccc|} & t_1 & t_2 & \cdots \\ e_1 & \cdots & \cdots & \cdots & f_1 \\ e_2 & \cdots & \cdots & \cdots & f_2 \\ e_3 & \cdots & \cdots & \cdots & f_3 \end{array}$$

et dont le coefficient d'indice (e_i, j) est le coefficient du terme t_j dans f_i . Dans la notation précédente on ajoute la colonne de droite simplement pour indiquer l'origine des coefficients de la matrice; cette colonne est donc optionnelle dans la notation. De plus, $M_{e,T_<(F)}(F)$ vérifie l'équation:

$$F = M_{e,T_<(F)}(F) \cdot T_<(F)$$

Pour alléger les notations on note $M_e(F)$ ou $M(F)$ la matrice $M_{e,T_<(F)}(F)$.

Réciproquement si M_e est une matrice étiquetée par $e = [e_1, \ldots, e_m]$, de taille $m \times l$, à coefficients dans $\mathbb{K}$, et $X = [t_1, \ldots, t_l]$ un vecteur de termes alors la représentation polynomiale étiquetée de M_e par rapport à X est le vecteur de m polynômes déterminé par l'équation $F = M \cdot X$ et d'étiquettes e.

Par exemple la matrice A_4' (équation 12) d'étiquettes $\left[z^2 f_2, z^2 f_1, yz\, f_1, xz\, f_1, y^2\, f_1\right]$:

$$A_4' = \begin{array}{c|cccccc|c} & xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 & \\ z^2f_2 & 1 & 3 & 2 & 4 & 22 & z^2f_4 \\ z^2f_1 & & 1 & 12 & 20 & 18 & z^2f_5 \\ yz\,f_1 & & & 1 & 11 & 13 & zf_7 \\ xz\,f_1 & & & & 1 & 18 & zf_8 \\ y^2\,f_1 & 1 & 11 & & 13 & 0 & yf_7 \end{array}$$

devient après mise sous forme triangulaire (on utilise la même notation˜mais on autorise uniquement une réduction de ligne par une ligne dont l'étiquette est strictement plus petite):

$$\widetilde{A_4'} = \begin{array}{c|cccccc|c} & xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 & \\ z^2f_2 & 1 & 3 & 2 & 4 & 22 & z^2f_4 \\ z^2f_1 & & 1 & 12 & 20 & 18 & z^2f_5 \\ yz\,f_1 & & & 1 & 11 & 13 & zf_7 \\ xz\,f_1 & & & & 1 & 18 & zf_8 \\ y^2\,f_1 & & & & & 1 & \end{array}$$

et la dernière ligne de cette matrice $\widetilde{A_4'}$ correspond au polynôme z^4 d'étiquette $y^2\,f_1$. Dans la description de l'algorithme il sera commode d'employer une notation simple pour décrire la façon de rajouter une ligne $f \in \mathbb{K}[x_1, \ldots, x_n]$ d'étiquette ε à une matrice étiquetée $M_e(F)$:

$$M_{e+[\varepsilon]}(F + [f]) = \begin{array}{c|c|c} & M_e(F) & \\ \varepsilon & \cdots & f \end{array}$$

Par exemple

$$xz\,f_1 \left| \begin{array}{c} \widetilde{A_4'} \\ \cdots \end{array} \right| z^4$$ désigne la matrice:

$$\begin{array}{c|cccccc|c} & xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 & \\ z^2f_2 & 1 & 3 & 2 & 4 & 22 & z^2f_4 \\ z^2f_1 & & 1 & 12 & 20 & 18 & z^2f_5 \\ yz\,f_1 & & & 1 & 11 & 13 & zf_7 \\ xz\,f_1 & & & & 1 & 18 & zf_8 \\ xz\,f_1 & & & & 1 & 18 & zf_8 \\ y^2\,f_1 & & & & & 1 & \end{array}$$

(remarquer qu'on trie les lignes et les colonnes).

12.2 Algorithme F_5 matriciel

On va décrire maintenant l'algorithme F_5 matriciel. Afin d'unifier le cas général et le cas particulier $\mathbb{F}_2$ on utilise la notation suivante: $\delta_{\mathbb{K},\mathbb{F}_2}$, le symbole de Kronecker, est égal à 1 si $\mathbb{K} = \mathbb{F}_2$ et 0 sinon. En effet dans le cas où cherche à calculer une base de Gröbner sur $\mathbb{F}_2$d'un système algébrique $[f_1, \ldots, f_m]$ on doit ajouter les équations de corps $x_i^2 - x_i$. Par conséquent, il faut tenir compte de nouvelles relations triviales

$$f^2 = f$$

provenant de l'action du morphisme de Frobenius (voir la section 14 pour la justification du nouveau critère). À noter que c'est cette version de F_5 matricielle qui a été utilisée pour résoudre le challenge 1 de HFE .

Avec ces notations on peut décrire très simplement l'algorithme F_5 matriciel:

Algorithme 21. *Algorithme F_5matriciel.*

Input: $\begin{cases} \textit{le corps } \mathbb{K} \\ F = [f_1, \ldots, f_m] \textit{ polynômes de degrés total } d_1 \leqslant \cdots \leqslant d_m, \\ \textit{un entier } D \end{cases}$

Output*: une D-base de Gröbner de F pour un ordre admissible $<$.*

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

for d ***from*** d_1 ***to*** D ***do*** // Boucle sur le degré

  for i ***from*** 1 ***to*** m ***do*** // Boucle sur les équations

    // Construire la nouvelle matrice $M^{(d)}([f_1, \ldots, f_i])$*:*

    if $d = d_i$ ***then***

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

    else

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

    // $J_{\textit{Critères}}$ *est un l'*idéal monomial:

$$J_{\textit{Critères}} := \mathrm{Id}\left(\mathrm{LT}\left(\widetilde{M^{(d-d_i)}}([f_1, \ldots, f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}])\right)\right)$$

    for *toute ligne f d'étiquette t f_i dans* $\widetilde{M^{(d-1)}}([f_1, \ldots, f_i])$ ***do***

      Soit k le plus grand entier tel que x_k divise t

      for j ***from*** $k + \delta_{\mathbb{K},\mathbb{F}_2}$ ***to*** n ***do***

        if $t\, x_j \notin J_{\textit{Critères}}$ ***then***

          // Si $\mathbb{K} = \mathbb{F}_2$ *alors* dans la matrice suivante on

          // remplace dans le produit $x_j f$ les carrés x_i^2 par x_i.

$$M^{(d)}([f_1, \ldots, f_i]) := \begin{array}{c} tx_j f_i \\ \\ \end{array} \left| \begin{array}{c} \widetilde{M^{(d)}}([f_1, \ldots, f_i]) \\ \cdots \end{array} \right| \begin{array}{c} x_j f \\ \\ \end{array}$$

    Calculer $\widetilde{M^{(d)}}([f_1, \ldots, f_i])$ *la réduction de Gauß (en respectant l'ordre des étiquettes).*

return *la représentation polynomiale de* $\widetilde{M^{(D)}}([f_1, \ldots, f_m])$

13 Algorithme F_5 version simplifiée

Soit I' un idéal de $\mathbb{K}[x_1, \ldots, x_n]$; on suppose qu'on a déjà calculé une base de Gröbner G'de I' pour un ordre fixé $<$ et qu'on veut calculer une base de Gröbner de l'idéal $(f_1) + I'$pour $f_1 \in \mathbb{K}[x_1, \ldots, x_n]$. Les différences entre l'algorithme de cette section avec l'algorithme F_5 matriciel (21) de la section précédente sont:

1. il n'est pas nécessaire de donner le degré maximal D des calculs.
2. c'est une version adaptative de l'algorithme matriciel (on ne considère pas systématiquement le produit par toutes les variables).

13.1 Signature d'un polynôme

On veut tout d'abord définir une signature unique et canonique pour tous les éléments de $T(I) \subset T$ les termes de tête de tous les polynômes de l'idéal I.

Définition 31. *Pour tout p dans l'idéal I on définit $w(p)$ comme étant $min_<\{g_1 \in \mathbb{K}[x_1, \ldots, x_n] \,|\, (g_1 f_1 - p) \in I'\} = min_< ((I' : (f_1))/I')$. On définit $v_1(p)$ comme étant le terme de tête de $w(p)$:*

$$v_1 : \begin{pmatrix} I & \longrightarrow & T(I) \\ p & \longmapsto & v_1(p) = \mathrm{LT}_<(w(p)) \end{pmatrix}.$$

Proposition 45. *Si p_1 et p_2 sont deux polynômes de I tels que* $\mathrm{LT}(p_1) \neq \mathrm{LT}(p_2)$ *alors* $v_1(p_1) \neq v_1(p_2)$.

Dans l'algorithme F_5, $v_1(p)$ sera la *signature* du polynôme p: elle est unique et ne dépend pas des calculs. Le nouveau critère utilise la signature pour éliminer des paires critiques. Par rapport à la version matricielle de F_5 la signature correspond à l'étiquette associée à chaque ligne de la matrice. Dans la représentation interne il faut inclure la signature dans la représentation interne des polynômes; mathématiquement on doit "agrandir" l'anneau des polynômes:

Définition 32. *On définit un nouvel anneau de polynômes $\mathcal{R} = T \times \mathbb{K}[x_1, \ldots, x_n]$. Pour tout $r \in \mathcal{R}$,si $r = (t, f)$ alors on définit la projection $\pi(r) = f \in \mathbb{K}[x_1, \ldots, x_n]$ et signature de r comme étant $\mathcal{S}(r) = t \in T$.*

On verra que durant l'exécution de l'algorithme F_5 on aura toujours $\mathcal{S}(r) = v_1(\pi(r))$.

Définition 33. *(admissibilité) On dit que $r \in \mathcal{R}$ est admissible s'il existe $g_1 \in \mathbb{K}[x_1, \ldots, x_n]$ tel que $g_1 f_1 - \pi(r) \in I'$ et $\mathrm{LT}(g_1) = \mathcal{S}(r)$.*

Remarque 17. *Avec les notations de l'exemple 11.2, p. 42 on a: $I' = \mathrm{Id}(f_3)$ et on considère $I = (f_2) + I'$:*

$$r_2 = (1, 3\,x^2 + 7xy + 22\,xz + 11\,yz + 22\,z^2 + 8\,y^2)$$
$$r_6 = (x, y^3 + 8\,y^2 z + xz^2 + \cdots) = (x + \cdots) f_2 + (\cdots) f_3$$

Remarque 18. *Pour tout $r \in R$ tel que $\pi(r) \in I'$ on a $\mathcal{S}(r) = 0$.*

Définition 34. *Soient $0 \neq \lambda \in \mathbb{K}$, $v \in T$, et $r = (u, p) \in \mathcal{R}$ on définit $\lambda r = (u, \lambda p)$ et $vr = (uv, vp)$.*

Remarque 19. *On doit aussi étendre la définition des opérateurs usuels sur $\mathcal{R}$:*

pour $r \in \mathcal{R}$, $\mathrm{LT}(r) = \mathrm{LT}(\pi(r))$ et $\mathrm{LC}(r) = \mathrm{LC}(\pi(r))$.
pour $r \in \mathcal{R}$, et $G \subset \mathbb{K}[x_1, \ldots, x_n]$, $\mathrm{NF}(r, G, <) = (\mathcal{S}(r), \mathrm{NF}(\pi(r), G, <))$.
pour $r, r' \in \mathcal{R}^2$, $\mathrm{lcm}(r, r') = \mathrm{lcm}(\mathrm{LT}(\pi(r)), \mathrm{LT}(\pi(r')))$
R est une base de Gröbner si $\pi(R)$ est une base de Gröbner.

13.2 Réductibilité, paire critique, S-polynôme au sens de F_5

Dans le reste de cette section on restreint les réduction valides:

Définition 35. *(normalisé)*
On dit que $r \in \mathcal{R}$ est normalisé si $\mathcal{S}(r)$ est 0 ou n'est pas top-réductible par I'. On dit que $(u, r) \in T \times \mathcal{R}$ est normalisé si $u\,r$ est normalisé. On dit encore qu'une paire critique orientée $(r, r') \in \mathcal{R}^2$ est normalisée si les deux conditions sont vraies:

(i) $u'\mathcal{S}(r') \prec u\mathcal{S}(r)$

(ii) (u, r) *et* (u', r') *sont normalisées où* $u = \frac{lcm(r,r')}{\mathrm{LT}(r)}$, $u' = \frac{lcm(r,r')}{\mathrm{LT}(r')}$.

Remarque 20. *Si G' est une base de Gröbner de I' l'implantation de la définition précédente est immédiate: r est normalisé si et seulement si $\mathcal{S}(r)$ n'est pas réductible par G'.*

Remarque 21. *(suite de l'exemple)* $r_6 = (x, y^3 + 8\,y^2 z + xz^2 + 18\,yz^2 + 15\,z^3)$
$x \times r_6$ *n'est pas normalisé car* $x\,\mathcal{S}(r_6) = x^2$ *et* $x^2 \in \mathrm{LT}(f_3)$
$y \times r_6$ *est normalisé car* $y\,\mathcal{S}(r_6) = xy$ *et* $xy \notin \mathrm{LT}(f_3)$

Définition 36. *(S-polynôme)*
Soit (r, r') une paire critique normalisée alors

$$\mathrm{Spol}(r, r') = \left(\frac{\mathrm{lcm}(r, r')}{\mathrm{LT}(r)} \mathcal{S}(r), \mathrm{Spol}(\pi(r), \pi(r')) \right) \in \mathcal{R}$$

Définition 37. *(strong reduction)*
Soient $r, r', r'' \in \mathcal{R}$ et R un sous ensemble de $\mathcal{R}$. Alors on dit que

(i) *r est fortement réductible par r', si $\mathrm{LT}(r')$ divise $\mathrm{LT}(r)$ et*

$$\frac{\mathrm{LM}(r)}{\mathrm{LM}(r')}\mathcal{S}(r') \leqslant \mathcal{S}(r)$$

Dans ce cas la réduction r par r' est:

$$r \xrightarrow{r'} (\mathcal{S}(r), \pi(r) - \frac{\mathrm{LM}(r)}{\mathrm{LM}(r')}\pi(r')) \in \mathcal{R}$$

(ii) *r est fortement réductible par R s'il existe $r' \in R$ tel que r est fortement réductible par r'. Notation*

$$r \xrightarrow{R} r''$$

$\xrightarrow{R^*}$ est la clôture réflexive transitive de $\xrightarrow{R}$.

Proposition 46. *Soient $r, r' \in \mathcal{R}$ et R un sous ensemble de $\mathcal{R}$. Alors*

(i) *si la paire (r, r') est normalisée $\mathrm{Spol}(r, r')$ est admissible.*

(ii) *pour $\lambda \in \mathbb{K}$ et $t \in T$, $(\lambda t)\, r$ est admissible.*

(iii) *si $r \xrightarrow{R^*} r'$ alors il existe $t \leqslant \mathcal{S}(r')$ tel que $(t, \pi(r'))$ est admissible.*

13.3 Nouveau critère F_5

Le théorème principal de cette section est aussi une nouvelle caractérisation des bases de Gröbner:

Théorème 47. *(critère F_5) Soit G' une base de Gröbner d'un idéal I' et f_1 un polynôme. Soit R un sous ensemble fini de $\mathcal{R}$ tel que:*

(i) *tous les éléments de $r \in R$ sont admissibles et unitaires.*

(ii) *$\{r \in R \mid \mathcal{S}(r) = 0\} = G'$ et $(1, f_1) \in R$.*

(iii) *pour toute paire critique normalisée $(r, r') \in R$, on a $\mathrm{Spol}(r, r') \xrightarrow{R^*} 0$.*

Alors R est une base de Gröbner (non réduite) de $I = (f_1) + I'$.

