Rapport de TER

Benjamin Auder

Mai 2006

Algorithmes de calcul des bases de Groebner et applications

encadré par Bernard Parisse (UJF Grenoble 1).

Table des matières

Note : la plupart des démonstrations théoriques concernant l'existence des bases de Groebner et les algorithmes de recherche de bases, ainsi que l'introduction sur les ordres monômiaux sont tirées du cours de Frédéric Chyzak cité en référence.

Qu'est ce qu'une base de Groebner ? A quoi sert-elle ? Comment en trouver ? Autant de questions auxquelles on répondra dans cet exposé.

1 Extension de la division euclidienne

Pour des polynômes en une indéterminée sur un anneau A, $A[X]$ étant principal on a une division euclidienne (pour tout I idéal de $A[X]$, il existe P dans $A[X]$ tel que $I = (P)$ où (P) représente l'idéal engendré par P).

$$\forall P, Q \in A[X], \quad \exists U, V \in A[X] / \quad P = UQ + V, d(V) < d(Q)$$

$d(P)$ étant le degré de P.

On peut en fait considérer que l'on a divisé par l'idéal $I = (N)$, car R est l'unique représentant de la classe de P modulo l'idéal $I = (N) = AN$. Dans le cas d'un anneau de polynômes à plusieurs indéterminées, dans lequel les idéaux ne sont pas nécessairement principaux, il semble donc naturel d'introduire une division par une famille de diviseurs, celle-ci étant définie comme l'association à P d'un de ses représentants dans $A/(Q_1, ..., Q_n)$ où Q_1,..,Q_n sont des polynômes de $A[X_1, ..., X_n]$, et $(Q_1, ..., Q_n)$ est l'idéal engendré par ces éléments.

Un tel représentant est-t-il unique ? Sous quelles conditions ? On répondra à ces questions plus loin.

1.1 Monoïde des monômes et ordres monômiaux

Un polynôme à plusieurs variables s'écrivant $\sum_{\alpha_1,...,\alpha_n} a_{\alpha_1,...,\alpha_n} X_1^{\alpha_1}..X_n^{\alpha_n}$ où les $a_{\alpha_1,...,\alpha_n}$ ne sont non nuls qu'en un nombre fini, éléments de l'anneau A. On définit alors un *monôme* de P comme étant une des expressions $X_1^{\alpha_1}...X_n^{\alpha_n}$ occurrant dans le polynôme. Un monôme est un cas particulier de polynôme.

Un *monoïde* M est un ensemble muni d'une loi interne associative pour laquelle il existe un élément neutre. On peut définir naturellement une structure de monoïde sur l'ensemble des monômes, ce qui sera très utile par la suite :

Neutre : $X_1^0...X_n^0$
Loi produit : $(X_1^{\alpha_1}..X_n^{\alpha_n})x(X_1^{\beta_1}...X_n^{\beta_n})$=$(X_1^{\alpha_1+\beta_1}...X_n^{\alpha_n+\beta_n})$
C'est le monoïde commutatif libre sur les n génerateurs $X_1, ..., X_n$, noté $[X_1, ..., X_n]$.

Un *ordre monomial* sur un monoïde M est une relation d'ordre strict $<$ qui est :

- totale : deux éléments peuvent toujours être comparés
- compatible avec le produit : si $m_1 < m_2$ alors $mm_1 < mm_2$ pour tout m dans M
- un bon ordre : tout ensemble non vide de monômes a un plus petit élément, ou de manière équivalente, il n'existe pas de suite strictement décroissante dans M

Remarque : le neutre défini plus haut est le plus petit élement du monôide des monômes, quel que soit l'ordre monômial ; on le note 1. En effet, dans le cas contraire on aurait $X_i < 1$ pour un certain i, puis par récurrence immédiate ${X_i}^{k+1} < {X_i}^k$, d'où l'existence d'une suite strictement décroissante, et donc l'ordre ne serait pas monômial. Ce serait aussi contradictoire avec le fait que $A[X_1, ..., X_n]$ est noetherien, comme démontré plus loin.

Notation : un multi-indice est défini comme suit : $\alpha = \alpha_1, ..., \alpha_n \in A^n$
On l'utilise dans les exemples qui suivent.

Exemples d'ordres dans le cas des monômes (qui sera le seul considéré par la suite) :

- ordre lexicographique (ordre du dictionnaire) : $X^\alpha <_{lex} X^\beta$ si $\alpha_k < \beta_k$ pour k=min$\{$i/$\alpha_i \neq \beta_i\}$, autrement dit si la première valeur non nulle de la suite $\alpha_1 - \beta_1, \alpha_2 - \beta_2, ...$ est strictement négative
- ordre lexicographique gradué (ordre du degré total raffiné par $<_{lex}$) : $X^\alpha <_{glex} X^\beta$ si $|\alpha| < |\beta|$ ou ($|\alpha| = |\beta|$ et $X^\alpha <_{lex} X^\beta$)
- ordre lexicographique renversé : $X^\alpha <_{revlex} X^\beta$ si $\alpha_k > \beta_k$ pour k=max$\{$i/$\alpha_i \neq \beta_i\}$, autrement dit si la dernière valeur non nulle de la suite $\alpha_1 - \beta_1, \alpha_2 - \beta_2, ...$ est strictement positive
- ordre lexicographique renversé gradué (ordre du degrtotal raffinpar $<_{grevlex}$: $X^\alpha <_{grevlex} X^\beta$ si $|\alpha| < |\beta|$ ou ($|\alpha| = |\beta|$ et $X^\alpha <_{revlex} X^\beta$)

Remarque : $<_{revlex}$ n'est pas un ordre monômial, car pour celui-ci $X_1 < 1$.
Il est très important d'avoir un ordre monômial pour que l'algorithme de réduction présenté plus loin se termine. Tous les autres ordres présentés sont bien des ordres monômiaux.

Il est clair que l'ordre lexicographique est un ordre monômial. Montrons que l'ordre lexicographique renversé gradué est bien un ordre monômial ; c'est un des ordres que l'on utilisera pour les algorithmes.
Par définition de $<_{grevlex}$ et du produit, l'ordre est total et compatible avec le produit.
Supposons qu'il existe une suite strictement décroissante de monômes $m_0, ..., m_k, ...$ pour grevlex.
Soit d_0 le degré total de m_0. Il n'y a alors qu'un nombre fini de degrés totaux possibles pour tous les autres monômes de la suite, à savoir $d_0, d_0 - 1, ..., 0$.
Etant donné qu'il y a une infinité de monômes dans la suite, on en déduit l'existence de $d \in [0..d_0]$ tel que tous les monômes de la suite aient pour degré total d à partir d'un certain rang.
Or il n'y a qu'un nombre fini de monômes à degré total fixé. On obtient donc une contradiction, et $<_{grevlex}$ est bien un ordre monômial.

Le point crucial dans la démonstration précedente est la décroissance des degrés totaux. L'ordre lexicographique renversé n'a pas cette propriété.

1.2 Terminologie utilisée

A présent on fixe un ordre monômial sur $[X_1, ..., X_n]$ et on fixe un anneau A commutatif.

Un polynôme en plusieurs variables étant sous la forme
P=$\sum_{\alpha_1,...,\alpha_n} a_{\alpha_1,...,\alpha_n} X_1^{\alpha_1}..X_n^{\alpha_n}$, on appelle *terme* le produit d'un élément de l'anneau A (appelé *coefficient*) par un monôme.

Le *monôme de tête* de P est le plus grand de ses monômes $X_1^{\alpha_1}..X_n^{\alpha_n}$ pour l'ordre monômial choisi. De même, le *terme de tête* de P est le terme de P qui divisé par son coefficient est égal au monôme de tête. On notera $mt(P)$ le monôme de tête d'un polynôme P, $tt(P)$ son terme de tête et $ct(P)$ son coefficient de tête.

