CALCUL DE RÉSOLVANTES AVEC LES MODULES DE CAUCHY

NICOLAS RENNERT*, ANNICK VALIBOUZE*

Abstract. This paper gives a new and efficient algorithm in order to compute some characteristic polynomials using Cauchy modules. This algorithm is used for the computation of absolute resolvents and multi-resolvents which are essential tools in constructive Galois theory.

Cet article décrit un algorithme nouveau et efficace pour calculer des polynômes caractéristiques d'endomorphismes dans un algèbre quotient en utilisant les modules de Cauchy. Cet algorithme est ensuite utilisé pour calculer des résolvantes et des multi-résolvantes absolues, outils de base de la théorie des corps et donc aussi de la théorie de Galois constructive.

1. Introduction

La résolvante est l'outil de base de la théorie de Galois effective dont le calcul à la main s'avère vite impossible. Ainsi, chercher à diminuer le temps consacré à son calcul améliore celui du calcul du groupe de Galois d'un polynôme. Historiquement les premiers à réaliser de tels calculs à l'aide de l'outil informatique, furent R. P. Stauduhar [20] et K. Girstmair [12] utilisant pour le premier une méthode basée sur des calculs numériques. Bien que le résultat ne soit pas certifié, cette méthode s'avère être la plus rapide dans le cas général. Le logiciel GALP [10] basée sur la librairie PARI/GP, utilise ces techniques pour déterminer le groupe de Galois de polynômes.

Les méthodes algébriques ont en revanche l'avantage d'être exactes. Parmi elles, les méthodes basées sur les fonctions symétriques s'avèrent efficaces pour des invariants particuliers (voir [21]). Cet article s'intéresse aux algorithmes qui calculent des résolvantes pour un invariant quelconque et qui sont basés sur l'opération de résultant.

Le résultant est un outil particulièrement puissant pour manipuler algébriquement les racines de polynômes qui depuis Lagrange a servi pour le calcul de la résolvante (voir [9]). Le premier qui sur machine a utilisé le résultant pour le calcul de résolvantes (absolues) est L. H. Soicher pour des invariants linéaires (voir [19]).

Jusqu'à présent les méthodes basées sur le résultant pour calculer la résolvante (absolue) se heurtaient à deux défauts : apparition de facteurs et de puissances “parasites” qui augmentent considérablement la taille des polynômes nécessaires et réduisent donc l'efficacité. Cet article présente un algorithme simple et rapide (voir théorème 4.4) qui calcule, sans formation de facteurs parasites, le polynôme caractéristique d'un endomorphisme multiplicatif associé à un invariant quelconque donné. Le polynôme caractéristique est une

Date: Juin 1997.

* projet GALOIS du GDR de Calcul Formel MEDICIS.

puissance de la résolvante pour ce même invariant. Ainsi le problème des facteurs "parasites" est résolu, reste encore celui de la puissance "parasite". L'algorithme proposé utilise essentiellement le résultant et les modules de Cauchy qui forment une base de Gröbner réduite de l'idéal des relations symétriques (pour l'ordre lexicographique). Si les calculs du théorème 4.4 sont réalisés modulo l'idéal des relations symétriques alors la puissances à laquelle nous obtenons la résolvante est réduite et les calculs intermédiaires accélérés. Si les calculs modulaires n'aboutissent pas à la résolvantes mais à une puissance strictement positive (ce qui est prévisible *a priori* alors nous utilisons alors le résultat de F. Lehobey (voir [16]) pour éliminer ces puissances en cours de calcul (ce travail fait référence aux résultats de article). Ainsi le problème de la puissance parasite est également résolu (voir algo 6.2).

L'algorithme de calcul des résolvantes est naturellement généralisé au cas des multi-résolvantes, c'est-à-dire les résolvantes relatives à un produit de groupes symétriques. Finalement sont exposées l'implantation de cette méthode, la comparaison avec d'autres algorithmes algébriques ainsi que des améliorations.

2. Notations et définitions

- $\mathcal{K}$ désigne un corps de caractéristique 0 ;
- x est une variable indéterminée sur $\mathcal{K}$;
- f est un polynôme unitaire de $\mathcal{K}[x]$ de degré n ;
- $x_1, \ldots, x_n$ sont des variables indéterminées sur $\mathcal{K}$;
- $\mathcal{K}[x_1, \ldots, x_n]$ désigne l'anneau des polynômes en les variables $x_1, \ldots, x_n$ et à coefficients dans $\mathcal{K}$;
- $\mathcal{K}(x_1, \ldots, x_n)$ désigne le corps des fractions de $\mathcal{K}[x_1, \ldots, x_n]$;
- Ψ est un polynôme appartenant à $\mathcal{K}[x_1, \ldots, x_n]$;
- $\alpha = (\alpha_1, \ldots, \alpha_n)$ est une liste (ordonnée) formée des n racines du polynôme f dans une clôture algébrique $\hat{\mathcal{K}}$ de $\mathcal{K}$;
- $\Psi(\alpha) = \Psi(\alpha_1, \ldots, \alpha_n)$;
- $\mathfrak{S}_n$ désigne le groupe symétrique de degré n .

Définition 2.1. *Le polynôme Ψ est dit d'***arité** *m si m est le plus petit entier j tel qu'il existe j entiers distincts $i_1, \ldots, i_j$ vérifiant $\Psi \in \mathcal{K}[x_{i_1}, \ldots, x_{i_j}]$.*

Définition 2.2. *Si $\Psi(\alpha) = 0$ alors Ψ est appelée une* **α-relation**.

Définition 2.3. *L'action du groupe symétrique $\mathfrak{S}_n$ sur Ψ est définie par :*

$$\tau.\Psi = \Psi(x_{\tau(1)}, \ldots, x_{\tau(n)}) \text{ pour tout } \tau \in \mathfrak{S}_n$$

Définition 2.4. *L'action du groupe symétrique $\mathfrak{S}_n$ sur $\hat{\mathcal{K}}^n$ est définie, pour tout $\sigma \in \mathfrak{S}_n$ et tout $\beta = (\beta_1, \ldots, \beta_n) \in \hat{\mathcal{K}}^n$, par $\beta_\sigma = (\beta_{\sigma(1)}, \beta_{\sigma(2)}, \ldots, \beta_{\sigma(n)})$.*

Définition 2.5. L'orbite de Ψ sous l'action d'un sous-groupe L de $\mathfrak{S}_n$, *notée $L.\Psi$, est définie par :*

$$L.\Psi = \{\tau.\Psi \mid \tau \in L\}$$

Définition 2.6. *Soient L et H deux sous-groupes de $\mathfrak{S}_n$ tels que L contienne H. Le polynôme Ψ est dit L-***primitif** H-**invariant** *si H est le stabilisateur de Ψ dans L :*

$$H = Stab_L(\Psi) = \{\tau \in L \mid \tau.\Psi = \Psi\} \qquad .$$

*Lorsque $L = \mathfrak{S}_n$, l'invariant Ψ est dit également H-***invariant primitif.**

Exemple 1. Soit $H = \mathfrak{S}_2 \times \mathfrak{S}_{n-2}$. Le polynôme $\Psi = x_1 + x_2$ est un $\mathfrak{S}_n$-primitif H-invariant.

Définition 2.7. *Soient deux sous-groupes L et H du groupe symétrique $\mathfrak{S}_n$ tels que L contienne H. L'ensemble $\{\tau_1, \ldots, \tau_e\}$ est appelé une* ***transversale à gauche de*** $L \mod H$ *si $\tau_1 H, \ldots, \tau_e H$ sont les différentes classes à gauches de $L \mod H$.*

Définition 2.8. *Pour $i = 0, \ldots, n$, la i-ième* **fonction symétrique élémentaire en** $x_1, \ldots, x_n$, *notée e_i, est définie par :*

$$\begin{array}{rrcl} & e_0 & = & 1 \qquad ; \\ \textit{et pour } i \geq 1 & e_i & = & \displaystyle\sum_{m \in \mathfrak{S}_n.(x_1 \cdots x_i)} m \qquad . \end{array}$$

Remarque 1. Rappelons que $f(x) = x^n - e_1(\alpha)x^{n-1} + e_2(\alpha)x^{n-2} + \cdots + (-1)^n e_n(\alpha)$.

Définition 2.9. *Les n fonctions* **fonctions interpolaires** *introduites par Ampère (voir* [2]*) vérifient :*

$$f_i \in \mathcal{K}[x_{i+1}, \ldots, x_n][x] \qquad \textit{et} \qquad deg_x(f_i) = i \qquad .$$

Et sont définies ainsi :

$$\begin{array}{rcl} f_n(x) & = & f(x) \qquad ; \\ f_i(x) & = & f_i(x, x_{i+1}, \ldots, x_n) = \dfrac{f_{i+1}(x) - f_{i+1}(x_{i+1})}{x - x_{i+1}} \qquad n-1 \geq i \geq 1 \qquad . \end{array}$$

Définition 2.10. *Les polynômes $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$ sont appelés les* **modules de Cauchy** *associés au polynôme f.*

Définition 2.11. *Soit L un sous-groupe du groupe symétrique $\mathfrak{S}_n$. L'idéal*

$$I_\alpha^L = \{r \in \mathcal{K}[x_1, \ldots, x_n] \mid (\forall \sigma \in L)\ (\sigma.r)(\alpha) = 0\} \qquad .$$

est un idéal radical appelé **idéal des α-relations invariantes par** L.

Définition 2.12. *L'idéal $I_\alpha^{\mathfrak{S}_n}$, noté $\mathcal{I}$, est connu sous le nom d'***idéal des relations symétriques entre les racines du polynôme** f.