Remarque 22. *Dans le théorème si on restreint la condition (iii) aux paires critiques de degré $\leqslant d$ alors on obtient une $d-$base de Gröbner.*

13.4 Réduction au sens de F_5

Lorsqu'on réduit un des polynômes on doit garder tous les polynômes admissibles et on doit modifier la définition usuelle de réduction; ainsi dans la fonction TopReduction(r, r') si r est réductible par r': il existe $t \in T$ tel que $\mathrm{LM}(r') \cdot t = \mathrm{LM}(r)$ et si r, r' sont unitaires et admissibles on peut trouver $(w, w') \in \mathbb{K}[x_1, \ldots, x_n]^2$ et $(h, h') \in {I'}^2$ tels que

$$\pi(r) = wf_1 + h,\ \pi(r') = w'f_1 + h' \text{ et } \mathcal{S}(r) = \mathrm{LT}(w),\ \mathcal{S}(r') = \mathrm{LT}(w').$$

Par conséquent

$$\pi(\tilde{r}) = \pi(r) - t\pi(r') = (w - tw')\, f_1 + (h - th')$$

et une condition suffisante pour que $\mathrm{LT}(w - tw') = \mathrm{LT}(w)$ est que $t\,\mathrm{LT}(w') < \mathrm{LT}(w)$ c'est à dire $t \cdot \mathcal{S}(r') < \mathcal{S}(r)$; dans ce cas $(\mathcal{S}(r), \pi(r) - t\pi(r'))$ est admissible.

Sinon, $t\,\mathrm{LT}(w') < \mathrm{LT}(w)$, et on doit construire un *nouveau polynôme* $(t\mathcal{S}(r'), \pi(r) - t\pi(r'))$ qui est admissible (si on pense l'algorithme en termes d'algèbre linéaire cela signifie qu'on doit échanger deux lignes dans la matric). Donc il est nécessaire de retourner plusieurs polynômes dans la fonction TOPREDUCTION et c'est donc une différence essentielle avec l'algorithme de Buchberger. De plus on doit s'assurer de ne pas générer deux polynômes avec la même signature: on maintient donc une liste de termes $\mathcal{A}$ contenant toutes les signatures déjà crées: un réducteur potentiel r' sera considéré seulement si $\mathcal{S}(r')$ n'est pas déjà dans $\mathcal{A}$.

Algorithme 22. TOPREDUCTION

Input: $r \in \mathcal{R},\ R \subset \mathcal{R}, \mathcal{A} \subset T$
Output: (r'', S) *où* $\begin{cases} r'' = \frac{r}{\mathrm{LC}(r)} \text{ si } r \text{ est réductible et } \varnothing \text{ sinon.} \\ S \text{ est un ensemble de } 0, 1 \text{ ou } 2 \text{ polynômes} \end{cases}$
if $\pi(r) = 0$ ***then***
  return $(\varnothing, \varnothing)$ // réduction à zéro !
for $r' \in R$ ***do***
  if $(t = \frac{\mathrm{LT}(r')}{\mathrm{LT}(r)} \in T)$ *et* $((t, r')$ *est normalisé) et* $((t\mathcal{S}(r') \notin \mathcal{A})$ ***then***
    if $\mathcal{S}(tr') \prec \mathcal{S}(r)$ ***then***
      return $(\varnothing, \{(\mathcal{S}(r), \pi(r) - t\,\pi(r'))\})$
    else
      $r' = (t\,\mathcal{S}(r'), t\,\pi(r') - \pi(r)) \in \mathcal{R}$
      $\mathcal{A} = \mathcal{A} \cup \mathcal{S}(r')$
      return $(\varnothing, \{r', r\})$
return $(\frac{1}{\mathrm{LC}(r)}\, r, \varnothing)$

Lemme 3. *Si* $\{r\} \cup R$ *est constitué de polynômes admissibles alors tous les polynômes retournés par* TOPREDUCTION *sont admissbles.*

Proof. Ceci découle de la discussion au début de ce paragraphe. □

On peut maintenant décrire la fonction REDUCTION qui est une fonction globale au sens de la réduction dans l'algorithme F_4.

Algorithme 23. REDUCTION

Input: F, R *listes de polynômes normalisés dans* $\mathcal{R}$
Output: *une liste de polynômes.*
$F =$ *éliminer dans* F *les doublons (pour la signature)*
$\mathcal{A} = \mathcal{S}(F)$, $R' = \varnothing$
while $F \neq \varnothing$ ***do***
  Prendre et retirer dans F *l'élément* r *avec la plus petit signature.*
  $(r', F') =$TOPREDUCTION$(r, R \cup R', \mathcal{A})$
  $R' = R' \cup \{r'\}$ *et* $F = F \cup F'$
return R'

13.5 Version simplifiée de l'algorithme F_5

On présente maintenant une version simplifiée de l'algorithme. On essaye d'être aussi proche que possible de la présentation de l'algorithme de Buchberger. Soit I' un idéal. On suppose qu'on a déjà calcule une base de Gröbner G' et on veut calculer la base de Gröbner de $(f_1) + I'$. On décrit maintenant l'algorithme utilisant le nouveau critère du théorème 47. L'implantation du test de normalisation et de calcul du S-polynôme est immédiat à partir des définitions 35 et 36 (voir aussi la remarque 20). L'algorithme fait appel à la fonction REDUCTION (algorithme 23) qui retourne une liste de polynômes réduits (comme dans l'algorithme F_4) . On verra dans l'exemple 13.7 qu'on peut remplacer cette fonction par une fonction de réduction totale.

Algorithme 24. *(F_5 version de base)*

***Input**: f_1 un polynôme et G' une base de Gröbner*
***Output**: une base de Gröbner de $(f_1) + G'$*
$R = \{(0, g') \mid g' \in G'\} \cup \{(1, f_1)\}$
$P = [(r, r') \mid r' \in G'$ *et* (r, r') *est normalisée*$]$
while $P \neq \varnothing$ ***do***
  Soit d_0 *le degré minimal de* P
  $P_{d_0} = \{p \in P \mid \deg(p) = d_0\}$
  $R_{d_0} =$REDUCTION$(\mathrm{Spol}(P_{d_0}), R)$
  for $r \in R_{d_0}$ ***do***
    $P = P \cup [(r, r') \mid r' \in R$ *et* (r, r') *normalisée*$]$
    $R = R \cup \{r\}$
return R

Remarque 23. *Pour un polynôme homogène f_1, une variante de l'algorithme est de ne garder dans la liste des paires critiques P que les paires de degré $\leqslant d$; dans ce cas on obtient une $d-$base de Gröbner. On note $d-$Gröbner F_5 cette variante de l'algorithme F_5.*

13.6 Preuve de l'algorithme F_5

On suppose que le polynôme f_1 est homogène. Soit $\tilde{R}$ l'ensemble de tous les polynômes qui sont générés pendant l'algorithme. On donne la preuve de l'algorithme.

Proposition 48. *Pour tout $r \in \tilde{R}$, r est admissible et normalisé.*

Proof. Au départ $(1, f_1)$ est admissible. Les nouveaux polynômes sont crées dans:

(i) dans le calcul du S-polynôme: $\mathrm{Spol}(P_d)$ et par définition la paire est normalisée.

(ii) dans TOPREDUCTION : on utilise alors le lemme 3.

□

Théorème 49. *Pour tout d, le résultat de $d-$Gröbner F_5 est une base de Gröbner jusqu'au degré d.*

Proof. Pour appliquer le théorème 47 on considère une paire critique normalisée $(r, r') \in R$ et d son degré. Soit r'' le résultat de la réduction de $\mathrm{Spol}(r, r')$ par R_{d-1} et $\tau = \mathrm{lcm}(\mathrm{LT}(r), \mathrm{LT}(r'))$.En notant $u = \frac{\tau}{\mathrm{LT}(r)}$ on a:

$$\mathcal{S}(r'') = u\,\mathcal{S}(r)$$
$$\text{et } \mathrm{LT}(\pi(r'')) < \mathrm{lcm}(\mathrm{LT}(r), \mathrm{LT}(r')) = u\,\mathrm{LT}(r)$$

et donc

$$\mathrm{Spol}(r, r') \xrightarrow{R_{d-1}} r'' \xrightarrow{R_{d-1} \cup \{r\}} 0$$

Comme $r'' \in R_d$ on en déduit $\mathrm{Spol}(r, r') \xrightarrow{R_d} 0$. D'après le lemme 48 on peut appliquer le théorème 47 et on en déduit que R_d est une base de Gröbner de l'idéal engendré par $(f_1) + I$ jusqu'au degré d. □

Théorème 50. *Si on suppose que le polynôme f_1 est homogène et qu'il n'y a pas de réduction à zéro pendant le calcul. On note R_d le résultat de* REDUCTION *dans l'algorithmeF_5 24 et R la valeur de R dans l'algorithme avant* REDUCTION. *Alors $Id(\mathrm{LT}(R)) \neq Id(\mathrm{LT}(R \cup R_d))$.*

Proposition 51. *Soit PSyz le module généré par les syzygies principales (triviales). Pour un système polynômial générique $(f_1, \ldots, f_m)$, alors Syz = PSyz.*

Théorème 52. *Lorsque l'algorithme trouve une réduction à zéro , $r_{i_k} \to 0$ alors ceci implique qu'il existe $\mathbf{s} \in$ Syz\PSyz avec $\mathrm{LT}(\mathbf{s}) = \mathcal{S}(r_{i_k})$.*

Corollaire 7. *Si le système initial est une suite régulière alors il n'y a pas de réduction à zéro.*

13.7 F_5: exemple pas à pas

On calcule maintenant la base de Gröbner pour un exemple tiré de l'article (Mora, T. and Möller, H.M. and Traverso, C., 1992). L'ordre utilisé est l'ordre DRL tel que $x > y > z > t$ et les coefficients sont les nombres rationnels ($\mathbb{Q}$):

$$\begin{cases} f_1 = yz^3 - x^2t^2 \\ f_2 = xz^2 - y^2t \\ f_3 = x^2y - z^2t \end{cases}$$

L'algorithme F_5 calcule successivement les bases de Gröbner des idéaux $\mathrm{Id}(f_3)$, $\mathrm{Id}(f_2, f_3)$ et $\mathrm{Id}(f_1, f_2, f_3) = \langle f_1 \rangle + \mathrm{Id}(f_2, f_3)$. Comme le dernier calcul est le plus significatif on décrit uniquement ce dernier cas. Les polynômes de la base de Gröbner $G' = \pi(R')$ de $\mathrm{Id}(f_2, f_3)$:

$R' = [r_3, r_2, r_4, r_5]$ où $r_3 = (0, f_3)$, $r_2 = (0, f_2)$, $r_4 = (0, x\,y^3\,t - z^4\,t)$, $r_5 = (0, z^6\,t - y^5\,t^2)$.

On note $\varphi' = \mathrm{NF}(., [r_3, r_2, r_4, r_5])$ la forme normale par rapport à $G\prime$.

$r_1 = (1, f_1)$

$R = R' \cup \{r_1\} = [r_3, r_2, r_4, r_5, r_1]$

Il y a quatre paires critiques: $p_7 = (x\,y\,z^3, x, r_1, y\,z, r_2)$, $p_8 = (x^2\,y\,z^3, x^2, r_1, z^3, r_3)$, $p_9 = (y\,z^6\,t, z^3\,t, r_1, y, r_5)$, $p_{10} = (x\,y^3\,z^3\,t, x\,y^2\,t, r_1, z^3, r_4)$.

Les signatures de $\mathcal{S}(p_7), \ldots, \mathcal{S}(p_{10})$ sont x, x^2, z^3, $x\,y^2$ sont toutes invariants par φ' donc les paires sont normalisées.

$P = [p_7, p_8, p_9, p_{10}]$

$\boxed{d_0 = 5}$, on calcule $\mathrm{Spol}(P_5)$ avec $P_5 = [p_7]$ et $P = [p_8, p_9, p_{10}]$

$r_6 = (x,\ y^3\,z\,t - x^3\,t^2)$ et $F := [r_6]$

On ajoute la règle de réécriture $x \to r_6$

Il n'y a aucune réduction possible de r_6 par G' donc le résultat retourné est $R_5 = [r_6]$

$R = [r_3, r_2, r_4, r_5, r_1, r_6]$

On met à jour la liste de paires critiques: $p_{11} = (y^3\,z^3\,t, z^2, r_6, y^2\,t,\ r_1)$, $p_{12} = (y^3\,z^6\,t, z^5, r_6, y^3, r_5)$, $p_{13} = (x\,y^3\,z\,t, x, r_6, z\,t, r_4)$, $p_{14} = (x^2\,y^3\,z\,t, x^2, r_6, y^2\,z\,t, r_3)$, $p_{15} = (x\,y^3\,z^2\,t, x\,z, r_6, y^3\,t\,, r_2)$. On vérifie que $\mathcal{S}(z^2\,r_6) = xz^2$ et $\mathcal{S}(z^5\,r_6) = x\,z^5$ sont réductible par φ' donc les paires p_{11} et p_{12} sont *éliminées* par le critère de F_5. Donc $P = [p_8, p_9, p_{10}, p_{13}, p_{14}, p_{15}]$.

$\boxed{d = 6}$, on calcule $\mathrm{Spol}(P_6)$ avec $P_6 = [p_8, p_{13}]$ et $P = [p_9, p_{10}, p_{14}, p_{15}]$

Comme on a la règle $x^2\,r_1 \longrightarrow x\,r_6$ on élimine la paire critique p_8.

En revanche on garde l'autre paire p_{13} puisque Rewritten?$(x, r_6) = false$ et Rewritten?$(z, r_4) = false$; par suite $r_7 = (x^2, z^5\,t - x^4\,t^2)$.

On ajoute la règle $x^2 \longrightarrow r_7$ et il n'y a pas de réduction possible de r_7 par R donc on retourne $R_6 = [r_7]$ et maintenant $R = [r_3, r_2, r_4, r_5, r_1, r_6, r_7]$.

Parmi toutes les paires critiques possibles on vérifie que (r_7, r_1), (r_7, r_6), (r_7, r_3) et (r_7, r_4) sont *éliminées* par le critère de F_5.

Les nouvelles paires normalisées sont $p_{16} = (z^6\,t, z, r_7, 1, r_5)$ et $p_{17} = (x\,z^5\,t, x, r_7, z^3\,t, r_2)$.

$\boxed{d = 7}$, on calcule $\mathrm{Spol}(P_7)$ avec $P_7 = [p_{15}, p_{16}, p_{17}, p_{14}]$ et $P = [p_9, p_{10}]$.

Comme on a la règle $x\,z\,r_6 \longrightarrow z\,r_7$ on élimine la paire critique p_{15}.

p_{16} est normalisée et on calcule $r_8 = (x^2\,z\,, y^5\,t^2 - x^4\,z\,t^2)$ puis on ajoute la règle $x^2\,z \to r_8$.

p_{17} est normalisée et on calcule $r_9 = (x^3\,\mathbf{F}_1, -x^5\,t^2 + y^2\,z^3\,t^2)$ puis on ajoute la règle $x^3 \to r_9$.

