Formes normales généralisées

Philippe Trébuchet, INRIA, Spaces/Lip6 Calfor

Équipe CALFOR/INRIA Projet SPACES
Université Pierre et Marie Curie
4 place Jussieu
75252 Paris Cedex 05

10 février 2004

Notations

Notations

- $\mathbb{K}$ un corps.
- $\mathbb{K}[x_1, \ldots, x_n]$ l'anneau des polynômes en n variables.
- $I = (f_1, \ldots, f_s)$ un idéal définissant une variété $\mathcal{V}$.
- $A = \mathbb{K}[x_1, \ldots, x_n]/I$ l'algèbre quotient

Résolution de systèmes

Cas où I est de dimension > 0.

Résolution de systèmes

Cas où I est de dimension > 0.
Pas de consensus sur la définition de “résoudre”

Résolution de systèmes

Cas où I est de dimension > 0.
Pas de consensus sur la définition de "résoudre"

Cas où I est de dimension 0.
on notera $\zeta_1, \ldots, \zeta_k$ les solutions.

.

Résoudre c'est permettre d'accéder facilement à une approximation des coordonnées des solutions

Résolution de systèmes

Cas où I est de dimension > 0.
Pas de consensus sur la définition de "résoudre"

Cas où I est de dimension 0.
on notera $\zeta_1, \ldots, \zeta_k$ les solutions.

.
Résoudre c'est permettre d'accéder facilement à une approximation des coordonnées des solutions

- Soit directement en renvoyant une telle approximation. (précision finie)

Résolution de systèmes

Cas où I est de dimension > 0.
Pas de consensus sur la définition de "résoudre"

Cas où I est de dimension 0.
on notera $\zeta_1, \ldots, \zeta_k$ les solutions.

.
Résoudre c'est permettre d'accéder facilement à une approximation des coordonnées des solutions

- Soit directement en renvoyant une telle approximation. (précision finie)
- Soit en calculant paramétrisation rationnelle de $\mathcal{V}$.

Résolution de systèmes

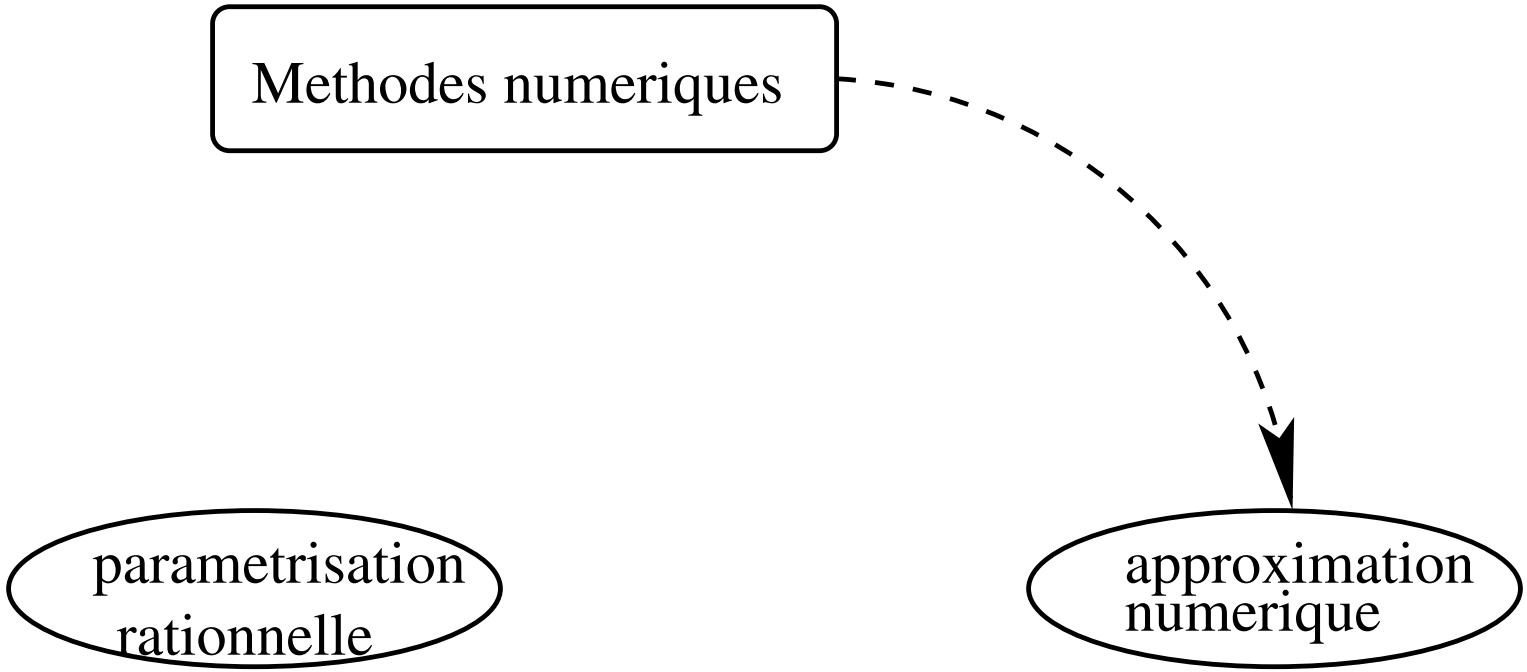
Cas où I est de dimension > 0.
Pas de consensus sur la définition de "résoudre"

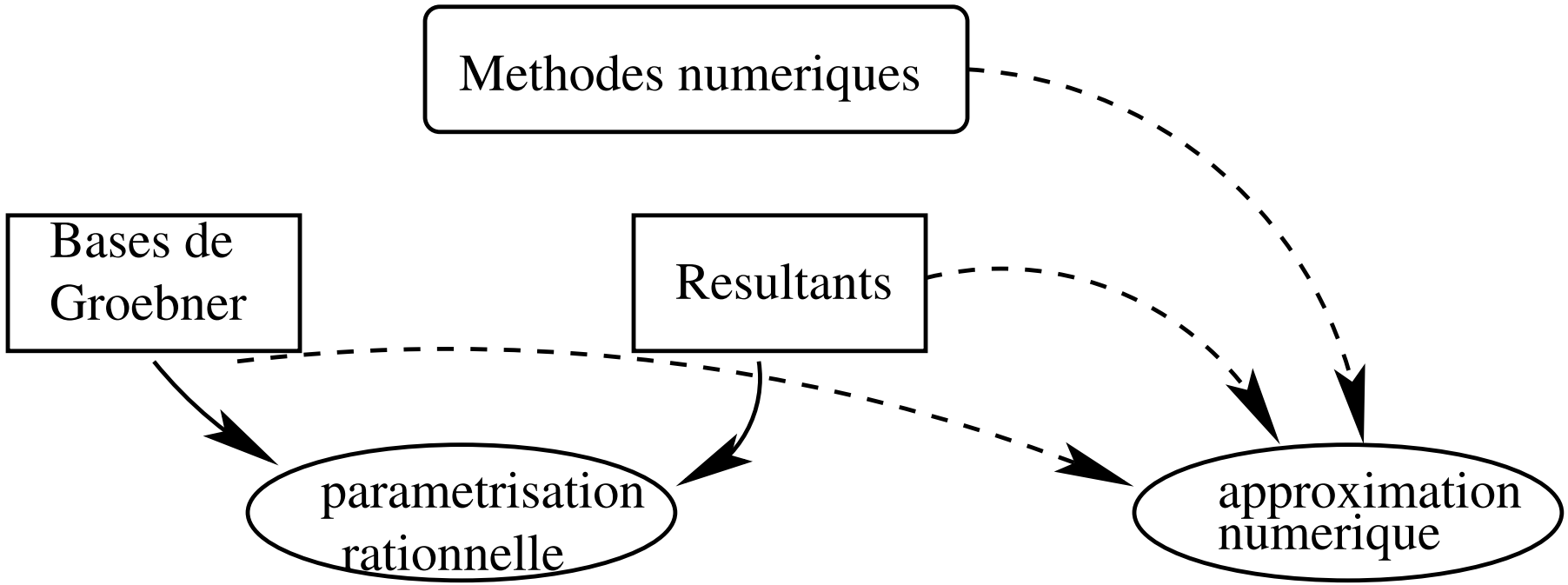
Cas où I est de dimension 0.
on notera $\zeta_1, \ldots, \zeta_k$ les solutions.

