Résolvantes et fonctions symétriques

Annick Valibouze
L.I.T.P
Université Pierre et Marie Curie
4, Place Jussieu
75252 Paris Cedex 05
GRECO DE CALCUL FORMEL No 60
UUCP: avb@litp.univ-p6-7.fr BITNET : avb@frunip11.bitnet

Abstract

In the present paper a model of transformations of polynomial equations (the so called "direct image"model) is studied. We express, in this model, some minimal polynomials and some resolvents relative to the Galois group of a polynomial in order to use a general algorithm of resolution. This algorithm can be effectively computed in MACSYMA with the extension SYM that manipulates symmetric polynomials. We give a few examples obtained by specializing the general algorithm for the Galois resolvent.

INTRODUCTION

Soit k un corps. On se donne un polynôme p de degré n et irréductible sur $k[x]$.
Soit $K = k(\alpha_1, \ldots, \alpha_n)$ le corps de l'ensemble $rac(p)$ des racines de p. Le groupe noté $Gal(K/k)$ des automorphismes de K laissant k invariant est le groupe de Galois de l'extension K/k. Le corps K étant le corps des racines d'un polynôme p, on notera également ce groupe $Gal_k(p)$, le groupe de Galois de p sur le corps k. Le calcul de son ordre ou le test de son inclusion dans un autre groupe (comprenant entre autre celle de sa résolubilité) peuvent se réaliser à l'aide de résolvantes.
Afin de simplifier les notations nous prendrons la convention suivante : soient S_n le groupe symétrique d'ordre n et $S_n(p)$ le groupe symétrique agissant sur les racines de p. Nous utiliserons implicitement l'isomorphisme de $S_n(p)$ dans S_n qui a σ associe τ tel que si $\sigma(\alpha_i) = \alpha_j$ alors $\tau(i) = j$, pour tout i, j dans $[1, \ldots, n]$.

1 Rappel de la définition d'image directe

Cette définition a été introduite dans [G,L,V].
Soit k un corps, et K la clôture algébrique de k. On note $\mathbf{N}$ l'ensemble des entiers naturels positifs.

Donnons-nous un entier s et considérons l'élément $C = (c_1, c_2, \ldots, c_s)$ de $\mathbf{N}^s$. Son *poids* $\mid C \mid$ est la somme $c_1 + \cdots + c_s$. Soit R_C l'algèbre de polynômes $k[x_1^{(1)}, \ldots, x_{c_1}^{(1)}, \ldots, x_1^{(s)}, \ldots, x_{c_s}^{(s)}]$. Une *transformation de type* C n'est rien d'autre qu'un polynôme f de R_C, auquel on associe une application :

$$f : K^{c_1} \times K^{c_2} \times \ldots \times K^{c_s} \longrightarrow K$$

que l'on désigne également par f.

L'ensemble $\mathbf{N}^s$ est naturellement partiellement ordonné. Si M est plus grand que C, cela induit une inclusion naturelle $R_C \subseteq R_M$.

Le *multidegré* D du s-uplet de polynômes d'une variable $(P_1, \ldots, P_s)$ est la suite $(d_1, \ldots, d_s)$ de leurs degrés.

Maintenant nous pouvons définir l'*image directe* ou *transformation* :

$$f_* : k[x]^s \longrightarrow k[x]$$

qui associe à chaque s-uplet ordonné de polynômes unitaires $P = (P_1, \ldots, P_s)$, de multidegré D plus grand que C, le polynôme unitaire $f_*(P)$ obtenu de la manière suivante : informellement parlé, c'est le polynôme d'une variable dont les racines sont les éléments de K obtenus par substitution dans f des variables $x_i^{(j)}$ par les différentes racines de P_j. Nous donnons maintenant une définition précise de $f_*(P)$:

Puisque le multidegré D est plus grand que C, on peut associer à chaque polynôme P_j de degré d_j l'ensemble $r(P_j) = (a_1^{(j)}, \ldots, a_{d_j}^{(j)})$ de ses d_j racines dans K ordonnées de façon arbitraire $(1 \leq j \leq s)$. Choisissons une application d'évaluation $E_a : R_D \longrightarrow K$ qui est un homomorphisme d'algèbre envoyant la variable $x_i^{(j)}$ sur $a_i^{(j)}$. Le produit S_D de groupes symétriques $S_{d_1} \times \cdots \times S_{d_s}$ agit naturellement sur R_D : par l'inclusion naturelle $R_C \subseteq R_D$, f devient un élément de R_D. Soit $O_{S_D}(f)$ son orbite sous S_D.

Finalement $f_*(P)$ est le polynôme dont les racines sont les images par l'application d'évaluation E_a des éléments de l'orbite de f, i.e. :

$$f_*(P)(x) = \prod_{g \in O_{S_D}(f)} (x - E_a(g)).$$

Pour ce présent papier les applications ne concernent que le cas $s = 1$ et $P = (p)$, mais l'algorithme `DIRECT` proposé ici pour le calcul de $f_*(p)$ se généralise simplement au cas $s > 1$.

2 Propriétés d'invariance

Nous énonçons ici des propriétés bien connues, permettant de justifier que les calculs de l'algorithme `DIRECT`, donn/'e plus loin, ne font pas sortir du corps de base.

