RAIRO. INFORMATIQUE THÉORIQUE ET APPLICATIONS

LIONEL DUCOS

Algorithme de Bareiss, algorithme des sous-résultants

RAIRO. Informatique théorique et applications, tome 30, n° 4 (1996), p. 319-347

<http://www.numdam.org/item?id=ITA_1996__30_4_319_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 30, n° 4, 1996, pp. 319-347)

ALGORITHME DE BAREISS, ALGORITHME DES SOUS-RÉSULTANTS (*)

par Lionel Ducos ([1])

Communiqué par C. Choffrut

Résumé. – *L'algorithme de Bareiss (calculant le déterminant d'une matrice) n'est autre qu'une méthode visant l'élimination optimale des composantes des vecteurs colonnes de cette matrice par triangularisation. L'algorithme des sous-résultants (calculant le résultant, mais aussi le pgcd, de deux polynômes) vise de même l'élimination optimale des coefficients de ces deux polynômes.*

L'objet de cet article est de réunir sous un même formalisme mathématique l'algorithme de Bareiss et celui des sous-résultants en insistant sur l'aspect intrinsèque des objets, et si possible, en évitant les complications calculatoires qui obscurcissent leurs exposés.

Enfin, une variante de l'algorithme des sous-résultants sera proposée : son intérêt est de calculer la chaîne des polynômes sous-résultants de façon « économique » (voir *section 6.2 page 23).*

Abstract. – *Bareiss' algorithm computes the determinant of a matrix. It eliminates in an optimal way the coordinates of its column vectors by fraction-free triangularisation. The subresultant algorithm also calculates the resultant and the greatest common divisor of two polynomials by eliminating optimally their coefficients.*

In this article, we relate these two algorithms in a single mathematical concept. Intrinsic aspects of this relation are emphasized, thus most of the calculations is avoided as they are long and complicated.

Finally, we give a variation of the subresultant algorithm which makes the computation of the chain of the subresultant polynomials more "economical" (see *section 6.2 page 23).*

INTRODUCTION

L'étude des algorithmes de Bariess et des sous-résultants amène à penser à l'existence d'un lien entre ces méthodes, bien qu'assez différentes par les objets qu'elles manipulent. Dans un premier temps, on étudiera la méthode de Bareiss « pas à pas » afin de donner naissance de façon naturelle à quelques formules simples d'algèbre multilinéaire. Après avoir totalement démontré l'algorithme de Bareiss, une présentation inhabituelle

(*) Reçu en février 1995.

([1]) Département de Mathématiques, Université de Poitiers, 40, avenue du Recteur Pineau, 86022 Poitiers Cedex, France. E-mail: ducos@mathlabo.univ-poitiers.fr

Informatique théorique et Applications/Theoretical Informatics and Applications
0988-3754/96/04/$ 7.00/

des polynômes sous-résultants permettra de retrouver des relations liant ces polynômes: similarité et pseudo-divisions entre autres. Certaines de ces relations justifieront l'algorithme des sous-résultants, mais toutes sont nécessaires pour démontrer une nouvelle variante de cet algorithme.

SOMMAIRE

1. L'ÉLIMINATION DE GAUSS ET L'APPLICATION $\natural$

Soit R un anneau commutatif intègre et M un R-module libre muni d'une base fixée. On note π_i la projection sur la i-ième coordonnée et $\Omega^n(M)$ le R-module des n-linéaires alternées de M dans R (que l'on nommera n-formes).

Étant donnés w, x, y, z quatre vecteurs de M, si l'on veut annuler la première coordonnée des vecteurs x, y, z par une élimination « à la Gauss » en se servant de w, la combinaison la plus simple est:

$$\begin{cases} x' = \pi_1(w)\, x - \pi_1(x)\, w \\ y' = \pi_1(w)\, y - \pi_1(y)\, w \\ z' = \pi_1(w)\, z - \pi_1(z)\, w \end{cases}$$

Ces combinaisons font apparaître une 2-application (application 2-linéaire alternée) :

$$\pi_1^\natural : \quad (a, b) \in M^2 \mapsto \pi_1(a)\, b - \pi_1(b)\, a \in M$$

Rappel rapide sur le produit extérieur : Si $g \in \Omega^n(M)$ et $f \in \Omega^m(M)$ sont deux formes multilinéaires alternées, on définit le produit extérieur de g et f (noté $g \wedge f$) par :

$$(v_1, \ldots, v_{n+m}) \in M^{n+m} \mapsto \sum_\sigma \operatorname{sgn}(\sigma)\, g(v_{\sigma_1}, \ldots, v_{\sigma_n})\, f(v_{\sigma_{n+1}}, \ldots, v_{\sigma_{n+m}})$$

où la somme porte sur les permutations σ de $\mathcal{S}_{n+m}$ telles que

$$\sigma_1 < \sigma_2 < \cdots < \sigma_n$$

et $\sigma_{n+1} < \cdots < \sigma_{n+m}$.

On voit bien que l'on peut prendre pour f une **application** $M^m \to N$ multilinéaire alternée tout en laissant la formule ci-dessus bien définie.

DÉFINITION 1 : *Soit f une application $M^m \to N$ multilinéaire alternée et $g \in \Omega^n(M)$, on définit le produit extérieur $g \wedge f$ par :*

$$M^{n+m} \to N :$$
$$(v_1, \ldots, \sigma_{n+m}) \mapsto \sum_\sigma \operatorname{sgn}(\sigma)\, g(v_{\sigma_1}, \ldots, v_{\sigma_n})\, f(v_{\sigma_{n+1}}, \ldots, v_{\sigma_{n+m}})$$

Par cette extension du produit extérieur, on a :

$$\pi_1^\natural = \pi_1 \wedge \mathrm{Id}_M$$

Si l'on réalise une étape supplémentaire pour éliminer la seconde coordonnée de y', z', on combine :

$$\begin{cases} y'' = \pi_2(x')\, y' - \pi_2(y')\, x' = \pi_2^\natural(x', y') \\ z'' = \pi_2(x')\, z' - \pi_2(z')\, x' = \pi_2^\natural(x', z') \end{cases}$$

et on s'aperçoit (la démonstration sera faite dans le paragraphe 2.1 p. 6, relation $(\star)$) que $y'' = \pi_1(w)\, \mu(w, x, y)$ et $z'' = \pi_1(w) \mu(w, x, z)$ où μ est la 3-application suivante :

$$(a, b, c) \in M^3 \mapsto \begin{vmatrix} \pi_1(a) & \pi_1(b) \\ \pi_2(a) & \pi_2(b) \end{vmatrix} c - \begin{vmatrix} \pi_1(a) & \pi_1(c) \\ \pi_2(a) & \pi_2(c) \end{vmatrix} b + \begin{vmatrix} \pi_1(b) & \pi_1(c) \\ \pi_2(b) & \pi_2(c) \end{vmatrix} a$$

Si l'on pose $\det_{\binom{1}{2}} = \pi_1 \wedge \pi_2 : (a, b) \in M^2 \mapsto \begin{vmatrix} \pi_1(a) & \pi_1(b) \\ \pi_2(a) & \pi_2(b) \end{vmatrix}$, et si l'on se permet l'extension du produit extérieur, on obtient alors :

$$\mu = \det_{\binom{1}{2}} \wedge \mathrm{Id}_M = \det^{\natural}_{\binom{1}{2}} \frac{y''}{\pi_1(w)}$$
$$= \det^{\natural}_{\binom{1}{2}} (w, x, z) \frac{z''}{\pi_1(w)} = \det^{\natural}_{\binom{1}{2}} (w, x, z)$$

C'est en utilisant ce genre de propriétés de divisibilité que Bareiss démontre dans [2] comment calculer le déterminant d'une matrice, à condition de pouvoir diviser par certaines valeurs ($\pi_1(w)$ par exemple). Le lecteur pourra aussi consulter [6] (pages 389 à 405) pour plus de détails sur la méthode de Bareiss et des améliorations possibles.

On voit maintenant la nécessité de définir une transformation $g \mapsto g^{\natural}$. On en ébauchera quelques propriétés avant de continuer l'étude de la méthode de Gauss.

NOTATION : Si $x \in M^n$ alors $x_i \in M$ désigne sa i-ième composante.

DÉFINITION 2 : *Si $g \in \Omega^{n-1}(M)$ est une $(n-1)$-***forme** *de M alors on appelle $g^{\natural} : M^n \to M$ la n-***application**

$$g \wedge \mathrm{Id}_M : v \in M^n \mapsto \sum_{i=1}^{n} (-1)^{n-i} g(v_1, \ldots, \not{v}_i, \ldots, v_n) v_i$$

PROPRIÉTÉ 1 : *Si $f \in \Omega^m(M)$ et $g \in \Omega^n(M)$, on a*

$$(f \wedge g)^{\natural} = f \wedge g^{\natural} \qquad g \wedge f^{\natural} = (-1)^{mn} f \wedge g^{\natural}$$

Si $f : M \mapsto N$ est un morphisme de R-modules alors $f \circ g^{\natural} = g \wedge f$.

Remarquer que $g^{\natural}$ est la seule application vérifiant la dernière relation pour toute f (g étant fixée).

2. ALGORITHME DE BAREISS

2.1. Premiers pas...

DÉFINITION 3 : *Si g est une application multilinéaire alternée de M^n sur un R-module (en particulier si g est une n-forme multilinéaire alternée de M), on nomme* $\ker g$ *le sous-module formé des éléments de M qui annulent g dès qu'ils font partie de ses arguments.*

Exemple et contre-exemple: Le vecteur $(0, 0, 1)$ fait partie de $\ker(\pi_1 \wedge \pi_2)$, mais n'est pas élément de $\ker(\pi_2 \wedge \pi_3)$.