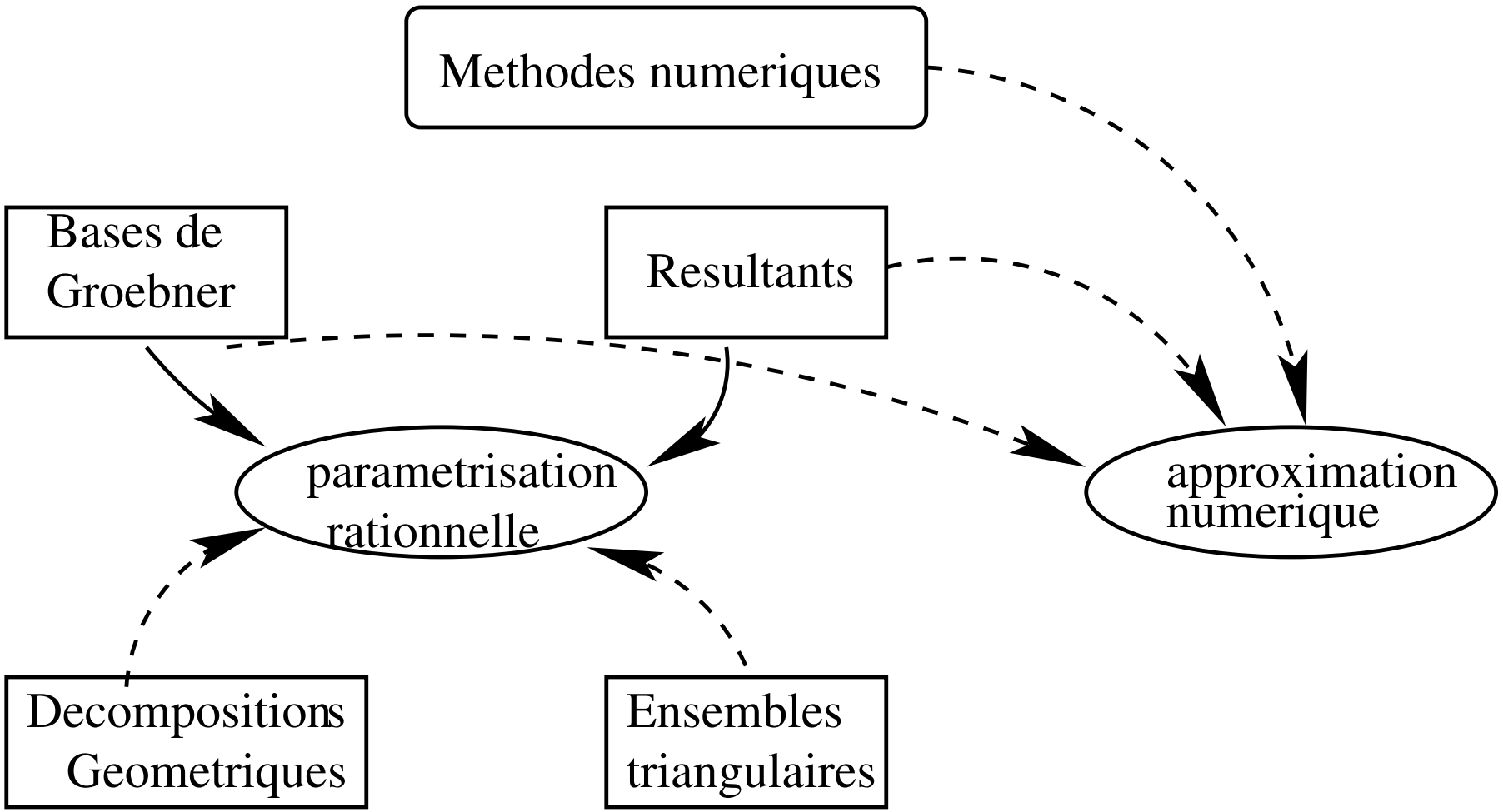
.
Résoudre c'est permettre d'accéder facilement à une approximation des coordonnées des solutions

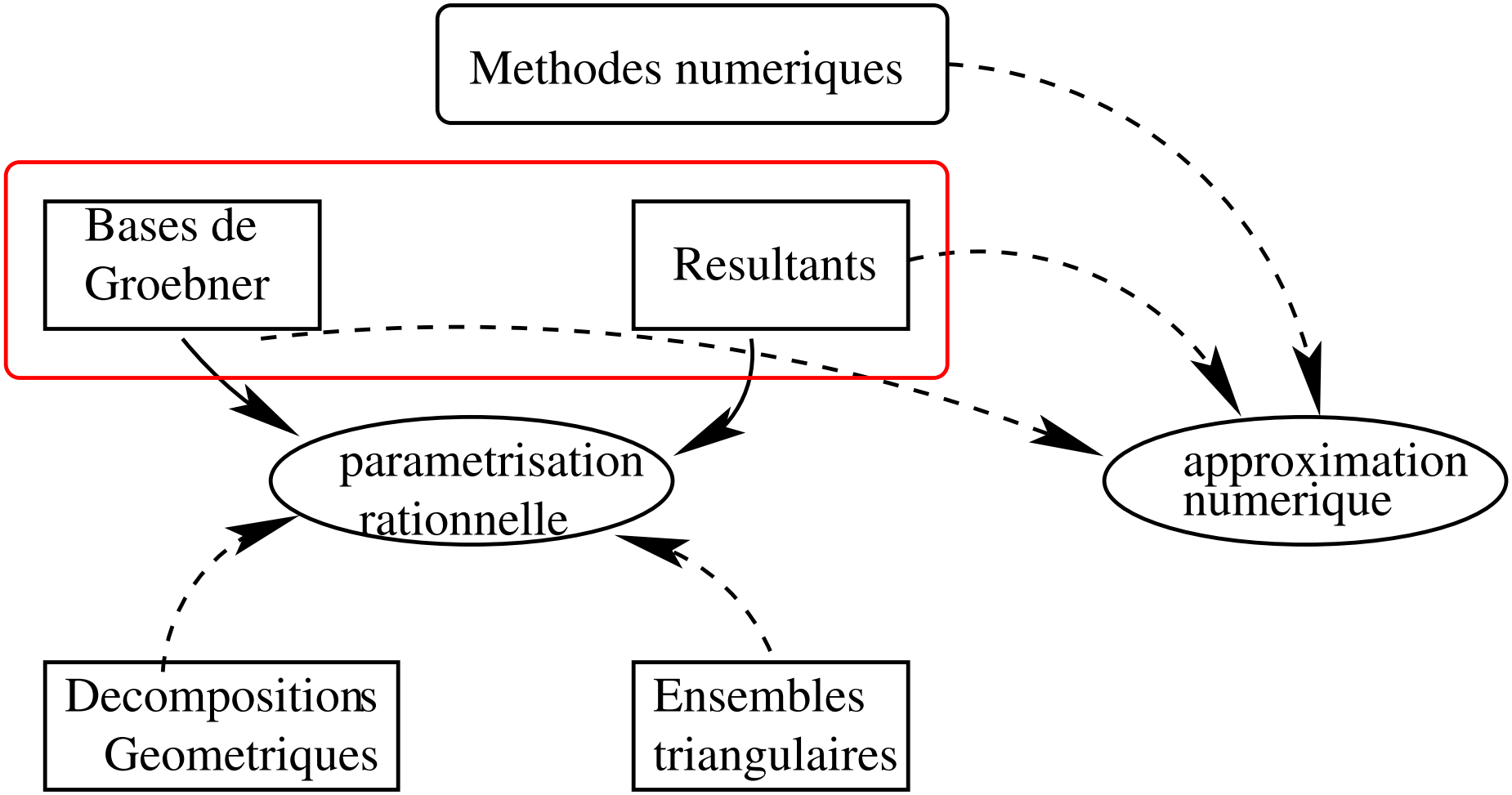
- Soit directement en renvoyant une telle approximation. (précision finie)
- Soit en calculant paramétrisation rationnelle de $\mathcal{V}$.
- soit en renvoyant décomposition de $\mathcal{V}$ en ensembles triangulaires.


parametrisation
rationnelle
approximation
numerique


Methodes numeriques
parametrisation rationnelle
approximation numerique


Methodes numeriques
Bases de Groebner
Resultants
parametrisation rationnelle
approximation numerique


Methodes numeriques
Bases de Groebner
Resultants
parametrisation rationnelle
approximation numerique
Decompositions Geometriques
Ensembles triangulaires


Methodes numeriques
Bases de Groebner
Resultants
parametrisation rationnelle
approximation numerique
Decompositions Geometriques
Ensembles triangulaires

Contributions

- Généralisation des calculs de Bases de Gröbner.
- Unification avec la théorie des résultants.
- Possibilité de prendre en compte certaines notions de stabilité.
- Implantations efficaces (du niveau du couple Gb/Rs).
- Applications pratiques.

Plan

Calculer les solutions.

Représentations de l'algèbre quotient A.

Calculer de meilleures représentations.

Calculer les solutions

Passer de A aux ζ_i

Théorème : *[Stickelberger]Si* $\mathcal{V} = \zeta_1, \ldots, \zeta_k$, $p \in \mathbb{K}[x_1, \ldots, x_n]$.
On appellera multiplication par p, $\mathcal{M}_p$ *l'opérateur* $A \to A$. *L'opérateur* $\mathcal{M}_p$ *a les* $f \to fp$
propriétés suivantes:

- *Les valeurs propres de* $\mathcal{M}_p$ *sont les nombres* $p(\zeta_1), \ldots, p(\zeta_k)$ *(comptés avec multiplicités).*

- *Les vecteurs propres communs à tous les* $\mathcal{M}_p^t$ *sont les évaluations en les* ζ_i.