Comme on a la règle $x^2\,r_6 \longrightarrow r_9$ on ne garde pas p_{14}. Il y a donc deux paires à traiter: $F = [r_8, r_9]$

Les éléments de F ne sont pas top réductible par R (dans la fonction REDUCTION) selon la description de l'algorithme 24 mais il est possible de *réduire totalement* r_9 par $y\,t^2 \times r_1$: alors $r_9 = (x^3\,, -x^5\,t^2 + x^2\,y\,t^4)$ et le résultat final est $r_9 = -\varphi'(r_9) = (x^3, x^5\,t^2 - z^2\,t^5)$

Le résultat de REDUCTION est donc: $R_7 = [r_9, r_8]$. Maintenant $R = [r_3, r_2, r_4, r_5, r_1, r_6, r_7, r_8, r_9]$.

Les paires critiques (r_9, r_1), (r_9, r_6), (r_9, r_7), (r_9, r_2), (r_9, r_3), (r_9, r_4), (r_9, r_5), (r_8, r_1), (r_8, r_6), (r_8, r_7), (r_8, r_9), (r_8, r_2) et(r_8, r_5) ne sont pas admissibles.

Les paires critiques sont $p_{18} = (x\,y^5\,t^2, x, r_8, y^2\,t, r_4)$ et $p_{19} = (x^2\,y^5\,t^2, x^2, r_8, y^4\,t^2, r_3)$.

$\boxed{d = 8}$, on calcule $\mathrm{Spol}(P_8)$ avec $P_8 = [p_9, p_{10}, p_{18}]$ et $P = [p_{19}]$.
p_9 est normalisée et $r_{10} = (z^3\,t\,, y^6\,t^2 - x^2\,z^3\,t^3)$ est calculée: on ajoute la règle $z^3\,t \to r_{10}$.
Comme on a la règle $x\,y^2\,t, r_1 \longrightarrow y^2\,t\,r_6$ on ne garde pas p_{10}
Comme on a la règle $x, r_8 \longrightarrow z\,r_9$ on ne garde pas p_{18}
Maintenant $r_{10} = \varphi'(r_{10}) = (z^3\,t\,, y^6\,t^2 - x\,y^2\,z\,t^4)$ est totalement réduit et le résultat du calcul est $R_8 = [r_{10}]$, $R = [r_3, r_2, r_4, r_5, r_1, r_6, r_7, r_8, r_9, r_{10}]$.
Toutes les nouvelles paires de la forme (r_{10}, r_i) avec $i = 1, \ldots, 8$ sont *éliminées* par le critère de F_5.
$\boxed{d = 9}$, on calcule $\mathrm{Spol}(P_9)$ avec $P_9 = [p_{19}]$ et $P = \varnothing$.
Comme on a la règle $x^2\,r_8 \longrightarrow x\,z\,r_9$ on ne garde pas la paire p_{19} et il n'y aucun calcul à effectuer.
$F = \varnothing$ et $R_9 = \varnothing$
L'algorithme se termine et retourne $G = \pi(R)$.

On remarque que le calcul n'a généré aucune paire critique inutile. Avec l'algorithme de Buchberger il y a 7 paires inutiles et 5 utiles.

14 Complexité. Systèmes sur-déterminés

Cette section résume des travaux en collaboration avec M. Bardet et B. Salvy et on se réfère aux articles (Bardet *et al.* , 2005; Bardet *et al.* , 2004; Bardet *et al.* , 2013; Bardet *et al.* , 2014)).

14.1 Introduction et motivation.

Les études de complexité des bases de Gröbner sont nombreuses: par exemple, le pire cas est bien connu depuis l'exemple de Mayr-Meyer et on sait que dans ce cas la complexité du calcul est doublement exponentiel. Heureusement, en pratique, le comportement des bases de Gröbner n'est jamais aussi catastrophique. Le but de cette section est donner des résultats de complexité non pas dans le pire cas mais dans le cas "générique" (plus exactement on va donner une définition précise de semi-régularité pour expliciter la notion de généricité).

On trouve des systèmes surdéterminés dans bon nombre d'applications: par exemple dans les codes correcteurs (décodage des codes cycliques , robotique, conception de filtres et la cryptographie. La sécurité de beaucoup de primitives cryptographiques repose sur la difficulté de la résolution des systèmes algébriques. Dans le contexte de la *cryptographie publique multivariée* les clés publiques sont directement données sous forme de polynômes en plusieurs variables. Plus généralement on peut ramener tout cryptosystème à un système algébrique et en évaluer la robustesse par le biais de l'analyse de complexité du système algébrique: cette démarche porte le nom de Cryptanalyse Algébrique. Dans la plupart des applications dans les corps finis on recherche les solutions non pas dans une extension algébrique du corps de base mais dans le corps lui même: ainsi sur $\mathbb{F}_2$ si on cherche à résoudre un système algébrique $[f_1, \ldots, f_m]$ ayant $m \geqslant n$ équations et n variables on doit lui adjoindre des équations de corps $x_i^2 - x_i = x_i\,(x_i - 1) = 0$ (pour $i = 1, \ldots, n$). On a donc en fait un système avec $m + n \geqslant 2\,n$ équations, donc un système surdéterminé.

Dans le contexte de la cryptographie publique multivariée il est très important de pouvoir distinguer un système "aléatoire" avec un système algébrique provenant d'une instance particulière: on verra que par exemple la clé publique de HFE semble constitué de polynômes aléatoires (par exemple en examinant la densité des équations); pourtant le calcul de la base de Gröbner est beaucoup plus facile que pour un système générique. Un moyen simple est de reporter sur un graphique le paramètre clé permettant l'analyse de complexité, à savoir le degré maximal atteint par le calcul de la base de Gröbner; c'est ce qui a été réalisé sur le dessin de la figure 2: la courbe rouge représente le degré maximal observé expérimentalement pour le calcul d'une base de Gröbner d'un système aléatoire de n équations quadratiques (denses) en n variables sur le corps $\mathbb{F}_2$ (par conséquent il faut ajouter encore n équations de corps de degré 2); les autres courbes proviennent du système HFE de taille n (et le paramètre d est le degré du polynôme secret):

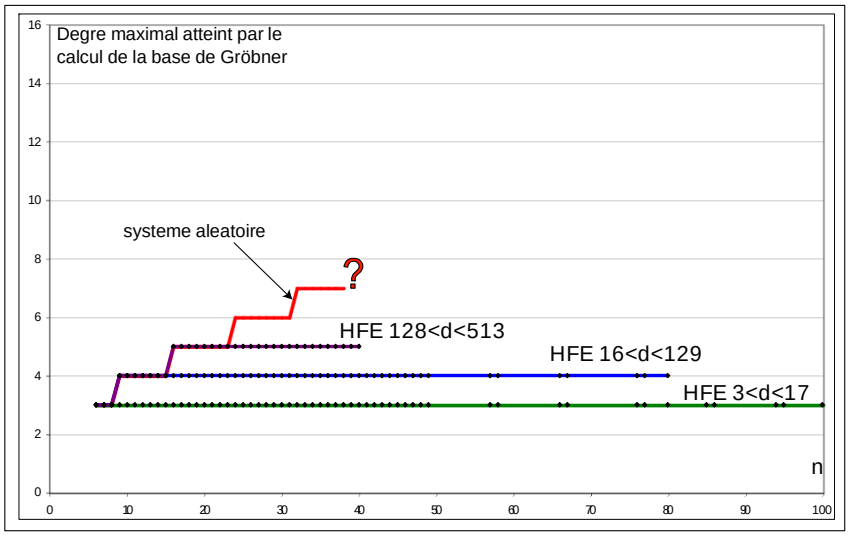


Figure 2: Complexité: degré maximal atteint.

Cependant, on remarque que, très vite, les calculs sont trop difficiles pour les systèmes aléatoires et il n'est pas possible d'obtenir les valeurs pour la courbe rouge même pour des valeurs petites ($n = 40$). Avec la connaissance de la courbe théorique en rouge il sera facile de distinguer un système aléatoire avec un système particulier: on peut stopper le calcul de la base de Gröbner dès qu'on observe une chute de degré strictement inférieure à la valeur prédite par la courbe théorique. De plus on voudrait déterminer la pente de la courbe rouge. Le présent chapitre répond à toutes ces questions: par exemple la pente est de $\frac{1}{11.14}$ (voir le tableau 16.1.2) et toutes les points de la courbe rouge peuvent être calculés (voir la figure 18.6).

Dans la suite la constante $\omega < 2.39$ désigne l'exposant de la complexité des matrices de multiplication.

14.2 Suites régulières et semi-régulières

D'après la définition géométrique de suite régulière (définition 19), dès que $m > n$, f_m devient un diviseur de zéro. Ainsi, si on calcule une base de Gröbner de l'idéal en utilisant une stratégie Normale (algorithme 7), à partir d'un certain degré on aura des réductions à zéro. Pour étendre la notion de suites régulières à des suites surdéterminées, il est donc nécessaire de modifier la définition de suite régulière: ainsi la définition 40 impose que f_i ne soit pas un diviseur de zéro lorsque $\deg(f_i)$ est inférieur au degré de régularité de l'idéal (définition 39).

14.2.1 Généricité

Les suites régulières représentent bien le comportement d'une suite dont les coefficients ont été tirés au hasard. Plus précisément, on peut montrer que, lorsque le corps des coefficients est infini, "presque toute" suite est une suite régulière, où plus rigoureusement que "être une suite régulière1' est une propriété générique au sens de la définition suivante :

Définition 38. *Considérons l'ensemble $E(n, m, d_1, ..., d_m)$ des suites $f_1, \ldots, f_m$ de m polynômes de $\mathbb{K}[x_1, \ldots, x_n]$ en n variables, de degrés $d_1, ..., d_m$. Lorsque le corps $\mathbb{K}$ est infini, on dit qu'une propriété des suites est générique si l'ensemble des suites vérifiant cette propriété est un ouvert non vide de Zariski, c'est à dire si elle est vérifiée par toutes les suites de E, sauf un ensemble algébrique de codimension au moins un.*

Montrer que la propriété de semi-régularité est générique est une conjecture difficile de Fröberg (Fröberg, 1997) démontrée seulement dans le cas $m = n + 1$ (la difficulté est de montrer que c'est un ensemble non vide).

14.2.2 Suites régulières et semi-régulières. Degré de régularité.

Afin d'unifier le cas général et le cas particulier $\mathbb{F}_2$ on utilise la même notation que dans le chapitre 11: $\delta_{\mathbb{K},\mathbb{F}_2}$ est le symbole de Kronecker qui est égal à 1 si $\mathbb{K} = \mathbb{F}_2$ et 0 sinon.

Si $\mathbb{K}$ est le corps $\mathbb{F}_2$ et si on cherche les solutions du système algébrique $(f_1, \ldots, f_m)$ il faut rajouter à l'idéal I engendré par $(f_1, \ldots, f_m)$ les équations de corps $x_i^2 - x_i$. Il faut alors remarquer que dans l'anneau quotient $\mathbb{F}_2[\overline{x_1}, \ldots, \overline{x_n}] = \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$, tout polynôme f de l'idéal $\langle f_1, \ldots, f_m\rangle$ est solution de l'équation triviale

$$f^2 = f$$

(dit autrement f est laissé invariant par le morphisme Frobenius $p \to p^2$).

On note $R_{\mathbb{K}}$ l'anneau de polynômes

$$R_{\mathbb{K}} = \mathbb{K}[x_1, \ldots, x_n] \text{ si } \mathbb{K} \neq \mathbb{F}_2$$

et

$$R_{\mathbb{F}_2} = \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2, \ldots, x_n^2\rangle \text{ si } \mathbb{K} = \mathbb{F}_2$$

(il s'agit des polynômes *sans carré).* Ainsi si $M_d(n)$ désigne le nombre de termes en n variables en degré d dans $R_{\mathbb{K}}$ il est facile de prouver que $M_d(n) = \binom{n+d-1}{d}$ et $M_d(n) = \binom{n}{d}$ si $\mathbb{K} = \mathbb{F}_2$. Par conséquent:

$$\sum_{d=0}^{\infty} M_d(n) z^d = \left(\frac{1 - \delta_{\mathbb{K},\mathbb{F}_2}\, z^2}{1 - z}\right)^n \tag{13}$$

Comme il n'existe pas de suite régulière lorsque le nombre de polynômes est plus grand que le nombre de variables, on doit modifier la définition usuelle de régularité en limitant le degré des non diviseurs de zéro (Bardet, 2004; Bardet *et al.* , 2004):

Définition 39. *Le degré de régularité d'un idéal homogène $I = \langle f_1, \ldots, f_m \rangle$ dans l'anneau $R_{\mathbb{K}}$ est*

$$d_{reg} = \min \{d \geqslant 0 \mid \dim_{\mathbb{K}}(\{f \in I, \ \deg(f) = d\}) = M_d(n)\}$$

Cette définition implique en particulier qu'un calcul de base de Gröbner pour un ordre du degré ne dépasse jamais le degré d_{reg}.

Définition 40. *Une suite de polynômes homogènes $(f_1, \ldots, f_m)$ est semi-régulière si pour tout $i = 1, \ldots, m$ et g tel que*

$$g \cdot f_i \in \langle f_1, \ldots, f_{i-1} \rangle \text{ et } \deg(g \cdot f_i) < d_{reg}$$

alors g est aussi dans $\langle f_1, \ldots, f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}} \rangle$.

Une suite de polynômes affines $(f_1, \ldots, f_m)$ est semi-régulière si $(f_1^H, \ldots, f_m^H)$ est semi-régulière ,où f_i^H est la partie homogène de plus haut degré.

Remarque 24. *La notion de généricité est également plus problématique dans $\mathbb{F}_2$ est: l'ensemble des suites de polynômes étant en effet fini, on peut seulement estimer la probabilité qu'une suite de polynômes soit semi-régulière. Pour des systèmes à coefficients dans $\mathbb{F}_2$, l'ensemble E de la définition 38 est de dimension zéro, et donc cette définition n'a plus de sens. Nous conjecturons cependant qu'une suite "tirée au hasard" sera semi-régulière sur $\mathbb{F}_2$, dans le sens ou la proportion de suites semi-régulières tend vers 1 quand n tend vers l'infini. Expérimentalement cette conjecture est parfaitement vérifiée pour les petites valeurs de n.*

14.3 Définition alternative de semi-régularité

Dans (Pardue & Richert, 2009) une définition légèrement différente de semi-régularité est donnée:

Définition 41. *(Pardue & Richert, 2009) Une suite de polynômes $(f_1, \ldots, f_m)$ de $\mathbb{K}[x_1, \ldots, x_n]^m$ est dites semi-régulière si pour tout $i = 1, \ldots, m$, la matrice de multiplication:*

$$(\mathbb{K}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_{i-1} \rangle)_{d-\deg(f_i)} \xrightarrow{f_i} (\mathbb{K}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_{i-1} \rangle)_d$$

est de rang plein pour tout d.

