École doctorale Blaise Pascal
UFR 922

Théorie des Invariants et Application à la Théorie de Galois effective

THÈSE

présentée et soutenue publiquement le 20 Septembre 2000

pour obtenir le grade de

DOCTEUR de l'UNIVERSITE PARIS 6

(Spécialité Informatique : Calcul Formel)

par

Ines Abdeljaouad

Président :	Mr. Daniel Lazard	Professeur à l'Université Paris 6
Rapporteurs :	Mr. Alain Lascoux	Directeur de Recherche au CNRS
	Mr. Antonio Machì	Professeur à l'Université de Rome
Examinateurs :	Mr. Jean-Marie Arnaudiès	Maître de conférence à l'Université Paris 6
	Mme Annick Valibouze	Professeur à l'Université Paris 6

Laboratoire d'Informatique Paris 6 — UMR 7503

Mis en page avec la classe thloria.

À ma mère et à mon père.

Remerciements

Je tiens à exprimer toute ma gratitude à ma directrice de thèse, Annick Valibouze, professeur à l'université Pierre et Marie Curie, pour m'avoir donné les moyens de mener à bien ce travail et pour m'avoir guidé dans mes études. Elle a dirigé mes travaux sans relâche, avec compétence et aussi une grande patience. Elle a eu le mérite de me faire partager son goût de la recherche, sa passion du savoir et sa persévérance. Qu'elle trouve ici l'expression de ma sincère reconnaissance.

Toute ma reconnaissance à Monsieur Daniel Lazard, professeur à l'Université Pierre et Marie Curie, pour m'avoir accueilli dans son équipe et qui m'a fait l'honneur de présider le jury de cette thèse.

Je remercie chaleureusement les professeurs Alain Lascoux et Antonio Machì d'avoir accepté d'être les rapporteurs attentionnés de cette thèse, de la lire et de la commenter.

Je remercie très sincèrement le Professeur Jean-Marie Arnaudiès d'avoir bien voulu accepter de faire partie du jury, mais également pour ses conseils avisés, sa disponibilité et son aide, ainsi qu'aux nombreux renseignements qu'il m'a communiqué. Qu'il trouve ici l'expression de ma profonde gratitude.

C'est avec une grande admiration que je rends hommage à toute l'équipe de Calcul Formel du LIP6 qui m'a accueillie, aidée et beaucoup appris. Que chaque membre de cette équipe dynamique, productive, solidaire et plurielle trouve ici l'expression personnelle de ma profonde reconnaissance. Je remercie également tous les collègues qui ont fréquenté le projet Galois.

La réalisation de cette thèse a été possible grâce au soutien matériel de l'UMS MEDICIS CNRS 658 et en particulier à Joël Marchand pour sa disponibilité. Merci aussi à tous les personnels du laboratoire, secrétaires, administrateurs systèmes et bibliothécaires.

Je tiens enfin à remercier les gens qui me sont chers. Malgré la distance, ma famille a toujours été dans mon coeur. Je remercie mon père et ma mère (à qui la thèse est dédiée), Ines, Khaled, Mahdi, Nidhal et Amine (6 mois) pour leur soutien.

Ma dernière pensée est pour mon tendre Naceur. Il a supporté l'éloignement de cinq ans en France, mais il a toujours été là pour m'aider à retrouver la confiance, la sérénité et la force de continuer.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Depuis le développement des mathématiques effectives, la théorie des invariants connaît un nouvel élan. L'avènement de l'ordinateur nous donne en effet, la possibilité de repousser les limites de ce qui est réellement calculable. Parmi les problèmes concrets à résoudre en théorie des invariants figurent :

- La mise au point d'algorithmes efficaces pour le calcul d'invariants de groupes,
- L'étude des *invariants classiques*.

Les deux premières parties de cette thèse suivent respectivement ces axes. Nous nous sommes intéressés à la théorie des invariants comme outil de la théorie de Galois. La troisième partie de cette thèse traite d'une méthode hybride de calcul du groupe de Galois et du corps de décomposition d'un polynôme à une variable.

$\star$

La première partie de ce document concerne la théorie des invariants moderne et propose de calculer des polynômes *invariants primitifs* de groupes. Ce sont des polynômes à plusieurs variables qui caractérisent à eux seuls les groupes finis.

Les travaux de E. Luther [54], A. Cayley [16], E.H. Berwick [10], H.O. Foulkes [27] permettent de déterminer des invariants primitifs associés à des groupes particuliers. La méthode naturelle pour le calcul d'un invariant primitif d'un groupe de permutations H consiste à sommer les monômes de l'orbite de $x_2x_3^2 \ldots x_n^{n-1}$ sous l'action de H. Le degré de l'invariant ainsi calculé est $\frac{n(n-1)}{2}$. Dans la perspective actuelle d'automatisation, il est nécessaire d'établir des algorithmes généraux efficaces de calcul d'invariants primitifs de degrés raisonnables ($\leq \frac{n(n-1)}{2}$) et en particulier ceux *relatifs* à des sous-groupes de S_n.

Nous introduisons dans le chapitre 1 une méthode de calcul de tous les invariants primitifs *réduits* ainsi qu'une implémentation d'un module informatique en GAP appelé « `PrimitiveInvariant` ».

Nous utilisons pour cette nouvelle méthode la représentation des polynômes en *listes de partitions d'entiers* donnée par C. Jordan dans [43]. Cette représentation des polynômes a permis à K. Girstmair [33] d'exhiber une technique de calcul d'*invariants primitifs absolus* de degré minimal et ceci pour tout groupe de permutations. Les deux premières sections

du chapitre 1 mettent en place les notations et les outils nécessaires à la généralisation de la méthode de Girstmair au calcul d'invariants primitifs relatifs ou absolues de degrés inférieurs à $\frac{n(n-1)}{2}$. Nous prouvons dans la troisième section du chapitre qu'un invariant primitif relatif est équivalent à un *ensemble essentiel* formé de listes d'entiers. C'est aussi l'utilisation des ensembles essentiels qui permet de calculer tous les invariants primitifs réduits. La quatrième section présente les algorithmes utilisés dans le calcul des invariants primitifs relatifs et en particulier ceux de degré minimal. La dernière section présente la représentation de données utilisée dans le module « `PrimitiveInvariant` » qui reprends les algorithmes de ce chapitre.

Le chapitre 2 rappelle une approche moins combinatoire et plus classique initiée par D. Hilbert [39] qui décrit un algorithme de calcul d'un système de générateurs d'anneaux d'invariants [69] : nous rappelons la principale méthode utilisée pour exprimer des invariants primitifs en fonction des générateurs de la base de l'anneau des invariants (voir par exemple les travaux de A. Colin [19]).

Nous reprenons dans la première section les principaux résultats utilisés dans le module informatique « `invar` » développé par G. Kemper [44] en MAGMA. Ce module détermine des *invariants fondamentaux* (i.e. *invariants primaires* et *secondaires*) qui génèrent tous les autres invariants. Nous présentons ensuite la méthode classique de calcul des invariants primitifs relatifs ou absolus à partir des invariants primaires et secondaires. Nous réalisons dans la deuxième section une comparaison entre les modules « `PrimitiveInvariant` » et « `invar` » pour le calcul des invariants primitifs. Cette comparaison met en évidence le fait que le module « `PrimitiveInvariant` » est plus performant pour le calcul des invariants primitifs relatifs que « `invar` » ; de plus, les invariants obtenues par ce dernier sont souvent de degré élevé.

$\star$

La deuxième partie de ce document étudie une méthode de calcul de résolvantes par la théorie des invariants classiques. Nous décrivons ainsi les invariants classiques et le passage vers la théorie moderne des invariants à l'aide d'algorithmes.

C'est grâce à un exemple donné par Boole il y a 150 ans que la théorie des invariants a vu le jour : le but étant de caractériser toutes les propriétés classiques et géométriques des formes binaires $f(x,y) = \sum_{k=0}^{n} C_n^k \, a_k \, x^k \, y^{n-k}$ invariantes par des transformations linéaires des variables x et y. Ainsi, pour $n = 2$ et $f(x,1) = a_2x^2 + 2a_1x + a_0$, Boole a noté que le discriminant $a_1^2 - a_0a_2$ est un *invariant classique* de formes binaires de degré 2 sous l'action de la translation $x \to x + c$ où c est une constante quelconque. Un autre exemple est donné par le résultant $R(f_1, f_2)$ qui peut aussi être considéré comme un invariant classique associé à deux formes binaires f_1 et f_2. Il caractérise les formes f_1 et f_2 par la propriété suivante : $R(f_1, f_2) = 0$ si et seulement si f_1 et f_2 ont des racines communes.
Des mathématiciens très célèbres ont effectué des calculs d'invariants classiques de formes

binaires comme par exemple J.J. Sylvester [70], P. Gordan [35], W.F. Meyer [56] et P.A. MacMahon [55]. Mais les notations utilisées restaient archaïques et même s'ils avaient des présomptions de preuves, les mathématiciens, comme A. Cayley [17] par exemple, n'avaient pas les moyens de les exprimer. Plusieurs aspects de la théorie des invariants ne sont pas expliqués de façon précise et suffisamment abstraite dans la littérature et c'est seulement à partir des travaux de D. Hilbert (voir [40]) que naquit la théorie modernes des invariants à l'aide d'un nouveau langage. L'idée est de déterminer un nombre fini d'invariants qui caractérisent tous les autres. A. Young [83], E. Study [68] et d'autres encore ont continué à travailler sur les invariants et à développer la théorie qui est devenue à la fois une branche classique et moderne des mathématiques. Au cours du 20^{e} siècle, plusieurs algorithmes de calculs d'invariants classiques ont été abandonnés, tandis que des progrès ont été fait en Algèbre. Le développement de la combinatoire et la modernisation de l'algèbre ont permis à la théorie des invariants modernes de se développer.

Le chapitre 3 présente un algorithme de calcul des invariants classiques de *degrés* et *poids* fixés, qui a donné lieu à une implémentation informatique. Cet algorithme permet également de déterminer un *système complet d'invariants classiques irréductibles*.

La première section définit la notion de *semi-invariants* et de *covariants* (c'est la forme générale des invariants classiques). Nous présentons ensuite un algorithme de calcul des invariants classiques qui se base sur la relation entre les invariants classiques et deux opérateurs différentiels particuliers. Nous réalisons dans cette même section une preuve de cette propriété théorique des invariants classiques. Nous étudions dans la deuxième section le lien entre les invariants classiques et modernes d'un point de vue de la théorie des groupes. Pour cela, nous décrivons les *notations symboliques* [47] qui permettent d'exprimer les invariants classiques en fonction de *polynômes-différences symétrisés*. Ces résultats nous font remarquer par exemple, qu'un invariant classique de degré d a un poids égal à $\frac{1}{2}nd$.

Le chapitre 4 automatise la *méthode de Berwick* pour le calcul de résolvantes en fonction d'invariants classiques et montre qu'il existe suffisamment d'invariants primitifs dont les résolvantes associées aient des coefficients invariants algébriques.

La résolvante est un polynôme à une variable résultant d'une transformation (par des invariants) sur les racines du polynôme $f(x, 1)$ et c'est un outil indispensable en théorie de Galois. De nombreux algorithmes furent développés depuis ceux de J.L. Lagrange [49] : R.P. Stauduhar [66] par des méthodes numériques, B. Soicher et J. McKay ont réalisés des calculs de résolvantes d'invariants linéaires avec des résultants (voir [63], [62] et [64]), A. Valibouze a mis au point un algorithme qui se base sur les polynômes symétriques ou les résultants, etc... Mais bien avant eux, A. Cayley et E.H. Berwick calculaient des résolvantes en se basant sur des invariants primitifs classiques dits *polynômes-différences* : les coefficients des résolvantes associées sont des invariants classiques. Nous montrons comment calculer ces résolvantes particulières et nous présentons de manière algorithmique la méthode de Berwick pour la détermination des *résolvantes de Lagrange*. La section

4.1 rappelle les principaux résultats en théorie de Galois et définit les résolvantes. Nous présentons dans la section 4.2 quelques propriétés des coefficients des résolvantes. La section 4.3 présente l'automatisation de la méthode de Berwick, l'avantage de cette méthode est qu'elle permet d'étudier en profondeur les invariants classiques et leur importance en théorie de Galois : le calcul des résolvantes et leur factorisation étant un outil servant à déterminer le groupe de Galois d'un polynôme à une variable.

$\star$

La troisième partie de ce document porte sur le groupe de Galois d'un polynôme et l'idéal des relations associé. Nous présentons une méthode nouvelle de calcul du groupe de Galois d'un polynôme qui utilise des méthodes d'algèbre linéaire en considérant le groupe de Galois comme un sous-groupe d'un groupe linéaire.

Jusqu'au 19$^{\text{e}}$ siècle, différentes méthodes coexistaient pour exprimer les racines du polynôme f en fonction de ses coefficients grâce à des fonctions simples (les opérations classiques classiques : addition, multiplication, soustraction, division) auxquelles les grecs ont rajouté l'extraction de racines. Ils connaissaient déjà des cas particuliers de la résolution de l'équation du second degré $a_2x^2 + 2a_1x + a_0 = 0$ par $x = \frac{1}{a_2}(-a_1 \pm \sqrt{a_1^2 - a_2a_0})$. De semblables formules avaient été trouvées pour les équations du troisième et quatrième degré par G. Cardan et un de ses disciples Ferrari [15]. C'est ce qui est communément appelé la *résolution par radicaux*. Les échecs répétés pour parvenir à une telle résolution dans le cas de l'équation du cinquième degré amenèrent E. Galois a introduire la notion de *groupe de l'équation* ou *groupe de Galois*. En effet, inspiré par les travaux de Lagrange, E. Galois a étudié une résolvante particulière (la *résolvante de Galois*) et a donné un critère qui indique le cas où un polynôme est résoluble par radicaux. Les travaux de J.M Arnaudiès et A. Valibouze [75] ont permis d'aboutir à une méthode déterministe de calcul du groupe de Galois, mais aussi [81], [62], [24] voir aussi [42]. L'intérêt de la méthode introduite dans le chapitre 5 est qu'elle combine deux approches différentes de la théorie de Galois. Elle étudie les relations entre les racines de f d'une part (les *idéaux de Galois*) et des équations qui caractérisent le groupe de Galois d'autre part (le *système de Hacque*).

Le chapitre 5 porte sur l'*idéal des relations* associé à une équation $f(x, 1)$. Cet idéal permet d'effectuer les calculs classiques dans le corps de décomposition de $f(x, 1)$ sans calculer explicitement ses racines (même lorsque le groupe de Galois n'est pas résoluble). Nous présentons ensuite les *idéaux de Galois* récemment introduits (voir [76]). Ils sont construits par inclusion grâce à la *méthode GI* et l'idéal de Galois contenant tous les autres est égal à l'idéal des relations de f. Nous présentons dans la section 5.3 la méthode *GI-complète* qui détermine l'idéal des relations de f et le groupe de Galois associé. Pour cela, nous avons implanté un algorithme qui calcule le groupe de Galois de f à partir des générateurs de l'idéal des relations entre les racines de f. La section 5.4 propose une méthode hybride appelée *la méthode de Hacque effective* qui est une méthode classique de calcul du groupe de Galois d'un polynôme irréductible f. Elle caractérise le groupe de

Galois en tant que sous-groupe du groupe classique linéaire $GL_n(k)$ grâce à un système d'équations. Cette méthode utilise les premières étapes de la méthode GI-complète avec la *méthode de Hacque* de la section 5.2.2 et nous obtenons ainsi une nouvelle approche effective de calcul du groupe de Galois. Nous concluons ce chapitre par la comparaison entre la méthode de Hacque effective et la méthode GI-complète.

$\star$

Parce que chaque système de calcul formel a sa particularité, nous utilisons divers logiciel pour effectuer des calculs : GAP est utilisée pour des manipulation des groupes, MAPLE ou ALDOR pour les polynômes, etc ...

Nous donnons en annexe A, le code du module « `PrimitiveInvariant` » en GAP qui reprend les algorithmes du chapitre 1. Nous présentons en annexe B des tables d'invariants primitifs relatifs ou absolus de degrés minimaux pour des sous-groupes du groupe de permutations S_9 déterminés par « `PrimitiveInvariant` », ainsi que leur temps de calcul. Nous réalisons dans l'annexe C quelques exemples d'invariants classiques et de semi-invariants célèbres calculés par la méthode du chapitre 3. Nous proposons ensuite les codes des différents algorithmes qui interviennent dans la deuxième partie de ce document.

Première partie

La Théorie des Invariants

Chapitre 1

Invariants primitifs

Sommaire

La synthèse des résultats de ce chapitre a été publié dans [3] et les invariants auxquels nous faisons référence sont des invariants dits *primitifs*: un invariant primitif d'un groupe de permutations est un polynôme qui à lui seul, permet d'identifier ce groupe. Nous introduisons dans ce chapitre, un nouvel outil de calcul de tous les invariants primitifs relatifs et absolus de groupes finis. Nous introduisons dans la première section quelques notation et définitions nécessaires à la compréhension des résultats donnés dans la section 1.2. Nous définissons dans la section 1.3 les ensembles essentiels et nous montrons le lien entre ces ensembles et les invariants primitifs. L'algorithme 1.4.3 nous donne des invariants différents et de degré minimum qui sont utilisés essentiellement pour la résolution d'équations polynomiales et les calculs de résolvantes dans la théorie de Galois (voir [74] et [59]).

Nous avons mis au point l'algorithme de Girstmair-Jordan qui, grâce à la représentation de données utilisée, calcule tous les polynômes invariants primitifs (relatifs ou absolus) à coefficients distincts de groupes finis. Enfin, dans la section 1.5, nous donnons les outils utilisés dans l'implantation en GAP des algorithmes de la section 1.4.

1.1 Définitions et Notations

Dans toute la suite, k désigne un corps commutatif de caractéristique nulle et, pour les n indéterminées $x_1, \ldots, x_n$ algébriquement indépendantes sur k, $k[x_1, \ldots, x_n]$ l'anneau des polynômes en ces variables et à coefficients dans k.

Le groupe symétrique de degré n, noté S_n, agit sur $k[x_1, \ldots, x_n]$ de façon naturelle par :

$$\begin{array}{ccc} S_n \times k[x_1,\ldots,x_n] & \longrightarrow & k[x_1,\ldots,x_n] \\ (\sigma\ ,\ P) & \mapsto & \sigma.P(x_1,\ldots,x_n) = P(x_{\sigma(1)},\ldots,x_{\sigma(n)}) \quad . \end{array}$$

Soient L un sous-groupe de S_n et P un polynôme de $k[x_1, \ldots, x_n]$. Le *stabilisateur de P sous l'action de L* est défini par : $Stab_L(P) = \{\sigma \in L \mid \sigma.P = P\} = L \cap Stab_{S_n}(P)$.

Le *stabilisateur d'une partie U de $k[x_1, \ldots, x_n]$ sous l'action de L* est

$$Stab_L(U) = \{\sigma \in L \mid \forall P \in U\ ,\ \sigma.P \in U\} \quad .$$

Soit H un sous-groupe de L tel que $H \subset L$, notons $e = [L : H]$ l'indice de H dans L. L'*orbite du polynôme P sous l'action de H* appelé aussi la *H-orbite de P* est définie par

$$H.P = Orb_H(P) = \{\sigma.P \mid \sigma \in H\} \quad .$$

Définition 1.1.1. Un polynôme $P \in k[x_1, \ldots, x_n]$ est un *H-invariant* si $H \subset Stab_{S_n}(P)$.

Définition 1.1.2. Un polynôme $P \in k[x_1, \ldots, x_n]$ est dit *H-invariant L-primitif* si

$$Stab_L(P) = H \quad .$$

Si $L = S_n$, alors P est appelé *H-invariant primitif (absolu)*.

Soit L' un groupe contenant H et contenu dans L. Si P est un H-invariant L-primitif alors, d'après la définition 1.1.2, P est également un H-invariant L'-primitif.

Un polynôme P est dit H-invariant L-primitif de *degré minimum* si le degré de tout polynôme H-invariant L-primitif est supérieur ou égal au degré de P. Parmi tous les H-invariants L-primitifs de degré minimum et à coefficient dans k, un polynôme dont le

nombre de monômes est minimum est appelé un *H-invariant L-primitif minimum.*

Exemple 1.1.3. Notons A_n le groupe alterné de degré n. Le déterminant de Vandermonde :

$$\delta_n = \prod_{1 \leq i < j \leq n} (x_i - x_j)$$

est un A_n-invariant primitif absolu. Nous montrerons plus loin que δ_n est un A_n-invariant primitif minimum.

Notation 1.1.4. Soient $\sigma_1, \ldots, \sigma_r$ des permutations de S_n, alors $< \sigma_1, \ldots, \sigma_r >$ désigne le sous-groupe de S_n engendré par les permutations $\sigma_1, \ldots, \sigma_r$.

Exemple 1.1.5. Soient $n = 4$, $H_1 =< (3,4), (1,2)(3,4), (1,3)(2,4) >$ un sous-groupe de S_4 et $H_2 =< (1,2)(3,4), (1,3)(2,4) >$ un sous-groupe de H_1. Les polynômes suivants sont des polynômes H_2-invariants H_1-primitifs de degré minimum :

$$x_2x_4 + x_1x_3 \qquad \text{et} \qquad x_2x_3 + x_1x_4 \qquad .$$

Un polynôme H_2-invariant H_1-primitif de degré $\frac{n(n-1)}{2} = 6$ est égal à :

$$a(x_2x_3^2x_4^3 + x_1x_4^2x_3^3 + x_4x_1^2x_2^3 + x_3x_2^2x_1^3) \qquad ,$$

où a est un élément non nul de k.

Notons $\{\tau_1, \ldots, \tau_e\}$ une transversale à gauche de L modulo H, (e étant l'indice de H dans L). Les classes à gauche de L modulo H sont $\tau_1 H, \ldots, \tau_e H$.

Remarque 1.1.6. Un polynôme P est un H-invariant L-primitif, lorsque pour tout $\sigma \in H$, $\sigma.P = P$ et pour tout $i \neq j$, $\tau_i.P \neq \tau_j.P$. Le nombre de conjugués de P sous l'action des τ_i est égal à l'indice de H dans L c'est à dire e.

Définition 1.1.7. Un *monôme* de $k[x_1, \ldots, x_n]$ est de la forme $x_1^{r_1} \ldots x_n^{r_n}$ où les r_i sont des entiers positifs ou nuls. Soit Q un monôme de $k[x_1, \ldots, x_n]$, nous notons

$$N_H(Q) \ = \sum_{Q' \in Orb_H(Q)} Q' \qquad .$$

Le polynôme $N_H(Q)$ est appelé *Trace réduite de Q par H.*

Définition 1.1.8. Soit U un ensemble fini de monômes de $k[x_1, \ldots, x_n]$. Le *(L,H)-groupe de U*, noté $H_L(U)$, est défini par :

$$H_L(U) \ = \bigcap_{Q \in U} Stab_L(N_H(Q)) \qquad .$$

Définition 1.1.9. Soit U un ensemble fini de monômes de $k[x_1, \ldots, x_n]$ dont les H-orbites sont deux à deux distinctes. Une *U-fonction élémentaire* est un polynôme $\mathcal{P}_U$ donné par la formule suivante :

$$\mathcal{P}_U = \sum_{Q \in U} a_Q N_H(Q)$$

où les a_Q sont des éléments de k non nuls et deux à deux distincts.

Exemple 1.1.10. Soient $U = \{x_1x_2, x_2x_3\}$ et $H =< (1,4) >$ un sous-groupe de S_4. La famille des U-fonctions élémentaires est donnée par $\mathcal{P}_U = a(x_1x_2 + x_2x_4) + b(x_2x_3)$ avec $a \neq b$ deux éléments non nuls de k.

Définition 1.1.11. Le *degré d'un ensemble U de monômes de $k[x_1, \ldots, x_n]$* est par définition égal au maximum des degrés des monômes dans U.

1.2 Résultats effectifs pour calculer des Invariants Primitifs

Le théorème et le corollaire suivants ont été énoncés dans [33], dans le cas où $L = S_n$. Nous présentons dans cette section une généralisation pour tout groupe L de S_n.

Théorème 1.2.1. *Soit U un ensemble fini de monômes appartenant à $k[x_1, \ldots, x_n]$ dont les H-orbites sont deux à deux distinctes. Toute U-fonction élémentaire $\mathcal{P}_U$ est un polynôme $H_L(U)$-invariant L-primitif.*

Preuve. Montrons qu'un polynôme $\mathcal{P}_U = \sum_{Q \in U} a_Q \, N_H(Q)$ (où $a_Q \in k$ sont non nuls et deux à deux distincts) est un $H_L(U)$-invariant L-primitif. D'après la définition 1.1.8 du (L,H)-groupe $H_L(U)$ et la définition 1.1.2 des H-invariants L-primitifs, ceci revient à montrer que :

$$Stab_L\left(\sum_{Q \in U} a_Q \, N_H(Q)\right) = \bigcap_{Q \in U} Stab_L(N_H(Q)) \qquad .$$

Soit σ un élément du groupe $Stab_L(\mathcal{P}_U)$, alors $\sigma.\mathcal{P}_U = \mathcal{P}_U$, soit :

$$\sigma.\mathcal{P}_U = \sum_{Q \in U} a_Q \, N_H(Q) \qquad .$$

Puisque les a_Q sont deux à deux distincts et que les traces réduites des éléments de U sont deux à deux distinctes, nous avons : $\sigma.N_H(Q) = N_H(Q)$ pour tout $Q \in U$ et donc $\sigma \in Stab_L(N_H(Q))$ pour tout $Q \in U$. D'où :

$$\sigma \in \bigcap_{Q \in U} Stab_L(N_H(Q)) \qquad .$$

Réciproquement, si $\sigma \in \bigcap_{Q\in U} Stab_L(N_H(Q))$ alors pour tout $Q \in U$ nous avons :

$$\sigma \in Stab_L(N_H(Q))$$

et donc $\sigma \in Stab_L(\sum_{Q\in U} a_Q\ N_H(Q))$. □

Remarque 1.2.2. En d'autres termes, le (L,H)-groupe $H_L(U)$ est le stabilisateur de chaque U-fonction élémentaire $\mathcal{P}_U$: $Stab_L(\mathcal{P}_U) = H_L(U)$.

Définition 1.2.3. Si $H_L(U) = H$, alors U est appelé *ensemble essentiel pour (L,H).*

Corollaire 1.2.4. *Soit U un ensemble fini de monômes. Si U est un ensemble essentiel pour (L,H) alors chaque U-fonction élémentaire est un H-invariant L-primitif.*

Proposition 1.2.5. *Si U est un ensemble essentiel pour (L,H), alors tout ensemble V contenant U est un ensemble essentiel pour (L,H).*

Preuve. $U \subset V \Rightarrow H_L(V) \subset H_L(U)$, d'autre part, $H_L(U) = H$ et $H \subset H_L(V)$, d'où l'égalité $H_L(V) = H$. □

Définition 1.2.6. Soit U un ensemble essentiel pour (L,H). Une *U-fonction primitive* est un polynôme $P = \sum_{Q\in U} a_Q\, N_H(Q)$ où les a_Q sont des éléments de k non nuls et où $Stab_L(P) = H$.

Remarque 1.2.7. Si U est un ensemble essentiel pour (L,H) alors chaque U-fonction élémentaire est une U-fonction primitive dont les coefficients sont deux à deux distincts.
Nous avons montré que toute U-fonction élémentaire où U est un ensemble essentiel pour (L,H), est un H-invariant L-primitif à coefficients distincts. Il reste à montrer que tout H-invariant L-primitif est égal à une U-fonction primitive où U est un ensemble essentiel pour (L,H). Ainsi, nous calculerons à l'aide d'ensembles essentiels pour (L,H) tous les H-invariants L-primitifs.

1.3 Ensembles Essentiels et Invariants primitifs

Proposition 1.3.1. *Soit Q un monôme de $k[x_1,\ldots,x_n]$. La trace réduite de Q par H est un polynôme H-invariant.*

Preuve. D'après la définition 1.1.7 de la trace réduite, pour toute transversale à gauche T de H modulo $Stab_H(Q)$, la trace réduite de Q est $N_H(Q) = \sum_{\tau\in T} \tau.Q$.
Fixons une telle transversale T et soit $\sigma \in H$. Alors σT est une autre transversale à gauche de H modulo $Stab_H(Q)$, d'où (puisque l'application $\tau \mapsto \sigma\tau$ est une bijection de T sur σT) :

$$\sigma.N_H(Q) = \sum_{\tau\in T} \sigma.(\tau.Q) = \sum_{\tau\in T} (\sigma\tau).Q = \sum_{\tau'\in\sigma T} \tau'.Q = N_H(Q) \qquad . \tag{1.1}$$

La trace réduite de Q est donc un H-invariant. □

Définition 1.3.2. Soit Q un monôme de $k[x_1,\ldots,x_n]$. Tout monôme Q' de $Orb_H(Q)$ est dit *représentant de* $N_H(Q)$. Le monôme Q' est aussi appelé *représentant de la H-orbite de Q.*

Ici, le corps k est un corps commutatif, de caractéristique quelconque et pas nécessairement algébriquement clos.
L'action naturelle de S_n sur $k[x_1,\ldots,x_n]$ est une action par automorphisme de k-algèbres graduées (la graduation considérée est la graduation canonique définie par le degré : pour tout $m \in \mathbf{N}$, la composante homogène de degré m de $k[x_1,\ldots,x_n]$ est donc le k-espace vectoriel $\mathcal{H}_m = k[x_1,\ldots,x_n]_m$ des polynômes homogènes de degré m). Nous en déduisons immédiatement qu'un polynôme $P \in k[x_1,\ldots,x_n]$ est H-invariant si et seulement si chacune de ses parties homogènes l'est.
Pour tout $m \in \mathbf{N}$, notons $\mathtt{Mon}_m(x_1,\ldots,x_n)$ l'ensemble des monômes de degré m : il est fini de cardinal C^m_{m+n-1}, et H-stable, donc réunion de H-orbites. C'est une base du k-espace vectoriel $\mathcal{H}_m$. Notons Ω_m une partie de $\mathtt{Mon}_m(x_1,\ldots,x_n)$ rencontrant chaque H-orbite ω de $\mathtt{Mon}_m(x_1,\ldots,x_n)$ suivant un singleton, qu'on notera $\{Q_\omega\}$. Il est clair que l'ensemble R^H_m des éléments de $\mathcal{H}_m$ qui sont H-invariants est un sous-k-espace vectoriel de $\mathcal{H}_m$.
L'ensemble R^H de tous les éléments de $k[x_1,\ldots,x_n]$ qui sont H-invariants est une sous-k-algèbre graduée de $k[x_1,\ldots,x_n]$, dont les composants homogènes sont les R^H_m pour m décrivant $\mathbf{N}$. Nous avons donc :

$$\mathtt{R}_H = \bigoplus\nolimits_{m\in\mathbf{N}} R^H_m \qquad . \tag{1.2}$$

Proposition 1.3.3. *Soit $m \in \mathbf{N}$. Avec les notations et hypothèses ci-dessus, une base du k-espace vectoriel R^H_m est la famille $(N_H(Q_\omega))_{\omega\in\Omega_m}$.*

Preuve. D'après la proposition 1.3.1, nous avons $N_H(Q_\omega) \in \mathtt{R}_{H,m}$ pour tout $\omega \in \Omega_m$.
Montrons d'abord que la famille $(N_H(Q_\omega))_{\omega\in\Omega_m}$ est k-linéairement indépendante. Pour tout $\omega \in \Omega_m$, les définitions montrent que $N_H(Q_\omega) = \sum_{Q'\in\omega} Q'$. Comme la famille $(\omega)_{\omega\in\Omega_m}$ est une partition de $\mathtt{Mon}_m(x_1,\ldots,x_n)$, la k-indépendance linéaire voulue en découle immédiatement.
Montrons enfin que la famille $(N_H(Q_\omega))_{\omega\in\Omega_m}$ engendre le k-espace vectoriel R^H_m. Pour tout $\sigma \in H$, nous avons (puisque l'application $M \mapsto \sigma.M$ est une permutation de $\mathtt{Mon}_m(x_1,\ldots,x_n)$) :

$$P = \sigma.P = \sum_{M\in\mathtt{Mon}_m(x_1,\ldots,x_n)} \lambda_M\ (\sigma.M) = \sum_{M\in\mathtt{Mon}_m(x_1,\ldots,x_n)} \lambda_{\sigma^{-1}.M}\ M \qquad ,$$

d'où $\lambda_{\sigma^{-1}.M} = \lambda_M$ pour tout $M \in \mathtt{Mon}_m(x_1,\ldots,x_n)$. La fonction $M \mapsto \lambda_M$ est donc constante sur chaque H-orbite $\omega \in \Omega_M$; notant c_ω la valeur de cette constante, nous avons donc, par regroupements de termes :

$$P = \sum_{\omega\in\Omega_M} c_\omega \Big(\sum_{M\in\omega} M\Big) = \sum_{\omega\in\Omega_M} c_\omega N_H(Q_\omega) \qquad . \tag{1.3}$$

Ce qui achève la démonstration. □

En tenant compte de (1.2), nous déduisons de la proposition 1.3.3 :

Corollaire 1.3.4. *Pour tout $m \in \mathbf{N}$, soit Ω_m une partie de $\mathtt{Mon}_m(x_1, \ldots, x_n)$ rencontrant chaque H-orbite ω de $\mathtt{Mon}_m(x_1, \ldots, x_n)$ suivant un singleton $\{Q_\omega\}$.*
La famille $(Q_\omega)_{\omega \in \bigcup_{m \in \mathbf{N}} \Omega_m}$ est une base du k-espace vectoriel R^H.

Chaque H-orbite pouvant être représentée par un de ses monômes, l'ensemble de ces représentant forme un *système de représentants des orbites de H* noté $\mathcal{S} = \{Q_\omega \mid \omega \in \cup_{m \in \mathbf{N}} \Omega_m$. Dans ce qui suit, nous fixons $\mathcal{S}$ un système de représentants des orbites de H.

Théorème 1.3.5. *Soit $P \in k[x_1, \ldots, x_n]$ un H-invariant L-primitif à coefficient distincts. Il existe un unique sous-ensemble U de $\mathcal{S}$ tel que P soit une U-fonction primitive.*

Preuve. Soit P un H-invariant L-primitif. D'après la proposition 1.3.3, P s'écrit sous la forme $P = \sum_{i=1}^{l} c_i \, N_H(Q_i)$ avec c_i des éléments de k deux à deux distincts et où Q_i sont des éléments de $\mathcal{S}$ pour tout $i \in [1, l]$. Le choix des monômes Q_i est unique.
Soit $U = \{Q_1, \ldots, Q_l\}$. D'après la définition 1.2.6, il reste à montrer que U est un ensemble essentiel pour (L,H), ce qui revient à montrer d'après la définition 1.2.3 que $H_L(U) = H$.
Le (L,H)-groupe de U est défini par $H_L(U) = \bigcap_{i=1}^{l} Stab_L(N_H(Q_i))$.
$H \subset H_L(U)$, en effet : pour tout $i \in [1, l]$ $H \subset Stab_L(N_H(Q_i))$ car les $N_H(Q_i)$ sont des H-invariants (voir proposition 1.3.1).
D'autre part, si $\sigma \in H_L(U)$ alors $\sigma \in \bigcap_{i=1}^{l} Stab_L(N_H(Q_i))$ et donc pour tout $i \in [1, l]$, $\sigma.N_H(Q_i) = N_H(Q_i) \Rightarrow \sigma \in Stab_L(P)$. Ainsi $H_L(U) = Stab_L(P) = H$. □

Corollaire 1.3.6. *Pour calculer tous les H-invariants L-primitifs à coefficients distincts, il suffit de calculer tous les ensembles essentiels pour (L,H).*

Preuve. D'après le théorème 1.3.5 et la remarque 1.2.7, tout H-invariant L-primitif à coefficients distincts s'écrit sous la forme d'une U-fonction élémentaire où U est un ensemble essentiel pour (L,H). D'autre part, toute U-fonction élémentaire où U est un ensemble essentiel pour (L,H) est un H-invariant L-primitif (voir corollaire 1.2.4). Ainsi, le calcul de tous les H-invariants L-primitifs à coefficients distincts, revient à calculer toutes les U-fonctions élémentaires c'est à dire à calculer tous les ensembles U essentiels pour (L,H). □

Pour le calcul de tous les H-invariants L-primitifs, il faut d'abord calculer tous les ensembles essentiels pour (L,H). Pour chaque ensemble essentiel U pour (L,H) il faut tester si chaque polynôme de la forme $\sum_{Q \in U} a_Q N_H(Q)$ où les a_Q sont non nuls, est un H-invariant L-primitif. Si les coefficients a_Q sont deux à deux distincts alors d'après le corollaire 1.3.6, les polynômes $\mathcal{P}_U = \sum_{Q \in U} a_Q N_H(Q)$ sont des H-invariants L-primitifs.
Sinon, si les coefficients a_Q ne sont pas distincts, alors il faut vérifier que le nombre de conjugués du polynôme $\sum_{Q \in U} a_Q N_H(Q)$ sous l'action des éléments de la classe de conjugaison de L par H, est égal à l'indice de H dans L (voir remarque 1.1.6).

1.4 Algorithmes de calculs des Ensembles Essentiels

Rappelons que $\mathcal{S}$ est un système de représentant des orbites de H. Pour calculer un H-invariant S_n-primitif, K. Girstmair utilise une technique semblable à celle de l'algorithme 1.4.3 sauf qu'il n'obtient qu'un seul ensemble essentiel pour (L,S_n) de degré minimum (voir [33]). De plus, le choix de cet ensemble se fait d'une manière aléatoire de sorte qu'il n'a pas toujours un H-invariant primitif absolu et minimum. Nous présentons, dans la première partie de ce paragraphe, un algorithme de calcul de l'ensemble essentiel de $\mathcal{S}$ de degré minimum et contenant tous les autres ensembles essentiels de degré minimum. Dans la deuxième partie de cette section, nous donnons un algorithme de calcul de tous les invariants primitifs relatifs ou absolus et de degré minimum ainsi qu'un algorithme de calcul de tous les ensembles essentiels de degrés $\leq \frac{n(n-1)}{2}$ de $\mathcal{S}$ et donc de calcul de tous les invariants primitifs relatifs ou absolus de degrés $\leq \frac{n(n-1)}{2}$ et à coefficients distincts (voir corollaire 1.3.6).

1.4.1 Système de Représentants et Ensembles Essentiels

Algorithme 1.4.1 (SystèmeReprésentant). *Cet algorithme calcule un système de représentants des H-orbites de monômes de degré $\leq \frac{n(n-1)}{2}$.*

Fonction `SystèmeReprésentant(n,H) ==`

`Entrées :` un entier n et un sous-groupe H de S_n.
`Sortie :` Un système de représentants des orbites de H de degré $\leq \frac{n(n-1)}{2}$.

```
1.    Soit A l'ensemble des monômes de degrés ≤ n(n-1)/2.
      Pour Tout m et m' dans A de même degré Faire
            Si Orb_H(m) = Orb_H(m')
                  Alors retirer m' de A
            Fin Si
      Fin Pour
2.    Retourner(A) ;
Fin.
```

Preuve. L'algorithme termine au bout d'un nombre fini d'étapes puisque l'ensemble $\mathcal{A}$ est fini. D'autre part, en remarquant que deux monômes de degrés distincts n'ont pas la même H-orbite, pour aller plus vite, il suffit de ne comparer que les monômes de même degré. □

Notations 1.4.1. Soit $\mathcal{A}$ un ensemble fini de monômes. Notons `premier`($\mathcal{A}$) la liste de tous les éléments de $\mathcal{A}$ de degré minimum et `rest`($\mathcal{A}$) la liste des éléments de $\mathcal{A}$ privé de `premier`($\mathcal{A}$).

Algorithme 1.4.2 (EnsembleEssentiel). *Soit $\mathcal{A}$ un sous-ensemble fini de $\mathcal{S}$. Nous présentons l'algorithme pour le calcul d'un ensemble essentiel pour (L,H) dans $\mathcal{A}$. L'ensemble U essentiel pour (L,H) obtenu grâce à cet algorithme, est parmi tous les ensembles essentiels dans $\mathcal{A}$, celui de degré minimum d qui contient tous les autres ensembles essentiels pour (L,H) dans $\mathcal{A}$ de degré d.*

Fonction `EnsembleEssentiel(`$\mathcal{A}$`) ==`

`Entrée :` `Un ensemble` $\mathcal{A}$ `fini de` $\mathcal{S}$.
`Sortie :` `Un ensemble essentiel` $U \subset \mathcal{A}$ `pour` (L,H), `s'il existe.`
1. $U := \{1\}$;
2. `Tant que` $H_L(U) \neq H$ `Et` $\mathcal{A} \neq \{\ \}$ `Faire`
   $U :=$ `rest(`$\mathcal{A}$`) ;`
   $\mathcal{A} :=$ `premier(`$\mathcal{A}$`) ;`
   `Fin Tant que`
3. `Si` $\mathcal{A} \neq \emptyset$
   `Alors` **Retourner**(U) ;
   `Sinon Pas d'ensemble essentiel pour` (L,H) `dans` $\mathcal{A}$.
   `Fin Si`
`Fin.`

Preuve. Puisque $\mathcal{A}$ est fini, cet algorithme récursif s'arrête au bout d'un nombre fini d'étapes avec $\mathcal{A} = \{\ \}$ ou $H_L(U) = H$. En effet : à chaque étape de l'algorithme, si $H_L(U) \neq H$ alors aucun sous-ensemble U n'est un ensemble essentiel pour (L,H). En effet, supposons qu'il existe $V \subset U$ un ensemble essentiel pour (L,H). Alors, d'après la proposition 1.2.5, U est aussi un ensemble essentiel pour (L,H) et donc $H_L(U) = H$, d'où l'absurdité.
Soit U l'ensemble essentiel pour (L,H) obtenu au bout d'un nombre fini d'étapes de la boucle 2. de l'algorithme. Parmi tous les autres ensembles essentiels pour (L,H) contenus dans $\mathcal{A}$, U est un ensemble essentiel de degré minimum parce que c'est le premier trouvé. Enfin, U contient tous les monômes de $\mathcal{A}$ de degré inférieur ou égal à d. Il contient donc tous les ensembles dans $\mathcal{A}$ essentiels pour (L,H) et de degré d. □

Remarque 1.4.2. Soit U un système de représentants des H-orbites des monômes de degré égal à $\frac{n(n-1)}{2}$, alors U contient nécessairement un représentant de la H-orbite du monôme $x_2x_3^2 \ldots x_n^{n-1}$. D'autre part, la trace réduite du représentant de ce monôme est un H-invariant L-primitif (voir la méthode classique de l'introduction). Ainsi d'après la proposition 1.2.5, U est aussi un ensemble essentiel pour (L,H).

1.4.2 Invariants Primitifs de Degré Minimum

Définition 1.4.3. Soit U un ensemble essentiel pour (L,H). Si U ne contient aucun ensemble essentiel pour (L,H) alors U est dit *réduit*. Toutes les U-fonctions primitives

sont alors *réduites*.