Remarquons qu'il est alors facile de calculer la forme de Chow de I :

$$Det(\mathcal{M}_{u_0+u_1x_1+\cdots+u_nx_n})$$

Passer de A aux ζ_i (suite)

Théorème : *[Roy et al.] Soient,* $\mathbb{K} = \mathbb{R}$, $h \in \mathbb{R}[\mathbf{x}_1, \ldots, x_n]$ *et* Q_h *la forme quadratique* $Q_h : A \to A$ $\quad p \to Tr(hp^2)$ *. Alors on a les deux propriétés suivantes :*

- *Le nombre de racines complexes* ζ_i *où* h *ne s'annule pas,* $h(\zeta_i) \neq 0$ *est le rang de* Q_h*.*

- *Le nombre de racines réelles* ζ_i *telles que* $h(\zeta_i) > 0$ *et telles que* $h(\zeta_i) < 0$ *est la signature de* Q_h*.*

Passer de A aux ζ_i (suite et fin)

Théorème : *[Rouillier] Soit* $u \in \mathbb{K}[x_1, \ldots, x_n]$, *on définit :*

- $f_u(T) = \Pi_{i=1..k}(T - u(\zeta_i))^{\mu_i}$
- $g_0(T) = \Sigma_{i=1..k}\mu_i\Pi_{j\neq i}(T - u(\zeta_j))$
- $g_l(T) = \Sigma_{i=1..k}\mu_i\zeta_l.i\Pi_{j\neq i}(T - u(\zeta_j))$

Passer de A aux ζ_i (suite et fin)

Théorème : *[Rouillier] Soit* $u \in \mathbb{K}[x_1, \ldots, x_n]$, *on définit :*

- $f_u(T) = \Pi_{i=1..k}(T - u(\zeta_i))^{\mu_i}$
- $g_0(T) = \Sigma_{i=1..k}\mu_i\Pi_{j\neq i}(T - u(\zeta_j))$
- $g_l(T) = \Sigma_{i=1..k}\mu_i\zeta_l.i\Pi_{j\neq i}(T - u(\zeta_j))$

$$f_u(\zeta) = 0 \rightarrow \zeta_i = \begin{pmatrix} g_1(\zeta)/g_0(\zeta) \\ \vdots \\ g_n(\zeta)/g_0(\zeta) \end{pmatrix}$$

Si u *sépare les* ζ_i *alors la famille précédente constitue une représentation univariée rationnelle.*

Passer de A aux ζ_i (suite et fin)

Théorème : *[Rouillier] Soit $u \in \mathbb{K}[x_1, \ldots, x_n]$, on définit :*

- $f_u(T) = \Pi_{i=1..k}(T - u(\zeta_i))^{\mu_i}$
- $g_0(T) = \Sigma_{i=1..k}\mu_i\Pi_{j\neq i}(T - u(\zeta_j))$
- $g_l(T) = \Sigma_{i=1..k}\mu_i\zeta_l.i\Pi_{j\neq i}(T - u(\zeta_j))$

$$f_u(\zeta) = 0 \rightarrow \zeta_i = \begin{pmatrix} g_1(\zeta)/g_0(\zeta) \\ \vdots \\ g_n(\zeta)/g_0(\zeta) \end{pmatrix}$$

Si u sépare les ζ_i alors la famille précédente constitue une représentation univariée rationnelle.

Proposition : *Un élément séparant peut être trouvé dans la famille :*

$$\mathcal{U} = \{x_1 + jx_2 + \cdots + j^{n-1}x_n, j = 0 \ldots nk(k-1)/2\}$$

On a aussi :

$$u \text{ sépare les } \zeta_i \Leftrightarrow deg(minpol((M_u))) == dim(A)$$

Représentation de l'algèbre quotient A

Travailler dans l'algèbre quotient A

Unicité de la représentation des éléments d'une même classe d'équivalence
Une telle représentation est appelée **Forme Normale** des éléments de la classe.

Travailler dans l'algèbre quotient A

Unicité de la représentation des éléments d'une même classe d'équivalence
Une telle représentation est appelée **Forme Normale** des éléments de la classe.

- $\mathcal{V}$ est composée de points $\Rightarrow$ A $\mathbb{K}$-espace vectoriel de dimension finie.
- On se ramène à chercher une base **monomiale** de A (notée B).

  $\mathtt{M}_k$ désignera l'ensemble de monômes de degré inférieur à k

Proposition : *Si on se donne une base B de A, alors on a :*

$$\mathbb{K}[x_1, \ldots, x_n] = \langle B \rangle \oplus \langle I \rangle$$

($\langle I \rangle$ désigne le $\mathbb{K}$-espace vectoriel engendré par I)

L'idée de Macaulay

Si B est donné, calculer une expression canonique d'un polynôme $p \in \mathbb{K}[x_1, \ldots, x_n]$ dans B est une chose facile:

L'idée de Macaulay

Si B est donné, calculer une expression canonique d'un polynôme $p \in \mathbb{K}[x_1, \ldots, x_n]$ dans B est une chose facile:

Algorithme 1. INPUT: $p \in \mathbb{K}[x_1, \ldots, x_n]$
$I = (f_1, \ldots, f_s)$
B *une base monomiale de* A
OUTPUT *la représentation de* p *dans* A.

- *k=0, reduced=*`false`
- *while reduced==false*
  - ⋆ *construire la matrice* $Mat_k(p)$

L'idée de Macaulay

Si B est donné, calculer une expression canonique d'un polynôme $p \in \mathbb{K}[x_1, \ldots, x_n]$ dans B est une chose facile:

Algorithme 1. INPUT: $p \in \mathbb{K}[x_1, \ldots, x_n]$
$I = (f_1, \ldots, f_s)$
B *une base monomiale de* A
OUTPUT *la représentation de* p *dans* A.

$$Mat_k(p) = \begin{array}{c} \begin{matrix} B^c & B \end{matrix} \\ \begin{pmatrix} \mathtt{M}_k f_1 \\ \vdots \\ \mathtt{M}_k f_s \\ p \end{pmatrix} \end{array}$$

- *k=0, reduced=*`false`
- *while reduced==false*
  - ⋆ *construire la matrice* $Mat_k(p)$

L'idée de Macaulay

Si B est donné, calculer une expression canonique d'un polynôme $p \in \mathbb{K}[x_1, \ldots, x_n]$ dans B est une chose facile:

Algorithme 1. INPUT: $p \in \mathbb{K}[x_1, \ldots, x_n]$
$I = (f_1, \ldots, f_s)$
B *une base monomiale de* A
OUTPUT *la représentation de* p *dans* A.

$$Mat_k(p) = \begin{array}{c} \begin{matrix} B^c & B \end{matrix} \\ \begin{pmatrix} \mathtt{M}_k f_1 \\ \vdots \\ \mathtt{M}_k f_s \\ p \end{pmatrix} \end{array}$$

- *k=0, reduced=*`false`
- *while reduced==false*
  - ⋆ *construire la matrice* $Mat_k(p)$
  - ⋆ *Ramener* $Mat_k(p)$ *à une forme échelon sans permuter les lignes*

L'idée de Macaulay

Si B est donné, calculer une expression canonique d'un polynôme $p \in \mathbb{K}[x_1, \ldots, x_n]$ dans B est une chose facile:

Algorithme 1. INPUT: $p \in \mathbb{K}[x_1, \ldots, x_n]$
$I = (f_1, \ldots, f_s)$
B *une base monomiale de* A
OUTPUT *la représentation de* p *dans* A.

$$Mat_k(p) = \begin{array}{c} \begin{matrix} B^c & B \end{matrix} \\ \begin{pmatrix} \mathtt{M}_k f_1 \\ \vdots \\ \mathtt{M}_k f_s \\ p \end{pmatrix} \end{array}$$

- *k=0, reduced=*`false`
- *while reduced==false*
  - ⋆ *construire la matrice* $Mat_k(p)$
  - ⋆ *Ramener* $Mat_k(p)$ *à une forme échelon sans permuter les lignes*
  - ⋆ *Si la dernière ligne n'a pas de coefficients non nuls en dehors des colonnes correspondant à* B
    reduced=`true`
- *retourner la dernière ligne de la matrice sous forme échelon.*

La construction de Macaulay

Si on arrive à caractériser des bases on pourra résoudre

La construction de Macaulay

Si les f_i sont $génériques$ de degré d_i alors on a :

- L'ensemble $E_0 = \{x_1^{\alpha_1} \dots x_n^{\alpha_n}, \alpha_i < d_i\}$ est une base of A.
- Les ensembles $E_i = \{x_1^{\alpha_1} \dots x_n^{\alpha_n}, \Sigma_{j=1..n}\alpha_j \leq \Sigma_{j=1..n}(d_j - 1) + 1, \ \forall n \geq j \geq i, \ \alpha_j < d_j\}$ sont les monômes par lesquels il faut multiplier les f_i

La construction de Macaulay

Si les f_i sont $génériques$ de degré d_i alors on a :