La définition de semi-régularité 40 est plus générale que la définition de semi-régularité au sens de la définition 41: en effet cette dernière définition implique que toute sous-suite $f_1, \ldots, f_i$ est encore semi-régulière ce qui n'est pas vrai avec notre définition (il suffit de considérer l'exemple $(f_1 = x_1^2, f_2 = x_1 x_2, f_3 = x_2^2)$). Par suite la propriété 54 du théorème 54 n'est plus vraie pour les suites semi-régulières au sens de Pardue-Richert, mais les résultats de complexité s'appliquent.

15 Complexité de l'algorithme F_5

Dans cette section on donne des résultats concernant la complexité de l'algorithme F_5: d'une part des résultats généraux sur le degré atteint par l'algorithme F_5 pour les suites régulières ou semi-régulières: ceci donne un moyen explicite de calculer d_{reg}. C'est l'objet de la présente section. Dans la section 18 on donne des résultats beaucoup plus précis permettant d'estimer le nombre d'opérations arithmétiques lorsque le système algébrique vérifie des hypothèse supplémentaires (position de Noether simultané). Ces derniers résultats ont été obtenus en collaboration avec M. Bardet and B. Salvy (Bardet *et al.* , 2014).

15.1 Degré maximal atteint par l'algorithme F_5

Le but de section est de prédire le degré maximal atteint par l'algorithme F_5 et d'en déduire une estimation de complexité.

Ceci est résumé dans le théorème suivant:

Théorème 53. *Pour une suite semi-régulière $(f_1, \ldots, f_m)$, il n'y a pas de réduction à 0 dans l'algorithme F_5 en degré inférieur à son degré de régularité d_{reg}; de plus d_{reg} est le degré en z du premier coefficient négatif de la série:*

$$\prod_{i=1}^{m} \left(\frac{1 - (1 - \delta_{\mathbb{K},\mathbb{F}_2})\, z^{d_i}}{1 + \delta_{\mathbb{K},\mathbb{F}_2} z^{d_i}} \right) \left(\frac{1 - \delta_{\mathbb{K},\mathbb{F}_2}\, z^2}{1 - z} \right)^n$$

où d_i est le degré total de f_i. Par conséquent, le nombre total d'opérations arithmétiques dans $\mathbb{K}$ nécessaire à F_5 (voir algorithme 21) est borné par

$$Cste \cdot M_{d_{reg}}(n)^{\omega}.$$

Preuve: On découpe la preuve en deux parties: d'abord le calcul explicite grâce une récurrence des tailles des matrices puis le calcule des séries génératrices permettant le calcul explicite du degré de régularité d_{reg}.

15.2 Récurrence sur la taille des matrices. Degré maximal

L'idée est on qu'on suit pas à pas l'algorithme F_5 et on va calculer la taille des matrices dans l'algorithme 21 qui sont de la forme suivante en degré d:

$$M^{(d)}([f_1, \ldots, f_i]) = \begin{array}{c|c|c} & \text{termes de degré } d \text{ en } x_1, \ldots, x_n \\ t_1\, f_1 & \cdots & \cdots \\ t_3\, f_2 & \cdots & \cdots \\ t_3\, f_3 & \cdots & \cdots \end{array}$$

où les t_i sont des termes de degré $d - \deg(f_i)$. Comme ces matrices sont de rang maximal, le degré où l'algorithme F_5 termine est précisément le degré d où le nombre de lignes dans la matrice A_d est *supérieur* au nombre de colonnes. Estimer le nombre de colonnes est facile car c'est exactement le nombre de termes en $x_1, \ldots, x_n$ en degré d; pour estimer le nombre de lignes il suffit d'observer comment on construit les matrices dans F_5 (voir algorithme 21):

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

Le critère F_5 implique que $t x_j f_i$ figure dans cette matrice si $t x_j \notin \mathrm{Id}(\mathrm{LT}(G_{j-1+\delta_{\mathbb{K},\mathbb{F}_2}}))$, où G_j est la base de Gröbner de $[f_1, \ldots, f_j]$. Par conséquent si on note $U_{d,i}(n)$ le nombre de lignes de la matrice $M^{(d)}([f_1, \ldots, f_i])$ on a la relation de récurrence pour $d \geqslant 2$:

$$U_{d,i}(n) = i \cdot \underbrace{M_{d-d_i}(n)}_{\substack{\text{nombre de termes} \\ \text{de degré } d - d_i}} - \underbrace{\sum_{j=1}^{i-1+\delta_{\mathbb{K},\mathbb{F}_2}} U_{d-d_i,j}(n)}_{\text{critères } F_5} \tag{14}$$

On note $h_{d,i}(n)$ la différence entre le nombre de lignes et le nombre des colonnes de $M^{(d)}([f_1, \ldots, f_i])$:

$$h_{d,i}(n) = M_d(n) - U_{d,i}(n)$$

La condition d'arrêt est donc équivalente à trouver le plus petit d tel que $h_{d,m}(n) < 0$. Cependant comme $h_{d,m}(n)$ est un polynôme en n il est plus simple de fixer d et de calculer N_d la plus grande racine réelle des polynômes $h_{d,i}(n)$. Par exemple dans le cas d'équations quadratiques, $m = n$ sur $\mathbb{F}_2$: en utilisant la relation de récurrence (14) on peut calculer explicitement:

$U_{0,i}(n) = 0$
$U_{1,i}(n) = 0$
$U_{2,i}(n) = i\binom{n}{0} - 0 = i$
$U_{3,i}(n) = i\binom{n}{1} - \sum_{j=1}^{i} U_{1,j}(n) = i\,n$
$U_{4,i}(n) = i\binom{n}{2} - \sum_{j=1}^{i} U_{2,j}(n) = i\,\frac{n\,(n-1)}{2} - \sum_{j=1}^{i} j = \frac{i\left(n^2-n-i-1\right)}{2}$

Puis:

$$\begin{aligned} h_{3,n}(n) &= M_3(n) - U_{3,n}(n) \\ &= \binom{n}{3} - n^2 \\ &= \frac{n\left(n^2-9\,n+2\right)}{6} \end{aligned}$$

Il suffit ensuite de calculer la plus grande racine réelle de ce polynôme (utiliser l'algorithme d'isolation des racines réelles **??**):

$$h_{3,n}(n) = n\left(n - 9/2 - 1/2\sqrt{73}\right)\left(n - 9/2 + 1/2\sqrt{73}\right)$$

la plus grande racine est dans ce cas: $9/2 + 1/2\sqrt{73} \approx 8.772$ et donc $N_3 = \mathbf{9}$. En poursuivant les calculs on trouve:

d	3	4	5	6	7	8	9
N_d	9	16	24	32	41	49	58

Table 1: Degré maximal sur F_2

Pour lire ce tableau il faut partir de la ligne inférieure:

(i) Lorsque le nombre de variables n est inférieur à $N_3 = 9$ alors le degré maximal atteint par l'algorithme F_5 est 3; par conséquent la taille de la plus grande matrice est $n^3 \times n^3$ et donc une complexité arithmétique globale en $O(n^9)$.

(ii) Pour $N_3 = 9 \leqslant n < N_4 = 16$ le degré maximal est 4 et donc une complexité de $O(n^{12})$.

(iii) Pour $N_4 = 16 \leqslant n < N_5 = 24$ le degré maximal est 5 et donc une complexité de $O(n^{15})$.

De cette façon on construit la courbe théorique (rouge) de la figure 18.6.

15.3 Série génératrice.

Pour obtenir un calcul plus explicite du degré maximal on va calculer la série génératrice:

$$H_m(z) = \sum_{d=0}^{\infty} h_{d,m}\,(n) z^d$$

En soustrayant l' équation (14) on trouve:

$$U_{d,m}(n) - U_{d,m-1}(n) = M_{d-d_m}(n) - U_{d-d_m,m-1+\delta_{\mathbb{K},\mathbb{F}_2}}(n)$$

Comme $h_{d,m}(n) = M_d(n) - U_{d,m}(n)$ on a immédiatement

$$h_{d,m}(n) - h_{d,m-1}(n) = -h_{d-d_m,m-1+\delta_{\mathbb{K},\mathbb{F}_2}}(n) \tag{15}$$

d'où en passant aux séries (on note δ pour $\delta_{\mathbb{K},\mathbb{F}_2}$dans cette preuve):

$$\begin{aligned} H_m(z) &= \sum_{d=0}^{\infty} h_{d,m-1}(n) z^d - z^{d_m} \sum_{d=0}^{\infty} h_{d,m-1+\delta}(n) z^d \\ &= \sum_{d=0}^{\infty} h_{d,m-1}(n) z^d - \delta\, z^{d_m} \sum_{d=0}^{\infty} h_{d,m}(n) z^d - (1-\delta)\, z^{d_m} \sum_{d=0}^{\infty} h_{d,m-1}(n) z^d \end{aligned}$$

et donc en groupant les termes:

$$\left(1+\delta\, z^{d_m}\right) H_m(z) = \left(1-(1-\delta)\, z^{d_m}\right) \sum_{d=0}^{\infty} h_{d,m-1}(n) z^d$$

et par suite:

$$\begin{aligned} H_m(z) &= \frac{1-(1-\delta)\, z^{d_m}}{1+\delta\, z^{d_m}} \sum_{d=0}^{\infty} h_{d,m-1}(n) z^d \\ &= \cdots \\ &= \prod_{i=1}^{m} \left(\frac{1-(1-\delta)\, z^{d_i}}{1+\delta\, z^{d_i}}\right) \sum_{d=0}^{\infty} h_{d,0}(n) z^d \\ &= \prod_{i=1}^{m} \left(\frac{1-(1-\delta)\, z^{d_i}}{1+\delta\, z^{d_i}}\right) \sum_{d=0}^{\infty} M_d(n) z^d \\ &= \prod_{i=1}^{m} \left(\frac{1-(1-\delta)\, z^{d_i}}{1+\delta\, z^{d_i}}\right) \left(\frac{1-\delta\, z^2}{1-z}\right)^n \end{aligned}$$

Pour calculer d_{reg} il suffit donc de développer cette série et de calculer le premier indice dont le coefficient est < 0. Ceci termine la preuve du théorème.

Corollaire 8. *Pour des équations quadratiques les séries sont:*

$H_m(z) = \frac{(1-z^2)^m}{(1-z)^n}$ *pour un corps quelconque* $\mathbb{K}$

$H_m(z) = \frac{(1+z)^n}{(1+z^2)^m}$ *pour le corps* $\mathbb{F}_2$

15.4 Exemple: NTRU

Exemple d'application des formule: NTRU.

Le problème fondamental dans cryptosystème NTRU (Hoffstein *et al.* , 1998) est le suivant: étant donné un polynôme $H(X)$ de degré n à coefficients dans $\mathbb{Z}_{2^q}$ on cherche $N(X), D(X)$ des polynômes de degré n à coefficients dans $\{0,1\}$ tels que:

$$H(X) = \frac{N(X)}{D(X)} \bmod (X^n - 1) \bmod q.$$

Il est clair que ce problème revient à résoudre un système *linéaire sous-determiné*:

$$\begin{cases} l_1(x_1, \ldots, x_n) = y_1 \\ \cdots \\ l_n(x_1, \ldots, x_n) = y_n \end{cases}$$

où l_i une équation linéaire à coefficients dans l'anneau 2^k. On cherche une solution $(x_1, \ldots, x_n, y_1, \ldots, y_n)$ dans $\{0,1\}^{2\,n}$. En utilisant les deux premiers bits des vecteurs de Witt, Smart Vercauteren et Silverman (Smart *et al.* , 2005) montrent que la recherche de la solution se ramène à la résolution de n équations quadratiques n variables dans $\mathbb{F}_2$. Dans (Bourgeois, 2006) l'auteur propose d'utiliser les 3ème et 4ème bit des vecteurs de Witt pour améliorer l'attaque précédente: on obtient ainsi n équations supplémentaires de degré 4 (resp. 8) pour le 3ème bit (resp. 4ème bit). Peut on évaluer le gain en complexité pour des valeurs cryptographiquement réalistes ($n = 251$ ou $n = 503$) ?

Il suffit d'appliquer la méthode du théorème 53 (avec $\delta_{\mathbb{K},\mathbb{F}_2} = 1$): on calcule les séries suivantes et on cherche le premier coefficient négatif:

1 er bit: $(1+z)^n\,(1+z^2)^{-n} = 1 + n\,z + \frac{n\,(n-3)}{2} z^2 + \cdots$
2 ème bit: $(1+z)^n\,(1+z^2)^{-n}\,(1+z^4)^{-n}$
3 ème bit: $(1+z)^n\,(1+z^2)^{-n}\,(1+z^4)^{-n}\,(1+z^8)^{-n}$

Dans le tableau suivant on reporte la valeur de d_{reg} ainsi obtenue:

n	1 bit	2bit	3bit
251	29	28	28
503	53	52	52

16 Caractérisation des suites semi-régulières.

Les propriétés des suites semi-régulières sont résumées dans le théorème 54; on aura besoin de la notation:

Définition 42. *Pour une série* $\sum_{i=0}^{\infty} a_i z^i$, *la notation* $\left[\sum_{i=0}^{\infty} a_i z^i\right]^+$ *désigne la série* $\sum_{i=0}^{\infty} b_i z^i$ *avec* $b_i = \begin{cases} a_i \text{ si } a_j > 0,\ \forall 0 \leqslant j \leqslant i \\ 0 \text{ sinon} \end{cases}$

Proposition 54. *Soit* $(f_1, \ldots, f_m)$ *une suite de* m *polynômes en* n *variables,* $d_i = \deg(f_i)$ *. Alors:*

(i) La suite $(f_1, \ldots, f_m) \subset R_{\mathbb{K}}$ *est semi-régulière si et seulement si la série de Hilbert de la suite* $(f_1^H, \ldots, f_m^H)$ *est donné par :*

$$[H_m(z)]^+$$

où $H_m(z) = \prod_{i=1}^{m} \left(\frac{1-(1-\delta_{\mathbb{K},\mathbb{F}_2})\,z^{d_i}}{1+\delta_{\mathbb{K},\mathbb{F}_2} z^{d_i}}\right) \left(\frac{1-\delta_{\mathbb{K},\mathbb{F}_2}\,z^2}{1-z}\right)^n$.

(ii) Pour un corps $\mathbb{K}$ *quelconque et* $m \leqslant n$, *la suite* $(f_1, \ldots, f_m)$ *est régulière si et seulement si elle est semi-régulière: dans ce cas les deux notions sont les mêmes.*

(iii) Le degré de régularité de l'idéal engendré par $(f_1, \ldots, f_m)$ est l'indice du premier coefficient négatif de la série $H_m(z)$.

Proof. ((Bardet *et al.* , 2005; Bardet, 2004)) On démontre (i) pour des polynômes homogènes. Considérons la suite exacte:

$$\begin{aligned} 0 &\to \left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}\rangle\right)_{d-d_i} \\ &\xrightarrow{f_i} \left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}\rangle\right)_{d} \\ &\to \left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i}\rangle\right)_{d} \\ &\to 0 \end{aligned}$$

alors tant que $d < d_{\text{reg}}$ la fonction de Hilbert correspondante vérifie (Cox *et al.* , 1998):

$$\mathrm{HF}_{\langle f_1,\ldots,f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}\rangle}(d-d_i) - \mathrm{HF}_{\langle f_1,\ldots,f_{i-1}\rangle}(d) + \mathrm{HF}_{\langle f_1,\ldots,f_{i}\rangle}(d) = 0 \tag{16}$$

pour tout $d < d_{\text{reg}}$. De plus, $\mathrm{HF}_{\langle f_1,\ldots,f_i\rangle}(d) = 0$ pour tout i,d; comme $\mathrm{HF}_{\langle 0\rangle}(d) = M_d(n)$ on retrouve la même relation de récurrence que l'équation (15). Par conséquent $\mathrm{HS}_{\langle f_1,\ldots,f_m\rangle}(z) = H_m$.