Notations 1.4.4. Pour U un ensemble fini de monômes, `partie`(U) est la liste de tous les sous-ensembles de U classés par ordre croissant suivant leurs tailles et `diff`(U,V) est la liste des sous-ensembles de U privée des ensembles contenant V.

Algorithme 1.4.3 (InvariantsPrimitifsDeDegréMinimum). *L'algorithme suivant calcule tous les ensembles essentiels réduits pour (L,H) et de degré minimum dans $\mathcal{S}$. À chaque ensemble essentiel U nous donnons en guise d'exemple les polynômes H-invariants L-primitifs à coefficients distincts $\mathcal{P}_U$.*

Fonction `InvariantsPrimitifsDeDegréMinimum(`L`,`H`) ==`

`Entrées :` Deux sous-groupes L et H de S_n vérifiant $H \subset L$.
`Sortie :` Une liste de polynômes H-invariants L-primitifs réduits, de degré minimum et à coefficients deux à deux distincts.

```
1.      A := SystèmeReprésentant(n,H) ;
2.      U := EnsembleEssentiel(A,L,H) ;
3.      I := { } ;
4.      E := partie(U) ;
5.      Pour tout V ⊂ E Faire
                Si H_L(V) = H Alors
                        Rajouter V dans I ;
                        E := diff(U,V) ;
                Fin Si
        Fin Pour
        (I contient tous les ensembles essentiels
        réduits pour (L,H) de degré minimum).
6.      Retourner(I) ;
7.      Pour tout U dans I, donner les polynômes P_U.
Fin.
```

Preuve. Soit $\mathcal{A}$ un système de représentants des H-orbites de monômes de degrés $\leq \frac{n(n-1)}{2}$ obtenu par l'algorithme 1.4.1 appliqué à (n,H). L'algorithme 1.4.2, termine avec un ensemble essentiel. En effet, U contient nécessairement un représentant de la H-orbite du monôme $x_2x_3^2 \ldots x_n^{n-1}$ et d'après la remarque 1.4.2 nous savons qu'au pire des cas, c'est à dire lorsque U est de degré $\frac{n(n-1)}{2}$, l'algorithme 1.4.1 termine avec la condition d'arrêt $H_L(U) = H$ et $\mathcal{A} = \{\ \}$.
L'ensemble U obtenu à l'étape 2. de l'algorithme, est un ensemble essentiel de degré minimum parmi les ensembles essentiels dans $\mathcal{A}$ et il contient tous les autres ensembles essentiels de degré minimum dans $\mathcal{A}$ (voir la preuve de l'algorithme 1.4.2). La boucle 5. de l'algorithme calcule tous les sous-ensembles V de U qui sont essentiels et réduits et donne les polynômes $\mathcal{P}_V$ qui sont des H-invariants L-primitifs à coefficients distincts (voir

corollaire 1.3.6). □

Remarque 1.4.5. Le calcul d'ensembles essentiels réduits revient à ne pas calculer les polynômes invariants primitifs somme de deux autres.

Remarque 1.4.6. Pour le calcul de tous les H-invariants L-primitifs de degré minimum dont les coefficients peuvent être égaux, il faut d'abord calculer tous les ensembles essentiels pour (L,H) de degré minimum (voir algorithme 1.4.3) et il faut vérifier que le nombre de conjugués des polynômes $\sum_{Q\in U} a_Q N_H(Q)$ avec $a_Q \in k^*$, sous l'action des éléments de la classe de conjugaison de L par H, est égal à l'indice de H dans L (voir remarque 1.1.6).

1.4.3 Remarques sur le calcul de tous les invariants primitifs

Nous avons montré dans les sections précédentes que le calcul de tous les invariants primitifs à coefficients distincts, est le calcul de toutes les U-fonctions primitives où U sont des ensembles essentiels. L'algorithme précédent nous donne tous les ensembles essentiels U réduits de degré minimum, ainsi nous obtenons toutes les U-fonctions élémentaires réduites par le simple calcul de polynômes dont les monômes sont les traces réduites des éléments de U et dont les coefficients sont deux à deux distincts. Le calcul des U-fonctions primitives qui ne sont pas des U-fonctions élémentaires nécessite, à part le calcul de l'ensemble essentiel U, de tester que le nombre de conjugués de ces polynômes est égal à l'indice de H dans L (voir remarques 1.4.6 et 1.1.6), et ce test est très coûteux à cause du nombre de polynômes à tester.
Le plus important, selon notre avis, dans le calcul d'invariants primitifs est de trouver les monômes qui forment ces polynômes c'est à dire les ensembles essentiels.

1.4.4 Algorithme de calcul des invariants primitifs de degrés $\leq \frac{n(n-1)}{2}$

Notation 1.4.7. Soit U un ensemble fini de monômes, `degre`(U) est le degré de l'ensemble U.

Algorithme 1.4.4 (Girstmair-Jordan). *Cet algorithme calcule tous les ensembles essentiels réduits pour (L,H) de degrés $\leq \frac{n(n-1)}{2}$. Nous allons voir que grâce à la représentation de données (paragraphe 1.5), nous calculons avec ce même algorithme tous les ensembles essentiels pour (L,H).*

Fonction `Girstmair-Jordan`(L,H) ==

```
Entrées :  Deux sous-groupes L et H de S_n vérifiant H ⊂ L.
Sortie :   Tous les ensembles essentiels réduits pour (L,H)
           de degré ≤ n(n−1)/2.
1.         A := SystèmeReprésentant(n,H) ;
2.         d := 1 ;
3.         I := { } ;
4.         Tant que d ≤ n(n−1)/2 Faire
                   U := EnsembleEssentiel(A,L,H) ;
5.                 Pour tout V ⊂ E Faire
                       Si H_L(V) = H Alors
                               Rajouter V dans I ;
                               E := diff(U,V) ;
                       Fin Si
               Fin Pour
      d := degre(U) + 1 ;
      Fin Tant que
      I contient tous les ensembles essentiels réduits
      pour (L,H) de degrés inférieurs ou égaux à n(n−1)/2 ;
6.    Retourner(I) ;
Fin.
```

Preuve. À chaque étape de la boucle 4. $\mathcal{A}$ change de valeur et ne contient plus que des monômes de degré supérieur strictement au degré du dernier ensemble essentiel obtenu. À la fin de la boucle 4., l'ensemble essentiel obtenu est de degré $\frac{n(n-1)}{2}$ et l'algorithme termine avec $d = \frac{n(n-1)}{2} + 1$.
Tous les ensembles essentiels obtenus sont réduits, en effet : soit U un ensemble essentiel obtenu à l'étape (p) de la boucle 4., alors U=`EnsembleEssentiel`($\mathcal{A}$,L,H) et $\mathcal{A}$ change de valeur et ne contient plus que les monômes de degré supérieur strictement au degré de U (voir l'algorithme 1.4.2). Ainsi, le nouvel ensemble essentiel calculé à la $(p+1)$-ème étape de la boucle 4. sera contenu dans ce nouvel $\mathcal{A}$ et l'intersection de l'ensemble essentiel obtenu à l'étape $(p+1)$ avec U est l'ensemble vide. D'autre part, la boucle 5. nous donne tous les ensembles essentiels réduits d'un même degré. □

1.5 Partitions de monômes et Invariants primitifs

Dans ce paragraphe est présentée la notion de *partition*, qui a été introduite pour la première fois par C. Jordan [43] et ses contemporains du siècle dernier. Nous mettons en évidence la correspondance entre monômes et partitions et nous montrons comment le problème de calcul d'invariants primitifs qui, à première vue, est purement algébrique se transforme en un problème de combinatoire des groupes et des ensembles.
En effet, vis-à-vis d'un groupe de permutation, un monôme ou son carré donnent la même information. Ce qui compte donc c'est la partition de l'ensemble $\{1, \ldots, n\}$ associée à un

monôme. Remarquons que la notion de partition définie ci-dessous ne coïncide pas avec la notion de partition habituellement donée en analyse combinatoire (au sens combinatoire, une *partition* de n est une suite $(\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n$ telle que $\sum_{k=1}^{k=n} k\alpha_k = n$).

1.5.1 Partitions

Définition 1.5.1. Une *partition* $T = (T_1, \ldots, T_s)$ *de l'ensemble* $\{1, \ldots, n\}$ est une liste de sous-ensembles non vides de $\{1, \ldots, n\}$ classés par cardinal décroissant et tels que les ensembles $T_1, \ldots, T_s$ soient deux à deux disjoints et que leur réunion soit égale à l'ensemble $\{1, \ldots, n\}$.

Le cardinal d'un sous-ensemble I de $\{1, \ldots, n\}$ est noté $\mid I \mid$. Notons $\mathcal{T}$ l'ensemble des partitions de $\{1, \ldots, n\}$. D'après la définition 1.5.1, une partition $T = (T_1, \ldots, T_s)$ de $\mathcal{T}$ vérifie donc les quatre propriétés suivantes :

(i) $\forall i \in [1, n] \;\; T_i \subset \{1, \ldots, n\}$.

(ii) $\bigcup_{i=1}^{s} T_i = \{1, \ldots, n\}$.

(iii) $\forall\, i, j \in [1, n] \;\; i \neq j \Longrightarrow T_i \cap T_j = \emptyset$.

(iv) $\forall i \in [1, n] \;\; \mid T_i \mid \neq 0$ et $\mid T_1 \mid \geq \mid T_2 \mid \geq \cdots \geq \mid T_s \mid \geq 1$.

1.5.2 Représentation des données

Notation 1.5.2. Soient I un sous-ensemble de $\{1, \ldots, n\}$ et β un entier. Par convention, notons :

$$x_I^{\beta} = \left(\prod\nolimits_{i \in I} x_i\right)^{\beta}.$$

Un monôme Q de $k[x_1, \ldots, x_n]$ peut toujours s'écrire de manière unique sous la forme :

$$Q = x_{T_1}^{\beta_1} \ldots x_{T_s}^{\beta_s}$$

où $\beta_1, \ldots, \beta_s$ sont des entiers deux à deux distincts et $(T_1, \ldots, T_s)$ est une partition de $\{1, \ldots, n\}$ vérifiant pour tout $i, j \in [1, s]$ tels que $i < j$: ou bien $\mid T_i \mid > \mid T_j \mid$ ou bien $\mid T_i \mid = \mid T_j \mid$ et $\beta_i < \beta_j$.

Introduisons l'application surjective Ψ définie par :

$$\begin{array}{ccccc} \Psi & : & \{\prod_{i=1}^{i=n} x_i^{r_i} \mid (r_1, \ldots, r_n) \in \mathbf{N}^n\} & \longrightarrow & \mathcal{T} \\ & & Q = x_{T_1}^{\beta_1} \ldots x_{T_s}^{\beta_s} & \mapsto & \Psi(Q) = (T_1, \ldots, T_s) \;\; . \end{array}$$

Définitions 1.5.3. Soit $T = (T_1, \ldots, T_s) \in \mathcal{T}$. Nous définissons le monôme Q_T et un ensemble de monômes $\mathcal{M}$ par :

$$Q_T = \prod\nolimits_{i=1}^{i=s} x_{T_i}^{(i-1)} \qquad \text{et} \qquad \mathcal{M} = \{Q_T \mid T \in \mathcal{T}\} \qquad .$$

Le *degré* d'une partition T est par définition égal au degré du monôme Q_T c'est à dire égal à $\sum_{i=1}^{i=s}(i-1) \mid T_i \mid$.

Proposition 1.5.4. *L'application* Φ *définie par :*

$$\begin{array}{ccccc} \Phi & : & \mathcal{T} & \longrightarrow & \mathcal{M} \\ & & T & \mapsto & Q_T \end{array}$$

est bijective.

Preuve. Évident. □

Exemple 1.5.5. Soient $Q = x_1x_2x_3^2x_4^3x_5^2x_6^2x_7$ et $n = 8$ alors :

$$\Psi(Q) = (\{1,2,7\},\{3,5,6\},\{8\},\{4\}) \quad .$$

En effet : $Q = (x_1x_2x_7)^1(x_3x_5x_6)^2{x_8}^0{x_4}^3$.

Exemple 1.5.6. Soient $n = 9$ et $T = (\{2,3,4,5\},\{7,8,9\},\{1\},\{6\})$ un élément de $\mathcal{T}$. Alors, $\Phi(T) = Q_T = x_7x_8x_9{x_1}^2{x_6}^3$ et le degré de T est égal à 8.

La correspondance entre partitions et monôme étant démontré, il suffit d'appliquer les algorithmes de la section 1.4 à l'ensemble des partitions plutôt qu'aux monômes et de faire le calcul d'ensembles essentiels de partitions. En effet, la manipulation des listes et des ensembles d'entiers (partitions) est beaucoup plus facile que la manipulation des monômes et des polynômes.

1.5.3 Calcul de tous les Invariants Primitifs avec des partitions

Nous remarquons tout d'abord que l'ensemble de partitions $\mathcal{T}$ est un ensemble fini et que le degré d'un ensemble de partitions varie entre 1 et $\frac{n(n-1)}{2}$.

En remplaçant dans les algorithmes 1.4.1 et 1.4.4 appelés *SystèmeReprésentant* et *Algorithme de Girstmair-Jordan*, les monômes par les partitions, nous obtenons un algorithme qui calcule tous les ensembles essentiels de partitions à partir desquels, nous obtenons tous les invariants primitifs à coefficients distincts appelés *invariants primitifs réduits*. Voici un exemple de la représentation des donnés utilisée dans l'implantation de l'algorithme **Girstmair-Jordan** dans le cas où $n = 3$.

la liste des partitions en degré 3 est égale à :

$$[[[1,2,3]],[[1,2],[3]],[[1,3],[2]],[[2,3],[1]],[[1],[2],[3]],$$
$$[[1],[3],[2]],[[2],[1],[3]],[[2],[3],[1]],[[3],[1],[2]],[[3],[2],[1]]] \quad .$$

Un système de représentants des S_2-orbites de partitions est égal à :

$$[[[1,2,3]],[[1,2],[3]],[[1,3],[2]],[[1],[2],[3]],[[1],[3],[2]],[3],[1],[2]]] \ .$$

La liste des ensembles essentiels réduits pour (S_2,S_3) est égale à :

$$[[[[1,2],[3]]],[[[1,3],[2]]],[[[1],[2],[3]]],$$
$$[[[1],[3],[2]]],[[[3],[1],[2]]]] \ .$$

La famille des polynômes S_2-invariants S_3-primitifs réduits est égale à :
$a(x_1x_2)^\eta x_3{}^\beta$, $a((x_1x_3)^\eta x_2{}^\beta+(x_2x_3)^\eta x_1{}^\beta)$, $a(x_1{}^\eta x_2{}^\beta x_3{}^\gamma+x_2{}^\eta x_1{}^\beta x_3{}^\gamma)$, avec η, β, γ trois entiers distincts et a un élément de k.

1.6 Exemples d'applications

L'implantation des algorithmes 1.4.3 et 1.4.4 dans le système de Calcul formel GAP (voir [61]), à été possible grâce à la manipulation des partitions. Pour plus de détails sur l'implantation le lecteur pourra aussi consulter [2] (voir [1] pour l'implantation en AXIOM).

1.6.1 Exemples d'utilisation

Les polynômes invariants primitifs suivis de * peuvent être obtenus par les méthodes classiques ou la méthode naturelle discutées dans l'introduction.

Notation 1.6.1. Le monôme $x_1^{r_1}x_2^{r_2}\dots x_n^{r_n}$ sera représenté par la liste $[r_1r_2\dots r_n]$.

Exemple 1.6.2. Soient $n=5$, $L=S_5$ et $H=<(2,3)(4,5),(4,5),(3,4)>$. Un H-invariant L-primitif minimum est égal à x_1. Un autre H-invariant L-primitif de degré minimum est $x_5+x_4+x_3+x_2$.

Remarque 1.6.3. Soient P un H-invariant L-primitif et Q un polynôme tels que $P+Q$ est un L-invariant. Alors Q est aussi un H-invariant L-primitif, car $\forall\sigma\in L\,,\ \sigma(P)=P$ équivaut à $\forall\sigma\in L\,,\ \sigma(Q)=Q$.

Exemple 1.6.4. Soient $n=5$, $L=<_{(2,3)(4,5),(2,4)(3,5),(4,5),(3,4)}>$ et $H=<_{(2,3),(2,4)(3,5)}>$. Nous adoptons la notation 1.6.1 et nous supposons que les entiers i,j,l,h,m sont deux à deux distincts. L'algorithme 1.4.4 calcule la liste suivante des H-invariants L-primitifs réduits :

$$
\begin{gathered}
[iiijj]+[ijjii],\\
[iijij]+[iijji]+[ijiij]+[ijiji],\\
[iiijh]+[iiihj]+[ijhii]+[ihjii],\\
[iijih]+[iihij]+[iijhi]+[iihji]+[ijiih]+[ihiij]+[ijihi]+[ihiji],\\
[iijjh]+[iijhj]+[ijijh]+[ijihj]+[ijhij]+[ihjij]+[ijhji]+[ihjji],\\
[iihjj]+[ihijj]+[ijjih]+[ijjhi],\\
[jiijh]+[jiihj]+[jjhii]+[jhjii],\\
[hiijj]+[hjjii],\\
[jijih]+[jihij]+[jijhi]+[jihji]+[jjiih]+[jhiij]+[jjihi]+[jhiji],\\
[hijij]+[hijji]+[hjiij]+[hjiji],\\
[iijhl]+[iijlh]+[ijihl]+[ijilh]+[ihlij]+[ilhij]+[ihlji]+[ilhji],\\
[iihjl]+[iihlj]+[ihijl]+[ihilj]+[ijlih]+[iljih]+[ijlhi]+[iljhi],\\
[iiljh]+[iilhj]+[ilijh]+[ilihj]+[ijhil]+[ihjil]+[ijhli]+[ihjli],\\
[jiihl]+[jiilh]+[jhlii]+[jlhii],\\
[hiijl]+[hiilj]+[hjlii]+[hljii],\\
[liijh]+[liihj]+[ljhii]+[lhjii],\\
[jihil]+[jilih]+[jihli]+[jilhi]+[jhiil]+[jliih]+[jhili]+[jlihi],\\
[hijil]+[hilij]+[hijli]+[hilji]+[hjiil]+[hliij]+[hjili]+[hliji],\\
[lijih]+[lihij]+[lijhi]+[lihji]+[ljiih]+[lhiij]+[ljihi]+[lhiji],
\end{gathered}
$$

$$A = [ijhlm] + [ijhml] + [ihjlm] + [ihjml] + [ilmjh] + [imljh] + [ilmhj] + [imlhj]*,$$

Les 14 autres orbites de A peuvent chacune être déterminées par une permutation de l'orbite.

$$[ijlhm] + [ijlmh] + [iljhm] + [iljmh] + [ihmjl] + [imhjl] + [ihmlj] + [imhlj]*,$$
$$[ijmhl] + [ijmlh] + [imjhl] + [imjlh] + [ihljm] + [ilhjm] + [ihlmj] + [ilhmj]*,$$
$$[jihlm] + [jihml] + [jhilm] + [jhiml] + [jlmih] + [jmlih] + [jlmhi] + [jmlhi]*,$$
$$[jilhm] + [jilmh] + [jlihm] + [jlimh] + [jhmil] + [jmhil] + [jhmli] + [jmhli]*,$$
$$[jimhl] + [jimlh] + [jmihl] + [jmilh] + [jhlim] + [jlhim] + [jhlmi] + [jlhmi]*,$$
$$[hijlm] + [hijml] + [hjilm] + [hjiml] + [hlmij] + [hmlij] + [hlmji] + [hmlji]*,$$
$$[lijhm] + [lijmh] + [ljihm] + [ljimh] + [lhmij] + [lmhij] + [lhmji] + [lmhji]*,$$
$$[mijhl] + [mijlh] + [mjihl] + [mjilh] + [mhlij] + [mlhij] + [mhlji] + [mlhji]*,$$
$$[hiljm] + [hilmj] + [hlijm] + [hlimj] + [hjmil] + [hmjil] + [hjmli] + [hmjli]*,$$
$$[himjl] + [himlj] + [hmijl] + [hmilj] + [hjlim] + [hljim] + [hjlmi] + [hljmi]*,$$
$$[lihjm] + [lihmj] + [lhijm] + [lhimj] + [ljmih] + [lmjih] + [ljmhi] + [lmjhi]*,$$
$$[mihjl] + [mihlj] + [mhijl] + [mhilj] + [mjlih] + [mljih] + [mjlhi] + [mljhi]*,$$
$$[limjh] + [limhj] + [lmijh] + [lmihj] + [ljhim] + [lhjim] + [ljhmi] + [lhjmi]*,$$
$$[miljh] + [milhj] + [mlijh] + [mlihj] + [mjhil] + [mhjil] + [mjhli] + [mhjli]*.$$

Exemple 1.6.5. Soient L et H deux sous-groupes de S_8 définis par :
$H =< (1,2,6,7)(3,4,8,5), (1,4,7,5)(2,3,8,6), (2,7,8)(4,5,6), (2,5,6)(3,4,8), (3,6,7)(4,5,8), (3,4,7,8,6,5) >$ et le groupe L contenant H est défini par : $L =< (1,2,6,7)(3,4,8,5), (1,4,7,5)(2,3,8,6) >$. Nous adoptons la notation 1.6.1 et nous supposons que les entiers i et h sont distincts.

Le polynôme suivant est un H-Invariant L-Primitif ; il est un H-Invariant L-Primitif minimum pour $i = 0$ et $h = 1$:

$$[iiiihhhh] + [iiihihhh] + [iihhhhii] + [ihihihih] + [ihihhihi] + [ihhiihhi] + [ihhihiih] +$$
$$[hiihihhi] + [hiihhiih] + [hihiihih] + [hihihihi] + [hhiiiihh] + [hhiihhii] + [hhhhiiii] \ .$$

Le polynôme suivant est un H-Invariant S_8-Primitif minimum :

$$[00001111] + [00010111] + [00101101] + [00110011] + [00111010] + [00111100] + [01001110] +$$
$$[01010101] + [01011001] + [01011010] + [01100011] + [01100110] + [01101001] + [01110100] +$$
$$[10001011] + [10010110] + [10011001] + [10011100] + [10100101] + [10100110] + [10101010] +$$
$$[10110001] + [11000011] + [11000101] + [11001100] + [11010010] + [11101000] + [11110000] \ .$$

Ces deux derniers calculs ont été réalisés en trois minutes sur une machine PC Pentium Pro 200 Mhz avec 512 Mo de mémoire RAM.

1.6.2 Coût de l'algorithme `Girstmair-Jordan`

Le temps de calcul et la mémoire utilisée pour le calcul d'invariants primitifs de degrés petits serons données en détail dans le chapitre 2.

Les temps nécessaire au calcul d'ensembles essentiels avec l'algorithme 1.4.3 se fait une fois pour toute. Les résultats son donnés sous forme de tableau dans le chapitre suivant. Le temps et la mémoire augmentent d'une façon raisonnable avec le degré n ; le tableau 1.1 marque en moyenne cette évolution pour n allant de 5 à 9 (sur une machine PC Pentium

Pro 500 Mhz avec 640 Mo de mémoire RAM).

Degré	Temps (CPU)	Mémoire (Mo)
5	10^{-3} s	$<< 74$
6	40 s	< 74
7	10 mn	74
8	50 mn	74
9	3 h	74

TAB. 1.1 – *Temps de calculs de l'algorithme 1.4.3*

La mémoire utilisée est stable lors du calcul d'ensembles essentiels pour n fixe. La mémoire augmente progressivement, mais d'une manière raisonnable. D'autre part, le temps de calcul augmente d'une façon plus importante avec n. Dans la plupart des cas, le calcul des invariants primitifs relatifs est plus rapide que le calcul d'invariants primitifs absolues, ce qui est une bonne nouvelle. Rappelons enfin que même si le calcul d'ensembles essentiels est coûteux en temps, il ne sera fait qu'une seule fois.

1.7 Conclusion

Les méthodes de calculs des polynômes invariants primitifs connus sont intuitifs et donnent souvent des polynômes de degrés élevés. L'algorithme de K. Girstmair calcule un invariant primitif absolu de degré minimum, mais pas nécessairement minimum. Le type de l'orbite d'un monôme ne dépend que de la partition définie par le multidegré, et donc le calcul doit se faire en utilisant ce codage, plutôt que les monômes proprement dits (qui sont d'utilisation malaisée dans les systèmes de calcul formel). Le module « `PrimitiveInvariant` » qui reprends les algorithmes présentés dans ce chapitre (voir Annexe A) détermine tous les invariants primitifs réduits relatifs et absolus et aussi ceux de degré minimum. La représentation de données utilisée dans « `PrimitiveInvariant` » peut se généraliser au calcul de polynômes invariants vérifiant d'autres propriétés par exemple dans le calcul des *invariants classiques* et autres types d'invariants utiles en théorie de Galois (voir chapitre 3).

Nous comparons dans le chapitre suivant le module « `PrimitiveInvariant` » avec une méthode classique de calcul d'invariants primitifs qui fait appel à des méthodes plus algébrique qui découlent des travaux de Hilbert à la fin du siècle dernier. La structure à décrire est celle de l'espace des invariants en tant que module libre sur un anneau de polynômes dont les générateurs sont les *invariants fondamentaux*.

Chapitre 2

Invariants fondamentaux

Sommaire

Il s'agit dans ce chapitre de décrire et d'analyser certaines méthodes pour le calcul de polynômes invariants. L'objectif recherché est de dégager le meilleur algorithme possible, c'est à dire le plus rapide théoriquement et pratiquement pour le calcul d'invariants primitifs. Les premiers algorithmes connus de constructions de *base d'anneaux d'invariants* sont donnés par D. Hilbert [39] et H. Weber [79]. L'idée est de trouver une famille de polynômes invariants par un groupe H qui, lorsqu'ils s'annulent, annulent tous les H-invariants. En fait, on montre qu'il existe de telles familles qui sont finies.
Par exemple, pour $n = 3$ et $H = S_2$, les polynômes puissances symétriques $x_1, x_2+x_3, x_2^2+x_3^2$ forment une famille de générateurs de tous les S_2-invariants de $k[x_1, x_2, x_3]$:

$$k[x_1, x_2, x_3]^{S_2} = k[x_1, x_2 + x_3, x_2^2 + x_3^2] \quad .$$

Lorsque le groupe H est quelconque, la détermination de tous les polynômes H-invariants est plus difficile et l'automatisation des méthodes devient nécessaire. Un algorithme de calcul de l'anneau des H-invariants est donnée dans le livre de B. Sturmfels [69]. Il résume les avancées faites dans ce domaine et propose de calculer une famille de polynômes invariants de degrés petits en calculant des invariants *primaires* et *secondaires* de degrés 1, 2, 3, ...
Un autre algorithme proposé par E. Dade construit des invariants en considérant des

produits d'orbites de formes linéaires particulières. Cet algorithme est très rapide mais calcule des invariants primaires de degrés élevés et, le lien qui existe entre les degrés de ces invariants et le nombre d'invariants secondaires impose un trop grand nombre de générateurs. M. Göbel [34] propose quand à lui un très bon algorithme qui calcule des générateurs d'anneaux d'invariants associés à des groupes de permutations.
Viennent enfin les travaux de G. Kemper [44] sur les groupes linéaires finis qui permettent de trouver des invariants primaires qui minimisent souvent le nombre d'invariants secondaires. Cette méthode utilise les résultats sur les *opérateurs de Reynold*, la *formule de Molien* et la *propriété de Cohen-Macaulay*. Ces travaux sont implantés en MAPLE et MAGMA sous la librairie `invar` (voir [45] et l'amélioration dans [46]) et ils nécessitent le calcul d'idéaux premiers et des bases de Gröbner.

Dans la section 2.1.1 sont introduit les invariants primaires et secondaires, appelés aussi *invariants fondamentaux*. Dans la section 2.1.2 est présenté l'algorithme de calcul des invariants primaires donné par G. Kemper et utilisé dans « `invar` ». La section 2.1.3 explique comment utiliser le module « `invar` » pour calculer des invariants primitifs et, dans la section 2.2 « `invar` » est comparé avec le module « `PrimitiveInvariant` » de GAP implantant les algorithmes du chapitre 1.

2.1 Calcul des invariants fondamentaux par G. Kemper

Soit H un sous-groupe fini du groupe algébrique linéaire $GL_n(k)$ de degré n sur k et $|H|$ son cardinal. Pour tout polynôme P de $k[x_1,\ldots,x_n]$, $deg(P)$ désignera le degré total de P en $x_1,\ldots,x_n$ et $R^H = k[x_1,\ldots x_n]^H$ l'anneau des polynômes H-invariants de $k[x_1,\ldots x_n]$. Le théorème sur la structure de l'anneau R^H est assez récent cf. l'article fondamental de R.P. Stanley ([65]) et les travaux de Hochster et Eagon ne faisant appel qu'à des notions classiques relativement simples de l'algèbre commutative sans algèbre homologique). Nous présentons dans ce qui suit les principales propriétés de l'anneau R^H.

2.1.1 Invariants fondamentaux

L'anneau R^H des polynômes invariants sous l'action de H est une algèbre de *Cohen-Macaulay* sur k, de *dimension de Krull* n. En d'autres termes, il existe une famille $\Pi_1,\ldots,\Pi_n$ de polynômes homogènes de R^H, algébriquement indépendants sur k, tel que R^H soit un module de type fini libre sur $k[\Pi_1,\ldots,\Pi_n]$.
Nous pouvons toujours trouver une $k[\Pi_1,\ldots,\Pi_n]$-base de ce module formée de polynômes homogènes, dont l'un d'entre eux est 1. Sa dimension est alors égale à :

$$\frac{\prod_{i=1}^{i=n} deg(\Pi_i)}{|H|} \quad .$$

Toute suite $(\Pi_1,\ldots,\Pi_n)$ de polynômes telle que R^H soit un module de type fini libre sur $k[\Pi_1,\ldots,\Pi_n]$ est appelée une suite d'*invariants primaires* de H. Une base $\Sigma_1,\ldots,\Sigma_e$

de $k[x_1,\ldots,x_n]^H$ comme module sur $k[\Pi_1,\ldots,\Pi_n]$ est appelée famille d'*invariants secondaires* de H. L'anneau R^H peut alors s'écrire sous la forme:

$$R^H = \bigoplus_{i=1}^{i=e} k[\Pi_1,\ldots,\Pi_n]\,\Sigma_i \tag{2.1}$$

appelée la *décomposition d'Hironaka* de $k[x_1,\ldots,x_n]^H$.

Exemple 2.1.1. Rappelons que S_n désigne le groupe des permutations de degré n. L'anneau $k[x_1,\ldots,x_n]$ est un $k[x_1,\ldots,x_n]^{S_n}$-module libre de dimension $n!$ et admet pour base la famille des monômes $\prod_{j=1}^{j=n} x_j^{i_j}$ tels que pour tout j, $0 \le i_j \le j-1$. Ainsi:

$$k[x_1,\ldots,x_n] \;=\; \bigoplus_{j\in[1,n],0\le i_j\le j-1} k[x_1,\ldots,x_n]^{S_n} x_1^{i_1}\ldots x_n^{i_n} \qquad .$$

Exemple 2.1.2. Les polynômes symétriques élémentaires sont des invariants primaires de l'anneau des polynômes symétriques $k[x_1,\ldots,x_n]^{S_n}$. Si $H \subset S_n$ alors $k[x_1,\ldots,x_n]^H$ est un module libre sur $k[x_1,\ldots,x_n]^{S_n}$ de dimension l'indice de H dans S_n. Pour le sous-groupe H de S_4 engendré par l'identité et la permutation $(2,3,4)$, nous avons:

$$k[x_1,\ldots,x_4]^H = k[\Pi_1,\ldots,\Pi_4]\,\Sigma_1 \bigoplus k[\Pi_1,\ldots,\Pi_4]\,\Sigma_2 \qquad ,$$

où $\Pi_1 = x_1$, $\Pi_2 = x_2+x_3+x_4$, $\Pi_3 = x_2^2+x_3^2+x_4^2$, $\Pi_4 = x_2^3+x_3^3+x_4^3$ et $\Sigma_1 = 1$, $\Sigma_2 = x_2^2x_3 + x_2x_4^2 + x_3^2x_4$.
L'anneau $k[x_1,\ldots,x_n]^H$ est la clôture intégrale de $k[x_1,\ldots,x_n]^{S_n}$ dans le corps des H-invariants du corps $k(x_1,\ldots,x_n)$ des fractions rationnelles à coefficients dans k en les indéterminées $(x_1,\ldots,x_n)$. Le fait que la dimension du $k[x_1,\ldots,x_n]^{S_n}$-module $k[x_1,\ldots,x_n]^H$ est $e = [S_n : H]$ découle alors du fait que $k(x_1,\ldots,x_n)$ est une extension de degré e du corps $k(x_1,\ldots,x_n)^{S_n}$ des fractions rationnelles *symétriques* à coefficients dans k en $(x_1,\ldots,x_n)$.

Soit la décomposition graduée:

$$R^H = \bigoplus_{d=0}^{\infty} R_d^H$$

où R_d^H désigne l'espace vectoriel des H-invariants de degré d (de dimension $C_{n+d-1}^{n-1} = \frac{(n+d-1)!}{(n-1)!d!}$).

Définition 2.1.3. La série de Hilbert de R^H est définie par:

$$\mathcal{H}(R^H,t) = \sum_{d=0}^{\infty} dim_k(R_d^H)\; t^d$$

Nous n'expliquerons pas dans les détails comment peuvent être obtenues des familles d'invariants fondamentaux, mais nous allons plutôt donner l'idée générale du calcul de ces invariants (voir algorithme 2.1.1) et utiliser ces résultats pour le calcul d'invariants primitifs.

2.1.2 Invariants primaires

Soit H un sous-groupe fini de $GL_n(k)$. Les degrés des invariants primaires choisis déterminent le nombre d'invariants fondamentaux de l'anneau des invariants R^H. En effet, la formule de Molien [69] nous donne une information complète sur le nombre des invariants secondaires en fonction de ceux des invariants primaires. ce nombre croit avec les degrés des invariants primaires. Nous renvoyons le lecteur à [44] et [72] pour le calcul des invariants secondaires et nous décrivons dans cette section la méthode utilisée par G. Kemper pour le calcul des invariants primaires.

G. Kemper propose un bon algorithme dans le sens où il calcule dans la plupart des cas, des invariants primaires de degrés minimaux. Pour cela, il utilise les *d'idéaux premiers* associés à un ensemble fini de polynôme. Nous présentons ci-après la proposition sur laquelle se base le calcul des invariants primaires et, pour plus de détails sur la construction des idéaux premiers, le lecteur pourra consulter [28].

Proposition 2.1.4 (Kemper). *Notons $R^H = k[x_1, \ldots, x_n]^H$ l'anneau des polynômes H-invariants et soit $\Pi_1, \ldots, \Pi_i$ une famille de polynômes homogènes H-invariants. Cette famille peut-être étendue à un système d'invariants primaires de l'anneau des H-invariants $k[x_1, \ldots, x_n]^H$ si et seulement si*

$$dim_k(\Pi_1, \ldots, \Pi_i) = n - i \qquad .$$

En particulier, $\Pi_1, \ldots, \Pi_i$ forment un système d'invariants primaires de $k[x_1, \ldots, x_n]^H$ si et seulement si $i = n$ et l'ensemble des zéros du système $\{\Pi_1 = 0, \ldots, \Pi_i = 0\}$ sur une clôture algébrique de k est réduit à zéro.

Preuve. Voir [44]. □

La dimension $dim_k(\Pi_1, \ldots, \Pi_i)$ est une dimension de variété algébrique, c'est-à-dire la dimension de la variété algébrique sur une clôture algébrique $\hat{k}$ de k définit par l'idéal de $k[x_1, \ldots, x_n]$ engendré par la suite $(\Pi_1, \ldots, \Pi_i)$.

Algorithme 2.1.1 (Invariants Primaires de « `invar` »). *Cet algorithme calcule une famille d'invariants primaires de degré petit.*

`Entrée :` `Générateurs d'un groupe fini` H.
`Sortie :` $\{\Pi_1, \ldots, \Pi_n\}$ `invariants primaires de` $k[x_1, \ldots, x_n]^H$.

`1.` $P_1 := \{\}$; $r := 1$;
$d := 1$ *le degré de l'invariant primaire à calculer* ;
$i := 1$ *le nombre d'invariants primaires* ;
`2.` `Tant que` $i \leq n$ `Faire`

```
3.          Soit π_d := Σ_{j=1}^{j=m_d} t_j b_j où les t_j sont des inconnues et les
            b_j sont les monômes unitaires H-invariants de degré d ;
4.          Pour j allant de 1 à r Faire
                    Calculer les restes résiduels a_j(t_1,...,t_{m_d})
                    de π_d par rapport à P_j ;
            Fin Pour ;
5.          Si ∃α_1,...,α_{m_d} tel que a_j(α_1,...,α_{m_d}) ≠ 0 ∀j Alors
                    Π_i = π_d(α_1,...,α_{m_d})
                    P_1,...,P_r bases de Gröbner des idéaux premiers
                    associés à {Π_1,...,Π_i} ;
                    i := i+1 ;
            Sinon d := d+1 ;
            Fin Si ;
      Fin Tant que ;
      Retourner({Π_1,...,Π_n}) ;
Fin.
```

Preuve. Nous calculons dans la partie `3.` de l'algorithme une forme générale d'un H-invariant homogène de degré d. Cet invariant contient tous les monômes H-invariants de degré d.
Dans la partie `4.` le calcul des restes résiduels a_j consiste à écrire le polynôme π_d dans la base P_j pour j allant de 1 à r. Ensuite nous recherchons, grâce aux bases de Gröbner, un polynôme H-invariant de degré d n'appartenant pas aux idéaux premiers associés à la famille d'invariants primaires déjà calculée.
La partie `5.` est la plus coûteuse. Dans son implantation, G. Kemper utilise des résultats donnés par P. Gianni [31] et T. Becker [9]. □

2.1.3 Calculs d'invariants primitifs à partir d'invariants fondamentaux

Lemme 2.1.5. *Soient H et L deux sous-groupes du groupe symétrique de degré n. Si $H \subset L$, alors le polynôme $\Theta = \sum_{\sigma \in H} \sigma.(x_1 x_2^2 \dots x_n^n)$ vérifie $Stab_{S_n}(\Theta) = H$.*

Preuve. Aucune hypothèse sur la caractéristique de k n'est nécessaire. Il est clair que $Stab_{S_n}(x_1 x_2^2 \dots x_n^n) = \{Id\}$, donc nous avons $\Theta = N_H(x_1 x_2^2 \dots x_n^n)$. Nous savons déjà que Θ est H-invariant. Si $\sigma \in S_n$ vérifie $\sigma.\Theta = \Theta$, du fait que $Stab_{S_n}(x_1 x_2^2 \dots x_n^n) = \{Id\}$, nous voyons qu'il existe $\phi \in S_H$ telle que $\phi(s) = \sigma s$ pour tout $s \in H$. En particulier, nous avons $\sigma = \sigma Id = \phi(Id) \in H$, d'où $Stab_{S_n}(\Theta) = H$. □

Soient H et L deux groupes finis tels que $H \subset L$ et e l'indice de H dans L. D'après le lemme 2.1.5, il existe toujours un polynôme H-invariant L-primitif (voir définition 1.1.2). En pratique, il n'est pas très intéressant d'utiliser ce polynôme car son degré est élevé et nous avons vu dans le chapitre 1 comment calculer tous les invariants primitifs de degré

petit. Il existe d'autres méthodes de calculs d'invariants primitifs (voir [23] qui se base sur les groupes par exemple), mais nous présentons dans ce qui suit une méthode de calcul des invariants primitifs se basant sur les invariants fondamentaux.

Test pour la primitivité des invariants

Tout polynôme H-invariant s'écrit sous la forme d'une combinaison linéaire d'invariants fondamentaux associés à l'anneau R^H. D'après l'égalité (2.1), un polynôme $P \in R^H$ s'écrit sous la forme :

$$P = \sum_{i=1}^{i=e} f_i(\Pi_1, \ldots, \Pi_n)\ \Sigma_i\ ,$$

où les f_i sont des polynômes en n variables et à coefficients dans k. Nous présentons dans cette section une méthode utilisée par A. Colin ([20] et [21]) pour tester si un polynôme H-invariant est L-relatif.
Notons T une transversale à gauche de L modulo H. Rappelons qu'un polynôme P est dit H-invariant L-primitif si P est invariant sous l'action de H et si :

$$\forall\ \sigma, \sigma^{'} \in T\ ,\ \sigma \neq \sigma^{'} \Longrightarrow \sigma.P \neq \sigma^{'}.P \qquad .$$

En d'autres termes, un polynôme H-invariant est dit H-invariant L-primitif si le discriminant $Disc$ du polynôme $T_P(X) = \prod_{\sigma \in T}(X - \sigma.P)$ est non nul :

$$Disc(T_P(X)) = \prod\nolimits_{\sigma,\sigma' \in T \text{ et } \sigma \neq \sigma'} (\sigma.P - \sigma^{'}.P) \neq 0 \qquad . \tag{2.2}$$

Au lieu de manipuler des polynômes génériques ce qui est plutôt coûteux sur machine, nous testons (2.2) sur des spécialisations du discriminant de $T_P(X)$ en des valeurs numériques. En fait, dès qu'une spécialisation de $Disc(T_P(X))$ est non nulle, $Disc(T_P(X))$ est non nul et P est un H-invariant L-primitif. Si au bout d'un certain nombre de spécialisations de $Disc(T_P(X))$ en des valeurs numériques nous trouvons toujours zéro, alors nous recommençons les tests avec un nouveau polynôme P appartenant à l'anneau R^H.

