Computing the Lagrange resolvent by effectiveness of Galois Theorem

Ines Abdeljaoued, Faiçal Bouazizi, Annick Valibouze

Computing the Lagrange resolvent by effectiveness of Galois Theorem

I. Abdeljaoued-Tej

LIM-EPT and ESSAI, Université 7 novembre à Carthage, 6 rue des métiers, La Charguia 2, 2036 Ariana, Tunisia

F. Bouazizi

Faculté des Sciences de Sfax, 3000 Sfax, Tunisia & LIP6, Université Pierre et Marie Curie, 4, place Jussieu, F-75252 Paris Cedex 05, France

A. Valibouze

LIP6, UPMC-Paris 6, 4, place Jussieu, 75252 Paris Cedex 05, France. +33 1 44 27 88 51

Abstract

In this article, we introduce a new method to calculate Lagrange resolvent. This technique is based on Lagrange's algorithm and it enables to calculate algebraically the resolvent. This algorithm is based on the fundamental theorem of symmetric functions:we generalize the effectivity of this theorem to any surgroup of the Galois's group of the polynomial.

Key words: Lagrange resolvent, minimal polynomial, Galois group, galoisian ideal, triangular ideal, 12F10 12Y05 11Y40

Introduction

The fundamental theorem of symmetric functions has various effective forms (see, for example, [10],[9] and [14]). The computer algebraic system `Maxima` has an important library on the subject (see `Symmetries` in [12]). Cauchy's method enables to reduct a

⋆ This research was partly supported by the Galois project

Email addresses: `i.tej@gnet.tn` (I. Abdeljaoued-Tej), `faical.bouazizi@lip6.fr` (F. Bouazizi), `annick.valibouze@upmc.fr` (A. Valibouze).

URL: `www-spiral.lip6.fr/~avb/` (A. Valibouze).

Preprint submitted to Elsevier

symmetric polynomial with respect to the ideal of symmetric relations $\mathfrak{S}$ generated by the triangular set of Cauchy moduli ([4]). The Galois theorem is not formulated in a constructive form but it generalizes the fundamental theorem of symmetric functions. It is stated not rigorously as follows: *Any polynomial expression on a field k in the roots of a univariate polynomial f with coefficients in k belongs to k if and only if it is invariant by the Galois group of f on k.* Its effective calculation is the stake of the effective Galois theory: let be a polynomial f in $k[x]$, calculate its Galois group G on k as well as the ideal $\mathfrak{M}$ of its relations (this maximum ideal $\mathfrak{M}$ contains the ideal of the symmetric relations). In order to have an effective calculation of $\mathfrak{M}$, it is necessary to calculate simultaneously the Galois group and thus resolvents ([15]). The resolvent offers a double advantage. It excludes groups and provides primitive elements of galoisian ideals, these intermediate ideals between $\mathfrak{S}$ and $\mathfrak{M}$. The resolvent is thus a fundamental tool of Galois theory.

When the resolvent is absolute, its coefficients are symmetric in the roots of f and, as a result, the fundamental theorem of symmetric functions can be applied. As the direct calculation of the coefficients being too expensive, Lagrange proposed two algorithms for its calculation. The first uses the technique of elimination, i.e. the *resultant* without naming it since it does not exist yet as a mathematical object ([7]). The second fact uses the Newton's functions of the resolvent's roots ([8], page 237). The `resolvent` function of `Maxima` implements this algorithm (see `resolvent:general` in library `Symmetries`). In [2], an algorithm based on triangular ideals and using resultants was worked out for resolvents (absolute ones or not).

We present, in section 3, a new algorithm devoted to the algebraic method which uses the Newton's functions of the resolvent's roots. This algebraic algorithm requires a generalization of the effectivity of the fundamental theorem of the symmetric function. We were inspired by Cauchy's method in order to "evaluate" the multivariate polynomials in the roots of f by using a galoisian ideal (see section 2). In particular, when the galoisian ideal is $\mathfrak{M}$, this evaluation produces the effectiveness of Galois theorem. We describe our algorithm in the free mathematics software system `SAGEmath` (see section 3.2). In order to measure the effectiveness of our algorithm (see section 5), we do not take account of optimizations of section 6. In section 6, we will note that the algorithm is naturally parallelizable and that we can apply to him a method of calculation of products in a ring quotiented by a triangular ideal ([3]). It presents other interests like detecting linear factors over $k[x]$ of resolvent while accelerating its calculation.

1. Reminder

In all this article, f is a univariate polynomial of degree n, with coefficients in a perfect field k, $\underline{\alpha} = (\alpha_1, ..., \alpha_n)$ denotes an n-tuple formed by n roots of f supposed distincts (where $n > 0$). The extension field $k(\underline{\alpha})$ of k is the decomposition field of f; recall that we have $k(\underline{\alpha}) = k[\underline{\alpha}]$. Let $x_1, \ldots, x_n$ be algebraically independent variables; we consider that they are ordered by $x_1 < x_2 < \ldots < x_n$; let $k[x_1, \ldots, x_n]$ be the ring of polynomials in these variables and with coefficients in k; $k(x_1, \ldots, x_n)$ is its field of fractions.
We adopt the notations and the results of [15] without citing them explicitely.