Fixons une fois pour toute une évaluation E des variables $x_1, \ldots, x_n$ en les racines du polynôme p. A partir de cette évaluation, si g est un élément (resp. un ensemble d'éléments) du corps $k(x_1, \ldots, x_n)$, on notera $\tilde{g}$ la valeur prise par g (resp. l'ensemble des valeurs prises par les éléments de g) en les racines de p via l'application d'évaluation E.

Remarque : Avec la convention d'identification de $S_n(p)$ à S_n, pour tout sous-groupe H de S_n on a $\widetilde{O_H(f)} = O_H(\tilde{f})$.

Donnons-nous f un élément de $k[x_1, \ldots, x_n]$. Nous supposons d'avance que les fonctions symétriques sont des polynômes.

Proposition 2.1 *Soient H un sous-groupe de S_n, et c le cardinal de l'orbite $O_H(\tilde{f})$. Si S est une fonction symétrique de c variables, alors $S(O_H(\tilde{f}))$ est invariante sous l'action de H sur les racines de p. En d'autres termes $S(O_H(\tilde{f}))$ appartient à l'ensemble K^H des invariants par H dans le corps des racines de p.*

Soit S une fonction symétrique. Si A est un ensemble de fonctions quelconques, on note $S(A)$ l'évaluation de S en les éléments de A. Si q est un polynôme d'une variable, on note $S(q)$ l'évaluation de S en les racines de q (i.e. $S(q) = S(rac(q))$).

preuve : Comme S est une fonction symétrique, il suffit de montrer que l'orbite $O_H(\tilde{f})$ reste invariante sous l'action de H. Ce qui est trivial puisque H est un groupe.

Corollaire 2.1 *Avec les hypothèses de la proposition 2.1, si H est le groupe $Gal(K/k)$, alors $S(O_{Gal(K/k)}(\tilde{f}))$ appartient au corps de base k.*

preuve : En effet d'après le théorème de Galois $k = K^{Gal(K/k)}$.

Corollaire 2.2 *Supposons que $O_{S_n}(f)$ comporte c éléments $f_1 \dots f_c$. Soit S une fonction symétrique de c variables. Alors $S(O_{S_n}(f))$ est symétrique en $x_1, \dots, x_n$.*

preuve : En effet, il suffit d'appliquer la proposition 3.1 à $H = S_n$ et prendre p le polynôme dont les racines sont $x_1, \dots, x_n$.

Corollaire 2.3 *Avec les hypothèses du corollaire précédent, si on évalue $S(O_{S_n}(f))$ en les racines de p alors le résultat $S(\tilde{f}_1, \dots, \tilde{f}_c)$ appartient au corps k.*

preuve : Comme d'après le corollaire précédent $S(O_{S_n}(f))$ appartient à $k(x_1, \dots, x_n)^{S_n}$, on peut l' exprimer, d'après le théorème fondamental des fonctions symétriques, comme fraction rationnelle sur k en les fonctions symétriques élémentaires des x_i. Alors l'évaluation de $S(O_{S_n}(f))$ en les racines du polynôme p revient à substituer aux fonctions symétriques élémentaires des x_i celles des racines de p qui sont (à un signe près) ses coefficients.

Remarques : En se placeant successivement dans s extensions associées aux blocs des racines des s polynômes $P_1, \dots, P_s$, ce dernier corollaire montre que $f_*(P_1, \dots, P_s)$ définie au paragraphe précédent a bien ses coefficients dans le corps de base k. On peut de même remarquer que si le groupe choisi pour l'orbite de f était le produit des groupes de Galois des polynômes $P_1, \dots, P_s$ au lieu de S_D, le polynôme $f_*(P_1, \dots, P_s)$, aurait également ses coefficients dans le corps de base. Mais alors, nous n'aurions pas de moyen de le calculer sans connaître les s groupes de Galois.

3 Polynômes minimaux

Proposition 3.1 *([Bastida] p.113) Soient F un corps et G le groupe des automorphismes de F, et $\alpha \in F$. Alors α est algébrique sur F^G si et seulement si $O_G(\alpha)$ est fini, et dans ce cas $\prod_{\beta \in O_G(\alpha)}(X - \beta)$ est son polynôme minimal sur F^G.*

Corollaire 3.1 *Soit H un sous-groupe de S_n et soit $h \in k(x_1, \dots, x_n)^H$, alors $h_*(\prod_{i=1}^n(x - x_i))$ est le polynôme minimal de $h(x_1, \dots, x_n)$ sur $k(x_1, x_2, \dots, x_n)^{S_n}$.*

preuve : Notons G le groupe des automorphismes de $k(x_1, \dots, x_n)^H$ et S_n/H une transversale de H dans S_n. Comme $O_{S_n}(h) = O_{S_n/H}(h) = O_G(h)$, puisque h est invariant par H, alors $h_*(\prod_{i=1}^n(x - x_i)) = \prod_{g \in O_{S_n/H}(h)}(x - g)$.

Ce corollaire permet d'exprimer la résolvante de Galois généralisée en termes d'image directe comme nous le verrons plus loin.