- L'ensemble $E_0 = \{x_1^{\alpha_1} \dots x_n^{\alpha_n}, \alpha_i < d_i\}$ est une base of A.
- Les ensembles $E_i = \{x_1^{\alpha_1} \dots x_n^{\alpha_n}, \Sigma_{j=1..n}\alpha_j \leq \Sigma_{j=1..n}(d_j - 1) + 1, \ \forall n \geq j \geq i, \ \alpha_j < d_j\}$ sont les monômes par lesquels il faut multiplier les f_i

Algorithme 2. INPUT: $f_0 \in \mathbb{K}[x_1, \dots, x_n]$, *dont on cherche l'opérateur de multiplication*
$f_1, \dots, f_s$ *des polynômes* $générique$
OUTPUT: *La matrice de l'opérateur de multiplication par* f_0 *dans* A

- *construire la matrice :*

$$\begin{array}{c} \begin{matrix} E_0 f_0 & E_1 f_1 & \dots & E_n f_n \end{matrix} \\ \left(\begin{array}{c|c} \mathtt{A} & \mathtt{C} \\ \hline \mathtt{B} & \mathtt{D} \end{array} \right) \end{array}$$

- *retourner le* $complément\ de\ Schur = A - CD^{-1}B$

Les méthodes matricielles

Étude d'un endomorphisme à la Sylvester:

$$\begin{array}{ccc} < E_0 > \times < E_1 > \times \cdots \times < E_s > & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n] \\ (p_0, p_1, \ldots, p_s) & \longrightarrow & \Sigma_{i=0..s} p_i f_i \end{array}$$

- E_0 doit être une base de A.
- E_i doit être assez gros pour que le complément de Schur du bloc `A` donne l'opérateur de multiplication par f_0.

Ex: Résultant creux (J. Canny, I. Emiris. . .), Résultant résiduel (L. Buse, J.P. Jouannolou. . .),

Les bases de Gröbner

Idée: Faire en multivariée un analogue de la division euclidienne

- Définir une notion de plus grand monôme $lm(p)$, pour un polynôme p

Les bases de Gröbner

Idée: Faire en multivariée un analogue de la division euclidienne

- Définir une notion de plus grand monôme $lm(p)$, pour un polynôme $p \Rightarrow$ introduction d'un ordre monomial.

Les bases de Gröbner

Idée: Faire en multivariée un analogue de la division euclidienne

- Définir une notion de plus grand monôme $lm(p)$, pour un polynôme $p \Rightarrow$ introduction d'un ordre monomial.

- Définir $\overline{p}^G$ la division d'un polynôme p par un ensemble de polynômes G

Les bases de Gröbner

Idée: Faire en multivariée un analogue de la division euclidienne

- Définir une notion de plus grand monôme $lm(p)$, pour un polynôme $p \Rightarrow$ introduction d'un ordre monomial.
- Définir $\overline{p}^G$ la division d'un polynôme p par un ensemble de polynômes G
- Vérifier que la division est canonique

Les bases de Gröbner

Idée: Faire en multivariée un analogue de la division euclidienne

- Définir une notion de plus grand monôme $lm(p)$, pour un polynôme $p \Rightarrow$ introduction d'un ordre monomial.
- Définir $\overline{p}^G$ la division d'un polynôme p par un ensemble de polynômes G
- Vérifier que la division est canonique $\Rightarrow$ Vérifier que les S-polynômes

$$S(p,q) = \frac{ppcm(lm(p), lm(q))}{lm(p)} p - \frac{ppcm(lm(p), lm(q))}{lm(q)} q$$

se réduisent à 0.

Les bases de Gröbner

Algorithme 3. INPUT : $f_1, \ldots, f_s$ *engendrant* I*.(de n'importe quelle dimension)*
Un ordre monomial admissible γ
OUTPUT : *Une représentation de* A

- $G = \{f_1, \ldots, f_s\}$
- **repeat**
  - $G' = G$
  - *for all pair* $(p, q),\ p, q \in G'$ *do*
    - $S = \overline{S(p,q)}^{G'}$
    - *if* $S \neq 0$ *then* $G = G \cup S$
- **until** $G' == G$
- *return* G

Les bases de Gröbner (Faugère, Lazard, Lombardi)

On remarque alors que :

- On peut remplacer l'algèbre polynomial par de l'algèbre linéaire

Les bases de Gröbner (Faugère, Lazard, Lombardi)

On remarque alors que :

- On peut remplacer l'algèbre polynomial par de l'algèbre linéaire

En effet dans les calculs de bases de Gröbner :

- On voit les polynômes de I comme relations de dépendance linéaire entre les monômes de leur support.

- On peut remplacer les réductions de S-polynômes par des échelonisations de matrices.

Réduire $S(p, q)$

Les bases de Gröbner (Faugère, Lazard, Lombardi)

On remarque alors que :

- On peut remplacer l'algèbre polynomial par de l'algèbre linéaire

En effet dans les calculs de bases de Gröbner :

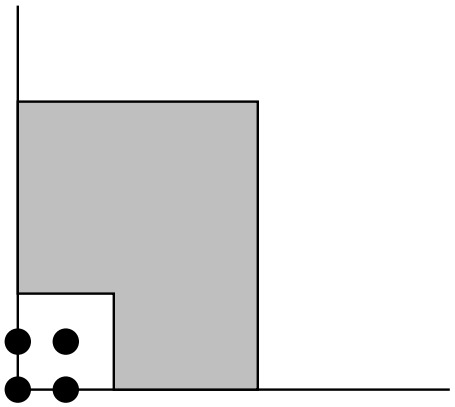
- On voit les polynômes de I comme relations de dépendance linéaire entre les monômes de leur support.

- On peut remplacer les réductions de S-polynômes par des échelonisations de matrices.

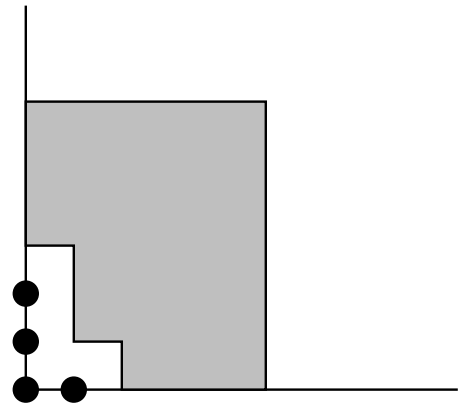
$$\left(\begin{array}{c} \text{monômes} \\ \frac{ppcm(\gamma(p),\gamma(q))}{\gamma(p)}p \\ \frac{ppcm(\gamma(p),\gamma(q))}{\gamma(q)}q \\ \vdots \\ \hline \\ \text{polynômes réducteurs} \end{array}\right)$$

Des problèmes avec les bases de Gröbner

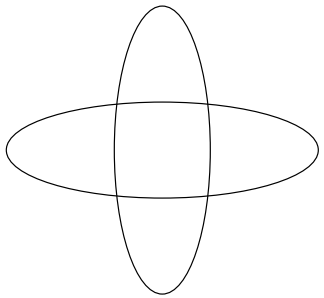
ex: $p_1 = ax_1^2 + bx_2^2 + n_1(x_1, x_2)$
$p_2 = cx_1^2 + dx_2^2 + n_2(x_1, x_2)$

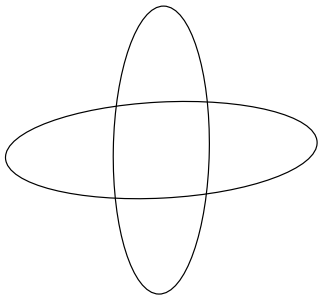


$n_1 = n_2 = 0$



$n_1 \equiv \varepsilon_1 x_1 x_2$
$n_2 \equiv \varepsilon_2 x_1 x_2$





Calculer de meilleures représentations

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$
- Une fonction de choix raffinant le degré, γ, est une fonction qui à un polynôme p associe $\gamma(p)$, un des monômes du support de p de degré maximal $deg(\gamma(p)) = deg(p)$.

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$
- Une fonction de choix raffinant le degré, γ, est une fonction qui à un polynôme p associe $\gamma(p)$, un des monômes du support de p de degré maximal $deg(\gamma(p)) = deg(p)$.
- Soit S un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$, S^+ est l'ensemble :

$$S^+ = x_1 S \cup \cdots \cup x_n S$$

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$
- Une fonction de choix raffinant le degré, γ, est une fonction qui à un polynôme p associe $\gamma(p)$, un des monômes du support de p de degré maximal $deg(\gamma(p)) = deg(p)$.
- Soit S un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$, S^+ est l'ensemble :

$$S^+ = x_1 S \cup \cdots \cup x_n S$$

- $P \subset \mathbb{K}[x_1, \ldots, x_n], M \subset \{x^\alpha,\ \alpha \in \mathbb{Z}^n\}$,$(P|M)$, est la matrice dont les colonnes sont indicées par les éléments de M et les lignes par ceux de P.

