UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES

École Doctorale Sciences Fondamentales et Appliquées

THÈSE

Présentée pour obtenir le titre de

Docteur en SCIENCES
de l'Université de Nice Sophia Antipolis

Spécialité :

MATHÉMATIQUES

par

Guillaume CHÈZE

Des méthodes symboliques-numériques et exactes pour la factorisation absolue des polynômes en deux variables

Soutenue publiquement le 16 Décembre 2004 à 14 heures 30
devant le jury composé de :

M. Manuel BRONSTEIN	INRIA - Sophia Antipolis	Examinateur
M. André GALLIGO	Université de Nice Sophia Antipolis	Directeur
M. Marc GIUSTI	CNRS - École Polytechnique	Rapporteur
M. Grégoire LECERF	CNRS - Université de Versailles	Examinateur
M. Maurice MIGNOTTE	Université de Strasbourg	Rapporteur
M. Bruno SALVY	INRIA - Rocquencourt	Rapporteur

Laboratoire J.A. Dieudonné, Parc Valrose, 06108 NICE Cedex 2

Remerciements

Je tiens tout d'abord à remercier André Galligo pour sa disponibilité, ses encouragements, son enthousiasme et son optimisme sans faille durant ces trois années.

Je suis également extrêmement reconnaissant envers Bruno Salvy, Marc Giusti et Maurice Mignotte qui ont accepté la lourde tâche de rapporteur.

Manuel Bronstein m'a fait l'honneur de faire parti du jury, je l'en remercie.

Ce fut un plaisir de discuter et travailler avec Grégoire Lecerf. Je le remercie chaleureusement d'avoir accepter le rôle d'examinateur.

Paul Zimmermann et Guillaume Hanrot m'ont accueilli et se sont intéressés à mes recherches. Je leur en suis très reconnaissant.

Je remercie également Mohamed Elkadi, ainsi que tous les membres du projet GALAAD (Bernard, Monique, Laurent, Jean-Pierre, Olivier, Philippe,...) pour leur soutien pendant ces trois années.

Je n'oublie pas tous les occupants et visiteurs du bureau 801 pour la chaleureuse ambiance qu'ils ont su y apporter (Jean-Gabriel, Olivier, Jamil, Thi Ha, Delphine, Didier, Maëlle, Bianca, Jean-Pascal, Matthieu, ...).

Je remercie aussi toutes les personnes du secrétariat, du service informatique, de la bibliothèque et de la reprographie pour leur efficacité et leur gentillesse.

Pour finir, je tiens à remercier Véronique pour sa patience et ses encouragements.

Introduction

La factorisation absolue d'un polynôme $P(X,Y) \in \mathbb{K}[X,Y]$ est sa décomposition en irréductibles dans $\overline{\mathbb{K}}[X,Y]$. L'anneau $\overline{\mathbb{K}}[X,Y]$ étant factoriel une telle décomposition existe toujours. Une question se pose alors : Comment obtenir pratiquement cette factorisation ?

Il existe de nombreux algorithmes permettant d'obtenir la factorisation absolue d'un polynôme, nous allons ici en dresser un petit historique[1]. Tout d'abord il faut remarquer que le cas des polynômes en deux variables est bien représentatif du problème. En effet, nous pouvons ramener l'étude de la factorisation absolue d'un polynôme en n variables à celle d'un polynôme en deux variables. Ce passage est possible grâce au théorème de Bertini. De nombreuses versions de ce théorème ont été données (voir [HS81], [Kal95], [Gao03]).