Réciproquement, considérons la suite exacte

$$\begin{aligned} 0 &\to K_{d-d_i} \\ &\left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}\rangle\right)_{d-d_i} \\ &\xrightarrow{f_i} \left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i-1+\delta_{\mathbb{K},\mathbb{F}_2}}\rangle\right)_{d} \\ &\to \left(k[x_1,\ldots,x_n]/\langle f_1,\ldots,f_{i}\rangle\right)_{d} \\ &\to 0 \end{aligned}$$

où K est le noyau de la fonction de multiplication par f_i, alors pour tout $d < d_{\text{reg}}$ le noyau $K_{d-d_i} = \{0\}$, et donc d'après la définition 40 la suite est semi-régulière.

La propriété (ii) découle de (i) et du théorème 35 page 20. La propriété (iii) est une conséquence de la définition 39. □

16.1 Développements asymptotiques du degré de régularité d_{reg}

Dans cette section nous présentons une analyse asymptotique du degré de régularité d'un idéal de dimension zéro défini par une suite semi-régulière sur un corps $\mathbb{K}$ ou sur $\mathbb{F}_2$.

16.1.1 Méthode du col.

Des méthodes de points cols et de points cols coalescents sont utilisées pour calculer un développement asymptotique du degré de régularité d_{reg} lorsque le nombre de variables n tend vers l'infini. Nous donnons de nombreux résultats explicites de ces développements, dont les premiers termes fournissent une approximation quasi exacte du degré de régularité et ceci même pour des petites valeurs de n. On cherche dans la série suivante le premier indice d pour lequel le coefficient de degré d est négatif.

$$H_m(z) = \prod_{i=1}^{m} \left(\frac{1-(1-\delta)\,z^{d_i}}{1+\delta z^{d_i}}\right) \left(\frac{1-\delta\,z^2}{1-z}\right)^n$$

Pour cela la méthode se résume à:

- écrire le d-th coefficient de la série en utilisant la formule de Cauchy:

$$\mathcal{I}_n(d) = s_{d,m}(n) = \frac{1}{2\imath\pi}\oint H_m(z)\frac{dz}{z^{d+1}} = \frac{1}{2\imath\pi}\oint e^{n\,f(z)}dz \tag{17}$$

  où le chemin d'intégration est un lacet simple, entourant l'origine et aucune autre pôle de $H_m(z)$.

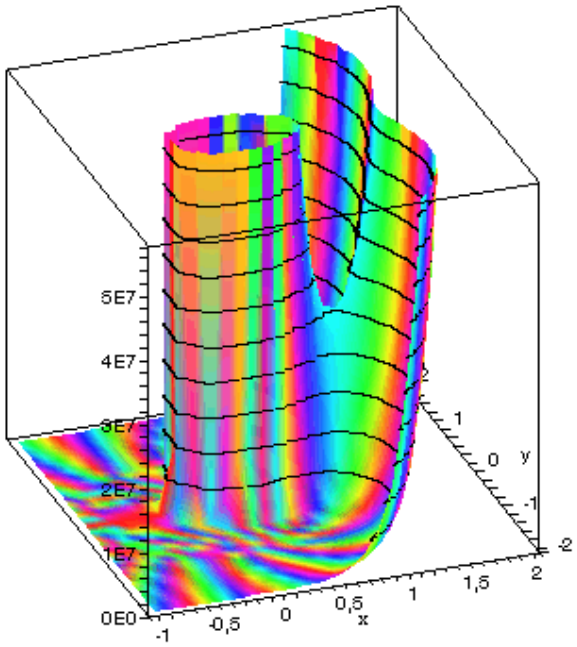


Fig 16.1.1: un point col simple.

- calculer le terme dominant de $\mathcal{I}_n(d)$ en fonction de d et n lorsque $n \to \infty$, d étant considéré comme un paramètre.
- trouver la valeur de d qui annule ce terme dominant: ceci nous donne le premier terme dans le développement asymptotique de d_{reg} .

En itérant ce processus on peut calculer les termes suivants dans le développement de d_{reg}.

Le développement asymptotique de $\mathcal{I}_n(d)$ est calculé en utilisant la méthode du col ou des points cols coalescents. L'idée de ces méthodes est de faire passer le chemin d'intégration par les points cols (les zéros de $f'(z)$ voir la figure 16.1.1) de la fonction à intégrer. On montre alors que la contribution des parties du chemin qui ne sont pas voisines des cols est asymptotiquement négligeable, et qu'au voisinage de ces cols la fonction à intégrer peut être approchée par une fonction gaussienne (pour la méthode des cols) ou une fonction d'Airy (pour la méthode des points cols coalescents) (Chester *et al.* , 1957).

16.1.2 Classification

On considère une suite semi-régulière constituée d'équations $(f_1, \ldots, f_m)$. Le tableau suivant résume le résultat de plusieurs théorèmes donne le développement asymptotique de d_{reg} lorsque $n \to \infty$ en fonction de la valeur du rapport entre le nombre d'équations et le nombre de variables $\frac{m}{n}$.

Légende des symboles utilisés dans le tableau:

k est une constante (qui ne dépend pas de n).

d_i est le degré total de f_i.

$H_k(X)$ est le k ème polynôme d'Hermite; $h_{k,1}$ est le plus grand zéro de H_k (tous les zéros de $H_k(X)$ sont réels).

$a_1 \approx -2.3381$ est le plus grand zéro de la fonction d'Airy (solution de $\frac{\partial^2 y}{\partial z^2} - z\, y = 0$).

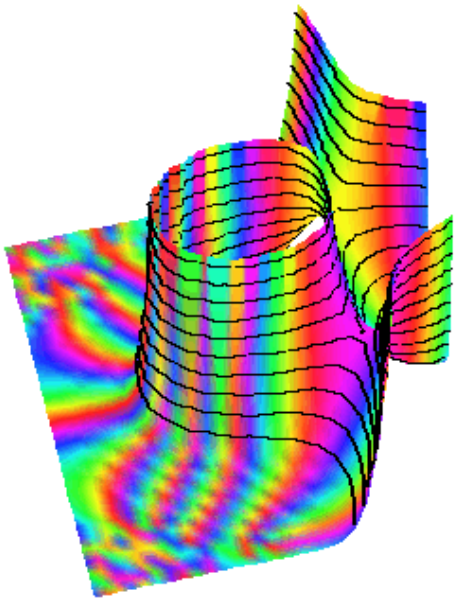
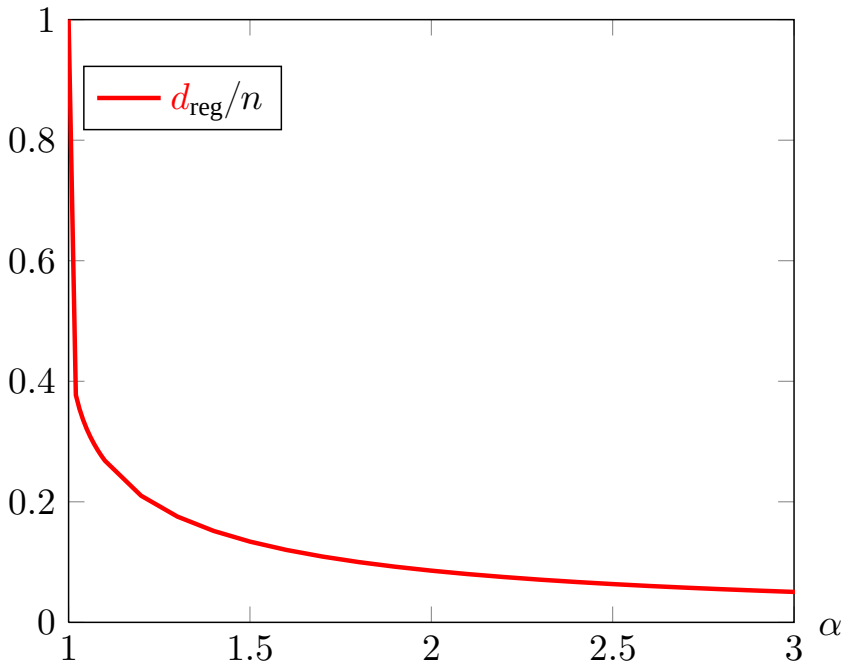


Fig 16.1.1: Points cols coalescents.

$\Phi(z) = \frac{z}{n}\frac{\partial}{\partial z}\log\left((1-z)^n\prod_{i=1}^{m}(1-z^{d_i})^{-1}\right) = \frac{z}{1-z} - \frac{1}{n}\sum_{i=1}^{m}\frac{d_i z^{d_i}}{1-z^{d_i}}$ et z_0 est la racine de $\Phi'(z)$ qui minimise $\Phi(z_0) > 0$.



Degré de régularité de $m = \alpha\, n$ équations quadratiques en n variables

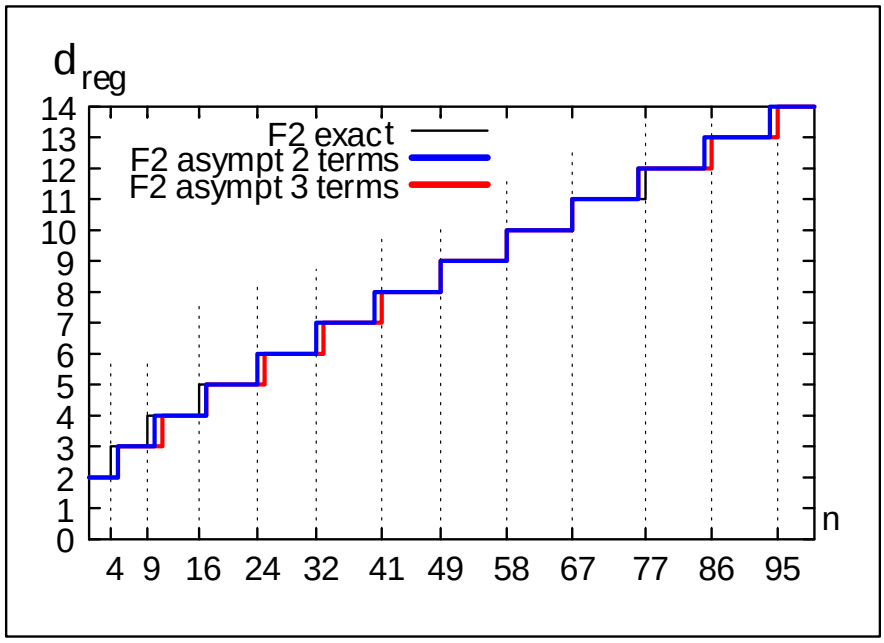


Figure 3: précision des développements asymptotiques.

Remarque 25. *Dans le cas où $m = n + 1$ on a $h_{1,1} = 0$ et donc $d_{reg} = \frac{m}{2} + o(1)$ ce qui est en accord avec le résultat de Szanto (Szanto, 2004).*

m	Degré	d_{reg}
$m < n$	$\mathbb{K}, d_i = 2$	$m + 1$ (Borne de Macaulay)
$n + 1$	$\mathbb{K}$	$\sum_{i=1}^{n+1} \frac{d_i - 1}{2}$ (A. Szanto)
$n + k$	$\mathbb{K}, d_i = 2$	$\frac{m}{2} - h_{k,1}\sqrt{\frac{m}{2}} + o(1)$
$n + k$	$\mathbb{K}$	$\sum_{i=1}^{n+k} \frac{d_i - 1}{2} - h_{k,1}\sqrt{\sum_{i=1}^{n+k} \frac{d_i^2 - 1}{6}} + o(1)$
$2\,n$	$\mathbb{K}, d_i = 2$	$\frac{n}{11.6569} + 1.04\,n^{\frac{1}{3}} - 1.47 + 1.71\,n^{-\frac{1}{3}} + O\left(n^{-\frac{2}{3}}\right)$
$k\ n$	$\mathbb{K}, d_i = 2$	$(k - \frac{1}{2} - \sqrt{k(k-1)})n + \frac{-a_1}{2(k(k-1))^{\frac{1}{6}}} n^{\frac{1}{3}} + O(1)$
$k\ n$	$\mathbb{K}$	$\Phi(z_0)\,n - a_1\left(-\frac{1}{2}\Phi''(z_0)z_0^2\right)^{\frac{1}{3}} + O\left(n^{\frac{1}{3}}\right)$
n	$\mathbb{F}_2, d_i = 2$	$\frac{n}{11.1360} + 1.0034\,n^{\frac{1}{3}} - 1.58 + O(n^{-\frac{1}{3}})$
$k\ n$	$\mathbb{F}_2, d_i = 2$	$\left(-k + \frac{1}{2} + \frac{1}{2}\sqrt{2k(k-5) - 1 + 2(k+2)\sqrt{k(k+2)}}\right) n$

Table 2: Généralisations de la borne de Macaulay

Remarque 26. *Afin d'illustrer la précision de ces développement asymptotique on trace sur même courbe (figure 3 p. 65), les points obtenus lors de la preuve du théorème 53 dans le cas de $m = n$ équations quadratiques sur $\mathbb{F}_2$, les courbes obtenus en partant du tableau en considérant deux ou trois termes.*

Le dessin montre que les courbes ne sont pas discernables et donc que les développements asymptotiques sont suffisants avec de petites valeurs de n.

17 Degré de régularité: autre cas et systèmes structurés

On considère maintenant des cas où le degré des équations varie aussi avec n, le nombre de variables.