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$
- Une fonction de choix raffinant le degré, γ, est une fonction qui à un polynôme p associe $\gamma(p)$, un des monômes du support de p de degré maximal $deg(\gamma(p)) = deg(p)$.
- Soit S un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$, S^+ est l'ensemble :

$$S^+ = x_1 S \cup \cdots \cup x_n S$$

- $P \subset \mathbb{K}[x_1, \ldots, x_n], M \subset \{x^\alpha,\ \alpha \in \mathbb{Z}^n\}$,$(P|M)$, est la matrice dont les colonnes sont indicées par les éléments de M et les lignes par ceux de P.
- Soient p_1, $p_2 \in \mathbb{K}[x_1, \ldots, x_n]$, le C-polynôme de p_1 et p_2 est $C(p_1, p_2) = \frac{ppcm(\gamma(p_1),\gamma(p_2))}{\gamma(p_1)} p_1 - \frac{ppcm(\gamma(p_1),\gamma(p_2))}{\gamma(p_2)} p_2$, et $deg_C(C(p_1, p_2)) = deg(ppcm(\gamma(p_1), \gamma(p_2))$.

Notations

- Une forme normale est :
  - B une base monomiale de A
  - Un algorithme pour projeter $\mathbb{K}[x_1, \ldots, x_n]$ sur $\langle B \rangle$
- Une fonction de choix raffinant le degré, γ, est une fonction qui à un polynôme p associe $\gamma(p)$, un des monômes du support de p de degré maximal $deg(\gamma(p)) = deg(p)$.
- Soit S un sous ensemble de $\mathbb{K}[x_1, \ldots, x_n]$, S^+ est l'ensemble :

$$S^+ = x_1 S \cup \cdots \cup x_n S$$

- $P \subset \mathbb{K}[x_1, \ldots, x_n], M \subset \{x^\alpha,\ \alpha \in \mathbb{Z}^n\}$,$(P|M)$, est la matrice dont les colonnes sont indicées par les éléments de M et les lignes par ceux de P.
- Soient p_1, $p_2 \in \mathbb{K}[x_1, \ldots, x_n]$, le C-polynôme de p_1 et p_2 est $C(p_1, p_2) = \frac{ppcm(\gamma(p_1),\gamma(p_2))}{\gamma(p_1)} p_1 - \frac{ppcm(\gamma(p_1),\gamma(p_2))}{\gamma(p_2)} p_2$, et $deg_C(C(p_1, p_2)) = deg(ppcm(\gamma(p_1), \gamma(p_2))$.

Macaulay revisité

Fournir un moyen de réécrire un polynôme p quelconque sous la forme

$$p = b + i,\ b \in \langle B \rangle,\ i \in \langle I \rangle$$

Macaulay revisité

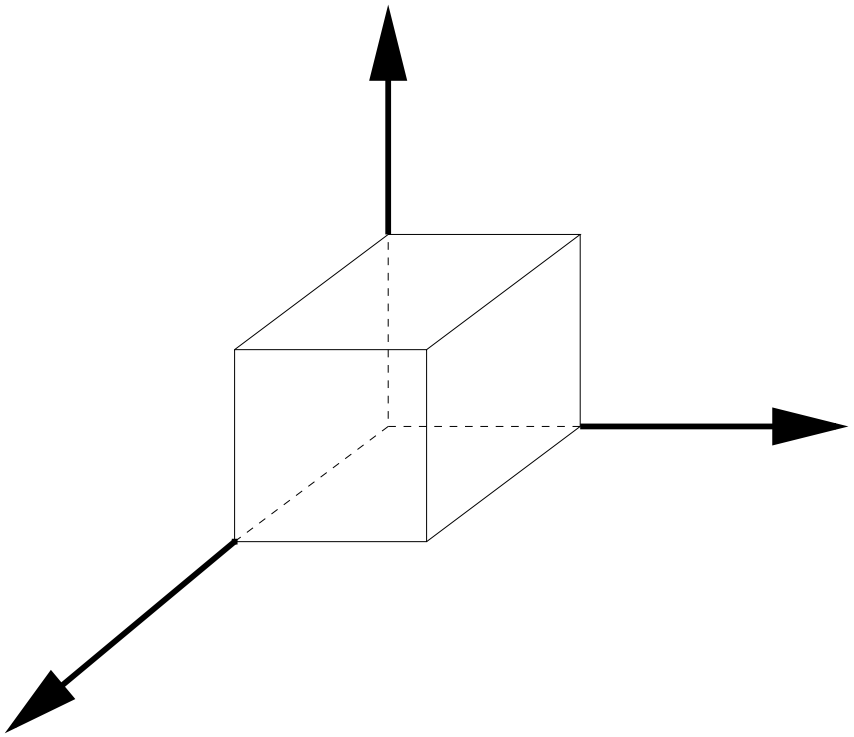
Fournir un moyen de réécrire un polynôme p quelconque sous la forme

$$p = b + i,\ b \in \langle B \rangle,\ i \in \langle I \rangle$$

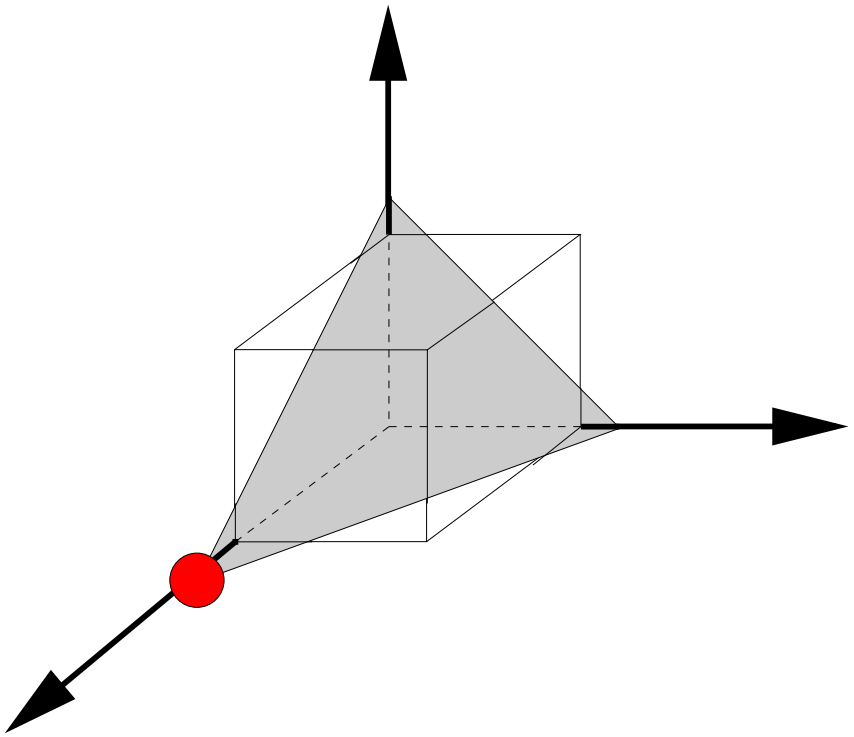
Algorithme 4. [Mourrain, T. ISSAC00] INPUT : $F = f_1, \ldots, f_s$ *polynômes génériques*
OUTPUT : *La représentation de A caractérisée par Macaulay.*

- $k = min(deg(f_i),\ i = 1..s)$.
- $P_k = F[k]$, $M_k = \{x_i^k,\ f_i \in P_k\}$
- **while** $k \leq \Sigma_{i=1..s}(d_i - 1) + 1$ **do**
  - $P_{k+1} = P_k^+ \cap B^+ \cup proj(P[k+1])$, $M_{k+1} = M_k^+ \cap B^+ \cup \{x_i^{k+1}, f_i \in P[k+1]\}$
  - *Résoudre le système linéaire* $(P_{k+1}|M_{k+1})X = P_{k+1}$.
  - $k = k + 1$
- *return* $P_i, i = min(deg(f_i),\ i = 1..s)..\Sigma_{i=1..s}(d_i - 1) + 1$