Une fois que nous nous sommes ramené au cas bivarié, nous pouvons essayer de nous ramener au cas de la factorisation rationnelle. (La factorisation rationnelle est la décomposition en irréductibles de $P(X,Y) \in \mathbb{L}[X,Y]$ dans $\mathbb{L}[X,Y]$.) Il faut donc déterminer a priori une extension $\mathbb{L}$ de $\mathbb{K}$ dans laquelle va se trouver un facteur absolument irréductible de P, puis factoriser P dans $\mathbb{L}[X,Y]$. Dans [Tra85], [Kal85b], [DT89] les auteurs proposent une méthode pour obtenir $\mathbb{L}$. Grâce à cette réduction et à l'utilisation de l'algorithme LLL nous pouvons obtenir un algorithme de factorisation absolue ayant une complexité polynomiale. Une autre démarche possible, développée par D. Duval puis J.-F. Ragot dans [Duv91], et [Rag97], repose sur l'étude du corps $K = \mathbb{K}(X)[Y]/(P(X,Y))$, et plus particulièrement sur l'étude des éléments algébriques de K sur $\mathbb{K}$.

Dans les années 90 des méthodes plus géométriques sont apparues. Ces méthodes étaient spécialement adaptées au cas $\mathbb{K} = \mathbb{Q}$. La plupart de ces méthodes nous permettent d'obtenir une factorisation approchée. C'est-à-dire nous obtenons à l'aide de ces algorithmes une approximation numérique des facteurs absolus de P. Dans [BCGW93] les auteurs proposent une méthode utilisant le lien entre les facteurs absolument irréductibles de P et les composantes connexes de $\{(x,y) \in \mathbb{C}^2 \mid P(x,y) = 0\}$. A. Galligo et ses coauteurs (voir [GW97], [Gal99], [GR02], [CGKW02]) ont utilisé une propriété géométrique plus forte (monodromie), afin d'obtenir des algorithmes plus fins. D. Rupprecht a poursuivi ce travail dans

[1]Pour plus de détails nous pouvons consulter [Kal90], [Kal92], [Gao03], [CG05].

sa thèse, et, dans l'article [Rup04]. L'algorithme de celui-ci effectue des calculs numériques puis rend une solution exacte. Dans [SVW01c], [SVW02c], [SVW03b] les auteurs utilisent la même approche théorique, mais effectue en pratique le calcul de l'action du groupe de monodromie.

Des techniques à base d'opérateurs différentiels ont aussi été utilisées pour donner un algorithme de factorisation absolu (voir par exemple [CSTU02]). Dans ce cas nous sommes en présence d'algorithmes symboliques.

Une autre approche symbolique a vue le jour au début des années 2000. Cette approche repose sur le travail de W. Ruppert de 1986, [Rup86]. Dans cet article W. Ruppert étudie une version effective du théorème d'Ostrowski, et pour cela il donne un nouveau moyen de tester l'irréductibilité absolue d'un polynôme. La technique utilisée a diverses conséquences, dont une amélioration significative du théorème de E. Noether effectif. Ce travail sera amélioré en 1999 dans [Rup99], et dans le livre de A. Schinzel, [Sch00], en 2000. A partir de ce test d'irréductibilité absolue S. Gao déduit un algorithme probabiliste de factorisation absolue (voir [Gao03]). Dans [KM03] et [GKM$^+$04] les auteurs poursuivent ce travail en donnant une formule pour le rayon d'irréductibilité absolue d'un polynôme, et un algorithme de factorisation approchée.

Pour finir nous remarquons que chaque algorithme de factorisation donne un algorithme qui permet de tester l'irréductibilité d'un polynôme. Cependant des techniques spécifiques ont été développées pour ce problème particulier. Par exemple, un algorithme de type Las Vegas a été donné par J.F Ragot (voir [Rag97], [Rag02]), celui-ci utilise les propriétés d'un polynôme $P(X,Y) \in \mathbb{Z}[X,Y]$ lorsque nous le réduisons modulo p. Une autre approche menée par S. Gao et ses coauteurs consiste à étudier le polytope de Newton du polynôme (voir [Gao01], [GR03], [GL]).

Nous venons donc de voir que la factorisation absolue peut être envisagée sous deux angles différents : l'approche numérique-symbolique et l'approche symbolique. Dans cette thèse nous allons étudier ces deux aspects.