Par définition d'un ordre monômial, le monôme de tête d'un produit de deux polynômes P et Q est le produit des monômes de tête $mt(P)$ et $mt(Q)$.
En effet, notons $P = \sum_{i=0}^{r} c_i M_i$ et $Q = \sum_{i=0}^{s} c'_i M'_i$ où les M_i, M'_i sont les monômes de P, Q, avec $m_r < ... < m_0$ et $m'_s < ... < m'_0$.
Alors $PQ = \sum_{i=0}^{r} \sum_{j=0}^{s} c_i c'_j m_i m'_j$
Or on a $m_i \leq m_0$ et $m'_j \leq m'_0$, d'où $m_i m'_j \leq m_0 m'_j$ et $m_0 m'_j \leq m_0 m'_0$, soit finalement $m_i m'_j \leq m_0 m'_0$.

Au passage, on a aussi montré que les coefficients de tête ainsi que les termes de tête se multiplient.

Remarque : ce qui précède est possible car l'anneau A est supposé commutatif ; on le supposera ainsi dans la suite, même si certains résultats resteraient valides dans un cadre plus géneral.

Une *partie stable* P d'un monôïde M est un sous-ensemble quelconque de M stable par la loi $*$ de M. C'est en quelque sorte l'équivalent d'un idéal dans le cadre plus général d'un monoïde.
Par exemple, dans $\mathbb{N}$, l'ensemble des entiers pairs est stable par addition, ce qui n'est pas le cas des entiers impairs (on en déduit au passage que le complémentaire d'une partie stable n'en est pas nécessairement une).

De façon analogue aux idéaux donnés par générateurs, étant donné une famille $\{s_i\}_{i\in I}$ d'éléments de M, l'ensemble

$$S = \{ms_i \in M/m \in M, i \in I\} = \cup_{i\in I} Ms_i$$

est une partie stable du monoïde M, appelée la partie stable de M *engendrée par la famille de générateurs* s_i. Toute partie stable peut être vue comme engendrée par un nombre fini de générateurs.

Dans la suite, on notera $mt(I)$ la partie stable constituée des monômes de tête des polynômes de l'idéal I, et on posera dans le cas d'un système de générateurs $G = \{g_1, ..., g_k\}, mt(G) = \{mt(g_1), ..., mt(g_k)\}$.

1.3 De la division euclidienne à la réduction

Algorithmiquement, la division euclidienne procède par une succession de "divisions élementaires", où l'on ne considère que les quotients Q qui sont des monômes et donc des restes qui ne sont pas minimaux au sens du degré. Dans le cas de plusieurs indéterminées, ces étapes sont appelées "réductions".

On a vu en introduction qu'il serait intéressant de "diviser" un polynôme par plusieurs autres polynômes. Une réduction d'un polynôme P par un ensemble de polynômes E consiste à effectuer les opérations suivantes :

- trouver un polynôme de E (noté Q) dont le terme de tête divise le terme dominant de P.
  Si un tel polynôme n'existe pas, la réduction est finie. On remarque

que pour des polynômes à coefficients dans un corps, on peut remplacer "terme" par "monôme"
- effectuer $P \leftarrow P - \frac{ct(P)}{ct(Q)} mQ$ où m est le monôme $\frac{mt(P)}{mt(Q)}$, puis revenir à l'étape précédente

L'algorithme est correct si on a un ordre monômial sur l'ensemble des monômes de $A[X_1, ..., X_n]$, dorénavant noté M.
On observe que contrairement à ce qui se passe pour la division euclidienne, le résultat d'une telle "division" n'est pas unique, qu'il s'agisse du reste ou bien des quotients successifs.

On vérifie par exemple que si l'on réduit $P = X^2Y + XY^2 + XY$ par $\{P_1, P_2\} = \{X + Y, X^2 + Y^2\}$ en suivant l'ordre lexicographique, on obtient $P = (XY + Y)P_1 - Y^2$ en réduisant par P_1 d'abord, et $P = (Y^2 + Y)P_1 + YP_2 - 2Y^3 - Y^2$ en commençant par P_2.

Dans le cas d'un anneau euclidien, on dit que l'on a un génerateur d'un idéal si l'on trouve un polynôme P dans l'idéal tel que tout autre polynôme Q de l'idéal s'écrive $Q = UP$ où U est un polynôme ; en d'autres termes, Q est divisible par P. On va généraliser cette idée au cas d'anneaux de polynômes à plusieurs variables en remplaçant la division par la réduction introduite ci-dessus : une base de Groebner G d'un idéal polynômial I est un système de générateurs de I tel que tout polynôme de $A[X_1, ..., X_n]$ a un unique reste dans la réduction par G. Il est intéressant de constater que si l'on peut alors garantir l'unicité du reste, le quotient quant à lui n'est pas unique.

2 Bases de Groebner d'un idéal

Après avoir donné une des principales propriétés d'une base de Groebner laissant entrevoir son utilité, donnons-en une définition plus précise avant d'étudier quelques unes de ses applications théoriques.

2.1 Définitions

Soit I un idéal de $A_n = K[X_1, ..., X_n]$ où K est un corps commutatif quelconque et $<$ un ordre monômial sur A_n. Un sous-ensemble **fini** G de $I - \{0\}$ est une *base de Groebner de I pour l'ordre* $<$ si l'une des propriétés équivalentes suivantes est vérifiée :

- 1. La partie stable de M engendrée par $mt(G)$ est égale à $mt(I)$

- 2. $mt(G)$ et $mt(I)$ engendrent le même idéal
- 3. Tout élément non nul de I est réductible par G
- 4. Pour tout P dans A_n, il existe un unique R dans A_n dont aucun monôme ne soit divisible par un monôme de $mt(G)$, et tel que $P - R$ soit dans l'déal I
- 5. Pour tout P dans I, P se réduit à zéro par G

Démonstration de l'équivalence :
$1 \Rightarrow 2$:
Supposons que

$$\cup_{g \in G} Mmt(g) = mt(I)$$

Passons alors aux idéaux. En notant (S) pour l'idéal de A_n engendré par la famille $\{s\}_{s \in S}$, on a les égalités :

$$(mt(I)) = \sum g \in G\big(Mmt(g)\big) = \sum_{g \in G} \big(mt(g)\big) = mt(G)$$

Donc $mt(G)$ et $mt(I)$ engendrent le même idéal.
$2 \Rightarrow 3$:
Supposons $(mt(G)) = (mt(I))$. Soit $P \in I - \{0\}$. On a d'abord l'égalité

$$mt(P) = \sum_{g \in G} q_g mt(g)$$

pour des polynômes q_g, puis en scindant en monômes, l'égalité

$$mt(P) = \sum_j c_j m_j mt(g_j)$$

pour des c_j de K, des monômes m_j et des g_j de G. Comme cette somme sur j est en fait une somme de termes, les termes en les monômes autres que $mt(P)$ doivent s'annuler, et on peut sans perte de généralité supposer que pour chaque j, $m_j mt(g_j) = mt(P)$. On a alors, pour j_0 l'un de ces j,

$$mt(P) = m_{j_0} mt(g_{j_0})$$

Donc P est réductible par G (au moins par g_{j_0}).
$3 \Rightarrow 4$:
Supposons que tout P non nul de I est réductible par G. Soit $P \in A_n$. On a l'existence énoncée au point (4) en prenant pour R le reste de la division de P par Q : alors $P - R$ est élément de I. Supposons que nous ayions deux écritures $P = H_i + R_i$ pour $i = \{1, 2\}$, avec $H_i \in I$ et des R_i dont aucun monôme n'est divisible par un monôme de $mt(G)$. Alors l'élément