Revenons aux s polynômes $P_1, \dots, P_s$ du paragraphe précédent et considérons l'élément
$\gamma = f(\alpha_1^{(1)}, \dots, \alpha_{c_1}^{(1)}, \dots, \alpha_1^{(s)}, \dots, \alpha_{c_s}^{(s)})$ de l'extension $k(rac(P_1), \dots, rac(P_s))$.

Corollaire 3.2 *Si $Gal_k(P_i) = S_{d_i}$ pour tout i de 1 à s, alors $f_*(P)$ est le polynôme minimal de γ sur k.*

Si le groupe de Galois de chaque polynôme P_i n'est pas S_{d_i}, on peut trouver le polynôme minimal de γ sur k avec des intervals I_i^j isolants chacune des racines $\alpha_i(j)$ choisies ($j = 1 \dots s$ et $i = 1, \dots, c_s$). Pour cela il suffit de remplacer l'étape (1) de l'algorithme 1 de R. [Loos] (p. 180-181) par le calcul de $f_*(P)$, de remplacer la factorisation “square-free” de l'étape (2) par une factorisation complète sur k, de généraliser l'étape (3), et de prendre $K = f(I_1^{(1)}, \dots, I_{c_1}^{(1)}, \dots, I_1^{(s)}, \dots, I_{c_s}^{(s)})$ dans la quatrième étape.

On retrouve ici que la résolvante de Galois (voir plus loin) est le polynôme minimal de l'élément primitif de l'extension $k(\alpha_1, \dots, \alpha_n)$.

4 Résolvantes en termes d'image directe

Nous conservons ici les notations du paragraphe précédent.
Nous allons donner des définitions connues de certaines résolvantes, mais en les exprimant chacune comme une image directe afin de pouvoir leur appliquer l'algorithme de résolution énoncé au paragraphe suivant. Nous énonçons également une formule permettant d'accélérer cet algorithme dans le cas des résolvantes de Galois et de Lagrange.

Dans toute cette partie on se donne un polynôme p irréductible dans $k[x]$ et de degré n.

4.1 Résolvante de Galois

Définition 4.1 *Soient $\underline{t} = (t_1, \ldots, t_n)$ un n-uplet d'entiers et $f(x_1, \ldots, x_n) = \sum_{i=1}^n t_i x_i$. Si $O_{S_n}(\tilde{f})$ est de cardinal $n!$ (i.e. $f_*(p)$ est sans facteur carré), alors f est appelée une fonction $n!$-valeurs.*

L'existence d'une telle fonction est connue, et il semble qu'il existe un nombre assez petit de valeurs de $\underline{t}$ qui ne vérifient pas cela.

Définition 4.2 *Soit f une fonction $n!$-valeurs. Le facteur irréductible dans $k[x]$ de $f_*(p)$ ayant $\tilde{f}$ comme racine est appelé la résolvante de Galois de p pour le corps k.*

Soit $G(x) = (x - \sigma_1 \tilde{f})(x - \sigma_2 \tilde{f}) \ldots (x - \sigma_r \tilde{f})$ la résolvante de Galois. Alors on sait que $Gal_k(p) = \{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ et que les autres facteurs irréductibles de $f_*(p)$ étant associés à des sous-groupes conjugués de $Gal_k(p)$ sous S_n. La factorisation de $f_*(p)$ sur k permet donc l'obtention de l'ordre du groupe de Galois.

Remarque : Soient q le polynôme dont les racines sont $t_1 \ldots t_n$, $g(\underline{u}, \underline{x}) = \sum_{i=1}^n u_i x_i$ et $h(\underline{u}) = \sum_{i=1}^n u_i \alpha_i$; alors on a $f_*(p) = g_*(p, q) = h_*(q)$.

Rappelons maintenant quelques définitions standards sur les fonctions symétriques (voir [Macdonald]).

Une *partition* est une séquence finie ou infinie d'entiers positifs rangés dans un ordre décroissant et contenant seulement un nombre fini de termes non nuls, appelé *longueur* de la partition. Si a est un n-uplet d'entiers positifs on notera par $P(a)$ la partition engendrée par une permutation de a.

Définition 4.3 *Soit $\underline{x} = (x_1, \ldots, x_n)$ et soit $I = (i_1, \ldots, i_n)$ une partition de longueur inférieure à n. Alors la forme monomiale, $M_I(\underline{x})$, donnée par la somme des monômes de l'orbite de $\underline{x}^I$ sous l'action de S_n est définie par :*

$$M_I(\underline{x}) = \sum_{\sigma \in S_n / G(I)} \underline{x}^{\sigma(I)},$$

où $G(I)$ est le stabilisateur de I sous l'action de S_n.

Les formes monomiales $M_{(r)}(\underline{x}) = \sum_{i=1}^n x_i^r$ sont appelées *fonctions puissance* (ou de Newton) et notées $p_r(\underline{x})$. De même les formes monomiales $e_r = M_{\underbrace{(1,1,\ldots,1)}_{r}}$ sont appelées *fonctions symétriques élémentaires*. Si q est un polynôme de degré n dans $k[x]$ et $r \leq n$ alors $(-1)^r e_r(rac(q))$ est le coefficient de x^r dans q.