PROPRIÉTÉ 2: *Soit f est une m-application et g une n-forme sur M. Alors*

$$\ker g \cap \ker f \subset \ker(g \wedge f)$$

Le théorème suivant est insipiré de ce qu'expose Quitté dans [10].

THÉORÈME 1: *Soit M et N deux R-modules, $f : M^{n+1} \to N$ une $(n+1)$-application et $g \in \Omega^{m-1}(M)$. On considère $x \in M^m$, $z \in M^n$ tels que $\forall\, j \in [1, n]$, $z_j \in \ker g$. Alors on a:*

$$f(g^\natural(x), z) = (g \wedge f)(x, z).$$

Démonstration

$$\begin{aligned}(g \wedge f)(x, z) &= \sum_{i=1}^{m} (-1)^{m-i} g(x_1, \ldots, \not{x}_i, \ldots, x_n) f(x_i, z) \\ &\text{car } \forall j, \quad z_j \in \ker g, \\ &= f\left(\sum_{i=1}^{m} (-1)^{m-i} g(x_1, \ldots, \not{x}_i, \ldots, x_n) \cdot x_i,\ z\right) \\ &\text{par linéarité de } f, \\ &= f(g^\natural(x), z). \quad \square\end{aligned}$$

DÉFINITION 4: *On dit que g est une n-forme* **pure** *si g est le produit extérieur de n 1-formes.*

THÉORÈME 2: *Si g est une n-forme* **pure** *alors* $\operatorname{im} g^\natural \subset \ker g$.

Démonstration: On a $g = g_1 \wedge g_2 \wedge \cdots \wedge g_n$ où g_i est une forme linéaire de M sur R, donc $\forall\, i$, $g_i \circ g^\natural = g \wedge g_i = 0$ et comme $g = g_1 \wedge g_2 \wedge \cdots \wedge g_n$, on a bien le résultat annoncé par:

$$\operatorname{im} g^\natural \subset \bigcap \ker g_i \subset \ker\left(\bigwedge g_i\right) = \ker g. \quad \square$$

Par exemple, en prenant f une 2-application de M dans N et g une n-forme **pure** de M, $w \in M^n$ et $x, y \in M$, grâce au théorème 2 on a $g^\natural(w, y) \in \ker g$ et grâce au théorème 1 on obtient:

$$f(g^\natural(w, x), g^\natural(w, y)) = (g \wedge f)(w, x, g^\natural(w, y))$$

Comme $g \wedge f$ est alternée, seul le terme $g(w) \cdot y$ de $g^\natural(w, y)$ n'est pas éliminé par les composantes de w:

$$\begin{aligned}(g \wedge f)(w, x, g^\natural(w, y)) &= (g \wedge f)(w, x, g(w)\, y - g(y)\, w) \\ &= (g \wedge f)(w, x, g(w)\, y)\end{aligned}$$

On a maintenant:

$$f(g^\natural(w, x), g^\natural(w, y)) = g(w)\,(g \wedge f)(w, x, y) \qquad (\star)$$

Si l'on pose $f = \pi_2^\natural$ et $g = \pi_1$ et que l'on reprend la deuxième élimination « à la Gauss » de la page 3, cette relation devient:

$$\begin{aligned}y'' &= \pi_2^\natural(\pi_1^\natural(w, x), \pi_1^\natural(w, y)) = \pi_1(w)\,(\pi_1 \wedge \pi_2^\natural)(w, x, y) \\ &= \pi_1(w)\, \det{}^\natural_{\binom{1}{2}}(w, x, y)\end{aligned}$$

(et de même avec z'', w, x, z) ce qui prouve (comme il était annoncé page 3) que $\mu = \det{}^\natural_{\binom{1}{2}}$.

On divise y'' et z'' par $\pi_1(w)$ (le pivot de la première étape) et on appelle de nouveau ces quotients exacts y'' et z'' pour simplifier. En réitérant une dernière fois l'élimination « à la Gauss » avec y'' et z'', grâce aux théorèmes 1 et 2, on a:

$$\begin{aligned}\pi_3^\natural(y'', z'') &= \pi_3^\natural(\det{}^\natural_{\binom{1}{2}}(w, x, y), \det{}^\natural_{\binom{1}{2}}(w, x, z)) \\ &= \det{}^\natural_{\binom{1}{3}}(w, x, y, \det{}^\natural_{\binom{1}{2}}(w, x, z)) \\ &= \det{}_{\binom{1}{2}}(w, x)\, \det{}^\natural_{\binom{1}{3}}(w, x, y, z)\end{aligned}$$

où $\det{}^\natural_{\binom{1}{3}} = \det{}_{\binom{1}{2}} \wedge \pi_3^\natural = \pi_1 \wedge \pi_2 \wedge \pi_3 \wedge \mathrm{Id}$.

On voit que $\pi_3^\natural(y'', z'')$ est divisible par $\det{}_{\binom{1}{2}}(w, x) = \pi_2(\pi_1^\natural(w, x))$ le pivot de la deuxième étape) et le résultat d'une telle division est $\det{}^\natural_{\binom{1}{3}}(w, x, y, z)$. On comprend alors comment le processus se poursuit avec un nombre de vecteurs plus importants.

Principe de l'algorithme de Bareiss: A l'aide des seules applications $\pi_i^\natural$, on peut calculer le déterminant d'une matrice. Pour ce faire, après chaque combinaison $\pi_i^\natural: (x, y) \mapsto x_i y - y_i x$ de deux vecteurs colonnes de l'étape i, on divise par le pivot de l'étape précédente afin de réduire au plus tôt la taille des coefficients des vecteurs tout en restant dans le module M.

2.2. Encore plus sur l'algorithme de Bareiss

Le lecteur trouvera dans [2] ou [6] des démonstrations de l'algorithme ébauché ci-dessus. La démonstration exposée ici est différente et donne une vision vectorielle sur l'algorithme: ce ne sont plus les coefficients de la matrice qui sont étudiés, mais les vecteurs colonnes qui la composent.

NOTATION: On note $\binom{i}{j}$ la liste **ordonnée** $\{i, i-1, \ldots, j+1, j\}$ si $i \geq j$, ou l'ensemble vide si $i < j$. Si K est une liste **ordonnée** $\{a, b, c, \ldots, z\}$, on pose:

$$\det{}_K = \pi_a \wedge \pi_b \wedge \pi_c \wedge \cdots \wedge \pi_z$$

et par exemple, si $i \geq j$:

$$\det{}_{\binom{i}{j}} = \pi_i \wedge \cdots \wedge \pi_j \qquad \det{}^{\natural}_{\emptyset} = \mathrm{Id}_M$$

Remarquer que la typographie n'est pas innocente: $\det_{\binom{i}{j}}$ est un déterminant dont la première ligne correspond à la projection $\pi_i, \ldots$ la dernière à π_j.

Encore un petit exemple... $\pi_1 \circ \det^{\natural}_{\binom{n}{2}} = \det_{\binom{n}{2}} \wedge \pi_1 = \det_{\binom{n}{1}}$.

Soit $(f_i)_{i \geq 1}$ une suite de 1-formes. On pose $g_i = f_1 \wedge f_2 \wedge \cdots \wedge f_i$ (g_i est donc pure). Par convention $g_0^{\natural} = \mathrm{Id}_M$. On choisit $X \in M^i$ ainsi que $y, z \in M$. On a alors pour $i \geq 1$:

$$\begin{aligned} & f_{i+1}^{\natural}(g_i^{\natural}(X, y), g_i^{\natural}(X, z)) \\ & \quad = g_i(X)(g_i \wedge f_{i+1}^{\natural})(X, y, z) \quad \text{relation } (\star) \text{ page 5} \\ & \quad = (g_{i-1} \wedge f_i)(X)\, g_{i+1}^{\natural}(X, y, z) \\ & \quad = f_i(g_{i-1}^{\natural}(X))\, g_{i+1}^{\natural}(X, y, z). \end{aligned}$$

Pour obtenir une preuve **complète** de l'algorithme de Bareiss (appliqué à une matrice A de dimension $d \times d$), il faudra poser: pour tout $i \in [1, d]$, A_i est la sous-matrice de A formée par ses i premiers vecteurs colonnes, $f_i = \pi_{\alpha_i}$ (et donc $g_i = \pi_{\alpha_1} \wedge \cdots \wedge \pi_{\alpha_i}$) en choisissant $\alpha_i \in [1, d] - E_{i-1}$ (où $E_{i-1} = \{\alpha_1, \ldots, \alpha_{i-1}\}$) tel que $\pi_{\alpha_i}(g_{i-1}^{\natural}(A_i))$ (le pivot de l'étape i) soit non nul. Dans le cas où un tel α_i n'existe pas, on conclut: $\det(A) = 0$

car les vecteurs de A_i sont liés. Si un tel α_i existe, en posant $X = A_i$, la dernière formule donne alors pour $i \geq 1$:

$$\frac{\pi^{\natural}_{\alpha_{i+1}}(\det{}^{\natural}_{E_i}(A_i, y), \det{}^{\natural}_{E_i}(A_i, z))}{\pi_{\alpha_i}(\det{}^{\natural}_{E_{i-1}}(A_i))} = \det{}^{\natural}_{E_{i+1}}(A_i, y, z) = \det{}^{\natural}_{E_{i+1}}(A_{i+1}, z)$$

où y est le $(i+1)$-ième vecteur colonne de A (colonne du pivot de l'étape $i+1$) et z un autre vecteur colonne de A.