Le premier chapitre constitue un état de l'art. Nous rappelons les définitions et propriétés de base, les théorèmes de Noether, Ostrowski et Bertini. A cette occasion nous donnerons une nouvelle borne pour le théorème de Bertini effectif. Ensuite, nous présenterons la remontée de Hensel, puis l'algorithme LLL. Pour finir ce chapitre nous décrirons plusieurs algorithmes de factorisation.

La partie I, de cette thèse correspond à nos contributions dans l'approche symbolique-numérique de la factorisation absolue. Dans le chapitre 2, nous présentons une méthode permettant d'obtenir une factorisation exacte à partir d'une factorisation approchée. Cette méthode sera utilisée dans le chapitre 3, où nous donnerons un algorithme de factorisation absolue. Cet algorithme reprend les idées exposées dans [Rup00] et [vH02], et, grâce à l'utilisation de l'algorithme LLL nous permet d'obtenir la factorisation absolue de polynômes de degré 200. Nous verrons sur des exemples que cet algorithme se compare favorablement aux méthodes existantes.

Dans la deuxième partie de cette thèse, nous présentons nos contributions pour l'approche symbolique. Le chapitre 4 correspond à un travail en cours avec G. Lecerf. Dans ce chapitre nous proposons un algorithme symbolique pour la factorisation absolue. Cet algorithme est l'adaptation de la technique "remonter-recombiner" pour l'algorithme de S. Gao. L'amélioration proposée par cet algorithme repose sur deux points : le système linéaire considéré est plus petit que celui utilisé par S. Gao, et, la méthode est déterministe.

Dans le chapitre 5, nous présentons une condition suffisante d'irréductibilité absolue. De cette condition nous en déduisons un algorithme de type Las Vegas pour tester l'irréductibilité absolue d'un polynôme à coefficients entiers. Cet algorithme tire profit de l'information contenue dans le polytope de Newton (comme dans [Gao01]), ainsi que du calcul modulaire (comme dans [Rag97]). L'algorithme obtenu ressemble donc à celui de J.-F. Ragot, mais nous permet de détecter plus rapidement l'irréductibilité absolue.

L'annexe A, présente une démonstration du théorème de Harris affine. Ce théorème est "l'outil" mathématique permettant la démonstration des algorithmes du type [GW97], [CGKW02], [SVW03b], et, joue un rôle important pour l'algorithme *Fac-knap* vu au chapitre 3. L'annexe B décrit sur un exemple l'algorithme *Fac-knap*. Pour finir et afin de compléter le chapitre 4, nous donnons dans l'annexe C une version préliminaire de l'article [CL].

Notations

Dans tout ce qui suit tous les anneaux considérés seront commutatifs et unitaires.

Nous conservons les notations traditionnelles pour l'ensemble des entiers $\mathbb{N}$, et les anneaux $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Q}_p$, $\mathbb{F}_p$ et $\mathbb{F}_q$. Enfin $\mathfrak{S}_n$ désigne le groupe symétrique à n éléments.

Nous désignons par $|S|$ le cardinal d'un ensemble S.

$\delta_{i,j}$ représente le symbole de Kronecker, c'est à dire $\delta_{i,j} = 0$ lorsque $i \neq j$, et, $\delta_{i,j} = 1$ lorsque $i = j$.

Soit $x \in \mathbb{R}$, nous notons :
$\lfloor x \rfloor$ la partie entière inférieure de x,
$\lceil x \rceil$ la partie entière supérieure de x,
$\lfloor x \rceil = \lfloor x + 1/2 \rfloor$.
Soit $x \in \mathbb{C}$, alors $\Re(x)$ désigne la partie réelle de x, et $\Im(x)$ désigne la partie imaginaire de x.