Exemple 2.1.6. Soient $n = 7$ et $H = <(1,6)(2,4)(3,5), (1,5), (2,3)(4,5), (1,4,5)(2,6,3), (1,4,5)(2,3,6)>$ un groupe d'ordre 72 dans S_7.

$$R^H = \bigoplus\nolimits_{i=1}^{i=4} k[\Pi_1, \ldots, \Pi_7]\Sigma_i\ ,$$

avec les invariants secondaires définis par :

$$\begin{aligned}
\Sigma_1 =& 1\ ,\\
\Sigma_2 =& x_1^2x_2 + x_1^2x_3 + x_1^2x_6 + x_1x_2^2 + x_1x_3^2 + x_1x_6^2 + x_2^2x_4 + x_2^2x_5 + x_2x_4^2 +\\
& x_2x_5^2 + x_3^2x_4 + x_3^2x_5 + x_3x_4^2 + x_3x_5^2 + x_4^2x_6 + x_4x_6^2 + x_5^2x_6 + x_5x_6^2\ ,\\
\Sigma_3 =& x_1^3x_2 + x_1^3x_3 + x_1^3x_6 + x_1x_2^3 + x_1x_3^3 + x_1x_6^3 + x_2^3x_4 + x_2^3x_5 + x_2x_4^3 +\\
& x_2x_5^3 + x_3^3x_4 + x_3^3x_5 + x_3x_4^3 + x_3x_5^3 + x_4^3x_6 + x_4x_6^3 + x_5^3x_6 + x_5x_6^3\ ,\\
\Sigma_4 =& x_1^5 + x_2^5 + x_3^5 + x_4^5 + x_5^5 + x_6^5\ ,
\end{aligned}$$

et les invariants primaires définis par :

$$\begin{aligned}
\Pi_1 &= x_1+x_2+x_3+x_4+x_5+x_6\ ,\\
\Pi_2 &= x_7\ ,\\
\Pi_3 &= x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2\ ,\\
\Pi_4 &= x_1x_2+x_1x_3+x_1x_6+x_2x_4+\\
&\quad x_2x_5+x_3x_4+x_3x_5+x_4x_6+\\
&\quad x_5x_6\ ,\\
\Pi_5 &= x_1^3+x_2^3+x_3^3+x_4^3+x_5^3+x_6^3\ ,\\
\Pi_6 &= x_1^4+x_2^4+x_3^4+x_4^4+x_5^4+x_6^4\ ,\\
\Pi_7 &= x_1^6+x_2^6+x_3^6+x_4^6+x_5^6+x_6^6\ .
\end{aligned}$$

Posons $L = <(2,3)(5,6),(1,6,3),(1,2,4)(3,5,6),(5,6)>$ un sous-groupe de S_7 contenant H. L'indice de H dans L est égal à 10. Le polynôme $T_P(X)$ est alors de degré 10 et nous trouvons grâce au test de primitivité (2.2) que le polynôme $P = \Pi_2\Pi_4\Sigma_1$ est un invariant L-primitif :

$$P = x_7(x_6+x_3+x_2)(x_5+x_4+x_1) \qquad .$$

Remarquons que le polynôme P est donné directement par « `PrimitiveInvariant` » et que ce module donne aussi le polynôme H-invariant L-primitif minimal qui est égal à :

$$x_4x_5+x_3x_6+x_2x_6+x_2x_3+x_1x_5+x_1x_4 \qquad .$$

Calcul d'Invariants Primitifs de degré minimal

Rappelons qu'un polynôme H-invariant L-primitif est dit de degré minimal, s'il n'existe pas d'autres H-invariant L-primitif de degré plus petit que lui. Pour avoir un polynôme invariant primitif de degré minimal, K. Gleissler et J. Klüners ont proposé dans [30] une méthode qui se base sur la propriété suivante : si H est un sous-groupe maximal de L, alors le degré minimal d des H-invariants L-primitifs est défini par :

$$d = min\{i \mid \mathrm{coeff}_i(\mathcal{H}(R^H,t)) \neq \mathrm{coeff}_i(\mathcal{H}(R^L,t))\} \qquad , \tag{2.3}$$

où $\mathrm{coeff}_i(\mathcal{H}(R^H,t))$ est le coefficient de t^i dans $\mathcal{H}(R^H,t)$ (idem pour $\mathrm{coeff}_i(\mathcal{H}(R^L,t))$). Ci-dessous, l'algorithme qui nous permet de calculer un polynôme invariant primitif de degré minimal.

Algorithme 2.1.2 (InvariantsPrimitifs). *La première étape de cet algorithme utilise les résultats de [46] :*

`Entrée :` `Deux groupes` H `et` L `tels que` H `sous-groupe maximal de` L.
`Sortie :` `Un polynôme homogène` P `de degré` $d \leq \frac{n(n-1)}{2}$ `tel que` $Stab_L(P) = H$.

```
1.    Calculer les séries de Hilbert H(R^H,t) et H(R^L,t).
      Déterminer d le plus petit des indices de H(R^H,t) et
      H(R^L,t) dont les coefficients associés sont distincts ;
2.    Calculer tous les H-invariants homogènes de degré d.
      Ne garder que ceux d'entre-eux qui sont L-relatifs.

3.    Retourner le polynôme qui contient le plus petit nombre
      de monômes parmi ceux de l'étape 2.

Fin.
```

En combinant les résultats des tests de primitivité (2.2) avec la propriété donnée par (2.3), nous pouvons déterminer des invariants primitifs de degré minimal.

2.2 Comparaison entre « `invar` » et « `PrimitiveInvariant` »

Rappelons que le module « `PrimitiveInvariant` » comprends les algorithmes présentés au chapitre 1 pour le calcul d'invariants primitifs de degré minimal par rapport à des groupes de permutations (voir Annexe A). Nous comparons dans cette section ce module avec la méthode découlant de l'algorithme de G. Kemper (voir algorithme 2.1.1 et la section 2.1.3).
Pour le calcul d'invariants primitifs de groupes de permutations, nous allons utiliser le module « `invar` » de G. Kemper implanté en MAGMA qui est un bon outil pour le calcul d'invariants fondamentaux.
Tous les calculs sont fait sur une machine *Compaq Ultimate de 2 x 533 Mhz et 2048 Mo.* Les comparaisons entre les deux modules sont donnés sous forme de figures où les temps (en ordonnées) sont donnés en secondes et les groupes (abscisses) sont donnés par leur numérotation dans la liste des sous-groupes de S_n (voir exemple de groupes à l'annexe B).

Nous commençons par comparer entre eux les deux logiciels de calcul formel MAGMA et GAP. Le tableau 2.1 nous donne le temps de calcul des orbites de polynômes à n variables sous l'action de sous-groupes du groupes symétriques S_n en GAP et en MAGMA et illustre bien la principale différence en temps d'exécutions des deux logiciels. En effet, il apparaît que MAGMA est plus performant que GAP.

En effectuant plusieurs calculs, nous avons remarqué que le temps nécessaire à la vérification de la primitivité d'un invariant (voir section 2.1.3) donné par « `invar` » est négligeable devant le temps de calcul des invariants fondamentaux. De ce fait, seul les temps propres à l'exécution des commandes de calcul d'invariants primaires et secondaires sont pris en compte.

n	GAP	MAGMA
4	0	0
5	1	0.5
6	6	1
7	11	2
8	20	4
9	24	6

TAB. 2.1 – *Comparaison entre* GAP *et* MAGMA

2.2.1 Résultats expérimentaux de S_4 et S_6

Nous avons réalisé les calculs d'invariants fondamentaux et d'invariants primitifs de tous les sous-groupes de S_4. Les temps de calculs sont extrêmement rapides (moins de 10^{-2} secondes pour chaque groupe).
La mémoire utilisée par les deux modules « `PrimitiveInvariant` » et « `invar` » est négligeable.

Le calcul le plus coûteux pour les deux modules a été la détermination des invariants du groupe alterné A_4 de S_4. Le module « `PrimitiveInvariant` » nous donne :
$x_2x_4^2x_3^3+x_3x_2^2x_4^3+x_4x_3^2x_2^3+x_1x_3^2x_4^3+x_3x_4^2x_1^3+x_4x_1^2x_3^3+x_1x_4^2x_2^3+x_2x_1^2x_4^3+x_4x_2^2x_1^3+x_1x_2^2x_3^3+x_2x_3^2x_1^3+x_3x_1^2x_2^3$ et son conjugué Σ_2. Le module « `invar` » nous donne :

$$\begin{aligned}\Pi_1 =& x_1+x_2+x_3+x_4,\\ \Pi_2 =& x_1^2+x_2^2+x_3^2+x_4^2,\\ \Pi_3 =& x_1^3+x_2^3+x_3^3+x_4^3,\\ \Pi_4 =& x_1^4+x_2^4+x_3^4+x_4^4\end{aligned}$$

$\Sigma_1 = 1$ et $\Sigma_2 = x_1^3x_2^2x_3+x_1^3x_2x_4^2+x_1^3x_3^2x_4+x_1^2x_2^3x_4+x_1^2x_2x_3^3+x_1^2x_3x_4^3+x_1x_2^3x_3^2+x_1x_2^2x_4^3+x_1x_3^3x_4^2+x_2^3x_3x_4^2+x_2^2x_3^3x_4+x_2x_3^2x_4^3$.
Le polynôme Σ_2 donné directement par « `invar` » est donc un invariant primitif. D'autre part, nous remarquons que les temps de calculs effectués sur les sous-groupes de S_6 pour déterminer des invariants primitifs par « `PrimitiveInvariant` » ou « `invar` » sont sensiblement identiques (l'utilisation de la mémoire reste raisonnable dans les deux cas). La figure 2.1 résume les variations de temps de calculs des invariants primitifs absolues de S_6 et la figure 2.2 nous donne une idée sur les temps de calculs pour des invariants primitifs relatifs.
En tenant compte du fait que les modules soient implantés dans des systèmes de Calcul formel différents et que le temps d'exécution d'une même opération est plus rapide sur MAGMA que sur GAP(voir tableau 2.1), nous conseillons pour $n = 6$, le module « `PrimitiveInvariant` » pour la détermination des invariants primitifs relatifs.

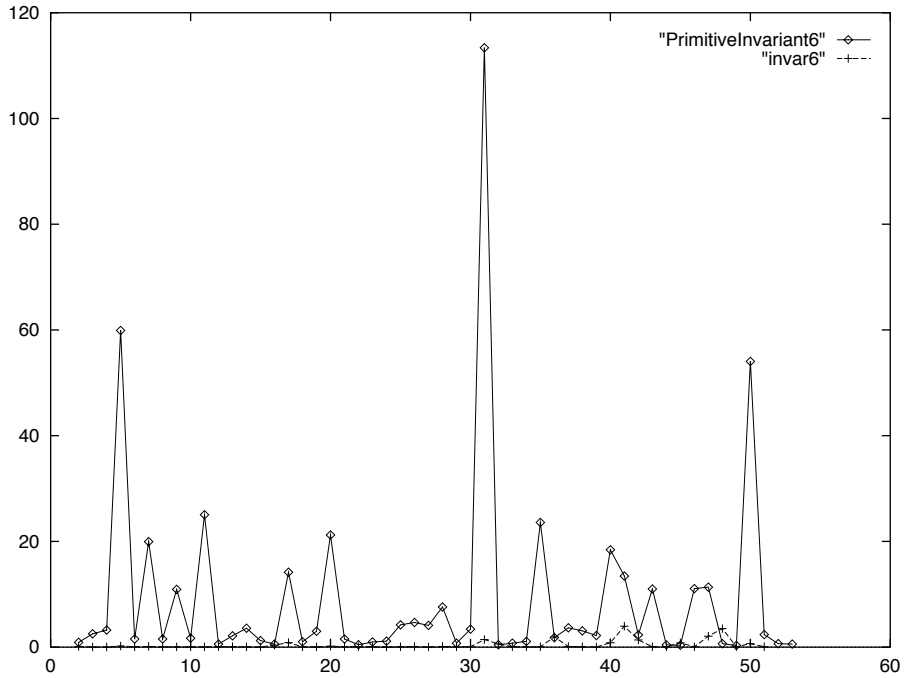

FIG. 2.1 – *Invariants primitifs absolus de* S_6

2.2.2 Résultats expérimentaux pour les sous-groupes de S_8

Pour des degrés plus élevés ($n = 8$), le module « `PrimitiveInvariant` » utilise en moyenne 70 Mo de mémoire alors que la variation de la taille de la mémoire utilisée par le module « `invar` » est beaucoup plus importante (en moyenne égale à 200Mo et atteint pour certains groupes 1Go sans raison apparente). Les temps de calculs aussi diffèrent. La figure 2.3 donne une idée sur la variation du temps de calculs des deux modules. Selon les groupes, il vaut mieux utiliser l'une ou l'autre méthode de calcul d'invariants primitifs. Mais nous remarquons par exemple que les deux algorithmes réagissent pratiquement de la même manière pour le groupe alternée (sur la mémoire utilisée et les temps de calculs d'invariants primitifs absolus).

Rappelons que d'une façon générale, les invariants fondamentaux que nous obtenons grâce à « `invar` » ne sont pas nécessairement des invariants primitifs. C'est ce qui nous amène à faire le test de primitivité de la section 2.1.3. Rappelons aussi que les invariants primitifs donnés par la manipulation des invariants fondamentaux ne sont pas nécessairement minimaux (cas possible lorsque H est sous-groupe maximal de L).

Remarquons enfin que pour certains groupes, les temps de calculs d'invariants primitifs absolus avec « `invar` » sont très supérieurs au temps de « `PrimitiveInvariant` ». Le tableau 2.2 nous donne en secondes la comparaison du temps de calculs de certains sous-groupes de S_8.

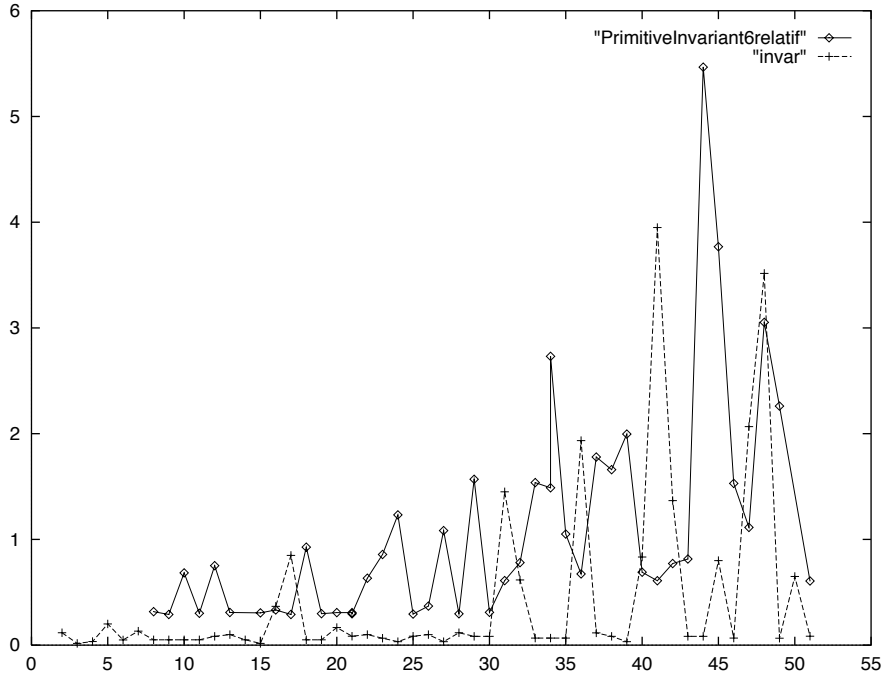

FIG. 2.2 – *Invariants primitifs relatifs de* S_6

Numéro	Groupe associé	« `invar` » en s.	« `PrimitiveInvariant` » en s.
43	$<(5,6)(7,8),(1,2)(3,4),(1,3,2,4)(5,7,6,8)>$	2498.333	2483.263
48	$<(5,6)(7,8),(5,7,6,8),(1,2)(3,4)>$	2656.267	1741.331
64	$<(7,8),(5,6),(3,4)(5,7)(6,8)>$	17902.683	360.731
81	$<(6,7,8),(2,3)(4,5),(2,4)(3,5)(7,8)>$	810.533	11148.599
110	$<(5,6)(7,8),(5,7)(6,8),(3,4)(7,8),(1,2)(7,8)>$	6629.984	6864.969
125	$<(7,8),(3,4)(5,6),(3,5)(4,6),(1,2)(5,6)>$	2546.833	1306.346
128	$<(6,7,8),(3,4,5),(1,2)(4,5)(7,8)>$	683.117	4756.493
133	$<(6,7,8),(4,5)(7,8),(1,2,3)>$	89256.916	6032.186
136	$<(7,8),(2,3,4,5,6),(3,6)(4,5)>$	8493.867	64.303
141	$<(6,7,8),(7,8),(2,3)(4,5),(2,4)(3,5)>$	70045.75	60.308

TAB. 2.2 – *Comparaison entre* « `invar` » *et* « `PrimitiveInvariant` »

Nous avons noté que, d'une façon générale, il n'y a pas une grande variation du temps de calcul des invariants primitifs par « `PrimitiveInvariant` », contrairement à « `invar` » (voir également les tableaux de l'annexe B.2).

2.3 Conclusion

La connaissance des invariants fondamentaux nous donne une caractérisation de l'anneau des invariants par une base de polynômes (alors que les invariants primitifs caractérisent des groupes). En adaptant les résultats de « `invar` », nous avons vu comment déterminer des invariants primitifs à partir des invariants fondamentaux (méthode utilisée notamment par Klüners pour déterminer des invariants primitifs relatifs) et nous avons comparé cette méthode avec le module « `PrimitiveInvariant` ». L'efficacité des algorithmes de ce cha-

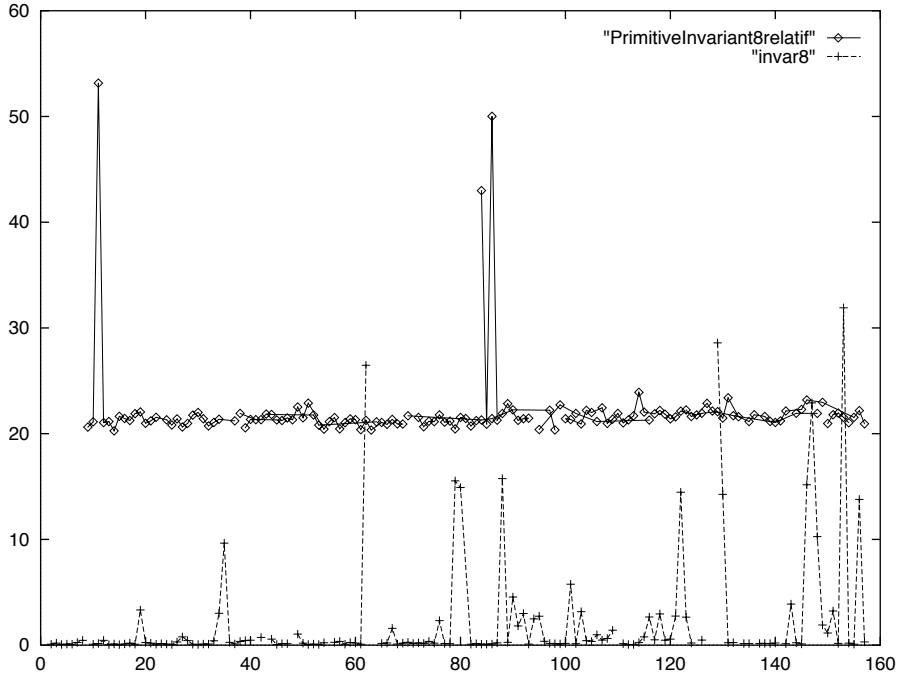

FIG. 2.3 – *Invariants primitifs relatifs de* S_8

pitre repose pour partie sur l'efficacité de l'implémentation des bases de Gröbner, ce qui conditionne le choix d'un système de calcul formel plutôt qu'un autre ; « `invar` » qui est le principal module utilisée pour la détermination des invariants primitifs est implémenté en MAGMA. Le module « `PrimitiveInvariant` » est implémenté dans GAP et le choix de ce système de calcul formel est conditionné par la richesse de la librairie des groupes en GAP. Les calculs ont montré, et ce malgré la différence entres les système de calcul formel MAGMA et GAP, que l'utilisation de « `PrimitiveInvariant` » est la plus intéressante, par rapport à « `invar` », pour la détermination des invariants primitifs relatifs. En effet, les invariants primitifs relatifs sont obtenus par « `PrimitiveInvariant` » en un temps raisonnable comparé au temps mis par « `invar` » pour calculer les invariants fondamentaux d'une part et tester parmi les combinaisons d'invariants primaires et secondaires ceux qui sont invariants primitifs d'autre part. Le module « `PrimitiveInvariant` » a un meilleur contrôle de la mémoire et du temps utilisés contrairement à « `invar` » qui explose (en temps et en mémoire) pour certains groupes comme nous le montre le tableau 2.2.

Par contre, les calculs montrent que la détermination d'invariants primitifs absolus est plus rapide avec « `invar` » que « `PrimitiveInvariant` » même si les tailles de calculs peuvent atteindre plusieurs centaines de Méga-octets par « `invar` » pour certains groupes (nous avons entamé une implémentation de « `PrimitiveInvariant` » en MAGMA pour avoir une idée plus claire sur la différence de temps de calculs). Ces deux modules sont complémentaires.

Parce que chaque système de calcul formel a sa spécificité, ses points forts et ses faiblesses

(GAP n'est par exemple pas encore bien développé pour la manipulation de polynômes : des erreurs sont courantes lors de calculs de stabilisateurs de polynômes et autres opérations de base), il serait préférable de développer des « passerelles » entre les divers systèmes de calcul formel quitte à avoir des versions moins efficaces, mais plus complètes et mieux structurées.

Deuxième partie

Les Invariants classiques dits Anciens

Chapitre 3

Covariants et Invariants classiques

Sommaire

Le principal objet de la théorie des « invariants classiques » est l'étude des propriétés géométriques des équations (voir par exemple [22]). Il s'agit dans ce chapitre, de rappeler les principaux résultats sur les *covariants* et, au niveau algorithmique, de mettre en évidence les invariants classiques et le lien entre ces polynômes et les invariants de groupes : les *polynômes-différences symétrisés* (le point de vue de la théorie des groupes est développé dans [41]). La littérature sur ce sujet est abondante ; en effet, la théorie des invariants est considérée comme un leitmotiv de l'Algèbre classique et moderne depuis au moins 1850. Parmi les anciens « invariantistes », nous pouvons citer P. Gordan [35], F. Klein, Fricke, Netto, Clebcsh, Aronhold, Cayley, Burnside [14], Salmon [60], Dickson (qui a écrit sur la théorie arithmétique des invariants)... Parmi les « invariantistes » modernes , citons Nagata, Fogarty, Gurevitch, D. Mumford [57], Hochster et toute l'école des groupes algébriques gravitant autour de Borel... et entre les deux, les grands du vingtième siècle : Young [83], Grace, Elliott, et bien sûr H. Weyl [80].

Une des principales difficultés rencontrée dans la théorie des invariants classiques réside dans les notations utilisées. Nous avons adopté les notations dites *symboliques* pour déterminer ces invariants classiques (voir [47]). Nous proposons dans un premier temps de

définir la notion de covariant qui est la forme générale des invariants classique. Nous présentons la méthode de Hilbert pour le calcul d'invariants classiques de *degré* et *poids* donnés (une implantation de l'algorithme associé est donnée en annexe C). Nous introduisons ensuite le lien entre les invariants classiques et les *racines formelles* et nous décrivons deux méthodes classiques pour déterminer des invariants classiques en fonction de polynômes-différences symétriques et vis-versa. La principale application (et motivation) de ce travail est en théorie de Galois : le calcul de *résolvantes* associées à des invariants classiques développé dans le chapitre 4.

3.1 Covariants classiques

Nous allons, tout au long de cette partie, travailler sur des formes binaires $f(x, y)$ homogène et de degré n en x et y définies par :

$$\begin{aligned} f(x,y) &= \sum_{k=0}^{k=n} C_n^k \, a_k \, x^k \, y^{n-k} \\ &= a_n \, x^n \, + \, C_n^1 \, a_{n-1} \, x^{n-1} \, y \, + \cdots + \, C_n^{n-1} \, a_1 \, x \, y^{n-1} \, + \, a_0 \, y^n \qquad , \end{aligned}$$

où $C_n^k = \frac{n!}{(n-k)!k!}$ est le symbole binomial. Les a_k sont appelés les *coefficients* de $f(x, y)$ et ils appartiennent à un corps k de caractéristique nulle. L'ensemble des formes binaires de degré n sera noté $\mathcal{F}_n$.
La caractéristique du corps k étant supposée nulle, l'utilisation des formes binaires sous la forme $f(x,y) = \sum_{k=0}^{k=n} C_n^k \, a_k \, x^k \, y^{n-k}$ plutôt que sous la forme $f(x,y) = \sum_{k=0}^{k=n} a_k \, x^k \, y^{n-k}$ nous évite de travailler avec des facteurs numériques entiers rationnels « parasites », qui s'introduisent lors des calculs.

3.1.1 Transformations linéaires

Un *changement linéaire de variables* $(c_{ij})_{1 \leq i,j \leq 2}$ est une transformation des variables x et y donnée par :

$$\begin{aligned} x &= c_{11}\bar{x} + c_{12}\bar{y} \qquad , \\ y &= c_{21}\bar{x} + c_{22}\bar{y} \qquad , \end{aligned} \tag{3.1}$$

où le déterminant $c_{11}c_{22} - c_{12}c_{21}$ est non nul. Le groupe $GL_2(k)$ des matrices inversibles d'ordre 2 agit sur l'ensemble des formes binaires $\mathcal{F}_n$. En effet, pour tout changement linéaire de variables $(c_{ij}) \in GL_2(k)$, la forme binaire $f(x, y)$ est transformée en une autre forme binaire $\bar{f}$ en les variables $\bar{x}$ et $\bar{y}$ définie par :

$$\bar{f}(\bar{x},\bar{y}) = \sum_{k=0}^{k=n} C_n^k a_k (c_{11}\bar{x} + c_{12}\bar{y})^k (c_{21}\bar{x} + c_{22}\bar{y})^{n-k} \qquad .$$

En regroupant les termes, nous obtenons :

$$\bar{f}(\bar{x},\bar{y}) \; = \sum_{k=0}^{n} C_n^k \, \bar{a}_k \, \bar{x}^k \, \bar{y}^{n-k} \qquad ,$$

où les coefficients $\bar{a}_k$ sont des polynômes en les a_i et c_{ij} :

$$\bar{a}_k \;=\; \sum_{m=0}^{n} \left(\sum_{i=m-n+k}^{\min(m,k)} C_k^i \; C_{n-k}^{m-i} \; c_{11}^i \; c_{12}^{m-i} \; c_{21}^{k-i} \; c_{22}^{n-k-m+i} \right) a_m \;\; .$$

Notons $\mathcal{P}$ l'anneau des polynômes en les variables $A_0, A_1, \ldots, A_n, X$ et Y et à coefficients dans k. Grâce au changement linéaire de variables (c_{ij}) défini par l'équation (3.1), nous définissons $\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}$ et $\bar{Y}$ par :

$$\begin{aligned} X &= c_{11}\bar{X} + c_{12}\bar{Y} \qquad , \\ Y &= c_{21}\bar{X} + c_{22}\bar{Y} \qquad , \\ \bar{A}_0 &= \sum_{m=0}^{n} C_n^m c_{12}^m \;\; c_{22}^{n-m} \;\; A_m \qquad , \\ \bar{A}_n &= \sum_{m=0}^{n} C_n^m c_{11}^m \;\; c_{21}^{n-m} \;\; A_m \end{aligned}$$

et pour $k \in [1, n-1]$, $\bar{A}_k \;=\; \sum_{m=0}^{n} \left(\sum_{i=m-n+k}^{\min(m,k)} C_k^i \; C_{n-k}^{m-i} \; c_{11}^i \; c_{12}^{m-i} \; c_{21}^{k-i} \; c_{22}^{n-k-m+i} \right) A_m$.

Exemple 3.1.1. Soit $f(x,y) \in \mathcal{F}_2$ définie par : $f(x,y) = a_2x^2 + 2a_1xy + a_0y^2$. Nous avons $\bar{f}(\bar{x},\bar{y}) = \bar{a}_2\bar{x}^2 + 2\bar{a}_1\bar{x}\bar{y} + \bar{a}_0\bar{y}^2$ avec $\bar{a}_0 = c_{22}a_0 + 2c_{12}c_{22}a_1 + c_{12}^2a_2$, $\bar{a}_1 = 2(c_{21}c_{22}a_0 + (c_{12}c_{21} + c_{11}c_{22})a_1 + c_{11}c_{12}a_2)$,$\bar{a}_2 = c_{21}^2a_0 + 2c_{11}c_{21}a_1 + c_{11}^2a_2$.

3.1.2 Covariants et Invariants classiques

Soient g un entier positif non nul et n un entier positif.

Définitions 3.1.2. Soit $M = A_0^{r_0}A_1^{r_1}\ldots A_n^{r_n}$ un monôme de $\mathcal{P}$. Le *degré* de M est égal à $\sum_{i=0}^{n} r_i$. Le *poids* de M est égal à $\sum_{i=0}^{i=n} i\, r_i$.
Le *degré* (respectivement *poids*) d'un polynôme J de $\mathcal{P}$ est égal au maximum des degrés (respectivement poids) des monômes de J.
Un polynôme $P \in \mathcal{P}$ est dit *homogène* si ses monômes sont de même degré.

Un polynôme non constant $J(A_0, A_1, \ldots, A_n, X, Y) \in \mathcal{P}$ est appelé *covariant classique de poids* g si pour tout changement linéaire de variables (c_{ij}), nous avons l'égalité suivante :

$$J(\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}, \bar{Y}) = (c_{11}c_{22} - c_{12}c_{21})^g J(A_0, A_1, \ldots, A_n, X, Y) \qquad . \tag{3.2}$$

Un covariant qui n'est pas fonction de X et Y est dit *invariant classique* (ou *invariant relatif*) *de poids* g.

Remarque 3.1.3. Soit $J(A_0, A_1, \ldots, A_n)$ un invariant classique et notons $J(f)$ le polynôme $J(a_0, a_1, \ldots, a_n)$ où les a_i sont les coefficients de la forme binaire f de degré n. L'équation

$J(f) = 0$ décrit une propriété de géométrie projective et l'ensemble $\{g \in \mathcal{F}_n \mid J(g) = 0\}$ est invariant sous l'action de $GL_2(k)$.

Le groupe $GL_2(k)$ est engendré par la réunion des deux sous-groupes (chacun isomorphe à $(k, +)$) $\mathcal{U}_+$ et $\mathcal{U}_-$, où $\mathcal{U}_+$ est l'ensemble des matrices de la forme $\left(\begin{smallmatrix} 1 & \lambda \\ 0 & 1 \end{smallmatrix}\right)$ avec $\lambda \in k$, et où $\mathcal{U}_-$ est l'ensemble des matrices de la forme $\left(\begin{smallmatrix} 1 & 0 \\ \mu & 1 \end{smallmatrix}\right)$ avec $\mu \in k$. Notons $\mathcal{T}_+$ (respectivement $\mathcal{T}_-$) le sous-groupe de $GL_2(k)$ formé des matrices de la forme $S_{\rho,\lambda} = \left(\begin{smallmatrix} \rho & \lambda \\ 0 & 1 \end{smallmatrix}\right)$, avec $(\rho, \lambda) \in k^* \times k$ (respectivement $S'_{\rho,\mu} = \left(\begin{smallmatrix} 1 & 0 \\ \mu & \rho \end{smallmatrix}\right)$, avec $(\rho, \mu) \in k^* \times k$). Les groupes $\mathcal{T}_+$ et $\mathcal{T}_-$ sont isomorphes au groupe des similitudes de k ; le groupe $\mathcal{U}_+$ est un sous-groupe distingué de $\mathcal{T}_+$, le groupe quotient $\mathcal{T}_+/\mathcal{U}_+$ étant isomorphe à k^*. De même, $\mathcal{U}_- \lhd \mathcal{T}_-$ et $\mathcal{T}_-/\mathcal{U}_-$ est isomorphe à k^*.

La définition suivante introduit ce que les anciens « invariantistes » appelaient autrefois *semi-invariants binaires* (terminologie de Cayley) :

Définitions 3.1.4. Un *semi-invariant* est un polynôme homogène de degré d de $\mathcal{P}$ qui est un *invariant classique* du groupe $\mathcal{T}_+$, c'est-à-dire qui, sous l'action de $S_{\rho,\lambda} \in \mathcal{T}_+$, est multiplié par ρ^g avec $g \in \mathbf{N}$.
L'entier g est appelé le *poids* du semi-invariant I. Pour tout monôme $M = \gamma A_0^{r_0} A_1^{r_1} \dots A_n^{r_n}$ de I tel que $\gamma \neq 0$, nous avons encore $\sum_{i=0}^{i=n} i\, r_i = g$, c'est-à-dire le semi-invariant I est *isobare de poids* g. De plus, $nd = 2g$, donc nd est pair.

Un *invariant absolu* est un polynôme stable sous l'action d'un groupe donné. Soit I un polynôme homogène de degré $d \geq 1$ en les coefficients de la forme binaire générale de degré n ; supposons que I soit un invariant absolu du groupe $SL_2(k)$. Alors nécessairement nd est pair et I est semi-invariant de poids $\frac{nd}{2}$ (la démonstration est facile, s'étend aux h-formes avec $h \geq 2$ quelconque et n'utilise que l'irréductibilité du déterminant général sur un corps commutatif : avec h quelconque, il faut remplacer $\frac{nd}{2}$ par $\frac{nd}{h}$).

Pour qu'un polynôme I à coefficient dans k, homogène de degré d en $(A_0, A_1, \dots, A_n)$ soit un semi-invariant de poids g, il faut et il suffit que les conditions suivantes soient satisfaites : I est un invariant absolu du groupe $\mathcal{U}_+$ et I isobare de poids $g = \frac{nd}{2}$.

Soit $T = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. Il est clair que $\mathcal{U}_+ \cup \{T\}$ engendre le groupe $SL_2(k)$, donc d'après la propriété vue ci-dessus, un semi-invariant homogène de degré d est un invariant classique si et seulement si il reste invariant par T. L'action de T revient à remplacer la liste $(A_0, A_1, \dots, A_n)$ par $((-1)^n A_n, (-1)^{n-1} A_{n-1}, \dots, A_0)$. Soit T' la matrice $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. L'action de T' revient à remplacer $(A_0, A_1, \dots, A_n)$ par $(A_n, A_{n-1}, \dots, A_0)$. Le sous-groupe de $GL_2(k)$ engendré par $\mathcal{U}_+ \cup \{T'\}$ contient $SL_2(k)$ comme sous-groupe d'indice 2. Comme $\det(T') = -1$, nous en déduisons facilement qu'un semi-invariant homogène de degré d de $\mathcal{P}$ est un invariant classique de $\mathcal{F}_n$ si et seulement si l'action de T' le multiplie par $(-1)^{\frac{nd}{2}}$.

3.1.3 Les opérateurs différentiels de Cayley

Supposons que le corps k est de caractéristique nulle. Soient Δ_1 et Δ_2 deux fonctions de $\mathcal{P}$ dans $\mathcal{P}$ qui, pour un polynôme $J \in \mathcal{P}$, sont définies par :

$$\begin{aligned} \Delta_1(J) &= \sum_{i=1}^{n} i\, A_{i-1} \frac{\partial(J)}{\partial A_i} \qquad \text{et} \\ \Delta_2(J) &= \sum_{i=1}^{n} (n-i+1)\, A_i \frac{\partial(J)}{\partial A_{i-1}} \qquad . \end{aligned} \tag{3.3}$$

Remarque 3.1.5. Un invariant classique (ou invariant relatif) I de poids $g = \frac{nd}{2}$ est un semi-invariant vérifiant

$$I(A_n, \ldots, A_0) = (-1)^g I(A_0, \ldots, A_n) \qquad . \tag{3.4}$$

Les opérateurs différentiels Δ_1 et Δ_2 sont des opérateurs de Aronhold de type particulier. Le lemme suivant permet de caractériser les invariants classiques à partir de ces opérateurs différentiels. Des preuves complètes ou des ébauches de preuves sont donnés par exemple dans [13]. Nous trouverons dans [50] d'autres méthodes algébriques pour décrire les invariants des formes binaires.

Lemme 3.1.6 (Hilbert). *Le polynôme $J \in \mathcal{P}$ est un invariant classique si et seulement si $\Delta_1(J) = \Delta_2(J) = 0$.*

Preuve. Considérons un invariant classique $J \in \mathcal{P}$. Le polynôme J vérifie l'égalité (3.2) pour tout changement de variable $(c_{ij}) \in GL_2(k)$ et en particulier pour les deux changements linéaires de variables engendrés par les deux matrices :

$$M_1 = \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \in GL_2(k) \qquad \text{et} \qquad M_2 = \begin{pmatrix} 1 & \mu \\ 0 & 1 \end{pmatrix} \in GL_2(k) \qquad ,$$

avec $\lambda \in k^*$ et $\mu \in k^*$. Soit le changement linéaire de variables défini par : $X \longrightarrow \bar{X}$ et $Y \longrightarrow \lambda \bar{X} + \bar{Y}$ où λ est une constante dans k^* qui correspond à la matrice M_1. Alors pour $k \in [0, n]$, $\bar{A}_k = A_k + \sum_{i=0}^{k-1} C_k^i\, \lambda^{k-i}\, A_i$. Notons $h_k = \bar{A}_k - A_k$.
En développant $J(\bar{A}_0, \ldots, \bar{A}_n)$ en série de Taylor en $(h_0, h_1, \ldots, h_n)$ à un ordre $m \in \mathbf{N}$, nous obtenons l'égalité suivante :

$$\begin{aligned} J(\bar{A}_0, \ldots, \bar{A}_n) &= J(A_0 + h_0, \ldots, A_n + h_n) \\ &= J(A_0, \ldots, A_n) + \sum_{\{\alpha \in \mathbf{N}^n \mid |\alpha| \in [1,m]\}} \frac{D^\alpha J(A_0, \ldots, A_n)}{\alpha!} (h_0^{\alpha_0} \ldots h_n^{\alpha_n}) \ , \end{aligned}$$

avec pour $\alpha = (\alpha_0, \ldots, \alpha_n) \in \mathbf{N}^n$, $\mid \alpha \mid = \alpha_0 + \cdots + \alpha_n$, $\alpha! = \alpha_0! \ldots \alpha_n!$ et D^α l'opérateur différentiel $(\frac{\partial}{\partial A_0})^{\alpha_0} \ldots (\frac{\partial}{\partial A_n})^{\alpha_n}$.

Pour ce changement de variables, $J(\bar{A}_0,\ldots,\bar{A}_n) = J(A_0,\ldots,A_n)$ et la différence entre ces deux polynômes doit être nulle ; soit :

$$\sum_{|\alpha|=1}^{|\alpha|=m} \frac{D^\alpha J(A_0,\ldots,A_n)}{\alpha!}(h_0^{\alpha_0}\ldots h_n^{\alpha_n}) \quad = 0 \qquad . \tag{3.5}$$

Le premier terme de l'égalité (3.5) peut-être considéré comme un polynôme en λ. En remplaçant h_k par sa valeur, nous obtenons le développement :

$$\sum_{|\alpha|=1}^{|\alpha|=m} \frac{D^\alpha J(A_0,\ldots,A_n)}{\alpha!} \prod_{k=0}^{n} (\sum_{i=0}^{k-1} C_k^i \, \lambda^{k-i} \, A_i)^{\alpha_k} = 0 \qquad . \tag{3.6}$$

Le coefficient de λ, obtenu avec $\mid \alpha \mid = 1$, est égal à $\sum_{i=1}^{n} i \, A_{i-1} \frac{\partial(J)}{\partial A_i} = \Delta_1(J)$. Pour un entier j quelconque, le coefficient de λ^j dans l'égalité (3.6), calculé avec $\mid \alpha \mid \leq j$, est égal à $\frac{1}{j!}\Delta_1^j(J)$. Il s'en suit l'égalité suivante :

$$J(\bar{A}_0,\ldots,\bar{A}_n) - J(A_0,\ldots,A_n) = \; \lambda \; \Delta_1(J) + \frac{\lambda^2}{2!}\Delta_1^2(J) + \frac{\lambda^3}{3!}\Delta_1^3(J) + \cdots .$$

Cette égalité est vrai quelque soit $\lambda \in k^*$, donc $J(\bar{A}_0,\ldots,\bar{A}_n) = J(A_0,\ldots,A_n)$ implique que $\Delta_1(J) = 0$. D'autre part, en considérant le changement linéaire de variables $X \longrightarrow \bar{X} + \mu\bar{Y}$ et $Y \longrightarrow \bar{Y}$ où μ est une constante dans k^*, nous obtenons pour $k \in [0,n]$:

$$\bar{A}_k = A_k + \sum_{i=1}^{n-k} C_{n-k}^{n-k-i} \, \mu^i \, A_{k+i} \qquad .$$

Posons $g_k = \sum_{i=1}^{n-k} C_{n-k}^{n-k-i} \, \mu^i \, A_{k+i}$. En recommençant le même procédé que ci-dessus, nous obtenons les égalités suivantes pour un polynôme $J \in \mathcal{P}$:

$$\begin{aligned} J(\bar{A}_0,\ldots,\bar{A}_n) &= J(A_0+g_0,\ldots,A_n+g_n) \\ &= J(A_0,\ldots,A_n) + \sum_{|\alpha|=1}^{|\alpha|=m} \frac{D^\alpha J(A_0,\ldots,A_n)}{\alpha!}(g_0^{\alpha_0}\ldots g_n^{\alpha_n}) \qquad . \end{aligned}$$

Si le polynôme J est un invariant classique, alors $J(\bar{A}_0,\ldots,\bar{A}_n) = J(A_0,\ldots,A_n)$. Il est donc nécessaire que la différence entre ces deux polynômes soit nulle :

$$\sum_{|\alpha|=1}^{|\alpha|=m} \frac{D^\alpha J(A_0,\ldots,A_n)}{\alpha!} \prod_{k=0}^{n} (\sum_{i=1}^{n-k} C_{n-k}^{n-k-i} \, \mu^i \, A_{k+i})^{\alpha_k} = 0 \qquad . \tag{3.7}$$

Le coefficient de μ dans le premier terme de l'équation (3.7) obtenu avec $\mid \alpha \mid = 1$ est égal à $\sum_{i=0}^{n-1} C_{n-i}^{n-i-1} \, A_{i+1} \frac{\partial(J)}{\partial A_i} = \Delta_2(J)$. Pour un entier j quelconque, le coefficient de μ^j dans l'égalité (3.7) obtenu avec $\mid \alpha \mid \leq j$ est égal à $\frac{1}{j!}\Delta_2^j(J)$. D'où

$$J(\bar{A}_0,\ldots,\bar{A}_n) - J(A_0,\ldots,A_n) = \; \mu \; \Delta_2(J) + \frac{\mu^2}{2!}\Delta_2^2(J) + \frac{\mu^3}{3!}\Delta_2^3(J) + \ldots$$

Cette égalité est vrai quelque soit $\mu \in k^*$, donc $J(\bar{A}_0, \ldots, \bar{A}_n) = J(A_0, \ldots, A_n)$ implique que $\Delta_2(J) = 0$. Nous avons donc montré que si J est un invariant classique, alors $\Delta_1(J) = \Delta_2(J) = 0$.

Réciproquement, supposons que $\Delta_1(J) = \Delta_2(J) = 0$ où J est un polynôme de $\mathcal{P}$ supposé homogène de degré d et montrons que J est un invariant classique. Pour cela, nous allons utiliser le lien entre invariants classiques et les semi-invariants.
Le polynôme J est un invariant classique de poids g si pour tout changement linéaire de variables $(c_{ij}) \in GL_2(k)$:

$$J(\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}, \bar{Y}) = (c_{11}c_{22} - c_{12}c_{21})^g J(A_0, A_1, \ldots, A_n, X, Y) \quad ,$$

Puisque $\Delta_2(J) = 0$, alors $J(A_0 + g_0, \ldots, A_n + g_n) = J(A_0, \ldots, A_n)$ et J est un semi-invariant.
Il reste donc à montrer que J vérifie la condition (3.4). En considérant le changement de variables $X \to Y$ et $Y \to X$, nous remarquons que $\bar{A}_k = -A_k + (A_{n-k} + A_k)$ et que par conséquent $J(A_n, \ldots, A_0) = J(-A_0 + (A_0 + A_n), \ldots, -A_n + (A_n + A_0))$. En supposant que le polynôme J est de poids g, nous déduisons le développement en série de Taylor suivant :

$$J(A_n, \ldots, A_0) - (-1)^g J(A_0, \ldots, A_n) = \sum_{|\alpha|=1}^{|\alpha|=m} \frac{D^\alpha J(A_0, \ldots, A_n)}{\alpha!} \prod_{k=0}^{n} (A_k + A_{n-k})^{\alpha_k}. \quad (3.8)$$

Enfin, en appliquant l'opérateur différentiel Δ_1 sur le deuxième terme de l'égalité (3.8), nous remarquons qu'il s'annule et que par conséquent :
$J(A_n, \ldots, A_0) - (-1)^g J(A_0, \ldots, A_n) = 0$. □

Remarque 3.1.7. En développant en série de Taylor l'égalité :

$$J(\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}, \bar{Y}) - (c_{11}c_{22} - c_{12}c_{21})^g J(A_0, A_1, \ldots, A_n, X, Y) \quad ,$$

où J est un polynôme homogène de poids g et (c_{ij}) un changement linéaire de variables, nous remarquons que le polynôme obtenu est un polynôme en les c_{ij} et à coefficients les différentes puissance de Δ_1 et Δ_2. C'est donc un moyen de prouver la condition suffisante du lemme 3.1.6, mais les calculs sont fastidieux et nous les avons simplifiés par les semi-invariants.