$$R_1 - R_2 = H_2 - H_1 \in I$$

est soit nul, soit réductible. Supposons ctte différence non nulle ; alors $mt(R_1 - R_2)$ est forcément un monômeparmi ceux de R_1 et R_2. ce monôme de tête est à la fois non divisible par un monôme de $mt(G)$, par définition des R_i, et divisible par l'un d'entre eux, par l'hypothèse faite du point (3). C'est une contradiction, et on a l'unicité de R.
$4 \Rightarrow 5$:
Soit $P \in I$. En application du point (4), on trouve un R, qui par la preuve d'existence et d'unicité précédente ne peut être que le reste de la division de P par G. Comme $R = (R - P) + P$ est un élément de I mais n'est pas réductible, cela signifie que R est nul.
$5 \Rightarrow 2$:
Soit P réductible par G. Alors $mt(P)$ est dans la partie stable engendrée par $mt(G)$, donc dans l'idéal engendré par $mt(G)$. Supposons le point (5). Comme tout élément non nul de I est alors réductible par G, on a obtenu dans ce cas l'inclusion

$$(mt(I)) \subseteq (mt(G))$$

L'autre inclusion découle de $G \subseteq I$.
$5 \Rightarrow 2$:
L'inclusion de la partie stable S engendrée par $mt(G)$ dans $mt(I)$ provient de $G \subseteq I$. Pour l'autre inclusion, supposons le point (2) et soit $m \in mt(I) \subseteq (mt(I)) = (mt(G))$. Par le même raisonnement que pour l'implication $2 \Rightarrow 3$, on écrit m sous la forme $m_{j_0} mt(g_{j_0})$ qui est un élément de S. La partie stable $mt(I)$ est donc incluse dans S, ce qui achève la démonstration.

2.2 Existence, unicité et représentation en escalier

A ce stade, rien n'indique s'il existe une base de Groebner pour un idéal quelconque donné par un système de générateurs.
En fait la réponse à la question d'existence est positive, et passe par le lemme suivant :

Lemme de Dixon : Pour tout sous-ensemble S de $\mathbb{N}^n$, il existe des vecteurs $v_1, ..., v_r \in \mathbb{N}^n$ tels que $S \subseteq (v_1 + \mathbb{N}^n) \cup ... \cup (v_r + \mathbb{N}^n)$.

Intuitivement, pour $n = 2$ cela revient à dire que toute partie du plan discret $\mathbb{N}^2$ est située "au-dessus" d'un escalier à r marches dont les coins sont les vecteurs v_i. Pour démontrer ce résultat, on va ramener l'étude dans $\mathbb{N}^n$ à celle dans $\mathbb{N}^{n-1}$

Démonstration : On effectue une récurrence sur $n \in \mathbb{N}$, la propriété du lemme

étant notée H_n.
Initialisation : n=1
Alors en supposant S non vide il suffit de choisir v=min S pour obtenir $S \subseteq v + \mathbb{N}$. Si S est vide le résultat est immédiat.
Hérédité : soit $n \in \mathbb{N} - \{0\}$ tel que H_{n-1} soit vraie, et donnons-nous un sous-ensemble S de $\mathbb{N}^n$, non vide (comme précédemment, le cas où S est vide ne pose pas de problèmes).
Posons $\pi : \mathbb{N}^n \to \mathbb{N}^{n-1}$ l'application donnée par

$$\pi(x_1, ..., x_n) = (x_2, ..., x_n)$$

Alors

$$\pi(S) = \{\pi(s) | s \in S\} \subseteq \mathbb{N}^{n-1}$$

et par hypothèse de récurrence, il existe $s_1, ..., s_r \in S$ tels que

$$\pi(S) \subseteq (\pi(s_1) + \mathbb{N}^{n-1}) \cup ... \cup (\pi(s_r) + \mathbb{N}^{n-1})(*)$$

En général, S n'est pas inclus dans $(s_1 + \mathbb{N}^n) \cup ... \cup (s_r + \mathbb{N}^n)$, car les premières composantes des vecteurs s_i sont en fait choisies arbitrairement. On va pallier ce défaut en ajoutant d'autres vecteurs dans S.
Soit M le plus grand nombre apparaissant en première composante des vecteurs s_i.
Pour $i \in [0, .., M-1]$, on définit

$$S_i = \{s \in S | s^1 = i\}$$

et

$$S_{\geq M} = \{s \in S | s^1 \geq M\}$$

On vérifie alors que $S = S_0 \cup ... \cup S_{M-1} \cup S_{\geq M}$. En effet, le membre de droite est inclus dans celui de gauche par définition, et tout élément de S a une première composante dans $\mathbb{N}$, donc suivant la valeur de cette première composante s est dans l'un des S_i ou dans $S \geq M$; d'où l'autre inclusion.

Montrons que $S_{\geq M} \subseteq (s_1 + \mathbb{N}^n) \cup ... \cup (s_r + \mathbb{N}^n)$.
Soit $s \in S_{\geq M}$. Ecrivons $s = (s^1, ..., s^n)$. D'après (*),
$(s^2, ..., s^n) \in (\pi(s_1) + \mathbb{N}^{n-1}) \cup ... \cup (\pi(s_r) + \mathbb{N}^{n-1})$, donc
$\exists k \in [1, .., r] | (s^2, ..., s^n) \in \pi(s_k) + \mathbb{N}^{n-1}$. Par suite, en prenant pour s_k la même notation que pour s, $(s^2, ..., s^n) = ({s_k}^2, ..., {s_k}^n) + a$, pour un certain $a \in \mathbb{N}^{n-1}$. Etant donné que l'on a $s^1 \geq {s_k}^1$ par définition de M, on en déduit $s = s_k + (s^1 - {s_k}^1, a_1, ..., a_{n-1})$, d'où $s \in (s_1 + \mathbb{N}^n) \cup ... \cup (s_r + \mathbb{N}^n)$, et l'inclusion annoncée.

Puisque la première composante des vecteurs de S_i est fixée, on peut identifier chaque ensemble S_i à un sous-ensemble de $\mathbb{N}^{n-1}$, et appliquer à nouveau l'hypothèse de récurrence : pour chaque i, il existe $s_{i,1}, ..., s_{i,r_i} \in S_i$ vérifiant

$$S_i \subseteq (s_{i,1} + \mathbb{N}^n) \cup ... \cup (s_{i,r_i} + \mathbb{N}^n)$$

Rassemblant alors les résultats, on obtient

$$S \subseteq (s_1 + \mathbb{N}^n) \cup ... \cup (s_r + \mathbb{N}^n) \cup \cup i = 0..n\big\{(s_{i,1} + \mathbb{N}^n) \cup ... \cup (s_{i,r_i} + \mathbb{N}^n)\big\}$$

D'où l'assertion au rang n+1, et donc pour tout n.

On montre alors la version polynômiale du lemme précedent : toute partie stable S de $M = [X_1, ..., X_n]$ est finiment engendrée. *Démonstration* : Soit S une partie stable (non vide) de M. On applique le lemme précédent au sous-ensemble de $\mathbb{N}^n$ fomé par les n-uplets d'exposants des termes contenus dans S, ensemble noté E : il existe $v_1, ..., v_r$ des vecteurs de $\mathbb{N}^n$ tels que $E \subseteq \cup(v_1 + \mathbb{N}^n) \cup ... \cup (v_r + \mathbb{N}^n)$. Alors avec les notations précédentes

$$S \subseteq \big(\prod_{i=1}^{n} {X_i}^{{v_1}_i} + M\big) \cup ... \cup \big(\prod_{i=1}^{n} {X_i}^{{v_r}_i} + M\big)$$

d'où le résultat.

Nous pouvons maintenant énoncer le **théorème d'existence des bases de Groebner** :

Pour tout ordre monomial $<$ sur $A_n = A[X_1, ..., X_n]$, tout idéal I non nul de A_n admet une base de Groebner.
Démonstration : Soit I un idéal non nul de A_n. d'après la seconde version du lemme de Dickson, i existe un système fini de générateurs de mt(I). Considérons un relèvement de ce système en un système déléments de I. Par la première définition des bases de Groebner (par légalité des parties stables), celui-ci s'avère être une base de Groebner de I pour $<$.