Soient $P(X,Y) = \sum_{i,j} a_{i,j} X^i Y^j$, et $Q(X,Y) = \sum_{i,j} b_{i,j} X^i Y^j$ deux polynômes de $A[X,Y]$ où A est un anneau. Nous posons :
$\deg_X(P)$ est le degré de P par rapport à la variable X,
$\deg_Y(P)$ est le degré de P par rapport à la variable Y,
$\deg(P)$ est le degré total de P,
$Res_Y(P,Q)$ est le résultant de P et Q vus comme des polynômes en Y,
$Disc_Y(P)$ est le discriminant de P vu comme un polynome en Y.
Lorsque $A = \mathbb{C}$, nous notons : $\|P\|_\infty = \max_{i,j} |a_{i,j}|$, et, $\|P\|_1 = \sum_{i,j} |a_{i,j}|$.

$Vect_{\mathbb{Z}}(v_1, \ldots, v_n)$ désigne le $\mathbb{Z}$ module engendré par $\{v_1, \ldots, v_n\}$. Autrement dit : $Vect_{\mathbb{Z}}(v_1, \ldots, v_n) = \{\sum_{i=1}^n \lambda_i v_i \mid \lambda_i \in \mathbb{Z}\}$.
Lorsque $\mathbb{K}$ est un corps $Vect_{\mathbb{K}}(v_1, \ldots, v_n)$ désigne le $\mathbb{K}$ espace vectoriel engendré par $\{v_1, \ldots, v_n\}$. Autrement dit : $Vect_{\mathbb{K}}(v_1, \ldots, v_n) = \{\sum_{i=1}^n \lambda_i v_i \mid \lambda_i \in \mathbb{K}\}$.

Soit $\mathbb{L}$ un corps contenant $\mathbb{K}$. Dans cette situation, nous notons $[\mathbb{L} : \mathbb{K}]$ la dimension de $\mathbb{L}$ vu comme un $\mathbb{K}$ espace vectoriel.

Supposons $[\mathbb{L} : \mathbb{K}] < \infty$, soit $x \in \mathbb{L}$, alors f_x désigne le polynome minimal de x sur $\mathbb{K}$.
Si $x \neq 0$ et $x \in \mathbb{L}$, alors m_x désigne l'homomorphisme (de $\mathbb{K}$ espace vectoriel) de multiplication par x dans $\mathbb{L}$, $P_{char}(x)$ désigne le polynome caractéristique de m_x, et $Tr_{\mathbb{L}/\mathbb{K}}$ désigne la trace de m_x. On rapelle que $P_{char}(x) = f_x^k$, où $k = [\mathbb{L} : \mathbb{K}[x]]$.
Soient $|\mathbb{L} : \mathbb{K}] = s$ et $(x_1, \ldots, x_s)$ un élément de $\mathbb{L}^s$. Nous définissons le discriminant $disc_{\mathbb{L}/\mathbb{K}}(x_1, \ldots, x_s)$ comme étant le déterminant de la matrice carrée de taille $s \times s$ avec pour coefficient (i, j) : $Tr_{\mathbb{L}/\mathbb{K}}(x_i x_j)$.

Table des matières

Chapitre 1

État de l'art

Dans ce chapitre nous allons donner les définitions et propriétés élémentaires liées à la factorisation absolue des polynômes en plusieurs variables. Nous verrons aussi les trois "grands" théorèmes de la factorisation absolue, et comment obtenir ceux-ci à partir du théorème de Noether. Nous obtiendrons à cette occasion une nouvelle estimation de la probabilité pour le théorème de Bertini effectif. Nous présenterons ensuite deux outils de base pour la factorisation que sont : la remontée de Hensel et l'algorithme LLL. Puis nous donnerons un aperçu des algorithmes de factorisation absolue existants. Nous distinguerons deux catégories d'algorithmes : les algorithmes symboliques et les algorithmes numériques. Pour finir nous rappellerons des critères d'irréductibilité absolue classiques et d'autres plus récents.