1.1. Orbits and group actions

Let L be a subgroup of the symmetric group S_n of degree n and H be a subgroup of L. The symmetric group S_n acts naturally on the field $k(x_1,...,x_n)$ by:

$$\begin{array}{lll} S_n \times k(x_1,...,x_n) \rightarrow k(x_1,...,x_n) & & \\ (\sigma, \quad P) & \mapsto & \sigma.P(x_1,...,x_n) = P(x_{\sigma(1)},...,x_{\sigma(n)}) \end{array}$$

Definition 1. The *orbit* $L.P$ of P under the action of L is defined by:

$$L.P = \{\sigma.P \mid \sigma \in L\}.$$

Definition 2. The *stabilizer* $\mathrm{Stab}_L(P)$ of a polynomial P in $k[x_1,...,x_n]$ with respect to L is defined by:

$$\mathrm{Stab}_L(P) = \{\sigma \in L \mid P = \sigma.P\}$$

and the *stabilizer* of H with respect to L is defined by:

$$\mathrm{Stab}_L(H) = \{\sigma \in L \mid \forall\, r \in H \;\; r = \sigma.r\}.$$

Definition 3. An *invariant of* L (or an L*-invariant*) is a polynomial P in $k[x_1,...,x_n]$ verifying:

$$L.P = \{P\}.$$

It is called an L*-primitif* H*-invariant* if

$$H = \mathrm{Stab}_L(P) = \{\sigma \in L \mid \sigma.P = P\}.$$

When $L = S_n$, the polynomial P is called a *primitif* H*-invariant*.

Examples 1.
- The *Vandermonde determinant* $\delta_n = \prod_{1\leq i<j\leq n}(x_i - x_j)$ is a *primitif* A_n*-invariant* where A_n is the alternating group of degree n.
- The polynomial $x_1x_2 + x_3x_4$ is a S_4*-primitif* $\mathfrak{D}_4$*-invariant* where

$$\mathfrak{D}_4 =< (1,2),(3,4),(1,3)(2,4) >$$

is the dihedral group
- The polynomials $x_1 + 2x_2 + ... + (n-1)x_{n-1}$ and $x_1x_2^2...x_{n-1}^{n-1}$ are *primitif* I_n*-invariants* where I_n is the identity group of degree n.

1.2. Symmetric polynomials

A polynomial s of $k[x_1,...,x_n]$ is called *symmetric* (in $x_1,x_2,...,x_n$) if $s = \sigma.s$ for all permutations $\sigma \in S_n$. Two important bases of the ring $k[x_1,...,x_n]^{S_n}$ of the symmetrical polynomials are pointed out below:

- the *elementary symmetric fonctions* $e_0, e_1, ..., e_n, ...$ in $x_1,...,x_n$, are defined by $e_0 = 1, e_r = 0$ for $r > n$ and for $r \in [\![1,n]\!]$

$$e_r \;=\; \sum_{m\in S_n.(x_1x_2...x_r)} m\;;$$

- the *Newton's functions* $p_0, p_1, ..., p_n, ...$ in $x_1, ..., x_n$ (also called *power functions*), are defined by

$$p_r = \sum_{i=1}^{n} x_i^r \ .$$

The *Girard-Newton formulae* ([6]) constitute a triangular system which makes it possible to pass from a basis to another: for all integers $r > 0$

$$p_r e_0 - p_{r-1} e_1 + \cdots + (-1)^{r-1} p_1 e_{r-1} + (-1)^r r.e_r = 0.$$

Put $a_i = (-1)^i e_i(\underline{\alpha})$ for $i = 1, \ldots, n$. The polynomial f can be written in the form

$$f = x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_n$$

and the Girard-Newton formulae give us:

$$p_r(\underline{\alpha}) + p_{r-1}(\underline{\alpha}) a_1 + \cdots + p_1(\underline{\alpha}) a_{r-1} + r a_r = 0.$$

The *fundamental theorem of symmetric polynomials* say that any symmetric polynomial over k can be expressed as a polynomial over k in elementary symmetric polynomials . Furthermore, by using Girard-Newton formulae, any symmetric polynomial on the roots of a univariate polynomial can be expressed as a polynomial expression over k of the power functions on these roots.

1.3. ***Triangular ideals***

Definition 4. A *triangular* set T is defined by:

$$T = \{f_1(x_1), ..., f_n(x_1, ..., x_n)\}$$

where every f_i is a monic polynomial on x_i and $\deg(f_i, x_i) > 0$. This triangular set T is called *separable* if any f_i verify for all $\beta = (\beta_1, ..., \beta_{i-1}) \in \widehat{k}^{i-1}$ such that

$$f_1(\beta_1) = f_2(\beta_1, \beta_2) \cdots f_{i-1}(\beta_1, ..., \beta_{i-1}) = 0,$$

the univariate polynomial $f_i(\beta_1, ..., \beta_{i-1}, x_i)$ does not admit a multiple root.

Example 1. For $n = 8$, the following triangular set T is separable:

$$\begin{aligned}
T = \{ \ & f_1 = x_1^8 + 9x_1^6 + 23x_1^4 + 14x_1^2 + 1, \\
& f_2 = x_2 + x_1, \\
& f_3 = x_3^3 + (x_1^7 + 8x_1^5 + 16x_1^3 + 3x_1)x_3^2 \\
& \qquad\qquad\qquad + (x_1^6 + 9x_1^4 + 21x_1^2 + 6)x_3 + x_1^7 + 9x_1^5 + 23x_1^3 + 14x_1, \\
& f_4 = x_4^2 + (x_1^7 + 8x_1^5 + 16x_1^3 + 3x_1)(x_4 + x_3) + x_3x_4 + x_3^2 + x_1^6 + 9x_1^4 + 21x_1^2 + 6, \\
& f_5 = x_5 + x_4 + x_3 + x_1^7 + 8x_1^5 + 16x_1^3 + 3x_1, \\
& f_6 = x_6 + x_3, \\
& f_7 = x_7 + x_4, \\
& f_8 = x_8 + x_5 \ \}
\end{aligned}$$

In fact: the polynomials f_2,f_5,f_6,f_7 and f_8 satisfies the condition because they are repectively linear on x_2, x_5, x_6, x_7 and x_8; the polynomial f_1 is irreductible over the perfect field $\mathbb{Q}$ and so separable; the polynomial $f_3(\alpha_1, x)$ is a factor of $f_1(x)$ over $\mathbb{Q}(\alpha_1)$ what involves its separability; finally,

$$f_4(x_1, x_3, x_4) \quad = \quad \frac{1}{x_3 - x_4}(f_3(x_1, x_3) - f_3(x_1, x_4)),$$

thus its separability.