Propriété 4.1 *Pour tout entier r, si f est une fonction $n!$-valeurs alors :*

$$p_r(f_*) = \sum \#G(J) \begin{pmatrix} r \\ j_1, \ldots, j_n \end{pmatrix} M_J(\underline{t}) M_J(\underline{x}) \tag{1}$$

où la somme est étendue à toutes les partitions $J = (j_1, \ldots, j_n)$ de poids r et de longueur inférieure ou égale à n.

preuve de (1) : On appelle Π_r la partie de $\mathbf{N}^n$ formée des éléments de poids r. Nous adopterons la notation σf pour la fonction associée à $f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$.
Notons $m(\underline{a}) = \begin{pmatrix} r \\ j_1, \ldots, j_n \end{pmatrix}$.

Comme par la définition des fonctions puissance $p_r(f_*) = \sum_{\sigma\in S_n}(\sigma f)^r$, nous calculons tout d'abord $(\sigma f)^r$ en posant pour $1 \leq i \leq n$, $\alpha_{(\sigma,i)} = x_{\sigma(i)}t_i$:

$$\begin{aligned}(\sigma f)^r &= (\alpha_{(\sigma,1)} + \cdots + \alpha_{(\sigma,n)})^r \\ &= \sum_{a\in\Pi_r} m_{P(a)}\alpha_{(\sigma,1)}^{a_1}\cdots\alpha_{(\sigma,n)}^{a_n},\end{aligned}$$

d'après la formule multinomiale et en utilisant $m(\underline{a}) = m(P(\underline{a}))$. D'où

$$(\sigma f)^r = \sum_{a\in\Pi_r} m_{P(a)}(\sigma\underline{x})^a\underline{t}^a.$$

On en déduit donc la fonction puissance :

$$\begin{aligned}p_r(f_*) &= \sum_{\sigma\in S_n}\sum_{a\in\Pi_r} m_{P(a)}(\sigma\underline{x})^a\underline{t}^a \\ &= \sum_{a\in\Pi_r} m_{P(a)}\underline{t}^a\sum_{\sigma\in S_n}(\sigma\underline{x})^a \\ &= \sum_{a\in\Pi_r} m_{P(a)}\underline{t}^a\ \#G(P(a))M_{P(a)}(\underline{x}).\end{aligned}$$

En posant pour toutes les suites a d'une même orbite $J = P(a)$ puis, en factorisant, il vient :

$$p_r(f_*) = \sum \#G(J)\, m_J M_J(\underline{x}) \sum_{a\in\mathbf{N}^n\ ,\ P(a)=J} \underline{t}^a,$$

où la première somme est étendue à toutes les partitions J de $\mathbf{N}^n$ de poids r.
La constatation $\sum_{a\in\mathbf{N}^n\ ,\ P(a)=J}\underline{t}^a = M_J(\underline{t})$ termine la démonstration.

Conclusion : Cette formule permet d'avoir les fonctions puissance des racines de $f_*(p)$. Si on les obtient effectivement, on en déduit facilement les coefficients du polynôme f_* (voir plus loin).

Lorsque $n = 4$, on obtient des informations (voir [Bastida] p.141) sur le groupe de Galois d'un polynôme p à l'aide de l'ordre du groupe de Galois de sa résolvante cubique. Dans le cas où cet ordre est 2, on ne peut pas savoir si $Gal_k(p)$ est le groupe diédral d'ordre 8 ou le groupe cyclique d'ordre 4. Mais alors, si on calcule l'ordre de $Gal_k(p)$ on peut lever l'indétermination (voir une application plus loin).

4.2 Résolvante de Lagrange (Vandermonde)

Définition 4.4 *Soit α une racine primitive n-ième de l'unité et $k = k(\alpha)$. Pour $f(\underline{x}) = \sum_{i=1}^n \alpha^i x_i$, l'image directe $f_*(p)$ est appelée la résolvante de Lagrange.*

Il est clair que si m n'est pas un multiple de n alors le coefficient de x^m dans $f_*(p)(x)$ est nul. Ce coefficient étant à un signe près la $(n! - m)$-ième fonction symétrique élémentaire des racines de $f_*(p)$ on en déduit avec les relations de [Girard]-Newton :

$$\sum_{i=0}^{r}(-1)^i e_i p_{r-i} = 0 \text{ pour tout } r \geq 0,$$

que $p_r(f_*(p))$ est nulle si r n'est pas un multiple de n. Pour p_r non nulle, la formule (1) donnée au paragraphe précédent s'applique ici avec $\underline{t} = \underline{\alpha}$.

Si on pose $E_r = ne_{rn}(f_*)$ et $P_r = p_{rn}(f_*)$ alors les relations de Girard-Newton deviennent :

$$rE_r = \sum_{i=0}^{r-1}(-1)^{n(r-i)+1}P_{r-i}E_i \tag{2}$$

Cette formule sera utile lors de la dernière étape de l'algorithme.