1.1 Définitions et premières propriétés

Nous commençons par rappeler la définition d'un anneau factoriel :

Définition 1. *Un anneau A est dit factoriel s'il est intègre et s'il vérifie les deux propriétés suivantes :*
- *Tout élément non nul de A s'écrit $a = up_1 \cdots p_n$ où u est un inversible de A et $p_1, \ldots, p_n$ sont des irréductibles de A.*
- *Cette écriture est unique à permutation prés et à des inversibles prés.*

Exemple 1. *$\mathbb{Z}$ et tous les corps sont des anneaux factoriels.*

Le théorème de Hilbert sur les anneaux factoriels nous dit la chose suivante :

Théorème 1. *Si A est un anneau factoriel alors $A[X]$ est factoriel.*

Il en découle en particulier que $\mathbb{K}[X_1, \ldots, X_n]$, où $\mathbb{K}$ est un corps, est un anneau factoriel.

Considérons alors un polynôme $P(X_1,\ldots,X_n)$ de $\mathbb{Q}[X_1,\ldots,X_n]$, P peut donc s'écrire comme un produit de polynômes irréductibles de $\mathbb{Q}[X_1,\ldots,X_n]$. La factorisation de f dans $\mathbb{Q}[X_1,\ldots,X_n]$ s'appelle la factorisation rationnelle de P. Plus généralement nous avons :

Définition 2. *Soient* $\mathbb{K}$ *un corps, et* $P \in \mathbb{K}[X_1,\ldots,X_n]$.
La factorisation de P *en irréductibles de* $\mathbb{K}[X_1,\ldots,X_n]$ *s'appelle la factorisation rationnelle de* P.

La factorisation rationnelle a beaucoup été étudiée du point de vue pratique, c'est-à-dire, divers algorithmes ont été donnés pour obtenir cette décomposition et ont une complexité polynomiale. Nous ne dresserons pas ici l'historique de ce problème car une première bibliographie a déjà été réalisé par E. Kaltofen dans [Kal90] et [Kal92] et plus brièvement par S. Gao dans [Gao03]. Nous avons donc des algorithmes efficaces pour réaliser une factorisation rationnelle. L'algorithme le plus récent et le plus efficace pour la factorisation rationnelle, connu de l'auteur, est celui développé par G. Lecerf dans [Lec04b]. Nous présenterons cet algorithme dans la section 1.4.6 page 26.

La factorisation rationnelle de P n'est pas la décomposition maximale de P sous forme de produit de polynômes. En effet pour obtenir le nombre maximum de facteurs irréductibles il suffit de regarder la décomposition de P dans $\overline{\mathbb{K}}[X_1,\ldots,X_n]$. Nous avons alors la définition suivante :

Définition 3. *Soient* $\mathbb{K}$ *un corps,* $P \in \mathbb{K}[X_1,\ldots,X_n]$.
La factorisation de P *en irréductibles de* $\overline{\mathbb{K}}[X_1,\ldots,X_n]$*, où* $\overline{\mathbb{K}}$ *désigne la clôture algébrique de* $\mathbb{K}$*, s'appelle la factorisation absolue de* P.

Il est important de remarquer que la factorisation absolue n'a pas qu'un rôle algébrique (décomposition en le plus de facteurs polynomiaux possibles), elle a aussi un sens géométrique. En effet soit $P = P_1 \cdots P_s$ la factorisation absolue de $P \in \mathbb{K}[X_1,\ldots,X_n]$. Cette factorisation est la factorisation de P dans $\overline{\mathbb{K}}[X_1,\ldots,X_n]$. Donc géométriquement cela revient à décomposer la variété $V(P)$ sous forme d'union d'hypersurfaces irréductibles $V(P_1) \cup \ldots \cup V(P_s)$ dans $\overline{\mathbb{K}}^n$.