3.1.4 Système complet d'invariants classiques irréductibles

D. Hilbert a prouvé qu'il existe un ensemble fini d'invariants classiques de formes binaires de degré n qui génèrent algébriquement tous les autres. Ce résultat a marqué l'Algèbre au 20$^{\text{e}}$ siècle. En effet, D. Hilbert a montré pour la première fois que l'anneau des polynômes à un nombre fini quelconques de variables sur un corps est *noethérien*. A cause du poids qui peut varier, les invariants classiques polynomiaux ne sont pas dotés de structure

simple : pour un poids donné, nous n'avons qu'un k-espace vectoriel d'invariants classiques polynomiaux. Le produit de deux invariants classiques en est encore un, mais le poids a changé. Si bien que les les invariants classiques polynomiaux en $(A_0, \ldots, A_n)$ ne forment pas une sous-k-algèbre de la k-algèbre des polynômes $k[A_0, \ldots, A_n]$.

D. Hilbert a prouvé que n étant donné, il existe une suite finie $(I_1, \ldots, I_m)$ d'invariants classiques en $(A_0, \ldots, A_n)$ possédant les propriétés suivantes :

(1) tout invariant classique en $(A_0, \ldots, A_n)$ appartient à $k[I_1, \ldots, I_m]$, et

(2) pour tout $i \in [1, n]$, l'invariant I_i n'appartient pas à $k[(I_j)_{j \in [1,m] \setminus \{i\}}]$.

La seconde propriété se traduit en disant que les I_i sont un système d'invariants classiques *irréductibles* (ces propriétés ne sont pas particulières aux formes binaires, elles s'étendent aux invariants classiques des formes à un nombre fini quelconque $h \geq 2$ de variables.

Pour exprimer que les deux propriétés (1) et (2) sont vraies, nous disons que $(I_1, \ldots, I_m)$ est un *système complet d'invariants classiques irréductibles*. Il n'y a en général pas unicité d'un système complet d'invariants irréductibles. La théorie classique des invariants explicite des systèmes complets pour les formes binaires de degré 3 et 4. Des algorithmes d'obtention d'un système complet pour les formes binaires de degré quelconque ont été proposés depuis longtemps par Clebsch, Gordan et d'autres. Nous verrons dans le chapitre 4 comment utiliser l'algorithme 3.1.1 afin d'obtenir les premiers éléments (classés selon les degrés) des invariants classiques irréductibles.

Algorithme 3.1.1 (InvariantClassique).
Cet algorithme calcule tous les invariants classiques homogènes de degré et poids fixés.

Fonction `InvariantClassique(n,d,g) ==`

`Entrées :` n, d, g `trois entiers positifs.`
`Sortie :` `Tous les invariants classiques de degré` d `et de poids` g `(s'il existe un tel invariant classique).`

1. `Si` $g \neq \frac{nd}{2}$ `alors Pas d'invariants classique de poids` g
   `Sinon :`

2. `Déterminer tous les monômes` M_i `en` $A_0, A_1, \ldots, A_n$ `de degré` d `et de poids` g`. Soit` r `leur nombre.`
3. `Calculer les dérivées de` $\sum_{i=1}^{r} X_i M_i$ `par rapport à` Δ_1 `et` Δ_2 `où` $X_1, \ldots, X_r$ `sont des inconnues.`
4. `Écrire les deux polynômes obtenus dans la précédente étape sous forme de polynômes à coefficients en` $X_1, \ldots, X_r$ `en les variables` $A_0, A_1, \ldots, A_n$.

5. Résoudre le système linéaire des équations des coefficients des précédents polynômes. Les inconnues sont $X_1, \ldots, X_r$.
6. Si $x_1, \ldots x_r$ sont les solutions du système de l'étape 4, les invariants (s'ils existent) de degré d et de poids g en $A_0, A_1, \ldots, A_n$ sont donnés par $\sum_{i=1}^{r} x_i M_i$.

Fin Si ;

Fin.

Exemple 3.1.8. Si nous fixons n et d, nous pouvons déterminer grâce à l'algorithme 3.1.1 l'invariant classique de degré d et de poids $g = \frac{1}{2}nd$, s'il existe. Soient $n = 8$, $d = 3$ et $g = 12$. La forme générale d'un polynôme en $A_0, A_1, \ldots, A_8$ de degré 3 et de poids 12 est égale à :

$$\begin{aligned} P \;=\; & X_1A_4^3 + X_2A_3^2A_6 + X_3A_2A_5^2 + X_4A_2^2A_8 + X_5A_0A_6^2 + X_6A_3A_4A_5 + X_7A_2A_4A_6 + \\ & X_8A_2A_3A_7 + X_9A_1A_5A_6 + X_{10}A_1A_4A_7 + X_{11}A_1A_3A_8 + X_{12}A_0A_5A_7 + X_{13}A_0A_4A_8 \quad . \end{aligned}$$

Les X_i pour $(i \in [1, 13])$ sont des inconnues. Les équations $\Delta_1(P) = 0$ et $\Delta_2(P) = 0$ sont alors :

$$\begin{aligned} \Delta_1(P) \;=\; & 12X_1A_3A_4^2 + X_2(6A_2A_3A_6 + 6A_3^2A_5) + X_3(2A_1A_5^2 + 10A_2A_4A_5) + \\ & X_4(4A_1A_2A_8 + 8A_2^2A_7) + 12X_5A_0A_5A_6 + X_6(3A_2A_4A_5 + 4A_3^2A_5 + 5A_3A_4^2) + \\ & X_7(2A_1A_4A_6 + 4A_2A_3A_6 + 6A_2A_4A_5) + X_8(2A_1A_3A_7 + 3A_2^2A_7 + 7A_2A_3A_6) + \\ & X_9(A_0A_5A_6 + 5A_1A_4A_6 + 6A_1A_5^2) + X_{10}(A_0A_4A_7 + 4A_1A_3A_7 + 7A_1A_4A_6) + \\ & X_{11}(A_0A_3A_8 + 3A_1A_2A_8 + 8A_1A_3A_7) + X_{12}(5A_0A_4A_7 + 7A_0A_5A_6) + \\ & X_{13}(4A_0A_3A_8 + 8A_0A_4A_7) \quad , \end{aligned}$$

$$\begin{aligned} \Delta_2(P) \;=\; & 12X_1A_4^2A_5 + X_2(10A_3A_4A_6 + 2A_3^2A_7) + X_3(6A_3A_5^2 + 6A_2A_5A_6) + \\ & 12X_4A_2A_3A_8 + X_5(8A_1A_6^2 + 4A_0A_6A_7) + X_6(5A_4^2A_5 + 4A_3A_5^2 + 3A_3A_4A_6) + \\ & X_7(6A_3A_4A_6 + 4A_2A_5A_6 + 2A_2A_4A_7) + X_8(6A_3^2A_7 + 5A_2A_4A_7 + A_2A_3A_8) + \\ & X_9(7A_2A_5A_6 + 3A_1A_6^2 + 2A_1A_5A_7) + X_{10}(7A_2A_4A_7 + 4A_1A_5A_7 + A_1A_4A_8) + \\ & X_{11}(7A_2A_3A_8 + 5A_1A_4A_8) + X_{12}(8A_1A_5A_7 + 3A_0A_6A_7 + A_0A_5A_8) + \\ & X_{13}(8A_1A_4A_8 + 4A_0A_5A_8) \quad . \end{aligned}$$

Ces deux équations nous permettent de déduire le système d'équations suivant :

$$\text{avec } \Delta_1 : \begin{cases} X_{11} + 4X_{13} = 0 \\ X_{10} + 8X_{13} + 5X_{12} = 0 \\ 12X_5 + X_9 + 7X_{12} = 0 \\ 4X_4 + 3X_{11} = 0 \\ 4X_{10} + 2X_8 + 8X_{11} = 0 \\ 5X_9 + 2X_7 + 7X_{10} = 0 \\ 2X_3 + 6X_9 = 0 \\ 8X_4 + 3X_8 = 0 \\ 7X_8 + 4X_7 + 6X_2 = 0 \\ 3X_6 + 6X_7 + 10X_3 = 0 \\ 4X_6 + 6X_2 = 0 \\ 5X_6 + 12X_1 = 0 \end{cases} \quad , \quad \text{avec } \Delta_2 : \begin{cases} X_{12} + 4X_{13} = 0 \\ 4X_5 + 3X_{12} = 0 \\ X_{10} + 5X_{11} + 8X_{13} = 0 \\ 2X_9 + 4X_{10} + 8X_{12} = 0 \\ 3X_9 + 8X_5 = 0 \\ 12X_4 + 7X_{11} + X_8 = 0 \\ 2X_7 + 7X_{10} + 5X_8 = 0 \\ 4X_7 + 6X_3 + 7X_9 = 0 \\ 6X_8 + 2X_2 = 0 \\ 10X_2 + 6X_7 + 3X_6 = 0 \\ 6X_3 + 4X_6 = 0 \\ 5X_6 + 12X_1 = 0 \end{cases}$$

Nous en déduisons l'invariant classique homogène de degré $d = 3$ et de poids $g = 12$ pour $n = 8$:

$$\begin{aligned} P &= 15A_4^3 + 24A_3^2A_6 + 24A_2A_5^2 + 3A_2^2A_8 + 3A_0A_6^2 - 36A_3A_4A_5 \\ &\quad -22A_2A_4A_6 - 8A_2A_3A_7 - 8A_1A_5A_6 + 12A_1A_4A_7 - 4A_1A_3A_8 \\ &\quad -4A_0A_5A_7 + A_0A_4A_8 \ . \end{aligned}$$

La détermination de tous les monômes en $A_0, A_1, \ldots, A_n$ de degré d donné a été faite grâce à une amélioration de l'algorithme de calcul de monômes de la section 1.5.2 du chapitre 1. Nous avons en effet, choisit une structure combinatoire de données qui ne tient compte que des degrés des monômes. L'implantation de l'algorithme 3.1.1 est faite en GAP et nous utilisons la fonction `solve` de MAPLE pour résoudre le système (3.3) en des temps de calculs très rapides (de l'ordre de quelques secondes). Cet algorithme est détaillé dans l'annexe C de ce mémoire.

Exemples 3.1.9. Pour $n = 2$ et $d = 2$, $A_0A_2 - A_1^2$ est un semi-invariant et il apparaît dans la factorisation de tous les semi-invariants de $n = 2$ et de degré d pair.
Pour $n = 4$ et $d = 2$, nous trouvons le semi-invariant $4A_1A_3 - 3A_2^2 - A_4A_0$. Pour $d = 6$, le polynôme $\frac{1}{2}(A_0A_3^2 - 2A_2A_3A_1 + A_4A_1^2 + A_2^3 - A_4A_0A_2)^2(4A_1A_3 - A_0A_4 - 3A_2^2)^3$ est un invariant classique. L'invariant $(A_0A_3^2 - 2A_2A_3A_1 + A_4A_1^2 + A_2^3 - A_4A_0A_2)(4A_1A_3 - A_0A_4 - 3A_2^2)$ est aussi retrouvé en appliquant l'algorithme 3.1.1 à $d = 5$. C'est cet effet de redondance qui fait que le nombre d'invariants pour n fixé soit fini.

Soit $f(x, y) = a_6\,x^6 + 6\,a_5\,x^5\,y + 15a_4x^4y^2 + 20a_3x^3y^3 + 15a_2x^2y^4 + 6a_1xy^5 + a_0y^6$ une forme binaire de degré 6 à coefficient dans $\mathbf{Q}$. Notons $\alpha_1, \ldots, \alpha_6$ les 6 racines de $g = f(x, 1) = 0$ dans une extension classique du corps de ses coefficients. Alors le semi-invariant $I_2(A_0, \ldots, A_6) = -10A_3^2 + 15A_2A_4 - 6A_1A_5 + A_0A_6$ vérifie :

$$\begin{aligned} I_2(a_0, \ldots, a_6) &= -10a_3^2 + 15a_2a_4 - 6a_1a_5 + a_0a_6 \\ &= -45(\frac{a_6}{6!})^2 \sum_{\sigma \in S_6} (\alpha_{\sigma(1)} - \alpha_{\sigma(2)})^2(\alpha_{\sigma(3)} - \alpha_{\sigma(4)})^2(\alpha_{\sigma(5)} - \alpha_{\sigma(6)})^2 \ . \end{aligned}$$

Notons $\Theta = (x_1 - x_2)^2(x_3 - x_4)^2(x_5 - x_6)^2$ et considérons le groupe $H = <$(5,6), (3,4), (3,5,4,6), (1,2), (1,3,2,4), (1,5,2,6)$>$. Le polynôme Θ est un H-invariant absolu de S_6 (voir définition 1.1.1 du chapitre 1). Le polynôme $I_2(a_0, \ldots, a_6)$ détermine le coefficient sous dominant de la *résolvante* $\mathcal{L}_{\Theta,g}^{S_6}$ (voir chapitre 4).

D'une façon générale, il existe un lien entre les invariants classiques et des fonctions particulières de racines du polynôme f. Nous utilisons dans la section suivante les *notations symboliques* pour un algorithme qui détermine les invariants classiques en fonction d'invariants de groupes (associés à la théorie moderne des invariants). C'est une méthode ancienne pour caractériser les invariants classiques que nous avons retrouvé dans différentes lectures (voir par exemple [60], [14] et [47]). Nous allons voir dans quel cas un

invariant classique est un invariant de groupe et nous implantons l'algorithme de passage entre invariants classiques et invariants de groupes.

3.2 Invariants classiques et Polynômes-différences

Soit $\mathcal{R} = \{0, 1, 2, 3, \ldots, u\}$ un alphabet contenant la lettre u et une infinité d'entiers positifs. Les éléments de $\mathcal{R}$ sont appelés *lettres symboliques*.

Définition 3.2.1. A chaque entier i de $\mathcal{R}$ nous associons les variables ν_i et μ_i et pour la lettre $u \in \mathcal{R}$ nous associons les variables x et y: $i \to \binom{\mu_i}{\nu_i}$, $u \to \binom{x}{y}$.
L'anneau des polynômes à coefficients dans un corps k et en les variables μ_i, ν_i, x et y associées à $\mathcal{R}$ est noté $\mathcal{U}$ et est appelé *Espace Symbolique*.

Nous présentons dans cette section la notion *d'opérateur symbolique* qui nous permet de représenter des polynômes de $\mathcal{P}$ en fonction des éléments de l'espace symbolique $\mathcal{U}$. Nous appelons ces fonctions les *polynômes-différences* et nous montrons dans la section 3.2.1 comment retrouver des invariants classiques en fonction des polynômes-différences et vis-versa.
Définition 3.2.2. l'*opérateur symbolique* U des formes binaires de degré n est un opérateur linéaire de l'espace Symbolique $\mathcal{U}$ dans l'anneau des polynômes $\mathcal{P}$. L'action de l'opérateur U sur un polynôme $P(\mu_1, \nu_1, \ldots) \in \mathcal{U}$ s'écrit sous la forme $< U \mid P(\mu_1, \nu_1, \ldots) >$ et elle est définie par:

$$\begin{aligned}
< U \mid \mu_i^k \nu_i^{n-k} > &= A_k \qquad \forall i \text{ entier } \in \mathcal{R} \\
< U \mid \mu_i^k \nu_i^l > &= 0 \qquad \text{si } k+l \neq n \text{ et } \forall i \text{ entier } \in \mathcal{R} \\
< U \mid x^k > &= (-Y)^k \\
< U \mid y^k > &= X^k \\
< U \mid \mu_1^i \nu_1^j \mu_2^k \nu_2^l \ldots x^p y^q > &= < U \mid \mu_1^i \nu_1^j >< U \mid \mu_2^k \nu_2^l > \ldots < U \mid x^p >< U \mid y^q > \quad .
\end{aligned}$$

Tout polynôme $J \in \mathcal{P}$ peut s'écrire sous la forme $< U \mid Q(\mu_1, \nu_1, \ldots) >$ où Q est un polynôme de $\mathcal{U}$ appelé *représentation symbolique* du polynôme J (respectivement J est appelé *Évaluation symbolique* de Q).

Définition 3.2.3. Soit $f(x,y) = \sum_{k=0}^{n} C_n^k a_k x^k y^{n-k}$ une forme appartenant à $\mathcal{F}_n$. La *fonction linéaire symbolique associée à* $f(x,y)$, notée $U(f)$, est définie par:

$$\begin{array}{rccc}
U(f) \ : & \mathcal{U} & \longrightarrow & k[x,y] \\
& J(A_0, A_1, \ldots, A_n, X, Y) & \mapsto & J(a_0, a_1, \ldots, a_n, x, y)
\end{array}$$

Exemple 3.2.4. Soit $n = 3$ et le polynôme $P = \mu_1\nu_1^2\mu_2\nu_2^2\mu_3\nu_3^2 + \mu_1\nu_1 + \mu_1^3\nu_3^3$ de $\mathcal{U}$. Alors $< U \mid P >= A_1^3 + A_0A_3$ et pour $f(x,y) = a_3\,x^3 + 3\,a_2\,x^2\,y + 3\,a_1\,x\,y^2 + a_0\,y^3$, nous avons:

$$\mu_1\nu_1^2\mu_2\nu_2^2\mu_3\nu_3^2 + \mu_1\nu_1 + \mu_1^3\nu_3^3 =< U(f) \mid P >= a_1^3 + a_0a_3 \quad .$$

Définition 3.2.5. Soient $v = \binom{v_1}{v_2}$ et $w = \binom{w_1}{w_2}$ deux vecteurs de dimension 2. Un *crochet* $[v, w]$ est définie par :

$$[v, w] \;=\; v_1 w_2 - v_2 w_1 \;=\; \det \begin{pmatrix} v_1 & w_1 \\ v_2 & w_2 \end{pmatrix} \;.$$

Pour $i, j, u \in \mathcal{R}$, $[i, j] = \mu_i \nu_j - \nu_i \mu_j$ et $[i, u] = \mu_i y - \nu_i x$.

Notation 3.2.6. Le changement linéaire de variables (c_{ij}) est noté $[c, d]$ avec $c = \binom{-c_{21}}{c_{11}}$ et $d = \binom{-c_{22}}{c_{12}}$. Dans ce cas et pour tout entier i de $\mathcal{R}$, la différence $[i, c]$ est égale à $\mu_i c_{11} + \nu_i c_{21}$ et $[i, d] = \mu_i c_{12} + \nu_i c_{22}$.

Proposition 3.2.7 (Kung-Rota). *Soient (c_{ij}) un changement linéaire de variables noté par $(c_{ij}) = [c, d]$ et soit une représentation symbolique :*

$$J(A_0, A_1, \ldots, A_n, X, Y) =< U \mid P(\mu_1, \nu_1, \mu_2, \nu_2, \ldots, x, y) >\in \mathcal{P}$$

Alors :

$$J(\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}, \bar{Y}) =< U \mid P([1, c], [1, d], [2, c], [2, d], \ldots, \frac{[u, c]}{[c, d]}, \frac{[u, d]}{[c, d]}) > .$$

Preuve. voir [47] page 33. □

La notation de la définition 3.2.5 est à la base de la représentation symbolique d'un polynôme $J(\bar{A}_0, \bar{A}_1, \ldots, \bar{A}_n, \bar{X}, \bar{Y})$ en fonction de $J(A_0, A_1, \ldots, A_n, X, Y)$ donnée par la proposition 3.2.7. Il existe donc une représentation symbolique d'un covariant J en polynômes fonction de différences. Un algorithme de calcul d'une représentation symbolique d'un covariant sera présenté dans la section suivante (voir [47]).

3.2.1 Les polynômes-différences symétrisés

Un *monôme-différence* est un élément M non constant de $\mathcal{U}$ qui s'écrit sous la forme de produit fini de crochets $[v, w]$ de lettres symboliques ($v, w \in \mathcal{R}$).

Exemple 3.2.8. Le produit de crochets

$$[3, 2]\,[4, 1]\,[6, u] = (\mu_3 \nu_2 - \mu_2 \nu_3)(\mu_4 \nu_1 - \mu_1 \nu_4)(\mu_6 y - \nu_6 x)$$

est un monôme-différence.

Définitions 3.2.9. La *multiplicité* d'un entier i dans un monôme-différence M est le nombre de crochets de M contenant i. Le monôme-différence M est dit *régulier de degré* d si pour tout i apparaissant dans les crochets de M, la multiplicité de i est égale à d.

Définitions 3.2.10. Le *poids* d'un monôme-différence M est le nombre de crochets de M qui ne contiennent que des entiers de $\mathcal{R}$. L'*ordre* de M est le nombre de crochets de M qui contiennent la lettre u. La *Longueur* de M est le nombre total de crochets dans M.

Exemple 3.2.11. Soit $M = [4,2]\,[3,1]\,[4,5]\,[7,u]$ un monôme-différence. Le poids de M est égal à 3 et son ordre est égal à 1. La multiplicité de 1 dans M est égale à 1, la multiplicité de 4 dans M est égale à 2 et nous déduisons que le monôme M n'est pas régulier.

Définitions 3.2.12. Un *polynôme-différence* est une combinaison linéaire fini de monômes-différences. Ces polynômes forment un sous-espace $\mathcal{B}$ de $\mathcal{U}$. Les polynômes-différences qui sont combinaisons linéaires de monômes-différences de poids g sont dits *polynômes-différences de poids g*.

Définitions 3.2.13. Soit P un polynôme de $\mathcal{B}$. Le polynôme-différence P est dit *régulier de degré d* si chacun de ses monômes est régulier de degré d.

Le théorème suivant nécessite que le corps k soit de caractéristique nulle (voir [80] et [47]).

Théorème 3.2.14 (Le premier théorème fondamental). *Soit P un élément de $\mathcal{B}$ qui est polynôme-différence de poids g. L'évaluation symbolique $< U \mid P >$ de P est un covariant de poids g.*
Réciproquement, Si J est un covariant de poids g de $\mathcal{P}$ alors, il existe un polynôme-différence $P \in \mathcal{B}$ de poids g tel que $J =< U \mid P >$.

Exemple 3.2.15. Posons $n = 3$ et soit $f(x,y) \in \mathcal{F}_3$. Soient $M = [1,u]^2[2,3]^2$ un monôme-différence et $N = \sum_{\sigma\in S_3} \sigma.M$ un polynôme-différence symétrique. Le polynôme N égal à $\sum_{\sigma\in S_3}(\mu_{\sigma(1)}y - \nu_{\sigma(1)}x)^2(\mu_{\sigma(2)}\nu_{\sigma(3)} - \nu_{\sigma(2)}\mu_{\sigma(3)})^2$, correspond au covariant appelé le *Hessien* et noté H_x :

$$H_x = (A_0A_2 - A_1^2)X^2 + (A_0A_3 - A_1A_2)XY + (A_1A_3 - A_2^2)Y^2 \qquad .$$

Remarque 3.2.16. Les coefficients d'un covariant classique par rapport à x sont des semi-invariants.

3.2.2 Représentation symbolique des invariants classiques

Le groupe symétrique S_d de degré d agit naturellement sur les entiers de $\mathcal{R}$: pour tout entier $i \in \mathcal{R}$ et pour toute permutation $\sigma \in S_d$, $\sigma(i)$ désigne l'image de i par σ.

D'après le théorème 3.2.14, un polynôme homogène J de $\mathcal{P}$ admet pour représentation symbolique un polynôme de $\mathcal{U}$. La forme générale d'une représentation symbolique du monôme $A_0^{d_0}A_1^{d_1}\ldots A_n^{d_n}X^{e_1}Y^{e_2}$ est donnée par :

$$(< U \mid \prod_{k=0}^{n} \mu_{i_1}^k \nu_{i_1}^{n-k} \dots \mu_{i_{d_i}}^k \nu_{i_{d_i}}^{n-k} >) < U \mid y^{e1} >< U \mid (-x)^{e_2} > ,$$

où les i_j sont des entiers de deux à deux distincts. Ainsi, une représentation symbolique de $A_0^{d_0} A_1^{d_1} \dots A_n^{d_n} X^{e_1} Y^{e_2}$ est de la forme :

$$(\prod_{k=0}^{n} \mu_{i_1}^k \nu_{i_1}^{n-k} \dots \mu_{i_{d_i}}^k \nu_{i_{d_i}}^{n-k})\, y^{e1}\, (-x)^{e_2} \quad . \tag{3.9}$$

Définition 3.2.17. Un polynôme P de $\mathcal{U}$ est dit *réduit* si pour chaque monôme de P et pour tout entier i apparaissant dans P, le degré total en les variables μ_i, ν_i dans P est égal à n.

Définition 3.2.18. Un *symétrique* $S(P)$ d'un polynôme réduit $P \in \mathcal{U}$ est défini par :

$$S(P) = \frac{1}{d!} \sum_{\pi \in S_d} \pi.P \qquad ,$$

où $\pi.P$ est l'image par la permutation π du polynôme P de $\mathcal{U}$ qui agit sur les d entiers symboliques de P. Le polynôme $S(P)$ est un *polynôme-différence symétrisé* de poids le poids de P.

Proposition 3.2.19. *Soit $J =< U \mid P >$ un invariant classique de $\mathcal{P}$ où P est un polynôme réduit en μ_i, ν_i de $\mathcal{U}$. Alors $J =< U \mid S(P) >$.*

Preuve. D'après la définition 3.2.2, tout covariant classique admet pour représentation symbolique un polynôme réduit. Soit $P \in \mathcal{U}$ tel que $J =< U \mid P >$. Montrer que $J =< U \mid S(P) >$ revient à montrer que $< U \mid M >=< U \mid S(M) >$ pour tout monôme M réduit de $\mathcal{U}$. Supposons $M = \prod_{i=0}^{n} (\mu_i^{k_i} \nu_i^{n-k_i})^{d_i}$ un monôme réduit avec $k_i \in [0, n]$ et $d_i \in \mathbf{N}$. L'évaluation symbolique de M par U est égale à $< U \mid M >= \frac{1}{(n+1)!} \sum_{\sigma \in S_{n+1}} \prod_{i=0}^{n} A_{k_i}^{d_i}$. Soit $S(M)$ le symétrique de M : $S(M) = \frac{1}{(n+1)!} \sum_{\sigma \in S_{n+1}} (\mu_{\sigma(i)}^{k_i} \nu_{\sigma(i)}^{n-k_i})^{d_i}$. L'évaluation symbolique de $S(M)$ est alors égale à celle de M : $< U \mid S(M) >= \frac{1}{(n+1)!} \sum_{\sigma \in S_{n+1}} \prod_{i=0}^{n} A_{k_i}^{d_i} =< U \mid M >$. □

La proposition 3.2.7 réalise le lien entre les covariants J qui vérifient l'équation (3.2) et des polynômes en les variables μ_i, ν_i (et plus exactement les éléments de $\mathcal{B}$). De plus, nous avons le résultat suivant :

Algorithme 3.2.1 (RepréSentationSymbolique). *Cet algorithme réécrit l'algorithme de détermination d'une représentation symbolique d'un covariant classique donné par J.P.S. Kung et G-C. Rota dans [47].*

Fonction `RepréSentationSymbolique(J) ==`

Entrée : Un covariant classique $J \in \mathcal{P}$.
Sortie : Une représentation symbolique $S(P)$ de J : $J =< U \mid P >$
où $S(P)$ est un polynôme-différence symétrisé de $\mathcal{U}$.

1. Pour tout monôme $M = A_0^{d_0}, \ldots, A_n^{d_n} X^{e_1} Y^{e_2}$ de J Faire
2. Remplacer M par $(\prod_{k=0}^{n} \mu_{i_1}^k \nu_{i_1}^{n-k} \ldots \mu_{i_{d_i}}^k \nu_{i_{d_i}}^{n-k})\, y^{e1}\, (-x)^{e_2}$
où les i_j sont des entiers deux à deux distincts ;
3. Calculer le symétrique $S(P)$ de P ;
Fin Pour ;
4. Retourner $S(P)$;

Fin.

Preuve. Nous utilisons dans l'étape 1. de l'algorithme 3.2.1 l'opérateur symbolique U donné dans la définition 3.2.2. Le polynôme $P \in \mathcal{P}$ obtenu dans l'étape 2. est bien une représentation symbolique du covariant J de poids g de départ. En effet, J.P.S. Kung et G-C. Rota ont montré dans [47] que :

$$P([1,c],[1,d],[2,c],[2,d],\ldots,\frac{[u,c]}{[c,d]},\frac{[u,d]}{[c,d]}) = [c,d]^g P(\mu_1,\nu_1,\mu_2,\nu_2,\ldots,\mu_n,\nu_n,x,y) \quad .$$

Cet algorithme prends donc en entrée un covariant J et calcule un polynôme-différence symétrisé $S(P)$ vérifiant $J =< U \mid S(P) >$. □

L'implantation de l'algorithme 3.2.1 au calcul d'une représentation symbolique d'un invariant classique est faite en MAPLE. Elle s'appuie sur la manipulation de listes. En effet, pour faciliter l'automatisation de cet algorithme, les monômes $A_0^{d_0} \ldots A_n^{d_n}$ sont considérés comme des listes $[d_0, \ldots, d_n]$ de longueur $n+1$ (voir annexe C).

Exemple 3.2.20. Pour $n = 2$, le discriminant $D = A_0A_2 - A_1^2$ est un invariant classique. La première étape de l'algorithme 3.2.1 appliqué à D, calcule le polynôme $P = \mu_1^2\nu_2^2 - \mu_1\mu_2\nu_1\nu_2$. La deuxième étape de l'algorithme calcule un symétrique de P qui n'est autre qu'une représentation de D en terme des racines formelles de $f \in \mathcal{F}_n$:

$$S(P) = \frac{1}{2}(\mu_1\nu_2 - \mu_2\nu_1)^2 = \frac{1}{2}[1,2]^2 \qquad \text{et} \qquad < U \mid \frac{1}{2}[1,2]^2 >= A_0A_2 - A_1^2 \quad .$$

Pour $f \in \mathcal{F}_2$, nous avons :

$$< U|\frac{1}{2}(\mu_1\nu_2 - \mu_2\nu_1)^2 >= a_0a_2 - a_1^2 \quad .$$

3.2.3 Invariants de groupes et Invariants classiques

Nous regroupons dans cette section les résultats connues ou moins connues pour exprimer des invariants classiques de $\mathcal{P}$ en fonctions de polynômes-différences de $\mathcal{B}$ qui sont à leur

tour des invariants de sous-groupes de S_n (voir par exemple [16] et [11]).
Soit n un entier non nul. L'ensemble $\mathcal{R}$ sera restreint à l'ensemble fini $\{1, 2, \ldots, n, u\}$ contenant les n premiers entiers positifs et la lettre u. Les variables μ_i, ν_i sont appelées *racines formelles de* $\mathcal{F}_n$.

Soit un covariant homogène $J \in \mathcal{P}$ de degré d, d'ordre t et de poids g et soit $\sum b_j M_j$ une représentation symbolique de J. Le polynôme $\sum b_j M_j$ est supposé être un polynôme-différence réduit. Chaque monôme-différence M_j à un poids égal à g, un ordre égal à t et une longueur $h = g + t$. D'autre part, nous remarquons que dans la représentation symbolique de J : $2h = dn + t$. Ainsi, le poids g d'un covariant homogène peut-être déduit de son degré d et de son ordre t grâce à l'égalité :

$$2g + t = dn \qquad .$$

Cette relation entre le degré, l'ordre et le poids d'un covariant classique est aussi donnée dans [60], [14] et [47].

Théorème 3.2.21. *L'espace vectoriel des covariants homogènes de $\mathcal{P}$ de degré d d'ordre t et de poids $g = \frac{1}{2}(dn - t)$ est isomorphe à l'espace des polynômes-différences symétrisés formés par les n premiers entiers. Chaque entier apparaît d fois et la lettre romaine apparaît t fois.*

Preuve. La preuve est donnée dans [47]. □

Pour avoir le lien entre les invariants classiques et les invariants de groupes, nous allons supposer que pour tout $i \in \mathcal{R}$, $\mu_i = \alpha_i$, $\nu_i = 1$ et $y = 1$. Ainsi, un crochet $[i, j] = \alpha_i - \alpha_j$ et $[u, i] = x - \alpha_i$. Les variables $\alpha_1, \ldots, \alpha_n$ sont considérées comme les racines de $f(x, 1) \in \mathcal{F}_n$:

$$\begin{aligned} f(x,1) &= \sum\nolimits_{k=0}^{n} C_n^k \, a_k \, x^k \\ &= a_n \prod_{i=1}^{n} [u, i] \\ &= a_n \prod_{i=1}^{n} (x - \alpha_i) \qquad . \end{aligned}$$

Il existe une relation entre les coefficients $C_n^k a_{n-k}$ de $f(x, 1)$ et ses racines :

$$C_n^k a_{n-k} = (-1)^k a_n \sum_{1 < i_1 < \ldots < i_k < n} \alpha_{i_1} \ldots \alpha_{i_k} \qquad .$$

Nous déduisons donc l'égalité :

$$a_{n-k} = (-1)^k \frac{a_n}{n!} \sum_{\pi \in S_n} \alpha_{\pi(1)} \ldots \alpha_{\pi(k)} \qquad . \tag{3.10}$$

La substitution (3.11) définit une représentation d'un polynôme de $\mathcal{P}$ en fonction des racines de $f(x,1)$:

$$\begin{array}{lcl} A_n & \leftarrow & a_n \quad , \\ A_{n-k} & \leftarrow & (-1)^k \frac{a_n}{n!} \sum_{\pi \in S_n} x_{\pi(1)} \dots x_{\pi(k)} \quad . \end{array} \tag{3.11}$$

avec $x_1, \dots, x_n$ n variables linéairement indépendantes.

Algorithme 3.2.2 (ReprésentationEnRacines). *Cet algorithme réécrit un polynôme en $A_0, \dots, A_n$ à coefficient dans k en fonction de A_n et des variables $x_1, \dots, x_n$ qui se spécialisent en les racines de $f(x,1)$. Il s'applique en particulier aux invariants classiques.*

Fonction `ReprésentationEnRacines(J) ==`

`Entrée :` Un polynôme J en les variables $A_0, \dots, A_n$ et à coefficient dans k.
`Sortie :` Un polynôme P en les variables $x_1, \dots, x_n$ et A_n.

`1.` P := 0 ;
`2.` Pour tout monôme $\lambda A_0^{d_0} \dots A_n^{d_n}$ de J Faire
`3.` M :=λ ;
Pour k allant de 1 à n Faire
M :=M $\left((-1)^{(n-k)} \frac{a_n}{n!} \sum_{\pi \in S_n} x_{\pi(1)} \dots x_{\pi(n-k)}\right)^{d_k}$;
Fin Pour ;
P :=P + λ_M $A_n^{d_n}$ M ;
Fin Pour ;
`4.` Retourner P ;

Fin.

Exemple 3.2.22. Pour $n = 4$, soit $J = 3A_2^2 - 4A_1A_3 + A_0A_4$ un invariant classique de degré 2 et de poids 4. En utilisant l'algorithme 3.2.2, nous obtenons le polynôme-différence symétrisé $N = \frac{1}{3}J$ avec :

$$N = \left(\frac{A_4}{4!}\right)^2 \sum_{\sigma \in S_4} (x_{\sigma(1)} - x_{\sigma(2)})^2 (x_{\sigma(3)} - x_{\sigma(4)})^2 \quad .$$

Le polynôme-différence symétrisé N est de degré 2 et de poids 4.

Pour exprimer un polynôme régulier de $\mathcal{B}$ symétrique en les racines de f en fonction de covariants, nous utilisons le théorème fondamental des fonctions symétriques et la substitution suivante :

$$\begin{array}{ccc} a_n & \leftarrow & A_n \quad , \\ (-1)^k \frac{a_n}{n!} \sum_{\pi \in S_n} x_{\pi(1)} \dots x_{\pi(k)} & \leftarrow & A_{n-k}. \end{array}$$

De plus, il y a une égalité entre les degrés et poids d'un polynôme-différence symétrisé d'une part et ceux de l'invariant classique associé d'autre part.

Exemple 3.2.23. Soient $n = 8$ et M un polynôme-différence symétrisé régulier de degré $d = 2$ et de poids $g = 8$ sous la forme :

$$M = (\frac{A_8}{8!})^2 \sum_{\sigma \in S_8} (x_{\sigma(1)} - x_{\sigma(2)})^2 (x_{\sigma(3)} - x_{\sigma(4)})^2 (x_{\sigma(5)} - x_{\sigma(6)})^2 (x_{\sigma(7)} - x_{\sigma(8)})^2 \quad .$$

Il existe un invariant classique homogène de degré 2 et de poids 8 égal à M : en utilisant l'algorithme **InvariantClassique** de la section 3.1.3, nous obtenons un invariant classique de degré $d = 2$ et de poids $g = 8$ égal à :

$$P \;=\; 35A_4^2 - 56A_3A_5 + 28A_2A_6 - 8A_1A_7 + A_0A_8 \quad .$$

Le polynôme-différence M est donc un multiple de P et l'algorithme 3.2.2 nous montre que $P = 12600M$. En notant $\alpha_1, \dots, \alpha_8$ les racines de $f(x,1) = \sum_{i=0}^{i=8} C_8^i \; a_i \; x^i$, nous avons :

$$35a_4^2 - 56a_3a_5 + 28a_2a_6 - 8a_1a_7 + a_0a_8 = \frac{(4a_8)^2}{5.8!} \sum_{\sigma \in S_8} (\alpha_{\sigma(1)} - \alpha_{\sigma(2)})^2 \; (\alpha_{\sigma(3)} - \alpha_{\sigma(4)})^2 \; (\alpha_{\sigma(5)} - \alpha_{\sigma(6)})^2 \; (\alpha_{\sigma(7)} - \alpha_{\sigma(8)})^2 \; .$$

3.3 Conclusion

Nous avons présenté dans ce chapitre des techniques connues ou anciennes de calcul d'invariants classiques. Ces invariants (associés à des formes binaires) sont homogènes et isobares et s'expriment en fonction de polynômes-différences symétrisés.
L'algorithme **InvariantClassique** que nous avons implanté et prouvé permet de déterminer des invariants classiques irréductibles qui représentent les premiers éléments (classés selon le degré croissant) d'un système complet d'invariants classiques irréductibles. Notre motivation est le calcul des coefficients de résolvantes particulières (résolvantes associés à des invariants primitifs sous la forme de polynômes-différences : voir chapitre 4).

Le langage a beaucoup évolué dans le domaine des invariants classiques, mais il y a encore des structures algébriques à dégager pour pouvoir progresser dans ce domaine. Par exemple, les algèbres de Hopf, les λ-anneau, les représentations des algèbres de Virasoro permettent de réécrire de manière compacte des multiples formules obtenues par Cayley, Mac Mahon, Sylvester et d'apprécier plus justement l'apport de ces auteurs à la théorie des invariants.

Chapitre 4

Application des Invariants classiques au calcul de résolvantes de Lagrange

Sommaire

Nous allons décrire dans ce chapitre un algorithme de calcul de *résolvantes* qui s'applique aux invariants primitifs s'écrivant sous forme de polynômes-différences symétrisés. C'est un algorithme qui automatise la méthode de Berwick pour le calcul de certaines résolvantes dont les coefficients sont des invariants classiques.

Nous introduisons dans la section 4.1 les notions de base de la théorie de Galois et en particulier les définitions des résolvantes *absolues* et *relatives*. Dans la section 4.2, nous présentons deux méthodes de calcul des résolvantes qui nous ont été inspirées pour la formalisation de la méthode de Berwick. A la section 4.3, nous présentons en application, l'automatisation de la méthode de Berwick pour le calcul de résolvantes par les invariants classiques.

4.1 Notations et Définitions

Le calcul du groupe de Galois d'un polynôme se fait en utilisant des propriétés de groupes et de certains polynômes assez particuliers appelés résolvantes. Le calcul de ces polynômes se fait en calculant d'abord les polynômes invariants primitifs (voir définition 1.1.2 du chapitre 1) : en effet, une résolvante d'un groupe H par rapport à L est un polynôme dont les racines sont les différents conjugués d'un H-invariant L-primitif (les groupes H et L sont deux groupes finis vérifiant $H \subset L$).

Notons $B = \{\sigma_1, \ldots, \sigma_e\}$ une transversale à gauche de L modulo H (e étant l'indice de H dans L). Les classes à gauche de L modulo H sont $\sigma_1 H, \ldots, \sigma_e H$.
Rappelons également que le groupe des permutations S_n agit naturellement sur l'ensemble des variables $\{x_1, \ldots, x_n\}$ et que pour tout $P \in k[x_1, \ldots, x_n]$ et $\sigma \in S_n$, $\sigma.P(x_1, \ldots, x_n) = P(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$.

Notations 4.1.1. L'anneau des polynômes en la variable T et à coefficients dans un corps k est noté $k[T]$ et notons $\alpha_1, \ldots, \alpha_n$ les racines de f dans une clôture classique $\hat{k}$ de k.

Une *relation entre les racines de* f est un polynôme R en n variables à coefficient dans k tel que $R(\alpha_1, \ldots, \alpha_n) = 0$.
Définition 4.1.2. Pour tout polynôme $f \in k[x]$ de degré n et $\alpha = (\alpha_1, \ldots, \alpha_n)$ un n-uplet de $\hat{k}$ formé de ses racines, notons $Gal_k(\alpha)$ le groupe de Galois de α défini par l'ensemble des permutations laissant invariantes les relations entre les racines de f :

$$Gal_k(\alpha) = \{\sigma \in S_n \mid \forall R \in k[x_1, \ldots, x_n] \text{ si } R(\alpha_1, \ldots, \alpha_n) = 0 \text{ alors } \sigma.R(\alpha_1, \ldots, \alpha_n) = 0\} \quad .$$

Fixons f un polynôme de $k[x]$ de degré n et posons $\alpha = (\alpha_1, \ldots, \alpha_n)$ où $\alpha_1, \ldots, \alpha_n$ sont les n racines de f dans $\hat{k}$.
Fixons également L et H deux sous-groupes de S_n vérifiant $H \subset L$ et choisissons Θ un H-invariant L-primitif.