A l'étape $d-1$, on voit que l'on calcule $\det{}^{\natural}_{E_{d-1}}(A)$. On peut connaître alors le déterminant de A car :

$$\pi_{\alpha_d}(\det{}^{\natural}_{E_{d-1}}(A)) = \det{}_{E_{d-1}} \wedge \pi_{\alpha_d}(A) = \operatorname{sgn}(i \mapsto E_i) \det(A).$$

Voici l'algorithme calculant le déterminant d'une matrice par la méthode de Bareiss décrite ci-dessus.

Algorithme de Bareiss

Donnée : $A = (x_1, x_2, \ldots, x_d) \in \mathcal{M}_d(R)$
Résultat : $\det(A)$

$E \leftarrow [\ ]$; $oldpivot \leftarrow 1$
for i in $1 \ldots d-1$ loop
  if $x_i = 0$ then return 0
  $(k, pivot) :=$ a Component Of (x_i) – ici, $pivot = \pi_k(x_i) \neq 0$ –
  $E := E \cup [k]$
  for j in $i+1 \ldots d$ loop
    $x_j \leftarrow \frac{pivot \times x_j - \pi_k(x_j) \times x_i}{oldpivot}$
  end loop
  $oldpivot \leftarrow pivot$
end loop – ici, $\forall j > i$, $\forall e \in E$, $\pi_e(x_j) = 0$ –
– ici, $|E| = d-1$ –
for j in $1 \ldots d$ loop
  if $j \notin E$ then return signature$(E \cup [j]) \times \pi_j(x_d)$
end loop

Dans l'algorithme ci-dessus, on utilise deux fonctions qui sont aComponentOf et signature : aComponentOf prend en argument un vecteur x non nul et renvoie le couple formé respectivement par le numéro

et la valeur d'une coordonnée non nulle de x ; signature est une fonction calculant la signature de la permutation $i \mapsto E_i$.

3. LA MEILLEURE ÉLIMINATION

3.1. Dans un module...

Annuler $m-1$ composantes en combinant m vecteurs est l'élimination maximale à laquelle on peut s'attendre dans le cas général : on ne peut pas annuler m composantes dans tous les cas, car alors m vecteurs de dimension m seraient toujours liés. De plus si K est une liste ordonnée d'entiers de cardinal $m-1$, alors $\det_K^\natural$ combine m vecteurs quelconques de manière à annuler les coordonnées indexées par K. La propriété 1 nous le prouve si l'on prend $f = \pi_k$, $k \in K$ et $g = \det_K$. En effet $\pi_k \circ \det_K^\natural = \det_K \wedge \pi_k = 0$ car $k \in K$.

La propriété suivante montre que $\det_K^\natural$ est une application de référence à laquelle on peut comparer toute autre élimination « maximale » ponctuelle.

PROPRIÉTÉ 3 : *Soit* $x \in M^m$, $y = \sum_{i=1}^m \lambda_i x_i \in M$ *et* $K \subset \mathbb{N}$ *tel que* $\forall\, k \in K,\ \pi_k(y) = 0$ *et* $|K| = m-1$. *Alors*

$$\lambda_i \det_K^\natural(x) = \alpha_i\, y \qquad \forall\, i \in [1,\, m]$$

où $\alpha_i = (-1)^{m-1} \det_K (x_1, \ldots, \not{x}_i, \ldots, x_m)$ *est le coefficient de* x_i *dans* $\det_K^\natural(x)$.

Démonstration : On a

$$\begin{aligned}
&\lambda_i \det_K^\natural(x) \\
&\quad = \det_K^\natural(x_1, \ldots, \lambda_i x_i, \ldots, x_m) \quad \text{par linéarité,} \\
&\quad = \det_K^\natural (x_1, \ldots, y, \ldots, x_m) \quad \text{car } \det_K^\natural \text{ est alternée,} \\
&\quad = (-1)^{m-i} \det_K (x_1, \ldots, \not{x}_i, \ldots, x_m)_y \quad \text{car } \forall\, k \in K,\ \pi_k(y) = 0, \\
&\quad = \alpha_i\, y. \quad \square
\end{aligned}$$

3.2. Les sous-résultants

Soit P et Q deux polynômes dans $R[X]$ de degrés respectifs p et q. Étant donné un entier $d \leq \min(p, q)$, on cherche une élimination portant sur les m polynômes $X^{q-d}P, \ldots, XP, P, X^{p-d}Q, \ldots, XQ, Q$ qui permettrait d'annuler les $m-1$ coefficients des monômes de plus hauts degrés. Tous

ces polynômes appartiennent au R-module $\sum_{i=0}^{p+q-d} R \cdot X^i$. D'après le paragraphe 3.1 (si on pose $\pi_i : T \in R[X] \mapsto$ coefficient en X^i de T), il y a une « plus belle » que les autres :

$$\det{}^{\natural}_{\binom{p+q-d}{d}} (X^{q-d}P, \ldots, XP, X^{p-d}Q, \ldots, XQ, Q)$$

On l'appelle **le sous-résultant de P et Q d'indice $d-1$** et on le note $S_{d-1}(P, Q)$.

Par exemple si l'on choisit $p \leq q$ et $d = p$, le sous-résultant considéré est $\det{}^{\natural}_{\binom{q}{p}} (X^{q-p}P, \ldots, XP, P, Q)$. Si l'on calcule ce déterminant en éliminant petit à petit les coefficients de Q, on s'aperçoit que l'on effectue les opérations élémentaires correspondant au calcul du pseudo-reste (abrégé par prem) de Q par P. On obtient finalement l'égalité :

$$S_{p-1}(P, Q) = \det{}^{\natural}_{\binom{q}{p}} (X^{q-p}P, \ldots, XP, P, Q) = \operatorname{prem}(Q, P)$$

4. SOUS-RÉSULTANTS

Le lecteur trouvera dans les ouvrages suivants des démonstrations des principales propriétés des polynômes sous-résultants, ainsi que de diverses méthodes pour prouver l'algorithme de sous-résultants :

Un des premiers articles sur le sujet, [3], donne une démonstration « polynomiale » de l'algorithme : les auteurs amennent progressivement les algorithmes de Collins et des sous-résultants. Le même style d'exposé est repris dans [6] (pages 285 à 299).

Dans [4], Cohen redémontre avec une méthode très concise l'algorithme des sous-résultants : il considère les formules de l'algorithme et montre qu'elle conduisent effectivement au calcul du résultant (en admettant que les divisions dans l'anneau de base sont exactes).

Dans [7], les auteurs s'attachent à définir les sous-résultants et démontrer des relations « susceptibles de passer à travers » les morphismes d'anneaux (spécialisation par exemple). On y voit que les propriétés fondamentales des polynômes sous-résultants.

Dans tous ces documents, la pseudo-division entre deux polynômes est considérée comme une opération élémentaire d'élimination. Le but premier de ce papier est de montrer que l'élimination élémentaire entre deux polynômes est l'opération $\pi_i^{\natural}$ (en considérant ces polynômes comme des vecteurs sur l'anneau de base), et surtout qu'il y a un bénéfice à en tirer.

4.1. Rappels

Dans cette section, les résultats les plus simples sur les sous-résultants seront montrés ou rappelés pour le lecteur non spécialiste. On reviendra sur les relations de similarité et de divisibilité entre les sous-résultants (elles sont déjà bien connues, voir les ouvrages précités ci-dessus) grâce à de nouvelles relations précises qui permettront de justifier non seulement l'algorithme des sous-résultants, mais aussi (et surtout) la variante proposée.

NOTATION : Si $T \in R[X]$ alors on note $X^{\langle i,j\rangle} T\,(i \geq j)$ la suite constituée de $X^i T$, $X^{i-1} T$, ..., X^{j+1}, $X^j T$. Par convention, si $i < j$ alors la suite est vide.

Dans [9], par définition, le d-ième sous-résultant de deux polynômes P et Q (de degrés p et q respectivement) est le déterminant polynômial de la matrice formée par les $X^{\langle q-d-1,0\rangle} P$ et les $X^{\langle p-d-1,0\rangle} Q$. On a donc

$$S_d(P, Q) = \det^{\natural}{}_{\binom{p+q-d-1}{d+1}} (X^{\langle q-d-1,0\rangle} P,\ X^{\langle p-d-1,0\rangle} Q)$$

On voit facilement avec cette écriture que :

• $S_d(P, Q) = (-1)^{(p-d)(q-d)} S_d(Q, P)$;

• $S_d(P, Q)$ est de degré inférieur à d (toutes les coordonnées entre $p+q-d-1$ et $d+1$ sont nulles) ;

• Les coefficients de $S_d(P, Q)$ sont des polynômes en les coefficients de P et Q. Cette propriété est intéressante pour la spécialisation des polynômes sous-résultants par exemple :

• NOTATION :

$$\begin{cases} s_d = \pi_d(S_d(P, Q)) & \text{si} \quad d < \min(p, q), \\ s_d = \mathrm{lc}(P)^{q-p} & \text{si} \quad d = p = \min(p, q), \\ s_d = \mathrm{lc}(Q)^{p-q} & \text{si} \quad d = q = \min(p, q). \end{cases}$$

Remarquer que, de façon générale, s_d est le coefficient en X^d de $S_d(P, Q)$.

Remarquer également que $s_{\min(p,q)} \in R - \{0\}$ et si $p = q = d$ alors $s_d = 1$.