Definition 5. The *Cauchy moduli* of f are polynomials $f_1, ..., f_n$ in $k[x_1, ..., x_n]$ defined inductively as follows:
- $f_1(x_1) = f(x_1)$ and
- for $i = 2, ..., n$:

$$f_i(x_i) = \frac{f_{i-1}(x_1, x_2, \dots, x_{i-2}, x_{i-1}) - f_{i-1}(x_1, x_2, \dots, x_{i-2}, x_i)}{x_{i-1} - x_i}.$$

The Cauchy moduli form a separable triangular set.

Definition 6. An ideal I is said *triangular* if it is generated by a separable triangular set.

Let I be a triangular ideal generated by the following separable triangular set:

$$T = \{f_1(x_1), f_2(x_1, x_2), ..., f_n(x_1, ..., x_n)\}.$$

The set T forms a minimal Gröbner basis for the lexicographic order (recall that $x_1 < x_2 < \cdots < x_n$). Reducing the polynomial P of $K[x_1, ..., x_n]$ by the ideal I consists on realizing successive Euclidean divisions for each polynomial f_i regarded as a polynomial in x_i for $i \in [\![1, n]\!]$. The remainder of this division is a normal form of P in the quotient ring $k[x_1, ..., x_n]/I$. The result of this reduction will be noted P mod I.

Algorithm `ReductionTriangulaire(P,I)`

Input: P $\in k[x_1, ..., x_n]$ and $f_1, ..., f_n$, a triangular basis of I

Output: P mod I

P mod I $\longleftarrow$ P

For $i \in [\![1, n]\!]$ Do

  P mod I $\longleftarrow$ Reste(P mod I, f_i, x_i)

Return P mod I

where `Reste(p, q, x)` is the remainder of euclidean division of `p` by `q` regarded as polynomials in `x`. The ideal of symmetric relations $\mathfrak{S}$ is triangular and is generated by the Cauchy moduli ([11]). Cauchy proposed in [4] an effective form of the fundamental theorem of symmetric functions which we rewrite in the following form:

Theorem 7. *(Cauchy, 1840) Let s be a symmetric polynomial of $k[x_1, \dots, x_n]$. Thus $s(\underline{\alpha})$ is the output of the algorithm* `ReductionTriangulaire`*$(s, \mathfrak{S})$.*

1.4. Ideal of $\underline{\alpha}$-relations and Galois group

Definition 8. A polynomial P of $k[x_1, ..., x_n]$ is called an $\underline{\alpha}$*-relation* if

$$P(\underline{\alpha}) = 0.$$

Definition 9. The ideal $\mathfrak{M}$ of $k[x_1, ..., x_n]$ defined by

$$\mathfrak{M} = \{R \in k[x_1, ..., x_n] \mid R(\underline{\alpha}) = 0\}$$

is called the *ideal of* $\underline{\alpha}$*-relations*.

Definition 10. The *Galois group* $G_{\underline{\alpha}}$ of $\underline{\alpha}$ over k is the stabilizer of $\mathfrak{M}$ in S_n:

$$G_{\underline{\alpha}} = \{\sigma \in S_n \mid (\forall R \in \mathfrak{M})\ \sigma.R \in \mathfrak{M}\}.$$

Theorem 11. *(Galois, 1897) Let $P \in k[x_1, \ldots, x_n]$. We put $\sigma.P(\underline{\alpha}) = P(\underline{\alpha})$ for all $\sigma \in G_{\underline{\alpha}}$ if and only if $P(\underline{\alpha}) \in k$.*

Definition 12. Let L be a subset of S_n. The ideal

$$I_{\underline{\alpha}}^L = \{R \in k[x_1, ..., x_n] \mid (\forall \sigma \in L)\ \sigma.R(\underline{\alpha}) = 0\}$$

is called the *ideal of* $\underline{\alpha}$*-relations invariant by* L. Such an ideal is called a *galoisian ideal of* f *over* k.

Remark 13.
- The ideal of $\underline{\alpha}$-relations $\mathfrak{M}$ is the ideal of $\underline{\alpha}$-relations invariants by $G_{\underline{\alpha}}$ or by I_n, the identity group of S_n.
- The ideal $\mathfrak{S} = I_{\underline{\alpha}}^{S_n}$ is the *ideal of symmetric relations* between the roots of f.

Definition 14. Let I be a galoisian ideal of f (over k). The *injector* of I in $\mathfrak{M}$ is given by:

$$\mathrm{Inj}(I, \mathfrak{M}) = \{\sigma \in S_n \mid (\forall R \in I)\ R(\alpha_{\sigma(1)}, ..., \alpha_{\sigma(n)}) = 0\}.$$

Example 2. $\mathrm{Inj}(\mathfrak{S}, \mathfrak{M}) = S_n$ and $\mathrm{Inj}(\mathfrak{M}, \mathfrak{M}) = G_{\underline{\alpha}}$.

We have the following identities:

$$\mathrm{Card}(V(I)) = \mathrm{Card}(\mathrm{Inj}(I, \mathfrak{M})) = \dim_k k[x_1, \ldots, x_n]/I \tag{1}$$

where $V(I)$ is the algebraic variety of I, the set of its zeros.