Définition 4.1.3. Soient $H \subset L$ deux groupes de permutations agissant sur l'ensemble des variables $\{x_1, \ldots, x_n\}$ et Θ un H-invariant L-primitif. La *H-résolvante L-relative* de α est définie par :

$$\mathcal{L}^L_{\Theta,\alpha} = \prod_{\sigma \in B} (T - (\sigma.\Theta)(\alpha_1, \ldots, \alpha_n))$$

Une H-résolvante S_n-relative est appelée une *H-résolvante absolue* (parfois notée $\mathcal{L}_{\Theta,f}$ et appelée *résolvante de lagrange*).
Le polynôme $\mathcal{L}^L_\Theta = \prod_{\sigma \in B}(T - (\sigma.\Theta))$ est à coefficient dans $R^L = k[x_1, \ldots, x_n]^L$ et c'est une H-résolvante L-relative dite *générique* .

Théorème 4.1.4. *Soient $f \in k[x]$ un polynôme unitaire de degré n. Soient des sous-groupes $H \subset L \subset S_n$ tels que $Gal_k(\alpha) \subset L$. Posons Θ un H-invariant L-relatif et $\alpha = (\alpha_1, \ldots, \alpha_n)$ les racines de f dans une clôture classique $\hat{k}$ de k . Alors :*

1. $\mathcal{L}^L_{\Theta,\alpha} \in k[T]$,

2. *S'il existe $\sigma \in L$ telle que $Gal_k(\alpha) \subset \sigma H \sigma^{-1}$, alors $(\sigma.\Theta)(\alpha_1, \ldots, \alpha_n) \in k$.*

3. *Réciproquement, si $(\sigma.\Theta)(\alpha_1, \ldots, \alpha_n) \in k$ et $(\sigma.\Theta)(\alpha_1, \ldots, \alpha_n) \in k$ est une racine simple de $\mathcal{L}^L_{\Theta,\alpha}$, alors $Gal_k(\alpha)$ est un sous-groupe de $\sigma H \sigma^{-1}$. Dans ce cas, les racines de f peuvent être ordonnées par $\alpha'_i = \alpha_{\sigma(i)}$ de telle sorte que $Gal_k(\alpha)$ soit un sous-groupe de H.*

Remarque 4.1.5. Il est toujours possible de transformer f (en un polynôme qui a le même groupe de Galois) pour obtenir des résolvantes qui n'admettent pas de racines doubles. Nous appliquons pour cela la *transformation de Tschirnhaus* sur f (voir [32]).

Pour calculer le groupe de Galois d'un polynôme f de degré n, A. Valibouze et J.M. Arnaudiès ont mis au point une méthode déterministe de calcul du groupe de Galois d'un polynôme s'appuyant sur la factorisation de résolvantes et les matrices de partitions. Dans ce cas H est un sous-groupe appelé *groupe test* et le calcul de la résolvante absolue associée à H est d'autant plus rapide que le degré de l'invariant primitif utilisé est petit (voir [73]). Cette méthode a été optimisée : le degré des résolvantes absolues est assez grand et leur factorisation est difficile mais souvent une factorisation partielle suffit (voir le travail de F. Lehobey sur la factorisation de résolvantes [52] et [53]). Une autre méthode appelée *la méthode de Stauduhar* (voir par exemple [66], [19] et [74]) utilise la factorisation de *résolvantes relatives* associées à des H-invariants L-primitifs relatifs ($H \subset L \subset S_n$). Elle utilise le 3. du théorème 4.1.4. Il s'agit de tester si la résolvante a un facteur simple linéaire. Le groupe de Galois est déterminé par inclusion successives dans le graphe des sous-groupes et la résolvante étant calculée par des approximations numériques des racines. La méthode analogue de K. Yokoyama qui consiste à utiliser des valeurs p-adiques pour déterminer si une racine de la résolvante appartient à k est robuste et efficace. Il est donc recommandé, pour calculer le groupe de Galois d'un polynôme, d'avoir plusieurs invariants primitifs relatifs et absolus de degrés petits, mais aussi de savoir calculer des résolvantes relatives et absolues.

4.2 Coefficients des Résolvantes

Les méthodes de calcul des résolvantes dans cette section se basent sur l'étude de leurs coefficients. D'après la définition 4.1.3, les coefficients des résolvantes génériques relatives à un sous-groupe L du groupe S_n (où n est le degré du polynôme dont on cherche le groupe

de Galois) sont des polynômes L-invariants. Nous pouvons alors, comme nous allons le voir à la section 4.2.3, écrire ces invariants en fonction d'invariants fondamentaux (chaque invariant s'exprime comme un polynôme à coefficients dans le corps de base d'invariants primaires et secondaires qui engendrent l'anneau R^L. Voir chapitre 2 et [19]). Une autre méthode plus ancienne utilisée par Cayley et Berwick consiste à déterminer les coefficients de résolvantes en fonction d'invariants classiques.
Précisons que les coefficients d'une résolvante quelconque ne sont pas nécessairement des invariants classiques. Dans la section 4.3, nous verrons qu'il y a suffisamment de telles résolvantes. Nous expliquons également la méthode de Berwick pour le calcul de résolvantes dont les coefficients sont des invariants classiques.

Soient H et L deux sous-groupes de S_n tels que $H \subset L$ et soit Θ un H-invariant L-primitif. D'après le théorème 4.1.4, si $\mathcal{L}^L_{\Theta,\alpha}$ admet une racine simple dans k alors le groupe de Galois G de α sur k est un sous-groupe de H. Considérons le polynôme f suivant :

$$\begin{aligned} f(x) &= (x-x_1)\cdots(x-x_n) \\ &= x^n - e_1x^{n-1} + e_2x^{n-2} - \cdots + (-1)^n e_n. \end{aligned}$$

Nous observons que les coefficients de f en tant que polynôme en x sont symétriques en les autres variables. En effet :

$$\begin{aligned} e_1 &= x_1 + x_2 + \cdots + x_n \\ e_2 &= x_1x_2 + x_1x_3 + \cdots + x_{n-1}x_n \\ &\vdots \\ e_n &= x_1x_2\cdots x_n \ . \end{aligned}$$

Les polynômes $e_1, e_2, \ldots, e_n$ sont appelés *polynômes symétriques élémentaires*, l'anneau des polynômes symétriques en $x_1, \ldots, x_n$ est noté $k[x_1, \ldots x_n]^{S_n}$ et il est engendré par les polynômes symétriques élémentaires. Notons $p_i(x) = x_1^i + x_2^i + \ldots x_n^i$ le polynôme somme de puissances $i$ème. Nous avons le résultat suivant :

Proposition 4.2.1. *Soit k un corps de caractéristique nulle. L'anneau des polynômes symétriques élémentaires est engendré par les n premiers polynômes sommes de puissances :*

$$k[x_1, \ldots, x_n]^{S_n} = k[e_1, \ldots, e_n] = k[p_1, \ldots, p_n]\,. \tag{4.1}$$

Preuve. Voir [77]. □

4.2.1 Résolvantes de Lagrange et invariants fondamentaux

La proposition 4.2.1 montre que les polynômes symétriques élémentaires et les polynômes puissances forment deux bases de l'anneau des polynômes symétriques. Il y a d'autres bases de $k[x_1, \ldots x_n]^{S_n}$ qui contiennent les *polynômes symétriques complets* ou bien les *polynômes de Schur* (les relations entre ses différentes bases sont d'un grand intérêt en algèbre combinatoire [51] ou dans la théorie des représentations). Les polynômes symétriques élémentaires sont également les plus célèbres des invariants fondamentaux.

4.2.2 Les polynômes de Schur

Le calcul des résolvantes absolues génériques se fait généralement en exprimant ses coefficients en fonction d'un nombre fini de polynômes en $x_1, \ldots, x_n$. L'écriture des invariants de degré d (coefficients des résolvantes) se fait ainsi dans une base de l'espace $k[x_1, \ldots, x_n]_d^{S_n}$ des polynômes symétriques de degré d. Notons $k[x_1, \ldots x_n]_d^{S_n}$ l'espace vectoriel des polynômes symétriques homogènes de degré d. Soit $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ une partition de l'entier d avec $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ et soit a_λ un polynôme homogène associé à λ définit par :

$$a_\lambda(x_1, \ldots, x_n) = \det \begin{pmatrix} x_1^{\lambda_1+n-1} & x_2^{\lambda_1+n-1} & \ldots & x_n^{\lambda_1+n-1} \\ x_1^{\lambda_2+n-2} & x_2^{\lambda_2+n-2} & \ldots & x_n^{\lambda_2+n-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{\lambda_n} & x_2^{\lambda_n} & \ldots & x_n^{\lambda_n} \end{pmatrix}.$$

Posons $D = \prod_{1 \leq i < j \leq n}(x_i - x_j)$. Le degré total de $a_\lambda(x_1, \ldots, x_n)$ est $d + \binom{n}{2}$ et les polynômes $s_\lambda = a_\lambda / D$ sont des polynômes symétriques homogènes de degré d appelés les *polynômes de Schur associés à la partition* λ.

Remarque 4.2.2. Le mot *partition* est utilisé ici en son sens combinatoire classique, qui diffère de celui introduit dans le chapitre 1.

Corollaire 4.2.3. *L'ensemble des polynômes de Schur associés aux partitions de longueur n de d forme une base du sous-espace vectoriel $k[x_1, \ldots x_n]_d^{S_n}$.*

Preuve. Voir [77]. □

Un invariant symétrique de degré d est une combinaison linéaire des polynômes de Schur associés aux partitions de longueur n de degré d. L'identification des coefficients des résolvantes se fait alors suivant leur degré. De la même manière et dans le cas particulier où L est un *groupe de réflexions*, les coefficients de la résolvante $\mathcal{L}_\Theta^L$ sont fonction des invariants primaires $\Pi_1, \ldots, \Pi_n$ de $k[x_1, \ldots, x_n]^L$:

4.2.3 Groupes de réflexions

Une matrice $A \in GL_n(k)$ est une réflexion si elle est diagonalisable et si une et une seule de ses valeurs propres n'est pas égale à 1. Un groupe fini H de $GL_n(k)$ est appelé groupe de réflexion s'il est engendré par des réflexions.

Théorème 4.2.4 (Chevalley). *L'anneau des polynômes invariants par un groupe $L \subset GL_n(k)$ est engendré par n polynômes invariants homogènes et algébriquement indépendants si et seulement si L est un groupe de réflexions.*

Preuve. La preuve de la condition nécessaire est donnée dans [69] et la condition suffisante est montrée dans [18]. □

Corollaire 4.2.5. *Soit L un sous-groupe de S_n. L'anneau R^L des polynômes invariants par L est engendré par n invariants homogènes algébriquement indépendants si et seulement si $L = S_{n_1} \times \cdots \times S_{n_s}$ avec $\sum_{i=1}^{i=s} n_i = n$ (L agit par blocs de n_i variables séparées).*

Preuve. Si $L = S_{n_1} \times \cdots \times S_{n_s}$ tels que $\sum_{i=1}^{i=s} n_i = n$ et sachant que L agit par blocs de n_i variables séparées (c'est-à-dire que S_{n_i} agit sur les $x_{n_1+\cdots+n_{i-1}+k}$ avec $k \in [1, n_i]$) alors $R^{S_{n_i}}$ est engendré par la famille E_{n_i} des polynômes symétriques élémentaires $e_1, \ldots, e_{n_i}$ qui agissent sur les variables $x_{n_1+\cdots+n_{i-1}+1}, \ldots, x_{n_1+\cdots+n_{i-1}+n_i}$. L'anneau R^L est par conséquent engendré par $E_{n_1}, \ldots, E_{n_s}$ qui sont algébriquement indépendants : $R^L = k[x_1, \ldots, x_n]^L = k[E_{n_1}, \ldots, E_{n_s}]$. La preuve de la condition nécessaire est dans [21]. □

D'une façon générale et pour un sous-groupe L de S_n, A. Colin propose de calculer la résolvante $\mathcal{L}^L_{\Theta,\alpha}$ en fonction des invariants fondamentaux. En effet, en notant $\Theta_i = \sigma_i.\Theta$ les e conjugués de Θ où $\{\sigma_1, \ldots, \sigma_e\}$ est une transversale à gauche de L par rapport à H et d'après la remarque 4.1.3, les coefficients de $\mathcal{L}^L_\Theta$ par rapport à $T^e, T^{e-1}, T^{e-2} \ldots$ sont respectivement égaux à :

$$\begin{aligned}
\mathrm{coeff}_e(\mathcal{L}^L_\Theta) &= 1 \quad , \\
\mathrm{coeff}_{e-1}(\mathcal{L}^L_\Theta) &= -\sum_{i=1}^{e} \Theta_i \quad , \\
\mathrm{coeff}_{e-2}(\mathcal{L}^L_\Theta) &= \sum_{1 \le i < j \le e} \Theta_i \Theta_j \quad , \\
&\vdots \\
\mathrm{coeff}_1(\mathcal{L}^L_\Theta) &= (-1)^{e-1} \sum_{1 \le i_1 < \ldots < i_{e-1} \le e} \Theta_{i_1} \cdots \Theta_{i_{e-1}} \quad , \\
\mathrm{coeff}_0(\mathcal{L}^L_\Theta) &= (-1)^e \, \Theta_1 \cdots \Theta_e \quad .
\end{aligned}$$

Les coefficients de $\mathcal{L}^L_\Theta$ sont des polynômes L-invariants. En les exprimant en fonction des invariants fondamentaux Π_i et Σ_i qui génèrent l'anneau $k[x_1, \ldots, x_n]^L$ à l'aide du module « `invar` », nous déduisons la valeur de $\mathcal{L}^L_\Theta$. En spécialisant ensuite $(x_1, \ldots, x_n)$ en $(\alpha_1, \ldots, \alpha_n)$ dans les invariants fondamentaux, nous déduisons de la même manière une spécialisation des coefficients de $\mathcal{L}^L_\Theta$ et donc la valeur de $\mathcal{L}^L_{\Theta,\alpha}$.

L'idée d'utiliser un système fini de polynômes pour déterminer les coefficients des résolvantes de Lagrange (et même des résolvantes relatives) et à la base de l'utilisation des invariants pour le calcul des résolvantes comme nous l'avons vu dans les sections 4.2.2 et 4.2.3. Nous proposons dans ce qui suit une explication de la méthode de Berwick pour la détermination de résolvantes dont les coefficients sont des invariants classiques.

D'une façon générale, nous allons voir qu'il existe un polynôme Θ, invariant sous l'action d'un groupe H par rapport à S_n, sous forme de polynôme-différence, tel que les coefficients de la résolvante $\mathcal{L}^L_\Theta$ soient des invariants classiques :

4.3 L'automatisation de la méthode de Berwick

Soit H un sous-groupe de S_n. La méthode de Berwick pour calculer la résolvante $\mathcal{L}_\Theta^{S_n}$ associée à un H-invariant absolu Θ consiste à exprimer ses coefficients en fonction d'un nombre fini d'invariants classiques. La généralisation de cette méthode et son automatisation sont données par l'algorithme 4.3.1 qui calcule un coefficient de degré l de cette résolvante en fonction d'invariants classiques irréductibles de degrés inférieurs ou égaux à l. Mais il y a d'abord des conditions à satisfaire par par Θ. Remarquons en effet, qu'une fonction polynomiale homogène et symétrique en les racines, par rapport aux différences des racines, n'est en général pas un invariant classique. Ainsi, le polynôme-différence symétrisé de $(x_1 - x_2)^2$ pour $n = 4$:

$$A_4^2((x_1-x_2)^2+(x_1-x_3)^2+(x_1-x_4)^2+(x_2-x_3)^2+(x_2-x_4)^2+(x_3-x_4)^2)$$

est le semi-invariant $3A_3^2 + 8A_4A_2$ mais pas un invariant classique.

4.3.1 Invariants primitifs et semi-invariants

Dans ce qui suit, notons $(Y_{i,j})_{(i,j)\in\{[1,n]^2 | i\neq j\}}$ une famille d'indéterminées. Soit I_β une partie finie de $\{(i,j) \in [1,n]^2 \mid i \neq j\}$. Notons $\mathcal{E} = \{\beta = (\beta_{i,j})_{(i,j)\in I_\beta}\}$.
Rappelons que $f(x,y) = a_n\, x^n + C_n^1\, a_{n-1}\, x^{n-1}\, y + \cdots + C_n^{n-1}\, a_1\, x\, y^{n-1} + a_0\, y^n$. Les α_i représentent génériquement une liste de racines déhomogénéisée de la forme f, c'est-à-dire :

$$f(x,y) = a_n \prod_{i=0}^{i=n} (x - \alpha_i y) \qquad .$$

Les variables A_i sont spécialisées en a_i et les x_i en α_i. Notons le crochet $[i,j] = x_i - x_j$. Nous utilisons le lemme suivant donné dans [60] :

Lemme 4.3.1. *soit P un polynôme homogène de degré d en les variables $(Y_{i,j})_{(i,j)\in\{[1,n]^2 | i\neq j\}}$ définit par :*

$$P = \sum_{\beta\in\mathcal{E}} (\lambda_\beta \prod_{(i,j)\in I_\beta} (Y_{i,j}^{\beta_{i,j}})) \qquad ,$$

avec $\lambda_\beta \neq 0$ pour tout $\beta \in \mathcal{E}$ dans P. Supposons que $F(x_1,\ldots,x_n) = P([i,j]_{(i,j)\in\mathcal{E}})$ soit un polynôme symétrique en $(x_1,\ldots,x_n)$ et régulier (c'est-à-dire que pour tout $\beta \in \mathcal{E}$, le degré de $\prod_{(i,j)\in I_\beta}([i,j]^{\beta_{i,j}})$ par rapport à chaque variable x_l soit un même entier d, indépendant de l et de β).
Alors la fonction polynomiale homogène de degré d de $(A_0,\ldots,A_n)$ définie par

$$I(A_0,\ldots,A_n) = (A_n)^d F(x_1,\ldots,x_n)$$

est un invariant relatif (c'est-à-dire un invariant classique) de f. En conséquence, nous avons $2g = nd$.

Preuve. Le lemme se déduit des faits suivants :

(1) Comme $F(x_1,\ldots,x_n)$ est symétrique de degré d en chaque x_i, il est clair, d'après le théorème fondamental sur les polynômes symétriques, que $I(A_0,\ldots,A_n)$ est une fonction polynomiale homogène de degré d en $(A_0,\ldots,A_n)$. Comme F ne dépend que des crochets $[i,j]$, nous voyons que $I(A_0,\ldots,A_n)$ est un semi-invariant de poids g.

(2) Quand nous remplaçons $(x_1,\ldots,x_n)$ par $(\frac{1}{x_1},\ldots,\frac{1}{x_n})$, d'après la deuxième partie de l'hypothèse, le polynôme $F(x_1,\ldots,x_n)$ se transforme en $\frac{(-1)^g}{(x_1\cdots x_n)^d}F(x_1,\ldots,x_n)$, donc $I(A_0,\ldots,A_n)$ se transforme en $(-1)^g I(A_n,\ldots,A_0)$. D'après les propriétés des semi-invariants, nous déduisons bien que $I(A_0,\ldots,A_n)$ est un invariant classique. □

Théorème 4.3.2. *Soient H un sous groupe de S_n avec $n \geq 5$ (n pair) et k un corps de caractéristique nulle. Dans le k-espace vectoriel des polynômes $f \in k[x]$ de degré $\leq n$, il existe un ouvert de Zariski tel que pour tout f irréductible appartenant à cet ouvert, il existe une résolvante de Lagrange $\mathcal{L}_\Theta^{S_n}$ associée à H dont les coefficients sont des invariants classiques et dont la spécialisation $\mathcal{L}_{\Theta,f}$ à f est séparable.*

Preuve. Soit H un sous-groupe de S_n, de cardinal h. Notons e l'indice de H dans S_n. Nous allons utiliser le lemme 4.3.1 pour construire des invariants primitifs de H qui sont aussi des invariants classiques de f et qui seront en général séparables.
Soit C l'ensemble $\{\{1,2\},\{2,3\},\ldots,\{n-1,n\},\{n,1\}\}$. Soit $P_{2,n}$ l'ensemble des parties à n éléments de l'ensemble des 2-parties de $[1,n]$. Considérons l'action à gauche naturelle de S_n sur $P_{2,n}$. Le stabilisateur S de C est le groupe (diédral de cardinal $2n$) engendré par le n-cycle $(1,2,\ldots,n)$ et la *symétrie centrale* $k \mapsto \overline{-k}$, où $\overline{j}$ désigne, pour tout $j \in \mathbf{Z}$, l'unique élément de $[1,n]$ congru à j modulo n.
Soit X une transversale à gauche de S_n modulo S. Considérons une famille $U = (U_\tau)_{\tau\in X}$ d'indéterminées sur $k(x_1,\ldots,x_n)$. Pour toute classe à gauche C de S_n modulo H, soit le polynôme $\Psi_C(U,x_1,\ldots,x_n) \in k[U,x_1,\ldots,x_n]$ défini par :

$$\begin{aligned}\Psi_C(U,x_1,\ldots,x_n) &= \prod_{s\in C}\Big(\sum_{\tau\in X} U_\tau \prod_{i=1}^{i=n}(x_{s\tau(\overline{i})} - x_{s\tau(\overline{i+1})})^2\Big) \\ &= \prod_{s\in C}\Big(\sum_{\tau\in X} U_\tau \prod_{i=1}^{i=n}(\delta_{s\tau(\overline{i}),s\tau(\overline{i+1})})^2\Big) \quad .\end{aligned} \tag{4.2}$$

Soit $u = (u_\tau)_{\tau\in X}$ une spécialisation de U dans k. Comme $n \geq 5$, le sous-groupe $\bigcap_{\tau\in X}\tau S\tau^{-1}$, qui est distingué et ne contient pas le groupe alterné A_n, est réduit à l'élément neutre. Nous déduisons que lorsque s décrit S_n, les polynômes $\sum_{\tau\in X} U_\tau \prod_{i=1}^{i=n}(\delta_{s\tau(\overline{i}),s\tau(\overline{i+1})})^2$ sont deux à deux distincts. Donc comme k est infini, l'ensemble $\mathcal{D}$ des spécialisations u de U telles que les polynômes $\Psi_C(U,x_1,\ldots,x_n)$ soient tous distincts est contenu dans le complémentaire d'une hypersurface algébrique (c'est-à-dire est dense au sens de Zariski).

Choisissons $u \in \mathcal{D}$, qui restera fixé dans ce qui suit.
Posons $\Theta_u(x_1, \ldots, x_n) = \Psi_H(U, x_1, \ldots, x_n)$ et rappelons que S_n/H dénote l'ensemble des classes à gauches de S_n modulo H. La résolvante de Lagrange

$$\mathcal{L}^{S_n}_{\Theta_u}(T) = \prod_{C \in S_n/H} (T - \Psi_C(U, x_1, \ldots, x_n)) = T^e + \sum_{k=1}^{k=e} I_k(u, \frac{A_0}{A_n}, \ldots, \frac{A_{n-1}}{A_n}) T^{e-k}$$

est attachée au groupe H. Pour tout $k \in [1, e]$, nous déduisons du lemme 4.3.1 que

$$J_k(u, A_0, \ldots, A_n) = (A_n)^{4hk} I_k(u, \frac{A_0}{A_n}, \ldots, \frac{A_{n-1}}{A_n})$$

est un polynôme homogène de degré $4hk$ en $(A_0, \ldots, A_n)$ qui est un invariant classique.
Si maintenant nous spécialisons les A_i en les éléments a_i de k (la forme f étant supposée k-irréductible donc séparable), la résolvante spécialisée $\mathcal{L}_{\Theta_u, f}(T)$ sera certes une résolvante de Lagrange de f associée à H, mais ne sera pas séparable qu'en général. Si nous notons $(\alpha_1, \ldots, \alpha_n)$ une liste des racines de f dans une clôture algébrique $\hat{k}$ de k, la résolvante $\mathcal{L}_{\Theta_u, f}$ est séparable si et seulement si pour toute permutation $s \in H$ distincte de l'identité, il existe au moins une permutation $\tau \in X$ telle que

$$\prod_{i=1}^{i=n} (\alpha_{s\tau(\bar{i})} - \alpha_{s\tau(\overline{i+1})})^2 \neq \prod_{i=1}^{i=n} (\alpha_{\tau(\bar{i})} - \alpha_{\tau(\overline{i+1})})^2 \, .$$

Nous voyons bien qu'en général, cette condition sera réalisée (si elle n'est pas réalisée, il y a des relations particulières entres les α_i). Nous sommes donc assurés que génériquement, quand le degré est pair, il y a bien « suffisamment » de résolvantes à coefficients invariants classiques qui soient séparables. □

Remarque 4.3.3. Le théorème 4.3.2 assure qu'il existe suffisamment d'invariants primitifs sous forme de polynômes-différences dont la résolvante de Lagrange associée soit séparable et admette pour coefficients des invariants classiques.

Ce résultat rend plausible l'existence, pour n'importe quelle forme spécialisée f irréductible, d'au moins une résolvante attachée à H qui soit séparable et dont les coefficients soient des invariants classiques. En fait, il y a bien d'autres constructions possibles que celles ci-dessus pour obtenir un H-invariant primitif Θ conduisant à une telle résolvante (voir exemple 4.3.4).

4.3.2 Calcul de résolvantes par la méthode de Berwick

Supposons $J_1, \ldots, J_r$ r invariants classiques irréductibles de degrés inférieurs ou égaux à l inclus dans un système complet d'invariants classiques irréductibles pour n fixé (voir section 3.1.4 du chapitre 3). Ces polynômes sont obtenus par l'algorithme **Invariant-Classique** du chapitre 3 et vérifient :

(1) tout invariant classique en $(A_0, \ldots, A_n)$ de degré inférieur ou égal à l appartient à $k[J_1, \ldots, J_r]$, et

(2) pour tout $i \in [1,n]$, l'invariant J_i n'appartient pas à $k[(J_j)_{j\in[1,r]\setminus\{i\}}]$.

Pour déterminer l'expression d'un invariant classique I dans $k[J_1,\ldots,J_r]$, nous considérons tous les monômes $M = J_1^{d_1}\ldots J_r^{d_r}$ tels que $\sum_{i=1}^{i=r} d_i = l$. L'invariant classique I est alors égal à $\sum \lambda_M M$ où $\lambda_M \in k$.

L'écriture (1) rappelle le cas des invariants des groupes de réflexions qui sont engendrés par un nombre fini d'invariants primaires, mais aussi les polynômes de Schur qui génèrent des espaces de polynômes symétriques de degré d donné. Si les coefficients de la résolvante $\mathcal{L}_\Theta^L$ sont des invariants classiques alors, en supposant que leur degré ne dépasse pas l, les coefficients de $\mathcal{L}_\Theta^L$ sont obtenus seulement avec $J_1,\ldots,J_r$.
D'après le théorème 4.3.2, il existe Θ un H-invariant S_n-primitif sous forme d'un polynôme-différence (voir égalité (4.2)) tel que $\mathcal{L}_\Theta^{S_n}$ ait comme coefficients des invariants classiques. Soient d le degré de Θ et g son poids (voir définitions 3.2.12). Les e conjugués de Θ sont aussi des polynômes-différences de même degré d et même poids g. Les coefficients de $\mathcal{L}_\Theta^{S_n}$ sont des polynômes-différences symétrisés ; les degrés de $\text{coeff}_{e-1}(\mathcal{L}_\Theta^{S_n})$, $\text{coeff}_{e-2}(\mathcal{L}_\Theta^{S_n})$, $\ldots, \text{coeff}_0(\mathcal{L}_\Theta^{S_n})$ sont respectivement égaux à $d, 2d, \ldots, ed$.

Algorithme 4.3.1 (MéthodeDeBerwick). *Rappelons que les invariants classiques sont des polynômes en les variables $A_0,\ldots,A_n$ et que les variables $a_0,\ldots,a_n$ désignent les coefficients de $f(x,1)$ (voir section 3.1 du chapitre 3).*
L'algorithme **MéthodeDeBerwick** *est une automatisation de la méthode de Berwick pour le calcul de résolvantes dont les coefficients sont des invariants classiques.*

Fonction `MéthodeDeBerwick(Θ) ==`

`Entrée :` Un polynôme H-invariant S_n-primitif homogène Θ sous forme d'un polynôme-différence vérifiant (4.2).
`Sortie :` La résolvante $\mathcal{L}_\Theta^{S_n}$ associée à Θ (et $\mathcal{L}_{\Theta,f}$).

1. Poser d le degré de Θ et e l'indice de H dans S_n ;
   Calculer les e conjugués de Θ : $\Theta_1,\ldots,\Theta_e$;
2. Soit $l = ed$;
   Calculer les r invariants classiques $J_1,\ldots,J_r$ irréductibles de degré $\leq l$ avec **InvariantClassique** ;
3. Pour k allant de 1 à e Faire
       Déterminer $\text{coeff}_{e-k}(\mathcal{L}_\Theta^{S_n})$ en fonction des invariants classiques $J_1,\ldots,J_r$ de degrés $\leq kd$;
   Fin Pour ;
4. Remplacer A_i par a_i dans $\mathcal{L}_\Theta^{S_n}$ $(i \in [1,n])$ pour obtenir $\mathcal{L}_{\Theta,\alpha}^{S_n}$;
   Retourner $\mathcal{L}_\Theta^{S_n}$ et $\mathcal{L}_{\Theta,\alpha}^{S_n}$;

Fin.

Le coefficient $\text{coeff}_{e-k}(\mathcal{L}_\Theta^{S_n})$ est un invariant classique de degré kd. Il est donc clair qu'il peut-être obtenu par une combinaison des invariants classiques irréductibles de degré $\leq kd$. En spécialisant les variables A_i en a_i (voir définition 3.2.3 du chapitre 3), nous exprimons les coefficients de $\mathcal{L}_\Theta^{S_n}$ qui sont des polynômes-différences symétrisés réguliers en les racines de $f(x,1)$ en fonction des coefficients a_i de $f(x,1)$. C'est ce que nous faisons à l'étape `4.` de l'algorithme.

L'exemple suivant reprends les diverses étapes du calcul de Berwick des équations de degré 6 en utilisant la théorie moderne des invariants primaires et secondaires. Il est en effet intéressant de voir ainsi liés sur cet exemple les deux aspects de la théorie des groupes finis et des invariants classiques (c'est-à-dire par rapport au groupe linéaire entier) :

Exemple 4.3.4. Fixons $n = 6$ et $f(x,y) = a_6x^6+6a_5x^5y+15a_4x^4y^2+20a_3x^3y^3+15a_2x^2y^4+6a_1xy^5+a_0y^6$. Il y a 4 sous-groupes maximaux de S_6 d'ordre 48, 72, 120 et 360. Les résolvantes correspondantes sont de degrés respectivement 15, 10, 6 et 2.
D'autre part, un système complet d'invariants classiques irréductibles de $\mathcal{F}_6$ est une famille de polynômes classiques de degrés 2, 4, 6 et 10. Ils interviennent chacun dans les coefficients des résolvantes absolues à calculer. Les calculs effectués dans [11] donnent :

1. Un invariant classique de degré 2 et de poids 6 est égal à :
$$I_2 = -10A_3^2 + 15A_2A_4 - 6A_1A_5 + A_0A_6 \quad ,$$
2. Un invariant classique de degré 4 et de poids 12 est choisi égal à :
$$J_4 = \frac{A_6^4}{1620} \sum_{\sigma\in S_6} (x_{\sigma(1)} - x_{\sigma(2)})^2(x_{\sigma(1)} - x_{\sigma(3)})^2(x_{\sigma(2)} - x_{\sigma(3)})^2 (x_{\sigma(4)} - x_{\sigma(5)})^2(x_{\sigma(4)} - x_{\sigma(6)})^2(x_{\sigma(5)} - x_{\sigma(6)})^2 \quad ,$$
3. Un invariant classique de degré 6 et de poids 18 est égal à $J_6 = -64I_2^3 + 81I_2J_4 + 27000K_6$ où K_6 est un invariant du covariant de degré 4 de la forme :
$$k_0x^4 + 4k_1x^3y + 6k_2x^2y^2 + 4k_3xy^3 + k_4y^4 = (A_2A_6 - 4A_3A_5 + 3A_4^2)x^4 + 2(A_1A_6 - 3A_2A_5 + 2A_3A_4)x^3y + (A_0A_6 - 9A_2A_4 + 8A_3^2)x^2y^2 + 2(A_0A_5 - 3A_1A_4 + 2A_2A_3)xy^3 + (A_0A_4 - 4A_1A_3 + 3A_2^2)y^4 \quad ,$$
et $K_6 = k_0k_2k_4 + 2k_1k_2k_3 - k_0k_3^2 - k_1^2k_4 - k_2^3$,
4. Un invariant classique de degré 10 et de poids égal à $\frac{10n}{2} = 30$ est donné par
$$\Delta = A_6^{10} \prod_{\sigma\in S_6} (x_{\sigma(1)} - x_{\sigma(2)})^2 \quad .$$

Posons H_{120} le sous-groupe maximal de S_6 d'ordre 120 (nous gardons les notations de [11]). Un invariant primitif de H_{120} qui soit polynôme-différence est égal à :
$$\Theta = A_6^2((x_1 - x_2)^2(x_3 - x_4)^2(x_5 - x_6)^2 + (x_1 - x_3)^2(x_5 - x_2)^2(x_6 - x_4)^2 + (x_1 - x_5)^2(x_6 - x_3)^2(x_4 - x_2)^2 + (x_1 - x_6)^2(x_4 - x_5)^2(x_2 - x_3)^2 + (x_1 - x_4)^2(x_2 - x_6)^2(x_3 - x_5)^2) \quad .$$

Il y a 6 conjugués de Θ et la résolvante associée à Θ est de degré 6 et ses coefficients sont des invariants classiques : $\mathcal{L}_\Theta^{S_6} = \prod_{\sigma \in S_6//H_{120}} (T - (\sigma.\Theta))$.
L'invariant primitif Θ est homogène de degré 2 et de poids 6. Le coefficient $\mathrm{coeff}_{6-k}(\mathcal{L}_\Theta^{S_6})$ de T^{6-k} dans la résolvante $\mathcal{L}_\Theta^{S_6}$ est donc de degré $2k$. Ainsi :

$$\begin{aligned}
\mathrm{coeff}_6(\mathcal{L}_\Theta^{S_6}) &= 1 \quad , \\
\mathrm{coeff}_5(\mathcal{L}_\Theta^{S_6}) &= \lambda_0 I_2 \quad , \\
\mathrm{coeff}_4(\mathcal{L}_\Theta^{S_6}) &= \lambda_1 I_2^2 + \lambda_2 J_4 \quad , \\
\mathrm{coeff}_3(\mathcal{L}_\Theta^{S_6}) &= \lambda_3 I_2^3 + \lambda_4 I_2 J_4 + \lambda_5 J_6 \quad , \\
\mathrm{coeff}_2(\mathcal{L}_\Theta^{S_6}) &= \lambda_6 I_2^4 + \lambda_7 I_2^2 J_4 + \lambda_8 I_2 J_6 + \lambda_9 J_4^2 \quad , \\
\mathrm{coeff}_1(\mathcal{L}_\Theta^{S_6}) &= \lambda_{10} I_2^5 + \lambda_{11} I_2^3 J_4 + \lambda_{12} I_2^2 J_6 + \lambda_{13} I_2 J_4^2 + \lambda_{14} J_4 J_6 + \lambda_{15} \Delta \quad , \\
\mathrm{coeff}_0(\mathcal{L}_\Theta^{S_6}) &= \lambda_{16} I_2^6 + \lambda_{17} I_2^4 J_4 + \lambda_{18} I_2^3 J_6 + \lambda_{19} I_2^2 J_4^2 + \lambda_{20} I_2 J_4 J_6 + \\
&\quad \lambda_{21} J_4^3 + \lambda_{22} J_6^2 + \lambda_{23} I_2 \Delta \quad .
\end{aligned}$$

Les constantes λ_i appartiennent au corps des coefficients k. En spécialisant les A_i en les coefficients a_i de $f(x,1)$, nous obtenons la résolvante $\mathcal{L}_{\Theta,f} = \mathcal{L}_{\Theta,\alpha}^{S_6}$; elle est donnée par Mr Gilham dans [11].

4.3.3 Conclusion

Nous avons proposé dans cette deuxième partie de la thèse de calculer des résolvantes en utilisant des invariants classiques. Le principal apport de cette étude est de montrer qu'il existe toujours un invariant (polynôme-différence) dont la résolvante associée ait comme coefficients des invariants classiques. Nous automatisons ainsi la méthode de Berwick au calcul des résolvantes en la généralisant à toutes les résolvantes de Lagrange (Berwick a été un des premiers avec Cayley à utiliser la technique de calcul des résolvantes avec les invariants classiques). Nous avons par la même occasion montré comment combiner les résultats modernes de la théorie des invariants (résolvantes, invariants fondamentaux) avec des invariants classiques (polynômes-différences symétrisés, semi-invariants) pour calculer l'outil de base de la théorie de Galois : les résolvantes.

Nos travaux futurs sur la théorie des invariants classique porteront sur la caractérisation des invariants primitifs dont les coefficients des résolvantes relatives associées sont invariants classiques. La méthode de Berwick sera alors généralisée aux résolvantes relatives.

Remerciements. Je tiens particulièrement à remercier Mr Arnaudiès qui a enrichi cette partie de la thèse par ses pertinentes remarques et son aide notamment dans la preuve du théorème 4.3.2.

Troisième partie

Application à la théorie de Galois

Chapitre 5

Méthode hybride pour le calcul du groupe de Galois

Sommaire

Il s'agit dans cette partie de faire la synthèse des deux premières parties de ce document, en présentant la principale application à ce travail : le calcul du groupe de Galois d'un polynôme irréductible. Nous reprenons pour cela les résultats de l'article publié dans *Notes Informelles de Calcul Formel* (voir [4]).
La section 5.1 présente quelques notations ainsi que la définition du groupe de Galois en tant que sous-groupe du groupe algébrique linéaire $GL_n(k)$. Dans la section 5.2, nous présentons une caractérisation du groupe de Galois et nous introduisons la *méthode de Hacque* donnée dans [37].
Dans la section 5.3, nous présentons la méthode de *GI-complète* pour le calcul du groupe de Galois d'un polynôme. Cette méthode reprends la méthode de *GI* pour le calcul de

l'*idéal des relations* (voir [76]). La méthode GI-complète se base sur l'algorithme 5.3.1 qui calcule le groupe de Galois de f à partir de l'idéal des relations. Dans la section 5.4, nous combinons la méthode de Hacque avec la méthode GI-complète en la *méthode de Hacque effective* afin de rendre le calcul du groupe de Galois plus efficace.

5.1 Groupe de Galois et $GL_n(k)$

Soit K une extension finie du corps k de degré $n > 1$. Fixons $e = (e_0, e_1, \ldots, e_m)$ une base du k-espace vectoriel K tels que $e_0 = 1$ et $m = n - 1$.

5.1.1 Notations et Définitions

L'anneau des k-endomorphismes de K est noté $\mathcal{L}_k(K)$ et notons $GL_k(K)$ l'ensemble des éléments inversibles de $\mathcal{L}_k(K)$. Le groupe de Galois de l'extension $k \mid K$ est par définition le groupe des k-automorphismes de K. Il est noté par $Gal_k(K)$.

Posons $\mathcal{M}_n(k)$ l'anneau des matrices $n \times n$ à coefficients dans k. Notons $M[., e]$, l'isomorphisme d'algèbres qui à tout k-endomorphisme de $\mathcal{L}_k(K)$ associe sa matrice dans la base e :

$$\begin{array}{rccc} M[., e] & : & \mathcal{L}_k(K) & \longrightarrow & \mathcal{M}_n(k) \\ & & g & \mapsto & M[g, e] \end{array}$$

Il est facile de vérifier que le groupe des matrices inversibles de $\mathcal{M}_n(k)$ noté $GL_n(k)$ est isomorphe à $GL_k(K)$.

Pour tout $\lambda \in K$, notons $\widehat{\lambda}$ l'endomorphisme multiplicatif de K dans K qui à x dans K associe $\widehat{\lambda}(x) = x\lambda$. Le corps $\widehat{K} = \{\widehat{\lambda} \in \mathcal{L}_k(K) \mid \lambda \in K\}$ est isomorphe au corps K. Le corps $\mathcal{K} = \{M[\widehat{\lambda}, e] \in \mathcal{M}_n(k) \mid \lambda \in K\}$ est naturellement isomorphe au corps $\widehat{K}$. Nous obtenons ainsi les isomorphismes suivants :

$$K \simeq \widehat{K} \simeq \mathcal{K}.$$

Le groupe des éléments inversibles de K (respectivement $\widehat{K}$) est noté K^* (respectivement $\widehat{K}^*$). Le groupe $\widehat{K}^*$ est isomorphe à K^* et est inclus dans $GL_k(K)$: $\widehat{K}^* = \{\widehat{\lambda} \in GL_k(K) \mid \lambda \in K^*\}$.
Posons $\mathcal{K}^* = \{M[\widehat{\lambda}, e] \in \mathcal{M}_n(k) \mid \lambda \in K^*\}$. Alors $\mathcal{K}^* \subset GL_n(k)$ et nous avons les isomorphismes suivant :

$$K^* \simeq \widehat{K}^* \simeq \mathcal{K}^*.$$

Définition 5.1.1. Soient G et H deux groupes tels que $H \subset G$. Le *normalisateur de H dans G*, noté $Nor[G;H]$, est égal à :

$$Nor[G;H] = \{a \in G \mid aHa^{-1} = H\}.$$

Définition 5.1.2. Soit $g \in GL_k(K)$. L'application g est *K-semi-linéaire* si pour tout $x \in K$ et pour tout $\lambda \in K$, il existe $s \in Gal_k(K)$ tel que $g(x\lambda) = g(x)s(\lambda)$.

5.1.2 Propriétés du groupe de Galois

Proposition 5.1.3. *Le groupe de Galois $Gal_k(K)$ est l'ensemble des éléments $GL_k(K)$ qui soient K-semi-linéaires et qui vérifient $g(1) = 1$.*

$$Gal_k(K) = \{g \in GL_k(K) \mid g \text{ est } K\text{−semi−lineaire et } g(1) = 1\}.$$

Preuve. Nous remarquons que pour tout $g \in Gal_k(K)$, g est K-semi-linéaire et $g(1) = 1$. Réciproquement, soit g une application K-semi-linéaire tel que $g(1) = 1$, montrer que $g \in Gal_k(K)$ revient à montrer que $\forall x \in k$, $g(x) = x$. Pour tout $x \in k$, il existe $s \in Gal_k(K)$ tel que $g(x) = s(x).g(1) = s(x) = x$. Ainsi, $g \in Gal_k(K)$. □

La proposition suivante dont nous donnons ici une preuve directe, est aussi une conséquence immédiate du lemme 2.1 de [36].