4.3 Résolvante de Galois généralisée

Dans le paragraphe sur la résolvante de Galois, si f est une fonction $n!$-valeurs alors $f(\alpha_1, \ldots, \alpha_n)$ est un élément primitif de $k(\alpha_1, \ldots, \alpha_n)$, et la résolvante de Galois est son polynôme minimal, ce qui confirme que le degré de l'extension est l'ordre du groupe de Galois (théorème de [Dedekind, §.166]-[Artin, sec. 2.H]). Maintenant considérons H un sous-groupe de S_n et la tour d'extensions

$$k(e_1, \ldots, e_n) = k(x_1, \ldots, x_n)^{S_n} \subset k(x_1, \ldots, x_n)^H \subset k(x_1, \ldots, x_n),$$

où $e_1, \ldots, e_n$ sont les fonctions symétriques élémentaires de $x_1, \ldots, x_n$.

Définition 4.5 *Soit $f(x_1, \ldots, x_n)$ un élément primitif de $k(x_1, \ldots, x_n)^H$ sur $k(e_1, \ldots, e_n)$, alors $f_*(p)$ est la résolvante de Galois généralisée.*

Nous rappelons ici que nous identifions les groupes agissants sur $[1, 2, \ldots, n]$ de ceux agissant sur les racines de p par l'isomorphisme donné dans l'introduction.

Remarque : Comme d'après 3.1 $f_*(\prod_{i=1}^n (x - x_i))$ est le polynôme minimal de $f(x_1, \ldots, x_n)$ sur $k(x_1, x_2, \ldots, x_n)^{S_n}$ et que f est primitif, le degré de $f_*(p)$ (i.e. le cardinal de $O_{S_n}(f)$) est l'indice du groupe H (le degré de l'extension).

Propriété 4.2 *Soit f un élément primitif de $k(x_1, \ldots, x_n)^H$. $Gal_k(p)$ est un sous-groupe d'un des groupes conjugués de H sous S_n si et seulement si il existe σ dans S_n tel que pour tout g dans $Gal_k(p)$ on ait $g(\sigma f) = \sigma f$.*

preuve : Le fait que $Gal_k(p)$ soit inclus dans H ou dans un de ses conjugués dépend de l'isomorphisme choisit pour identifier $Gal_k(p)$ à un sous-groupe de S_n. En choisissant le bon isomorphisme il suffit de montrer que $Gal_k(p) \subset H$ si et seulement si pour tout g dans $Gal_k(p)$ on a $gf = f$. De gauche à droite, c'est trivial pour tout f dans $k(x_1, \ldots, x_n)^H$ et de droite à gauche également puisque f est un élément primitif.

Mais cette proposition n'offre pas de méthode calculatoire, comme ce corollaire qui en découle :

Corollaire 4.1 *Soit f un élément primitif de $k(x_1, \ldots, x_n)^H$. Si $Gal_k(p)$ est un sous groupe d'un des groupes conjugués de H sous S_n alors $f_*(p)$ a une racine dans k. Si $f_*(p)$ n'a pas de racines multiples et qu'une de ses racines est dans k, alors $Gal_k(p)$ est un sous-groupe d'un des groupes conjugués de H sous S_n.*

Remarque : Ce corollaire est déjà connu (voir [Arnaudiès] et [Lefton]).

preuve : Comme précédemment, nous supposerons que l'isomorphisme choisit est le bon. Si le groupe de Galois est un sous groupe de H alors comme d'après la proposition précédente la racine $\tilde{f}$ de $f_*(p)$ est invariante sous l'action de $Gal_k(p)$, elle appartient donc à k. Maintenant si $\tilde{f}$ est dans k, elle est invariante par $Gal_k(p)$. Or si $g\tilde{f} = \tilde{f}$ alors ou bien $gf = f$, ou bien $gf = h \neq f$ et $\tilde{h} = \tilde{f}$, ce qui est exclu puisque $f_*(p)$ est sans facteur carré. On conclut avec la proposition précédente.

Dans son article J.M. Arnaudiès applique ce résultat aux sous-groupes résolubles transitifs maximaux afin de déterminer si un polynôme est résoluble par radicaux. Dans son article P. Lefton prend, entre autres, comme exemples le groupe alterné et le groupe cyclique engendré par le cycle $\sigma = (12 \ldots n)$.

Exemple : Prenons A_n le groupe alterné (d'indice 2). L'élément primitif est alors le déterminant de Vandermonde : $f(\underline{x}) = \prod_{i<j}(x_i - x_j)$ dont le conjugué sous S_n est $-f$. D'où $f_*(p) = x^2 - (\tilde{f})^2 = x^2 - disc(p)$. On retrouve bien que $Gal_p(k)$ est un sous-groupe de A_n si son discriminant est un carré dans k.

5 Algorithme général de résolution et implantation

5.1 L'algorithme `DIRECT`

Tout d'abord nous allons voir une proposition permettant de déduire de l'action du produit de groupes symétriques S_C sur f celle de S_D sur f vue dans R_D.
En fait nous l'énonçons dans le cas où $s = 1$ (i.e. $C = (c_1) = (m)$ et $D = (d_1) = (n)$). Le cas $s > 1$ en est qu'une simple généralisation.