When the injector of I in $\mathfrak{M}$ is a group, the ideal I is said *pure*. A galoisian ideal I included in $\mathfrak{M}$ is pure if and only if its injector in itself (we have $\mathrm{Inj}(I, I) = \mathrm{Stab}_{S_n}(I)$) contains the Galois group $G_{\underline{\alpha}}$; which is equivalent to

$$\mathrm{Inj}(I, I) = \mathrm{Inj}(I, \mathfrak{M})\ ;$$

which is, using (1), still equivalent to

$$\mathrm{Card}(\mathrm{Inj}(I, I)) = \dim k[x_1, \ldots, x_n]/I \quad .$$

In [2], the authors show that a pure galoisian ideal is triangular. For example, the ideals $\mathfrak{S}$ and $\mathfrak{M}$ are pure. The galoisian ideals considered are for the majority pure otherwise we obtain this case by permutations of its relations (see [16]). The reduction modulo a pure galoisian ideal consists on n Euclidean divisions with its generators (see section 1.3).

1.5. *Minimal and characteristic polynomials and Resolvents*

We consider $I \subset \mathfrak{M}$, a galoisian ideal with injector L in $\mathfrak{M}$, H a group included in L and P an L-primitif H-invariant.

Let $\widehat{P}$ be the multiplicative endomorphism of $k[x_1, ..., x_n]/I$ induced by P:

$$\begin{array}{rccc} \widehat{P}: & k[x_1,...,x_n]/I & \rightarrow & k[x_1,...,x_n]/I \\ & \Theta & \longmapsto & \Theta.P \end{array}$$

The *characteristic polynomial* of $\widehat{P}$ is an element of $k[x]$ of degree $\mathrm{Card}(L)$ given by

$$\chi_{\widehat{P},I} = \prod_{\sigma \in L} (x - \sigma.P(\underline{\alpha})). \tag{2}$$

This results is easily obtained from Identity (1) and from the Stickelberger's Theorem expressing the characteristic polynomial of such multiplicative endomorphisms (for ideals of finished variety not necessarily radical). We can also affirm that $\chi_{\widehat{P},\mathfrak{S}} \in k[x]$ by the Galois Theorem since the injector L contains the Galois group.

When k is a perfect field, the *minimal polynomial* of the endomorphism $\widehat{P}$ is the square free factor over k of the characteristic polynomial:

$$\mathrm{Min}_{\widehat{P},I} = \prod_{\psi \in \{Q(\underline{\alpha}) \mid Q \in L.P\}} (x - \psi).$$

The *resolvent L-relative* of $\underline{\alpha}$ by P is, by definition, the polynomial

$$R_{P,I} = \prod_{Q \in L.P} (x - Q(\underline{\alpha})). \tag{3}$$

It belongs to $k[x]$ since its coefficients are invariant by L and since that L contains the Galois group $G_{\underline{\alpha}}$. When the resolvent is square free, it is identified with the minimal polynomial. We have the identity:

$$\chi_{\widehat{P},I} = R_{P,I}^{\mathrm{Card}(H)} \quad . \tag{4}$$

When $L = S_n$, the resolvent does not depend on the order $\alpha_1, ..., \alpha_n$ of the roots of f. It is said *absolute* and called *resolvent of f by P*.

Example 3. Let put $n = 3$, $f = (x - x_1)(x - x_2)(x - x_3)$ and $P = x_1x_2^2$. We have

$$R_{P,\mathfrak{S}} = (x - x_1x_2^2)(x - x_1x_3^2)(x - x_2x_1^2)(x - x_2x_3^2)(x - x_3x_1^2)(x - x_3x_2^2) \quad .$$

1.6. General assumptions

Let us assume that I is a galoisian ideal knowed by a reduced Gröbner basis (this basis is used for performing the reduction modulo I). We denote by L the injector of I in a maximal ideal $\mathfrak{M}$ containing I. We fix a subgroup H of L and a an L-primitif H-invariant P in $k[x_1, ..., x_n]$.

2. Effectiveness of Galois Theorem

We seek to evaluate polynomials which are invariant by any injector in $\mathfrak{M}$ of a galoisian ideal.
As mentioned in Introduction, when the injector L is S_n, the considered polynomials are symmetric and there are many methods to evaluate them on the roots of f. They are effective forms of fundamental theorem of symmetric polynomials (voir section 1.2). When $L = G_{\underline{\alpha}}$ it is about the effective form of Theoroem 11 of Galois. We gather here these two theorems into an effective one (i.e. for any injector L).

Theorem 15. *Let I be a galoisian ideal of f included in $\mathfrak{M}$, the ideal of $\underline{\alpha}$-relations, and L the injector of I in $\mathfrak{M}$. Let $R \in k[x_1, ..., x_n]$ such that $\sigma.R = R$ for all $\sigma \in L$. Then $R(\underline{\alpha})$ belongs to the field k of coefficients of f and*

$$R - R(\underline{\alpha}) \in I.$$

In other words, $R(\underline{\alpha})$ is calculated as the reduction of R modulo I. *Proof*: Since L is the injector of I in $\mathfrak{M}$, the Galois group $G_{\underline{\alpha}}$ is included in L. Let put $\lambda = R(\underline{\alpha})$. As the polynomial R is invariant by L, it is also invariant by $G_{\underline{\alpha}}$. Therefore, by Theorem 11, $\lambda \in k$ and $R - \lambda \in k[x_1, \ldots, x_n]$. For any $\sigma \in L$, we have

$$\sigma.(R - \lambda)(\underline{\alpha}) = \sigma.R(\underline{\alpha}) - \lambda = R(\underline{\alpha}) - \lambda = 0 \quad ,$$

because R is L-invariant. So, by definition of galoisian ideals, we obtain the result $R - R(\underline{\alpha}) \in I$.