Proposition 5.1.4. *Le normalisateur de $\widehat{K}^*$ dans $GL_k(K)$ est égal à l'ensemble des applications de $GL_k(K)$ qui sont K-semi-linéaires.*

$$Nor[GL_k(K); \widehat{K}^*] = \{g \in GL_k(K) \mid g \text{ est } K\text{−semi−lineaire}\}.$$

Preuve. Soit g un k-endomorphisme appartenant à $Nor[GL_k(K); \widehat{K}^*]$. Alors pour tout $\lambda \in K^*$, $g \circ \widehat{\lambda} \circ g^{-1} \in \widehat{K}^*$. Pour tout $\lambda \in K^*$, il existe $\mu \in K^*$ tel que $g \circ \widehat{\lambda} = \widehat{\mu} \circ g$; soit s une application de K^* dans K^* tel que $s(\lambda) = \mu$; s est une application bijective de K^* dans K^* parce qu'elle est surjective de K^* dans K^*. Montrer que g est K-semi-linéaire, revient à montrer que g vérifie $g(\lambda x) = g(x)s(\lambda)$ où $s \in Gal_k(K)$:

(i) Pour tout $\lambda \in K^*$, nous avons $g \circ \widehat{\lambda} = \widehat{s(\lambda)} \circ g$; soit pour tout $x \in K$, $g \circ \widehat{\lambda}(x) = \widehat{s(\lambda)} \circ g(x)$ et donc $g(x\lambda) = s(\lambda)g(x)$.

(ii) Vérifions que la bijection s est un k-morphisme de K.
Soient λ, $\mu \in K$ et $x \in K$ alors, d'après (i), $g(x(\lambda + \mu)) = g(x)s(\lambda + \mu)$. D'autre part, $g(x(\lambda + \mu)) = g(x\lambda) + g(x\mu) = g(x)s(\lambda) + g(x)s(\mu) = g(x)(s(\lambda) + s(\mu))$. Ainsi, comme $g \neq 0$, $s(\lambda + \mu) = s(\lambda) + s(\mu)$.
De la même façon, $g(x(\lambda\mu)) = g(x)s(\lambda\mu)$ et $g(x(\lambda\mu)) = g(x\lambda)s(\mu) = g(x)s(\lambda)s(\mu)$. Puisque $g \neq 0$, nous obtenons $s(\lambda\mu) = s(\lambda)s(\mu)$. Enfin, $g(1) = g(1)s(1)$ et donc $s(1) = 1$.

Ainsi $s \in Gal_k(K)$ et l'application g est bien K-semi-linéaire. Réciproquement, soit g une application K-semi-linéaire. Montrons que pour tout $\lambda \in K^*$, $g \circ \widehat{\lambda} \circ g^{-1} \in \widehat{K}^*$.

Par définition, pour tout $\lambda \in K$ et $x \in K$, il existe $s \in Gal_k(K)$ tel que $g(\lambda x) = g(x)s(\lambda)$. En particulier, pour tout $\lambda \in K^*$ et pour tout $x \in K$, $g \circ \widehat{\lambda} \circ g^{-1}(x) = g \circ \widehat{\lambda}(g^{-1}(x)) = g(g^{-1}(x)\lambda) = g(g^{-1}(x))s(\lambda) = xs(\lambda)$. D'où $g \circ \widehat{\lambda} \circ g^{-1} = \widehat{s(\lambda)}$.
Puisque $\lambda \in K^*$ et $s \in Gal_k(K)$, l'élément $s(\lambda)$ est inversible (i.e. $s(\lambda) \in K^*$) et nous en déduisons que l'application $g \circ \widehat{\lambda} \circ g^{-1} \in \widehat{K^*}$. □

Théorème 5.1.5. *D'après les propositions 5.1.3 et 5.1.4, nous avons :*

$$Gal_k(K) = \{g \in Nor[GL_k(K); \widehat{K^*}] \mid g(1) = 1\}.$$

Grâce à l'isomorphisme $GL_n(k) \simeq GL_k(K)$, le groupe de Galois en tant que sous-groupe de $GL_n(k)$ s'exprime alors sous la forme suivante :

$$Gal_k(K) = \{A \in Nor[GL_n(k); \mathcal{K}^*] \mid A(e_0) = e_0\}.$$

5.2 Groupe de Galois $Gal_k(K)$ sur $GL_n(k)$

Nous présentons dans un premier temps une base de $\mathcal{K}^*$ et une caractérisation des éléments de $Nor[GL_n(k); \mathcal{K}^*]$. Ensuite, nous introduisons le *système de hacque* du groupe de Galois $Gal_k(K)$.

Dans toute cette partie, nous nous donnons une matrice A de $GL_n(k)$.

Lemme 5.2.1. *Si la matrice A de $GL_n(k)$ vérifie $A(e_0) = e_0$ alors elle s'écrit sous la forme :*

$$A = \begin{pmatrix} 1 & \gamma_{1,0} & \cdots & \gamma_{m,0} \\ 0 & \gamma_{1,1} & \cdots & \gamma_{m,1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \gamma_{1,m} & \cdots & \gamma_{m,m} \end{pmatrix} \tag{5.1}$$

où $\gamma_{i,j} \in k$ pour $(i,j) \in [1,m] \times [0,m]$ avec $det(\gamma_{i,j})_{i,j\in[1,m]} \neq 0$.

Preuve. Évident.

□

5.2.1 Caractérisation du groupe de Galois $Gal_k(K)$

Pour $j \in [0,m]$, posons $M_j = M[\widehat{e_j}, e]$ la matrice de l'endomorphisme multiplicatif $\widehat{e_j}$ dans la base e. Soit $\lambda \in K$ et $\lambda_0, \ldots, \lambda_m \in k$ vérifiant $\lambda = \sum_{j=0}^{m} \lambda_j e_j$.
L'écriture

$$M[\widehat{\lambda}, e] = \sum_{j=0}^{m} \lambda_j M_j \tag{5.2}$$

donne une décomposition des éléments du corps $\mathcal{K}$.
Pour que λ appartienne au corps K^*, les λ_j ($j \in [0,m]$) ne doivent pas être tous nuls. Ainsi, $(M_0, M_1, \ldots, M_m)$ est une base de $\mathcal{K}^*$.

La matrice A appartient à $Nor[GL_n(k); \mathcal{K}^*]$ si elle vérifie $A\mathcal{K}^*A^{-1} = \mathcal{K}^*$, ce qui est équivalent à :

$$\forall i \in [1,m]\ ,\ \exists\ (\mu_{i,0}, \ldots, \mu_{i,m}) \in k^n - \{(0,\ldots,0)\}\ \ AM_i = \sum\nolimits_{j=0}^{m} \mu_{i,j} M_j A \qquad . \tag{5.3}$$

Corollaire 5.2.2. *Soit $A \in GL_n(k)$ et soit le système d'équations linéaires* (5.4) *déduit de* (5.3) *:*

$$AM_i = \sum\nolimits_{j=0}^{m} x_{i,j} M_j A\ ,\ i \in [1,m], \tag{5.4}$$

où les $x_{i,j}$ sont des inconnues. La matrice A appartient à $Nor[GL_n(k); \mathcal{K}^*]$ si et seulement si le système d'équations (5.4) *admet au moins une solution $\mu = (\mu_{i,j})_{i\in[1,m],j\in[0,m]}$ dans $k^{m\times n}$.*

Théorème 5.2.3. *La matrice A de $GL_n(k)$ appartient au groupe de Galois $Gal_k(K)$ si et seulement si*

- *elle s'écrit sous la forme* (5.1)*,*
- *le système* (5.4) *admet au moins une solution*

Preuve. D'après le théorème 5.1.5, le lemme 5.2.1 et la section 5.2.

□

Définition 5.2.4. Soit $B = (b_{i,j})_{i,j\in[1,n]}$ une matrice de $\mathcal{M}_n(k)$ où les $b_{i,j}$ sont des inconnues. Posons $X = (x_{i,j})_{i\in[1,m],j\in[0,m]}$ où les $x_{i,j}$ sont aussi des inconnues. Le système d'équations :

$$B(e_0) = e_0 \qquad \text{et} \qquad BM_i = \sum\nolimits_{j=0}^{m} x_{i,j} M_j B\ ,\ i \in [1,m] \tag{5.5}$$

est appelé le *système d'équations du groupe de Galois $Gal_k(K)$ dans la base e.*

Corollaire 5.2.5. *Une matrice A appartient au groupe de Galois $Gal_k(K)$ si et seulement si le système d'équations du groupe de Galois dans la base e* (5.5) *admet une solution $B = A$ et $X = \mu$ pour un certain $\mu \in k^{m\times n}$.*

5.2.2 Système de Hacque de $Gal_k(K)$

Soit $u \in K$ un élément primitif de l'extension K du corps k (i.e. $k[u] = K$) et soit $u^n - (a_m u^m + \cdots + a_1 u + a_0)$ son polynôme minimal.

Pour $j \in [0, m]$, nous pouvons poser $e_j = u^j$. Notons M_0 la matrice identité. Dans la base $(1, u, \ldots, u^m)$, $M_j = {M_1}^j$ où M_1, la matrice de $\widehat{u}$, s'écrit :

$$M_1 = \begin{pmatrix} 0 & 0 & \ldots & 0 & a_0 \\ 1 & 0 & \ldots & 0 & a_1 \\ 0 & 1 & \ldots & 0 & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & a_m \end{pmatrix}$$

Lemme 5.2.6. *Le système d'équations du groupe de Galois* (5.5) *dans la base* $(1, u, \ldots, u^m)$ *est équivalent à :*

$$B(e_0) = e_0 \qquad \text{et} \qquad BM_1 = \sum_{j=0}^{m} x_j M_j B\,. \tag{5.6}$$

Preuve. Le système (5.5) entraîne le système (5.6) est évident. Réciproquement soit $i \in [2, m]$ et supposons que (5.6) soit vérifié. Alors

$$BM_iB^{-1} = (BM_1B^{-1})^i = \left(\sum_{j=0}^{m} x_j M_j\right)^i = \sum_{j=0}^{m} y_j M_j$$

où les y_j appartiennent à $k[x_0, \ldots, x_m]$ car $(M_0, M_1, \ldots, M_m)$ est une base de $\mathcal{K}^*$.

□

Lemme 5.2.7. *Si une matrice* $A \in GL_n(k)$ *et* $\mu = (\mu_0, \ldots, \mu_m)$ *vérifient le système* (5.6)*, (i.e.* $B = A$ *et* $(x_0, \ldots, x_m) = \mu$ *sont des solutions), alors* A *s'écrit sous la forme* (5.1) *et* $\mu_j = \gamma_{1,j}$ *pour* $j \in [0, m]$*.*

Preuve. Il suffit d'exprimer AM_1 et $M_1^j A$ pour $j \in [0, m]$, dans la base $(1, u, \ldots, u^m)$.

□

Théorème 5.2.8 (Hacque). *Une matrice* A *appartient au groupe de Galois* $Gal_k(K)$ *si et seulement si la matrice* A *est inversible et*

$$A = \begin{pmatrix} 1 & \gamma_{1,0} & \cdots & \gamma_{m,0} \\ 0 & \gamma_{1,1} & \cdots & \gamma_{m,1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \gamma_{1,m} & \cdots & \gamma_{m,m} \end{pmatrix} \qquad \text{tel que} \qquad AM_1 = \sum_{j=0}^{m} \gamma_{1,j}\, M_j A$$

avec $\gamma_{i,j} \in k$ *pour* $(i, j) \in [1, m] \times [0, m]$*.*

Définition 5.2.9. Le système $AM_1 = \sum_{j=0}^{m} \gamma_{1,j}\, M_j A$ du théorème de Hacque est appelé *système de Hacque.*

Exemple 5.2.10. Pour $f = x^6 + 2$ de degré $d = 6$, le groupe de Galois est un groupe transitif de S_6 et c'est un sous-groupe de $PGL(2, 5)$. Soient $V = x_3 + 2\,x_2 + 3\,x_1$ un I_6-invariant

$PGL(2,5)$-relatif et $\mathcal{L}_V^{PGL(2,5)}$ la résolvante de f par V.

$$\begin{aligned}\mathcal{L}_V^{PGL(2,5)} &= (T^{12}+15444\,T^6+343064484)(T^{12}-21164\,T^6+188183524)\\ &\quad (T^{12}-572\,T^6+470596)(T^6-3456)^2(T^6+128)^2(T^6+2)^2\\ &\quad (T^{12}+1012\,T^6+19307236)^2(T^6-54)^4\end{aligned}$$

Cette résolvante a un facteur irréductible simple F qui n'est autre que le polynôme minimal de l'extension galoisienne de $\mathbf{Q}$. Soit $F = T^{12} - 572\,T^6 + 470596$. Alors $n = 12$ et :

$$M_1 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & -470596 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & 0 & 0 & 572 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \end{pmatrix}$$

Le système de Hacque a 144 équations et autant d'inconnues. Les matrices A de $Gal_k(K)$ s'écrivent sous la forme :

$$A := \begin{bmatrix} 1 & \gamma_{1,0} & \gamma_{2,0} & \gamma_{3,0} & \gamma_{4,0} & \gamma_{5,0} & \gamma_{6,0} & \gamma_{7,0} & \gamma_{8,0} & \gamma_{9,0} & \gamma_{10,0} & \gamma_{11,0} \\ 0 & \gamma_{1,1} & \gamma_{2,1} & \gamma_{3,1} & \gamma_{4,1} & \gamma_{5,1} & \gamma_{6,1} & \gamma_{7,1} & \gamma_{8,1} & \gamma_{9,1} & \gamma_{10,1} & \gamma_{11,1} \\ 0 & \gamma_{1,2} & \gamma_{2,2} & \gamma_{3,2} & \gamma_{4,2} & \gamma_{5,2} & \gamma_{6,2} & \gamma_{7,2} & \gamma_{8,2} & \gamma_{9,2} & \gamma_{10,2} & \gamma_{11,2} \\ 0 & \gamma_{1,3} & \gamma_{2,3} & \gamma_{3,3} & \gamma_{4,3} & \gamma_{5,3} & \gamma_{6,3} & \gamma_{7,3} & \gamma_{8,3} & \gamma_{9,3} & \gamma_{10,3} & \gamma_{11,3} \\ 0 & \gamma_{1,4} & \gamma_{2,4} & \gamma_{3,4} & \gamma_{4,4} & \gamma_{5,4} & \gamma_{6,4} & \gamma_{7,4} & \gamma_{8,4} & \gamma_{9,4} & \gamma_{10,4} & \gamma_{11,4} \\ 0 & \gamma_{1,5} & \gamma_{2,5} & \gamma_{3,5} & \gamma_{4,5} & \gamma_{5,5} & \gamma_{6,5} & \gamma_{7,5} & \gamma_{8,5} & \gamma_{9,5} & \gamma_{10,5} & \gamma_{11,5} \\ 0 & 0 & \gamma_{2,6} & \gamma_{3,6} & \gamma_{4,6} & \gamma_{5,6} & \gamma_{6,6} & \gamma_{7,6} & \gamma_{8,6} & \gamma_{9,6} & \gamma_{10,6} & \gamma_{11,6} \\ 0 & 0 & \gamma_{2,7} & \gamma_{3,7} & \gamma_{4,7} & \gamma_{5,7} & \gamma_{6,7} & \gamma_{7,7} & \gamma_{8,7} & \gamma_{9,7} & \gamma_{10,7} & \gamma_{11,7} \\ 0 & 0 & \gamma_{2,8} & \gamma_{3,8} & \gamma_{4,8} & \gamma_{5,8} & \gamma_{6,8} & \gamma_{7,8} & \gamma_{8,8} & \gamma_{9,8} & \gamma_{10,8} & \gamma_{11,8} \\ 0 & 0 & \gamma_{2,9} & \gamma_{3,9} & \gamma_{4,9} & \gamma_{5,9} & \gamma_{6,9} & \gamma_{7,9} & \gamma_{8,9} & \gamma_{9,9} & \gamma_{10,9} & \gamma_{11,9} \\ 0 & 0 & \gamma_{2,10} & \gamma_{3,10} & \gamma_{4,10} & \gamma_{5,10} & \gamma_{6,10} & \gamma_{7,10} & \gamma_{8,10} & \gamma_{9,10} & \gamma_{10,10} & \gamma_{11,10} \\ 0 & 0 & \gamma_{2,11} & \gamma_{3,11} & \gamma_{4,11} & \gamma_{5,11} & \gamma_{6,11} & \gamma_{7,11} & \gamma_{8,11} & \gamma_{9,11} & \gamma_{10,11} & \gamma_{11,11} \end{bmatrix}$$

où les $\gamma_{i,j}$ vérifient, entre autres, les équations suivantes qui font partie du système de Hacque :

$$\gamma_{2,0} - \gamma_{1,0}^2 + 470596\gamma_{1,2}\gamma_{1,5} - \gamma_{1,3}(-470596\gamma_{1,4} - 470596\gamma_{1,5}) - \gamma_{1,4}(-470596\gamma_{1,3} - 470596\gamma_{1,4} - 470596\gamma_{1,5}) - \gamma_{1,5}(-470596\gamma_{1,2} - 470596\gamma_{1,3} - 470596\gamma_{1,4} - 470596\gamma_{1,5}) = 0$$

$$\gamma_{2,1} - 2\gamma_{1,0}\gamma_{1,1} + 470596\gamma_{1,3}\gamma_{1,5} - \gamma_{1,4}(-470596\gamma_{1,4} - 470596\gamma_{1,5}) - \gamma_{1,5}(-470596\gamma_{1,3} - 470596\gamma_{1,4} - 470596\gamma_{1,5}) = 0$$

$$\gamma_{2,2} - 2\gamma_{1,0}\gamma_{1,2} - \gamma_{1,1}^2 + 470596\gamma_{1,4}\gamma_{1,5} - \gamma_{1,5}(-470596\gamma_{1,4} - 470596\gamma_{1,5}) = 0$$

$$\gamma_{2,3} - 2\gamma_{1,0}\gamma_{1,3} - 2\gamma_{1,1}\gamma_{1,2} + 470596\gamma_{1,5}^2 = 0$$

$$\gamma_{2,4} - 2\gamma_{1,0}\gamma_{1,4} - 2\gamma_{1,1}\gamma_{1,3} - \gamma_{1,2}^2 = 0$$

$$\gamma_{2,5} - 2\gamma_{1,0}\gamma_{1,5} - 2\gamma_{1,1}\gamma_{1,4} - 2\gamma_{1,2}\gamma_{1,3} = 0$$

$$\gamma_{2,6} - \gamma_{1,1}\gamma_{1,5} - \gamma_{1,2}(\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,3}(\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,4}(\gamma_{1,2} + 572\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,5}(\gamma_{1,1} + 572\gamma_{1,2} + 572\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) = 0$$

$$\gamma_{2,7} - \gamma_{1,2}\gamma_{1,5} - \gamma_{1,3}(\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,4}(\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,5}(\gamma_{1,2} + 572\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) = 0$$

$$\gamma_{2,8} - \gamma_{1,3}\gamma_{1,5} - \gamma_{1,4}(\gamma_{1,4} + 572\gamma_{1,5}) - \gamma_{1,5}(\gamma_{1,3} + 572\gamma_{1,4} + 572\gamma_{1,5}) = 0$$

$$\gamma_{2,9} - \gamma_{1,4}\gamma_{1,5} - \gamma_{1,5}(\gamma_{1,4} + 572\gamma_{1,5}) = 0$$

$$\gamma_{2,10} - \gamma_{1,5}^2 = 0$$
$$\gamma_{2,11} - \gamma_{1,1}\gamma_{1,5} - \gamma_{1,2}(\gamma_{1,4}+\gamma_{1,5}) - \gamma_{1,3}(\gamma_{1,3}+\gamma_{1,4}+\gamma_{1,5}) - \gamma_{1,4}(\gamma_{1,2}+\gamma_{1,3}+\gamma_{1,4}+\gamma_{1,5}) - \gamma_{1,5}(\gamma_{1,1}+\gamma_{1,2}+\gamma_{1,3}+\gamma_{1,4}+\gamma_{1,5}) = 0$$
$$\gamma_{3,0} - \gamma_{1,0}\gamma_{2,0} + 470596\gamma_{1,1}\gamma_{2,11} - \gamma_{1,2}(-470596\gamma_{2,5} - 470596\gamma_{2,10} - 470596\gamma_{2,11}) - \gamma_{1,3}(-470596\gamma_{2,4} - 470596\gamma_{2,5} - 470596\gamma_{2,9} - 470596\gamma_{2,10} - 470596\gamma_{2,11}) - \gamma_{1,4}(-470596\gamma_{2,3} - 470596\gamma_{2,4} - 470596\gamma_{2,5} - 470596\gamma_{2,8} - 470596\gamma_{2,9} - 470596\gamma_{2,10} - 470596\gamma_{2,11}) - \gamma_{1,5}(-470596\gamma_{2,2} - 470596\gamma_{2,3} - 470596\gamma_{2,4} - 470596\gamma_{2,5} - 470596\gamma_{2,7} - 470596\gamma_{2,8} - 470596\gamma_{2,9} - 470596\gamma_{2,10} - 470596\gamma_{2,11}) = 0$$
$$\gamma_{3,3} - \gamma_{1,0}\gamma_{2,3} - \gamma_{1,1}\gamma_{2,2} - \gamma_{1,2}\gamma_{2,1} - \gamma_{1,3}\gamma_{2,0} + 470596\gamma_{1,4}\gamma_{2,11} - \gamma_{1,5}(-470596\gamma_{2,5} - 470596\gamma_{2,10} - 470596\gamma_{2,11}) = 0$$
$$\gamma_{3,4} - \gamma_{1,0}\gamma_{2,4} - \gamma_{1,1}\gamma_{2,3} - \gamma_{1,2}\gamma_{2,2} - \gamma_{1,3}\gamma_{2,1} - \gamma_{1,4}\gamma_{2,0} + 470596\gamma_{1,5}\gamma_{2,11} = 0$$
$$\gamma_{3,5} - \gamma_{1,0}\gamma_{2,5} - \gamma_{1,1}\gamma_{2,4} - \gamma_{1,2}\gamma_{2,3} - \gamma_{1,3}\gamma_{2,2} - \gamma_{1,4}\gamma_{2,1} - \gamma_{1,5}\gamma_{2,0} = 0$$
$$572\gamma_{6,5} + \gamma_{11,5} - \gamma_{1,0}\gamma_{11,5} - \gamma_{1,1}\gamma_{11,4} - \gamma_{1,2}\gamma_{11,3} - \gamma_{1,3}\gamma_{11,2} - \gamma_{1,4}\gamma_{11,1} - \gamma_{1,5}\gamma_{11,0} = 0$$
$$572\gamma_{6,6} + \gamma_{11,6} - \gamma_{1,0}\gamma_{11,6} - \gamma_{1,1}(\gamma_{11,5} + 572\gamma_{11,11}) - \gamma_{1,2}(\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,3}(\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,4}(\gamma_{11,2} + 572\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,8} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,5}(\gamma_{11,1} + 572\gamma_{11,2} + 572\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,7} + 572\gamma_{11,8} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) = 0$$
$$572\gamma_{6,7} + \gamma_{11,7} - \gamma_{1,0}\gamma_{11,7} - \gamma_{1,1}\gamma_{11,6} - \gamma_{1,2}(\gamma_{11,5} + 572\gamma_{11,11}) - \gamma_{1,3}(\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,4}(\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,5}(\gamma_{11,2} + 572\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,8} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) = 0$$
$$572\gamma_{6,8} + \gamma_{11,8} - \gamma_{1,0}\gamma_{11,8} - \gamma_{1,1}\gamma_{11,7} - \gamma_{1,2}\gamma_{11,6} - \gamma_{1,3}(\gamma_{11,5} + 572\gamma_{11,11}) - \gamma_{1,4}(\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,10} + 572\gamma_{11,11}) - \gamma_{1,5}(\gamma_{11,3} + 572\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,9} + 572\gamma_{11,10} + 572\gamma_{11,11}) = 0$$
$$572\gamma_{6,9} + \gamma_{11,9} - \gamma_{1,0}\gamma_{11,9} - \gamma_{1,1}\gamma_{11,8} - \gamma_{1,2}\gamma_{11,7} - \gamma_{1,3}\gamma_{11,6} - \gamma_{1,4}(\gamma_{11,5} + 572\gamma_{11,11}) - \gamma_{1,5}(\gamma_{11,4} + 572\gamma_{11,5} + 572\gamma_{11,10} + 572\gamma_{11,11}) = 0$$
$$572\gamma_{6,10} + \gamma_{11,10} - \gamma_{1,0}\gamma_{11,10} - \gamma_{1,1}\gamma_{11,9} - \gamma_{1,2}\gamma_{11,8} - \gamma_{1,3}\gamma_{11,7} - \gamma_{1,4}\gamma_{11,6} - \gamma_{1,5}(\gamma_{11,5} + 572\gamma_{11,11}) = 0$$
$$572\gamma_{6,11} + \gamma_{11,11} - \gamma_{1,0}\gamma_{11,11} - \gamma_{1,1}(\gamma_{11,5} + \gamma_{11,10} + \gamma_{11,11}) - \gamma_{1,2}(\gamma_{11,4} + \gamma_{11,5} + \gamma_{11,9} + \gamma_{11,10} + \gamma_{11,11}) - \gamma_{1,3}(\gamma_{11,3} + \gamma_{11,4} + \gamma_{11,5} + \gamma_{11,8} + \gamma_{11,9} + \gamma_{11,10} + \gamma_{11,11}) - \gamma_{1,4}(\gamma_{11,2} + \gamma_{11,3} + \gamma_{11,4} + \gamma_{11,5} + \gamma_{11,7} + \gamma_{11,8} + \gamma_{11,9} + \gamma_{11,10} + \gamma_{11,11}) - \gamma_{1,5}(\gamma_{11,1} + \gamma_{11,2} + \gamma_{11,3} + \gamma_{11,4} + \gamma_{11,5} + \gamma_{11,6} + \gamma_{11,7} + \gamma_{11,8} + \gamma_{11,9} + \gamma_{11,10} + \gamma_{11,11}) = 0$$

...

Dans ce cas, et grâce à une identification entre les générateurs des groupes candidats et le système de Hacque, nous déduisons que le groupe de Galois est isomorphe au groupe Diédral de cardinal 12 dans S_6.

La méthode de Hacque nécessite le calcul du polynôme minimal de l'extension galoisienne engendrée par les racines d'un polynôme f de degré d afin de calculer le groupe de Galois $Gal_k(K)$ de f. Ce calcul réalisé directement, par une résolvante dite de Galois, est connu comme exponentiel (en $d!$) (voir [78] et [48]). D'autre part, comme le polynôme minimal est de degré l'ordre du groupe de Galois, il n'est pas intéressant d'utiliser la méthode de Hacque spécialement lorsque l'ordre du groupe de Galois est trop gros.
Nous introduisons dans la section suivante la méthode de *GI-complète* qui a pour avantage de permettre, au bout d'un certain nombre d'étapes d'obtenir au moins l'un des résultats suivants :

- le polynôme minimal cherché, en évitant le calcul exponentiel de la résolvante de Galois,
- le groupe de Galois de f,
- la détermination de l'extension galoisienne (ce qui est équivalent à la détermination de l'idéal des relations et entraîne la détermination presque immédiate du groupe de Galois.

5.3 La méthode GI-complète

Soit f un polynôme à coefficient dans k et de degré d et Ω_f un n-uplet des racines de f dans une clôture algébrique $\hat{k}$ de k. D'une façon générale $\alpha = (\alpha_1, \ldots, \alpha_d)$ désigne un d-uplet d'éléments de $\hat{k}$.

5.3.1 Idéaux de Galois et Groupe de décomposition

Rappelons que S_d désigne le groupe symétrique de degré d. Pour tout $\sigma \in S_d$ et $\alpha = (\alpha_1, \ldots, \alpha_d) \in \hat{k}^d$ posons $\sigma.\alpha = (\alpha_{\sigma(1)}, \ldots, \alpha_{\sigma(n)})$. Notons $k[T]$ l'anneau des polynômes en la variable T à coefficients dans le corps k et rappelons que $k[x_1, \ldots, x_d]$ désigne l'anneau des polynômes en les variables $x_1, \ldots, x_d$ à coefficients dans le corps k. Soient J un ensemble de polynômes de $k[x_1, \ldots, x_d]$ et $\sigma \in S_d$. Alors $\sigma(J) = \{\sigma.P \mid P \in J\}$.

Définition 5.3.1. L'*idéal des Ω_f-relations*, noté I_{Ω_f}, est défini par :

$$I_{\Omega_f} = \{P \in k[x_1, \ldots, x_d] \mid P(\Omega_f) = 0\} .$$

Définition 5.3.2. Le groupe de Galois de Ω_f est défini par :

$$G_{\Omega_f} = \{\sigma \in S_d \mid \forall P \in I_{\Omega_f},\ \ \sigma.P(\Omega_f) = 0\} .$$

Un polynôme $P \in k[x_1, \ldots, x_d]$ est une *α-relation* si $P(\alpha) = 0$.

Définition 5.3.3. Soit L un sous-groupe de S_d. L'*idéal I_α^L des α-relations L-invariantes* est défini par :

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

L'idéal $I_\alpha^{S_d}$ est appelé *idéal des relations symétriques* et, d'après la définition 5.3.1, I_α est l'*idéal des α-relations* associé au groupe identité de S_d.

Exemple 5.3.4.

Définition 5.3.5. Le *groupe de décomposition $Gr(I)$ d'un idéal $I \subset k[x_1, \ldots, x_d]$* est défini par :

$$Gr(I) = \{\sigma \in S_d \mid \sigma(I) = I\} \qquad .$$

Remarque 5.3.6. $Gr(I)$ est un groupe et il est facile de montrer l'égalité suivante :

$$Gr(I) = \{\sigma \in S_d \mid \sigma(I) \subset I\} \qquad .$$

Remarque 5.3.7. D'après la définition 5.3.2, le groupe de Galois de Ω_f est le groupe de décomposition de l'idéal I_{Ω_f} des Ω_f-relations :

$$G_{\Omega_f} = \{\sigma \in S_d \mid \sigma(I_{\Omega_f}) = I_{\Omega_f}\} = Gr(I_{\Omega_f}) \qquad .$$

Remarque 5.3.8. L'idéal des Ω_f-relations vérifie $I_{\Omega_f}^{G_{\Omega_f}} = I_{\Omega_f}$ et si $H \subset G_{\Omega_f}$ alors $I_{\Omega_f}^H = I_{\Omega_f}$.

5.3.2 Détermination du Groupe de décomposition d'un idéal

Théorème 5.3.9. *Soient $g_1, \ldots, g_s$ des générateurs de l'idéal I dans $k[x_1, \ldots, x_d]$. le groupe de décomposition $Gr(I)$ de l'idéal I est le plus grand groupe G de S_d vérifiant :*

$$\forall\, i \in [1, s] \text{ et } \forall\, j \in [1, r] \quad \tau_j \,.\, g_i \in I \;, \tag{5.7}$$

où $\tau_1, \ldots, \tau_r$ sont les générateurs de G.

Preuve. Soit $\sigma \in Gr(I)$ alors $\sigma(I) = I$. En particulier pour chaque générateur g_i de I, nous avons $\sigma.g_i \in I$. Ainsi, $Gr(I)$ vérifi la condition (5.7). Reste à prouver que $Gr(I)$ est le plus grand groupe vérifiant la condition (5.7).
Soit $\tau \in S_d$ tel que :

$$\forall\, i \in [1, s] \;,\; \tau.g_i \in I \tag{5.8}$$

Soit $g \in I$, alors il existe $t_1, \ldots, t_d$ dans $k[x_1, \ldots, x_d]$ tels que $g = t_1 g_1 + \cdots + t_d g_d$. De la même manière, $\tau.g$ est une combinaison linéaire de $\tau.g_1, \ldots, \tau.g_d$ sur $k[x_1, \ldots, x_d]$. En utilisant l'identité (5.8), nous avons $\tau.g \in I$ pour tout $g \in I$. Alors $\tau \in Gr(I)$. □

Nous proposons un algorithme qui calcule le groupe de décomposition d'un idéal. Les calculs intermédiaires du Groupe de décomposition d'un idéal utilisent la notion de *base de Gröbner* d'un idéal de $k[x_1, \ldots, x_d]$ qui est un ensemble canonique de générateurs relativement à un ordre donné sur les monômes $x_1, \ldots, x_d$.
La théorie des bases de Gröbner est omniprésente en Calcul formel pour les problèmes traitant d'idéaux. Cette théorie fut développée par B. Buchberger dans [12] et des avancées remarquables sont donnés dans [25] et [26]. La principale application des bases de Gröbner utilisée dans cette section porte sur le problème d'appartenance à un idéal (voir [28]) comme nous allons le voir dans l'algorithme 5.3.1.

Algorithme 5.3.1. *L'algorithme calcule $Gr(I)$ et il est exécuté avec l'appel :* `GDI(Base de Gröbner de` I`, L)` *où I est un idéal et L un groupe de Liste Candidats contenant $Gr(I)$ (voir section 5.3.3).*
Définissons les fonctions utilisées dans l'algorithme :
- La fonction `Retourner` *renvoie le groupe de décomposition de l'idéal I.*
- Soit g une permutation de S_d. La fonction `Ajouter`*(g, G) renvoie le groupe engendré par la permutation g et les permutations de G.*

Fonction `GDI(Base de Gröbner de` I`,` *Liste Candidats*`) ==`

```
Entrée :  Une base de Gröbner g_1,...,g_s de I ;
          Un groupe L contenant Gr(I) ;
Sortie :  Le groupe de décomposition Gr(I) de l'idéal I.
1.        Pour f ∈ I Faire
2.                G := {} ;
3.                Pour g ∈ L Faire
4.                        Si g.f ∈ I Alors
5.                            G := Ajouter(g, G) ;
                          Fin Si ;
                  Fin Pour ;
6.                L := G ;
          Fin Pour ;
7.        Retourner(G) ;
Fin.
```

Preuve. A chaque étape, le groupe L se rapproche un peu plus du groupe de décomposition de I. En effet, Le groupe converge vers un groupe qui laisse fixe l'idéal I de plus, d'après le théorème 5.3.9, l'algorithme `IDG` se termine en un nombre fini d'étapes.
Le test d'appartenance à l'idéal I (voir étape 4.) est possible dès que nous considérons une base de Gröbner de I. En effet, $g.f \in I$ si et seulement si le reste de la réduction de $g.f$ par I est nul.

□

D'après la remarque 5.3.7, l'algorithme 5.3.1 appliqué à l'idéal I_{Ω_f} calcule le groupe de Galois $G_{\Omega_f} = Gal_k(K)$ de f. dans ce cas, l'algorithme `GDI` prends en entrée un groupe L contenant $Gr(I)$ déterminé à partir de la liste *Liste Candidats* donné dans la section 5.3.3.

Pour calculer $Gal_k(K)$ en utilisant la méthode GI-complète, nous devons d'abord déterminer une base de Gröbner de I_{Ω_f}.

Une méthode de calcul de I_{Ω_f} due à Tchebotarev (voir [71]) et à N. Yokoyama [58] consiste à factoriser f dans des extensions successives de k jusqu'au corps de décomposition de f. L'inconvénient de cette méthode est que son coût est très élevé et qu'elle utilise une approche inverse de la méthode de la section 5.3.3 sur les groupes.
La *méthode GI* [76] calcule une chaîne d'idéaux de Galois et en particulier I_{Ω_f}. L'algorithme de calcul d'une base de Gröbner des idéaux de Galois est en $O(d)$. En effet, les idéaux de Galois sont Cohen-Macaulay : la base de Gröbner à d générateurs. De plus, les idéaux de Galois ont la bonne propriété d'avoir des bases de Gröbner pour l'ordre du degré qui le sont aussi pour l'ordre lexicographique. Le calcul des générateurs se fait grâce à une modification de l'algorithme de calcul des ensembles triangulaires donné par P. Aubry (voir [8] et [7]).

Nous présentons dans la section 5.3.3 la méthode GI pour le calcul de l'idéal des relations.

Cette méthode permet aussi de calculer le groupe de Galois d'un polynôme.

5.3.3 Détermination des générateurs de I_{Ω_f} pour le calcul de G_{Ω_f}

La méthode GI calcule les générateurs de l'idéal I_{Ω_f} à l'aide de *résolvantes relatives* en descendant dans le graphe des sous-groupes de S_d jusqu'à arriver à l'idéal des Ω_f-relations donné par le groupe identité.

Soient $\Theta \in k[x_1, \ldots, x_d]$ et L un sous-groupe de S_d contenant le groupe de Galois G_{Ω_f} de f . Une *résolvante L-relative de Ω_f par Θ* est un polynôme sur k donné par :

$$\mathcal{L}^L_{\Theta,\Omega_f} = \prod\nolimits_{\psi \in L.\Theta} (T - \psi(\Omega_f)) \quad .$$

Si $L = \mathcal{S}_d$, alors cette résolvante, notée $\mathcal{L}_{\Theta,f}$, est appelée *H-résolvante absolue de f par Θ*.

Soient H un sous-groupe de S_d tel que $H \subset L$ et Θ un H-invariant L-primitif. L'invariant Θ est dit *L-séparable pour Ω_f* si et seulement si $\Theta(\Omega_f)$ est une racine simple de la résolvante $\mathcal{L}^L_{\Theta,\Omega_f}$.

Notation 5.3.10. Pour un ensemble $E \subset k[x_1, \ldots, x_d]$, l'idéal engendré par E dans $k[x_1, \ldots, x_d]$ est noté $< E >$.

Théorème 5.3.11 (Valibouze). *Soient H et L deux sous-groupes de S_d tels que $H \subset L$ et $G_{\Omega_f} \subset L$. Soient Θ un H-invariant L-primitif L-séparable pour Ω_f et F le polynôme minimal de $\Theta(\Omega_f)$ sur k. Alors*

$$I^H_{\Omega_f} = I^L_{\Omega_f} + < F(\Theta) > \quad .$$

Le polynôme F est un facteur irréductible simple de la résolvante $\mathcal{L}^L_{\Theta,\Omega_f}$.

Preuve. voir **théorème 3.27** dans [76]. □

La méthode GI se base sur le théorème 5.3.11 pour le calcul d'une chaîne d'idéaux de Galois. Elle utilise tout d'abord l'idéal des relations symétriques $I^{S_d}_{\Omega_f}$ et un sous-groupe L contenant le groupe de Galois. Les premières étapes de la méthode GI produisent un idéal de Galois $I^L_{\Omega_f}$ et une liste *Liste Candidats* de sous-groupe de L candidats à être groupe de Galois de f.

Soient H un élément de *Liste Candidats*, Θ un H-invariant L-primitif L-séparable et $\mathcal{L}^L_{\Theta,\Omega_f}$ une H-résolvante L-relative qui vérifient les conditions du théorème 5.3.11. Nous calculons un facteur F de irréductible simple sur k de $\mathcal{L}^L_{\Theta,\Omega_f}$. Nous réduisons ainsi la liste *Liste Candidats* et nous déterminons $I^H_{\Omega_f}$. Si H est un sous-groupe de G_{Ω_f} alors d'après la remarque 5.3.8, nous avons :

$$I_{\Omega_f} = I^L_{\Omega_f} + < F(\Theta) > \quad ,$$

ceci est en particulier vrai lorsque H est égal au groupe identité de S_d.
Si la *Liste Candidats* est réduite à un seul élément, alors c'est le groupe de Galois de f. Sinon, nous calculons une base de Gröbner de I_{Ω_f} et nous appliquons l'algorithme `IDG` à cette base et à *Liste Candidats*.

Remarque 5.3.12. Il est souvent intéressant de calculer l'idéal I_{Ω_f} qui est utilisé pour des calculs sur le corps de décomposition de f. Souvent, la connaissance de I_{Ω_f} est plus instructive que celle de G_{Ω_f}.

5.4 La méthode de Hacque effective

Soit f un polynôme à une variable, séparable, à coefficient dans k et de degré d et K son corps de décomposition de degré n comme supposé dans la section précédente. Nous avons vu dans la section 5.2.2 que la méthode de Hacque ne peut être implantée sans la connaissance d'un polynôme minimal d'un élément primitif de l'extension galoisienne $k \mid K$. Pour que la méthode de Hacque soit efficace, nous cherchons à déterminer dans la section 5.4.1 un facteur particulier de la résolvante de Galois pour déterminer un polynôme minimal de l'extension galoisienne de k.
Le but de la section 5.4.2 est de savoir dans quelles conditions la méthode de Hacque pourra s'appliquer. Il s'agit alors de la combiner avec la méthode GI-complète.

5.4.1 Polynôme minimal d'un élément primitif de $k \mid K$

Pour calculer un polynôme minimal sur k d'un élément primitif de l'extension $k \mid K$, la méthode historique consiste à calculer et à factoriser la *résolvante de Galois du polynôme* f (voir [78]). En effet, tout facteur irréductible simple de cette résolvante est le polynôme minimal d'un élément primitif de K sur k. Cette résolvante étant de degré $d!$, il est bien évident que son calcul est voué à l'échec dès le degré $d = 6$. Or, chaque facteur sur k de la résolvante de Galois suffit à déterminer un polynôme minimal d'un élément primitif de K sur k. L'idée présentée ici, est de calculer un facteur sur k d'une résolvante relative particulière en utilisant les informations données dans les premières étapes de la méthode GI.

Soient $V \in k[x_1, \ldots, x_d]$, I_d le groupe identité de S_d et $H \subset L$ deux sous-groupes de S_d où $G_{\Omega_f} \subset L$. Rappelons qu'une résolvante L-relative de Ω_f par V est un polynôme à une variable T sur k égal à $\mathcal{L}^L_{V,\Omega_f} = \prod_{\psi \in L.V}(T - \psi(\Omega_f))$ et que si $L = S_d$ alors cette résolvante est notée $\mathcal{L}_{V,f}$ et est appelée *résolvante de f par V*.

Remarque 5.4.1. Si V est un H-invariant L-primitif alors la résolvante relative $\mathcal{L}^L_{V,\Omega_f}$ est de degré l'indice de H dans L (noté $[H : L]$) et elle est un facteur de la résolvante absolue $\mathcal{L}_{V,f}$.

Définition 5.4.2. Soit $V \in k[x_1, \ldots, x_d]$. La résolvante $\mathcal{L}_{V,f}$ est dite *résolvante de Galois* si elle n'a que des racines simples et si V est un I_d-invariant S_d-primitif.

Proposition 5.4.3. *Il existe toujours plusieurs polynômes V tels que la résolvante $\mathcal{L}_{V,f}$ soit de Galois et pour un tel V, les racines de la résolvante de Galois associée sont des éléments primitifs de l'extension algébrique $k \mid K$.*

Preuve. Puisque le corps k est un corps parfait infini et f un polynôme irréductible (voir [49] et [29]). □

D'après la proposition 5.4.3, nous pouvons toujours trouver un polynôme $V \in k[x_1, \ldots, x_d]$ tel que $\mathcal{L}_{V,f}$ soit une résolvante de Galois. Soit F un des facteurs irréductibles simples sur k de cette résolvante de Galois. Sans perte de généralités, nous pouvons supposer que $V(\alpha)$ est une des racines du polynôme F sur k et F est son polynôme minimal sur k.
Soit L un sous-groupe de S_d contenant le groupe de Galois G_{Ω_f}. D'après la remarque 5.4.1, le degré de la résolvante relative $\mathcal{L}^L_{V,\Omega_f}$ est égal à l'ordre de L dans S_d et $\mathcal{L}^L_{V,\Omega_f}$ est un facteur sur k de la résolvante de Galois $\mathcal{L}_{V,f}$.
Ainsi, s'il faut éviter de calculer toute la résolvante de Galois, le problème est alors de savoir calculer la résolvante $\mathcal{L}^L_{V,\Omega_f}$ pour un groupe L contenant le groupe de Galois G_{Ω_f}.
Une fois un tel groupe L déterminé, le calcul de $\mathcal{L}^L_{V,\Omega_f}$ est réalisable algébriquement sur machine dès lors que l'ordre du groupe L est assez petit (voir [59] et [52] et les premières étapes de la méthode GI-complète).