Définition 2.13. *Soit I_n le groupe identité dans $\mathfrak{S}_n$. L'idéal $I_\alpha^{I_n}$, noté I_α, est connu sous le nom d'***idéal des α-relations.**

Remarque 2. Le quotient $\mathcal{K}[x_1, \ldots, x_n]/\mathcal{I}$ est appelé ***algèbre de décomposition universelle.***

Définition 2.14. *Le sous-groupe G_α du groupe symétrique $\mathfrak{S}_n$ défini par*

$$G_\alpha = \{\sigma \in \mathfrak{S}_n \mid (\forall r \in \mathcal{I}_\alpha)\ (\sigma.r)(\alpha) = 0\} \qquad .$$

est connu abusivement sous le nom de ***groupe de Galois de*** f, *on le nommera groupe de Galois de α.*

Notation 2.15. *Soit I un idéal de $\mathcal{K}[x_1, \dots, x_n]$. Pour $\Theta \in \mathcal{K}[x_1, \dots, x_n]$ notons $\hat{\Theta}$ la multiplication par la classe de Θ dans l'anneau quotient $\mathcal{K}[x_1, \dots, x_n]/I$. C'est un endomorphisme du $\mathcal{K}$-espace vectoriel $\mathcal{K}[x_1, \dots, x_n]/I$. Le polynôme caractéristique de l'endomorphisme $\hat{\Theta}$ sera noté $\chi_{\Theta,I}$. Si $I = \mathcal{I}$ alors il sera noté simplement χ_Θ.*

Définition 2.16. *Soit L un sous-groupe du groupe symétrique $\mathfrak{S}_n$ tel que L contienne le groupe de Galois G_α du polynôme f. La L-**relative résolvante de** f **par** Ψ (associée à la numérotation α), notée $\mathcal{L}_{\Psi,L,\alpha}$, est le polynôme de $\mathcal{K}[T]$ défini comme suit :*

$$\mathcal{L}_{\Psi,L,\alpha}(T) = \prod_{\Theta \in L.\Psi} (T - \Theta(\alpha)) \qquad . \tag{1}$$

Si $L = \mathfrak{S}_n$, la résolvante est notée $\mathcal{L}_{\Psi,f}$ et appelée **résolvante (absolue) de** f **par** Ψ.

Comme le groupe L contient le groupe de Galois G_α et que le corps $\mathcal{K}$ est parfait, par la théorie de Galois, la résolvante appartient bien au corps $\mathcal{K}[T]$.

3. Polynômes caractéristiques et résolvantes

Dans ce paragraphe, le polynôme f est supposé séparable (i.e ses racines sont distinctes deux à deux) et L désigne un sous-groupe de $\mathfrak{S}_n$ qui contient le groupe de Galois G_α.

Proposition 1. *La variété de l'idéal I_α^L dans une clôture algébriques de $\mathcal{K}$, notée $V(I_\alpha^L)$, est l'orbite de la liste α sous l'action du groupe L :*

$$V(I_\alpha^L) = \{\alpha_\sigma \mid \sigma \in L\} \qquad .$$

Comme le polynôme f est séparable, le cardinal de cette variété est celui du groupe L.

Preuve. La preuve est simple (voir [22]). □

Pour un idéal I radical, le théorème de Stickelberger donne cette expression explicite du polynôme caractéristique de l'endomorphisme $\hat{\Psi}$ associé à un idéal I (voir Notation 2.15) :

$$\chi_{\Psi,I}(T) = \prod_{\beta \in V(I)} (T - \Psi(\beta)) \qquad .$$

Comme l'idéal I_α^L est radical et que le polynôme f est séparable, il vient :

$$\chi_{\Psi,I_\alpha^L}(T) = \prod_{\sigma \in L} (T - (\sigma.\Psi)(\alpha)) \qquad .$$

En particulier pour $I = \mathcal{I}$, l'idéal des relations symétriques :

$$\chi_\Psi(T) = \prod_{\sigma \in \mathfrak{S}_n} (T - (\sigma.\Psi)(\alpha)) \qquad . \tag{2}$$

Soient H un sous-groupe du groupe L et $\mathcal{T}$ une transversale à gauche de $L \mod H$. Si Ψ est un L-primitif H-invariant nous avons l'identité suivante bien connue :

$$\mathcal{L}_{\Psi,L,\alpha}(T) = \prod_{\tau \in \mathcal{T}} (T - (\tau.\Psi)(\alpha)) \qquad . \tag{3}$$

En effet, pour tout $\tau \in \mathcal{T}$ et pour tout $h \in H$, $\tau.\Psi = \tau h.\Psi$. De plus si $\sigma \in \mathcal{T}$ avec $\sigma \neq \tau$, alors $\sigma.\Psi \neq \tau.\Psi$.

Et nous obtenons (voir [5] pour $L = \mathfrak{S}_n$) :

$$\chi_{\Psi,I_\alpha^L} \quad = \quad (\mathcal{L}_{\Psi,L,\alpha})^{\mathrm{card}(H)} \tag{4}$$

où il apparaît que le polynôme caractéristique s'exprime comme une puissance de la résolvante absolue.

Dans le prochain paragraphe nous verrons comment se calcule le polynôme caractéristique χ_Ψ lié à l'algèbre de décomposition universelle, puis dans les suivants nous chercherons à calculer la résolvante absolue à partir de la trame de l'algorithme de calcul du polynôme caractéristique.

4. Calcul du polynôme caractéristique dans l'algèbre de décomposition universelle

Ce paragraphe énonce la formule de J.M. Arnaudiès et l'algorithme de J.L. Lagrange puis il termine par un nouvel algorithme inspiré de celui de Lagrange.

Rappel 1. Soient u et v deux polynômes de $\mathcal{K}[x]$, avec $u(x) = a\prod_{i=1}^m (x-\beta_i)$ où $\beta_1, \dots, \beta_m$ appartiennent à une clôture algébrique de $\mathcal{K}$. Le résultant en x des deux polynômes u et v, noté $\mathrm{Res}_x(u,v)$, peut s'exprimer ainsi :

$$\mathrm{Res}_x(u,v) = a^{\deg(v)} \prod_{i=1}^m v(\beta_i) \qquad . \tag{5}$$

4.1. La méthode d'Arnaudiès.

la méthode proposée dans [4] est basée sur le résultant à n variables.

Notons x_{n+1} une indéterminée supplémentaire et d le degré total de Ψ en $x_1, \dots, x_n$. L'homogénéisé de Ψ, noté Ψ^*, est donné par :

$$\Psi^* = x_{n+1}^d \Psi(\frac{x_1}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}}) \qquad .$$

Notons $\mathrm{Res}(g_1, \dots, g_n)$ le résultant de n polynômes $g_1, \dots, g_n$ homogènes non constants à n variables.

Théorème 4.1. *(Arnaudiès) Posons $s_i = e_i - e_i(\alpha)$ pour $i = 1, \dots, n$. Alors le polynôme caractéristique est donné par :*

$$\chi_\Psi(T) \quad = \quad Res(s_1^*, \dots, s_n^*, T x_{n+1}^d - \Psi^*) \qquad .$$

Ce théorème démontré dans [4] se déduit de la formule de Poisson-Perron (voir [3], p. 243). Le calcul de résultant à n variables s'avère très coûteux, cette méthode n'est donc pas utilisable en pratique.

4.2. La méthode de Lagrange.

Ce paragraphe traduit en terme de résultants, la méthode que J.L. Lagrange proposait pour calculer la résolvante absolue $\mathcal{L}_{\Psi,f}$ (voir [9]) et qui calcule le polynôme caractéristique.

Soit $(U_i)_{0 \le i \le n}$ la suite finie définie inductivement comme suit :

$$U_0(T, x_1, \dots, x_n) = T - \Psi(x_1, \dots, x_n) \qquad ,$$

$$U_i(T, x_{i+1}, \dots, x_n) = Res_{x_i}(f(x_i, \dots, x_n), U_{i-1}(T, x_i, \dots, x_n)) \qquad 1 \le i \le n \quad .$$

Alors, d'après le rappel 1 et puisque f est unitaire :

$$U_n(T) = \prod_{i_n=1}^{n} \prod_{i_{n-1}=1}^{n} \cdots \prod_{i_1=1}^{n} (T - \Psi(\alpha_{i_1}, \dots, \alpha_{i_n})) \quad . \tag{6}$$

Les facteurs sur $\mathcal{K}$ du polynôme $U_n(T)$ sont

- le polynôme caractéristique χ_Ψ ;
- des facteurs dits **parasites** provenant des égalités deux à deux, trois à trois, ..., n à n des indices apparaissant dans les produits. Effectivement les nombres algébriques tels $\Psi(\alpha_1, \alpha_1, \alpha_3, \dots, \alpha_n)$, $\Psi(\alpha_1, \alpha_1, \alpha_1, \alpha_3, \dots, \alpha_n), \dots, \Psi(\alpha_1, \alpha_1, \dots, \alpha_1)$ sont racines de $U_n(T)$ et ne sont pas racines du polynôme caractéristique.

Ces facteurs parasites étaient déjà signalés par J.L. Lagrange. Ce sont des polynômes caractéristiques associés à des invariants d'*arité* strictement inférieure à celle de l'invariant Ψ. Par décroissance stricte de l'arité, il est donc toujours possible de déduire le polynôme caractéristique χ_Ψ à partir du polynôme U_n. Donnons un exemple explicite :

Exemple 2. Choisissons l'invariant $\Psi = x_1 + 2x_2$ et $n = 2$. Alors

$$U_2(T) = \chi_\Psi \cdot \chi_{3x_1}$$

où $\chi_{3x_1} = Res_x(f(x), T - 3x)$. De plus $\mathcal{L}_{\Psi,f} = \chi_\Psi$ puisque Ψ est un I_2-invariant primitif.

La méthode proposée au paragraphe 4.3 évite la formation des facteurs parasites et calcule donc directement le polynôme caractéristique. J.L. Lagrange, qui cherchait à calculer la résolvante, signalait également la puissance parasite provenant des symétries de l'invariant Ψ et que nous retrouvons dans la formule (4). Il est possible de diminuer la puissance en cours de calcul (voir le paragraphe 6 consacré au calcul de la résolvante).

4.3. Méthode des modules de Cauchy.

Notons $f_1, \dots, f_n$ les modules de Cauchy associés au polynôme f (voir Définition 2.10).

Définition 4.2. *Soient r un entier positif et s un entier compris entre 1 et n. La r-ième* **fonction symétrique complète**, *notée $h_r(x_1, \dots, x_s)$, est la somme des monômes de degré total r en $x_1, \dots, x_s$, avec $h_0(x_1, \dots, x_s) = 1$.*

Le théorème suivant permet de les calculer efficacement il se démontre facilement.