3. Algebraic algorithm for the resolvent

3.1. The principle of calculation

From Theorem 15, we can bring out an algorithm to find the resolvent $R_{P,I}$ inspired on Lagrange algorithm ([8], page 237) restricted to the absolute resolvent (i.e. $L = S_n$ and $I = \mathfrak{S}$). Its method is to calculate the power functions of the roots of the absolute resolvent based on the effectiveness of the fundamental theorem of symmetric polynomials; then he deduce the coefficients of the resolvent with Girard-Newton formulae. We present a simular algorithm for any L- relative resolvents based on Theorem 15.

First of all, the following well known lemma provides us a way to calculate, without duplication, the orbit $L.P$ of P under the action of L:

Lemma 16. *Let d be the index of H in L. Then the orbit $L.P$ consists of d distinct polynomials $\tau.P$ where τ runs through a left coset of L mod H.*

We fix $i \in [\![1, d]\!]$. According to Fomula (3) defining the resolvent $R_{P,I}$, the ith power function $p_i(R_{P,I})$ of its roots is given by:

$$p_i(R_{P,I}) \;=\; \sum_{Q \,\in\, L.P} Q^i(\underline{\alpha}) \quad ;$$

this is the evaluation in $\underline{\alpha}$ of the polynomial

$$p_i(L.P) \;=\; \sum_{Q \,\in\, L.P} Q^i \quad .$$

Let check with the following lemma that $p_i(L.P)$ is an L-invariant:

Lemma 17. *Let $L.P = \{P_1, \ldots, P_d\}$ and s be a symmetric polynomial in $k[x_1, ..., x_d]$. Then, the polynomial $s(P_1, \ldots, P_d)$ is an L-invariant.* *Proof*: For any $\sigma \in L$

$$\sigma.s(P_1, \ldots, P_d) = s(\sigma.P_1, \ldots, \sigma.P_d) = s(P_{\tau(1)}, \ldots, P_{\tau(d)})$$

where $\tau \in S_d$ since $\sigma \in L$ and $\{P_1, \ldots, P_d\}$ is the orbit of P under the action of L. As s is a symmetric polynomial, the lemma is proven.

Theorem 18. *For each $i \in \mathbb{N}$, the value in k of the i-th power function $p_i(R_{P,I})$ of $R_{P,I}$ is given by*

$$p_i(R_{P,I}) \;=\; \sum_{Q \,\in\, L.P} Q^i \mod I.$$

Proof: In our case, the polynomial s of Lemma 17 is the ith power sum p_i. Then $p_i(L.P) = \sum_{Q \,\in\, L.P} Q^i$ is L-invariant. By Theorem 15, its evaluation $p_i(R_{P,I})$ on the roots of f is given by $p_i(R_{P,I}) \;=\; \sum_{Q \,\in\, L.P} Q^i \mod I$.

Since it is about finding the values in k of $p_1(R_{P,I}), \ldots, p_d(R_{P,I})$ where d, the index of H in L, is the degree of the resolvent, our algorithm avoids to develop each polynomial $\sum_{Q \,\in\, L.P} Q^i$ in order to reduce it modulo I. We explain below the process selected.

We fix $\overline{R} = R \mod I$, for any $R \in k[x_1, \ldots, x_n]$, and $p_i = p_i(R_{P,I})$. We calculate $\overline{Q}$, $Q \in L.P$, we keep them in a list `lp` and we build `lpp=(1,...,1)` of length d. At the ith step, $1 \leq i \leq d$, we suppose that `lpp` contains $\overline{Q^{i-1}}$, $Q \in L.P$, of the previous step, and we keep the list `lp` of the first step. The power function p_i is computed as follows:

(a) p_i:=0

(b) Browse lists `lp` and `lpp` simultaneously in order to replace every polynomial

$$u = \overline{Q^{i-1}}$$

of `lpp` by $\overline{u * \overline{Q}}$ where $\overline{Q}$ is the element extracted from `lp`; this new element is $\overline{Q^i}$.

(c) For any (reduced) polynomial u of `lpp` Do $p_i := \overline{p_i + u}$.

3.2. The algorithm ABV

All functions are described in the `SAGEmath`'s language.
For our algorithm called `ABV` which compute relative resolvents we need tree additional functions: `somme_mod`, `Orbite` and `pui2polynome` which are described in the follows of the function `ABV`.

Function ABV
Input:
- n the degree of the polynomial f on the variable x
- I a galoisian ideal given with its reduced Gröbner basis
- L the injector of I in a maximal ideal containing I
- H a subgroup of L
- P an L-primitif H-invariant

Output: $R_{P,I}$, the L-relative resolvent of $\underline{\alpha}$ by P, for any $\underline{\alpha} \in V(\mathtt{I})$.

```
def ABV(P,L,H,I,n,x):
     d=gap.Index(L,H)
     lp=Orbite(P,L,H,n)
     lp=[s.mod(I) for s in lp]
     lpp=[1 for i in range(d)]
     pui=[d]
     for i in range(d):
         for j in range(d):
             lpp[j]=(lp[j]*lpp[j]).mod(I)
         pui= pui + [somme_mod(lpp,I)]
     Resolvante=pui2polynome(d,[s for s in pui],x)
     return Resolvante
```

where:
- The function somme_mod(lpp,I) returns the sum of the reduced elements of the list lpp modulo l.
- The function Orbite(P,L,H,n) calculates the orbit of P under the action of L:

```
def Orbite(P,L,H,n)
    from sage.groups.perm_gps.permgroup import from_gap_list
    Sn=SymmetricGroup(n)
    rc= gap.RightCosets(L,H)
    rc=gap.List(rc,'i->Representative(i)')
    LTransv=[s for s in rc]
    return [P * si for si in from_gap_list(Sn,"%s" % LTransv)]
```