Propriété 5.1 *Soit $f \in k[x_1, \ldots, x_m]$ telle que*

$$p_r(O_{S_m}(f)) = \sum_{I \in E} c_I M_I(x_1, \ldots, x_m).$$

C'est à dire que cette fonction puissance, symétrique en $x_1, \ldots x_m$ (comme nous l'avons déjà constaté), est connue sur la base des formes monomiales de $k[x_1, \ldots, x_m]^{S_m}$. Si $m \leq n$ et F est le prolongement de f à $k[x_1, \ldots, x_n]$ alors :

$$p_r(O_{S_n}(F)) = \sum_{I \in E} \binom{n - lg(I)}{m - lg(I)} c_I M_I(x_1, \ldots, x_n),$$

où $lg(I)$ est la longueur de la partition I.

En particulier si f est symétrique (où multisymétrique si $s > 1$), alors $p_r(O_{S_m}(f)) = f^r$ peut être obtenue directement sur la base des formes monomiales (voir [V1]).

L'algorithme est donné pour $s = 1$, mais il se généralise facilement au cas $s > 1$. On se donne p un polynôme de degré n en une variable et f un polynôme de m variables, avec $m \leq n$. On suppose que le cardinal de $O_{S_n}(f)$ est c. Les quatres étapes de l'algorithme sont les suivantes :

1- Calculer les fonctions puissance de $O_{S_m}(f)$ jusqu'à l'ordre c, sur la base des formes monomiales de $k[x_1, \ldots, x_m]^{S_m}$. Ceci est possible d'après le Corollaire 2.2.

2- En déduire à l'aide de la Proposition 5.1 celles de $O_{S_n}(f)$ jusqu'à l'ordre c, sur la base des formes monomiales de $k[x_1, \ldots, x_n]^{S_n}$.

3- Décomposer chacune de ces fonctions puissance en les fonctions symétriques élémentaires des racines de p. Ceci peut se faire directement où par l'intermédiaire d'une autre base de l'algèbre des polynômes symétriques comme celle des fonctions puissance obtenues à partir des fonctions symétriques élémentaires à partir des relations de Girard-Newton appliquées aux racines de p.

4- L'étape 3 ayant permis l'obtention des fonctions puissance des racines de $f_*(p)$, on en déduit les fonctions symétriques élémentaires (et donc les coefficients) avec les relations de Girard-Newton.

Remarque : Le choix de la recherche des fonctions puissance n'est bien entendu pas limitatif car on peut les obtenir à partir de toute autre base de l'anneau des fonctions symétriques. Mais en général ce sont elles qui offrent le plus de facilités combinatoires.

5.2 Programmation sous MACSYMA

Nous disposons sous `MACSYMA` d'un module de manipulations de fonctions symétriques, nommé `SYM`, permettant de réaliser l'algorithme précédent [voir V3]. Nous allons reprendre l'algorithme précédent étape par étape en mettant en gras les fonctions de `SYM`.

5.2.1 Etape 1

Cette étape peut se réaliser de différentes manières. Le premier cas est celui où le cas particulier étudié permet d'obtenir facilement une formule (ex. pour la résolvante de Galois). Si on ne dispose pas d'une telle formule, la méthode générale consiste à calculer l'orbite de f sous l'action de S_m avec la fonction **orbit** (où **multi_orbit** pour le cas $s > 1$). On a vu au paragraphe 4.3 (*Remarque* p.8) que l'invariance suivant un groupe de la fonction f peut permettre d'obtenir plus rapidement cette orbite. Une fois cette orbite obtenue, pour en calculer les fonctions puissance sur la base des formes monomiales, on utilise la fonction **pui_direct** de `SYM`.

5.2.2 Etape 2

Elle se réalise sans problème avec des commandes `MACSYMA`.

5.2.3 Etape 3

Si on désire décomposer directement en les fonctions symétriques élémentaires du polynôme p, on utilise la fonction **elem** sinon on calcule les fonctions puissance des racines de p avec la fonction **ele2pui** puis on décompose avec la fonction **pui**.

5.2.4 Etape 4

On utilise la fonction **pui2ele** de `SYM`.

5.3 Calcul de la résolvante de Galois

On utilise la proposition 4.1 afin de réaliser la première étape. Tout d'abord on calcule les fonctions puissance sans spécialiser les valeurs entières des $t_1, t_2, \ldots, t_n$. On utilise alors les fonctions **ltreillis**, **card_stab** et **multinomial** de `SYM`. La fonction **ltreillis** ramène la liste des partitions de poids n et de longueurs bornées. Comme pour un même degré cette étape réalisée dans le programme ci-dessous par la fonction `newton_gen` est la même pour tous les polynômes, on sauve ce résultat dans un fichier si cela n'a pas encore été fait, sinon on le charge. Puis on spécialise les t_i en des entiers. De même, cette étape pouvant être commune à plusieurs (en réalité presque tous) polynômes d'un même degré, on sauve ce calcul dans un fichier.