Proposition 55. *Soit $[f_1, \ldots, f_m]$ un système semi-régulier dans $\mathbb{K}[x_1, \ldots, x_n]$. On suppose qu'il y a $m = \beta^n$ équations que le degré des équation est de la forme$\deg(f_i) = \alpha n$ alors*

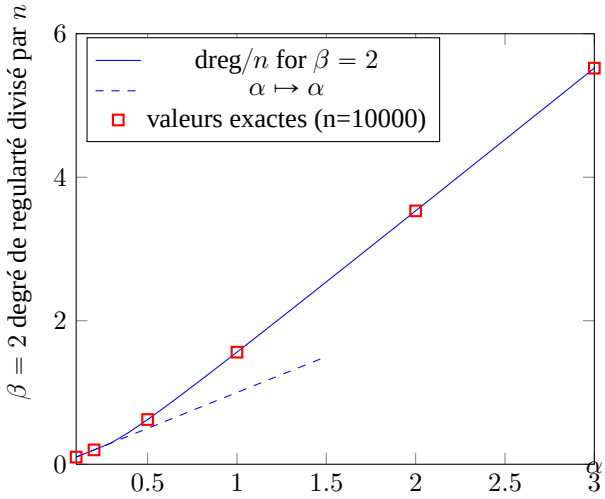
$$\frac{dreg}{n} \approx \begin{cases} \alpha \text{ if } \alpha < \alpha_0 \\ f(\alpha) \text{ if } \alpha_0 \leqslant \alpha < 6 \end{cases}$$

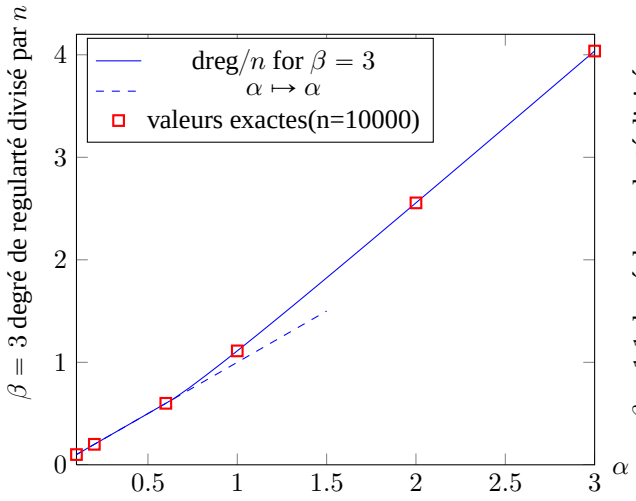
où $f(\alpha)$ est la racine réelle l'équation:

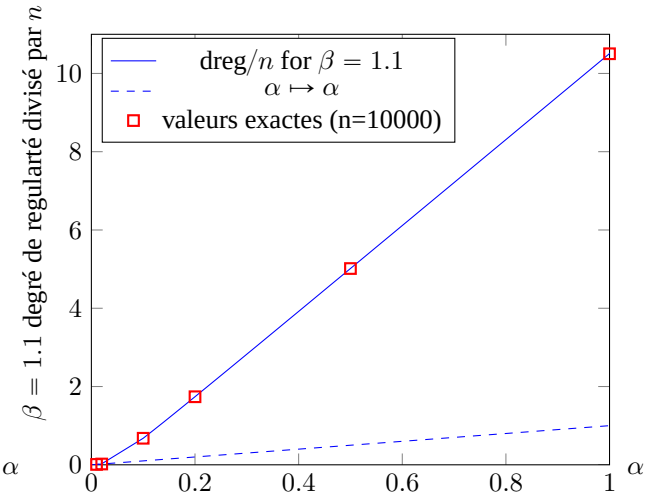
$$\ln(1+l) - l\ln(l) + l\ln(1+l) - \ln(1+l-\alpha) + \ln(l-\alpha)l - \ln(l-\alpha)\alpha - \ln(1+l-\alpha)l + \ln(1+l-\alpha)\alpha = \ln(\beta)$$

et α_0 est le nombre réel tel que $f(\alpha_0) = \alpha_0$. Approximativement:

β	α_0
2	0.293815373
3	0.641794121
1.1	0.019208159







Proposition 56. *Soit $[f_1, \ldots, f_m]$ un système semi-régulier dans $\mathbb{K}[x_1, \ldots, x_n]$. On suppose que le nombre d'équations est $m = n^{n(\beta-1)}$ et que le degré de chaque équation est $\deg(f_i) = n^\beta$ avec $\beta > 1$. Alors*

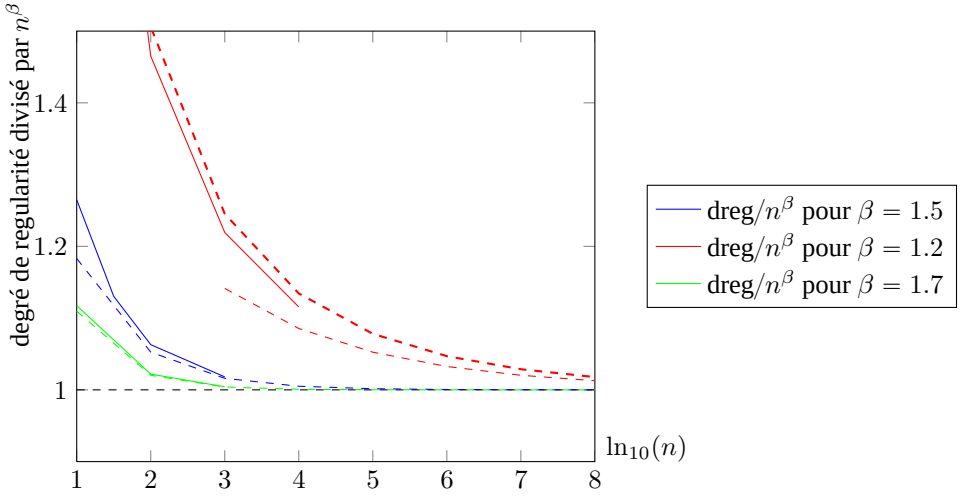
$$\textit{dreg} \approx C_1(n)\, n^\beta.$$

où

$$C_1(n) = 1 + a_\beta \frac{1}{n^{\beta-1}} + b_\beta \frac{1}{n^{2\beta-2}} + \cdots$$

avec

β	a_β
1.1	0.99998675
1.2	0.65265401
1.5	0.54537396
1.7	0.54252680
2	0.54222139



17.1 Systèmes multi-homogènes

Les résultats suivants sont issus des articles (Faugère *et al.* , 2011) et (Faugère *et al.* , 2014a).

Nombre de variables	Degré	d_{reg}
$n = n_1 + n_2$	Bilinéaire $(1, n_1)$	$2 + \min(n_1, n_2)$
$n = n_1 + \cdots + n_l$	Multilinéaire $(1, \ldots, 1)$	$2 + \sum n_i - \max(\lceil n_i + 1 \rceil)$
$n = n_1 + \cdots + n_l$	Multi-homogène $(d_1, d_2, \ldots, d_l)$	$2 + \sum n_i - \max(\left\lceil \frac{n_i+1}{d_i} \right\rceil)$

17.2 Systèmes invariant par l'action d'un groupe abélien G

Les résultats suivants sont issus de l'article (Faugère & Svartz, 2013).

Nombre de variables	Type de système	d_{reg}
n	Équations quadratiques / G-Degré $0, 1$	2
n	Équations quadratiques / G-Degré $0, 1, \ldots, k$	$k + 1$
n	Équations cubiques / G-Degré 0	???

18 Amélioration des bornes de complexité de F_5

Même si les résultats obtenus jusqu'à maintenant sont optimaux pour l'estimation du paramètre de complexité d_{reg} l'analyse de la complexité arithmétique reste sommaire: une fois connu d_{reg} on estime de la plus grande matrice apparaissant dans le calcul: c'est une matrice $M_{d_{\text{reg}}}(n) \times M_{d_{\text{reg}}}(n)$ et donc le coût total est estimé à $M_{d_{\text{reg}}}(n)^3$. S'il est vrai que pour un système semi-régulier les matrices sont de tailles croissantes on va voir, et l'expérience le confirme, que ce n'est pas la dernière matrice qui est la plus coûteuse à traiter: cette dernière matrice est en effet quasi-triangulaire et il y a peu de travail à effectuer pour la rendre triangulaire. Les résultats suivants sont repris de (Bardet *et al.* , 2014).

On considère des polynômes *homogènes* $(f_1, \ldots, f_m)$ dans $\mathbb{K}[x_1, \ldots, x_n]$, d_i est le degré total de f_i et on se limite explicitement à l'ordre $<_{\text{DRL}}$. Dans cette étude $m \leqslant n$.

18.1 Position de Noether

Définition 43. *Les variables $(x_1, \ldots, x_m)$ sont en position de Noether par rapport au système $(f_1, \ldots, f_m)$ si dans $\mathbb{K}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_m\rangle$, $\overline{x_i}$ est un entier algébrique sur $\mathbb{K}[x_{m+1}, \ldots, x_n]$ et de plus $\mathbb{K}[x_{m+1}, \ldots, x_n] \cap \langle f_1, \ldots, f_m\rangle = \langle 0\rangle$.*

Géométriquement cela signifie que le système est de dimension $n - m$ et que dans la clôture algébrique de $\mathbb{K}$ le nombre de solution (avec multiplicité) reste invariant quelque soit la spécialisation des variables $(x_{m+1}, \ldots, x_n)$.

La proposition suivante donne une caractérisation commode:

Proposition 57. *((?), lemme 4.1). Les variables $(x_1, \ldots, x_m)$ sont en position de Noether par rapport au système $(f_1, \ldots, f_m)$ si et seulement si pour tout $1 \leqslant j \leqslant m$ il existe $n_j \in \mathbb{N}$ tel que $x_j^{n_j} \in \mathrm{LT}_{<_{DRL}}(\langle f_1, \ldots, f_m\rangle)$ c'est à dire $x_j \in \sqrt{\mathrm{LT}_{<_{DRL}}(\langle f_1, \ldots, f_m\rangle)}$.*

18.2 Position de Noether simultanée.

F_5 étant incrémental il est nécessaire d'avoir une propriété plus forte:

Définition 44. *(SNP) Les variables $(x_1, \ldots, x_m)$ sont en position de Noether simultané par rapport au système $(f_1, \ldots, f_m)$ si $(x_1, \ldots, x_i)$ est en position de Noether par rapport au système $(f_1, \ldots, f_m)$ pour tout $i \in \{1, \ldots, m\}$.*

En utilisant la proposition 57 on obtient la définition suivante:

Définition 45. *(SNP). Soient des polynômes* homogènes *$(f_1, \ldots, f_m)$ dans $\mathbb{K}[x_1, \ldots, x_n]$, et l'ordre DRL $<_{DRL}$. Si G_i est une base de Gröbner de $(f_1, \ldots, f_i)$ pour l'ordre $<_{DRL}$ pour $1 \leqslant i \leqslant m \leqslant n$. On dit que $(f_1, \ldots, f_m)$ est en position de Noether simultanée si*

$$\text{pour tout } i \in \{1, \ldots, m\} \text{ on a } x_i \in \sqrt{LT(G_i)} \text{ et } x_i \notin \sqrt{LT(G_{i-1})}.$$

Remarque 27. *C'est une propriété qui est vraie pour des polynômes aléatoires.*

Proposition 58. *Si $(f_1, \ldots, f_m)$ est en SNP alors $(f_1, \ldots, f_m)$ est une suite régulière.*

Lemme 4. *Si $(f_1, \ldots, f_m)$ est en SNP alors pour tout $1 \leqslant i \leqslant m$,*

$$(f_1, \ldots, f_i, x_{i+1}, \ldots, x_n)$$

est une suite régulière.

Proof. D'après la définition 45, $\sqrt{LT(G_i)} \subset \{x_1, \ldots, x_i\}$ et donc

$$(f_1, \ldots, f_i, x_{i+1}, \ldots, x_n)$$

est un idéal zéro-dimensionnel. □

18.3 Structure d'une base DRL.

Dans la suite on suppose que G_i est une base de Gröbner de $(f_1, \ldots, f_i)$ pour l'ordre $<_{\text{DRL}}$calculé par l'algorithme F_5. Par conséquent on pour chaque $g \in G$ on a aussi la signature $\mathcal{S}(g) = s_g = (i_g, t_g) \in \mathbb{N} \times T$. On rappelle que ceci implique (voir section 11) l'existence d'une écriture:

$$g = (t_g + \cdots)\, f_{i_g} + (\cdots)\, f_{i_g+1} + \cdots$$

Le lemme suivant donne la structure des termes de têtes d'une base de Gröbner pour un ordre DRL:

Lemme 5. *Pour tout polynôme g apparaissant dans le calcul de G_i dont la signature est $s_g = (i, t)$ avec $t \in \mathbb{K}[x_1, \ldots, x_i]$ on a* $\mathrm{LT}(g_i) \in \mathbb{K}[x_1, \ldots, x_i]$.

Proof. Soit k le plus grand j tel $x_j | \mathrm{LT}(g)$;supposons que $k > i$. Alors (d'après algorithme F_5) il existe $(g_1, \ldots, g_i)$ tels que

$$g = \sum_{j=1}^{i} g_j\, f_j \text{ avec } \mathrm{LT}(g_i) = t \in \mathbb{K}[x_1, \ldots, x_i]$$

et g_i est réduit modulo $[f_1, \ldots, f_{i-1}]$. Comme $\mathrm{LT}(g) = 0 \mod [x_k, \ldots, x_n]$ on a (voir la proposition 7) $g = 0 \mod [x_k, \ldots, x_n]$; a fortiori $g = 0 \mod [x_{i+1}, \ldots, x_n]$. De plus t est encore le terme de tête de $g_i \mod [x_{i+1}, \ldots, x_n]$ et donc $g_i \neq 0 \mod [x_{i+1}, \ldots, x_n]$. Ainsi $(x_{i+1}, \ldots, x_n, f_1, \ldots, f_i)$ n'est pas une suite régulière ce qui est contraire au lemme 4. □

Le théorème suivant est plus précis et il donne la structure des signatures des polynômes d'une base de Gröbner:

Théorème 59. *Si le système $(f_1, \ldots, f_m)$ est en SNP alors pour tout $(s_g, g) \in G_i$, $\mathrm{LT}(g) \in \mathbb{K}[x_1, \ldots, x_i]$ et s_g (la signature) est de la forme $s_g = (j, t)$ avec $j \leqslant i$ et t est un terme dans $\mathbb{K}[x_1, \ldots, x_{j-1}]$.*

Proof. Dans la preuve de ce théorème $\mathcal{T}_i$ est l'ensemble des termes en $x_1, \ldots, x_i$ et $\mathcal{T} = \mathcal{T}_n$. On raisonne par l'absurde et on suppose que l'ensemble

$$\mathcal{A} = \{(s_f, f) \in G_i \mid \text{tel que } 1 \leqslant i \leqslant n,\ s_f = (i, t) \text{ avec } t \in \mathcal{T} \backslash \mathcal{T}_{i-1}\}$$

tel que. On prend (s_h, h) le plus petit élément de $\mathcal{A}$ (c'est à dire avec la plus petite signature s_h). Donc $s_h = (i, s)$ avec $s \notin \mathcal{T}_{i-1}$. La seule façon de créer ce polynôme h dans F_5 est d'ajouter dans la matrice une ligne $t \cdot (h_0, s_{h_0})$ où $h_0 \in G_i$, $s_{h_0} = (i, t_0)$ et t est un terme de degré $\geqslant 1$; ainsi $s_h = (i, t \cdot t_0)$. Comme (s_h, h) est le plus petit élément de $\mathcal{A}$ on sait que $t_0 \in \mathcal{T}_{i-1}$ et donc $\mathrm{LT}(h_0) \in \mathcal{T}_i$ (d'après le lemme 5). Comme $t \cdot t_0 = s \notin \mathcal{T}_{i-1}$, on peut trouver un indice $l \geqslant i$ tel que x_l divise t. Maintenant cette ligne (s_h, h) de la matrice a été réduite par une autre ligne: on peut donc trouver $k \leqslant i$, $g \in G_k$ et $w \in \mathcal{T}$ tels que $k \leqslant i$, $s_g = (k, v) < s_h$ et

$$w \cdot \mathrm{LT}(g) = \mathrm{LT}(h) = t \cdot \mathrm{LT}(h_0). \tag{18}$$

À cause de la minimalité de h on a $v \in \mathcal{T}_{k-1}$ et $\mathrm{LT}(g) \in \mathcal{T}_k$. Comme $\frac{t}{\gcd(t,w)} h_0$ est réductible par g et que $t \cdot h_0$ est le plus petit élément de $\mathcal{A}$ on en déduit que $\gcd(t, w) = 1$; par suite t divise $\mathrm{LT}(g) \in \mathcal{T}_k$ et donc $k = i = l$, $t \in \mathcal{T}_i$. De l'équation (18) on déduit immédiatement que $w \in \mathcal{T}_i$; mais si x_i divise w alors $\mathrm{LT}(g)$ divise $\frac{w}{x_i} \mathrm{LT}(g) = \frac{t}{x_i} \mathrm{LT}(h_0)$ et comme $\frac{t}{x_i} h_0$ est strictement inférieur à h dans $\mathcal{A}$ ceci est impossible. Donc, en fait, $w \in \mathcal{T}_{i-1}$ et $s_{w \cdot g} = (i, w \cdot v)$ avec $w \cdot v \in \mathcal{T}_{i-1}$. On obtient ainsi une contradiction car l'indice de la ligne h est $(i, t \cdot t_0)$ avec $x_i = x_l$ divisant t et donc a fortiori $t \cdot t_0$ qui est $<_{\text{DRL}} w \cdot v \in \mathcal{T}_{i-1}$ (c'est une propriété de l'ordre DRL): l'opération élémentaire sur la ligne est donc interdite. □