Remarques :
La factorisation absolue est la décomposition maximale au sens suivant :
Soit $\mathbb{M}$ un corps contenant $\overline{\mathbb{K}}$, et $P(X,Y)$ un polynôme de $\mathbb{K}[X,Y]$. Soit $P(X,Y) = P_1(X,Y)\cdots P_s(X,Y)$ la factorisation de $P(X,Y)$ dans $\mathbb{M}[X,Y]$.
Notons $P_i(X,Y) = a_{m,i}(X)Y^{m_i} + \cdots + a_{0,i}(X)$ les facteurs de $P(X,Y)$ dans $\mathbb{M}[X,Y]$. Alors pour tout $x \in \mathbb{K}$, nous avons $a_{k,i}(x) \in \overline{\mathbb{K}}$. En effet $P(x,Y)$ est à coefficients dans $\mathbb{K}$, ses racines sont donc des éléments de $\overline{\mathbb{K}}$, et nous en déduisons alors $a_{k,i}(x) \in \overline{\mathbb{K}}$. De ce fait $a_{k,i}(X) \in \overline{\mathbb{K}}[X]$. Pour voir cela, il suffit d'exprimer les coefficients de $a_{k,i}(X)$ après une interpolation en des points de $\mathbb{K}$. En effet nous exprimons les coefficients de $a_{k,i}(X)$ comme étant solutions d'un système

de Vandermonde à coefficients dans $\mathbb{K}$ et avec pour second membre un vecteur à coefficients dans $\overline{\mathbb{K}}$.
Ainsi nous obtenons $P_i(X,Y) \in \overline{\mathbb{K}}[X,Y] \subset \mathbb{M}[X,Y]$. Donc même si nous prenons un corps contenant $\overline{\mathbb{K}}$, nous n'obtiendrons pas de nouveaux facteurs. Cela signifie en particulier que si $P(X,Y) \in \mathbb{Q}[X,Y]$ alors factoriser P dans $\mathbb{C}[X,Y]$ revient à factoriser P dans $\overline{\mathbb{Q}}[X,Y]$.

Le polynôme $P(X,Y) = Y^2 - 2X^2 = (Y+\sqrt{2}X)(Y-\sqrt{2}X) \in \mathbb{Q}[X,Y]$ est irréductible dans $\mathbb{Q}[X,Y]$, mais n'est pas absolument irréductible. Sur cet exemple nous remarquons que les coefficients de X des facteurs absolument irréductibles sont conjugués sur $\mathbb{Q}$. Cette remarque se généralise et on obtient le lemme suivant :

Lemme 1 (Lemme fondamental). *Soient $\mathbb{K}$ un corps parfait, et $\overline{\mathbb{K}}$ sa clôture algébrique. Soit P un polynôme de $\mathbb{K}[X,Y]$ irréductible dans $\mathbb{K}[X,Y]$, qui s'écrit : $P(X,Y) = Y^n + a_{n-1}(X)Y^{n-1} + ... + a_0(X)$ avec $\deg(a_i(X)) \leq n-i$. Soit $P = P_1 \cdots P_s$ la factorisation en irréductibles de P dans $\overline{\mathbb{K}}[X,Y]$. Soit $K = \mathbb{K}[\alpha]$ l'extension de $\mathbb{K}$ engendrée par les coefficients de P_1.*

Dans ce cas nous pouvons écrire les P_i sous la forme :

$$P_i(X,Y) = Y^m + b_{m-1}(\alpha_i, X)Y^{m-1} + ... + b_0(\alpha_i, X)$$

où P_i est irréductible dans $\overline{\mathbb{K}}[X,Y]$, $b_k \in \mathbb{K}[Z,X]$, $\deg_X(b_k) \leq m-k$, et $\alpha_1,...,\alpha_s$ sont les différents conjugués de $\alpha = \alpha_1$.

Démonstration. Comme le corps $\mathbb{K}$ est parfait nous pouvons donc appliquer le théorème de l'élément primitif, nous avons alors $K = \mathbb{K}[\alpha]$. On désigne par α_1,...,α_k les différents conjugués de α (avec $\alpha_1 = \alpha$). On considère de plus σ_1,...,σ_k les $\mathbb{K}$ homomorphismes de $\mathbb{K}[\alpha]$ dans $\overline{\mathbb{K}}$ tels que $\sigma_i(\alpha) = \alpha_i$.