Macaulay revisité un exemple

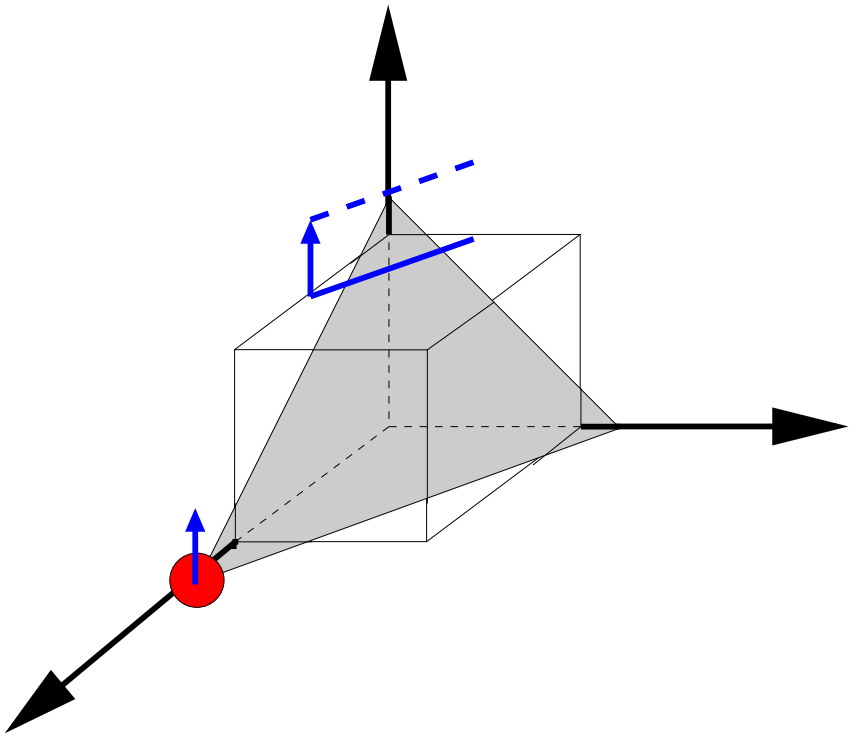


Macaulay revisité un exemple



$$f_i = x_i^{d_i} + \dots$$

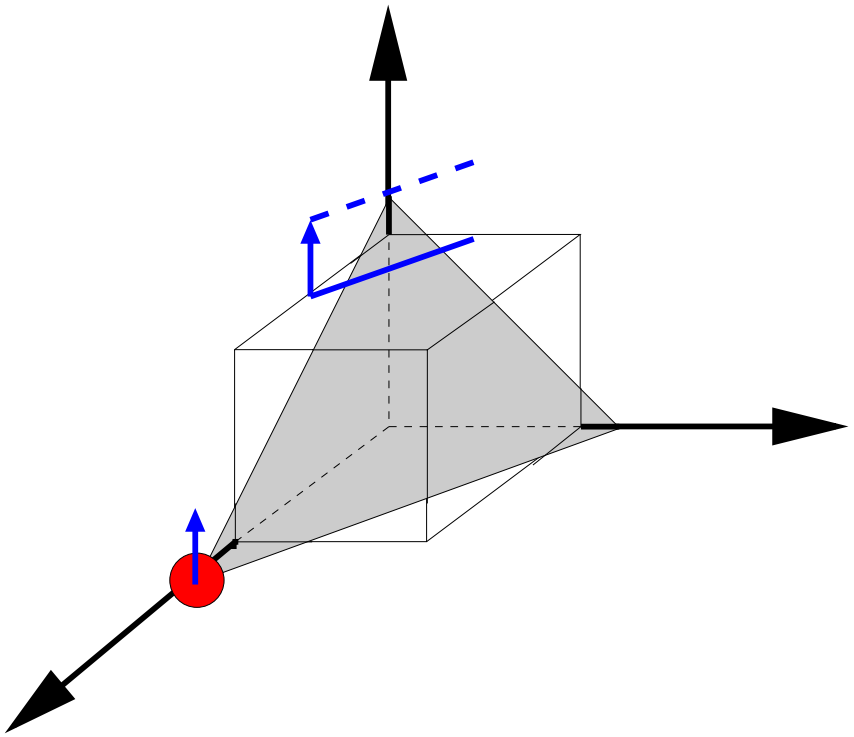
Macaulay revisité un exemple



$$f_i = x_i^{d_i} + \dots$$

$$x_1 f_i = x_1 x_i^{d_i} + \alpha_2 x_1^{d_1} x_2 + \cdots + \alpha_n x_1^{d_1} x_n + \dots$$

Macaulay revisité un exemple

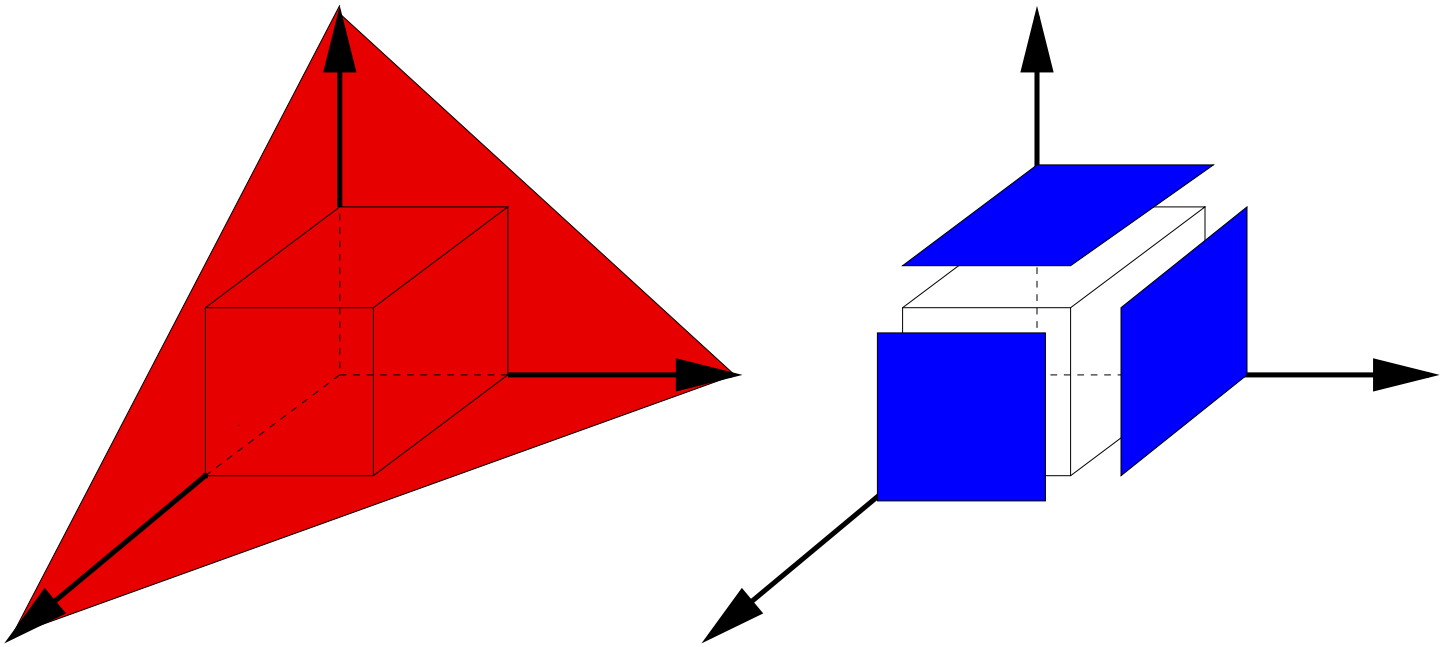


$$f_i = x_i^{d_i} + \dots$$

$$x_1 f_i = x_1 x_i^{d_i} + \alpha_2 x_1^{d_1} x_2 + \cdots + \alpha_n x_1^{d_1} x_n + \dots$$

$$\begin{pmatrix} & \cdots\cdots\cdots\cdots\cdots & \\ 1 & \dots \quad \alpha_2 \quad \alpha_3 \quad \dots \quad \alpha_n & \dots \\ & \cdots\cdots\cdots\cdots\cdots & \end{pmatrix}$$

Comparaison avec l'original



Comparaison avec l'original

n	5	6	7	8	9	10	11
taille	5	6	7	8	9	10	11
des	20	30	42	56	72	90	110
matrices	**30**	**60**	105	168	252	360	495
	20	**60**	**140**	**280**	504	840	1320
	5	30	105	**280**	**630**	**1260**	2310
		6	42	168	504	**1260**	**2772**
			7	56	252	840	2310
				8	72	360	1320
					9	90	495
						10	110
							11
Somme	80	192	448	1024	2304	5120	11264
Macaulay	462	1 716	6 435	24 310	92 378	352 716	1 352 078
Nb points	32	64	128	256	512	1024	2048

Table 1: Taille des systèmes à inverser

Que faire si B n'est pas connu?

Algorithme 5. [Mourrain] INPUT : $F = f_1, \ldots, f_n$
L *un* $\mathbb{K}$*-espace vectoriel connexe à* 1

OUTPUT : *La structure multiplicative de* A

1) $K_0 = \langle f_1, \ldots, f_n \rangle$, $n = 0$

2) **repeat**

- $K_{n+1} = K_n^+ \cap L$
- $n = n + 1$

3) **until** $K_n == K_{n-1}$

4) Calculer B *un supplémentaire de* K_n *dans* L

5) Si $B^+ \not\subset L$, $L = L^+$ *et retour à l'étape 1)*

Que faire si B n'est pas connu?

Pour prouver la correction de cet algorithme, on a besoin d'un nouveau critère :

- $\mathbb{K}[x_1, \ldots, x_n] = < B > \oplus < I >$.

  $\Longleftrightarrow$

- Les opérateurs de multiplication commutent

Que faire si B n'est pas connu?

Pour prouver la correction de cet algorithme, on a besoin d'un nouveau critère :

- $\mathbb{K}[x_1, \ldots, x_n] = < B > \oplus < I >$.

  $\Longleftrightarrow$

- Les opérateurs de multiplication commutent

Propriétés

$\oplus$ très bonne stabilité numérique.

$\oplus$ Possibilité de prendre en compte la géométrie du problème.

$\ominus$ Calcul **TRES** coûteux.

Calculer mieux pour calculer moins

Ce qui ne se passe pas bien :

- On a relégué le calcul effectif de B à la dernière étape.