Théorème 4.3. *(Machì-Valibouze) Soit le polynôme $f(x) = x^n + a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_n$ à coefficients dans $\mathcal{K}$. Posons $a_0 = 1$. Alors, pour $1 \le r \le n$, le (r)-ième module de Cauchy associé à f est donné par :*

$$f_r(x_r) = \sum_{i=0}^{r} h_i(x_r, \dots, x_n) a_{r-i} \quad . \tag{7}$$

En particulier $f_n(x_n) = \sum_{i=0}^{n} h_i(x_n) a_{n-i} = f(x_n)$ et $f_1(x_1) = h_1(x_1, \dots, x_n) + a_1$.

Remarque 3. A. Cauchy donna la formule jusqu'à $n = 4$. Une démonstration simple proposée par A. Lascoux peut-être faite à l'aide des différences divisées (voir [15]).

Le théorème suivant donne une méthode explicite du calcul du polynôme caractéristique :

Théorème 4.4. *Soit la suite finie $R_0, R_1, \ldots, R_n$ définie inductivement par :*

$$\begin{array}{rcll} R_0(T, x_1, \ldots, x_n) & = & T - \Psi \qquad \text{et} & \\ R_i(T, x_{i+1}, \ldots, x_n) & = & Res_{x_i}(f_i(x_i), R_{i-1}(T, x_i, \ldots, x_n)) & 1 \leq i \leq n \end{array} .$$

Alors le polynôme caractéristique est donné par

$$\chi_\Psi(T) = R_n(T) \qquad . \tag{8}$$

Preuve. Nous montrons le théorème par récurrence sur n, le degré du polynôme f.

Pour $n = 1$, $f = (x - \alpha_1)$, comme le polynôme f est unitaire nous avons bien :

$$Res_{x_1}(f(x_1), T - \Psi(x_1)) = (T - \Psi(\alpha_1)) = \chi_\Psi(T) \qquad .$$

Montrons-le également pour $n = 2$. Nous avons $f_2(x_1, x_2) = x_1 + x_2 - e_1(\alpha)$ et donc $R_1(T, x_1) = \mathrm{Res}_{x_1}(f_1(x_1, x_2), T - \Psi(x_1, x_2)) = T - \Psi(x_2, e_1(\alpha) - x_2))$. D'où

$$\begin{array}{rcl} R_2(T) & = & Res_{x_2}(f(x_2), T - \Psi(x_2, e_1(\alpha) - x_2)) \\ & = & (T - \Psi(\alpha_1, e_1(\alpha) - \alpha_1))(T - \Psi(\alpha_2, e_1(\alpha) - \alpha_2)) \\ & = & (T - \Psi(\alpha_1, \alpha_2))(T - \Psi(\alpha_2, \alpha_1)) \\ & = & \chi_\Psi(T) \text{ d'après la formule (2)} \qquad . \end{array}$$

Supposons désormais que le théorème soit vrai jusqu'à $n - 1$ dans tout anneau commutatif unitaire, donc en particulier dans $\mathcal{K}[x_n]$. Ainsi, en posant, $F(x) = f_{n-1}(x, x_n) = f_{n-1}(x)$ et en notant $F_{n-1}(x_{n-1}), F_{n-2}(x_{n-2}, x_{n-1}), \ldots, F_1(x_1, \ldots, x_{n-1})$ les modules de Cauchy associés à F, nous avons $F_j(x_j, \ldots, x_{n-1}) = f_j(x_j, \ldots, x_n)$ pour $j = 1, \ldots, n-1$. Notons $\beta_1(x_n), \ldots, \beta_{n-1}(x_n)$ les $n - 1$ racines de $F(x)$ dans une clôture algébrique de $\mathcal{K}[x_n]$. Par hypothèse de récurrence et puisque $F(x)$ est unitaire, nous avons

$$R_{n-1}(T, x_n) = \prod_{\tau \in \mathfrak{S}_{n-1}} \left(T - \Psi(\beta_{\tau(1)}(x_n), \ldots, \beta_{\tau(n-1)}(x_n), x_n)\right) \qquad ,$$

et donc

$$R_n(T) = \prod_{i=1}^{n} \prod_{\tau \in \mathfrak{S}_{n-1}} (T - \Psi(\beta_{\tau(1)}(\alpha_i), \ldots, \beta_{\tau(n-1)}(\alpha_i), \alpha_i)) \qquad . \tag{9}$$

Or, pour $i = 1, \ldots, n$, pour $j = 1, \ldots, n - 1$ et pour $\tau \in \mathfrak{S}_{n-1}$, nous savons que $\beta_{\tau(j)}(\alpha_i)$ est une racine de $F(x) = f_2(x, \alpha_i) = \frac{f(\alpha_i) - f(x)}{\alpha_i - x}$ et est donc une racine de f distincte de α_i. Ainsi, les $n!$ n-uplets $(\beta_{\tau(1)}(\alpha_i), \ldots, \beta_{\tau(n-1)}(\alpha_i), \alpha_i)$ sur lesquels s'évalue l'invariant Ψ dans le produit (9) sont les $n!$ n-uplets $(\alpha_{\sigma(1)}, \ldots, \alpha_{\sigma(n)})$ où σ parcourt $\mathfrak{S}_n$. Ceci achève la démonstration. □

Remarque 4. Comme le polynôme f est supposé unitaire, alors chaque module de Cauchy $f_i(x_1, \ldots, x_i)$ est également unitaire en x_i pour $i = 1, \ldots, n$. Dans le cas où f n'est pas unitaire, et où a est son coefficient dominant, il suffit de poser $g(x) = a^{n-1} f(x/a)$ qui, lui, est un polynôme unitaire. Pour garder le polynôme f, il faut alors introduire le coefficient dominant de $f = f_n$ ainsi que ceux des modules de Cauchy $f_1, \dots, f_{n-1}$ dans les calculs des résultants successifs.

5. Réductions avec des bases de Gröbner

5.1. Bases de Gröbner.

L'idéal $\mathcal{I}$ est engendré par les n polynômes $e_1 - e_1(\alpha), \dots, e_n - e_n(\alpha)$. Mais ces polynômes n'en forment pas une base de Gröbner (voir [6]).

Rappelons le théorème historique de Cauchy (voir [8]) dans lequel il énonce comment les utiliser pour évaluer les polynômes symétriques : [8]

> *Soit $F(x_1, \dots, x_n)$ un polynôme à coefficients dans $\mathcal{K}$ et symétrique en les variables $x_1, \dots, x_n$. Pour éliminer $x_n, \dots, x_1$ dans le polynôme F, il suffit de diviser successivement F par les divers termes de la suite $f_1(x_1), \dots f_{n-1}(x_{n-1})$, $f_n(x_n)$, en considérant chaque f_i comme une fonction de x_i. Le dernier reste obtenu sera indépendant de $x_1, \dots, x_n$ et donnera la valeur $F(\alpha_1, \dots, \alpha_n)$ en fonction des coefficients de f.*

Autrement dit, lorsque le polynôme f est séparable, les modules de Cauchy du polynôme f, forment une base de Gröbner réduite de l'idéal $\mathcal{I}$ des relations symétriques. Pour évaluer un polynôme symétrique il suffit de le *réduire* modulo l'idéal $\mathcal{I}$.

Notation 5.1. *Pour $i \in [1, n]$, la notation $\mathcal{I}_i$ désigne l'idéal engendré dans $\mathcal{K}[x_i, \dots, x_n]$ par les modules de Cauchy $f_i, \dots, f_n$.*

Il est bien connu qu'alors les polynômes $f_i, \dots, f_n$ forment une base de Gröbner réduite pour l'ordre lexicographique de l'idéal $\mathcal{I}_i$ (voir [6]). Ainsi, en considérant $\mathcal{K}[x_i, \dots, x_n]$ comme un sous-anneau de $\mathcal{K}[x_1, \dots, x_n]$:

$$\begin{aligned} \mathcal{I}_i &= \mathcal{I} \cap \mathcal{K}[x_i, \dots, x_n] \\ &= \{P \in \mathcal{K}[x_i, \dots, x_n] \mid (\forall \sigma \in \mathfrak{S}_n)\ (\sigma.P)(\alpha) = 0\} \quad . \end{aligned}$$

En particulier $\mathcal{I}_1 = \mathcal{I}$.

Le théorème de Cauchy se généralise facilement ainsi:

Théorème 5.2. *Supposons que le polynôme f soit réductible sur $\mathcal{K}$: $f = gh$ où $g, h \in \mathcal{K}[x]$ avec $deg(g) = m$ et $deg(h) = p = n - m$. Choisissons la numérotation des racines du polynôme f de telle sorte que $\alpha_1, \dots, \alpha_m$ soient les m racines du polynôme g. Notons $g_1, \dots, g_m$ les modules de Cauchy du polynôme g et $h_1, \dots, h_p$ ceux du polynôme h. Alors les n polynômes $g_1, \dots, g_m, h_1, \dots, h_p$ forment une base de Gröbner réduite de l'idéal $I_\alpha^{\mathfrak{S}_m \times \mathfrak{S}_p}$.*

Preuve. D'après [22], l'idéal $I_\alpha^{\mathfrak{S}_m \times \mathfrak{S}_p}$ est engendré dans $\mathcal{K}[x_1, \dots, x_n]$ par l'idéal dans $\mathcal{K}[x_1, \dots, x_m]$ des relations symétriques entre les racines du polynôme g et l'idéal dans $\mathcal{K}[x_{m+1}, \dots, x_n]$ des relations symétriques entre les racines du polynôme h :

$$\mathcal{I}_\alpha^{\mathfrak{S}_m \times \mathfrak{S}_p} = \mathcal{I}^{\mathfrak{S}_m}_{(\alpha_1, \dots, \alpha_m)} + \mathcal{I}^{\mathfrak{S}_p}_{(\alpha_{m+1}, \dots, \alpha_n)}$$

$\square$

5.2. Réduction modulo un idéal triangulaire.

Définition 5.3. *Un ensemble T de n polynômes de $\mathcal{K}[x_1, \ldots, x_n]$ est dit* ***triangulaire*** *de $\mathcal{K}[x_1, \ldots, x_n]$ si $T = \{p_1(x_1), \ldots, p_n(x_1, \ldots, x_n)\}$ où chaque polynôme p_i est unitaire en x_i avec degré$(p_i, x_i) > 0$.*

Définition 5.4. *Un idéal $\mathcal{J}$ est dit* ***triangulaire*** *s'il est engendré par un ensemble triangulaire.*

Exemple 3. L'idéal $\mathcal{I}$ est un idéal triangulaire, ainsi que l'idéal $I_\alpha^{\mathfrak{S}_m \times \mathfrak{S}_p}$ défini au théorème 5.2.