- The function pui2polynome(p,x) calculates a univariate polynomial of degree d in x from the $d+1$ power functions $p_0 = d, p_1, \ldots, p_d$ of its roots; the variable p is the list of this power functions:

```
 def pui2polynome(p,x):
   a=[p[0]]
   pol=x^a
   for i in range(1,d+1):
       ai=p[i]+sum(p[j]*a[i-j] for j in range(1,i))
       a=a+[-1/i*ai]
   return  pol=pol+a[i]*x^(d-i)
```

4. Example

Take for example the polynomial $f = x^6 + 2$. Its galoisian ideal I is defined by

$$\begin{aligned}
I = \langle f_1 &= 24x_6 + x_3^3x_2^3x_1 + 8x_3^3x_2^2x_1^2 + 6x_3^3x_2x_1^3 + 5x_3^3x_1^4 + 8x_3^2x_2^3x_1^2 + 4x_3^2x_2^2x_1^3 \\
&\quad +8x_3^2x_2x_1^4 + 6x_3x_2^3x_1^3 + 8x_3x_2^2x_1^4 - 4x_3x_2x_1^5 + 12x_3 + 5x_2^3x_1^4 + 12x_2 + 14x_1, \\
f_2 &= 24x_5 - 5x_3^3x_2^4 - 7x_3^3x_2^3x_1 - 16x_3^3x_2^2x_1^2 - 7x_3^3x_2x_1^3 - 5x_3^3x_1^4 - 8x_3^2x_2^4x_1 \\
&\quad -12x_3^2x_2^3x_1^2 - 12x_3^2x_2^2x_1^3 - 8x_3^2x_2x_1^4 - 12x_3x_2^4x_1^2 - 16x_3x_2^3x_1^3 - 12x_3x_2^2x_1^4 \\
&\quad +8x_3 - 5x_2^4x_1^3 - 5x_2^3x_1^4 - 2x_2 - 2x_1, \\
f_3 &= 24x_4 + 5x_3^3x_2^4 + 6x_3^3x_2^3x_1 + 8x_3^3x_2^2x_1^2 + x_3^3x_2x_1^3 + 8x_3^2x_2^4x_1 + 4x_3^2x_2^3x_1^2 + \\
&\quad 8x_3^2x_2^2x_1^3 + 12x_3x_2^4x_1^2 + 10x_3x_2^3x_1^3 + 4x_3x_2^2x_1^4 + 4x_3x_2x_1^5 + 4x_3 + \\
&\quad 5x_2^4x_1^3 + 14x_2 + 12x_1, \\
f_4 &= x_3^4 + x_3^3x_2 + x_3^3x_1 + x_3^2x_2^2 + x_3^2x_2x_1 + x_3^2x_1^2 \\
&\quad +x_3x_2^3 + x_3x_2^2x_1 + x_3x_2x_1^2 + x_3x_1^3 + x_2^4 + x_2^3x_1 + x_2^2x_1^2 + x_2x_1^3 + x_1^4, \\
f_5 &= x_2^5 + x_2^4x_1 + x_2^3x_1^2 + x_2^2x_1^3 + x_2x_1^4 + x_1^5, \\
f_6 &= x_1^6 + 2\rangle.
\end{aligned}$$

The group $L = \langle(1,3)(2,4),(1,3,4)(2,5,6),(2,3)(4,5),(3,5)(4,6),(3,4,5,6)\rangle$ of order 128 is the injector of I. Let

$$H = \langle(1,2)(3,4)(5,6),(1,3,5)(2,4,6),(3,5)(4,6)\rangle,$$

be a subgroup of L of index 10 in L.
The package `PrimitiveInvariant` of `GAP` (see [1]) calculates the following L-primitif H-invariant

$$P = x_3x_6 + x_1x_6 + x_4x_5 + x_2x_5 + x_1x_4 + x_2x_3.$$

Let $L.P = \{Q_1, ..., Q_{10}\}$ be the orbit of P under the action of L computed with `Orbite`$(P, L, H, 6)$ where 6 is the degree of f.
The first and second power functions of the roots of the resolvent are computed as follows:

$$p_1(R_{P,I}) = \overline{\overline{\overline{\overline{\overline{Q}_1 + \overline{Q}_2} + \overline{Q}_3} + \cdots + \overline{Q}_9} + \overline{Q}_{10}} = 0 \quad \text{and}$$

$$p_2(R_{P,I}) = \overline{\overline{\overline{\overline{\overline{Q}_1^2 + \overline{Q}_2^2} + \overline{Q}_3^2} + \cdots + \overline{Q}_9^2} + \overline{Q}_{10}^2} = 0.$$

By the same way:

$$\begin{array}{llll}
p_3(R_{P,I}) = -6, & p_4(R_{P,I}) = 0, & p_5(R_{P,I}) = 0, & p_6(R_{P,I}) = 36, \\
p_7(R_{P,I}) = 0, & p_8(R_{P,I}) = 0, & p_9(R_{P,I}) = -24, & p_{10}(R_{P,I}) = 0.
\end{array}$$

We save these ten values in the variable `pui` and we deduce the resolvent after executing `pui2polynome(pui,10)`. The function `ABV`$(P, L, H, I, 6, x)$ returns

$$R_{P,I} = x^{10} + 2x^7 - 4x^4 - 8x.$$

5. Time and comparisons with other methods

We will compare our algorithm `ABV` with two others algebraic methods. The first algorithm, which we call `Algo2`, is described in [2] and is based on resultants. It computes the characterisctic polynomial (i.e. a power of the resolvent). Its implementation is available in `Maxima` with version 2 of the library `Symmetries` (not yet distributed). The second algorithm, called `Algo3`, computes on `Maple` the matrix of the endomorphism $\widehat{P}$ with the function `MultiplicationMatrix` then $\mathrm{Min}_{\widehat{P},I}$ with the function `MinimalPolynomial`.