Remarque : pour tester si f est une fonction $n!$-valeurs, la pratique a montré qu'il est plus aisé de calculer d'abord la résolvante $f_*(p)$ puis de tester si elle sans facteur carré, que de calculer son discriminant avec le déterminant de la matrice $(p_{i+j-2}(f_*(p)))_{1\leq i,j\leq n!}$. Voici donc le programme en `MACSYMA` :

```
/* VALEUR 3 POUR pui, CAR LES POLYNOMES SONT PARTITIONNES*/
pui : 3$
/* treillis EST UNE LISTE DE PARTITIONS DE POIDS r A EXACTEMENT n ELEMENTS
   ON ADJOINT A CHAQUE PARTITION LE PRODUIT DU CARDINAL DE SON STABILISATEUR,
   CALCULE AVEC LA FONCTION card_stab de SYM, SOUS L'ACTION DU GROUPE SYMETRIQUE
   S_n, AVEC UN COEFFICIENT MULTINOMIAL, CALCULE AVEC LA FONCTION
   multinomial de SYM */
newton_gen(treillis,r,n):=
      maplist(lambda([part],
                     cons(card_stab(part,"=")* multinomial(r,part),
                          part)),
              treillis)$
/* EVALUATION D'UNE FORME MONOMIALE A PARTIR DE SA PARTITION ASSOCIEE part
   CETTE PARTITION COMPORTANT EXACTEMENT n ELEMENTS.
   UTILISATION DE LA FONCTION permut de SYM */
ev_sym(part,v,n):=block([r],
   apply("+", maplist(lambda([u],
                             (r:1,
                              for i:1 thru n do
                                  if not(v[i]=0 and u[i]=0)
                                     then r:r*v[i]^u[i],
                              r)),
                      permut(part))))$
/* LA FONCTION Galois PREND COMME ARGUMENTS :
     - n LE DEGRE DU POLYNOME
     - puissances_init LES FONCTIONS PUISSANCE DE SES RACINES (DE 0 A n!)
                       OBTENUES AVEC LA FONCTION ele2pui DE SYM
     - v LA VALEUR DES ENTIERS POUR  f=somme(v_i*x_i) DE 1 A n
     - etape VALANT 1, 2 OU 3 SELON CE QUI A DEJA ETE SAUVEGARDE */
Galois(n,puissances_init,v,etape):=
      block([d,newton,g,p,coeff_resol,newton_base,nom ],
/* DEGRE DE LA RESOLVANTE */
   d:factorial(n),
/* VALEURS COMMUNES A TOUS LES POLYNOMES DE DEGRES n
     UTILISATION DE LA FONCTION ltreillis de SYM : PARTITION DE POIDS r ET
     DE LONGUEUR INFERIEURE A n */
  nom : concat('newton_base,n,".",'l),
```

```
  if etape = 1 then
       (newton_base : makelist(newton_gen(ltreillis(r,n),r,n),r,1,d),
        apply(save,[[nom],'newton_base]))
     else if etape = 2 then load(nom),
/* CALCUL DES FONCTIONS PUISSANCE, DE 1 A n!, DE l'ORBITE DE LA FONCTION
   somme_{i=1}^n t_i*x_i , OU LES x_i SONT ENCORE GENERIQUES */
  nom : concat('newton_spe,n,".",'l),
  if not(etape=3) then
     (for i:1 thru d do
          (p[i]:maplist(lambda([mon],
                              cons(ev_sym(rest(mon),v,n)*first(mon),
                                   rest(mon)) ),
                       first(newton_base)),
            newton_base : rest(newton_base)),
            p[0]:d,
            apply(save,[[nom],'p]))
    else load(nom),
/* PUIS DECOMPOSITION EN LES FONCTIONS PUISSANCE DU POLYNOME DONNE
   AVEC LA FONCTION pui de SYM */
   p[0]:d,
   newton: cons(d,
               makelist((kill(p[i-1]),pui(puissances_init,p[i],[])),
                      i,1,d)),
   kill(p[d]),
/* CALCUL DE LA RESOLVANTE  A PARTIR DES FONCTIONS PUISSANCE DE SES RACINES */
   coeff_resol:pui2ele(d,newton),
   kill(newton),
   g: x**d,
   for i:1 thru d do
          (coeff_resol:rest(coeff_resol),
            g:g+(-1)**i*x**(d-i)*first(coeff_resol)),
   sqfr(g))$
```

Pour le cas où le polynôme est de degré 4, on calcule le degré de la résolvante de Galois de sa résolvante cubique avec la fonction `resolcub` qui prend comme argument la liste des fonctions symétriques élémentaires des racines du polynôme considéré :

```
resolcub(fse):= block([fse_resolcubique,fp],
/* FONCTIONS SYMETRIQUES ELEMENTAIRES DES RACINES DE SA RESOLVANTE CUBIQUE */
    fse_resolcubique : [fse[2],fse[1]*fse[3]-4*fse[4],
                        fse[4]*(fse[1]^2-4*fse[2])+fse[3]^2],
/* FONCTIONS PUISSANCE */
     fp : ele2pui(6,cons(3,fse_resolcubique)),
     Galois(3,fp,[1,2,3],1))$
```