Calculer mieux pour calculer moins

Ce qui ne se passe pas bien :

- On a relégué le calcul effectif de B à la dernière étape.

Ce qu'il faut faire pour imiter les bases de Gröbner :

- essayer dès le départ de deviner quel sera B.
- vérifier que B est bien une base de A.
- éviter de calculer des polynômes qui vont $trop\ loin\ de\ B$

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les* C*-pol de degré* $k+1$

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les* C*-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les* C*-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.
- $M_{k+1} = \{M_k^+ \cap b^+\}$

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les* C*-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.
- $M_{k+1} = \{M_k^+ \cap b^+\}$
- $PseudoSolve\,(P_{k+1} | M_{k+1}) X = P_{k+1}$

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(0-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les C-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.
- $M_{k+1} = \{M_k^+ \cap b^+\}$
- $PseudoSolve\,(P_{k+1}|M_{k+1})X = P_{k+1}$
- *Réduire les C-pol par rapport aux* P_j

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(*0*-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les* C*-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.
- $M_{k+1} = \{M_k^+ \cap b^+\}$
- $PseudoSolve\,(P_{k+1}|M_{k+1})X = P_{k+1}$
- *Réduire les* C*-pol par rapport aux* P_j
- *Suivant que les* C*-pol se réduisent à* 0 *ou non et suivant que* $(P_{k+1}|M_{k+1})$ *est de rang maximal ou non, mettre à jour* b, P_{k+1}, k, M_{k+1}

End `While`

Calculer mieux pour calculer moins

Algorithme 6. [T., thèse 2002] FORME NORMALE

INPUT : $F = f_1, \ldots, f_s$ *définissant* I *(0-dimensionnel)*
γ *une* **fonction** *de choix raffinant le degré.*

Initialisation : *Sélectionner les* f_i *de degré minimal*
$b = (\gamma(f_i))$, $k = deg(f_i)$, $P_k = \{f_i\}$, $M_k = \{\gamma(f_i)\}$

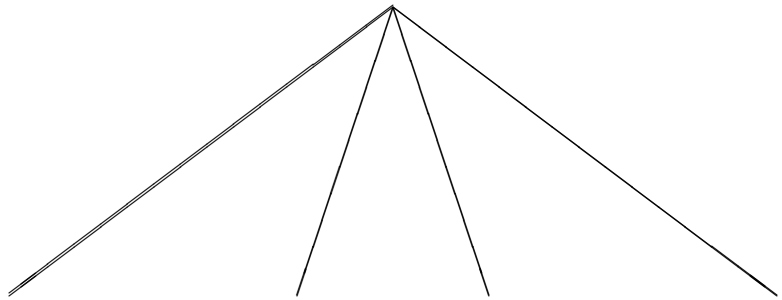
Core Loop : `While` $(newmon || k \leq Maxdeg(F))$ *do*

- *Calculer les C-pol de degré* $k+1$
- *Calculer* $P_{k+1} = P_k^+ \cap b^+$ *et prendre en compte les* f_i *de degré* $k+1$.
- $M_{k+1} = \{M_k^+ \cap b^+\}$
- $PseudoSolve\,(P_{k+1}|M_{k+1})X = P_{k+1}$
- *Réduire les C-pol par rapport aux* P_j
- *Suivant que les C-pol se réduisent à* 0 *ou non et suivant que* $(P_{k+1}|M_{k+1})$ *est de rang maximal ou non, mettre à jour* b, P_{k+1}, k, M_{k+1}

End `While`

OUTPUT : $\{P_j, j = 0..k\}$ *qui permettent de construire une forme normale pour tout* k $\forall k \in \mathbb{N}$

$r = \#M_{k+1} - Rang((M_{k+1}|P_{k+1}))$
$c = \#\{\text{non réduits à } 0\}$



$\mathbf{r = 0}$ && $\mathbf{c = 0}$	$\mathbf{r = 0}$ && $\mathbf{c \neq 0}$	$\mathbf{r \neq 0}$ && $\mathbf{c \neq 0}$	$\mathbf{r \neq 0}$ && $\mathbf{c = 0}$
Passer à l'étape suivante	Traiter les non réduits et repartir à leur degré	Rajouter à b et traiter les non 0	Rajouter à b Passer à l'étape suivante

Un exemple

$$F = \begin{cases} x^2 + xy \\ y^2 + xy \\ xy^3 \end{cases}$$

$\gamma =$ ask user

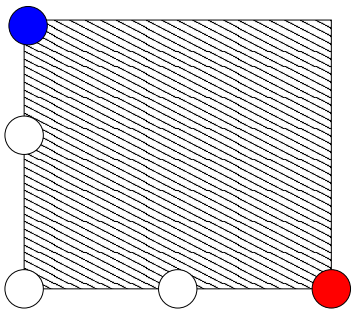
Calculer le quotient :

Un exemple

$$F = \begin{cases} x^2 + xy \\ y^2 + xy \\ xy^3 \end{cases}$$

$\gamma =$ ask user

Calculer le quotient :

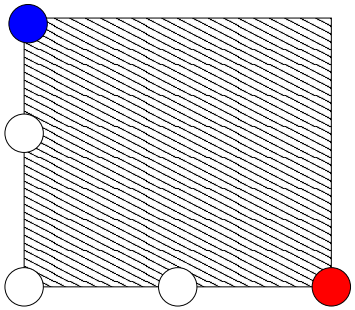


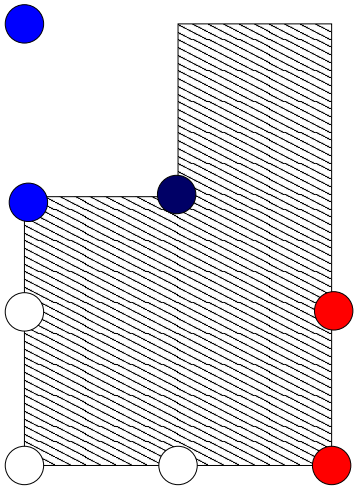
Un exemple

$$F = \begin{cases} x^2 + xy \\ y^2 + xy \\ xy^3 \end{cases}$$

$\gamma =$ ask user

Calculer le quotient :



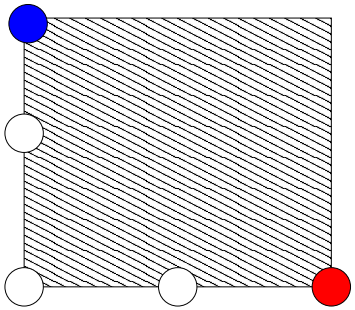


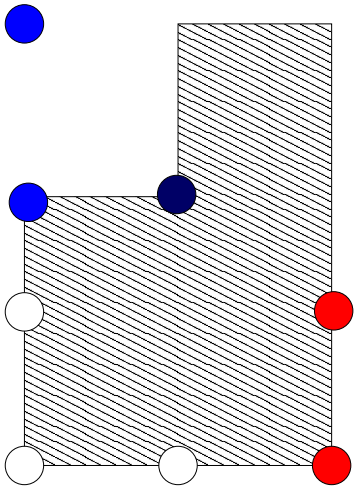
Un exemple

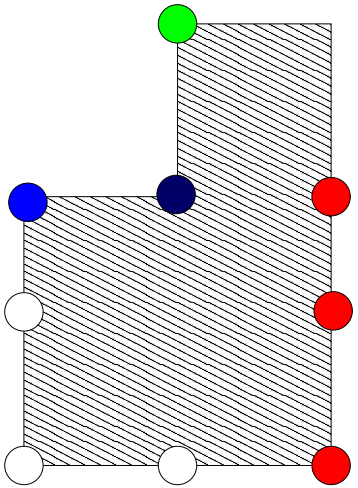
$$F = \begin{cases} x^2 + xy \\ y^2 + xy \\ xy^3 \end{cases}$$

$\gamma =$ ask user

Calculer le quotient :







Que faire si on ne raffine pas le degré?

Un exemple : l'ordre `Lex`
Les Problèmes :

- On peut avoir besoin de polynômes non encore $réduits$.
- On ne peut pas déterminer à l'avance quels polynômes seront ou non réduits.
- On doit éviter les dead-lock (i.e. le polynôme p dépendant du polynôme q et le polynôme q dépendant du polynôme p).

Que faire si on ne raffine pas le degré?

Un exemple : l'ordre `Lex`
Les Problèmes :

- On peut avoir besoin de polynômes non encore $réduits$.
- On ne peut pas déterminer à l'avance quels polynômes seront ou non réduits.
- On doit éviter les dead-lock (i.e. le polynôme p dépendant du polynôme q et le polynôme q dépendant du polynôme p).

Les solutions:

- Mettre de côté les polynômes non réduits
- à la fin de chaque tour de boucle déterminer quels sont les polynômes réduits.
- utiliser une forme linéaire pour guider les choix de monômes.

Un environnement pour le calcul symbolique et numérique

Un environnement pour le calcul symbolique et numérique

SYmbolic Numeric APplicationS

Un environnement pour le calcul symbolique et numérique

SYmbolic Numeric APplicationS

SYNAPS

Un environnement pour le calcul symbolique et numérique

SYmbolic Numeric APplicationS

Plateforme logicielle commune.