Soit $\mathcal{J}$ un idéal triangulaire, engendré par l'ensemble triangulaire :

$$\{p_1(x_1), p_2(x_1, x_2), \ldots, p_n(x_1, \ldots, x_n)\} \qquad .$$

Réduire le polynôme Θ de $\mathcal{K}[x_1, \ldots, x_n]$ par l'idéal $\mathcal{J}$ revient a réaliser successivement des divisions euclidiennes par chaque polynôme p_j considéré comme un polynôme en x_j ($j \in [1, n]$). Le reste de cette division appartiendra au quotient $\mathcal{K}[x_1, \ldots, x_n]/\mathcal{J}$.

Notation 5.5. *Pour $\Theta \in \mathcal{K}[x_i, \ldots, x_n][T]$ le résultat de cette réduction sera noté $\Theta \mod \mathcal{J}$.*

algorithme 5.1. *Réduction*

---

entrées : $\Theta \in \mathcal{K}[x_1, \ldots, x_n][T]$,
    $p_1, \ldots, p_n$ *: une base triangulaire de* $\mathcal{J}$.
sortie : $\Theta \mod \mathcal{J}$
Début
  resultat $\leftarrow \Theta$
  Pour *j* ***De*** 1 *à n*
    Répète *resultat* $\leftarrow$ *Reste*$(resultat, p_j, x_j)$
  Fin Pour
  resultat
Fin

Où *Reste*(p, q, x_k) est le reste de la division euclidienne de p par q, considérés comme des polynômes de $\mathcal{K}[x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n, T][x_k]$.

6. Calcul de la résolvante absolue

La formule (4) montre que le polynôme caractéristique est une puissance de la résolvante. Cette partie est donc consacrée à l'élimination de cette puissance. Le paragraphe 6.1 rappelle les résultats de F. Lehobey sur ce point et le paragraphe 6.4 montre comment certaines puissances s'éliminent lorsque les calculs sont réalisés dans l'algèbre de décomposition universelle.

Les modules de Cauchy du polynôme f sont notés $f_1, \ldots, f_n$.

6.1. Méthode de Lehobey.

Dans la pratique, le calcul de la résolvante $\mathcal{L}_{\Psi,f}(T)$ ne se réalise pas sans réduire la puissance parasite due aux symétries de l'invariant Ψ à chaque étape de l'algorithme 6.1 (voir (4)). La formation de cette puissance est illustrée au travers de l'exemple suivant :

Exemple 4. Fixons $n = 3$ et choisissons $\Psi = 2x_3 + x_1 + x_2$ un $\mathfrak{S}_3$-primitif H-invariant où $H = \mathfrak{S}_2 \times \mathfrak{S}_1$. D'après la définition de la suite $(R_j)_j$ du Théorème 4.4, il vient :

$$R_{n-1}(T, x_3) = \mathrm{Res}_{x_2}(f_2(x_2, x_3), Res_{x_1}(f_1(x_1, x_2, x_3), T - \Psi)) \qquad .$$

Il est alors possible d'éliminer des symétries à ce niveau puisque :

$$R_2(T, x_3) \quad = \quad (T - (2x_3 + \beta_1(x_3) + \beta_2(x_3)))^2 \qquad ,$$

où $\beta_1(x_3), \beta_2(x_3)$ sont les racines de $f_2(x)$ dans $\mathcal{K}[x_3][x]$. En posant $V_2(T, x_3)^2 = R_2(T, x_3)$, nous avons finalement :

$$\mathcal{L}_{\Psi,f}(T) = \mathrm{Res}_{x_3}(f_3(x_3), V_2(T, x_3)) \qquad .$$

Nous constatons que nous obtenons la résolvante au lieu de son carré donné par la formule (4). Cet exemple montre également qu'il est préférable d'éliminer d'abord les symétries.

L'exemple 4 suppose connu le polynôme V_2 qui se calcule par la méthode de F. Lehobey (voir [16]). Elle s'appuie sur le Théorème 6.1 et aboutit à l'algorithme 6.1 exposés ci-après.

Le Théorème 6.1 explicite le degré de chaque puissance superflue qui apparaît après un calcul de résultant.

Théorème 6.1. *(Lehobey) Soient A un anneau intègre de caractéristique nulle, $f \in A[x]$ un polynôme unitaire, de degré n ; H_n un sous-groupe de $\mathfrak{S}_n$; $\Psi \in A[x_1, \ldots, x_n]$ un H_n-invariant primitif ; $f_1, \ldots, f_n$, les n modules de Cauchy de f et $H_i = \mathrm{Stab}_{\mathfrak{S}_i}(\Psi)$, pour $1 \le i \le n-1$, les $n-1$ stabilisateurs sur $\mathfrak{S}_i$ de Ψ considéré comme un élément de $A[x_{i+1}, \ldots, x_n][x_1, \ldots, x_i]$. Soit l'entier m_i défini par :*

$$m_i \quad = \quad \frac{Card(H_i)}{Card(H_{i-1})} \qquad 1 \le i \le n \qquad ,$$

avec $Card(H_0) = 1$. Alors la suite finie $V_0, \ldots, V_n$ définie récursivement par:

$$V_0(T, x_1, \ldots, x_n) = T - \Psi \qquad ;$$

$$V_i(T, x_{i+1}, \ldots, x_n) = (Res_{x_i}(f_i(x_i), V_{i-1}(T, x_i, \ldots, x_n)))^{1/m_i}$$

calcule la résolvante de f par Ψ :

$$\mathcal{L}_{\Psi,f}(x) = V_n(x) \qquad .$$

6.2. La racine r-ième.

L'algorithme de calcul de la résolvante (6.1) ainsi que les algorithmes qui en découlent, font appel à la fonction *monicPerfectNthRoot*() définie de la façon suivante :
Soient A un anneau commutatif de caractéristique nulle quelconque, m un entier positif et p un polynôme de $\mathcal{K}[x_1, \ldots, x_m]$ unitaire en x_i ($i \in [1 \ldots m]$), pour tout $k \in \mathbb{N}$ la fonction *monicPerfectNthRoot*(p^k, k, x_i) retourne le polynôme p. Cette fonction est basée sur le travail de P. Henrici [13].

6.3. L'algorithme de calcul de la résolvante.

algorithme 6.1. *Résolvante*

---

entrées : f *: un polynôme de* $\mathcal{K}[x]$,
  Ψ *: un polynôme de* $\mathcal{K}[x_1, \ldots, x_n]$ H*-invariant primitif,*
  $PuissancesParasites$ *: une liste d'entiers contenant les degrés des puissances parasites.*
sortie : la résolvante de f *par* Ψ
Début
  $resultat \leftarrow T - \Psi$
  Pour *cm* ***Dans*** *modulesDeCauchy*$(f, [x_1, \ldots, x_n])$
    v ***Dans*** $[x_1, \ldots, x_n]$
    pp ***Dans*** $PuissancesParasites$
    Répète $resultat \leftarrow$ *monicPerfectNthRoot*(*resultant*$(cm, resultat, v), pp, T)$
  Fin Pour
  $resultat$
Fin

Où :

- la fonction *modulesDeCauchy*$(f, [x_1, \ldots, x_n])$ renvoie la liste $[f_1(x_1), \ldots, f_n(x_n)]$ des modules de Cauchy de f
- *monicPerfectNthRoot*(f, n, y) calcule la racine n-ième de f comme polynôme en y (voir section 6.2).
- La liste des puissances parasites $PuissancesParasites$ est fonction de l'invariant. Elle peut être calculée en utilisant des stabilisateurs (voir théorème 6.1) avec le logiciel GAP [17, 11] ou par la factorisation des calculs intermédiaires.

Preuve. Cet algorithme découle des théorèmes 4.4 et 6.1. □

Exemple 5. Prenons le polynôme $f = x^6 - 243x^2 + 729$ et l'invariant $\Psi = x_5x_6 + x_3x_4 + x_1x_2$. Le temps nécessaire au calcul du polynôme caractéristique puis à celui de la résolvante à partir de la formule (4) est de 1 heure et 40 minutes. Avec l'algorithme 6.1 le temps nécessaire au calcul de la résolvante n'est plus que de 1 seconde.

6.4. Calculs modulo l'idéal des relations symétriques.

L'un des inconvénients de l'algorithme 6.1 est la croissance des degrés en les variables $x_1, \ldots, x_n$ dans les calculs intermédiaires du polynôme $resultat$. Nous étudions donc le moyen de contrôler cette croissance par réductions utilisant des divisions euclidiennes. Nous constaterons que la méthode employée aboutit parfois à l'élimination de plusieurs variables lors d'un même pas de boucle. L'élimination prématurée des variables par réduction n'occasionne donc aucune puissance superflue. Une réduction est alors équivalente à autant de pas de l'algorithme 6.1 que de variables éliminées prématurément par réduction.

6.4.1. *Réduction de polynômes.*

Soit $i \in [1, n]$. Puisque $f_i, \ldots, f_n$ est une base de Gröbner réduite de l'idéal $\mathcal{I}_i$ de dimension zéro (voir Notation 5.1), l'anneau quotient $\mathcal{K}[x_i, \ldots, x_n]/\mathcal{I}_i$ est engendré en tant que $\mathcal{K}$-module par les monômes m tels que, pour $j \in [i, n]$, degré$(m, x_j) < j =\deg_{x_j}(f_j(x_j))$ (voir Définition 2.10). Autrement dit :

$$\mathcal{K}[x_i, \ldots, x_n]/\mathcal{I}_i \simeq \bigoplus_{0 \leq b_j < j} \mathcal{K}.x_i^{b_i} \ldots x_n^{bn} \qquad .$$

Ainsi les degrés en x_j $(0 < j \leq n)$ des polynômes de l'algorithme 6.1 peuvent être contrôlés pour ne pas dépasser $j - 1$.