• le polynôme $S_d(P, Q)$ appartient à l'idéal de $R[X]$ engendré par P et Q car combinaison linéaire des $X^{\langle p-d-1,0\rangle} P$ et des $X^{\langle q-d-1,0\rangle} Q$;

Dans cette combinaison linéaire, les coefficients de $X^{q-d-1}P$ et de $X^{p-d-1}Q$ sont respectivement :

$$(-1)^{p-d} \det_{\binom{p+q-d-1}{d+1}} (X^{\langle q-d-2,0\rangle}P,\ X^{\langle p-d-1,0\rangle}Q)$$
$$= (-1)^{p-d} \operatorname{lc}(Q)\, s_{d+1}$$

et

$$(-1)^{p-d-1} \det_{\binom{p+q-d-1}{d+1}} (X^{\langle q-d-1,0\rangle}P,\ X^{\langle p-d-2,0\rangle}Q)$$
$$= (-1)^{p-d-1} \operatorname{lc}(P)\, s_{d+1}$$

DÉFINITION 5 : *On dit qu'un sous-résultant $S_d(P, Q)$ est dégénéré si son degré est strictement inférieur à son indice d.*

Rappel de quelques autres propriétés des sous-résultants (*voir* [7], [5]) :

- $\operatorname{res}(P, Q) = S_0(P, Q)$;
- Dans $K[X]$ (où K est le corps des fractions de R), si

$$d = \deg(\gcd(P, Q))$$

alors $S_d(P, Q)$ est de degré d ;

- Les sous-résultants représentent les meilleures éliminations des coefficients des monômes des plus hauts degrés de deux polynômes (*voir* le paragraphe 3.2, page 9).

Le prochain théorème (théorème 3) permettra à lui seul de démontrer l'algorithme des sous-résultants : en effet, les résultats sur les similarités (section 4.2) et divisibilité (section 4.3) entre sous-résultants n'en sont que des « applications numériques » où les x_i, y_i et z_l sont des polynômes bien choisis. De plus il représente une généralisation du théorème 1 dont nous nous sommes servis pour démontrer l'algorithme de Bareiss. C'est donc ce théorème qui joue le rôle d'une passerelle entre les deux algorithmes.

DÉFINITION 6 : *Soit $x \in M^m$, y, $z \in M^n$. On dit que z est* **échelonné sur** y **modulo** x *si z_j est exprimable linéairement en fonction des x_i, de y_i et des y_l qui ont déjà servi pour les z_l précédents.*

Afin d'illustrer cette définition, voici deux exemples où z est **échelonné sur** y **modulo** x :

- pour $j = 1, 2, \ldots, n$, $z_j \in R \cdot y_i + \sum_{l<j} R \cdot y_l + \sum_{i=1}^{m} R \cdot x_i$;
- pour $j = n \ldots 1$, $z_j \in R \cdot y_i + \sum_{l>j} R \cdot y_l + \sum_{i=1}^{m} R \cdot x_i$.

Il faut souligner que le sens de parcours des valeurs $1 \ldots n$ par j est important.

THÉORÈME 3 : *Soit M et N deux R-modules, $f : M^{n+1} \to N$ une $(n+1)$-application et $g \in \Omega^{m-1}(M)$. On considère $x \in M^m$, $y, z \in M^n$, $\lambda \in R^n$ tel que z soit* **échelonné sur** y **modulo** x *et λ_j soit le coefficient de y_j dans l'écriture de z_j. On suppose de plus que $\forall j \in [1, n]$, $z_j \in \ker g$. Alors on a :*

$$f(g^\natural(x), z) = \left(\prod_{l=1}^{n} \lambda_l\right)(g \wedge f)(x, y).$$

Démonstration :

$$\begin{aligned}
&\left(\prod_{l=1}^{n} \lambda_l\right)(g \wedge f)(x, y) \\
&\quad = (g \wedge f)(x, \lambda_1 \cdot y_1, \ldots, \lambda_n \cdot y_n) \quad \text{par linéarité de } g \wedge f, \\
&\quad = (g \wedge f)(x, z_1, \ldots, z_n) \quad \text{car } g \wedge f \text{ est alternée} \\
&\qquad \text{et } z \text{ échelonné sur } y \text{ modulo } x, \\
&\quad = f(g^\natural(x), z) \quad \text{grâce au théorème 1.} \quad \square
\end{aligned}$$

4.2. Relations de similarité entre sous-résultants

Le but du théorème suivant est de trouver des relations entre les sous-résultants de P et Q afin de mieux comprendre leur structure. On utilisera aussi ces résultats pour donner une variante de l'algorithme des sous-résultants.

LEMME 1 : *Soit $0 < d \leq \min(p, q)$ et $S_{d-1}(P, Q)$ dont le degré e est strictement inférieur à $d - 1$ ($\deg(0) < 0$). On considère $i < d - 1$ et $\delta \in [1, d - e[$. On pose $I = \binom{p+q+\delta-d}{d}$ et $J = \binom{i+\delta}{i+1}$. Alors on a :*

$$\det{}^\natural_J(X^{\langle \delta, 0\rangle} S) = s_d^\delta \det{}^\natural_{I \cup J}(X^{\langle q+\delta-d, 0\rangle} P, X^{\langle p+\delta-d, 0\rangle} Q)$$

Démonstration : La démonstration suivante est inspirée de celle que donne Lazard dans [8]. On pose :

$$f = \det{}^\natural_{\binom{i+\delta}{i+1}} \qquad g = \det{}_{\binom{p+q+\delta-d}{d}} \qquad x' = X^{\langle q-d, 0\rangle} P \cup X^{\langle p-d, 0\rangle} Q$$

$$x = X^{\langle \delta, 1\rangle} X^{q-d} P \cup x' \qquad y = X^{\langle \delta, 1\rangle} X^{p-d} Q \qquad z = X^{\langle \delta, 1\rangle} S$$

Comme $\deg(S) + \delta = e + \delta < d$, chaque z_k est de degré strictement inférieur à d, donc dans le noyau de g. De plus, si $\alpha = (-1)^{p-d} \operatorname{lc}(P)\, s_d$, on a

$$S \in \alpha X^{p-d} Q + \sum_{j<p-d} R \cdot X^j Q + \sum_{j \leq p-d} R \cdot X^j P$$

et en multipliant cette appartenance par $X^k (k \in [1, \delta])$, on voit que z remplit la condition d'échelonnement du théorème 3 et que λ_k est égal à α. Ainsi, en posant $A = \det^{\natural}_I(x)$ et $B = \det^{\natural}_{I \cup J}(x, y)$, le théorème fournit:

$$\det^{\natural}_J (A,\ X^{\langle \delta, 1 \rangle} S) = \alpha^\delta B = s_d^\delta (-1)^{\delta(p-d)} \operatorname{lc}(P)^\delta B$$

Mais $A = \operatorname{lc}(P)^\delta \det^{\natural}_{\binom{p+q-d}{d}} (x') = \operatorname{lc}(P)^\delta S$, si bien que le membre de gauche de l'égalité devient:

$$\det^{\natural}_J (\operatorname{lc}(P)^\delta S,\ X^{\langle \delta, 1 \rangle} S) = \operatorname{lc}(P)^\delta (-1)^\delta \det^{\natural}_J (X^{\langle \delta, 0 \rangle} S)$$

Le membre de droite devient à son tour après permutation des arguments de B:

$$s_d^\delta (-1)^\delta \operatorname{lc}(P)^\delta \det^{\natural}_{I \cup J} (X^{\langle q+\delta-d, 0 \rangle} P,\ X^{\langle p+\delta-d, 0 \rangle} Q)$$

On conclut en simplifiant de chaque côté par $(-1)^\delta \operatorname{lc}(P)^\delta$. □

THÉORÈME 4: *On désigne par S_i le i-ième sous-résultant de P et Q. On suppose que S_{d-1} est de degré $e < d - 1$. Alors:*

1. *S_{d-1} non nul implique*
   (a) $\deg(S_d) = d$ *(où* $d < \min(p, q)$*)*;
   (b) $S_{d-1} \sim S_e$;
   (c) $\operatorname{lc}(S_{d-1})^{d-e} = s_d^{d-e-1} \operatorname{lc}(S_e)$;
   (d) $\forall\, \delta \in [0,\ d - e[,\ \operatorname{lc}(S_{d-1})^\delta\, S_{d-1} \in s_d^\delta \cdot R[X]$;
2. *Si* $s_d \neq 0$ *alors* $\forall\, i \in]e,\ d - 1[,\ S_i = 0$;
3. *Si* $s_d \neq 0$ *et* $S_{d-1} = 0$ *alors* $S_d = \gcd(P, Q)$ *dans* $K[X]$ *où* K *est le corps des fractions de* R, *(ou* $\gcd(P, Q) \in \{P, Q\}$ *si* $d = \min(p, q)$*)*;
4. $\forall\, \delta \in [1,\ d - e[,\ \det^{\natural}_{\binom{e-1+\delta}{e}} (X^{\langle \delta, 0 \rangle} S_{d-1}) \in s_d^\delta \cdot R[X]$.

Remarques:

• Par 1.(*b*) et 1.(*c*), si $p = q = d$ alors S_e est un multiple de S_{d-1} car $s_d = 1$;

• L'assertion 1.(d) est due à Lazard dans [8]. Cette formule sera utilisée pour obtenir la variante proposée de l'algorithme des sous-résultants ;

• Par 3., si $\deg(S_d) = d$ et S_d n'est pas le pgcd de P et Q dans $K[X]$, alors S_{d-1} est non nul ;

• L'assertion 4. servira dans la variante de l'algorithme des sous-résultants.