18.4 Nombre d'élément d'une base de Gröbner DRL.

Le théorème suivant donne une nouvelle borne très précise sur le nombre de polynômes dans la base de Gröbner:

Théorème 60. *Soit $(f_1, \ldots, f_m)$ un système homogène pour lequel les variables $(x_1, \ldots, x_n)$ sont en SNP. Soit G_i la base de Gröbner réduite de $(f_1, \ldots, f_i)$ pour l'ordre DRL et pour $1 \leqslant i \leqslant m$, alors le nombre de polynômes de degré d dans $G_i \backslash G_{i-1}$ est borné par $N_{d,i}$, où*

$$\sum_{d=0}^{\infty} N_{d,i} z^d = z^{d_i} \prod_{j=1}^{i-1} \frac{1 - z^{d_j}}{1 - z} \tag{19}$$

Proof. On fait la preuve par récurrence sur i. Si $i = 1$ alors par définition de la position de Noether, la base est réduite à un seul polynôme dont le terme de tête est $x_1^{d_1}$ et donc l'équation (19) est correcte. Supposons la propriété vraie pour $i - 1$. Considérons $g \in G_i$; on peut toujours l'écrire (algorithme F_5): $g = g_i f_i + \cdots + g_1 f_1$,pour des polynômes g_i. Alors d'après le lemme 5 on peut supposer que $\mathrm{LT}(g_i) \in \mathbb{K}[x_1, \ldots, x_{i-1}]$ et $\mathrm{LT}(g_i)$ réduit par rapport à G_{i-1}; or le nombre termes dans $\mathbb{K}[x_1, \ldots, x_{i-1}]$ qui ne sont pas top-réductibles par $\langle f_1, \ldots, f_{i-1} \rangle$ est exactement $N_{d,i} = \mathrm{HF}_{\langle f_1, \ldots, f_{i-1} \rangle}(d - d_i)$ dans $\mathbb{K}[x_1, \ldots, x_{i-1}]$ (on peut imaginer qu'on substitue $x_i = \cdots = x_n = 0$). En appliquant le théorème 35 avec $m = i - 1$ on a:

$$\sum_{d \geqslant 0} \mathrm{HF}_{\langle f_1, \ldots, f_{i-1} \rangle}(d)\, z^d = \sum_{d=0}^{\infty} N_{d+d_i,i} z^d = \frac{\prod_{j=1}^{i-1} (1 - z^{d_j})}{(1 - z)^{i-1}}$$

ce qui prouve le théorème. □

Corollaire 9. *Pour des équations quadratiques $\sum_{d=0}^{\infty} N_{d,i} z^d = z^2 (1 + z)^{i-1}$ et donc $N_{d,i} = \binom{i-1}{d-2}$.*

Si de plus $m = n$, le nombre total d'éléments dans la base de Gröbner est majoré par $\sum_{i=1}^{n} \sum_{d=2}^{i+1} \binom{i-1}{d-2} = \sum_{i=1}^{n} 2^{i-1} = 2^n - 1$.

Remarque 28. *En fait, en pratique, la borne du théorème 60 est exacte. On peut aussi comparer le résultat de ce théorème avec la borne du corollaire 5: on trouvait que le nombre d'éléments étaient bornés par $\leqslant n\, D(I) = n\, 2^n$. Dans le cas d'un SNP la borne 9 est meilleure.*

18.5 Complexité arithmétique de F_5

Théorème 61. *Le nombre total d'opérations arithmétique utilisé par l'algorithme F_5 pour calculer une base de Gröbner basis d'un système homogène $(f_1, \ldots, f_m)$ pour lequel les variables $(x_1, \ldots, x_n)$ sont en SNP est borné par:*

$$N_{F_5} = \sum_{i=1}^{m} \sum_{d=d_i}^{D} N_{d,i} \binom{i + d - 1}{d} \binom{n + d - 1}{d}, \tag{20}$$

où $D = 1 + \sum_{j=1}^{m} (d_j - 1)$ et $N_{d,i}$ est donné dans le théorème 60.

Lorsque $m = n$ et $\deg(f_i) = d_i = 2$ la formule se simplifie en

$$N_{F_5} = \sum_{d=0}^{n-1} \binom{2d + 2}{d} \binom{n + d + 1}{d + 2} \binom{n + d + 2}{2d + 3} - \binom{n + 1}{2} \tag{21}$$

Proof. Les opérations arithmétiques proviennent des réductions durant l'élimination de Gauß. Pour tout $1 \leqslant i \leqslant m$ et tout $d_i \leqslant d \leqslant D$, d'après le théorème 60 il y au plus $N_{d,i}$ polynômes dans $M^{(d)}([f_1, \ldots, f_i])$ qui n'était pas dans $M^{(d)}([f_1, \ldots, f_{i-1}])$ et on doit les réduire par $\widetilde{M^{(d)}}([f_1, \ldots, f_{i-1}])$. D'après le lemme 5 les termes de têtes du résultat sont dans $\mathbb{K}[x_1, \ldots, x_i]_d$, ce qui limite à $\binom{i+d-1}{d}$ le nombre de lignes impliquées dans la réduction chacune de ces lignes ayant au plus $\binom{n+d-1}{d}$ éléments non nuls. On obtient ainsi la formule (20) . Pour la formule (21) voir dans (Bardet *et al.* , 2014). □

Lorsqu'on applique l'algorithme F_5 on se trouve face à deux stratégies: en degré d utiliser autant que possible les calculs effectués en degré $< d$ ou générer des matrices très creuses en utilisant uniquement des produits des équations initiales: dans le premier cas on a tendance à générer des matrices assez denses mais qui ont une structure triangulaires par blocs; dans le deuxième cas on génère une sous matrice de la matrice de Macaulay.

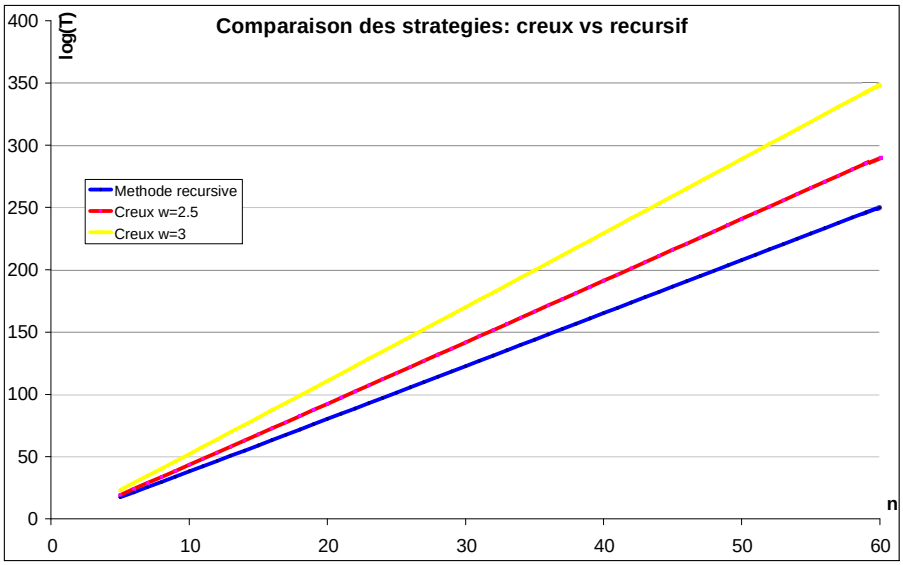


Figure 4: comparaison de deux stratégies.

La question naturelle est de déterminer quelle est la meilleure stratégie ? On peut faire le calcul explicitement pour des valeurs de n et comparer la formule (21) et la formule obtenue dans le théorème 53: par exemple pour $n = 30$ équations quadratiques sur $\mathbb{Q}$ on trouve

$$\begin{array}{rcl} N_{F_5} & = & 2^{122.6} \text{ par l'équation (21)} \\ & & 2^{56.7\,\omega} \text{ avec le théorème 53} \end{array}$$

par conséquent la borne est bien meilleure pour la valeur réaliste de $\omega = 3$. Pour d'autres valeurs de n et de $\omega = 3$ ou 2.5 on reporte sur un dessin les différentes valeurs des bornes obtenues pour l'algorithme F_5. Même si le dessin montre que la nouvelle borne est bien meilleure que la borne du théorème 53, il est cependant délicat de conclure en faveur d'une méthode ou d'une autre puisqu'on compare des bornes supérieures.

La borne (20) permet aussi de répondre à une autre question: quelle est l'étape la plus coûteuse ? Plus exactement si on considère un système quadratique SNP ayant n équations et n variables (21) la dernière étape, en degré $d_{\text{reg}} = n+1$, nécessite de générer la matrice dont la taille est la plus grande: cette étape est elle la plus coûteuse ? Afin de déterminer son maximum, on trace maintenant la fonction

$$F : d \longmapsto F(d) = \binom{2d+2}{d}\binom{n+d+1}{d+2}\binom{n+d+2}{2d+3}.$$

On étudie la fonction décroissante $f(d) = \frac{F(d+1)}{F(d)} - 1 = \frac{(2\,d+3)(n+d+2)(n+d+3)(n-d-1)}{(d+3)^2(d+1)(2\,d+5)} - 1$. On cherche la racine d_0 de f dans l'intervalle $\frac{n}{2} < d_0 < n$; lorsque $d > d_0$ on $F(d) < F(d_0)$. Asymptotiquement on cherche d_0 sous la forme $d_0 \approx \lambda n$ et on trouve:

$$f(\lambda n) = \frac{-2\lambda(2\,\lambda^3 + \lambda^2 - \lambda - 1)n^4 + \cdots}{\cdots}$$

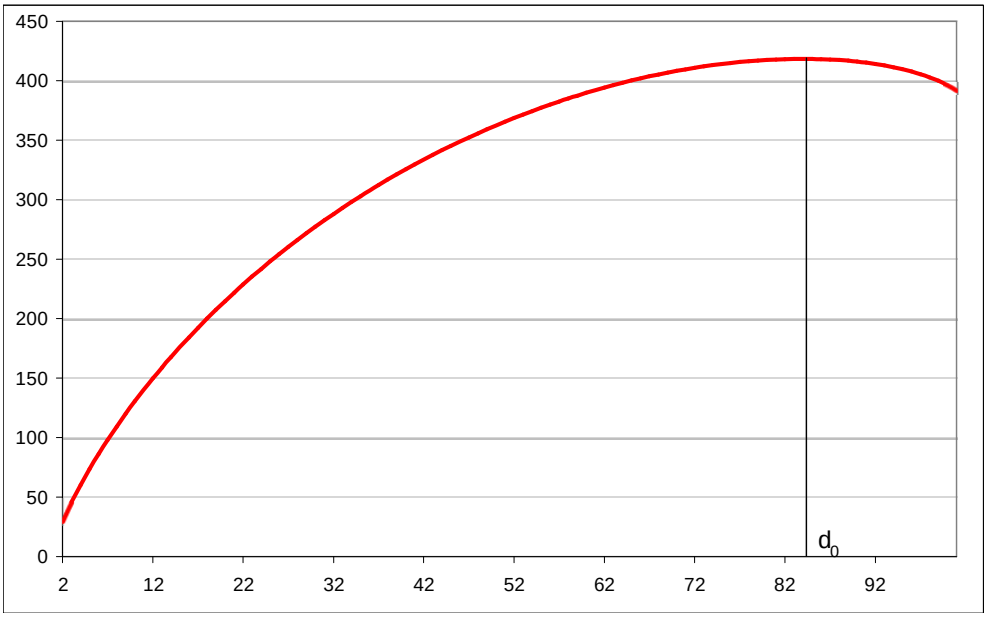


Figure 5: Coût relatif des algorithmes.

Par conséquent le coefficient de n^4 dans le numérateur doit être nul: ainsi on trouve $d_0 \approx 0.83n$. Maintenant $N_{F_5} = \sum_{d=0}^{m-1} F(d) - \binom{n+1}{2}$ est majoré par $n\,F(0.83\,n)$ et en utilisant la formule:

$$\log\left(\binom{an+b}{cn+d}\right) \approx (a\log(a) - c\log(c) - (a-c)\,\log(a-c))n$$

on trouve $\log(F(d_0)) \approx F_0\,n$ avec $F_0 \approx 4.3$. On obtient donc le résultat suivant:

Théorème 62. *Le nombre total d'opérations arithmétique utilisé par l'algorithme F_5 pour calculer une base de Gröbner d'un système quadratique homogène $(f_1, \ldots, f_m)$ pour lequel les variables $(x_1, \ldots, x_n)$ sont en SNP est borné par:*

(i) pour l'algorithme F_5 sous forme matricielle:

$$N_{F_5} \approx 2^{4.3\,n+o(n)}$$

(ii) par la méthode d'élimination de Gauß de la matrice de Macaulay:

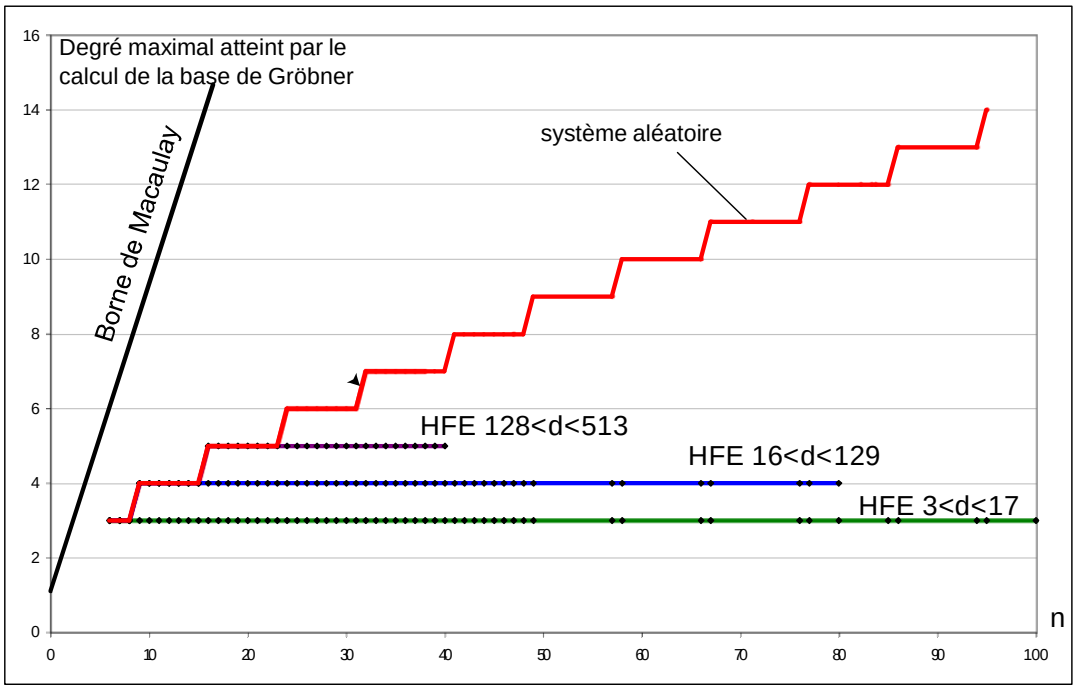
$$N_{Macaulay} \approx \frac{1}{4\pi^{\frac{3}{2}}\sqrt{n}} 2^{6\,n}$$