On dit qu'une base de Groebner G d'un idéal I est **réduite** si chaque polynôme de G est irréductible par les autres éléments de G, et si chaque coefficient de tête des polynômes de G vaut 1 (on comprend donc l'utilité d'avoir choisi un corps et non un anneau). Montrons d'abord que l'on ne change pas l'idéal engendré par G si on remplace un des polynômes (appelons-le g_0) par le reste de sa réduction pa les autre spolynômes (notés $g_1, ..., g_t$).
Si $P = \sum_{i=0}^{t} \alpha_i g_i$, et $g_0 = \sum_{i=1}^{t} \beta_i g_i + r$, alors on a

$P = \alpha_0 r + \sum_{i=1}^{t}(\alpha_i + \alpha_0\beta_i)g_i$, d'où l'invariance annoncée.

Il reste à montrer l'unicité d'une base de Grebner réduite dans le cas d'un déal engendré :

Supposons que l'on ait $G = \{g_1, ..., g_t\}$ et $F = \{f_1, ..., f_l\}$ deux bases de Groebner réduites de I engendré par un nombre fini de polynômes. Soit $g_i \in G$. g_i se réduit à zéro par F car F est une base de Groebner de I, donc il existe $f_j \in F, \alpha \in \mathbb{N}^n$ vérifiant $tt(g_i) = tt(X^\alpha f_j)$. Par le même argument on trouve $g_k \in G, \beta \in \mathbb{N}^n$ tel que $tt(f_j) = tt(X^\beta g_k)$. D'où $tt(g_i) = tt(X^{\alpha+\beta} g_k)$. Or G est réduite, donc $\alpha = \beta = 0$, puis $tt(g_i) = tt(g_k)$, ce qui implique $g_i = g_k$.

Ainsi, $\forall g_i \in G, \exists f_j \in F | tt(g_i) = tt(f_j)$ (car on a normalisé les coefficients de tête à 1). De plus f_j est unique car F est réduite, d'où une injection des termes de tête de G dans ceux de F. En échangeant les rôles de G et F, on obtient aussi une injection des termes de tête de F dans ceux de G, d'où une bijection σ entre les termes de tête de G et de F d'après le théor'eme de Cantor-Bernstein.

Il reste alors à montrer que pour tout i, si $\sigma(tt(g_i)) = tt(f_j)$, alors $\sigma(g_i) = f_j$ (et donc $g_i = f_j$). Raisonnons par l'absurde en supposant $g_i \neq f_j$.

Ecrivons alors $tt(g_i - f_j) = c_\alpha X^\alpha$, où $\alpha \in \mathbb{N}^n$. En tant qu'élément de I $g_i - f_j$ se réduit à zéro par G, et donc pour un certain g_k de G on a $X^\alpha = X^\beta tt(g_k)$.
Si X^α est dans g_i, alors G ne serait pas réduite car on pourrait réduire g_i par g_k.
Si X^α est dans f_j, alors f_j ne serait pas réduit non plus : contradiction.

Dans le cas particulier de la dimension 2, le terme de tête d'un polynôme est représenté par un point à coordonnées entières dans le plan, et toute partie stable engendrée par les termes de tête d'une base de Groebner d'un idéal I dans A_2 prend la forme suivante :

2.3 Quelques propriétés et applications

L'existence des bases de Groebner nous donne une réponse constructive à la question de la finitude de la présentation des idéaux polynômiaux.

Théorème de Hilbert : Tout idéal I de $A_n = K[X_1, ..., X_n]$ admet un système fini de générateurs, ou, de façon équivalente, toute chaîne infinie

croissante (pour l'inclusion) d'idéaux deA_n stationne.

Démonstration : Toute base de Groebner a un systˋeme fini de générateurs, ce qui prouve le premier point. Pour léquivalence annoncée, supposons d'abord que toute chaîne infinie croissante d'idéaux de A_n stationne. Etant donné un idéal I qui ne soit pas finiment engendré, on peut trouver une suite infinie d'éléments dont chaque terme n'est pas dans l'idéal engendré par la sous-suite finie des termes précédents (par récurrence en extrayant des éléments de I). On crée ainsi une suite infinie strictement croissante d'idéaux, ce qui contredit l'hypothèse.
Donc tout idéal est finiment engendré.

Réciproquement, supposons que tout idéal soit finiment engendré et donnons-nous une chaîne infinie croissante d'idéaux. L'union de tous les idáux de la chaîne est un nouvel idéal, qui est alors finiment engendré. Soit un système fini de générateurs de l'union ; il existe un idéal de la chaîne qui contient tous ces générateurs. cet idéal, de même que tous les suivants dans la chaîne, est égal à l'union.

Essayons à présent de voir à quoi peut servir une base de Groebner, en dehors des considération théoriques précédentes.

Une caractéristique intéressante des bases de Groebner est la possibilité d'exprimer très simplement l'capion d'un idéal engendré avec l'ensemble des polynômes en les variables $X_i, i \geq i_0$ où $i_0 \in [1, .., n]$ lorsqu'on choisit l'ordre lexicographique $X_1 > X_2 > ... > X_n$. Plus précisément, notons $K[X^{i_0}]$ l'esemble des polynômes en les indéterminées $X_{i_0}, ..., X_n$ à coefficients dans K. Alors si G est une base de Groebner de I engendré par $P_1, ..., P_k$, $G \cap K[X^{i_0}]$ est une base de Groebner de $I \cap K[X^{i_0}]$.

Démonstration : Soit $i_0 \in [1, .., n]$. Il s'agit de montrer $I \cap K[X^{i_0}] = (G \cap K[X^{i_0}])$ où (.) désigne l'idéal engendré dans $K[X^{i_0}]$.
Une inclusion est évidente car $G \subseteq I$, montrons la seconde :
Soit $P \in I \cap K[X^{i_0}]$. En particulier $P \in I$, donc P se réduit à zéro par G : on peut écrire
$P = \sum_{j=1}^{k} \alpha_j g_j$ pour des g_j éléments de G. Or Les g_j sont tous dans $K[X^{i_0}]$ car chaque réduction se fait sur un élément de $K[X^{i_0}]$, par récurrence finie immédiate : le monôme de tête du polynôme réducteur est à chaque étape un diviseur du monôme de tête du polynôme courant, qui est dans $K[X^{i_0}]$ par hypothèse de récurrence. Conclusion :

$P \in (G \cap K[X^{i_0}])$.

Cette propriété permettra de résoudre des systèmes polynômiaux algorithmiquement pas triangulation.

Une autre application intéréssante consiste en la réecriture d'un polynôme comme fonction polynômiale d'autres polynômes. Par exemple on sait que les sommes de Newton s'expriment à l'aide des polynômes symétriques élémentaires, mais la simple application de la formule récurrente $P_d = \sum_{k=1}^{d-1}(-1)^{k-1}S_k P_{d-k} + (-1)^{d-1}dS_d$ valable pour tout $d \in \mathbb{N}$ devient vite laborieuse.
Supposons que l'on cherche à exprimer $P \in K[X_1, ..., X_n]$ en fonction de $P_1, ..., P_r \in K[X_1, ..., X_n]$: $P = Q(P_1, ..., P_r)$ où $Q \in K[T_1, ..., T_r]$. Considérons l'anneau de polynômes $A = K[X_1, ..., X_n, T_1, ..., T_r]$. Si l'on peut écrire

$$P = a_1(T_1 - P_1) + ... + a_r(T_r - P_r) + R$$

où $R \in K[T_1, ..., T_r]$ et $a_1, ..., a_r \in A$, on substitue alors P_i à T_i pour tout i, et on peut choisir Q=R.
Le théoréme suivant permet la décomposition adéquate de P :

Théorème : Soient $P, P_1, ..., P_r \in K[X_1, ..., X_n]$. Soit I l'idéal

$$I =< T_1 - P_1, ..., T_r - P_r >$$

dans l'anneau A. Soit G une base de Groebner de I pour l'ordre lexicographique donné par $X_1 \geq ... \geq X_n \geq T_1 \geq ... \geq T_r$.
Alors P peut être écrit comme un polynôme en $P_1, ..., P_r$ si et seulement si le reste R (noté P^G par la suite) dans la réduction de P par G est dans $K[T_1, ..., T_r]$. Dans ce cas, $P = R(P_1, ..., P_r)$.