Démonstration :

• Si on pose $i = e$ alors le lemme précédent nous donne :

$$\forall\, \delta \in [1,\ d-e[,$$
$$\mathrm{lc}(S_{d-1})^{\delta} S_{d-1} = s_d^{\delta} \det{}_K^{\natural} (X^{\langle p+\delta-d,0\rangle} P,\ X^{\langle p+\delta-d,0\rangle} Q).$$

Alors S_{d-1} non nul implique :

– s_d non nul *i.e.* $\deg S_d = d$ si $d < \min(P,\ Q)$ d'où 1.(a) ;

– 1.(b) et 1.(c) en posant $\delta = d - e - 1$;

– 1.(d) en faisant varier δ.

• Si on pose $\delta = d - i - 1$ alors le lemme précédent nous donne :

$$\forall\, i \in]e,\ d-1[,$$
$$0 = s_d^{\delta} \det{}^{\natural}_{\binom{p+q-i-1}{i+1}} (X^{\langle q+\delta-d,0\rangle} P,\ X^{\langle p+\delta-d,0\rangle} Q) = s_d^{\delta}\, S_i.$$

Alors $s_d \neq 0$ implique 2. L'assertion 3. n'est qu'une conséquence de 2. lorsque $e < 0$ (*voir* les rappels sur les sous-résultants).

• Le lemme donne exactement la propriété 4. si l'on pose $i = e - 1$. □

4.3. Relations de divisibilité entre sous-résultants

Dans cette section, nous allons démontrer l'algorithme des sous-résultants. C'est un algorithme qui calcule le résultant de deux polynômes grâce à une suite de pseudo-divisions euclidiennes de certains de leurs sous-résultants. Ici encore, des nouvelles relations de divisibilité seront démontrées : elles seront utilisées dans la variante de l'algorithme des sous-résultants.

On choisit P et Q dans $R[X]$, avec $p = \deg(P) \geq \deg(Q) = q$. On abrègera encore le i-ième sous-résultant de P et Q par S_i.

Premier cas : le premier sous-résultant calculé. On a toujours :

$$\begin{aligned}\mathrm{prem}(P,\ -Q) &= \det{}^{\natural}_{\binom{p}{q}} (-X^{\langle p-q,0\rangle} Q,\ P) \\ &= \det{}^{\natural}_{\binom{p}{q}} (P,\ X^{\langle p-q,0\rangle} Q) = S_{q-1}\end{aligned}$$

Deuxième cas : le second sous-résultant calculé.

LEMME 2 : *Si S_{q-1} est non nul et de degré e, on considère $\delta \in [0, q-e]$ et on pose $I = \binom{p+\delta}{q+1}$ et $J = \binom{e+\delta}{e}$. Alors :*

$$\det{}_J^{\natural}(Q, X^{\langle\delta,0\rangle}S_{q-1}) = \mathrm{lc}(Q)^{(p-q)\delta+1} \det{}_{I\cup J}^{\natural}(X^{\langle\delta,0\rangle}P, X^{\langle p-q+\delta,0\rangle}Q)$$

Démonstration : On pose $A = \det{}_{I\cup J}^{\natural}(X^{\langle p-q+\delta,0\rangle}Q, X^{\langle\delta,0\rangle}S_{q-1})$. Comme le coefficient de P dans S_{q-1} est $(-1)^{p-q+1}\mathrm{lc}(Q)^{p-q+1}$, et qu'il en est de même pour celui de X^kP dans X^kS_{q-1} pour $k \in [0, \delta]$, il vient

$$\begin{aligned} A &= ((-1)^{p-q+1}\mathrm{lc}(Q)^{p-q+1})^{\delta+1} \det{}_{I\cup J}^{\natural}(X^{\langle p-q+\delta,0\rangle}Q, X^{\langle\delta,0\rangle}P) \\ &= \mathrm{lc}(Q)^{(p-q+1)(\delta+1)} \det{}_{I\cup J}^{\natural}(X^{\langle\delta,0\rangle}P, X^{\langle p-q+\delta,0\rangle}Q). \end{aligned}$$

De plus $A = \mathrm{lc}(Q)^{p-q+\delta} \det{}_J^{\natural}(Q, X^{\langle\delta,0\rangle}S_{q-1})$. On obtient alors l'égalité anoncée en simplifiant par $\mathrm{lc}(Q)^{p-q+\delta}$. □

THÉORÈME 5 : *Si S_{q-1} est non nul et de degré e, alors*

1. $\forall\, \delta \in [0, q-e]$, $\det{}_{\binom{e+\delta}{e}}^{\natural}(Q, X^{\langle\delta,0\rangle}S_{q-1}) \in \mathrm{lc}(Q)^{(p-q)\delta+1} \cdot R[X]$;
2. $\mathrm{prem}(Q, -S_{q-1}) = \mathrm{lc}(Q)^{(p-q)(q-e)+1}S_{e-1}$.

Démonstration : Le 1. est immédiat à partir du lemme précédent. Pour obtenir 2. en considérant le lemme, il faut poser $\delta = q - e$. □

Remarque : le 1. sera utilisé dans la variante de l'algorithme des sous-résultants.

Troisième cas : les autres sous-résultants.

LEMME 3 : *On choisit $d < q$ tel que S_d et S_{d-1} soient respectivement non dégénéré et non nul. On a $d = \deg(S_d)$ et on pose $e = \deg(S_{d-1})$, $\delta \in [0, d-e]$, ainsi que $I = \binom{p+q-d+\delta}{d+1}$ et $J = \binom{e+\delta}{e}$. Alors :*

$$\det{}_J^{\natural}(S_d, X^{\langle\delta,0\rangle}S_{d-1}) = \mathrm{lc}(S_d)^{\delta+1} \det{}_{I\cup J}^{\natural}(X^{\langle q-d+\delta,0\rangle}P, X^{\langle p-d+\delta,0\rangle}Q)$$

Démonstration : On pose :

$$f = \det{}_{\binom{e+\delta}{e}}^{\natural} \qquad g = \det{}_{\binom{p+q-d+\delta}{d+1}} \qquad x' = X^{\langle q-d-1,0\rangle}P \cup X^{\langle p-d-1,0\rangle}Q$$

$$x = X^{\langle\delta,0\rangle}X^{q-d}P \cup x' \qquad y = X^{\langle\delta,0\rangle}X^{p-d}Q \qquad z = X^{\langle\delta,0\rangle}S_{d-1}$$

Chaque z_k est de degré strictement inférieur à $d+1$, donc dans le noyau de g. De plus, si $\alpha = (-1)^{p-d} \operatorname{lc}(P)\, s_d$, on a

$$S_{d-1} \in \alpha\, X^{p-d} Q + \sum_{j<p-d} R \cdot X^j Q + \sum_{j \le q-d} R \cdot X^j P$$

et en multipliant cette appartenance par X^k ($k \in [0,\ \delta]$), on voit que z remplit la condition d'échelonnement du théorème 3 et que λ_k est égal à α. Ainsi, en posant $A = \det^{\natural}_I(x)$ et $B = \det^{\natural}_{I \cup J}(x,\ y)$, le théorème fournit :

$$\det{}^{\natural}_J\,(A,\ X^{\langle \delta, 0\rangle} S_{d-1}) = \alpha^{\delta+1} B = \operatorname{lc}(S_d)^{\delta+1} (-1)^{(\delta+1)(p-d)} \operatorname{lc}(P)^{\delta+1} B$$

Mais $A = \operatorname{lc}(P)^{\delta+1} \det^{\natural}_{\binom{p+q-d-1}{d+1}}(x') = \operatorname{lc}(P)^{\delta+1} S_d$, si bien que le membre de gauche de l'égalité devient :

$$\operatorname{lc}(P)^{\delta+1} \det{}^{\natural}_J(S_d,\ X^{\langle \delta, 0\rangle} S_{d-1})$$

Le membre de droite devient à son tour après permutation des arguments de B :

$$\operatorname{lc}(S_d)^{\delta+1} \operatorname{lc}(P)^{\delta+1} \det{}^{\natural}_{I \cup J}\,(X^{\langle q-d+\delta, 0\rangle} P,\ X^{\langle p-d+\delta, 0\rangle} Q)$$

On obtient alors le résultat annoncé en simplifiant par $\operatorname{lc}(P)^{\delta+1}$. □

THÉORÈME 6 : *Toujours avec les mêmes notations :*

1. $\forall\, \delta \in [0,\ d-e]$, $\det^{\natural}_{\binom{e+\delta}{e}}(S_d,\ X^{\langle \delta, 0\rangle} S_{d-1}) \in \operatorname{lc}(S_d)^{\delta+1} \cdot R[X]$;
2. $\operatorname{prem}(S_d,\ S_{d-1}) = \operatorname{lc}(S_d)^{d-e+1}\, S_{e-1}$;
3. *si* S_{c-1} *est le sous-résultant (éventuellement dégénéré) similaire à* S_d *alors* $\operatorname{prem}(S_{c-1},\ -S_{d-1}) = \operatorname{lc}(S_{c-1}) \operatorname{lc}(S_d)^{d-e}\, S_{e-1}$.

Démonstration : Le 1. s'obient directement du lemme précédent. Le 2. vient lorsque l'on pose $\delta = d-e$. Quant au 3., comme S_{c-1} et S_d sont similaires, on a $\operatorname{lc}(S_d) \operatorname{prem}(S_{c-1},\ \ldots) = \operatorname{lc}(S_{c-1}) \operatorname{prem}(S_d,\ \ldots)$. En utilisant le 2., on conclut facilement. □

Remarque : le 1. sera utilisé dans la variante de l'algorithme des sous-résultants.