5.4.2 La méthode de Hacque effective et la méthode GI-complète

Déterminer un groupe L vérifiant les conditions de la section précédente s'inscrit naturellement dans une méthode dont le but est aussi de calculer le groupe de Galois.
Rappelons que la méthode qui calcule le groupe de Galois d'un polynôme en utilisant les idéaux de Galois est appelée la méthode de GI-complète. Cette méthode calcule d'abord l'idéal des relations en les racines de f (voir [6]) ensuite le groupe de Galois grâce à l'algorithme 5.3.1.
La méthode de Hacque utilise les premières étapes de la méthode GI pour déterminer un polynôme minimal d'un élément primitif de l'extension $k \mid K$ et puis le système de Hacque qui caractérise le groupe de Galois de f.

Il est naturel de comparer cette méthode avec la méthode de GI-complète. Pour cela, supposons que les premières étapes de la méthode GI nous donnent l'idéal de Galois $I^L_{\Omega_f}$ et une liste *Liste Candidats* de groupes candidats à être groupe de Galois et en particulier le groupe L contenant G_{Ω_f}. A ce stade, nous avons le choix entre la méthode de Hacque ou la méthode de GI-complète. Soit V un I_d-invariant L-primitif.

Pour la méthode de Hacque, nous devons calculer et factoriser la résolvante L-relative $\mathcal{L}^L_{V,\Omega_f}$. Soit F un facteur irréductible simple sur k de la résolvante. Nous déterminons le

groupe de Galois grâce à une identification des groupes candidats de *Liste Candidats* avec le système de Hacque de F.
D'autre part, la méthode GI-complète calcule un facteur irréductible simple F sur k de la résolvante L-relative $\mathcal{L}^L_{V,\Omega_f}$. Après le calcul de $I_{\Omega_f} = I^L_{\Omega_f} + < F(V) >$ et de sa base de Gröbner, nous appliquons l'algorithme 5.3.1 pour déduire le groupe de Galois de f.

Comme le polynôme minimal d'un élément primitif de l'extension galoisienne de k est de degré l'ordre du groupe de Galois, nous sommes amenés à utiliser la méthode de Hacque comme dernière étape de la méthode GI plutôt que la méthode GI-complète lorsque le groupe de Galois est de petite taille. Réciproquement, lorsque le groupe de Galois est d'ordre élevé. C'est alors la méthode GI-complète qui sera la moins coûteuse.
Dans ce cas, s'il existe un groupe H de *Liste Candidats* contenu dans L et contenant G_{Ω_f} alors, au lieu de calculer $\mathcal{L}^L_{V,\Omega_f}$, nous allons considérer une résolvante L-relative associée à H : soient Θ un H-invariant L-relatif et $\mathcal{L}^L_{\Theta,\Omega_f}$ la résolvante associée. Le degré de $\mathcal{L}^L_{\Theta,\Omega_f}$ est plus petit que $\mathcal{L}^L_{V,\Omega_f}$ et elle est donc plus facile à factoriser. Nous obtenons de la même manière I_{Ω_f} et nous déduisons après le calcul d'une base de Gröbner de I_{Ω_f} le groupe de Galois de f grâce à `GDI`.

5.5 Exemple de calcul du groupe de Galois pour $d = 8$

Notons $T_1, \ldots, T_{50} = S_8$ les 50 sous-groupes transitifs du groupe symétrique S_8 de degré 8 (à conjugaison près) et rappelons que pour deux sous-groupes H et L de S_8 avec $H \subset L$, un polynôme est dit H-invariant L-primitif si son stabilisateur dans L est H. Posons I_8 le groupe identité de S_8.

Considérons le polynôme irréductible $f(x) = x^8 + 4x + 2$, Ω_f un vecteur de $\mathbf{C}^8$ des racines (distinctes) de f et G_{Ω_f} le groupe de Galois de Ω_f sur $\mathbf{Q}$.

1. Le polynôme f étant irréductible sur $\mathbf{Q}$, la classe de conjugaison de G_{Ω_f} est une des 50 classes de sous-groupes transitifs de S_8.
   Les 8 modules de Cauchy $f_1(x_1), f_2(x_1, x_2), \ldots, f_8(x_1, \ldots, x_8)$ du polynôme f engendrent l'idéal des relations symétriques entre les racines de f. Ce qui nous donne une base de l'idéal de Galois $I^{S_8}_{\Omega_f}$.

   En calculant le discriminant du polynôme f, nous remarquons qu'il se factorise en $2^{19}7^4$. Le groupe de Galois de f est donc un groupe impair puisque le discriminant de f n'est pas carré (ce n'est donc pas un sous-groupe du groupe alterné de S_8). Soit T_{45} le sous-groupe (maximal) d'indice 35 dans $T_{50} = S_8$ (et de cardinal 1152) engendré par les permutations :

$$< (5,6)(7,8), (5,7)(6,8), (6,7,8), (1,2)(3,4), (1,3)(2,4), (7,8), (6,7), (3,4), (2,3), (1,5)(2,6)(3,7)(4,8) > \quad .$$

Le degré des T_{45}-résolvantes absolues est égal à l'indice de T_{45} dans S_8 (i.e. 35). Soit $\Theta_1 = x_1x_2x_3x_4 + x_5x_6x_7x_8$ un T_{45}-invariant S_8-primitif. La factorisation sur $\mathbf{Q}$ de la résolvante $\mathcal{L}^{S_8}_{\Theta_1,\Omega_f}$ est :

$$(T-1)T^8(T^2-8)^5(T^4-8T^2+14)^4$$

(*calculée avec le module SYM du système de calcul formel Maxima à partir de* $I^{S_8}_{\Omega_f}$).

$(T-1)$ est un facteur irréductible simple sur $\mathbf{Q}$: $G_{\Omega_f} \subset T_{45}$. Remarquons que le cardinal de la variété de l'idéal $I^{T_{45}}_{\Omega_f}$ est égal à 1152. À partir des générateurs de $I^{T_{45}}_{\Omega_f} = I^{S_8}_{\Omega_f} + < \Theta_1 - 1 >$ nous calculons, avec le logiciel GB, la base de Gröbner de $I^{T_{45}}_{\Omega_f}$ pour l'ordre lexicographique :

$$< x_8+x_7+x_6+x_5, x_7^2+x_7x_6+x_7x_5+x_6^2+x_6x_5+x_5^2, x_6^3+x_6^2x_5+x_6x_5^2+x_5^3, x_5^4+x_1^4+1, x_4+x_3+x_2+x_1,$$
$$x_3^2+x_3x_2+x_3x_1+x_2^2+x_2x_1+x_1^2, x_2^3+x_2^2x_1+x_2x_1^2+x_1^3, x_1^8+x_1^4+2 >$$

2. Soit le groupe test T_{35}, sous-groupe d'ordre 128 du groupe T_{45} dont $\Theta_2 = x_7x_8 + x_5x_6 + x_3x_4 + x_1x_2$ est un T_{35}-invariant T_{45}-primitif. La résolvante $\mathcal{L}^{T_{45}}_{\Theta_2,\Omega_f}$ calculée à partir de $I^{T_{45}}_{\Omega_f}$ se factorise sur $\mathbf{Q}$ en

$$T(T^8 - 12T^6 - 48T^4 + 192T^2 - 3584) \quad .$$

Alors $G_{\Omega_f} \subset T_{35}$ et une base de Gröbner de l'idéal $I^{T_{35}}$ pour l'ordre lexicographique est égale à :

$$< x_1^8 + x_1^4 + 2, x_2 + x_1, x_3^2 + x_1^2, x_4 + x_3, x_5^4 + x_1^4 + 1, x_6 + x_5, x_7^2 + x_5^2, x_8 + x_7 > \quad .$$

Remarquons que $I^{S_8}_{\Omega_f} \subset I^{T_{45}}_{\Omega_f} \subset I^{T_{35}}_{\Omega_f}$.

3. Soit T_{26} le sous-groupe de T_{35} dont l'ordre est 64 (i.e. d'indice 2 dans T_{35}) et notons Θ_3 un des T_{26}-invariant T_{35}-primitif T_{35}-séparable calculés avec le module GAP `PrimitiveInvariant`. Le calcul de la résolvante $\mathcal{L}^{T_{35}}_{\Theta_3,\Omega_f}$ aboutit à $(T-96)(T+608)$. Nous en déduisons que $G_{\Omega_f} \subset T_{26}$ et :

$$\begin{aligned} I^{T_{26}}_{\Omega_f} \;=\; & < 2x_8 - x_5x_2x_1^7 - x_5x_2x_1^3, x_7 + x_5, 2x_6 + x_5x_2x_1^7 + x_5x_2x_1^3, x_5^4 + x_1^4 + 1, \\ & x_4 + x_1, x_3 + x_2, x_2^2 + x_1^2, x_1^8 + x_1^4 + 2 > \quad . \end{aligned}$$

4. La liste *Liste Candidats* obtenue avec la méthode GI contient à ce stade les sous-groupes impairs de T_{26} :

$$\textit{Liste Candidats} \;= \{T_1, T_7, T_{16}, T_{17}\} \quad .$$

Nous avons le choix entre calculer la I_8-résolvante T_{26}-relative et ensuite calculer le système de Hacque ou encore appliquer la méthode de GI-complète en calculant

l'idéal des relations. Mais puisque le degré de la résolvante à calculer est encore élevé (degré égal à 64), nous utilisons dans ce cas, une résolvante discriminante soit la T_7-résolvante T_{26}-relative. En effet, nous remarquons que $T_1 \subset T_7 \subset T_{16}$ et que $T_7 \subset T_{17}$. Soit Θ un T_7-invariant T_{26}-primitif. La résolvante $\mathcal{L}^{T_{26}}_{\Theta,\Omega_f}$ est un polynôme irréductible. La *Liste Candidats* est donc réduite à $\{T_1\}$ et le groupe de Galois de f est (à conjugaison près) égal à T_1.

5.6 Conclusion

Suivant la complexité du problème, nous choisirons la résolution d'équations grâce au système de Hacque ou la méthode GI-complète. Les deux méthodes sont complémentaires l'une de l'autre selon que le groupe de Galois est d'ordre élevé ou non et elles constituent une approche algébrique du calcul du groupe de Galois.
Il existe d'autres méthodes très efficaces qui se basent sur des méthodes numériques : le travail récent de J. Klüners et K. Geissler [30] abouti au calcul du groupe de Galois de polynômes irréductibles sur $\mathbf{Q}$ de degré $d = 15$ dans le système KASH. Il s'agit de combiner plusieurs méthodes de calcul de groupe de Galois en prenant comme ossature principale celle de R.P. Stauduhar [66] et la méthode p-adique de Yokoyama pour leur approximation des racines [82]. J. Klüners et K. Geissler apportent en plus des améliorations intéressantes pour éviter le calcul complet des résolvantes relatives lorsque des sous-groupes du groupe de Galois sont déterminés (par des techniques modulaires ou des suites de Sturm,...). Ils obtiennent ainsi un programme dont l'efficacité est effectivement remarquable.
Les implantations numériques sont combinés avec des méthodes inspirées des méthodes algébriques de calcul du groupe de Galois comme celle des matrices de partitions [10], [64] et [5]. C'est cette méthode de matrices des partitions (améliorée par les matrices des groupes [74]) qui, combinée avec les idéaux de Galois, forme la méthode GI sur laquelle nous travaillons.
Notons au passage que les méthodes p-adiques et numériques, ne sont plus applicables directement lorsque le corps des coefficients de f n'est plus $\mathbf{Q}$ mais $\mathbf{Q}(x)$ ou tout autre corps. D'autre part, la méthode de Hacque étant une nouvelle approche du calcul du groupe de Galois par une caractérisation grâce à un système d'équations, une étude approfondie du système de Hacque reste à faire pour résoudre certains problèmes de Galois inverse.

En conclusion, chaque méthode connue permettant de calculer le groupe de Galois d'un polynôme a ses faiblesses ; certaines lorsque l'ordre du groupe est élevé, d'autres lorsqu'il est petit, une autre (la numérique) lorsque les racines sont trop rapprochées, d'autres encore lorsque les coefficients ne sont pas numériques ... Il ne faudrait donc pas opposer ces méthodes, mais au contraire, considérer qu'elles se complètent en s'emboîtant algorithmiquement afin d'aboutir à des programmes efficaces.

Conclusion - Perspectives

En étendant les travaux effectués dans le cadre de la théorie des invariants, nous avons présenté dans cette thèse un aperçu détaillé des principales méthodes qui interviennent en théorie de Galois effective : le calcul d'invariants primitifs et le calcul de résolvantes de Lagrange à l'aide des résultats de la théorie des invariants classiques et modernes.

Les algorithmes étudiés dans un premier temps sont des algorithmes de calculs de polynômes invariants primitifs. Nous avons amélioré dans ce contexte la technique de K. Girstmair considérée comme une approche automatique de calcul des invariants primitifs absolus. Nous avons implanté le module « `PrimitiveInvariant` » de calcul de tous les invariants primitifs relatifs ou absolus de petits degrés, dans un système de calcul formel (GAP). Ce module permet de déterminer tous les invariants primitifs réduits relatifs ou absolues.
Nous avons ensuite réalisé une comparaison entre nos algorithmes de calcul et le module `invar` de calcul d'invariants de G. Kemper. Utilisant le système de calcul formel MAGMA, particulièrement bien adapté à la manipulation d'objets algébriques, ce module reste moins efficace que « `PrimitiveInvariant` » en terme de temps de calcul d'invariants primitifs relatifs.

Nous nous sommes intéressés aux invariants classiques et à leur lien avec les invariants de groupes. En accordant une place essentielle à la manipulation de listes (dans l'implantation des polynômes), nous avons prouvé et implanté un algorithme de calcul d'invariants classiques de degré et poids donnés. Nous avons également implanté une méthode ancienne pour exprimer des invariants classiques en fonction de polynômes-différences symétrisés qui soient des invariants de groupes. Nous avons montré qu'il existe suffisammant d'invariants primitifs dont les résolvantes de Lagrange associées aient pour coefficients des invariants classiques. Nous avons enfin présenté de manière automatique la méthode utilisée par E.H. Berwick pour le calcul de ces résolvantes.

Nous avons travaillé dans la troisième partie sur les idéaux de Galois. Nous introduisons la méthode de Hacque effective pour le calcul du groupe de Galois. C'est une méthode hybride qui caractérise le groupe de Galois par un système d'équations en tant que sous-groupe du groupe algébrique linéaire. Même si les résultats obtenus par des méthodes

analytiques (ou numériques) sont souvent plus performants, nous pensons que pour comprendre les phénomènes algébriques, pour maîtriser les structures utilisées en théorie de Galois et pour généraliser les calculs à des corps quelconques (corps non rationnels), les méthodes algébriques restent une bonne solution. De plus, il est toujours possible de les combiner avec d'autres méthodes.

D'autres investigations sur les invariants classiques et modernes feront l'objet de futurs développements et notamment sur une librairie en GAP sur les invariants des groupes de permutations (en collaboration avec Mr. Thiéry [72]). Il serait également intéressant d'étudier les propriétés des invariants dans des domaines d'applications comme l'imagerie (voir [38] et [67]).

Annexe A

Le module « PrimitiveInvariant » sous GAP

Ce module est disponible sur les machines de l'UMS Medicis de l'école polytechnique [1]. C'est un programme écrit en GAP qui généralise un algorithme donné par K. Girstmair pour le calcul d'invariants primitifs absolus et minimaux.

L'utilisation de ce programme est très facile : en effet il suffit d'introduire deux sous groupes L et H du groupe symétrique de degré n vérifiant H inclus dans L.
La fonction « MinimalPrimitiveInvariants(n, L, H) » calcule tous les représentants de polynômes H-invariants L-primitifs de degré minimal.
La fonction « AllPrimitiveInvariants(n, L, H) » calcule tous les représentants des polynômes H-invariants L-primitifs de degrés allant jusqu'à $\frac{n(n-1)}{2}$.
En application à la théorie de Galois et au calcul de résolvantes, n désigne le degré du polynôme f, L le groupe candidat et H le groupe test.

Ce module comprends plusieurs fonctions. Nous commençons par calculer des combinaisons de listes qui représentent les poids des monômes a n variables. Ensuite, nous présentons les fonctions nécessaires au calcul des partitions et des ensembles essentiels.
Les corrections faites le 23/09/1999 pour le passage à la version 4 de GAP portent sur les fonctions `ConcatenationString` et `Sublist`.

A.1 Combinatoire des listes

1. Un type t de m est une liste $(t_1, \ldots, t_r)$ vérifiant $t_1 + \cdots + t_r = m$. La fonction `Calculsoustype(m,r)` calcule tous les types de m de longueur r. La fonction `Calcultype(m)` calcule la liste de tous les types de l'entier m. La longueur varie de 1 a m.

```
Calculsoustype := function(m,r)
```

```
  local resultat, k, sou, t, res, i;
  resultat := [];
  if (r=1) then
    resultat := [[m]];
  elif (r=m) then
      for i in [1..m] do resultat[i] := 1; od;
      resultat := [resultat];
    else
      k := m-1;
      while (k > r-2) do
        sou := Calculsoustype(k,r-1);
        repeat
          if Length(sou)>0 then
            t := sou[1];
            if ( t[1] > m-k ) then
              if Length(sou)>1 then
                sou := sou{[2..Length(sou)]};
              else
                sou:=[];
              fi;
            else
              res := Concatenation([m-k],t);
              Append(resultat,[res]);
              if Length(sou)>1 then
                sou := sou{[2..Length(sou)]};
              else
                sou:=[];
              fi;
            fi;
          fi;
        until (sou=[]);
        k := k-1;
      od;
  fi;
  return(resultat);
end;

Calcultype := function(m)
  local sortie, sar, s, r;
  sortie := [];
  for r in [1..m] do
    sar := Calculsoustype(m,r);
    Append(sortie,sar);
  od;
```

```
  return(sortie);
end;
```

2. La fonction *w(t)* calcule le poids du type $t = (t_1, \ldots, t_r)$. Le poids de t est par définition égal a $w(t) = t_2 + \cdots + (i-1)t_i + \cdots + (r-1)t_r$. La fonction `listpoidstypes(m)` calcule tous les poids possibles et arrange les types de m par ordre croissant suivant leurs poids. `Listpartition` s'applique à un type et une liste d'entiers ordonnés d'une certaine manière et la transforme en partition de $\{1..m\}$. Enfin, la fonction `insert(val,l,i)` fait ce que son nom l'indique, c'est a dire insérer dans la liste `l` à l'indice `i` la valeur `val`.

```
w := function(t)
  local j, somme, e, s;
  j := 1;
  somme := 0;
  s := Length(t);
  if s>1 then
    for e in [2..s] do
      somme := somme + ( t[e] * j );
      j := j + 1;
    od;
  fi;
  return(somme);
end;

Listpoidstypes  := function(m)
  local l, s, pc, rc, t;
  l:=[];
  s := Calcultype(m);
  Sort(s, function(l1,l2) return w(l1)<w(l2); end);
  pc := 0; rc := rec(po := pc , ty := []);
  for t in s do
    if (w(t)<> pc) then
      Add(l,rc);
      pc := w(t);
      rc := rec(po := pc , ty := [t]);
    else
        Add(rc.ty,t);
    fi;
  od;
  Add(l,rc);
  return(l);
end;
```

```
Listpartition := function(t,liste)
  local l, k, res, i, j, cou;
  l := 1;
  k := t[1];
  res := [Set(liste{[l..k]})];
  for i in [2..Length(t)] do
    l := k+1;
    k := t[i]+k;
    cou := Set(liste{[l..k]});
    Add(res,cou);
  od;
  return(res);
end;

insert := function(e,l,i)
  return(Concatenation(l{[1..(i-1)]},[e],l{[i..Length(l)]}));
end;
```

3. La fonction `Setpartition(m,t)` nous donne l'ensemble $T(t)$ de toutes les partitions de m de type t. Les fonctions `OnPart` et `OnSPart`, calculent l'image d'une partition ou d'une liste de partitions par une permutation.

```
Setpartition := function(m,t)
  local temp, s, resultat;
  temp := Arrangements([1..m],m);
  resultat := [];
  for s in temp do
    AddSet(resultat, Listpartition(t,s));
  od;
  return resultat;
end;

OnPart := function (part,g)
  local l1,s;
  l1:=[];
  for s in part do
    Append(l1,[OnSets(Set(s),g)]);
  od;
  return l1;
end;;

OnSPart := function (lpart,g)
```

```
  local l1,s;
  l1:=Set([]);
  for s in lpart do
      Append(l1,[OnPart(s,g)]);
  od;
  return Set(l1);
end;;
```

A.2 Calculs d'orbites de partitions

1. `Orbitepartition` détermine l'orbite d'une partition T par un groupe. L'action est induite par celle du groupe sur une liste d'entiers. `systemeDeRepresentant(m,H,T)` nous donne un système de représentants des orbites de T par H, où les T sont des partition's de type t variant dans l'ensemble $T(t)$. Cette fonction ne donne que les partitions T d'orbites distinctes.

```
Orbitepartition := function(Hrelatif,T)
  local res, g, K;
  res := [];
  K := Elements(Hrelatif);
  for g in K do
    AddSet(res,OnPart(T,g));
  od;
  return(res);
end;

Systemerep := function(m,Hrelatif,t)
  local sortie, don, tempo, cou, T, a;
  sortie := [];
  don := [];
  tempo := Setpartition(m,t);
  for T in tempo do
    cou := Set(Orbit(Hrelatif,T,OnTuplesSets));
    if (cou in don)=false then
      AddSet(sortie,T);
      AddSet(don,cou);
    fi;
  od;
  a:= rec(par :=[], orb := []);
  a.par := sortie;
  a.orb := don;
  return(a);
```

```
end;
```

2. `Partmpoids` calcule toutes les partitions du système de représentants des orbites du groupe $Hrealtif$ de même poids. `Interstabor(G,A)` calcule le groupe $H(A)$, égal à l'intersection des stabilisateurs par rapport a G, de l'orbite de T par H, òu T est dans A.

```
Partmpoids := function(m,Hrelatif,lis)
local sortie, t, cou;
sortie := rec(par :=[], orb := []);
cou := lis.ty;
for t in cou do
sortie.par:=Concatenation(sortie.par,Systemerep(m,Hrelatif,t).par);
sortie.orb:=Concatenation(sortie.orb,Systemerep(m,Hrelatif,t).orb);
od;
return(sortie);
end;
```

```
Interstabor := function(Gabsolu,A)
  local res, K, T, U;
  res := Gabsolu;
  for U in A do
    K := Stabilizer(Gabsolu,U,OnSPart);
    res := Intersection(K,res);
  od;
  return res;
end;
```

3. `Essentielset` calcule des ensembles essentiels. C'est aussi grâce à ces ensembles particuliers qu'on obtient des invariants.

```
Essentielset := function(m,Hrelatif,Gabsolu)
  local U0, G0, G1, G2, A, liste, s;
  liste := Listpoidstypes(m); Print(liste);
  s := liste[1];
  U0 :=Partmpoids(m,Hrelatif,s).par;
  liste := liste{[2..Length(liste)]};
  G0 := Gabsolu;
  G1 := Gabsolu;
  while (G1 <> Hrelatif and liste <>[]) do
```

```
    s := liste[1];
    A := Partmpoids(m,Hrelatif,s);
    G2 := Interstabor(Gabsolu,A.orb);
    G1 := Intersection(G2,G0);
    if (G1 <> G0) then
      if (G1 = G2) then
        U0 := A.par;
      else
        U0 := Concatenation(U0,A.par);
      fi;
      G0 := G1;
    fi;
    liste := liste{[2..Length(liste)]};
  od;
  return(U0);
end;
```

4. Les fonctions suivantes transforment des partitions (respectivement une orbite de partitions) en monôme (resp. en polynômes). `Polynomessentiel` transforme un ensemble essentiel en polynôme.

```
Polynompart := function(T)
  local a, i, j, l, s, z;
  a := "";
  s := 1;
  T := T{[2..Length(T)]};
  if Length(T)<>0 then
    if s=Length(T) then
      for l in T do
        for j in l do
          if j=l[Length(l)] then
            a := Concatenation(a,"x_",String(j));
          else
            a := Concatenation(a,"x_",String(j));
            #a := Concatenation(a,"*");
          fi;
        od;
      od;
    else
      for l in T do
        z :=1;
        for j in l do
          if s=1 then
```

```
            a := Concatenation(a,"x_",String(j));
            a := Concatenation(a,"*");
          elif s=Length(T) then
            if z<Length(l) then
              a := Concatenation(a,"x_",String(j));
              a := Concatenation(a,"^",String(s));
              a := Concatenation(a,"*");
            else
              a := Concatenation(a,"x_",String(j));
              a := Concatenation(a,"^",String(s));
            fi;
          else
            a := Concatenation(a,"x_",String(j));
            a := Concatenation(a,"^",String(s));
            a := Concatenation(a,"*");
          fi;
          z := z+1;
        od;
        s := s+1;
      od;
    fi;
  fi;
  return(a);
end;

Polynomorbit := function(H,A)
  local P, T, i;
  P := "";
  for T in A do
    if P<>"" then P := Concatenation(P,"+");fi;
    P := Concatenation(P,Polynompart(T));
  od;
  return(P);
end;

Polynomessentiel := function(H,Ur)
  local Pou, A, T, i, Sir;
  i := 1;
  Pou := "";
  if Length(Ur.par)=1 then
    Pou := Concatenation(Pou,Polynomorbit(H,Ur.orb[1]));
  else
    for Sir in Ur.orb do
      if Pou= "" then Pou := Concatenation(Pou,String(i),"(");
```

```
        else
          Pou := Concatenation(Pou,"+",String(i));
          Pou := Concatenation(Pou,"(");
        fi;
        Pou := Concatenation(Pou,Polynomorbit(H,Sir));
        Pou := Concatenation(Pou,")");
        i := i+1;
      od;
    fi;
    return(Pou);
  end;
```

A.3 Ensembles Essentiels et Invariants primitifs réguliers

1. La fonction `stabinter(Hrelatif,A)` calcule la liste des orbites des partitions de A sous l'action du groupe `Hrelatif`.
   La fonction `MinimalPrimitiveInvariants(m,Gabsolu,Hrelatif)` nous rend une liste de chaînes représentant chacune un polynômes invariants primitifs de degré minimal. La fonction `Listessentiel` nous donne la liste des ensembles essentiels de tous les poids.

```
stabinter := function(Hrelatif,A)
  local res, T;
  res := [];
  for T in A do
    res := Concatenation(res,[Orbitepartition(Hrelatif,T)]);
  od;
  return res;
end;

MinimalPrimitiveInvariants := function(m,Gabsolu,Hrelatif)
local A, U, n, i, j, test, S, Pou, P, L, Ur, sortie;
Pou := [];
A := Essentielset(m,Hrelatif,Gabsolu);
n := Length(A);
sortie := [];
Ur := rec(par := [], orb := []);
for i in [1..n] do
U := [A[i]];
L := stabinter(Hrelatif,U);
```

```
if ( Interstabor(Gabsolu,L) = Hrelatif ) then
Ur.par := U; Ur.orb := L;
Pou := Concatenation(Pou,[Polynomessentiel(Hrelatif,Ur)]);
Add(sortie,U);
fi;
od;
A := Difference(A,sortie);
for i in [1..n-1] do
for j in [i+1..n] do
U := A{[i..j]};
L := stabinter(Hrelatif,U);
test :=true;
for S in sortie do
if IntersectionSet(S,U)=S then
test:=false;
fi;
od;
if test and ( Interstabor(Gabsolu,L) = Hrelatif) then
Ur.par := U; Ur.orb := L;
Pou := Concatenation(Pou,[Polynomessentiel(Hrelatif,Ur)]);
Add(sortie,U);
fi;
od;
od;
return(Pou);
end;

   Listessentiel := function(m,Hrelatif,Gabsolu)
     local U0, G0, G1, G2, A, liste, s, L, res;
     res := [];
     liste := Listpoidstypes(m);
     s := liste[1];
     L := Partmpoids(m,Hrelatif,s);
     U0 := L;
     liste := liste{[2..Length(liste)]};
     G0 := Gabsolu;
     G1 := Gabsolu;
     while (liste <>[]) do
       while (G1 <> Hrelatif) do
         s := liste[1];
         A := Partmpoids(m,Hrelatif,s);
         G2 := Interstabor(Gabsolu,A.orb);
         G1 := Intersection(G2,G0);
         if (G1 <> G0) then
```

```
        if (G1 = G2) then
          U0 := A.par;
        else
          U0 := Concatenation(U0,A.par);
        fi;
        G0 := G1;
      fi;
      liste := liste{[2..Length(liste)]};
    od;
    Add(res,U0);
    U0 := L;
    G0 := Gabsolu;
    G1 := Gabsolu;
  od;
  return(res);
end;
```

2. La fonction `TransitiveMinimalPrimitiveInvariants` calcule la liste des invariants primitifs absolus de tous les groupes transitifs de S_n.
   La fonction `AllPrimitiveInvariants(m,Gabsolu,Hrelatif)` nous rends une liste de chaînes représentant des polynômes invariants primitifs de tous les degrés possibles.

```
AllPrimitiveInvariants := function(m,Gabsolu,Hrelatif)
  local A, sortie, U, n, i, j, test, S, P, B, Pou, Ur, L;
  Pou := [];
  B := Listessentiel(m,Hrelatif,Gabsolu);
  for A in B do
    n := Length(A);
    sortie := [];
    Ur := rec(par :=[], orb:=[]);
    for i in [1..n] do
      U := [A[i]];
      L := stabinter(Hrelatif,U);
      if ( Interstabor(Gabsolu,L) = Hrelatif ) then
        Ur.par := U; Ur.orb := L;
        Pou := Concatenation(Pou,[Polynomessentiel(Hrelatif,Ur)]);
        Add(sortie,U);
      fi;
    od;
    A := Difference(A,sortie);
    for i in [1..n-1] do
      for j in [i+1..n] do
```

```
          U := A{[i..j]};
          L := stabinter(Hrelatif,U);
          test :=true;
          for S in sortie do
            if IntersectionSet(S,U)=S then
              test:=false;
            fi;
          od;
          if test and ( Interstabor(Gabsolu,L) = Hrelatif ) then
           Ur.par := U; Ur.orb := L;
           Pou:=Concatenation(Pou,[Polynomessentiel(Hrelatif,Ur)]);
           Add(sortie,U);
          fi;
        od;
      od;
    od;
    return(Pou);
 end;

TransitiveMinimalPrimitiveInvariants := function(n,k)
  local G;
  G:=SymmetricGroup(n);
  return(MinimalPrimitiveInvariants(n,G,Subgroup
        (G,GeneratorsOfGroup(TransitiveGroup(n,k)))));
end;
```

Annexe B

Implantations et Résultats Expérimentaux

Nous présentons dans cette partie les résultats expérimentaux du module informatique « `PrimitiveInvariant` » sous forme de tables. Les calculs ont été effectués sur des machines (Compaq XP/1000 - Alpha EV6 de 500 Mhz 640 Mo).

Les deux premières colonnes des tables désignent repectivement l'indice des groupes L et $H \subset L$ dans la liste des représentants des sous-groupes de S_n. Cette liste est obtenue grâce aux coommandes suivantes du sytème de 'calcul formel GAP :

- `S := SymmetricGroup(n) ; ;`
- `a := ConjugacyClassesSubgroups(S) ; ;`
- `List(a,Representative) ; ;`

Par exemple, pour $n = 4$, le tableau suivant représente la liste des représentants des sous-groupes de S_n et leur indices :

Numéro	Groupe associé	Numéro	Groupe associé
2	$<(4,5)>$	11	$<(4,5),(1,2,3)>$
3	$<(2,3)(4,5)>$	12	$<(4,5),(2,3),(2,4)(3,5)>$
4	$<(3,4,5)>$	13	$<(1,2,3,4,5),(2,5)(3,4)>$
5	$<(2,3)(4,5),(2,4)(3,5)>$	14	$<(2,3)(4,5),(2,4)(3,5),(3,4,5)>$
6	$<(2,3)(4,5),(2,4,3,5)>$	15	$<(4,5),(1,2,3),(2,3)>$
7	$<(4,5),(2,3)>$	16	$<(1,2,3,4,5),(2,5)(3,4),(2,4,5,3)>$
8	$<(1,2,3,4,5)>$	17	$<(2,3)(4,5),(2,4)(3,5),(3,4,5),(4,5)>$
9	$<(3,4,5),(4,5)>$	18	$<(1,3,2),(2,4,3),(2,3)(4,5)>$
10	$<(3,4,5),(1,2)(4,5)>$	19	S_5

TAB. B.1 – *Représentants des sous groupes de* S_4

La troisième colonne des tables des sections B.1 et B.2 représente des invariants primitifs relatifs ou absolus de degré minimal. Leur temps de calculs donnés dans la quatrième

colonne des tables.

Si nous avons choisi le logiciel de calcul formel GAP, c'est à cause de la simplicité de ses fonctions et de la structure des objets qu'il manipule, bien adapté aux modèles de calcul utilisés. C'est aussi à cause de la richesse de sa bibliothèque sur les groupes (la bibliothèque des groupes n'existe presque pas en AXIOM et ALDOR).

B.1 Invariants primitifs relatifs et absolus pour S_4

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
5	2	$x_4 + x_3$	41
6	2	$x_2x_4 + x_1x_3$	46
6	3	x_2	15
7	2	$x_4 + x_3$	18
8	3	$x_4 + x_3$	18
8	4	$x_3x_4^2 + x_4x_2^2 + x_2x_3^2$	42
9	2	$1(x_3x_4) + 2(x_2x_4 + x_1x_3)$	90
9	3	x_2	24
9	5	$x_2x_4 + x_1x_3$	27
9	6	$x_4 + x_3$	17
9	7	$x_2x_4^2 + x_3x_2^2 + x_4x_1^2 + x_1x_3^2$	39
10	2	$x_4 + x_3$	16
10	4	$x_4 + x_3 + x_2$	14
10	5	$x_3x_4 + x_1x_2$	33
11	2	$1(x_3x_4) + 2(x_2x_4 + x_1x_3)$	109
11	3	$1(x_4 + x_3) + 2(x_2)$	60
11	4	$x_3x_4^2 + x_4x_2^2 + x_2x_3^2$	243
11	5	$1(x_3x_4 + x_1x_2) + 2(x_2x_4 + x_1x_3)$	70
11	6	$x_4 + x_3$	19
11	7	$x_2x_4^2 + x_3x_2^2 + x_4x_1^2 + x_1x_3^2$	52
11	8	$x_4 + x_3 + x_2$	15
11	9	$x_3x_4 + x_1x_2$	30
11	10	$x_2x_3^2x_4^3 + x_3x_4^2x_2^3 + x_4x_2^2x_3^3 + x_1x_4^2x_3^3 + x_3x_1^2x_4^3 + x_4x_3^2x_1^3 + x_1x_2^3x_4^3 + x_2x_4^2x_1^3 + x_4x_1^2x_2^3 + x_1x_3^3x_2^3 + x_2x_1^2x_3^3 + x_3x_2^2x_1^3$	87

TAB. B.2 – *Invariants primitifs de S_4*

B.2 Invariants primitifs relatifs et absolus pour S_9

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
9	3	$x_9 + x_8$	174280
10	2	x_7	174890
10	3	$x_7x_9 + x_6x_8$	464810
11	3	$x_9 + x_8$	174470
12	3	x_5	174840
13	4	$x_9 + x_8$	175030
14	3	x_5	175310
14	4	$x_5x_9 + x_4x_8$	463240
15	3	$x_9 + x_8$	174600
16	2	x_7	174810
16	5	$x_7x_9 + x_6x_8$	463350
17	2	x_7	175160
17	4	$x_7x_9 + x_6x_8$	462970
18	3	$x_9 + x_8$	175080
19	3	$x_9 + x_8$	174960
20	3	$x_9 + x_8$	174770
21	4	$x_9 + x_8$	173610
22	3	x_5	174700
23	5	$x_9 + x_8$	174640
25	2	$x_9 + x_8$	174590
25	6	$x_8x_9^2 + x_9x_7^2 + x_7x_8^2$	1763590
26	6	x_6	173600
27	2	x_7	175330
28	6	x_6	174410
29	6	x_6	176530
30	8	$x_9 + x_8 + x_7$	175990
31	6	x_6	175940
32	6	x_6	175310
33	8	$x_6x_8 + x_5x_7 + x_4x_9$	462550
34	2	x_7	174790
35	8	$x_9 + x_8 + x_7$	175040
36	5	$x_9 + x_8$	174830
37	7	$x_9 + x_8 + x_7$	174610
38	7	$x_6x_8 + x_5x_7 + x_4x_9$	463100
39	5	$x_9 + x_8$	175240
40	8	$x_9 + x_8 + x_7$	174780
41	8	x_3	175130
43	4	$x_9 + x_8$	175420
43	13	$x_9 + x_8 + x_7 + x_6$	173880
44	2	x_7	175190
44	3	$1(x_8x_9) + 2(x_7x_9 + x_6x_8)$	467150
44	9	$x_7x_9 + x_6x_8$	463530
44	10	$x_9 + x_8$	174850
44	11	$x_7x_9^2 + x_8x_7^2 + x_9x_6^2 + x_6x_8^2$	1682230
45	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5) + 4(x_4) + 5(x_3)$	173480

TAB. B.3 – *Invariants primitifs de S_9*

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
46	16	$x_5x_7 + x_4x_6$	461580
47	2	x_7	175100
47	5	$x_7x_9 + x_6x_8$	463650
47	16	$x_7 + x_6$	175300
48	4	$x_9 + x_8$	174750
48	21	$x_9 + x_8 + x_7 + x_6$	175300
49	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	175320
49	4	$x_5x_9 + x_4x_8$	462130
49	9	x_5	174610
49	14	$x_9 + x_8$	174690
49	20	$x_5x_9 + x_5x_8 + x_4x_7 + x_4x_6$	461620
50	3	x_5	174710
50	4	$x_5x_9 + x_4x_8$	463310
50	13	$x_5x_9 + x_4x_8 + x_3x_7 + x_2x_6$	462060
50	14	$x_9 + x_8$	174510
51	3	x_5	175120
51	4	$x_5x_9 + x_4x_8$	460930
51	14	$x_9 + x_8$	174070
51	21	$x_5x_9 + x_4x_8 + x_3x_7 + x_2x_6$	461180
52	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174540
52	11	x_5	174650
52	12	$x_9 + x_8$	174540
52	18	$x_7x_9 + x_6x_8$	462520
53	2	$1(x_9 + x_8) + 2(x_7) + 3(x_6) + 4(x_5)$	174800
53	16	x_3	174850
54	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	173310
54	9	x_5	174560
54	12	$x_9 + x_8$	175270
54	15	$x_5x_9 + x_5x_8 + x_4x_7 + x_4x_6$	462170
55	2	x_7	175150
55	5	$x_7x_9 + x_6x_8$	464060
55	16	$x_7 + x_6$	174640
56	2	x_7	174590
56	3	$1(x_8x_9) + 2(x_7x_9 + x_6x_8)$	465620
56	10	$x_9 + x_8$	174420
56	15	$x_5x_7x_9 + x_5x_6x_8 + x_4x_7x_8 + x_4x_6x_9$	1639810
56	18	$x_7x_9 + x_6x_8$	460870
57	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174700
57	4	$x_5x_9 + x_4x_8$	463160
57	14	$x_9 + x_8$	174710
57	18	x_3	174550
58	2	$1(x_9 + x_8) + 2(x_7) + 3(x_6) + 4(x_5)$	174660
58	3	$1(x_8x_9) + 2(x_7x_9 + x_6x_8) + 3(x_7x_8 + x_6x_9) + 4(x_6x_7) + 5(x_5x_9 + x_5x_8)$	464790
58	4	$1(x_8x_9) + 2(x_7x_9 + x_6x_8) + 3(x_7x_8 + x_6x_9) + 4(x_6x_7) + 5(x_5x_9 + x_4x_8)$	463060
58	10	x_5	174630
58	14	$x_7x_9 + x_6x_8$	462350
58	17	$x_5x_7 + x_4x_6$	461680
58	22	$x_5x_7x_9 + x_5x_6x_8 + x_4x_7x_8 + x_4x_6x_9$	1629510
59	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174600
59	9	x_5	174640
59	19	$x_5x_9 + x_5x_8 + x_4x_7 + x_4x_6$	462420
59	22	$x_9 + x_8$	174660
60	2	x_7	174700
60	3	$1(x_8x_9) + 2(x_7x_9 + x_6x_8)$	465930
60	10	$x_9 + x_8$	174440
60	19	$x_5x_7x_9 + x_5x_6x_8 + x_4x_7x_8 + x_4x_6x_9$	1623190
60	20	$x_7x_9 + x_6x_8$	462010

TAB. B.4 – *Invariants primitifs de* S_9

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
61	3	x_5	174210
61	4	$x_5x_9 + x_4x_8$	462950
61	14	$x_9 + x_8$	174670
62	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174090
62	11	x_5	173080
62	20	$x_7x_9 + x_6x_8$	461590
62	22	$x_9 + x_8$	174120
63	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174350
63	4	$x_5x_9 + x_4x_8$	461970
63	11	x_5	174750
63	14	$x_9 + x_8$	174600
63	19	$x_5x_9 + x_5x_8 + x_4x_7 + x_4x_6$	462060
64	4	$x_9 + x_8$	175030
64	13	$x_9 + x_8 + x_7 + x_6$	174510
65	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174910
65	4	$x_5x_9 + x_4x_8$	463170
65	14	$x_9 + x_8$	174540
65	15	x_3	174390
66	4	$x_9 + x_8$	174340
66	13	$x_9 + x_8 + x_7 + x_6$	174300
67	2	x_7	174820
67	5	$x_7x_9 + x_6x_8$	465330
67	16	$x_7 + x_6$	174500
68	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174710
68	12	$x_9 + x_8$	174820
68	19	$x_5x_9 + x_5x_8 + x_4x_7 + x_4x_6$	460640
69	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	175100
69	12	$x_9 + x_8$	175700
69	20	$x_7x_9 + x_6x_8$	462190
70	3	x_5	175230
70	22	$x_9 + x_8$	175070
71	3	x_5	175030
71	4	$x_5x_7 + x_4x_6$	462660
71	14	$x_9 + x_8$	174850
72	2	x_7	175170
72	5	$x_7x_9 + x_6x_8$	463870
72	16	$x_7 + x_6$	171730
73	5	$x_9 + x_8$	173840
73	23	$x_5 + x_4$	175130
74	3	$1(x_9 + x_8) + 2(x_7 + x_6) + 3(x_5)$	174620
74	18	x_3	174600
74	22	$x_9 + x_8$	174350
75	4	$x_9 + x_8$	174710
75	21	$x_9 + x_8 + x_7 + x_6$	174450
76	7	$x_9 + x_8 + x_7$	174380
77	6	x_6	174430
77	8	$x_6x_9 + x_5x_8 + x_4x_7$	462570
78	8	x_3	174760
79	6	x_6	174920
79	7	$x_6x_9 + x_5x_8 + x_4x_7$	461750