Démonstration : Soit $G = \{g_1, ..., g_N\}$ une base de Groebner de I pour l'ordre lexicographique donné ci-dessus. L'algorithme de réduction nous donne la relation

$$P = a'_1 g_1 + ... + a'_N g_N + P^G$$

où $a'_1, ..., a'_N \in A$. Comme $< g_1, ..., g_N >= I$, on peut écrire les g_i à l'aide des polynômes générateurs $T_i - P_i$, et obtenir après réarrangement des termes

$$P = a_1(T_1 - P_1) + ... + a_N(T_N - P_N) + P^G$$

où $a_1, ..., a_N \in A$.
Si $P^G \in K[T_1, ..., T_r]$, alors on obtient $P = P^G(P_1, ..., P_r)$.

Réciproquement, supposons qu'il existe un polynôme Q dans $K[T_1, ..., T_r]$ tel que $P = Q(P_1, ..., P_r)$. On va montrer dans ce cas que P^G est aussi dans $K[T_1, ..., T_r]$.
On commence par remarquer que $P(X_1, ..., X_n) - Q(T_1, ..., T_r)$ est dans I, car I contient tous les polynômes qui s'annulent en $T_1 = P_1, ..., T_r = P_r$. On peut alors écrire

$$P = a_1(T_1 - P_1) + ... + a_r(T_r - P_r) + Q$$

où $a_1, ..., a_r \in A$. Par le même argument que ci-dessus, on peut réecrire cette relation avec les g_i :

$$P = b_1 g_1 + ... + b_N g_N + Q$$

pour des b_i éléments de A. On a de maniière évidente $P^G = Q^G$.
Supposons que Q soit réductible par un certain g_i. Alors le monôme de tête de g_i ne fait pas intervenir les indéterminées $X_1, ..., X_n$, par choix de l'ordre lexicographique (sinon il ne diviserait pas le monôme de tête de Q, qui lui est dans $M[T_1, ..., T_r]$). Par suite, $g_i \in K[T_1, ..., T_r]$, et donc le reste dans la réduction de Q par g_i ne fait intervenir que les indéterminées $T_1, ..., T_r$. Par récurrence finie immédiate, $Q^G \in K[T_1, ..., T_r]$, et donc $P^G \in K[T_1, ..., T_r]$, ce qui achève la preuve.

Voici finalement Un propriété des bases de Groebner liée à la finitude de la présentation des idéaux polynômiaux :

Tout idéal I de $A_n = K[X_1, ..., X]$ admet un ensemble fini de générateurs qui est une base de Groebner pour tous les ordres monomiaux possibles sur $M = [X_1, ..., X_n]$. Un tel système de gén'erateurs est appelé une **base de Groebner universelle**.

3 Algorithme de calcul

Jusque là on a vu ce qu'était une base de Groebner, et on en a démontré l'existence pour tout idéal de A_n. Mais on ne sait toujours pas comment, étant donné un système de générateurs G, trouver de manière constructive une base de Groebner de l'idéal egendré par G. C'est ce qu'on va faire ici.

3.1 Caractérisation en terme de S-polynômes

Un exemple simple motive la définition qui va suivre : soit I l'idéal $A_1(X-1) + A_1X$. L'"escalier" associé aux générateurs $X-1$ et X est réduit à la partie stable des pusisances de X à exposant strictement positif.

Pourtant, le polynôme $1(X-1)+(-1)X$ vaut 1 et la partie stable associée à l'idéal est aisni l'ensemble de tous les monômes en X. De manière générale, le phénomène est qu'un polynôme P peut très bien être dans un idéal $I = \sum_{i=1}^{r} A_n P_i$ sans que son monôme de tête ne soit dans la partie stable $\cup_{i=1}^{r} Mmt(P_i)$, ce qui a lieu lorsqu'une combinaison $\sum_{i=1}^{r} l_i P_i$ provoque une annulation des termes de tête des $l_i P_i$.

Définition des S-polynômes : Soient deux polynômes non nuls P_1 et P_2, et posons $m_1 = mt(P_1)$, $m_2 = mt(P_2)$ et $m = ppcm(m_1, m_2) = n_1 m_1 = n_2 m_2$. On appelle S-polynôme de P_1 et P_2 le polynôme suivant :

$$Spoly(P_1, P_2) = l_1 P_1 + l_2 P_2 \quad pour \quad l_1 = ct(P_2)n_1, \quad l_2 = -ct(P_1)n_2$$

Quel est le lien entre ces polynômes et les bases de Groebner ?

On voit que tout S-polynôme d'éléments d'un système de générateurs G est dans l'idéal I engendré par G, et donc doit être réduit à zéro par les éléments de G si G se destine à être une base de Groebner pour I. Or l'exemple précedent nous montre que le calcul de S-polynome peut mener à des polynômes non réductibles par G après annulation des termes de tête. Intuitivement on comprend donc que plus G réduit à zéro de S-polynômes, plus G est proche d'une base de Groebner car les S-polynômes ont été crées pour diminuer strictement le terme dominant entre deux polynômes.
On a en fait le résultat suivant :

Propriété caractéristique des bases de Groebner : Soit $G = \{P_k\}_{1 \leq k \leq r}$ un système de générateurs non nuls d'un idéal de polynômes. Tous les S-polynômes $Spoly(P_i, P_j)$ se réduisent à zéro par G si et seulement si G est une base de Groebner de l'idéal.

Démonstration : On montre l'implication directe en vérifiant que tout élément de l'idéal se réduit à zéro par G ; l'autre sens est trivial pardéfinition des bases de Groebner car tout S-polynôme est dans l'idéal. Raisonnons par l'absurde : soit P irréductible par P et exprimé sous la forme $\sum_{i=1}^{r} L_i P_i$. Sans perte de généralité, on peut supposer que le monôme $\delta = max_{1 \leq i \leq r}\{mt(L_i P_i)\}$ est minimal parmi les écritures de P en terme des P_i etque pour un entier k bien choisi, on a la relation $\delta = mt(L_i P_i) > mt(L_j P_j)$ dès que $1 \leq i \leq k < j \leq r$. Alors,

$$P = \sum_{i=1}^{k} tt(L_i)P_i + \sum_{i=1}^{k}(L_i - tt(L_i))P_i + \sum_{i=k+1}^{r} L_i P_i = \sum_{i=1}^{k} tt(L_i)P_i + \sum_{i=1}^{r} L'_i P_i$$

où dans la dernière somme on a $mt(L'_i P_i) < \delta$ dès que le polynôme L'_i est non nul. Sans plus de perte de généralité, on peut encore supposer que k est minimal parmi les écritures de P sous cette forme qui minimisent δ. Notons que k vaut au moins 2, sinon δ serait monôme de tête de P et P serait réductible par P_1. Observons que δ est divisible par le ppcm des monômes de tête de P_1 et P_2, car par la propriété multiplicative des monômes de tête $mt(P_i)$ divise δ pour $i \in \{1, 2\}$.

On introduit alors les monômes n_1 et n_2 qui interviennent dans la définition de $Spoly(P_1, P_2)$, ainsi que des constantes $\lambda_1, \lambda_2 \in K$ et le monôme $m = \frac{\delta}{ppcm(mt(P_1), mt(P_2))}$, pour obtenir :

$$tt(L_1)P_1 + tt(L_2)P_2 = \lambda_1 mct(P_2) n_1 P_1 + \lambda_2 mct(P_1) n_2 P_2 = \lambda_1 m Spoly P_1, P_2) + (\lambda_1 + \lambda_2) ct(P_1) m n_2 P_2$$

Par construction, le premier polynôme de cette dernière somme a son monôme de tête strictement plus petit que δ alors que le monôme de tête du second polynôme est exctement δ, à moins qu'il ne soit nul. On obtient ainsi une contradiction à la minimalité de k.