5. ALGORITHMES

Ici encore, on choisit deux polynômes P et Q dans $R[X]$ et on note S_i leur i-ième sous-résultant. Dans la section 5.1, on traite rapidement l'algorithme

des sous-résultants (toutes les relations mathématiques utilisées par cette méthode sont explicitées dans les théorèmes 4, 5 et 6).

La section 5.2 est consacrée exclusivement au développement de la variante proposée de l'algorithme des sous-résultants. Cette section est beaucoup plus importante car toutes les relations utiles à cette méthode n'ont pas été exposées dans les sections précédentes.

5.1. L'algorithme des sous-résultants

L'algorithme des sous-résultants est maintenant totalement démontré: à l'aide des sous-résultants éventuellement dégénérés notés S_{i-1}, on calcule le résultant de deux polynômes de $R[X]$.

1. On se donne deux polynômes P et Q, le degré de P étant supérieur à celui de Q (échanger P et Q si nécessaire).

2. Calcul direct de S_{q-1} (de degré d) par $\operatorname{prem}(P, -Q)$.

3. Calcul de S_{d-1} (de degré e) en fonction de Q, S_{q-1} et $\deg(P)$ (théorème 5.2).

4. Calcul de $\operatorname{lc}(S_d)$ en fonction de Q, S_{q-1} et $\deg(P)$ (théorème 4.1.(*c*)). Calcul de S_{e-1} (de degré f) en fonction de S_{q-1}, S_{d-1} et $\operatorname{lc}(S_d)$ (théorème 6.3).

5. Calcul de $\operatorname{lc}(S_e)$ en fonction de $\operatorname{lc}(S_d)$, S_{d-1} et $\deg(S_{q-1})$ (théorème 4.1.(*c*)). Calcul de S_{f-1} en fonction de S_{d-1}, S_{e-1} et $\operatorname{lc}(S_e)$ (théorème 6.3).

6. Répéter le 5. jusqu'au calcul de $\operatorname{lc}(S_0)\ldots$

7. $\operatorname{res}(P, Q) = \operatorname{lc}(S_0)$ (théorème 4.1(*c*) ou 4.2).

Algorithme des sous-résultants

Données: $P, Q \in R[X]$ $\quad \deg(P) \geq \deg(Q) \geq 1$
Résultat: $\operatorname{res}(P, Q)$

$\delta \leftarrow \deg(P) - \deg(Q)$; $(P, Q) \leftarrow (Q, \operatorname{prem}(P, -Q))$; $s \leftarrow \operatorname{lc}(P)^{\delta}$
loop – ici, $Q = S_{e-1}$, $s = \operatorname{lc}(S_e)$, $\delta = \deg(S_{e-1}) - \deg(S_{d-1}) -$
if $Q = 0$ then return 0
$\delta \leftarrow \deg(P) - \deg(Q)$
if $\deg(Q) = 0$ then return $\frac{\operatorname{lc}(Q)^{\delta}}{s^{\delta-1}}$
$(P, Q) \leftarrow \left(Q, \frac{\operatorname{prem}(P, -Q)}{\operatorname{lc}(P) \cdot s^{\delta}}\right)$; $s \leftarrow \frac{\operatorname{lc}(P)^{\delta}}{s^{\delta-1}}$
end loop – ici, $P = S_{d-1}$ –

5.2. Une variante de l'algorithme des sous-résultants...

Un deuxième algorithme est considéré. C'est un algorithme dont les calculs sont différents de ceux de l'algorithme des sous-résultants, afin d'éviter une croissance inutile des différentes quantités calculées :

• Le calcul du coefficient dominant de S_e par le théorème 4.1(c) fait apparaître une fraction de deux éléments de R avec certains exposants. Il est coûteux de calculer d'abord deux puissances puis de les « éliminer » en les divisant. Un calcul économique est donné par le 1.(d) du même théorème (considérer le coefficient dominant de S_{d-1} et S_e) :

$$\mathrm{lc}\,(S_e) = \frac{\mathrm{lc}\,(S_{d-1})^{d-e}}{s_d^{d-e-1}} = \frac{\dfrac{\dfrac{\mathrm{lc}\,(S_{d-1})^2}{s_d} \times \mathrm{lc}\,(S_{d-1})}{s_d} \times \cdots \times \mathrm{lc}\,(S_{d-1})}{s_d}$$

où tous les quotients sont exacts. Dans le terme de droite, seuls des éléments de « hauteur » 1 ou 2 apparaissent : après chaque multiplication, on effectue une simplification.

• Le calcul de S_{e-1} par le théorème 6 est aussi coûteux. Après avoir calculé le pseudo-reste de S_{c-1} par S_{d-1}, on le divise par un élément de R de « hauteur » $d - e + 1$, qui peut être énorme... Pour rendre économique le calcul de S_{e-1}, il faudrait commencer à diviser pendant que l'on effectue les éliminations élémentaires de la pseudo-division. Mais en fait, le principe de la pseudo-division est ici mal adapté. En effet, il se peut que l'on ne puisse pas réduire les restes intermédiaires de la pseudo-division de S_{c-1} par S_{d-1}.

Par un principe de calcul différent (qui sera exposé plus tard), on peut effectuer une simplification après chaque élminiation. Ce qui a pour effet de générer des termes de « hauteur » 2 au maximum...

Voici le principe de l'algorithme économique (la chaîne de tous les polynômes sous-résultants non nuls y sont calculés) :

1. On se donne deux polynômes P et Q, le degré de P étant supérieur à celui de Q (échanger P et Q si nécessaire).

2. Calcul direct de S_{q-1} (de degré d) par $\mathrm{prem}\,(P, -Q)$.

3. Calcul de S_d en fonction de Q, S_{q-1} et $\deg(P)$ (théorème 4).
   Calcul de S_{d-1} (de degré e) en fonction de Q et S_{q-1} (théorème 5.2).

4. Calcul de S_e en fonction de S_d et S_{d-1} (théorème 4).
   Calcul de S_{e-1} en fonction de S_d et S_{d-1} (théorème 6.2).
5. Répéter 4. jusqu'au calcul de S_0...
6. $\operatorname{res}(P, Q) = S_0$.

Pour rendre économique le calcul de S_e en fonction de S_d et S_{d-1}, il suffit d'utiliser le théorème 4.1.(*d*). On obtient alors:

$$S_e = \frac{\dfrac{\dfrac{\operatorname{lc}(S_{d-1})^2}{s_d} \times \operatorname{lc}(S_{d-1})}{s_d} \times \dfrac{\ddots \times \operatorname{lc}(S_{d-1})}{s_d} \times S_{d-1}}{s_d}$$

où tous les quotients sont exacts.

Calcul de S_e
Données: d, s_d, S_{d-1} Résultat: S_e
if $d-1 = \deg(S_{d-1})$ then return S_{d-1} $y \leftarrow \operatorname{lc}(S_{d-1})$ for j in $\deg(S_{d-1})+1 \ldots d-2$ loop $\quad y \leftarrow \frac{y \cdot \operatorname{lc}(S_{d-1})}{s_d}$ end loop return $\frac{y \cdot S_{d-1}}{s_d}$

Il est moins facile de rendre économique le calcul de S_{e-1} en fonction de S_d et S_{d-1}. C'est l'objet de tout ce qui suit jusqu'à la fin de la section 5.2.

PROPRIÉTÉ 4: *Soit K une suite ordonnée d'entiers et E un sous-ensemble de $R[X]$ tel que $|E| = 1 + |K|$. Alors:*

$$\det^{\natural}_{K+1}(X \cdot E) = X \cdot \det^{\natural}_{K}(E)$$

Démonstration: Si T est un polynôme de $R[X]$, on a $\pi_{i+1}(X \cdot T) = \pi_i T$ et $\mathrm{Id}_{R[X]}(X \cdot T) = X \cdot \mathrm{Id}_{R[X]}(T)$. Ainsi

$$\begin{aligned}\det^{\natural}_{K+1}(X \cdot E) &= \left(\bigwedge \pi_{i+1}\right) \wedge \mathrm{Id}_{R[X]}(X \cdot E) \\ &= \left(\bigwedge \pi_i\right) \wedge X \cdot \mathrm{Id}_{R[X]}(E) = X \cdot \det^{\natural}_{K}(E). \quad \square\end{aligned}$$

LEMME 4: *Soit $T \in R[X]$ de degré e et $i \geq e$. Alors*

$$\det^{\natural}_{\binom{i}{e}}(X^{\langle i-e+1,0\rangle}T) = \pi^{\natural}_e(X \det^{\natural}_{\binom{i-1}{e}}(X^{\langle i-e,0\rangle}T),\ T)$$

Démonstration:

$$\begin{aligned}\det^{\natural}_{\binom{i}{e}}(X^{\langle i-e+1,1\rangle}T,\ T) &= \pi^{\natural}_e(\det^{\natural}_{\binom{i}{e+1}}(X^{\langle i-e+1,1\rangle}T),\ T)\\ &\quad \text{grâce au théorème 1}\\ &= \pi^{\natural}_e(\det^{\natural}_{\binom{i-1}{e}}(X^{\langle i-e,0\rangle}T),\ T)\\ &\quad \text{grâce au théorème 4.}\ \square\end{aligned}$$