TAB. B.5 – *Invariants primitifs de S_9*

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
80	7	$x_9+x_8+x_7$	174390
81	24	$x_8x_9^2+x_7x_8^2+x_6x_7^2+x_9x_5^2+x_5x_6^2$	1629170
82	3	x_5	174640
83	24	x_4	174290
84	2	x_7	175090
85	24	x_4	175070
86	3	x_9+x_8	173370
86	6	$x_9+x_8+x_7$	175370
86	9	$x_8x_9+x_6x_7$	463300
87	2	$1(x_9+x_8)+2(x_7)+3(x_6)$	174890
125	2	$1(x_9+x_8)+2(x_7)+3(x_6)+4(x_5)+5(x_4)+6(x_3)$	176250
125	3	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)+5(x_5x_9+x_5x_8)$ $+6(x_5x_7+x_5x_6)+7(x_4x_9+x_4x_8)+8(x_4x_7+x_4x_6)+9(x_4x_5)$ $+10(x_3x_9+x_3x_8)$	471540
125	4	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)+5(x_5x_9+x_4x_8)$ $+6(x_5x_8+x_4x_9)+7(x_5x_7+x_4x_6)+8(x_5x_6+x_4x_7)+9(x_4x_5)$ $+10(x_3x_9+x_2x_8)$	468050
125	5	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)+5(x_5x_9+x_4x_8)$ $+6(x_5x_8+x_4x_9)+7(x_5x_7+x_4x_6)+8(x_5x_6+x_4x_7)+9(x_4x_5)$ $+10(x_3x_9+x_3x_8)$	464990
125	10	$1(x_9+x_8)+2(x_7+x_6)+3(x_5)+4(x_4)+5(x_3)$	176030
125	12	$1(x_7x_8x_9+x_6x_8x_9)+2(x_6x_7x_9+x_6x_7x_8)+3(x_5x_8x_9+x_4x_8x_9)$ $+4(x_5x_7x_9+x_5x_6x_8+x_4x_7x_8+x_4x_6x_9)+5(x_5x_7x_8+x_5x_6x_9+x_4x_7x_9+x_4x_6x_8)$ $+6(x_5x_6x_7+x_4x_6x_7)+7(x_4x_5x_9+x_4x_5x_8)+8(x_4x_5x_7+x_4x_5x_6)$ $+9(x_3x_8x_9)$	1719150
125	14	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)$ $+5(x_5x_9+x_5x_8+x_4x_9+x_4x_8)+6(x_5x_7+x_5x_6+x_4x_7+x_4x_6)$ $+7(x_4x_5)+8(x_3x_9+x_3x_8+x_2x_9+x_2x_8)+9(x_3x_7+x_3x_6+x_2x_7+x_2x_6)$ $+10(x_3x_5+x_2x_4)$	466760
125	16	$1(x_8x_9)+2(x_7x_9+x_7x_8+x_6x_9+x_6x_8)+3(x_6x_7)$ $+4(x_5x_9+x_5x_8+x_4x_9+x_4x_8)+5(x_5x_7+x_4x_6)+6(x_5x_6+x_4x_7)$ $+7(x_4x_5)+8(x_3x_9+x_3x_8)$	468190
125	17	$1(x_8x_9)+2(x_7x_9+x_7x_8+x_6x_9+x_6x_8)+3(x_6x_7)$ $+4(x_5x_9+x_5x_8+x_4x_9+x_4x_8)+5(x_5x_7+x_4x_6)+6(x_5x_6+x_4x_7)$ $+7(x_4x_5)+8(x_3x_9+x_3x_8+x_2x_9+x_2x_8)$ $+9(x_3x_7+x_2x_6)$	462980
125	22	$1(x_7x_8x_9+x_6x_8x_9)+2(x_6x_7x_9+x_6x_7x_8)$ $+3(x_5x_8x_9+x_4x_8x_9)+4(x_5x_7x_9+x_5x_6x_8+x_4x_7x_8+x_4x_6x_9)$ $+5(x_5x_7x_8+x_5x_6x_9+x_4x_7x_9+x_4x_6x_8)+6(x_5x_6x_7+x_4x_6x_7)$ $+7(x_4x_5x_9+x_4x_5x_8)+8(x_4x_5x_7+x_4x_5x_6)+9(x_3x_8x_9+x_2x_8x_9)$ $+10(x_3x_7x_9+x_3x_6x_8+x_2x_7x_8+x_2x_6x_9)$	1705440
125	45	$x_3x_5x_7x_9+x_3x_5x_6x_8+x_3x_4x_7x_8+x_3x_4x_6x_9$ $+x_2x_5x_7x_8+x_2x_5x_6x_9+x_2x_4x_7x_9+x_2x_4x_6x_8$	4300620
125	46	x_3	176900
125	53	$x_3x_5x_7+x_3x_4x_6+x_2x_5x_6+x_2x_4x_7$	1634650
125	58	$x_3x_5+x_2x_4$	464950

TAB. B.6 – *Invariants primitifs de* S_9

L	H	H-invariant L-primitif minimal	Temps (CPU) en 10^3 s
148	3	x_5	175550
148	4	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)+5(x_5x_9+x_4x_8)$	464980
148	14	x_9+x_8	175040
148	21	$1(x_8x_9+x_6x_7)+2(x_7x_9+x_7x_8+x_6x_9+x_6x_8)+3(x_5x_9+x_4x_8+x_3x_7+x_2x_6)$	465010
148	51	$x_9+x_8+x_7+x_6$	175240
148	71	$x_9+x_8+x_5+x_4$	175880
149	3	$1(x_9+x_8)+2(x_7+x_6)+3(x_5)$	175810
149	4	$x_5x_9+x_4x_8$	464500
149	11	x_5	175200
149	13	$x_5x_9+x_4x_8+x_3x_7+x_2x_6$	465600
149	14	$1(x_9+x_8)+2(x_7+x_6)+3(x_5+x_4)$	175750
149	19	$x_5x_9+x_5x_8+x_4x_7+x_4x_6$	463630
149	50	$x_7x_9+x_6x_8$	464050
149	63	x_5+x_4	175740
150	5	x_9+x_8	175590
150	23	x_5+x_4	175390
150	73	$x_9+x_8+x_7+x_6$	175620
151	5	x_9+x_8	175660
151	23	x_5+x_4	175570
151	73	$x_9+x_8+x_7+x_6$	175700
152	2	$1(x_9+x_8)+2(x_7)+3(x_6)+4(x_5)$	175630
152	5	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)$ $+5(x_5x_9+x_4x_8)$	466810
152	16	$1(x_9+x_8)+2(x_7+x_6)+3(x_5+x_4)+4(x_3)$	175370
152	53	x_7+x_6	174980
152	55	x_3	175190
152	72	$x_5x_7+x_4x_6$	464520
153	2	$1(x_9+x_8)+2(x_7)+3(x_6)+4(x_5)$	174830
153	5	$1(x_8x_9)+2(x_7x_9+x_6x_8)+3(x_7x_8+x_6x_9)+4(x_6x_7)$ $+5(x_5x_9+x_4x_8)$	466190
153	16	$1(x_9+x_8)+2(x_7+x_6)+3(x_5+x_4)+4(x_3)$	175420
153	47	x_3	175570
153	53	x_7+x_6	175040
153	67	$x_3x_7+x_3x_6+x_2x_5+x_2x_4$	461750
155	7	$1(x_9+x_8+x_7)+2(x_6+x_5+x_4)$	173550
155	37	$x_9+x_8+x_7+x_6+x_5+x_4$	175390
155	76	$x_6x_8x_9+x_5x_7x_8+x_4x_7x_9+x_3x_5x_6+x_2x_4x_5$ $+x_2x_3x_9+x_1x_4x_6+x_1x_3x_7+x_1x_2x_8$	1618110
156	6	$1(x_9+x_8+x_7)+2(x_6)+3(x_5)$	175390
156	8	$1(x_8x_9+x_7x_9+x_7x_8)+2(x_6x_9+x_5x_8+x_4x_7)$ $+3(x_6x_8+x_5x_7+x_4x_9)$	464410
156	29	x_6+x_5	175430
156	33	$x_6x_9+x_5x_8+x_4x_7$	464760
156	77	$x_8x_9^2+x_9x_7^2+x_7x_8^2$	1630390

TAB. B.7 – *Invariants primitifs de S_9*

Annexe C

Le calcul d'invariants classiques

C.1 Les Invariants Classiques

La fonction `indice(t)` calcule le poids de la liste `t` et `degre(t)` comme son nom l'indique calcule le degré du monôme qui corresponds à la liste `t`.

$$\texttt{Calculsoustype}(8,3) = [[3,3,2,0,0,0,0,0],[4,2,2,0,0,0,0,0],[4,3,1,0,0,0,0,0],$$
$$[5,2,1,0,0,0,0,0],[6,1,1,0,0,0,0,0]].$$

Ces listes sont de poids respectivement égal à 7, 6, 5, 4 et 3 et de degré 8.
fonction `calcultype(n,d,g)` calcule toutes les listes qui corresondent à des monômes en `n` variables, de degré `d` et de poids `g`. Par exemple :

```
b :=calcultype(8,3,12) ;
```

- `[[0,0,0,0,3,0,0,0,0]`,`[0,0,0,2,0,0,1,0,0]`,`[0,0,1,0,0,2,0,0,0]`,
- `[0,0,2,0,0,0,0,0,1]`,`[1,0,0,0,0,0,2,0,0]`,`[0,0,0,1,1,1,0,0,0]`,
- `[0,0,1,0,1,0,1,0,0]`,`[0,0,1,1,0,0,0,1,0]`,`[0,1,0,0,0,1,1,0,0]`,
- `[0,1,0,0,1,0,0,1,0]`,`[0,1,0,1,0,0,0,0,1]`,`[1,0,0,0,0,1,0,1,0]`,
- `[1,0,0,0,1,0,0,0,1]]`.

La fonction `monome(t)`transforme une liste en un monôme. Par exemple :
`monome([1,2,3,4,5,6,7,8,9])=` $A_0^1A_1^2A_2^3A_3^4A_4^5A_5^6A_6^7A_7^8A_8^9$.

La fonction `polynome(n,d,g)` calcule la forme générale du polynôme en n variables de degré d et de poids g :

`polynome(8,3,12)`$=X_1A_4^3+X_2A_3^2A_6+X_3A_2A_5^2+X_4A_2^2A_8+X_5A_0A_6^2+X_6A_3A_4A_5+X_7A_2A_4A_6+X_8A_2A_3A_7+X_9A_1A_5A_6+X_10A_1A_4A_7+X_11A_1A_3A_8+X_12A_0A_5A_7+X_13A_0A_4A_8$.

Les fonctions `derive1(t)` et `derive2(t)` définissent respectivement les opérateurs différentiels Δ_1 et Δ_2 appliqué au monôme associé à liste `t`. Les fonctions `Delta1(n,d,g)` et `Delta2(n,d,g)` calculent les dérivés associés au polynôme `polynome(n,d,g)` par rapport à `Delta1(n,d,g)` et `Delta2(n,d,g)` :

`Delta1(8,3,12)=` $((X_11+4X_13)A_8A_3+(X_10+5X_12+8X_13)A_7A_4+(X_9+12X_5+7X_12)A_6A_5)A_0+((4X_4+3X_11)A_8A_2+(2X_8+8X_11+4X_10)A_7A_3+(2X_7+5X_9+7X_10)A_6A_4+(2X_3+6X_9)A_5^2)A_1+(8X_4+3X_8)A_7A_2^2+((7X_8+6X_2+4X_7)A_6A_3+(6X_7+10X_3+3X_6)A_5A_4)A_2+(6X_2+4X_6)A_5A_3^2+(12X_1+5X_6)A_4^2A_3$.

`Delta2(8,3,12)=` $(X_12+4X_13)A_8A_5+(4X_5+3X_12)A_7A_6)A_0+((X_10+5X_11+8X_13)A_8A_4+(2X_9+4X_10+8X_12)A_7A_5+(3X_9+8X_5)A_6^2)A_1+((12X_4+7X_11+X_8)A_8A_3+(2X_7+7X_10+5X_8)A_7A_4+(4X_7+6X_3+7X_9)A_6A_5)A_2+(6X_8+2X_2)A_7A_3^2+((10X_2+6X_7+3X_6)A_6A_4+(6X_3+4X_6)A_5^2)A_3+(12X_1+5X_6)A_5A_4^2$.

A l'aide de la fonction `solve` de MAPLE, nous trouvons l'invariant classique de degré $d=3$ et de poids $g=12$ en quelques secondes :

$P=15A_4^3+24A_3^2A_6+24A_2A_5^2+3A_2^2A_8+3A_0A_6^2-36A_3A_4A_5-22A_2A_4A_6-8A_2A_3A_7-8A_1A_5A_6+12A_1A_4A_7-4A_1A_3A_8-4A_0A_5A_7+A_0A_4A_8$.

Cet invariant classique correspond à un polynôme-différence en les racines formelles de $f(x,1)\in\mathcal{F}_8$ (forme binaire de degré $n=8$). En effet, la représentation symbolique de P est donnée par :

Voici d'autres exemples plus triviaux : pour $n=4$, le calcul des semi-invariants de degré $d=2$ et de poids $g=2$ nous donne le polynôme $H=A_0A_2-A_1^2$. Pour $d=2$ et $g=4$, $I=3A_2^2-4A_1A_3+A_0A_4$ et enfin pour $d=3$ et $g=6$, nous posons $J=A_2^3+A_1^2A_4+A_0A_3^2-2A_1A_2A_3-A_0A_2A_4$.

C.2 Implantation en GAP

```
Calculsoustype := function(m,r)
  local resultat, k, sou, t, res, i, l;
  resultat := [];
  if (r=1) then
    resultat := [[m]];
  elif (r=m) then
      for i in [1..m] do resultat[i] := 1; od;
      resultat := [resultat];
    else
      k := m-1;
      while (k > r-2) do
```

```
        sou := Calculsoustype(k,r-1);
        repeat
          if Length(sou)>0 then
            t := sou[1];
            if ( t[1] > m-k ) then
              if Length(sou)>1 then
                sou := sou{[2..Length(sou)]};
              else
                sou:=[];
              fi;
            else
              res := Concatenation([m-k],t);
              Append(resultat,[res]);
              if Length(sou)>1 then
                sou := sou{[2..Length(sou)]};
              else
                sou:=[];
              fi;
            fi;
          fi;
        until (sou=[]);
        k := k-1;
      od;
  fi;
  for l in resultat do
    for i in [r+1..m] do l[i]:=0; od;
  od;
  return(resultat);
end;

indice := function(t)
  local j, somme, e, s;
  j := 0;
  somme := 0;
  s := Length(t);
  if s>1 then
    for e in [1..s] do
      somme := somme + ( t[e]  j );
      j := j + 1;
    od;
  fi;
  return(somme);
end;
```

```
degre := function(t)
  local somme, e;
  somme := 0;
  for e in [1..Length(t)] do
      somme := somme + t[e];
  od;
  return(somme);
end;

calcultype := function(n,d,g)
  local sortie, sar, s, r, l, resultat, i, res;
  if d<= n then
    sortie := [];
    resultat := [];
    for r in [1..d] do
      sar := Calculsoustype(d,r);
      Append(sortie,sar);
   od;
    for l in sortie do
      for i in [d+1..n+1] do l[i]:=0; od;
    od;
  else
    resultat:=[];
    sar:=Partitions(d);
    sortie := [];
    for l in sar do
      if Length(l)=n+1 then
        Add(sortie,l);
      elif Length(l)<n+1 then
        for i in [Length(l)+1..n+1] do l[i]:=0; od;
        Add(sortie,l);
      fi;
    od;
  fi;
  for l in sortie do
    Append(resultat, Arrangements(l,n+1));
  od;
  res := [];
  for l in resultat do
    if indice(l)=g then Add(res,l); fi;
  od;
  return(res);
end;
```

```
monome := function(t)
  local res, j;
  res:=Concatenation("A_", String(0),"^",String(t[1]));
  for j in [2..Length(t)] do
    res := Concatenation(res,"A_",String(j-1),"^",String(t[j]));
  od;
  return(res);
end;

polynome := function(n,d,g)
  local a, t, res;
  a:=calcultype(n,d,g);
  res:= Concatenation("X_", String(1), "", monome(a[1]));
  for t in [2..Length(a)] do
    res := Concatenation(res,"+X_",String(t),"",monome(a[t]));
  od;
  return(res);
end;

derive1 := function(t)
  local res, i, u, e,j;
  j:=2; while t[j]=0 do j:=j+1; od;
  if j=Length(t) then res:="0"; else
u:=t{[1..Length(t)]}; u[j-1]:=u[j-1]+1; u[j]:=u[j]-1; e:=(j-1)t[j];
    res:= Concatenation(monome(u), "", String(e));
    for i in [j+1..Length(t)] do if t[i]>0 then
u:=t{[1..Length(t)]}; u[i-1]:=u[i-1]+1; u[i]:=u[i]-1; e:=(i-1)t[i];
      res := Concatenation(res,"+",monome(u),"",String(e));
    fi; od;
  fi;
  return(res);
end;

Delta1 := function(n,d,g)
  local a, res, i;
  a:=calcultype(n,d,g);
  res:= Concatenation("X_", String(1), "(", derive1(a[1]), ")");
  for i in [2..Length(a)] do
res := Concatenation(res, "+X_", String(i), "(", derive1(a[i]), ")");
  od;
  return(res);
end;

derive2 := function(t)
```

```
    local res, i, u, e, j, n;
    n:=Length(t);
    j:=1; while t[j]=0 do j:=j+1; od;
    if j=n then res:="0"; else
      u:=t{[1..n]}; u[j+1]:=u[j+1]+1; u[j]:=u[j]-1; e:=(n-j)t[j];
      res:= Concatenation(monome(u), "", String(e));
      for i in [j+1..n-1] do if t[i]>0 then
        u:=t{[1..n]}; u[i+1]:=u[i+1]+1; u[i]:=u[i]-1;
        e:=(n-i)t[i] ;
        res := Concatenation(res,"+",monome(u),"",String(e));
      fi; od;
    fi;
    return(res);
  end;

  Delta2 := function(n,d,g)
    local a, res, i;
    a:=calcultype(n,d,g);
    res:= Concatenation("X_", String(1), "(", derive2(a[1]), ")");
    for i in [2..Length(a)] do
 res := Concatenation(res, "+X_", String(i), "(", derive2(a[i]), ")");
    od;
    return(res);
  end;
```

C.3 Représentation symbolique d'un covariant

Cette partie de l'annexe traite de la section 3.2.2 du chapitre 3. Nous présentons dans ce qui suit l'implantation en MAPLE de l'algorithme **RepresentationSymbolique**:

```
restart:
with(combinat);
u := proc(d)
local m,j,e,i,k;
m := nops(d):
j:=0: e:=[]:
for i from 1 to m do
if d[i]<0 then
for k from j to j+d[i]-1 do
e:=[op(e),i-1,m-i]:
od:
j:=j+d[i];
```

```
 fi;
 od;
 e := [op(e),seq(0,i=nops(e)+1..2*(m-1))];
 e;
 end:


> u([1,2,0,0,0]);
                        [0, 4, 1, 3, 1, 3, 0, 0]

> u([0,4,0,0,0]);

                        [1, 3, 1, 3, 1, 3, 1, 3]


 su:=proc(d)
 local e,a,res,i,es,n,l;
 e := u(d);
 n := nops(e);
 a := permute([seq(i,i=1..n/2)]);
 res := 0:
 for l in a do
 es := 1:
 for i from 1 to n/2 do
 if l[i]<>0 then es := es*mu[l[i]]^(e[2*i-1])*nu[l[i]]^(e[2*i]);fi;
 od;
 res := res+es;
 od;
 1/(nops(d)-1)!*res:
 end:

> su([0,4,0,0,0]);

          mu[1]*nu[1]^3*mu[2]*nu[2]^3*mu[3]*nu[3]^3*mu[4]*nu[4]^3


 lu:=proc(e)
 local res,i;
 res := 1;
 for i from 1 to nops(e) do
 res := res*A[i-1]^(e[i]);
 od;
 res;
 end:
```

```
> lu([1,2,3,0,1,2]);

          A[0]*A[1]^2*A[2]^3*A[4]*A[5]^2
```

Ci-joint la fonction qui permet de calculer des invariants classiques en fonction de polynômes-différences symétrisés :

```
with(combinat);

racine := proc(n,b)
ss := [seq(i, i=1..n)];
s := permute(ss):
e := subs(A[n]=a[n],b);
for j from 0 to n-1 do
g := [];
for f in s do t:=[];
for k from 1 to n-j do t:=[op(t),mu[f[k]]]; od;
t := [op(t),seq(1,o=n-j+1..n)];
g := [op(g),convert(t,`*`)];
od;
oo := convert(g,`+`);
e := subs(A[j]=a[n]/n!*oo, e);
od;
e;
end;


> racine(2,A[2]*A[0]-A[1]^2);

           -1/4*a[2]^2*(mu[2]-mu[1])^2
```

Bibliographie

La partie à gauche de chaque entrée est ce qui apparaît dans le texte lorsque nous faisons une citation :

[1] I. Abdeljaouad. Calculs d'invariants primitifs minimaux et implantation en Axiom. Mémoire de Stage du DEA Algorithmique de l'École polytechnique, France, Juin 1996.

[2] I. Abdeljaouad. Package PrimitiveInvariant sous GAP. inclus au logiciel GAP. *ftp ://ftp-gap.dcs.st-and.ac.uk/pub/gap/gap-3.4.4/deposit/gap/PrimitiveInvariant.g*, Septembre 1997.

[3] I. Abdeljaouad. Calculs d'invariants primitifs de groupes finis. *RAIRO - Informatique Théorique et Programmation*, 33(1) :59–77, Janvier-Février 1999.

[4] I. Abdeljaouad and A. Valibouze. The Hacque method and the complete GI-method for computing the Galois group. *Notes informelles de calcul formel, Palaiseau, France*, 2000-08, 1999. Présentation orale à AAECC'13, Hawaii en Novembre 1999.

[5] J.M. Arnaudiès and A. Valibouze. Lagrange resolvents. *Rapport Interne LITP, 93-61*, December 1993.

[6] J.M. Arnaudiès and A. Valibouze. Lagrange resolvents. *Journal of Pure and Applied Algebra, Special Issue MEGA*, 117-118 :23–40, 1997.

[7] P. Aubry. *Ensembles Triangulaires de polynômes et résolution de systèmes algébriques. (implantation en AXIOM).* PhD thesis, Université Pierre et Marie Curie, Paris VI, 1998.

[8] P. Aubry and A. Valibouze. Using Galois ideals for computing relative resolvents. *Présenté à MEGA'98*, 1998. A paraître à JSC en 2000.

[9] T. Becker, V. Weispfenning, and H. Kredel. *Gröbner bases : a computational approach to commutative algebra.* Number 141 in Graduate texts in mathematics. Springer-Verlag, 1993.

[10] E.H. Berwick. The condition that a quintic equation should be soluble by radicals. *Proc. London Math. Soc.*, 2(14) :301–307, 1915.

[11] E.H. Berwick. On Soluble Sextic Equations. *Proceedings of the London Mathematical Society*, 29(1674) :1–28, december 1927.

[12] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Innsbruck, 1965.

[13] W.S. Burnside. *The theory of equations*, volume 2. Dover, 1886.

[14] W.S. Burnside and A.W. Panton. *The theory of equations, with an introduction to the theory of binary algebraic forms.* Dublin, London, Hodges, Figgis, 1886.

[15] G. Cardano and T.R. Witmer. *Ars magna or The rules of algebra, 1501-1576.* New York : Dover, 1993.

[16] A. Cayley. *On a new auxiliary equation in the theory of equation of fifth order.* Philosophical transactions of the Royal Society of London, 1861. CLL.

[17] A. Cayley. *Collected mathematical papers. Vol. I-XIII, Index, Reprint.* New York and London : Johnson Reprint Corporation., 1961.

[18] C. Chevalley. *Invariants of finite groups generated by reflexions*, volume 77. American Journal of Mathematics, 1955.

[19] A. Colin. Formal computation of Galois groups with relative resolvents. *AAECC'95, Lecture Notes in Computer Science*, 948, 1995.

[20] A. Colin. Relative resolvents and partition tables in Galois group computations. *Proceeding ISSAC'97, ACM*, 1997.

[21] A. Colin. *Théorie des invariants effective, Application à la théorie de Galois et à la résolution de systèmes algébriques. Implantation en AXIOM.* PhD thesis, École polytechnique, Palaiseau, France, 1997.

[22] J.A. Dieudonné and J.B. Carell. *Invariant theory - Old and New.* Académic Press, New York, 1971.

[23] Y Eichenlaub. *Problèmes effectifs de théorie de Galois en degré 8 à 11.* PhD thesis, Université Bordeaux I, 1996.

[24] Y. Eichenlaub and M. Olivier. Computation of Galois groups for polynomials with degree up to eleven. Preprint Université Bordeaux I, 1995.

[25] J.C. Faugère. *Résolution des systèmes d'équations algébriques.* PhD thesis, Université Pierre et Marie Curie, Paris VI, 1994.

[26] J.C. Faugère. New generations of Gröbner bases algorithms. *Colloque MEGA'98, to appear in Workshop Solving Systems of Equations, MSRI, Berkeley*, 1998.

[27] H.O. Foulkes. The resolvents of an equation of seventh degree. *Quart. J. Math. Oxford Ser.*, 2, 1931.

[28] R. Fröberg. *An Introduction to Gröbner Bases.* Pure and Applied Mathematics, A Wiley-Interscience Series of Texts, Monographs, and Tracts, 1998.

[29] E. Galois. *Oeuvres Mathématiques.* publiées sous les auspices de la SMF, Gauthier Villard, 1879.

[30] K. Geissler and J. Klüners. Galois group Computation for Rational polynomials. *Journal of Symbolic Computation*, 11 :1–23, 2000.

[31] P. Gianni, B. Trager, and G. Zacharias. Gröbner bases and Primary Decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6 :149–267, 1988.

[32] K. Girstmair. On the computation of resolvents and Galois groups. *Manuscripta. Math.*, 43, 1983.

[33] K. Girstmair. On invariant polynomials and their application in field theory. *Maths of Comp.*, 48(178), 1987.

[34] M. Göbel. Computing Bases for Permutation-Invariant Polynomials. *Journal of Symbolic Comutation*, 19, 1995.

[35] P. Gordan. *Vorlesungen Uber Invariantentheorie*. Leipzig, 1885-1887, 1837-1912.

[36] M. Hacque. Théorie de Galois des anneaux presque-simples. *Journal of Algebra*, 108, 1987.

[37] M. Hacque. Caractérisation des groupes de Galois comme sous-groupes de groupes algébriques linéaires. Communication privée, 1995.

[38] I. Herman. On the Projective Invariant Representation of Conics in Computer Graphics. *Computer Graphics Forum*, 8(4) :301–314, Décembre 1989.

[39] D. Hilbert. Über die vollen invariantensysteme. *Math. Ann.*, 42 :313–370, 1893.

[40] D. Hilbert and Hermann R. *Hilbert's invariant theory papers*. Brookline, Mass. : Math Sci Press, 1978.

[41] R. Howe. The classical groups and invariants of binary forms. *Bull. Amer. M. S.*, pages 27–85, 1984.

[42] A. Hulpke. *Techniques for the computattion of Galois groups*. PhD thesis, RWTH Aachen, 1996.

[43] C. Jordan. *Traité des substitutions et des équations algébriques*. Gauthier-Villard, Paris, 1870.

[44] G. Kemper. Calculating invariant rings of finite groups over arbitrary fields. *J. Symbolic Computation*, 21, 1996.

[45] G. Kemper. The invar package for calculating rings of invariants. *IWR Prépublication* **97-08**, *Heidelberg*, 1996.

[46] G. Kemper and Steel A. Some algorithm in Invariant theory of finite groups. *Proceedings of the Euroconference on Computational Methods for Representations of Groups and Algebras*, 1999. to appear, Progress in Mathematics, Basel, Birkhäuser.

[47] J.P.S. Kung and G-C. Rota. The invariant theory of binary forms. *Bulletin of the American Mathematical Society*, 10(1), 1984.

[48] J.L. Lagrange. *Réflexions sur la résolution algébrique des équations*, volume IV. Mémoires de l'Académie de Berlin, 1771. Oeuvres de Lagrange.

[49] J.L. Lagrange. *Réflexions sur la résolution algébrique des équations, Volume III*, pages 205–421. Gauthier-Villars, Paris, 1869.

[50] A. Lascoux. Opérateurs différentiels sur l'anneau des polynômes symétriques. Manuscrit.

[51] A. Lascoux and M.-P. Schützenberger. Symmetrization operators on polynomial rings. *Funct. Anal.*, pages 77–78, 1987.

[52] F. Lehobey. Resolvent computations by resultants without extraneous powers. In *Proceedings of ISSAC'97*. ACM, 1997.

[53] F. Lehobey. *Calcul et factorisation interactive de résolvantes de Lagranges en théorie de Galois effective*. PhD thesis, Université de Rennes 1, 09 1999.

[54] E. Luther. Ueber die factoren des algebraisch lôsbaren irreducible gleichungen vom sechsten grade und ihren resolvanten. *Journal für Math.*, 37, 1848.

[55] P.A. MacMahon and G.E. Andrews. *Percy Alexander MacMahon : collected papers 1854-1929*. Cambridge, Mass. : MIT Press, c1978-c1986.

[56] W.F. Meyer. *Sur les progrès de la théorie des invariants projectifs*. Paris : Gauthier-Villars, 1897.

[57] D. Mumford and J. Fogarty. *Geometric invariant theory*. Berlin ; New York : Springer-Verlag,, 1982.

[58] M. Noro and K. Yokoyama. Factoring polynomials over algebraic extension fields. to appear, 1997.

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

[60] G. Salmon. *Modern Higher Algebra*. Chelsea Publishing Company, Bronx, New York., 1885.

[61] M. Schönert et al. *GAP – Groups, Algorithms, and Programming*. Lehrstuhl D für Mathematik, Rheinisch Westfalische Technische, Rheinisch Westfalische Technische Hochschule, Aachen, Germany, fifth edition, 1995.

[62] L. Soicher. *The computation of the Galois groups*. PhD thesis, Departement of computer science, Concordia University, Montreal, Quebec, Canada, 1981.

[63] L.H. Soicher. An algorithm for computing Galois groups. In *Computational Group Theory*, pages 291–296. Academic Press, London, 1984.

[64] L.H. Soicher and J. McKay. Computing Galois groups over the rationals. *Journal of number theory*, 20 :273–281, 1985.

[65] R.P. Stanley. Invariants of finite groups and their applications to combinatorics. *Bull. Amer. Math. Soc.*, 1(3) :451–511, 1979.

[66] R.P. Stauduhar. The determination of Galois groups. *Mathematics of Computation*, 27(124) :981–996, 1973.

[67] J.A. Stiver, P.J. Antsaklis, and D. Lemmon. An invariant-based approach to the design of hybrid control systems containing clocks. *Lecture Notes in Computer Science*, 1066, 1996.

[68] E. Study. *Einleitung in die Theorie der invariaten lenearer Transformationen auf Grund der Vektorenrechnung*. Braunschweig, F. Vieweg, 1923.

[69] B. Sturmfels. *Algorithms in Invariant Theory*. Springer Verlag, New York, 1993.

[70] J.J. Sylvester. *Collected Mathematical papers, Volume I-IV*. Cambridge Univ. Press, New York, 1904–1912.

[71] N. Tchebotarev. *Grundzüge des Galois'shen Theorie*. P. Noordhoff, 1950.

[72] N.M. Thièry. *Invariants algébriques de graphes et reconstruction ; une étude expérimentale*. PhD thesis, Université Lyon I, June 1999. N° d'ordre : 167-99.

[73] A. Valibouze. Computation of the Galois group of the Resolvent Factors for the Direct and Inverse Galois problem. *AAECC'95 Conference. LNCS 948, Paris*, pages 456–468, July 1995.

[74] A. Valibouze. Galois groups of all polynomials with applications up to degree 7. Private communication, August 1995.

[75] A. Valibouze. Théorie de Galois constructive, septembre 1995. Mémoire d'habilitation à diriger les recherches, L.I.T.P., Université Paris 6.

[76] A. Valibouze. Etude des relations algébriques entre les racines d'un polynôme d'une variable. *Bulletin of the Belgian Mathematical Society*, 6 :507–535, 1999.

[77] A. Vandermonde. *Collected Papers*. Mémoire de l'Académie des Sciences de Paris, 1771.

[78] B.L. Van Der Waerden. *A Modern Algebra*, volume 1. Ungar, New York, 1953.

[79] H. Weber. *Lehrbuch der Algebra*. Number page 223–228 dans 56 in 2. Viehweg Verlag, 1899.

[80] H.. Weyl. *The classical groups, their invariants and representations*. Princeton, N.J., London, Princeton university press ; G. Cumberlege, Oxford university press, 1946.

[81] R.L. Wilson. A method for the determination of the Galois group. *American Mathematical Society*, 1949.

[82] K. Yokoyama. A modular method for computing the Galois group of polynomials. *Journal Pure Appl. Algebra*, 117-118 :617–636, 1994.

[83] A. Young. *The collected papers of Alfred Young 1873-1940*, volume 21. Mathematical expositions. Toronto, University of Toronto Press, 1977.

Notations générales

$\mathbf{N}, \mathbf{Z}$	l'ensemble des nombres naturels, l'anneau des entiers relatifs
$\mathbf{Q}, \mathbf{C}$	le corps des rationnels, le corps des complexes
C_n^k	forme binomiale égale à $\frac{n!}{k!(n-k)!}$
S_n	le groupe symétrique de degré n
$< E >$	le groupe, l'idéal, généré par les éléments de E
$card(H) =\mid H \mid$	cardinal du groupe fini H
k, k^*, $\hat{k}$	un corps de caractéristique nulle, les éléments non nuls du corps k, une clôture algébrique de k
$k[x]$	l'anneau des polunômes en la variable x et à coefficient dans k
$x_1, \ldots, x_n$	n variables deux à deux distinctes
$k[x_1, \ldots, x_n]$	l'anneau des polynômes en les variables $x_1, \ldots, x_n$ à coefficient sur k
$R^H = k[x_1, \ldots, x_n]^H$	l'anneau des polynômes invariants par le groupe H
R_d^H	l'espace des polynômes H-invariants de degré $d \in \mathbf{N}$
$\mathcal{H}(R^H, t)$	la série de Hilbert de R^H
$\text{coeff}_i(R(t))$	le coefficient de t^i dans le polynôme $R(t)$
$\Pi_1, \ldots, \Pi_n$	des polynômes de $k[x_1, \ldots, x_n]$ qui sont invariants primaires
$\Sigma_1, \ldots, \Sigma_e$	des polynômes de $k[x_1, \ldots, x_n]$ qui sont invariants secondaires
Ω_f	un vecteur de $\hat{k}$ contenant les racines d'un polynôme $f \in k[x]$
I_{Ω_f}	idéal des Ω_f-relations
$I_{\Omega_f}^{S_n}$	idéal des relations symétriques du polynôme f ($f \in k[x]$)
$\alpha = \alpha_1, \ldots, \alpha_n$	désigne un n-uplet d'éléments de $\hat{k}$
I_α^L	idéal des α-relations L-invariantes appelé aussi idéal de Galois
$P(\Omega_f)$	l'évaluation de $P \in k[x_1, \ldots, x_n]$ en les racines de f
$\sigma.P$	l'action d'un élément $\sigma \in S_n$ sur $P \in k[x_1, \ldots, x_n]$, page 4
$\mathcal{L}_{\Theta,\Omega_f}^L$	la résolvante L-relative de Ω_f par Θ
$\mathcal{L}_{\Theta,f}$	la résolvante (absolue) de f par Θ
$Stab_L(P)$	le groupe satbilisateur du polynôme P sous l'action du groupe L

$Stab_L(U)$	le groupe satbilisateur d'une partie $U \in k[x_1, \ldots, x_n]$ sous l'action du groupe L
$Orb_H(P) = H.P$	l'orbite du polynôme $P \in k[x_1, \ldots, x_n]$ suivant le groupe H (apppelé aussi H-orbite de P, page 4)
$N_H(Q)$	la Trace réduite de Q par H, page 5
$H_L(U)$	le (L, H)-groupe de l'ensemble fini U des monômes de $k[x_1, \ldots, x_n]$ (H et L étant deux groupes finis vérifiant $H \subset L$)
$\mathcal{P}_U$	une U-fonction élémentaire (U ensemble fini de monômes de $k[x_1, \ldots, x_n]$)
$[1, n] = \{1, \ldots, n\}$	L'ensemble des entiers allant de 1 à n
$\mid I \mid$	cardinal d'un sous-ensemble I de $\{1, \ldots, n\}$
$\mathcal{T}$	l'ensemble des partitions de $\{1, \ldots, n\}$
$x_I^\beta = (\prod_{i \in I} x_i)^\beta$	le monôme formé des variables d'indice les entiers de $I \in \{1, \ldots, n\}$ et de degré $\beta \in \mathbf{N}$, page 15
Q_T	monôme égal à $\prod_{i=1}^{s} x_{T_i}^{(i-1)}$ où $T = (T_1, \ldots, T_s)$ est une partition de $\mathcal{T}$
$\mathcal{M}$	l'ensemble des monômes Q_T où $T \in \mathcal{T}$
$[r_1, \ldots, r_n]$	l'écriture des monômes choisi qui correspond par exemple à $A_1^{r_1} \ldots A_n^{r_n}$ où les A_i sont des indéterminées.
$\mathcal{R}$	lettres symboliques contenant des entiers de $\mathbf{N}$ et la lettre u
x, y	variables associés à $u \in \mathcal{R}$
$[i, j]$	la différence $\mu_i \nu_j - \mu_j \nu_i$ où $i, j \in \mathcal{R}$
$[i, u]$	la différence $\mu_i x - \nu_i y$ où $i, u \in \mathcal{R}$
$f(x, y)$	forme binaire de degré n, voir page 38
ν_i, μ_i	variables associés à $i \in \mathcal{R}$ racines de $f(x, y) = 0$
c_{ij}	un changement de variables sur x et y, voir page 38
$\pi(i)$	image de l'entier $i \in \mathbf{N}$ par la permutation π
dim_k	La dimension d'un espace vectoriel sur le corps de base k
min	Le plus petit élément d'un ensemble fini d'entiers
$deg(P)$	le degré du polynôme $P \in k[x_1, \ldots, x_n]$
$\text{coeff}_\text{i}(P(t))$	Le coefficient de t^i dans le polynôme $P(t)$
$\mathcal{B}$	espace vectoriel des polynômes différences
$\mathcal{F}_n$	Formes binaires de degré n

Index

Glossaire

SystèmeReprésentant : prends en entrée un entier n et un groupe H et renvoie un système de représentants des orbites de H de degré $\frac{n(n-1)}{2}$ (voir page 10).

EnsembleEssentiel : prends en entrée un ensemble fini $\mathcal{A}$ de monômes et nous rends un ensemble essentiel de $\mathcal{A}$ (voir page 11).

InvariantsPrimitifsDeDegréMinimum : prends en entrée deux groupes $H \subset L$ et rends une liste de polynômes H-invariants L-primitifs réduits de degré minimum et à coefficients deux à deux distincts (voir page 12).

Girstmair-Jordan : prends en entrée deux groupes $H \subset L$ et rends tous les ensembles essentiels réduits pour (L,H) de degré $\geq \frac{n(n-1)}{2}$ (voir page 13).

InvariantsPrimitifs : l'algorithme 2.1.2 calcule les invariants primitifs de degré minimum (voir page 27).

InvariantClassique : c'est l'algorithme 3.1.1 qui retrace la méthode de Hilbert pour le calcul d'invariants classiques de degré et poids donnés (voir page 44).

ReprésentationSymbolique : l'algorithme 3.2.1 prends en entrée un covariant classique J et renvoie une représentation symbolique de J (voir page 50).

ReprésentationEnRacines : cet algorithme s'applique à un invariant classique et rends un polynôme-différence symétrisé qui est la représentation de l'invariant dans $\mathcal{B}$ (voir page 53).

MéthodeDeBerwick : Cet algorithme est une automatisation de la méthode de Berwick pour le calcul des résolvantes (voir page 64).

GDI : cette fonction prends en entrée une base de Gröbner d'un idéal I, et une liste fini de groupes. Elle calcule le groupe de décomposition de l'idéal I (voir page 78).

Résumé

La théorie des invariants et la théorie de Galois sont deux thèmes centraux en Calcul Formel qui ont fait l'objet de nombreux travaux. Cette thèse est consacrée à l'étude des principales méthodes de la théorie des invariants qui interviennent en théorie de Galois effective. La première partie de la thèse présente un module « `PrimitiveInvariant` » qui détermine tous les invariants primitifs réduits (relatifs ou absolus). Ce sont des polynômes qui caractérisent les sous-groupes du groupe symétrique et sont à la base de la fondation de la théorie des groupes. La deuxième partie de la thèse revisite la théorie des invariants classiques pour les utiliser dans le calcul de résolvantes particulières (la résolvante est l'outil de base de la théorie de Galois effective). Nous prouvons qu'il existe toujours un invariant primitif (polynôme-différence) dont les coefficients de la résolvante associée soient des invariants classiques. Nous implémentons un algorithme de calcul de ces invariants classiques et nous automatisons la méthode de Berwick pour le calcul des résolvantes de Lagrange en exprimant les coefficients (polynômes-différences symétrisés) en fonction d'invariants classiques. Dans la troisième partie, nous présentons une méthode hybride de calcul du groupe de Galois d'un polynôme. Cette méthode combine un résultat de M. Hacque pour identifier le groupe de Galois avec un système d'équations et la méthode des idéaux de Galois développée récemment dans notre équipe.

Mots-clés: Calcul Formel, Théorie des invariants, Théorie de Galois, Groupe de permutations, invariant primitifs, covariant, idéaux de Galois, Algèbre linéaire.

Abstract

The Invariant theory and the Galois theory are the main problems in Computer Algebra. They have been studied by many authors. This thesis is devoted to the study of the interaction between the computation of the invariants and the Galois theory. First of all, we give the package « `PrimitiveInvariant` » written in GAP for computing all the primitive invariants (relative or absolute). Then, using covariants, we present a generalisation of the Berwick method's to compute Lagrange resolvents (a resolvent is the basic tool in the computation of the Galois group of an equation) and we compute the classical invariants which are invariants of groups. We finally propose two algebraic methods for computing the Galois group of an irreducible polynomial f which we call the effective Hacque method and the complete GI-method. We combine the Hacque method and the first steps of the complete GI-method to obtain an implementable method which is a new approach for computing the Galois group.

Keywords: Computer Algebra, Invariant theory, Galois Theory, Permutation group, primitive invariant, covariant, Galois ideal, Linear Algebra.