Proof. Pour la matrice de Macaulay en degré $d_{\text{reg}} = n+1$ il y a $M_{d_{\text{reg}}}(n) = \binom{n+d_{\text{reg}}-1}{d_{\text{reg}}} = \binom{2\,n}{n+1}$ colonnes et $n\,M_{d_{\text{reg}}-2}(n) = n\,\binom{2\,n-2}{n-1}$ lignes. La complexité de l'élimination de Gauß est donc bornée par $n\,M_{d_{\text{reg}}-2}(n)\,M_{d_{\text{reg}}}(n)^2$. La fonction `asympt` de Maple donne le résultat. □

18.6 Conclusion

Dans cette section on a étendu la définition de suite semi-régulière pour les systèmes surdéterminées: on conjecture que presque tout système est une suite semi-régulière (en caractéristique 0 c'est une conséquence de la conjecture

de (Fröberg., 1985)). Pour ces systèmes on donne des équivalents asymptotiques très précis du degré de régularité. Par exemple dans le cas de $2\,n$ équations en n variables on améliore la borne de Macaulay d'un facteur 11. De plus on répond à la question posée dans l'introduction et on peut étendre la courbe théorique (figs 2 et 18.6):



De plus le calcul d'une base de Gröbner d'un système semi-régulier ayant $m = \alpha n$ équations et n variables se fait en temps simplement exponentiel; par exemple un système ayant aléatoire avec 80 équations quadratiques est impossible à résoudre par les techniques Gröbner. Les systèmes algébriques constituent donc une source intéressante de problème difficile qui peuvent être utilisé pour concevoir de nouveaux cryptosystèmes.

Pour des suites réguliers vérifiant une propriété plus forte (être en position de Noether simultanée) on donne des estimations plus précises de la complexité arithmétique de F_5 en utilisant la structure d'une base DRL calculée par cet algorithme: on améliore ainsi l'exposant de la borne de complexité.

authordate4.bst - As authordate3, but with downstyle titles.

References

Arri, A., & Perry, J. 2011. The F5 Criterion revised. *Journal of Symbolic Computation*, **46**(2), 1017–1029. Preprint online at `arxiv.org/abs/1012.3664`.

Ars, G., Faugère, J.-C., Imai, H., Kawazoe, M., & Sugita, M. 2004. Comparison between XL and Gröbner Basis Algorithms. *In:* LEE, Pil Joong (ed), *AsiaCrypt 2004*. LNCS. Springer. to appear.

Auzinger and Stetter H. 1998. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. *Int. Series of Numerical Math.*, **86**, 11–30.

Bardet, M., Faugère, J.C, & B., Salvy. 2004 (Nov.). On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. *Pages 71–75 of:* Valibouze, A (ed), *ICPSS Paris*.

Bardet, Magali. 2004. *Étude des systèmes algébriques surdeterminés. Applications aux codes correcteurs et à la cryptographie.* Ph.D. thesis, Université Paris 6.

Bardet, Magali, Faugère, Jean-Charles, & Salvy, Bruno. 2005. Asymptotic Behaviour of the Index of Regularity of Semi-Regular Quadratic Polynomial Systems. *In: Proceedings of the8th MEGA (Effective Methods in Algebraic Geometry)*. 15 pages.

Bardet, Magali, Faugère, Jean-Charles, Salvy, Bruno, & Spaenlehauer, Pierre-Jean. 2013. On the Complexity of Solving Quadratic Boolean Systems. *Journal of Complexity*, **29**(1), 53–75.

Bardet, Magali, Faugère, Jean-Charles, & Salvy, Bruno. 2014. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, Sept., 1–24.

Becker T. and Weispfenning V. 1993. *Groebner Bases, a Computationnal Approach to Commutative Algebra*. Graduate Texts in Mathematics. Springer-Verlag.

Bourgeois, G. 2006 (Juin). *Attaque algébrique de NTRU à l'aide des vecteurs de Witt*. http://arxiv.org/ftp/cs/papers/0605/0605136.pdf.

Buchberger, B. 1987. History and Basic Features of the Critical-Pair/Completion Procedure. *Journal of Symbolic Computation*, **3**(1 and 2), 3–38.

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

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

Buchberger B. 1979. A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Basis. *Pages 3–21 of: Proc. EUROSAM 79*. Lect. Notes in Comp. Sci., vol. 72. Springer Verlag.

Buchberger B. 1985. Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory. *Chap. 6, pages 184–232 of:* Reidel Publishing Company (ed), *Recent trends in multidimensional system theory*. Bose.

Caniglia L. and Galligo A. and Heintz J. 1988. Some new effectivity bounds in computational geometry. *Pages 131–152 of: Proceedings of AAECC-6*. Lect. Notes in Comp. Sci., vol. 357. Springer Verlag.

Caniglia L. and Galligo A. and Heintz J. 1991. Equations for the projective closure and effective Nullstellensatz. *Discrete Applied Math.*, **33**, 11–23.

Chester, C., Friedman, B., & F.Ursell. 1957. An extension of the method of steepest descents. *Proc. Camb. Philos. Soc*, **53**, 599–611.

Courtois, Nicolas, Shamir, Adi, Patarin, Jacques, & Klimov, A. 2000. Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations. *Pages 392–407 of: Eurocrypt'2000*. Lecutures Notes in Computer Science, vol. 1807. Springer Verlag.

Cox, D., Little, J., & O'Shea, D. 1998. *Using Algebraic Geometry*. Springer Verlag, New York.

Cox, D., Little, J., & O'Shea, D. 2007. *Ideals, Varieties and Algorithms*. 3rd ed. 2007. corr. 2nd printing, 2008, xvi edn. Undergraduate Texts in Mathematics. Springer Verlag, New York. 560 p. 93 illus., Hardcover.

D., Bayer, & M., Stillman. 1987. A theorem on refining division orders by the reverse lexicographic orders. *Duke J. Math.*, **55**, 321–328.

Eder, Christian, & Faugère, Jean-Charles. 2014 (Apr.). *A survey on signature-based Gröbner basis computations*.

Eder, C. 2008. A new attempt on the F5 Criterion. *The Computer Science Journal of Moldova*, **16**, 4–14.

Eder, C., & Perry, J. 2010. F5C: A Variant of Faugère's F5 Algorithm with reduced Gröbner bases. *Journal of Symbolic Computation, MEGA 2009 special issue*, **45**(12), 1442–1458. dx.doi.org/10.1016/j.jsc.2010.06.019.

Eder, C., & Perry, J. 2011. Signature-based Algorithms to Compute Gröbner Bases. *Pages 99–106 of: ISSAC 2011: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*.

Eder, C., & Roune, B. H. 2013. Signature Rewriting in Gröbner Basis Computation. *Pages 331–338 of: ISSAC 2013: Proceedings of the 2013 international symposium on Symbolic and algebraic computation*.

Faugère, J.-C. 2002. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). *Pages 75–83 of: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. ISSAC '02. New York, NY, USA: ACM.

Faugère, Jean-Charles, & Mou, Chenqi. 2011. Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices. *Pages 115–122 of: Proceedings of the 36th international symposium on Symbolic and algebraic computation*. ISSAC '11. New York, NY, USA: ACM.

Faugère, Jean-Charles, & Rahmany, Sajjad. 2009. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. *Pages 151–158 of: ISSAC '09: Proceedings of the 2009 international symposium on Symbolic and algebraic computation*. ISSAC '09. New York, NY, USA: ACM.

Faugère, Jean-Charles, & Svartz, Jules. 2013. Gröbner Bases of ideals invariant under a Commutative group : the Non-modular Case. *Pages 347–354 of: Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*. ISSAC '13. New York, NY, USA: ACM.

Faugère, Jean-Charles, Safey El Din, Mohab, & Spaenlehauer, Pierre-Jean. 2011. Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree : Algorithms and complexity. *Journal of Symbolic Computation*, **46**(4), 406–437.

Faugère, Jean-Charles, Spaenlehauer, Pierre-Jean, & Svartz, Jules. 2014a (July). Sparse Gröbner Bases: the Unmixed Case. *In: ISSAC 2014*. 20 pages, Corollary 6.1 has been corrected.

Faugère, Jean-Charles, Gaudry, Pierrick, Huot, Louise, & Renault, Guénaël. 2014b. Sub-cubic Change of Ordering for Gröner Basis: A Probabilistic Approach. *Pages 170–177 of: ISSAC '14 - Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. ISSAC '14. Kobe, Japan: ACM.

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

Faugère, Jean-Charles and Lachartre, Sylvain. 2010 (July). Parallel Gaussian Elimination for Gröbner bases computations in finite fields. *Pages 89–97 of:* Moreno-Maza, M., & Roch, J.L. (eds), *Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*. PASCO '10. ACM, New York, NY, USA.

Fröberg, R. 1997. *An introduction to Gröbner bases*. Pure and Applied Mathematics. Chichester: John Wiley and Sons Ltd.

Fröberg., Ralf. 1985. Hilbert series of graded algebras. *Math. Scand.*, **56**(2), 117–144.

F.S., Macaulay. 1916. *The algebraic theory of modular systems*. Cambridge library. John Wiley and Sons Ltd.

G., Ars. 2005 (June). *Applications des bases de Gröbner Ã la cryptographie*. Ph.D. thesis, UniversitÃ© de Rennes 1.

Gao, S., Guan, Y., & Volny IV, F. 2010. A new incremental algorithm for computing Gröbner bases. *Pages 13–19 of: ISSAC '10: Proceedings of the 2010 international symposium on Symbolic and algebraic computation*. ACM.

Gao, S., Volny IV, F., & Wang, D. 2010. *A new algorithm for computing Groebner bases*. http://eprint.iacr.org/2010/641.

Gao, S., Volny IV, F., & Wang, D. 2011. *A new algorithm for computing Groebner bases (rev. 2011)*. http://www.math.clemson.edu/~sgao/papers/gvw.pdf.

Gao, S., Volny IV, F., & Wang, D. 2013. *A new algorithm for computing Groebner bases (rev. 2011)*. http://www.math.clemson.edu/~sgao/papers/gvw_R130704.pdf.

Gebauer, R., & Moller, H. M. 1986 (July). Buchberger's Algorithm and Staggered Linear Bases. *Pages 218–221 of: Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation*.

Gebauer, R., & Möller, H.M. 1988. On an Installation of Buchberger's Algorithm. *Journal of Symbolic Computation*, **6**(2 and 3), 275–286.

Gerdt V.P. 1995. Involutive Polynomial Bases. *In: PoSSo on software*. Paris, F.

Giovini A. and Mora T. and Niesi G. and Robbiano L. and Traverso C. 1991. One sugar cube, please, or Selection strategies in the Buchberger Algorithm. *In:* S. M. Watt (ed), *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*. ACM Press.

Giusti, M. 1994. Some effectivity problems in polynomial ideal theory. *Pages 159–171 of: Proc. Int. Symp. on Symbolic and Algebraic Computation EUROSAM 84, Cambridge (England)*. LNCS, vol. 174. Springer.

Hoffstein, J., Pipher, J., & Silverman., J.H. 1998. NTRU: a ring-based public key cryptosystem. *Pages 267–288 of: ANTS III*. Lectures Notes in Computer Science, vol. 1423. Springer Verlag.

Huang, L. 2010. *A new conception for computing Gröbner basis and its applications*. http://arxiv.org/abs/1012.5425.

J., Apel, & R., Hemmecke. 2002. *Detecting unnecessary reductions in an involutive basis computation.* Tech. rept. RISC Linz Report Series.

J., Sylvester. 1853. 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.

Lang, S. 2002. *Algebra (3nd Ed.)*. Graduate Texts in Mathematics – vol. 211. New York: Springer-Verlag.

Lazard D. 1981. Resolution des systemes d'equations algebriques. *Theor. Comp. Science*, **15**, 77–110.

Lazard D. 1983. Gaussian Elimination and Resolution of Systems of Algebraic Equations. *Pages 146–157 of: Proc. EUROCAL 83*. Lect. Notes in Comp. Sci, vol. 162.

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

Möller H.M. 1993. Systems of algebraic equations solved by means of endomorphisms. *Pages 43–56 of: Proceedings of AAECC-10*. Lect. Notes in Comp. Sci., vol. 673.

Mora, T. and Möller, H.M. and Traverso, C. 1992. Gröbner Bases Computation Using Syzygies. *Pages 320–328 of:* Wang, Paul S. (ed), *ISSAC 92*. ACM Press.

Pan, S., Hu, Y., & Wang, B. 2012. *The Termination of Algorithms for Computing Gröbner Bases*. http://arxiv.org/abs/1202.3524.

Pan, S., Hu, Y., & Wang, B. 2013. The Termination of the F5 Algorithm Revisited. *Pages 291–298 of: ISSAC 2013: Proceedings of the 2013 international symposium on Symbolic and algebraic computation*.

Pardue, Keith, & Richert, Benjamin. 2009. Syzygies of semi-regular sequences. *Illinois J. Math.*, **53**(1), 349–364.

Rouillier, F. 1999. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, **9**(5), 433–461.

Roune, B. H., & Stillman, M. 2012a. Practical Gröbner Basis Computation. *In: ISSAC 2012: Proceedings of the 2012 international symposium on Symbolic and algebraic computation*.

Roune, B. H., & Stillman, M. 2012b. *Practical Gröbner Basis Computation*. http://arxiv.org/abs/1206.6940.

Smart, Nigel, Vercauteren, Fre, & Silverman, Joseph H.. 2005. An algebraic approach to NTRU $(q = 2^n)$ via Witt vectors and overdetermined systems of nonlinear equations. *Pages 278–298 of: Security in Communication Networks (SCN 2004)*. Lecutures Notes in Computer Science, vol. 3352. Springer-Verlag.

Sun, Y., & Wang, D. K. 2011. A generalized criterion for signature related Gröbner basis algorithms. *Pages 337–344 of: ISSAC 2011: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*.

Szanto, A. 2004. Multivariate subresultants using Jouanolou s resultant matrices. *Journal of Pure and Applied Algebra*. to appear.

Van der Waerden B.L. 1991. *Algebra*. Springer-Verlag. seventh edition.

Volny, F. 2011. New algorithms for computing Gröbner bases. Ph.D. thesis, Clemson University.

V.P. Gerdt and Yu.A.Blinkov. 1998. Involutive Bases of Polynomial Ideals. *athematics and Computers in Simulation*, **45**, 519–542.