THÉORÈME 7: *Soit $d \leq \min(p,\ q)$. On suppose S_{d-1} non nul et de degré e. On définit les polynômes B_δ par:*

$$B_\delta = \frac{\det^{\natural}_{\binom{e+\delta-1}{e}}(X^{\langle\delta,0\rangle}\,S_{d-1})}{s_\delta^d} \qquad \forall\,\delta \in [0,\ d-e]$$

Si $\delta < d-e$ alors les polynômes B_δ sont dans $R[X]$ et on a la relation de récurrence pour $\delta \in [0,\ d-e[$:

$$s_d\,B_{\delta+1} = \pi^{\natural}_e(X \cdot B_\delta,\ S_{d-1})$$

Démonstration: Le théorème 4.4 indique que les polynômes B_δ sont dans $R[X]$ si $\delta < d-e$. Si l'on substitue S_{d-1} à T et l'on pose $\delta = i-e \in [0, d-e[$, le lemme précédent donne:

$$\det^{\natural}_{\binom{e+\delta}{e}}(X^{\langle\delta+1,0\rangle}\,S_{d-1}) = \pi^{\natural}_e(X\ \det^{\natural}_{\binom{e+\delta-1}{e}}(X^{\langle\delta,0\rangle}S_{d-1}),\ S_{d-1})$$

Or le membre de droite de cette égalité n'est autre que $\pi^{\natural}_e(s_d^\delta X \cdot B_\delta,\ S_{d-1})$. On termine la preuve du résultat annoncé en divisant par s_d^δ. $\square$

THÉORÈME 8: *Soit $d < q = \min(p,\ q)$ tel que $d = \deg(S_d)$. On pose $e = \deg(S_{d-1})$ et on définit les polynômes A_δ par:*

$$A_\delta = \frac{\det^{\natural}_{\binom{e+\delta}{e}}(S_d,\ X^{\langle\delta,0\rangle}S_{d-1})}{\mathrm{lc}(S_d)^{\delta+1}} \qquad \forall\,\delta \in [0,\ d-e]$$

Alors les polynômes A_δ sont tous dans $R[X]$, $A_{d-e} = S_{e-1}$ et on a la relation de récurrence pour $\delta \in [1, d-e]$:

$$A_\delta = \frac{\pi_{e+\delta}(S_d)\, B_\delta - \mathrm{lc}\,(S_{d-1})\, A_{\delta-1}}{\mathrm{lc}\,(S_d)}$$

Démonstration : Le théorème 6.1 indique que A_δ appartient à $R[X]$, le théorème 6.2 donne

$$A_{d-e} = \frac{\mathrm{prem}\,(S_d,\ -S_{d-1})}{\mathrm{lc}\,(S_d)^{d-e+1}} = S_{e-1}$$

En développant $\pi_{e+\delta} \wedge \det^{\natural}_{\binom{e+\delta-1}{e}}$, on obtient

$$\begin{aligned}
&\det^{\natural}_{\binom{e+\delta}{e}}(S_d,\ X^{\langle\delta,0\rangle} S_{d-1}) \\
&\quad = \pi_{e+\delta}(S_d)\ \det^{\natural}_{\binom{e+\delta-1}{e}}(X^{\langle\delta,0\rangle} S_{d-1}) \\
&\qquad - \mathrm{lc}\,(S_{d-1})\ \det^{\natural}_{\binom{e+\delta-1}{e}}(S_d,\ X^{\langle\delta-1,0\rangle} S_{d-1})
\end{aligned}$$

En divisant par $\mathrm{lc}\,(S_d)^\delta$ cette égalité ($\delta \in [1, d-e]$), le membre de gauche devient $\mathrm{lc}\,(S_d)\, A_\delta$, le premier terme du membre de droite se simplifie en B_δ et le second devient $\mathrm{lc}\,(S_{d-1})\, A_{\delta-1}$. On arrive ainsi à la formule annoncée. □

Maintenant il est clair que, pendant le calcul séquentiel des A_δ (pour $\delta \in [0, d-e]$) et des B_δ (où $\delta \in [0, d-e[$), les éléments générés ont une « hauteur » ne dépassant pas 2.

Calcul de S_{e-1}

Données : S_d, S_{d-1}
Résultat : S_{e-1}

$(e, r, s) \leftarrow (\deg(S_{d-1}),\ \mathrm{lc}\,(S_{d-1}),\ \mathrm{lc}\,(S_d))$
$(A, B) \leftarrow \left(\frac{\pi_e(S_d)\cdot S_{d-1} - r\cdot S_d}{s},\ S_{d-1}\right)$
for i in $e+1 \ldots \deg(S_d) - 1$ loop – ici, $A = A_{i-e-1}$, $B = B_{i-e-1}$ –
  $B \leftarrow \frac{\pi_{e-1}(B)\cdot S_{d-1} - r\cdot X\cdot B}{s}$; $A \leftarrow \frac{\pi_i(S_d)\cdot B - r\cdot A}{s}$
end loop – ici, $A = A_{i-e}$, $B = B_{i-e}$ –
return $\frac{\pi_{e-1}(B)\cdot S_{d-1} - r\cdot(X\cdot B + A)}{s}$

Cas exceptionnel: le cas du second sous-résultant calculé S_{d-1} éventuellement dégénéré (*voir* le théorème 5) peut être traité ainsi:

THÉORÈME 9: *Soit* $d = q = \min(p, q)$ *et* $e = \deg(S_{q-1})$. *On pose:*

$$A_\delta = \frac{\det^{\natural}_{\binom{e+\delta}{e}}(Q,\ X^{\langle\delta,0\rangle} S_{q-1})}{\mathrm{lc}(Q)^{(p-q)\delta+1}} \qquad \forall\, \delta \in [0,\ q-e]$$

Alors les polynômes A_δ *sont tous dans* $R[X]$, $A_{q-e} = S_{e-1}$ *et on a la relation de récurrence pour* $\delta \in [1,\ q-e]$*:*

$$A_\delta = \frac{\pi_{e+\delta}(Q)\, B_\delta - \frac{\mathrm{lc}(S_{q-1})\, A_{\delta-1}}{\mathrm{lc}(Q)^{p-q-1}}}{\mathrm{lc}(Q)}$$

Démonstration: On s'inspire complètement de la démonstration du théorème précédent. Le théorème 5.1 indique que A_δ appartient à $R[X]$ et le théorème 5.2 donne en particulier

$$A_{q-e} = \frac{\mathrm{prem}(Q,\ -S_{q-1})}{\mathrm{lc}(Q)^{(p-q)(q-e)+1}} = S_{e-1}$$

En développant $\pi_{e+\delta} \wedge \det^{\natural}_{\binom{e+\delta-1}{d}}(Q,\ X^{\langle\delta,0\rangle} S_{q-1})$, puis en divisant par $\mathrm{lc}(Q)^{(p-q)\delta+1}$, on obtient les résultats annoncés. □

Calcul de S_{e-1} (exception)

Données: Q, S_{q-1}, $\delta = \deg(P) - \deg(Q)$
Résultat: S_{e-1}

$(e, r, t, s) \leftarrow (\deg(S_{q-1}),\ \mathrm{lc}(S_{q-1}),\ \mathrm{lc}(Q),\ \mathrm{lc}(Q)^\delta)$
$(A, B) \leftarrow \left(\frac{\pi_d(Q)\cdot S_{q-1} - r\cdot Q}{t},\ S_{q-1}\right)$
for i in $e+1 \ldots \deg(Q) - 1$ loop – ici, $A = A_{i-e-1}$, $B = B_{i-e-1}$ –
  $B \leftarrow \frac{\pi_{e-1}(B)\cdot S_{q-1} - r\cdot X\cdot B}{s}$; $A \leftarrow \frac{\pi_i(Q)\cdot B - \frac{r\cdot A}{t^{\delta-1}}}{t}$
end loop – ici, $A = A_{i-e}$, $B = B_{i-e}$ –
return $\frac{\pi_{e-1}(B)\cdot S_{q-1} - r\cdot(X\cdot B + A)}{s}$

Voici maintenant le corps du nouvel algorithme faisant appel aux petites procédures vues ci-dessus:

Algorithme économique

Données: $P, Q \in R[X]$ $\quad \deg(P) \geq \deg(Q) \geq 1$
Résultat: $\text{res}(P, Q)$

$\delta \leftarrow \deg(P) - \deg(Q)$; $P \leftarrow \text{prem}(P, -Q)$
if $P = 0$ then return 0
$Z \leftarrow$ *calcul de* S_e *avec* $\deg(Q)$, $\text{lc}(Q)^{\delta}$, P
if $\deg(Z) = 0$ then return $\text{lc}(Z)$
$P \leftarrow$ *calcul de* S_{e-1} *avec* Q, P, δ
loop
  $Q \leftarrow Z$ $\quad$ – ici, $Q = S_d$, $P = S_{d-1}$ –
  if $P = 0$ then return 0
  $Z \leftarrow$ *calcul de* S_e *avec* $\deg(Q)$, $\text{lc}(Q)$, P
  if $\deg(Z) = 0$ then return $\text{lc}(Z)$
  $P \leftarrow$ *calcul de* S_{e-1} *avec* Q, P
end loop

6. IMPLÉMENTATION ET EXPÉRIMENTATION

6.1. Une dernière amélioration

L'algorithme économique ne doit pas être implémenté tel qu'il a été présenté dans le paragraphe précédent: en effet lorsque l'on parcourt la procédure calculant S_{e-1} en fonction de S_d et S_{d-1}, les polynômes A et B générés sont de degrés supérieurs à $e = \deg(S_{d-1})$. En fait on peut limiter les calculs en utilisant les polynômes de degrés strictements inférieurs à e...