The following table shows the CPU execution times in seconds with $n = \deg(f)$, $c = \mathrm{Card}(L)$, $d = \deg(R_{P,I})$, D is the total degree of the invariant P, N is the monomial's number of P and r is its arity. The polynomials belongs to $\mathbb{Q}[x]$.

n	c	(D,N,r)	d	`ABV`	`Algo2`	`Algo3`
4	4!	$(6,12,4)$	2	0.18	0.6	2.96
5	5!	$(4,25,5)$	12	2.63	9.04	8.14
5	5!	$(4,20,5)$	24	15.94	19.58	766.27
5	5!	$(3,6,5)$	60	88.09	96.60	2361.34
6	6!	$(6,30,6)$	6	2.52	170.1	318
6	128	$(2,6,6)$	10	7.97	11.43	18.50
6	6!	$(2,12,6)$	15	2.60	3.57	860.92
6	6!	$(7,45,6)$	20	4.22	12079.5	571.97
6	6!	$(3,18,6)$	30	120	876.22	4212.70
6	6!	$(2,8,6)$	45	118.15	214.05	1577.51
8	128	$(6,32,8)$	2	8.15	11.92	14.92
8	1152	$(2,8,8)$	9	17.56	337.42	994.84
9	9!	$(1,8,9)$	9	5.16	67.45	867.84

Remark 19. Algorithm `ABV` is based on reductions modulo I, `Algo2` computes a characteristic polynomial of degree the dimension over k of $R = k[x_1, \ldots, x_n]/I$ and `Algo3` produces a square matrix of dimension $\dim_k(R)^2$. Then, in the preceeding table, the value of the order c of the injector L of the ideal I is essential because this value also represents the dimension of R over k.

Comments

We note that `Algo3` is slower than function `ABV` and `Algo2`; it tells us nothing about the multiplicities of roots of the resolvent when it is not separable. To determine these multiplicities, we must find the characteristic polynomial in a longer time than that required by the minimal polynomial, and then calculate a $\frac{c}{d}$th root (see Identity (4)). In terms of efficiency, this method offers no interest.

We also note that `Algo2` is often slower than function `ABV`. Lagrange already noticed

the same thing in his memory [8] by writing on page 240:
"... mais, comme on ne voit pas de cette manière de quel degré devrait être cette équation finale en x, qu'on pourrait même parvenir à une équation en x d'un degré plus haut qu'elle ne devrait être, ce qui est l'inconvénient ordinaire des méthodes d'élimination, nous avons cru devoir montrer comment on peut trouver cette équation a priori et s'assurer du degré précis auquel elle doit monter ".* What Lagrange expressed and which we translate here is that elimination's methods introduce power parasites and, moreover, these power are unknown; while with the power functions, he found directly the resolvent. Today, we know that this power equals to the order of the stabilizer of the invariant P since he calculated the characteristic polynomial $\chi_{\widehat{P},\mathfrak{S}}$ of degree $n!$.

Remark 20. Note that `Algo2` is far more efficient than that proposed by Lagrange. Indeed, the Lagrange's method which is restricted to absolute resolvents (i.e. $L = S_n$) enables to eliminate the variables $x_n, \ldots, x_1$ of the polynomial $x - P$ with respect to polynomials $f(x_n), \ldots, f(x_1)$; he computes polynomial g of degree n^n where $\chi_{\widehat{P},\mathfrak{S}}$ is a factor. Next, with division of g by its"parasite's factors", which can be calculated by eliminations too, he extracts the divisor $\chi_{\widehat{P},\mathfrak{S}}$ of g.

By using `Algo2`, elimination is achieved with the Cauchy moduli (here $L = S_n$) of respective degrees n, $n-1, \ldots, 1$ en $x_n, \ldots, x_1$ and the result is the polynomial $\chi_{\widehat{P},\mathfrak{S}}$ of degree $n!$.

Our function `ABV` does not include the optimizations propozed in the following section. Nevertheless, this comparison demonstrates the efficiency of the function `ABV`.

6. Further Developpements

6.1. Parallelization.

We assume $L.P = \{P_1, \ldots, P_d\}$. The algorithm `ABV` is parallelizable as follows:
Step 1 In parallel, for $j = 0, ..., d$, calculate the list l_j of $\overline{P_j^i}$, $i = 1, \ldots, d$:
(a) $l_j = [P_j \mod I]$
(b) For $i = 1, \ldots, d \quad l_j = l_j + [\, l_j[1] * l_j[j-1] \mod I\,]$.
Step 2 In parallel, for $i = 0, ..., d$, calculate the ith power function p_i by using the function `somme_mod(ll,I)` where `ll` is the list composed by the ith element of every l_j, $j = 1, \ldots, d$.

6.2. Efficient method for products under $k[x_1, \ldots, x_n]/I$.

When the ideal I is triangular, our algorithm can be greatly improved by optimizing the multiplication of polynomials modulo I. Indeed, we can incorporate the method described in [3] based on assessment techniques and on interpolation. This optimization is applicable to the function `ABV` and also in Step 1 of the parallel version. **Detecting roots over k.** We can lighten the calculations when there is $Q \in L.P$ such that $\lambda = Q$

* "... but as we are not able to see the degree of the final equation in x, even more we could reach an equation in x of a degree higher than it should be (and this is the drawback of elimination's methods), we felt obliged to show how this equation can be found in advance and ensure the precise degree to which it must climb."

mod I belongs to k. When $I = \mathfrak{M}$, by the Galois Theorem and Theorem 15, the resolvent has a root $Q(\underline{\alpha})$ in k if and only if $Q \mod \mathfrak{M}$ belongs to k. For any ideal I the resolvent may have a root in k but no polynomial Q of $L.P$ satisfies $\overline{Q} \in k$. However, the reciprocal of the preceding assertion is true as expressed in the following lemma:

Lemma 21. *Let $Q \in L.P$ and $\overline{Q} = Q \mod I$. If $\overline{Q} \in k$ then $x - \overline{Q}$ is a factor over k of the resolvent $R_{P,I}$; even more, for any $i \geq 0$,*

$$p_i(L.P \backslash \{Q\}) \mod I$$

is the ith power function s_i of the roots of $\frac{R_{P,I}}{x-\overline{Q}}$.