Notation 6.2. *Soit I un idéal quelconque, la notation*

$$\textbf{\textit{expression}} \mod I$$

*qui désigne le résultat de la réduction de **expression** modulo l'idéal I, sous-entend que tous les calcul d'**expression** sont effectués modulo l'idéal I.*

6.4.2. *Élimination automatique des puissances superflues.*
Commençons par des exemples qui illustrent comment, après une réduction, des variables peuvent être éliminées prématurément évitant ainsi, d'une part, des calculs de résultants et, d'autre part, l'apparition de puissances superflues.

Exemple 6. Cet exemple montre que l'arité de l'invariant est importante. Prenons un polynôme f de degré 4. Supposons que $\Psi = x_4 + 2x_3$ où 2 est donc l'arité de Ψ (voir Définition 2.1). Soit $S_0 = (T - \Psi) \mod \mathcal{I} = T - (x_4 + 2x_3)$, puis $S_1 =\mathrm{Res}_{x_3}(f_3(x_3, x_4), S_0) = T^3 + (2 - x_4)T^2 + T(3x_4^2) + 5x_4^3 + 6x_4^2 = S_1 \mod \mathcal{I}_4$. Alors

$$\begin{aligned} S_2 &= \mathrm{Res}_{x_4}(f_4(x_4), S_1) \\ &= T^{12} + 9T^{11} + 33t^{10} + 63T^9 + 40T^8 - 120T^7 - 346T^6 - 336T^5 \\ &\quad +2348T^4 + 7332T^3 + 7416T^2 + 3888T + 5864 \\ &= \mathcal{L}_{\Psi,f} \qquad . \end{aligned}$$

Le calcul de la résolvante s'est alors réalisé en évitant deux résultants et par conséquent sans l'apparition de puissances superflues induites par ces résultants. De manière générale, si $m \leq 1$ et $\Psi \in \mathcal{K}[x_m, \ldots, x_n]$ est un $\mathfrak{S}_{n-m}$-primitif K-invariant alors le calcul de la résolvante se réalise sans les m premiers résultants définissant la suite $(R_j)_j$ du théorème 4.4. Ces premiers résultants sont nécessaires au calcul du polynôme caractéristique mais parfaitement inutiles à celui de la résolvante. De plus, ils introduisent une puissance superflue égale à $m!$. En effet, si la suite $(R_j)_j$ du théorème 4.4 est modifiée en ignorant les m premiers termes et en posant $R_{m+1} = T - \Psi(x_m, \ldots, x_n)$, alors

$$\chi_\Psi = R_n^{m!} \qquad \text{et} \tag{10}$$

$$R_n = (\mathcal{L}_{\Psi,f})^{\mathrm{card}(K)} \qquad . \tag{11}$$

Exemple 7. Prenons $f = x^4 + x^3 + 2$ et choisissons $\Psi = x_3x_4 + x_1x_2$ un $\mathcal{D}_4$-invariant primitif (où $\mathcal{D}_4$ est le groupe diédral). Le calcul du polynôme

$$S_0 = (T - \Psi) \mod \mathcal{I}_1$$

aboutit à $S_0 = x_3^2 + (-2\ x_4 - 1)\ x_3 - x_4^2 - x_4 + T$. Les variables x_1 et x_2 sont éliminées sans un calcul de résultant. Le calcul du polynôme

$$S_1 = \mathrm{Res}_{x_3}(f_3(x_3), S_0) \mod \mathcal{I}_4$$

aboutit à $S_1 = T^3 - 8\ T - 2$. Comme le degré du polynôme S_1 est égal à celui de la résolvante $\mathcal{L}_{\Psi,f}$, le Théorème 6.3 nous assure que ces polynômes sont égaux.

Théorème 6.3. *Posons $I_{n+1} = (0)$ et définissons la suites finie d'entiers $(k_i)_{0\leq i\leq j}$ ($k_i \in [0, n]$) et la suite finie de polynômes $(S_i)_{0\leq i\leq j}$ de la manière suivante :*

- *l'entier naturel $n - k_i + 1$ est l'arité du polynôme S_i qui appartient à $\mathcal{K}[T][x_{k_i}, \ldots, x_n]$;*
- *l'entier j est le plus petit entier i tel que $k_i = n + 1$ (i.e l'arité de S_i est nulle) ;*
- *Au départ :*

$$S_0(T, x_{k_0}, \ldots, x_n) = T - \Psi \mod \mathcal{I}_1$$

et pour $i \in [0, j[$

$$S_{i+1}(T, x_{k_{(i+1)}}, \ldots, x_n) = Res_{x_{k_i}}(f_{k_i}(x_{k_i}), S_i(T, x_{k_i}, \ldots, x_n))^{1/m_{k_i}} \mod \mathcal{I}_{k_i+1} \qquad ,$$

où l'entier m_{k_i} est celui défini au Théorème 6.1.

Alors le dernier terme de la suite est une puissance de la résolvante cherchée :

$$S_j(T)^m = \mathcal{L}_{\Psi,f}(T) \qquad m > 0 \qquad . \tag{12}$$

Preuve. Cette démonstration se déroule en trois étapes. La première montre que l'on peut remplacer chaque V_i de la suite $(V_i)_i$ définie au Théorème 6.1 par $V_i \mod \mathcal{I}_{i+1}$. La seconde montre que l'on peut réaliser tous les calculs de la suite $(V_i)_i$ modulo les idéaux $\mathcal{I}_j$ ($j \in [1, n]$) et la troisième étape montre comment éviter des calculs en tenant compte des variables éliminées prématurément.

a) Soit $W = V_i \mod \mathcal{I}_{i+1}$, c'est à dire, d'après l'algorithme 6.2, $V_i = \sum_{j=i+1}^{n} \$ f_j(x_j) + W$ où $\$$ désigne un polynôme quelconque (en l'occurrence un quotient d'une division euclidienne par $f_j(x_j)$). Alors dans les résultants successifs définissant $V_{i+1}, \ldots, V_n$, le polynôme V_i peut-être remplacé par le polynôme W sans que le polynôme V_n soit modifié, puisque ces résultants sont réalisés avec les polynômes $f_{i+1}(x_{i+1}), \ldots, f_n(x_n)$.
b) Maintenant posons $P = \mathrm{Res}_{x_i}(f_i(x_i), V_{i-1}) = V_i^{m_i}$ et $S = P \mod \mathcal{I}_{i+1}$, c'est à dire, d'après l'algorithme 6.2, $P = \sum_{j=i+1}^{n} \$ f_j(x_j) + S$. Alors $S = W^{m_i} \mod \mathcal{I}_{i+1}$. Ainsi, en remplaçant V_i par $S^{1/m_i} \mod \mathcal{I}_{i+1}$ dans la suite $(V_j)_j$, le résultat $V_n = \mathcal{L}_{\Psi,f}$ n'est pas modifié.
c) Tenons compte maintenant des variables de W. Supposons que le polynôme V_i soit remplacé par W dans la suite $(V_j)_j$. Nous avons donc

$$V_{i+1}^{m_{i+1}} = \mathrm{Res}_{x_{i+1}}(f_{i+1}(x_{i+1}), W) \qquad .$$

Si W ne dépend pas de x_{i+1}, alors $V_{i+1}^{m_{i+1}} = W^{i+1}$. Ainsi, si V_{i+1} est remplacé par W dans la suite $(V_j)_j$, alors $V_n = \mathcal{L}_{\Psi,f}^{i+1/m_{i+1}}$.

$\square$

Remarque 5. Dans l'équation (12), l'entier m sera égal à 1 si les résultants évités par élimination prématurée des variables (voir partie c) de la démonstration correspondent à des symétries de l'invariant Ψ et non pas au fait que l'invariant Ψ n'est pas séparable. Par exemple, si Ψ est un H-invariant primitif, où H est un sous-groupe propre de $\mathfrak{S}_n$, et que $\Psi - \Psi(\alpha) \in \mathcal{I}$, alors $i_0 = 0$, $S_{i_0} = T - \Psi(\alpha) \in \mathcal{K}[T]$ et $\mathcal{L}_{\Psi,f} = S_{i_0}^{[\mathfrak{S}_n:H]}$. L'identification du groupe de Galois du polynôme f se réalise avec les facteurs simples des résolvantes. Donc, en pratique, dès que le programme détecte que le saut de plusieurs résultants n'aboutira pas au calcul exact de la résolvante (i.e. $m = 1$), il sera stoppé pour choisir un autre invariant du même groupe ou pour transformer le polynôme f. Cette détection se fait avec les degrés des modules de Cauchy et les entiers m_i de Lehobey dès que le calcul d'un résultant est évité.