3.2 Algorithme de Buchberger basique

Du théorème ci-dessus, on déduit un premier algorithme pour calculer une base de Groebner :

Entrée : un ensemble fini G' de polynômes P_i non nuls, un ordre monomial $<$.
Sortie : une base de Groebner G pour le même idéal.

- 1. Initialiser G à G' et S à l'ensemble des paires d'éléments de G
- 2. Tant que S 'est pas vide, faire :
- a) Choisir une paire {P,P'} et la retirer de S
- b) Calculer Spoly(P,P') et le réduire par G
- c) Si le reste R est non nul, alors :
- i) Adjoindre à S toutes les paires {P,R} pour P dans G
- ii) Adjoindre R à G
- 3. Renvoyer G

Correction et terminaison de l'algorithme : Un invariant de cet algorithme est que l'ensemble G ne contient que les générateurs P_i initiaux et des recombinaisons finies de ceux-ci à coefficients polynômiaux : l'idéal engendré par G est donc constant. De plus, si l'algorithme termine, la sortie G réduit à

zéro chacun des S-polynômes de ses éléments pris deux à deux. le théorème précédent fournit donc la correction de l'algorithme.

Pour la terminaison, on remarque que la partie stable engendrée par les monômes de tête des éléments de G croît srictement à chaque adjonction dans G. En considérant les idéaux engendrés successivement par cette partie stable, on obtient une suite strictement décroissante d'idéaux, puisque l'idéal engendrépar une partie stable admet en tant qu'espace vectoriel sur K la base constituée exactement des monômes de la partie stable. Par noetherianité de A_n, cette suite d'idéaux ne peut être infinie et il ne peut donc y avoir qu'un nombre fini d'adjonctions dans G.

3.3 Optimisations

L'algorithme se termine donc toujours en renvoyant un résultat correct, mais en combien de temps ? En pratique on se rend compte que l'on perd beaucoup de temps à effectuer les réductions des S-polynômes, alors que celles-ci ne sont pas toujours fructueuses. Il nous faudrait donc un critère permettant de trancher si un calcul va être utile avant, critère dont le coût sera négligeable par rapport au calcul.

Monômes de tête premiers entre eux :
Lorsque les monômes de tête de P et Q (dans A_n) sont premiers entre eux, ce qui revient à dire que les indéterminées apparaissant dans l'un ne se retrouvent pas dans l'autre, alors Spoly(P,Q) se réduit à zéro par P et Q (ie (P,Q) est une base de Groebner de l'idéal engendré par P et Q).

Démonstration : On suppose sans perte de généralité les polynômes P et Q normalisés : ct(P)=ct(Q)=1. Ecrivant P=mt(P)+u, Q=mt(Q)+v, on obtient étant donnque ppcm(mt(P),mt(Q))=mt(P)mt(Q) :

$$Spoly(P,Q) = mt(Q)P - mt(P)Q$$

puis, après calculs

$$Spoly(P,Q) = uQ - vP$$

On vérifie alors que le degré total de Spoly(P,Q) est égal au maximum des degrés totaux de uQ et de vP. Ainsi Spoly(P,Q) sera réductible par P et Q, ce qui signifie que Spoly(P,Q) se réduit à zéro par G vu que P et Q sont dans G.
Il suffit de constater que les monômes de tête de uQ et vP sont distincts et ne peuvent ainsi pas s'annuler. En effet, si c'était le cas on aurait mt(u)mt(Q)=mt(v)mt(P),

puis mt(P) divise mt(u) car mt(P) et mt(Q) sont premiers entre eux, ce qui est absurde.

Un second critère très utile est le suivant (avec $G = \{g_1, ..., g_N\}$) :
Si $mt(g_k)$ divise $ppcm(mt(g_i), mt(g_j))$, et que les paires $(g_i, g_k), (g_k, g_j)$ ont déjà été introduites au cours de l'algorithme, alors il est inutile de traiter la paire (g_i, g_j).

Démonstration : Il suffit de montrer que l'idéal engendré par l'ensemble Z des S-polynômes des éléments de G est le même que celui engendré par Z-$\{Spoly(g_i, g_j)\}$. Pour cela, on vérifie que

$$Spoly(g_i, g_j) = \frac{ppcm(mt(g_i), mt(g_j))}{ppcm(mt(g_i), mt(g_k))} Spoly(g_i, g_k) - \frac{ppcm(mt(g_i), mt(g_j))}{ppcm(mt(g_j), mt(g_k))} Spoly(g_j, g_k)$$

Et le critère est justifié.

J'ai également utilisé deux stratégies dans les algorithmes : je réduis les S-polynômes en priorité par les polynômes les plus anciennement entrés dans la base en construction, car ceux-ci sont souvent de degré plus petit que les autres.
J'ordonne également les paires de polynômes dans l'ordre croissant de leur ppcm afin de ne pas faire exploser les dergés dès lors que c'est possible.

4 Présentation de l'implémentation

Maintenant que l'on a appris à calculer une base de Groebner, on peut se convaincre rapidement de l'utilité d'un programme faisant le calcul à notre place en essayant par exemple de trouver la base de Groebner réduite d'un idéal engendré par trois polynômes à trois variables non triviaux.

4.1 Structures de données

J'ai choisi de représenter un polynôme par un tableau de termes, car il faut pouvoir accéder rapidement à chaque terme lors des opérations que l'on fait dessus. On suprimera et ajoutera des éléments avec une complexité linéaire, mais ce défaut est partiellement compensé en reconstruisant entièrement un polynôme dans le cas de la soustraction (opération la plus basique après l'ajout de terme) ; ainsi on évite les défauts d'un algorithme qui insérerait un par un les opposés des termes de Q à P dans le calcul de P-Q.

On limite autant que possible la complexité de l'ajout de terme en maintenant un invariant sur toutes les opérations que l'on effectue : dans la soustraction, la multiplication, le calcul de S-polynôme et la réduction, les polynômes donnés en entrée sont supposés triés dans l'ordre décroissant et les résultats sont ordonnésde la même manière. Ainsi on peut s'arranger pour que chaque appel à addTerm se contente d'ajouter un terme en fin de tableau, cette opération se faisant en temps constant (complexité amortie) ; le seul cas où on est obligé de supprimer un terme dans le tableau est lorsque le terme ajouté est l'opposé d'un de ceux déjà présents, mais on peut trouver des astuces pour éviter ce cas, et on ne fait jamais d'insertion dans tabTermes (le tableau des termes d'u polynôme).

Exemple : L'utilisateur (qui n'est pas forcément au courant de l'ordre dans lequel les termes doivent être rangés) entre $P = X^2+Y^3, Q = X^2Y^2+Z, R = X^2 + Y + Z^5$ et demande le calcul d'une base de Groebner. Le programme souhaitant utiliser l'ordre grevlex commence par réordonner les termes de chaque polynôme pour obtenir $P = Y^3+X^2, Q = X^2Y^2+Z, R = Z^5+X^2+Y$ (complexité d'un tri : $Nlog(N)$ en nombre de comparaisons, où N est le nombre de termes d'un polynôme). Supposons que le programme ait besoin de calculer $P-R$: il commencer par créer un nouveau polynôme vide, auquel il ajoute d'abord le plus grand des termes de tête de P et R, c'est à dire $-Z^5$; ensuite il ajoute le plus grand terme entre $-X^2$ et Y^3 : c'est Y^3. A létape suivante il doit choisir entre ajouter $-X^2$ ou.. X^2 ; la première méthode serait alors d'ajouter d'abord X^2, puis $-X^2$, ce qui necessiterait d'effacer le dernier élement de tabTermes juste après l'avoir ajouté. C'est un peu bête de procéder ainsi, donc on choisi de vérifier l'égalité des termes dans la soustraction avant d'appeler la fonction addTerm.