Ce qui suit n'apporte rien mathématiquement, mais au prix d'un dernier théorème, les calculs en machines seront accélérés.

NOTATION: Soit $K \subset \mathbb{N}$. On nomme Π_K la projection linéaire qui à un polynôme $T = \sum_i t_i X^i$ associe $\sum_{i \in K} t_i X^i$.

THÉORÈME 10: *Soit* $d \leq \min(p, q)$ *tel que* $d = \deg(S_d)$ *si* $d < \min(p, q)$. *On suppose* S_{d-1} *non nul. On pose* $e = \deg(S_{d-1})$ *et* $K = [0, e-1]$. *En reprenant les notations des théorèmes* 7, 8 *et* 9, *on considère* $C_\delta = \Pi_K(A_\delta)$

et $D_\delta = \Pi_K(B_\delta)$ *pour tout* $\delta \in [0, d-e]$. *Dans ces conditions on a* $C_{d-e} = S_{e-1}$ *et les relations de récurrence pour* $\delta \in [1, d-e]$ *:*

- $D_\delta = \frac{\pi_{e-1}(D_{\delta-1})\, D_0 - \pi_e(S_{d-1})\, \Pi_K(X D_{\delta-1})}{\mathrm{lc}(S_d)}$
- *Si* $d = q = \min(p, q)$ *alors* $C_\delta = \frac{\pi_{e+\delta}(Q)\, D_\delta - \frac{\mathrm{lc}(S_{q-1})\, C_{\delta-1}}{\mathrm{lc}(Q)^{p-q-1}}}{\mathrm{lc}(Q)}$
- *Si* $d < \min(p, q)$ *alors* $C_\delta = \frac{\pi_{e+\delta}(S_d)\, D_\delta - \mathrm{lc}(S_{d-1})\, C_{\delta-1}}{\mathrm{lc}(S_d)}$

Démonstration: Pour obtenir les formules de récurrence concernant C_δ, il suffit de prendre les formules des théorèmes 8 et 9 et de leur appliquer Π_K (les expressions sont linéaires). De même pour obtenir la formule de récurrence concernant D_δ, il faut prendre la formule du théorème 7 et lui appliquer Π_K :

$$\begin{aligned} s_d\, D_\delta &= s_d\, \Pi_K(B_\delta) = \Pi_K \circ \pi_e^\natural(X \cdot B_{\delta-1},\ S_{d-1}) \\ &= \pi_e(X \cdot B_{\delta-1})\, \Pi_K(S_{d-1}) - \pi_e(S_{d-1})\, \Pi_K(X \cdot B_{\delta-1}) \\ &= \pi_{e-1}(D_{\delta-1})\, D_0 - \mathrm{lc}(S_{d-1})\, \Pi_K(X \cdot D_{\delta-1}). \quad \square \end{aligned}$$

6.2. Résultats de tests sur machines

Afin de rendre compte de l'efficacité de cette variante de l'algorithme des sous-résultants, nous donnons juste deux séries de tests.

Pour la première série, on reprend les tests fournis dans [1] (pages 268 à 279):

test 1 $P = X^8 + X^6 - 3X^4 - 3X^3 + 8X^2 + 2X - 5$
$Q = 3X^6 + 5X^4 - 4X^2 - 9X + 21$

test 2 $P = 3X^4 + 5X^3 + 5X^2 - 2X + 1$
$Q = 3X^3 + 3X^2 + 3X - 4$

test 3 $P = 2X^6 + X^5 + 4X^4 + 3X^3 + 5X^2 - 2X + 2$
$Q = X^6 - X^5 - X^4 + 4X^3 - 2X^2 + X + 1$

test 4 $P = X^6 + X^5 - X^4 - X^3 + X^2 - X + 1$
$Q = 6X^5 + 5X^4 - 4X^3 - 3X^2 + 2X - 1$

test 5 $P = 3X^9 + 5X^8 + 7X^7 - 3X^6 - 5X^5 - 7X^4 + 3X^3 + 5X^2 + 7X - 2$
$Q = X^8 - X^5 - X^2 - X - 1$

test 6 $P = 3X^8 + 4X^7 - 7X^5 - 9X^3 + X^2 - X + 2$
$Q = 3X^5 + 12X^4 - 7X^2 + 1$

test 7 $P = X^5 + 5X^4 + 10X^3 + 5X^2 + 5X + 2$
$Q = X^4 + 4X^3 + 6X^2 + 2X + 1$

test 8 $P = 2X^4 + 3X^3 + 5X^2 - 2X - 1$
$Q = X^3 + X^2 + 2X - 1$

test 9 $P = 6X^8 + 7X^7 - 3X^6 + 16X^5 + 20X^4 + 8X^3 + 17X^2 + 6X + 18$
$Q = 2X^7 + X^6 + 3X^5 + X^4 - 2X^3 + 9X^2 - 7X + 3$

test 10 $P = 2X^4 - X^3 - 14X^2 + 17X - 5$
$Q = 6X^3 - 7X^2 + 4X - 1$

Le tableau suivant résume les performances de trois algorithmes étudiés en ce qui concerne les tailles (*i.e.* le nombre de chiffres) des plus grands entiers générés sur chaque test. Bien sûr, il n'y est pas question avec ces premiers tests de comparer des temps de calculs car ceux-ci sont trop petits (en ce qui concerne les algorithmes des sous-résultants et sa variante) pour être significatifs.

Nous donnons aussi un jeu de calculs de résultants avec des polynômes plus ou moins paramétrés et de degrés plus importants. Ici, nous comparons les temps de réponse de l'algorithme des sous-résultants et de sa variante. L'implémentation des algorithmes est faite en Axiom version 2.0a (*resultant* est la fonction standard d'Axiom utilisant l'algorithme des sous-résultants).

Tailles maximales des entiers générés.

	Algorithme des sous-résultants	Algorithme économique	Algorithme de Bareiss
test 1	11	9	10
test 2	6	6	7
test 3	16	11	12
test 4	11	8	10
test 5	23	16	17
test 6	28	20	21
test 7	5	5	6
test 8	1	1	2
test 9	21	16	18
test 10	11	9	10

test 11 $P = aX^6 + bX^5 + cX^4 + dX^3 + eX^2 + fX + g$
$Q = P'$

test 12 $P = X^5 + aX^4 + bX^3 + cX^2 + dX + e$
$Q = X^5 + fX^4 + gX^3 + hX^2 + iX + j$

test 13 $P = X^7 + aX^3 + bX^2 + cX + d$
$Q = X^7 + eX^3 + fX^2 + gX + h$

test 14 $P = X^{20} + aX^{15} + b$
$Q = X^{20} + cX^5 + d$

test 15 $P = (X + a)^{15}$
$Q = (X + z)^{15}$

test 16 $P = X^{30} + aX^{20} + 2aX^{10} + 3a$
$Q = X^{25} + 4bX^{15} + 5bX^5$

test 17 $P = (a + X)^{90}$

$Q = (a - X)^{60}$

test 18 $P = \sum_{j=0}^{75} a^{75-j} X^j$

$Q = \sum_{j=0}^{75} j a^j X^j$

test 19 $P = \sum_{j=0}^{200} X^j$

$Q = 1 + \sum_{j=0}^{100} j X^j$

test 20 $P = 1 + \sum_{j=1}^{900} j X^j$

$Q = 1 + \sum_{j=1}^{900} j^2 X^j$

Le tableau suivant résume les performances des deux algorithmes. Les temps sont exprimés en secondes. La machine utilisée pour tester les deux algorithmes est une station Sun Sparc-10 SunOS 4.1.3, 4 × 72 Mhz, 512 Mo de mémoire centrale.

Temps de calculs (en secondes).

	resultant (Axiom)	Algorithme économique		resultant (Axiom)	Algorithme économique
test 11	187	22	test 16	2 184	136
test 12	7 345	246	test 17	257	204
test 13	2 881	177	test 18	4 397	69
test 14	3 590	202	test 19	265	10
test 15	1 491	877	test 20	1 125	33

RÉFÉRENCES

1. A. G. AKRITAS, *Elements of Computer Algebra with Applications,* John Wiley and Sons, 1989.
2. E. H. BAREISS, Sylverter's Identity and Multistep Integer-Preserving Gaussian Elimination, *Math. Comp.*, 1968, *22*, p. 565-578.
3. W. S. BROWN et J. F. TRAUB, On Euclid's Algorithm and Theory of Subresultants, *Ass. Comp. Mach.*, Octobre 1971, *18* (4), p. 505-514.
4. H. COHEN, *A Course in Computational Algebraic Number Theory*, ch. 3, Springer-Verlag, 1993, p. 116-123.
5. L. DUCOS, Calcul du résultant et du pgcd dans les anneaux de polynômes, *Mémoire de D.E.A.*, juin 1994.
6. G. LABAHN, K. O. GEDDES et S. R. CZAPOR, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1992.
7. T. RECIO, L. GONZÁLEZ-VEGA, H. LOMBARDI et M.-F. ROY, Spécialisation de la suite de sturm et sous-résultants (I). *Informatique théorique et Applications*, décembre 1990, *24* (6), p. 561-588.
8. D. LAZARD, Sous-résultants, Manuscrit non publié.
9. R. LOOS, Generalized Polynomial Remainder Sequences, *Symbolic and Algebraic Computation*, Computing, Springer-Verlag, 1982, Supplementum (4), p. 115-137.
10. C. QUITTÉ, Une démonstration de l'algorithme de Bareiss par l'algèbre extérieure, manuscrit non publié.