6.4.3. *Algorithme avec réduction.*

algorithme 6.2. *algorithme de calcul de la résolvante avec réductions :*

---

entrées : f : un polynôme de $\mathcal{K}[x]$,
Ψ : un polynôme de $\mathcal{K}[x_1, \ldots, x_n]$ H-invariant primitif,
puissancesParasites : une liste d'entiers contenant les degrés des puissances parasites.
sortie : la H-résolvante de f par Ψ
Début
varutiles $\leftarrow$ *variables*(Ψ)
cm $\leftarrow$ *modulesDeCauchy*(f, *varutiles*)
resultat $\leftarrow T - \Psi$
Tant que *arite*(*resultat*) $\neq 1$
Répète $v \leftarrow$ *mainVariable*(*resultat*)
$k \leftarrow$ *position*(v, *varutiles*)
pp $\leftarrow$ *puissancesParasites*[k]
resultat $\leftarrow$ $resultant(cm[k], resultat, v)^{\frac{1}{pp}} \mod \mathcal{I}_{k+1}$
Fin Tant que
resultat
Fin

Où :

- la fonction *arite*(P) renvoie l'arité de P,
- la fonction *position*(v, lv) renvoie la position de v dans la liste lv,
- la fonction *mainVariable*(P) renvoie la variable principale de P,
- la notation $l[k]$ désigne le k-ième élément de la liste l,
- la fonction *variables*(P) renvoie la liste des variables de P,
- la fonction *modulesDeCauchy*($f, [x_1, \ldots, x_n]$) renvoie la liste $[f_1(x_1), \ldots, f_n(x_n)]$ des modules de Cauchy de f,

- l'idéal $\mathcal{I}_{k+1}$ est défini notation 5.1,
- la notation $P \mod \mathcal{I}_{k+1}$ est décrite en notation 6.2,
- le calcul de $P^{\frac{1}{k}} \mod \mathcal{I}$ est réalisé à l'aide de la fonction *monicPerfectNthRoot*(p, k, v) décrite au paragraphe 6.2.
- La liste des puissances parasites *PuissancesParasites* est fonction de l'invariant. Elle peut être calculée en utilisant des stabilisateurs (voir théorème 6.1) avec le logiciel GAP [17, 11] ou par la factorisation des calculs intermédiaires.

7. Généralisation aux multi-résolvantes

Soient $F = (f^{(1)}, \ldots, f^{(p)})$ un p-uplet de polynômes de $\mathcal{K}[x]$ de degrés respectifs $d_1, \ldots, d_p$ et $d = d_1 + \cdots + d_p$. Donnons-nous $x_1, \ldots, x_d$ des indéterminées. Soit Ω_F la liste des racines des polynômes de F, ordonnées ainsi : $\Omega_F = (\Omega^{(1)}, \ldots, \Omega^{(p)})$ où $\Omega^{(i)}$ est une liste quelconque des racines du polynôme $f^{(i)}$ pour $i \in [1, p]$.

Fixons $\Psi \in \mathcal{K}[x_1, \ldots, x_d]$ et posons $\mathcal{S} = \mathfrak{S}_{d_1} \times \cdots \times \mathfrak{S}_{d_p}$.

Définition 7.1. ***La multi-résolvante absolue*** *de F par Ψ, notée $\mathcal{L}_{\Psi,F}$, est la résolvante relative $\mathcal{L}_{\Psi,\mathcal{S},\Omega_F}$. La multi-résolvante de F est donc le polynôme*

$$\mathcal{L}_{\Psi,F} = \prod_{\Theta \in \mathcal{S}.\Psi} (y - \Theta(\Omega_F))$$

Pour calculer une multi-résolvante, il suffit d'utiliser les algorithmes de calcul des résolvantes conçus pour des invariants avec des coefficients non numériques. Dans la pratique, le corps $\mathcal{K}$ peut être remplacé par un anneau intègre de caractéristique nulle (voir Théorème 6.1) . En ce cas l'invariant Ψ est considéré successivement comme un polynôme en $(x_1, \ldots, x_{d_1})$, puis en $(x_{d_1+1}, \ldots, x_{d_1+d_2})$,... puis enfin en $(x_{d_1+\cdots+d_{p-1}+1}, \ldots, x_d)$. L'algorithme de calcul d'une multi-résolvante est donc le suivant.

algorithme 7.1. *Multi-résolvante absolue*

---

entrées : $F = [f^{(1)}, \ldots, f^{(p)}]$ *: une liste de polynômes de* $\mathcal{K}[x]$,
  Ψ *: un polynôme de* $\mathcal{K}[x_1, \ldots, x_n]$ *qui est H-invariant primitif,*
  lpp : la liste des degrés des puissances parasites.
  lvar : la liste des variables $x_1, \ldots, x_n$.
sortie : la multi-résolvante de F *par* Ψ.
Début
  $resultat \leftarrow T - \Psi$
  $cm \leftarrow [\,]$
  $toutesVariables \leftarrow variables(\Psi)$
  Pour i ***Dans*** $[1..p]$
  Répète $f \leftarrow F[i]$
    $varutiles \leftarrow variablesDuBloc(\Psi, lvar, i, F)$
    $cm \leftarrow cons(modulesDeCauchy(f, varutiles), cm)$
  Fin Pour
  $resultat \leftarrow resultat \mod \; < cm >$

Tant que $arite(resultat) \neq 1$
Répète $v \leftarrow$ $mainVariable(resultat)$
$k \leftarrow$ $position(v, toutesVariables)$
$pp \leftarrow$ $lpp[k]$
$resultat \leftarrow$ $resultant(cm[k], resultat, v)^{\frac{1}{pp}} \mod < cm[k+1..n] >$
Fin Tant que
$resultat$
Fin

Où :

- $variablesDuBloc(\Psi, lvar, i, F)$ renvoie les variables du polynôme Ψ associées au polynôme $F[i]$,
- la fonction $arite(P)$ renvoie l'arité du polynôme P,
- la fonction $position(v, lv)$ renvoie la position de l'élément v dans la liste lv,
- la fonction $mainVariable(P)$ renvoie la variable principale du polynôme P,
- la notation $l[k]$ désigne le k-ième élément de la liste l,
- la notation $l[k..j]$ désigne les éléments k, $k+1$, ..., j de la liste l,
- la fonction $cons(e, l)$ renvoie la liste composée de l'élément e et de la liste l,
- la fonction $variables(P)$ renvoie la liste des variables du polynôme P,
- la fonction $modulesDeCauchy(f, [x_1, \ldots, x_n])$ renvoie la liste $[f_1(x_1), \ldots, f_n(x_n)]$ des modules de Cauchy du polynôme f,
- la notation $< cm >$ désigne l'idéal engendré par les polynômes de cm,
- l'expression $P \mod \mathcal{I}$ qui désigne la réduction du polynôme P modulo l'idéal $\mathcal{I}$ est décrite en notation 6.2,
- le calcul de $P^{\frac{1}{k}} \mod \mathcal{I}$ est réalisé à l'aide de la fonction $monicPerfectNthRoot(p, k, v)$ décrite au paragraphe (6.2).

Voyons sur un exemple comment ce calcul est réalisé avec la méthode proposée dans cet article :

Exemple 8. Prenons les polynômes $f = x^2 + 5x - 89$ et $g = x^2 - 9x - 23$ et l'invariant $\Psi = x_1x_3 + x_2x_4^3$.

Soient $f_1(x_1, x_2) = x_2 + x_1 - 9$ et $f_2(x_1) = x_1^2 - 9x_1 - 23$, les modules de Cauchy associés au polynôme f. Soient $g_1(x_3, x_4) = x_4 + x_3 + 5$ et $g_2(x_3) = x_3^2 + 5x_3 - 89$, les modules de Cauchy associés au polynôme g. Nous avons la liste $cm = [g_1, g_2, f_1, f_2]$. Calculons la suite $(R_i)_i$ correspondant à la suite des valeurs prises par la variable $resultat$ de l'algorithme 7.1.

$$\begin{aligned}
R_0 &= T - \Psi \mod < cm >= (-115x_1 + 1026)x_3 - 1015x_1 + T + 9135 \qquad , \\
R_1 &= \text{resultant}_{x_3}(R_0, g_2) \mod < f_1, f_2 > \\
&= T^2 + (-1455T + 6343920)x_1 + T + 13140T - 73902264 \qquad , \\
R_2 &= \text{resultant}_{x_1}(R_1, f_2) \\
&= T^4 + 13185T^3 - 138809523T^2 + 200429193960T + 316431786384576 \qquad .
\end{aligned}$$

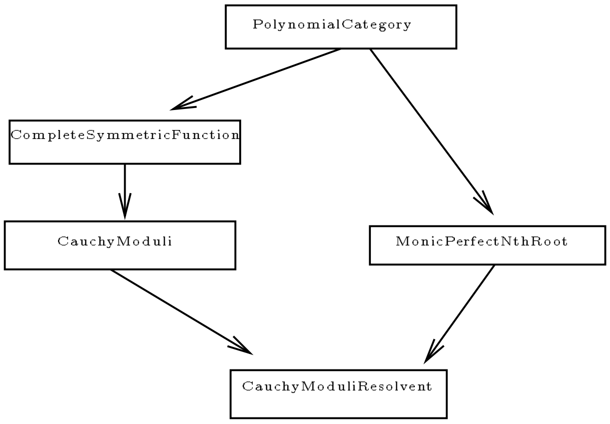
Alors $R_2 = \mathcal{L}_{\Psi,(f,g)}$.

8. Implantation

Le système de calcul formel *Axiom* a été choisi pour l'implantation des différents algorithmes (voir [14]). Ce choix est motivé par la grande modularité d'*Axiom*, parfaitement adaptée aux mathématiques. L'implantation se veut aussi générale que possible. Ainsi le choix du domaine de polynômes est laissé libre à l'utilisateur des différents programmes.

8.1. **L'organisation générale.**

Le schéma suivant décrit l'organisation des différents *paquets*(*packages*),*domaines*(*domains*) et *catégories*(*categories*).



8.2. **Un exemple commenté.**

Voici comment en *Axiom* obtenir la résolvante $\mathcal{L}_{\Theta,f}$ comme polynôme en la variable y, où $f = x^5 + 4x^4 - 17x^3 - x^2 + x + 1$ et $\Theta = x_4x_5^2 + x_3x_4^2 + x_2x_3^2 + x_5x_1^2 + x_1x_2^2$.