Ainsi on ne fait rien à cette étape, et on se positionne sur le terme suivant dans Q (pour P c'est terminé), dont on ajoute l'opposé en fin de tableau. En conclusion on obtient $P - R = -Z^5 + Y^3 - Y$, polynôme ordonné en décroissant selon grevlex. Cet exemple illustr bien comment l'ordonnancement des termes est conservé tout au long de l'algorithme.

Un terme est quant à lui pleinement défini par son coefficient (qui est un élement de $F_p = \mathbb{Z}/p\mathbb{Z}$ avec p premier pour simplifier l'écriture du programme) et le vecteur des exposants de ses indéterminées, appartenant à $\mathbb{N}^n$ où n est le nombre d'indéterminées. Le nombre n (qui sera appelé la taille du terme) n'ayant pas à évoluer dans le temps, on utilise cette fois un simple pointeur en allouant juste l'espace mémoire nécessaire au stockage de n valeurs entières.

Aussi bien pour les polynômes que pour les termes, on définit des fonctions permettant d'accéder et de modifier l'état de l'objet quand cela est nécessaire. On obtient donc finalement le code suivant pour la réduction d'un polynôme (dont les termes sont triés) par un ensemble de polynômes, en réduisant en priorité par les polynômes les plus anciennement introduits dans la base en construction, au motif que ceux-ci sont souvent de degré plus petit que ceux calculés plus récemment, ce qui permet des calculs moins lourds :

```
// réduction de *this (polynôme courant) par tabPol.
bool red=true ; // indique si une réduction a été effectuée dans la boucle
// dans le cas où on réduit le polynôme nul, on renvoie simplement
// l'argument de la réduction (mot clé this)
if (tabTermes.size()==0) return *this ;
// sinon, on initialise le résultat au polynôme courant PolPlusVarLex
pol_res=PolPlusVarLex(*this) ;
// termes de tête temporaires et quotient de division entre termes
TermLex HTi, HTres, divi ;

while (red) { // on s'arrete donc lorsqu'on fait un parcours sans reduction
  // red vaudra true seulement si un polynôme peut réduire pol_res
  red=false ;
  // HTres initialisé au terme de tête du résultat intermédiaire
  HTres=pol_res[0] ;
  for (int i=0 ; i<(int)tabPol.size() ; i++) {
    if (tabPol[i].getSize()≠0) {
      HTi=tabPol[i][0] ;
      // tentative de reduction par le monome de tete de tabPol[i]
      divi=HTi.divise(HTres) ;
      if (divi.getCoeff()≠0) {
        divi.setCoeff(HTres.getCoeff()/HTi.getCoeff()) ;
        pol_res=pol_res-tabPol[i]*divi ;
        if (pol_res.getSize()==0) return pol_res ;
        HTres=pol_res[0] ;
        red=true ;
      }
    }
  }
}

return pol_res ;
```

Où tabTermes est un tableau d'objets de type TermLex, représentant chacun un terme en les indéterminées $X_1, ..., X_n$
std : :vector<TermLex> tabTermes ;

Comme indiqué plus haut, on travaille dans F_p et il est donc agréable de disposer d'une classe encapsulant la description ainsi que les opérations élémentaires pouvant s'effectuer sur F_p.

Toutes les opérations de comparaisons, d'addition et de soustraction n'ont pas besoin d'être détaillées ici, seule l'inversion modulaire est intéressante : on utilise la relation matricielle

$$\begin{pmatrix} b \\ r \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & q \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

où a=bq+r, |r|<b, pour obtenir par récurrence immédiate

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \prod_i \begin{pmatrix} 0 & 1 \\ 1 & q_i \end{pmatrix} \begin{pmatrix} p \\ x \end{pmatrix}$$

dans le cas où a=p premier, b=x entier dans [0,..,p-1] (les q_i étant les quotients successifs). D'où le code suivant, dans lequel la première ligne de la matrice est notée (a1 a2) et la seconde (a3 a4) au cours des itérations :

```
// on applique l'algorithme d'Euclide pour trouver une relation xu+pv=1 ;
// alors x^{-1} = u[p] ; ici value est x
int a1=0,a2=1,a3=1,a4=p/value ; // les coefficients actualises de la matrice
int q=value,r=abs(p%value) ; // quotients et restes
int aux,aux2 ; // variables temporaires

while (r>0) {
   a1=a3 ;
   aux2=a2 ;
   a2=a4 ;
   a3* =-a4 ;
   a4=aux2+a4*(q/r) ;
   aux=q%r ;
   q=r ;
   r=abs(aux) ;
}

return abs(a2%p) ;
```

4.2 Opérations sur les polynômes

On présente ici respectivement le code pour le calcul de la soustraction (*this-pol) ainsi que du S-polynôme (Spoly(*this,pol)).

Soustraction :

```
PolPlusVarLex pol_res=PolPlusVarLex();
int i=0;
int j=0;
while (i<(int)tabTermes.size() && j<(int)pol.getSize()) {
  if (tabTermes[i].infLex(pol[j])) {
    pol_res.addTerm(-pol[j]);
    j++;
  }
  else {
    pol_res.addTerm(tabTermes[i]);
    i++;
  }
}
if (i<(int)tabTermes.size())
  for (j=i; j<(int)tabTermes.size(); j++) pol_res.addTerm(tabTermes[j]);
  else for (i=j; i<pol.getSize(); i++) pol_res.addTerm(-pol[i]);
return pol_res;
```

S-polynôme :

```
if (tabTermes.size()==0 || pol.getSize()==0) return PolPlusVarLex();
  TermLex term_ppcm=tabTermes[0].ppcm(pol[0]); // calcul du ppcm
des monomes de tete utile pour la suite
  TermLex term1=TermLex(tabTermes[0].getSize()),
term2=TermLex(tabTermes[0].getSize());

// divisions du ppcm par HT(f), HT(g)
for (int i=0; i<tabTermes[0].getSize(); i++) {
  term1.addElt(term_ppcm[i]-tabTermes[0][i]);
  term2.addElt(term_ppcm[i]-pol[0][i]);
}

// multiplications par HC(f), HC(g)
term1.setCoeff(pol[0].getCoeff());
term2.setCoeff(tabTermes[0].getCoeff());
```

PolPlusVarLex pol_res=PolPlusVarLex((*this)*term1-pol*term2) ;
return pol_res ;

4.3 Optimisations futures

Les premières optimisations ont consisté à passer tous les arguments par référence constante afin qu'ils ne soient ni recopiés ni modifiés ; la vitesse de l'algorithme augmente alors assez. J'ai aussi utilisé les optimisations proposées par g++ au niveau du code généré (-O3, -funroll-loops..etc).

J'ai utilisé des vecteurs pour décrire les polynômes mais je ne tire quasiment aucun profit de l'algèbre linéaire qui permettrait de rendre considérablement plus efficaces les réductions ainsi que les choix de paires de polynômes dont on calcule les S-polynômes, comme dans les algorithmes F4 et F5 de J-C. Faugère.

5 Applications réalisées

5.1 Appartenance à un idéal, égalité de deux idéaux, test de principalité

On a vu qu'une base de Groebner G d'un idéal I a la propriété de réduire à zéro tout polynôme de I. Réciproquement, si un polynôme P de A_n se réduit à zéro par G, alors P s'écrit $P = \sum_{i=1}^{k} \alpha_i Q_i$ où les Q_i sont des éléments de G. On en déduit qu'alors P appartient à I.

Pour tester si un polynôme appartient à un idéal engendré par un système de générateurs, il suffit donc de calculer un base de Groebner de cet idéal, puis de réduire le polynôme par la base trouvée : si on obtient zéro, alors le polynôme est dans l'idéal ; sinon, il n'y est pas.
Le code prenant en paramètre un polnôme et un système de géérateurs est alors trivial à écrire.

En ce qui concerne l'égalié de deux idéaux I_1 et I_2 donnés par des systèmes de générateurs, on calcule d'abord les bases de Groebner réduites de I_1 et I_2, notées G_1 et G_2 respectivement ; puis celles-ci étant uniques, il suffit de vérifier si elles sont égales. Algorithmiquement, on vérifie simplement que chaque polynôme de G_1 est dans G_2 en les prenant un par un.