GALAAD ; SPACES ; . . . travail collaboratif.

- Intégration de logiciels "libres" dans un environnement **homogène**.
- Noyau en C++ (anciennement ALP).
- Distribution `cvs@cvs-sop.inria.fr`, GPL+runtime exception.

Structures de **données de base** : **vecteurs, matrices (dense, Toeplitz, Hankel, creuse), polynômes en une ou plusieurs variables, . . .**

SYNAPS

Un environnement pour le calcul symbolique et numérique

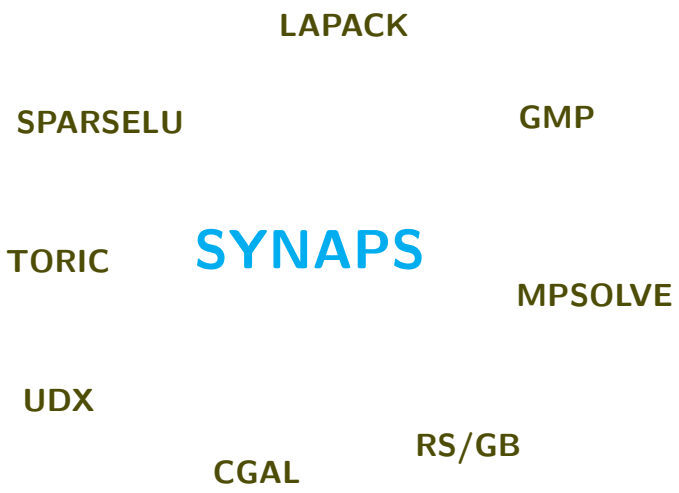
SYmbolic Numeric APplicationS

Plateforme logicielle commune.

GALAAD ; SPACES ; . . . travail collaboratif.

- Intégration de logiciels "libres" dans un environnement **homogène**.
- Noyau en C++ (anciennement ALP).
- Distribution `cvs@cvs-sop.inria.fr`, GPL+runtime exception.

Structures de **données de base** : **vecteurs, matrices (dense, Toeplitz, Hankel, creuse), polynômes en une ou plusieurs variables, . . .**



Un environnement pour le calcul symbolique et numérique

SYmbolic Numeric APplicationS

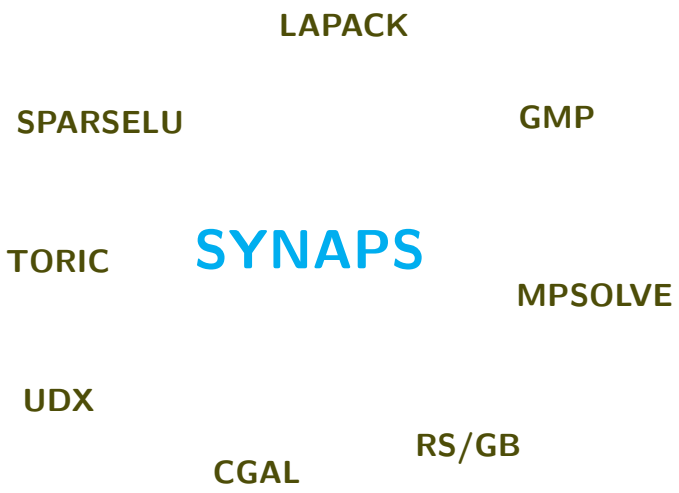
Plateforme logicielle commune.

GALAAD ; SPACES ; . . . travail collaboratif.

- Intégration de logiciels "libres" dans un environnement **homogène**.
- Noyau en C++ (anciennement ALP).
- Distribution `cvs@cvs-sop.inria.fr`, GPL+runtime exception.

Structures de **données de base** : **vecteurs, matrices (dense, Toeplitz, Hankel, creuse), polynômes en une ou plusieurs variables, . . .**

Modules spécialisés



Un environnement pour le calcul symbolique et numérique

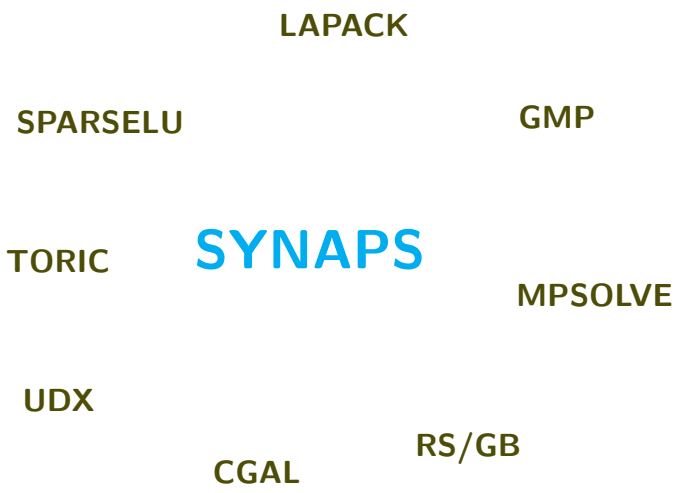
SYmbolic Numeric APplicationS

Plateforme logicielle commune.

GALAAD ; SPACES ; . . . travail collaboratif.

- Intégration de logiciels "libres" dans un environnement **homogène**.
- Noyau en C++ (anciennement ALP).
- Distribution `cvs@cvs-sop.inria.fr`, GPL+runtime exception.

Structures de **données de base** : **vecteurs, matrices (dense, Toeplitz, Hankel, creuse), polynômes en une ou plusieurs variables, . . .**



Modules spécialisés

- **Courbes et Surfaces, liens avec CGAL, ECG, GAIA.**
- **Résolution : intégration, génération d'outils, protocoles.**
- **Factorisation et décomposition.**

Applications

- **Vision artificielle**
  Equations de Kruppa

- **Géométrie algorithmique**
  Calcul de cylindres passant par 4 ou 5 points

- **Robotique**
  Modèle géométrique direct du robot parallèle

- **Pharmacognosie**
  chromatographie de partage centrifuge

La Chromatographie de Partage Centrifuge

Problème: Accélérer et rendre plus robuste la simulation de CPC.

- $v = \frac{V_stat}{V_mob}$, $x_0 = [I^-]$,$x_1 = [Cl^-]$,$x_2 = [A_1^-]$, $x_3 = [A_2^-]$, $x_4 = v[B^+, I^-]$, $x_5 = v[B^+, Cl^-]$, $x_6 = v[B^+, A_1^-]$, $x_7 = v[B^+, A_2^-]$, $x_8 = [C]$
- les constantes chimiques, $K_C = a_6 = \frac{[C]}{[A_1^-][A_2^-]}$, $K_{Cl} = \frac{[Cl^-][B^+,I^-]}{[I^-][B^+,Cl^-]} = va_7$, $K_{A_1} = \frac{[A_1^-][B^+,I^-]}{[I^-][B^+,A_1^-]} = va_8$, $K_{A_2} = \frac{[A_2^-][B^+,I^-]}{[I^-][B^+,A_2^-]} = va_8$

- conservations de la matière:

$$a_1 = x_0 + x_4$$
$$a_2 = x_1 + x_5$$
$$a_3 = x_2 + x_6 + x_8$$
$$a_4 = x_3 + x_7 + x_8$$
$$a_4 = x_4 + x_5 + x_6 + x_7$$

- pour les équilibres chimiques:

$$a_6 x_2 x_3 = x_8$$
$$a_7 x_0 x_5 = x_1 x_4$$
$$a_8 x_0 x_6 = x_2 x_4$$
$$a_9 x_0 x_8 = x_3 x_4$$

Précision dans les calculs de F.N.	Temps	Mémoire	$max(\|f_i(\zeta_j\|))$
double	$0.01s$	1M	erreur
128 bits	$0.09s$	1M	10^{-12}
∞	$0.23s$	2M	10^{-12}

Travail récent

- **Travail sur les variétés de dimension positive**
  Démonstration d'un théorème fort de Bézout dans le cadre bi-homogène [Safey, T. Issac2004 soumis], amélioration des meilleures bornes connues sur le nombre de composantes connexes d'une variété réelle.

- **Mesures de stabilité**
  Implantation d'une arithmétique des infinitésimaux [1], et connexion avec l'implantation SYNAPS existante.

[1] http://www-sop.inria.fr/galaad/personnel /trebuchet/mpai.tgz

Perspectives

Mise en place d'un critère efficace pour le cadre des idéaux de dimension positive

- Régularité de Castelnuovo-Mumford du gradué.
- Utilisation de critères de detection de la régularité[Quillen, Bayer-Stilman].
- Caractérisation de conformations géométriques où l'on peut conclure sans calculs supplémentaires.
- Implantation

Perspectives

Ce qu'il reste à faire :

- Etude poussée de la stabilité du résultat (définition de différentes notions et évaluations).
- Optimisation des calculs de C-polynômes (critères de Buchberger, critère F_5...)
- Ouverture vers d'autres domaines (Cryptologie, équations différentielles)