Proof. We always have $Q(\underline{\alpha}) = Q \mod \mathfrak{M}$. Suppose that $\lambda = Q \mod I$ belongs to k. Then $Q(\underline{\alpha}) = Q \mod I$ since $I \subset \mathfrak{M}$ and $Q(\underline{\alpha}) = Q \mod \mathfrak{M} = \lambda$. Therefore $x - \lambda$ is a factor of the resolvent $R_{P,I}$. We have $s_0 = d - 1$ and, for $i \geq 1$, $p_i(L.P \backslash \{Q\}) \mod I = (p_i(L.P) - \lambda^i) \mod I = (p_i(L.P) \mod I) - \lambda^i = s_i$. $\square$

When $\lambda \in k$, the polynomial Q is removed from the orbit $L.P$. These compute $s_0 =$ $, s_1, \ldots, s_{d-1}$, the d first power functions of the roots of $\frac{R_{P,I}}{x-\lambda}$. According to Lemma 21, this is possible since $s_0 = d - 1$ and, for $i = 1, \ldots, d - 1$, we have:

$$s_i = p \mod I \; for \; i \in [\![1, d-1]\!],$$

where $p = p_i(L.P \backslash \{Q\})$. Here, the polynomial p is not invariant by L and Theorem 15 does not apply. But since $p - s_i$ is an $\underline{\alpha}$-relation invariant by L, we get s_i as the reduction of p modulo I.

Conclusion

We have exploited the properties of galoisian ideals to develop our algorithm. For this, we generalized the fundamental theorem of symmetric functions and give an effective form of Galois Theorem. The time comparisons between two others techniques show the effectiveness of our algorithm `ABV`. Propositions for its optimization will provide significant gains. In the other hand, we are working on an implementation of the parallel version including the results of [3]. This implementation will be developed on `SAGEmath`.

References Computer Algebra Systems

- `SAGE` http://www.sagemath.org/
- `GAP` http://www.gap-system.org/
- `PrimitiveInvariant` `GAP`-Package of I. Abdeljaouad
  http://www-gap.mcs.st-and.ac.uk/Gap3/Contrib3/contrib.html
- `MAXIMA` http://maxima.sourceforge.net
- `Symmetries` in `MAXIMA`, author A. Valibouze
  http://maxima.sourceforge.net/docs/manual/en/maxima_32.html#SEC125
- `MAPLE` http://www.maplesoft.com

References

[1] I. Abdeljaouad. Calculs d'invariants primitifs de groupes finis. *Theor. Inform. Appl.*, 33(1):59–77, 1999.

[2] P. Aubry and A. Valibouze. Using Galois ideals for computing relative resolvents. *J. Symbolic Comput.*, 30(6):635–651, 2000.

[3] A. Bostan, M. Chowdhury, J. Van der Hoeven, and É. Schost. Homotopy methods for multiplication modulo triangular sets, 2009. Technical Report http://arxiv.org/abs/0901.3657v1, Arxiv. To appear in JSC.

[4] A. Cauchy. Usage des fonctions interpolaires dans la détermination des fonctions symétriques des racines d'une équation algébrique donnée. *Oeuvres*, **5**:473 Extrait 108, 1840.

[5] E. Galois. Oeuvres Mathmatiques, dites par la SMF *Gauthier-Villars, Paris*, 1897.

[6] A. Girard. Invention nouvelle en algèbre. *Amsterdam*, 1629.

[7] J.-L. Lagrange. Réflexions sur la résolution algébrique des équations. *Prussian Academy*, 1770.

[8] J.-L. Lagrange. *Oeuvres, Tome VIII, Notes sur la théorie des équations algébriques, Note X.* Publiées sous les auspices du ministère de l'instruction publique, 1808.

[9] A. Lascoux and M.P. Schützenberger. Formulaire raisonné de fonctions symétriques. Publication interne L.A. 248, Laboratoire LITP, Université Paris 7, France, 1985.

[10] I.G. Macdonald. *Symmetric Functions and Hall Polynomials, second ed.* Oxford: Clarendon Press. ISBN 0-19-850450-0 (paperback, 1998), 1995.

[11] N. Rennert and A. Valibouze. Calcul de résolvantes avec les modules de Cauchy. *Experiment. Math.*, 8(4):351–366, 1999.

[12] W. Schelter. *Manuel de Maxima*, 2001. (`http://maxima.sourceforge.net`).

[13] R.P. Stauduhar. The determination of Galois groups. *Math. Comp.*, 27:981–996, 1973.

[14] A. Valibouze. Théorie de Galois constructive, 1994. HDR, UPMC-Paris 6, France.

[15] A. Valibouze. Étude des relations algébriques entre les racines d'un polynôme d'une variable. *Bull. Belg. Math. Soc. Simon Stevin*, 6(4):507–535, 1999. (Version longue du rapport LIP6 1997/014).

[16] A. Valibouze. Classes doubles, idéaux de Galois et résolvantes. *Rev. Roum. de Math. Pures et Appl.*, 52 no 1, 2007.