8.2.1. *Définition des domaines utilisés.*

La variable `R` représente l'anneau de base (dans ce cas $\mathbb{Z}$) :

```
R:=Integer
```

La variable `POL` représente l'anneau de polynôme $\mathtt{R}[x]$ (ici $\mathbb{Z}[x]$). La variable `x` est définie comme le polynôme x :

```
POL:=UnivariatePolynomial(x,R)
x:POL:=x
```

Le choix de la représentation des polynômes de $\mathtt{R}[x_1, \ldots, x_n, y]$ est relativement vaste. Des tests ont montré que les performances varient de façon significative selon l'implantation choisie. La représentation récursive des polynômes est particulièrement adaptée aux algorithmes 6.1 et 6.2. La variable OVL désigne le domaine des variables de l'anneau de polynôme $\mathtt{R}[x_1, \ldots, x_n, y]$. La variable `DP` désigne l'anneau de polynômes $\mathtt{R}[x_1, \ldots, x_6, y]$:

```
lv:=[x6,x5,x4,x3,x2,x1,y]
OVL:= OrderedVariableList(lv)
DP:=SparseMultivariatePolynomial(R,OVL)
```

Les variables `x1`,...,`x6` représentent les polynômes x_1, ... ,x_6 de `DP` :

```
x1:=monomial(1$DP,variable(x1)$(OVL)::(OVL),1)$DP;
x2:=monomial(1$DP,variable(x2)$(OVL)::(OVL),1)$DP;
x3:=monomial(1$DP,variable(x3)$(OVL)::(OVL),1)$DP;
x4:=monomial(1$DP,variable(x4)$(OVL)::(OVL),1)$DP;
x5:=monomial(1$DP,variable(x5)$(OVL)::(OVL),1)$DP;
x6:=monomial(1$DP,variable(x6)$(OVL)::(OVL),1)$DP;
```

8.2.2. *Instanciation du paquet principal.*
La variable `DRES` représente une instance du paquet `CauchyModuliResolvent` paramétré par les types définis précédemment. La signature du paquet `CauchyModuliResolvent` est la suivante :

```
CauchyModuliResolvent(R,E,OV,P,X,Y): Public == Private where
  R      :  EuclideanDomain
  OV     :  OrderedSet
  E      :  OrderedAbelianMonoidSup
  P      :  PolynomialCategory(R,E,OV)
  X      :  Symbol
  Y      :  OV
```

- `R` correspond à l'anneau de base ;
- `OV` correspond au domaine des exposants du domaine de polynômes multivariés ;
- `E` correspond au domaine des variables du domaine de polynômes multivariés ;
- `P` correspond au domaine de polynômes multivariés ;
- `X` correspond à la variable x du polynôme de départ ;

- `Y` correspond à la variable de la résolvante.

```
DRES:=CauchyModuliResolvent(R,IndexedExponents(OVL),OVL,DP,'x,'y::OVL)
```

8.2.3. *Utilisation du paquet.*

- La variable `thetaG1` représente l'invariant Θ ;
- la variable `f` représente le polynôme f ;
- la variable `lpp` représente la liste des puissances parasites (voir Théorème 6.1).

```
thetaG1:=x4*x5^2+x3*x4^2+x2*x3^2+x5*x1^2+x1*x2^2
f:=x^5+4*x^4-17*x^3-x^2+x+1
lpp:=[1,1,1,1,1,5]
```

La résolvante de `f` par `thetaG1` s'obtient par la commande suivante :

```
r1:=resolvent(f,thetaG1,lpp)$DRES
```

Le résultat est le suivant :

```
thetaG1:=x4*x5^2+x3*x4^2+x2*x3^2+x5*x1^2+x1*x2^2
                 2     2          2         2         2
   (106)  x4 x5  + x1 x5 + x3 x4  + x2 x3  + x1 x2
Type: SparseMultivariatePolynomial(Integer,OrderedVariableList [x6,x5,x4,x3,x2,x1,y])
                                         Time: 0.05 (IN) + 0.02 (OT) = 0.07 sec
```

```
f:=x^5+4*x^4-17*x^3-x^2+x+1
            5     4      3    2
   (107)  x  + 4x  - 17x  - x  + x + 1
                                          Type: UnivariatePolynomial(x,Integer)
                                                      Time: 0.03 (IN) = 0.03 sec
```

```
r1:=resolvent(f,thetaG1,[1,1,1,1,1,5])$DRES

   (108)
      24       23          22            21              20                 19
     y   - 390y   + 31765y   + 4780444y   - 647704873y   - 29613862040y
                  18                    17                      16
     + 4952685405792y   + 162526229247165y   - 19020045312189109y
                            15                         14
     - 789049792414412487y   + 28429669450472325254y
                                13                            12
     + 2033502682726556521322y   + 207403428999349413609 04y
                                   11                             10
     - 1053157069581828323779746y   - 32557903559648254823580435y
                                     9                                  8
     - 104470564098416503608748049y  + 1046832608146878671031579933 1y
                                         7                                 6
     + 1947779175080644889929551296 53y  + 14185016608356486076603065157 1y
                                          5                                    4
     - 400491723788446681756730512462 82y  - 53522901889381647545240729282254 8y
                                           3
     - 678630031684626941978361684969815y
                                              2
     + 545593557189635850529783667695314 24y
     + 618019689068720129370987442816431340y
     + 264607005419904174635295601510373654 4

Type: SparseMultivariatePolynomial(Integer,OrderedVariableList [x6,x5,x4,x3,x2,x1,y])
             Time: 0.02 (IN) + 27.36 (EV) + 0.07 (OT) + 9.32 (GC) = 36.76 sec
```

9. TEMPS DE CALCULS ET COMPARAISONS AVEC *SYM*

Le logiciel *SYM* est une extension du système de calcul formel *Maxima* (voir [18] et [21]).

Les temps donnés correspondent aux temps “machine” donnés par les différents logiciels. Ils ont été mesurés sur un Pentium Pro à 200 Mhz doté de 512 Mo de mémoire vive.

9.1. Table des polynômes invariants utilisés dans les calculs des résolvantes et multi-résolvantes.

Pour un degré donné n, la notation T_i désigne les sous-groupes transitifs de $\mathfrak{S}_n$ décrits dans l'article [7]. Les invariants ci-dessous ont été calculés par le programme **Primitive Invariants** (voir [1]).

L	H	L-primitif H-invariant
$\mathfrak{S}_6$	T_1	$x_5x_6^2 + x_4x_5^2 + x_3x_4^2 + x_2x_3^2 + x_6x_1^2 + x_1x_2^2$
$\mathfrak{S}_6$	T_2	$(x_5x_6 + x_3x_4 + x_1x_2) + 2(x_4x_6 + x_3x_5 + x_2x_6 + x_2x_4 + x_1x_5 + x_1x_3) + 3(x_4x_5 + x_2x_3 + x_1x_6)$
$\mathfrak{S}_6$	T_3	$(x_5x_6 + x_4x_5 + x_3x_4 + x_2x_3 + x_1x_6 + x_1x_2)$
$\mathfrak{S}_6$	T_4	$(x_6x_5^2 + x_4x_6^2 + x_5x_4^2 + x_3x_5^2 + x_4x_3^2 + x_6x_2^2 + x_2x_4^2 + x_3x_2^2 + x_1x_6^2 + x_5x_1^2 + x_1x_3^2 + x_2x_1^2) + 2(x_3x_5x_6 + x_2x_4x_5 + x_2x_3x_6 + x_1x_4x_6 + x_1x_3x_4 + x_1x_2x_5) + 3(x_3x_6^2 + x_6x_3^2 + x_2x_5^2 + x_5x_2^2 + x_1x_4^2 + x_4x_1^2) + 4(x_3x_4x_6 + x_2x_5x_6 + x_2x_3x_5 + x_1x_4x_5 + x_1x_3x_6 + x_1x_2x_4) + 5(x_3x_4x_5 + x_2x_4x_6 + x_1x_5x_6 + x_1x_2x_3)$
$\mathfrak{S}_6$	T_5	$x_4x_6^2 + x_3x_5^2 + x_6x_2^2 + x_2x_4^2 + x_5x_1^2 + x_1x_3^2$
$\mathfrak{S}_6$	T_6	$x_3x_5x_6 + x_2x_4x_5 + x_2x_3x_6 + x_1x_4x_6 + x_1x_3x_4 + x_1x_2x_5$
$\mathfrak{S}_6$	T_{12}	$x_3x_5x_6 + x_3x_4x_6 + x_2x_5x_6 + x_2x_4x_5 + x_2x_3x_4 + x_1x_4x_6 + x_1x_4x_5 + x_1x_3x_5 + x_1x_2x_6 + x_1x_2x_3$
$\mathfrak{S}_6$	T_{14}	$x_3x_5x_4x_6(x_3x_5+x_4x_6) + x_2x_4x_5x_6(x_2x_4+x_5x_6) + x_4x_5x_1x_6(x_4x_5+x_1x_6) + x_2x_5x_3x_6(x_2x_5+x_3x_6) + x_1x_3x_5x_6(x_1x_3+x_5x_6) + x_2x_6x_1x_5(x_2x_6+x_1x_5) + x_3x_4x_2x_6(x_3x_4+x_2x_6) + x_1x_4x_3x_6(x_1x_4+x_3x_6) + x_1x_2x_4x_6(x_1x_2+x_4x_6) + x_2x_3x_1x_6(x_2x_3+x_1x_6) + x_2x_3x_4x_5(x_2x_3+x_4x_5) + x_3x_4x_1x_5(x_3x_4+x_1x_5) + x_1x_4x_2x_5(x_1x_4+x_2x_5) + x_1x_2x_3x_5(x_1x_2+x_3x_5) + x_1x_3x_2x_4(x_1x_3+x_2x_4)$
$\mathfrak{S}_6$	$\mathfrak{S}_2 \times \mathfrak{S}_2 \times \mathfrak{S}_2$	$x_1x_2 - x_3x_4$
$\mathfrak{S}_8$	T_4	$x_7x_8 + x_3x_6 + x_2x_5 + x_1x_4 + 2(x_6x_8 + x_3x_5 + x_2x_4 + x_1x_7)$
$\mathfrak{S}_8$	T_{47}	$x_1x_2x_3x_4 + x_5x_6x_7x_8$
$\mathfrak{S}_9$	T_{14}	$(x_4x_6x_7x_9 + x_4x_5x_6x_8 + x_3x_5x_8x_9 + x_3x_5x_6x_7 + x_3x_4x_7x_8 + x_2x_7x_8x_9 + x_2x_5x_6x_9 + x_2x_4x_5x_7 + x_2x_3x_6x_8 + x_2x_3x_4x_9 + x_1x_6x_8x_9 + x_1x_5x_7x_8 + x_1x_4x_5x_9 + x_1x_3x_7x_9 + x_1x_3x_4x_6 + x_1x_2x_6x_7 + x_1x_2x_4x_8 + x_1x_2x_3x_5) + 2(x_4x_5x_7x_8 + x_4x_5x_6x_9 + x_3x_7x_8x_9 + x_3x_5x_6x_8 + x_3x_4x_6x_7 + x_2x_6x_8x_9 + x_2x_5x_6x_7 + x_2x_4x_7x_9 + x_2x_3x_5x_9 + x_2x_3x_4x_8 + x_1x_6x_7x_9 + x_1x_5x_8x_9 + x_1x_4x_6x_8 + x_1x_3x_5x_7 + x_1x_3x_4x_9 + x_1x_2x_7x_8 + x_1x_2x_4x_5 + x_1x_2x_3x_6)$
$\mathfrak{S}_9$	T_{10}	$x_6x_7x_8x_9 + x_5x_6x_7x_8 + x_4x_5x_8x_9 + x_4x_5x_6x_7 + x_3x_5x_7x_9 + x_3x_4x_8x_9 + x_3x_4x_7x_8 + x_3x_4x_5x_6 + x_2x_5x_7x_9 + x_2x_4x_7x_9 + x_2x_4x_6x_9 + x_2x_4x_6x_8 + x_2x_3x_7x_8 + x_2x_3x_6x_7 + x_2x_3x_4x_5 + x_1x_7x_8x_9 + x_1x_5x_6x_9 + x_1x_4x_6x_8 + x_1x_4x_5x_9 + x_1x_3x_6x_8 + x_1x_3x_5x_8 + x_1x_3x_5x_7 + x_1x_2x_8x_9 + x_1x_2x_6x_7 + x_1x_2x_5x_6 + x_1x_2x_3x_9 + x_1x_2x_3x_4$
$\mathfrak{S}_r$	$A_r \times \mathfrak{S}_r$	$\prod_{1<=i<j<r} (x_i - x_j)$