Pour vérifier qu'un idéal engendré par un système de générateurs est principal, il suffit de vérifier si la base de Groebner réduite contient un unique polynôme.

5.2 Mise sous forme implicite d'une paramétrisation

Une paramétrisation étant donnée par un système de la forme
$X_1 = \frac{P_1(T_1,...,T_m)}{Q_1(T_1,...,T_m)}$
...
$X_p = \frac{P_p(T_1,...,T_m)}{Q_p(T_1,...,T_m)}$
où les P_i et Q_i sont dans A_n et $0 \leq p \leq n$, le problème est de trouver les relations les plus précises possibles reliant les X_i entre eux.

On réalise cela en ajoutant une équation au système avec une inconnue supplémentaire notée U, afin que les dénominateurs ne s'annulent pas : $U \prod i = 1pQ_i - 1 = 0$.

On calcule donc une base de Groebner de l'idéal engendré par les polynômes suivants, pour l'ordre lexicographique $lex(U, T_1, ..., T_m, X_1, ..., X_p)$:

$$U \prod_{i=1}^{p} Q_i - 1$$

$$X_1 Q_1 - P_1$$

$$...$$

$$X_p Q_p - P_p$$

Finalement, on a éliminé autant que possibles les m+1 premières indéterminées, et on renvoie simplement les équations n'impliquant que les X_i. S'il n'y en a pas, cela signifie qu'il n'y a pas d'équation implicite polynômiale décriant la paramétrisation.
Le code reprend les étapes précédentes en les adaptant aux stuctures de données utilisées :

Un entier n est initialisé avec le nombre de relations données.
a) calcul du nombre maximum d'indeterminees rencontrees dans les numérateurs et dénominateurs
b) mise a jour des polynomes pour que tous les monomes aient le même nombre d'indéterminées (m)
c) mise a jour des polynomes pour que tous les monomes soient de taille n+m+1, l'indéterminée ajoutée est U

d) copie de sauvegarde des denominateurs, utiles pour le calcul $U \prod i = 1 p Q_i - 1 = 0$
e) actualisation des denominateurs : $\forall i, Q_i \leftarrow Q_i X_{i+m+1}$
f) initialisation des polynomes necessaires au calcul
g) calcul
h) suppression des polynomes faisant intervenir U ou les T_i, c'est à dire $\{X_i, i \in [1, ...m+1]\}$
i) décalage des indeterminees pour ramener tous les polynômes qui restent à n indéterminées
j) affichage du resultat (ordonne selon lex)

Justification :

L'ordre choisit élimine le plus d'indéterminées possibles de U à T_n, donc on obtient bien le plus de relations possibles décrivant la paramétrisation.

5.3 Résolution d'un système polynômial

On a besoin pour cette application d'implémenter la division euclidienne pour des polynômes à une variable nécessaire au calcul de pgcd, ainsi qu'une fonction substituant un élément de F_p à une indéterminée.
La division euclidienne n'étant qu'une version simplifiéee de la réduction présentée précédemment, seule un détail de la substitution présente un intérêt ici ; on utilise une méthode "diviser pour régner" pour calculer les puissances des indéterminées, comme ceci :

calcul de $a^b[p]$
if (mem=1 ; mem<b ; mem∗ =2) **return puiss(a,b/2) ;**
else return (puiss(a,b/2)*a)%p ;

Voici les différentes étapes de l'algorithme :

Un vecteur solution initialisé à la liste vide, et i à n.
a) si $i \leq 0$, arrêter l'algorithme et renvoyer la liste en cours de construction
b) si $G \cap K[X_i]$ est vide, faire $i \leftarrow i - 1$, et lancer p fois l'algorithme sur les listes $l \leftarrow l + k$ où $k \in [0, .., p-1]$ (ceci correspond au cas où il y a une infinité de solutions dans le cas d'un corps infini) ; on va donc p fois en c). Sinon, aller en c) sans changer i
c) calcul du pgcd des polynômes de $G \cap K[X_i]$
d) resolution pgcd=0

e) $i \leftarrow i-1$, et lancer l'algorithme (ie : retourner en a)) sur chaque liste obtenue par concaténation : $l \leftarrow l+k$ où k est solution de l'équation résolue en d)

La justification se fait en remarquant que résoudre un systéme d'équations en une indéterminée revient à annuler le pgcd des polynômes du systéme, et que G engendrant le même idéal que les polynômes générateurs de départ le sous-ensemble de K définie par $\forall P \in IP(x) = 0$ est égal à $\{x \in K | \forall P \in GP(x) = 0$.

5.4 Démonstration de quelques théorèmes géométriques

Tout théorème de géométrie euclidienne pouvant se ramener à un calcul sur des polynômes, on montre ici à l'aide des bases de Groebner que les trois bissectrices d'un triangle sont concourantes. On montre ensuite le même résultat sur les médiatrices, médianes et hauteurs (c'est alors plus simple).

On se ramène à un triangle dans ${\mathbb{R}_+}^2$, dont un côté est le segment d'origine (0,0) arrivant en (1,0), un autre est le segment d'origine (0,0) arrivant en (X_1, X_2), et le dernier côté part donc de (1,0) pour arriver en (X_1, X_2). On calcule alors les équations des droites en introduisant de nouvelles indéterminées correspondant aux inconnues des équations.
X_1 et X_2 étant les paramètres strictement positifs du sommet du triangle, on obtient après quelques calculs les équations suivantes :

$$X_3 - X_5 = 0$$
$$X_4 - X_6 = 0$$
$$X_5 - X_7 = 0$$
$$X_6 - X_8 = 0,$$

correspondant au fait que les droites doivent se couper.

$$-2X_1X_2X_3X_4 + X_2^2X_3^2 - X_2^2X_4^2 = 0$$
$$-2X_1X_2X_5X_6 + 2X_1X_2X_6 + X_2^2X_5^2 - 2X_2^2X_5 + X_2^2 + 2X_2X_5X_6 - 2X_2X_6 = 0$$
$$2X_1^3X_2X_8 - 2X_1^2X_2^2X_7 - X_1^2X_2^2X_8^2 + X_1^2X_2^2 - 2X_1^2X_2X_7X_8 - 2X_1^2X_2X_8$$
$$+2X_1X_2^3X_8 + 2X_1X_2^2X_7^2 - 2X_1X_2^2X_8^2 + 2X_1X_2X_7X_8 - 2X_2^4X_7 + X_2^4 + 2X_2^3X_7X_8$$
$$- 2X_2^3X_8 - X_2X_7^2 + 2X_2^2X_8^2,$$

correspondant aux trois équations de droites.

La méthode utilisée est la suivante : on cherch à éliminer les paramètres $X_3, ..., X_8$ en les exprimant de manière unique en fonction de X_1 et X_2, ce qui montrera qu'il existe une (unique) solution pour chaque couple (X_1, X_2) d'éléments de $\mathbb{R}_+$, et donc pour tout triangle du plan (par rotation et homothéties). Ceci se fait en calculant une base de Groebner des polynômes décrivant les trois droites pour l'ordre $lex(X_8, ..., X_1)$; on arrive normalement à des équations de la forme $X_3 = f_3(X_1, X_2), X_4 = f_4(X_1, X_2, X_3), ..etc$, d'où l'existence du point d'capion par résolution du système triangulaire obtenu.

L'algorithme se termine très rapidement sur tous les problèmes sauf les bissectrices. Pour ce dernier, il boucle trop longtemps pour avoir un résultat en temps raisonnable.

Bibliographie :

Frédéric Chyzak - *Bases de Groebner, algorithme de Buchberger et applications*
Niels Lauritzen - *Concrete abstract algebra*
David Cox/John Little/Donal O'Shea - *Ideals, Varieties and Algorithms*
Bruno Buchberger - *Groebner Bases : A Short Introduction for Systems Theorists*
Fabio Berto - *Systèmes non lineaires et Optimisation*