6 Exécutions

Les illustrations choisies ci-dessous sont des exemples se trouvant dans [Dickson]. Soient $p(x) = x^3 + x^2 + x + 1$ et $f(x_1, x_2) = x_2 - x_1$ (voir p. 162). On calcule en `(c3)` les fonctions puissance des racines de ce polynôme et en `(c4)` la forme "sqfree" de $f_*(p)$. Cette résolvante étant sans facteur carré, f est une fonction $n!$-valeurs est le groupe de Galois est d'ordre 2 d'après `(d5)`. Ensuite on choisit le polynôme $q(x) = x^4 + x^3 + x^2 + x + 1$ (voir p.170). Le groupe de Galois de sa résolvante cubique est d'ordre 2 d'après `(d6)` et `(d7)`. On calcule alors en `(d8)` et `(d9)` l'ordre du groupe de Galois de q. Et comme d'après `(d8)` et `(d9)` l'ordre de $Gal_{\mathbf{Q}}(q)$ est 4, c'est le groupe cyclique d'ordre 4. En `(c10)` et `(c11)` on trouve les temps d'exécutions lorsque l'on utilise des résultats sauvegardés dans des fichiers.

```
(c3) fp:ele2pui(3,[3,-1,1,-1]);
Time= 850 msec.
(d3)         [3, - 1, - 1, - 1]
(c4) Galois(3,fp,[-1,1,0],1);
Totaltime= 28616 msec.  GCtime= 6450 msec.
```

```
                     6      4      2
(d4)                x  + 4 x  + 4 x  + 16
(c5) factor(%);
Time= 750 msec.
                       2        2              2
(d5)                 (x  + 4) (x  - 2 x + 2) (x  + 2 x + 2)
(c6) resolcub([-1,1,-1,1]);
Totaltime= 21683 msec.  GCtime= 3250 msec.
                      6       5       4        2
(d6)                 x  - 12 x  + 40 x  - 140 x  + 48 x + 19
(c7) factor(%);
Totaltime= 4300 msec.  GCtime= 3200 msec.
                     2               2              2
(d7)               (x  - 9 x + 19) (x  - 4 x - 1) (x  + x - 1)
(c8) Galois(4,ele2pui(4,[4,-1,1,-1,1]),[1,2,-1,-2],1);
Totaltime= 5385650 msec.  GCtime= 1152850 msec.
      24       22        20          18          16           14             12
(d8) x   + 50 x   + 375 x   - 14000 x   - 73625 x   + 4031250 x   - 2410625 x
               10               8                6                 4
 - 689531250 x   + 1897181250 x  + 33836281250 x  + 13666112500 x
                2
 + 226676250000 x  + 45422265625
(c9) factor(%);
Totaltime= 105400 msec.  GCtime= 14066 msec.
        4       2                 4       2
(d9) (x  - 20 x  - 15 x + 155) (x  - 20 x  + 15 x + 155)
   4       2                 4       2                  4       2
 (x  + 20 x  - 45 x + 55) (x  + 20 x  + 45 x + 55) (x  + 25 x  + 5)
   4       2
 (x  + 25 x  + 125)
(c10) Galois(4,ele2pui(4,[4,-1,1,-1,1]),[1,2,-1,-2],2)$
Totaltime= 5230583 msec.  GCtime= 1106550 msec.
(c11) Galois(4,ele2pui(4,[4,-1,1,-1,1]),[1,2,-1,-2],3)$
Totaltime= 238450 msec.  GCtime= 49050 msec.
```

References

[Arnaudiès] 1976, Jean-Marie Arnaudiès, *Sur la résolution explicite des équations de degré 5, quand elles sont résolubles par radicaux*, Bull. Sc. math., 2^e série, 100, 241-254.

[Artin], E., 1959, *Galois Theory*, Notre Dame Mathematical Lectures No. 2, Notre Dame, IN: Notre Dame University Press.

[Bastida], Julio R., 1984 , *Field Extensions and Galois Theory*, Encyclopedia of mathematics and its applications, Vol. 22, Addison-Wesley.

[Dedekind], R., *Über die Theorie der ganzen algebraishen Zahlen*, in [Dirichlet, supp.XI].

[Dickson] Leonard E., *Algebraic theories*, Dover publications, INC., New-York.

[Dirichlet], P.G.L, 1984, *Vorlesungen über Zahlentheorie*, Braunschweig : Vieweg.

[Girard], 1629, Invention Nouvelle en Algèbre, Amsterdam

[G,L,V] 1988, M. Giusti, D. Lazard, A. Valibouze, *Algebraic transformation of polynomial equations, symmetric polynomials and elimination*, Proceedings of ISSAC-88 (Roma, Italy), Springer-Verlag

[Lefton], Phyllis, Octobre 1977, *Galois resolvents of permutation groups*, Am. Math. Monthly, 642-644

[Loos], R., *Computing in Algebraic Extensions*, Computer Algebra Symbolic and Algebraic Computation, Springer-Verlag, 173-187

[Macdonald], I.G., 1979, *Symmetric functions and Hall polynomials*, Clarendon Press, Oxford

[V1] 1897, A. Valibouze, *Fonctions symétriques et changements de bases*, Proceedings of the European Conference on Computer Algebra EUROCAL '87 (Leipzig, RDA), Springer-Verlag

[V3] 1989, A. Valibouze, *Symbolic computation with symmetric polynomials, an extension to Macsyma*, Proceedings of the conference Computers and Mathematics (MIT, Cambridge, Mass), Springer-Verlag