9.2. **Calcul de résolvante absolue.**

Dans cette section sont comparés les algorithmes 6.1 et 6.2 et le logiciel *SYM* qui intègre différents algorithmes de calcul de résolvantes dont un algorithme général basé sur le calcul des fonctions puissances des racines de la résolvante. Lorsque la forme de l'invariant induit des formules closes de ces fonctions puissances, le logiciel *SYM* offre des algorithmes spécialisés.

9.2.1. *Algorithmes généraux.*

Prenons $f = x^6 - 243x^2 + 729$.

Groupes	Algorithme 6.2	Algorithme 6.1	SYM
T_1	41 minutes	$\star$	$\star$
T_2	41 minutes	$\star$	$\star$
T_3	57 minutes	$\star$	$\star$
T_4	1 heure et 20 minutes	$\star$	$\star$
T_5	1 heure et 30 minutes	6 heures	$\star$
T_6	2 heures	$\star$	$\star$
T_{12}	2 heures	5 heures et 55 minutes	7 heures 7 minutes
T_{14}	11 minutes	6 heures	$\star$
$\mathfrak{S}_2 \times \mathfrak{S}_2 \times \mathfrak{S}_2$	3 minutes 10 secondes	3 minutes et 30 secondes	6 minutes et 8 secondes

$\star$: calcul non terminé.

9.2.2. *Algorithmes spécialisés.*

Les résultats suivants montrent les performances des différents algorithmes de l'article et des algorithmes spécialisés de SYM, pour des invariants de faible arité. Fixons $f = x^7 + 8x^6 - 5x^4 - 18x^2 + x - 1$

Invariants	Algorithme 6.2	Algorithme 6.1	SYM
$x_1x_2 - x_3x_4$	3 minutes 10 secondes		2,5 secondes
$x_1 + 2x_2 + 3x_3$	171 secondes	128 secondes	2 heures 23 minutes
$x_1x_2^2x_3^3$	18 minutes et 26 secondes	48 minutes	16 minutes et 16 secondes
$x_1x_2x_3$	200 secondes	250 secondes	2,5 secondes

9.3. **Calculs de multi-résolvantes absolues :**

Fixons $F = (f^{(1)}, f^{(2)}, f^{(3)}, f^{(4)})$ où :

$$\begin{aligned}
f^{(1)} &= x^3 + 6095519139889209290x^2 + 14888192843637255648x + 3495901233607471790 \\
f^{(2)} &= x^3 + 17595319154460550344x^2 + 11186788950436463054x + 11049608948604938510 \\
f^{(3)} &= x + 8358981594339387494 \\
f^{(4)} &= x + 4843668258729955558
\end{aligned}$$

Les temps mesurés sont les suivant :

Groupes	Algorithme 7.1	SYM
$A_4 \times \mathfrak{S}_4$	0,27 seconde	15 heures et 20 minutes
T_{47}	0,06 seconde	$\star$
T_4	5,46 secondes	$\star$

$\star$: temps de calcul supérieur à 2 jours.

Fixons maintenant $P = (p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)})$

$$\begin{aligned}
p^{(1)} &= x^4 + 119x^3 + 442x^2 + x - 1\\
p^{(2)} &= x^3 + 1759531915446055034 4x^2 + 1118678895043646305 4x + 1104960894860493851 0\\
p^{(3)} &= x + 8358981594339387494\\
p^{(4)} &= x + 4843668258729955558
\end{aligned}$$

Groupes	Algorithme 7.1
T_{14}	0,05 seconde
T_{10}	1 heure 28 minutes

10. Conclusions

Cet article a montré comment éviter les deux principaux défauts des méthodes de calcul de résolvantes basées sur le résultant (facteurs parasites et puissances superflues). L'algorithme auquel nous avons aboutit (algorithme 6.2) a permis de calculer des résolvantes qui n'étaient pas calculables par d'autres méthodes (par exemple : la résolvante de T_1 en degré 6). Les algorithmes spécialisés implantés dans SYM peuvent être beaucoup plus rapides (cas des résolvantes produits, symétriques) mais ce n'est pas toujours le cas (résolvantes linéaires, résolvantes monomiales). Pour les invariants de grande arité, les temps de calcul restent élevés et au delà du degré 7, de nombreux exemples n'aboutissent pas. En revanche, l'algorithme de multi-résolvante (algorithme 7.1) est très efficace.

References

[1] Abdeljaouad (Ines). – *Calcul d'invariants primitifs de groupes finis.* – Rapport technique, LIP6, 1997.

[2] Ampère (André-Marie). – *Fonctions interpolaires.* – Annales de M.Gergonne, 1826.

[3] Arnaudiès (Jean-Marie). – *Elimination.* – Thèse de PhD, Université Paul Sabatier deToulouse, 1989.

[4] Arnaudiès (Jean-Marie). – *Théorème de Bézout, Formule de Poisson-Perron et Courbes Rationnelles Planes Génériques.* – Rapport technique n° 92-24, LITP, 1992.

[5] Arnaudiès (Jean-Marie) et Valibouze (Annick). – *Résolvantes de Lagrange.* – Rapport interne n° 93-61, Laboratoire Informatique Théorique et Programmation, décembre 1993.

[6] Becker (Thomas), Weispfenning (Volker) et Kredel (Heinz). – *Gröbner bases : a computational approach to commutative algebra.* – Springer-Verlag, 1993.

[7] Butler (Gregory) et McKay (John). – The transitive groups of degree up to eleven. *Commun. Algebra*, vol. 11, 1983, pp. 863–911.

[8] Cauchy (Augustin Louis). – *Usage des fonctions interpolaires dans la détermination des fonctions symétriques des racines d'une équation algébrique donnée.*, p. 473. – Paris, Gauthier-Villars, 1882.

[9] de Lagrange (Joseph-Louis). – *Réflexions sur la résolution algébrique des équations*, pp. 205–421. – Prussian Academy, 1770, *Œuvres de Lagrange*, volume 4.

[10] Eichenlaub (Yves) et Olivier (Michel). – Galp. – Logiciel, 1996. http://www.math.u-bordeaux.fr/A2X/Logiciels.html.

[11] The GAP Group, Lehrstuhl D für Mathematik, RWTH Aachen, Germany and School of Mathematical and Computational Sciences, U. St. Andrews, Scotland. – *GAP – Groups, Algorithms, and Programming, Version 4*, 1997.

[12] Girstmair (Kurt). – On the computation of resolvents and Galois groups. *Manuscr. Math.*, vol. 43, 1983, pp. 289–307.

[13] Henrici (Peter). – Automatic computations with power series. *Journal of the Association for Computing Machinery*, vol. 3, n° 1, January 1956, pp. 10–15.

[14] Jenks (Richard D.) et Sutor (Robert S.). – *Axiom : The scientific computation system.* – NAG, New-York, 1992.
[15] Lascoux (Alain) et Pragacz (P.). – S-function series. *J. Phys.*, vol. 21, n° 22, 1988, pp. 4105–4114.
[16] Lehobey (Frédéric). – Resolvent computations by resultants without extraneous powers. *In : Proceedings of ISSAC'97.* ACM.
[17] Schönert (Martin) et al. – *GAP – Groups, Algorithms, and Programming.* – Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, 1995, fifth édition.
[18] Schelter (William). – MAXIMA. – GNU software. maintenu par William Schelter.
[19] Soicher (Leonard H.). – An algorithm for computing Galois groups. *In : Computational Group Theory*, pp. 291–296. – London, Academic Press, 1984.
[20] Stauduhar (Richard P.). – The determination of Galois groups. *Mathematics of Computation*, vol. 27, n° 124, October 1973, pp. 981–996.
[21] Valibouze (Annick). – SYM, symbolic computation with symmetric polynomials an extension to MACSYMA. *In : Computers and Mathematics (Proceeding of a conference 13-17, 1989 Massachusetts Institute of Technology, Cambridge, MA).* pp. 308–320. – Springer-Verlag.
[22] Valibouze (Annick). – Etude des relations algebriques entre les racines d'un polynôme d'une variable. *Bulletin of the Belgian Math. Soc.*, noto appear, 1998.

Les rapports internes et techniques sont disponibles sur le serveur du LIP6 (http://www.lip6.fr/reports/)

LIP6., Université Paris VI, 4 place Jussieu, F-75252 Paris Cedex 05
E-mail address: avb@medicis.polytechnique.fr