Dans un premier temps nous allons montrer que $k \leq s$.

Soit M l'extension de $\mathbb{K}$ engendrée par tous les coefficients des P_i. Alors M est une extension de $\mathbb{K}$ de degré fini car les coefficients des P_i sont algébriques, et on a $\overline{\mathbb{K}} \supset M \supset K \supset \mathbb{K}$.

Nous pouvons donc prolonger à M les $\mathbb{K}$ homomorphismes σ_i (voir [Esc97] p.89 la proposition 4.3). De ce fait pour chaque σ_i nous considérons un prolongement $\tilde{\sigma_i}$ de σ_i à M. On désignera par $\hat{\sigma_i}$ le morphisme d'anneaux de $M[X,Y]$ dans $\overline{\mathbb{K}}[X,Y]$ qui restreint à M est $\tilde{\sigma_i}$ et qui envoie X sur X et Y sur Y.

On a $P = P_1...P_s \Rightarrow \hat{\sigma_i}(P) = \hat{\sigma_i}(P_1)...\hat{\sigma_i}(P_s) = P$ car $\hat{\sigma_i}$ laisse $\mathbb{K}$ invariant. Les $\hat{\sigma_i}(P_j)$ sont des polynômes irréductibles de $\overline{\mathbb{K}}[X,Y]$. Les P_j et les $\hat{\sigma_i}(P_j)$ sont unitaires. L'unicité de la décomposition en facteurs irréductibles dans $\overline{\mathbb{K}}[X,Y]$ donne : $\hat{\sigma_i}(P_1) = P_j$.

De plus si $\hat{\sigma_i}(P_1) = \hat{\sigma_j}(P_1)$ avec $i \neq j$ alors on aurait en identifiant les coefficients de ces deux polynômes $\sigma_i = \sigma_j$ sur $\mathbb{K}[\alpha]$, ce qui est absurde. Donc $\hat{\sigma_i}(P_1) \neq \hat{\sigma_j}(P_1)$ lorsque $i \neq j$. De ce fait $k \leq s$, car l'application ev_{P_1} allant de $\{\hat{\sigma_1},...,\hat{\sigma_k}\}$ dans $\{P_1,...,P_s\}$ qui a $\hat{\sigma_i}$ associe $\hat{\sigma_i}(P_1)$ est injective.

A présent nous allons montrer que $k = s$.

Supposons $k < s$, alors $\prod_{i=1}^{k} \hat{\sigma_i}(P_1)$ divise P. Si ce polynôme est dans $\mathbb{K}[X, Y]$ nous aboutirons à une contradiction et nous pourrons alors conclure.

Comme $P_1 \in \mathbb{K}[\alpha][X, Y]$, nous pouvons écrire : $P_1 = \sum_{a,b} c_{a,b}(\alpha) X^a Y^b$ avec $c_{a,b}(Z) \in \mathbb{K}[Z]$.

Ainsi $\prod_{i=1}^{k} \hat{\sigma_i}(P_1) = \prod_{i=1}^{k} (\sum_{a,b} c_{a,b}(\alpha_i) X^a Y^b) = Q(X, Y)$. Étudions les coefficients de Q. Le coefficient de $X^a Y^b$ est obtenu ainsi :

$$\sum_{\substack{i_1+\ldots+i_k=a \\ j_1+\ldots+j_k=b}} c_{i_1,j_1}(\alpha_1)\ldots c_{i_k,j_k}(\alpha_k).$$

Ce polynôme est symétrique en $\alpha_1, \ldots, \alpha_k$. Nous pouvons donc exprimer ce polynôme, comme un polynôme en les polynômes symétriques élémentaires en les $\alpha_1, \ldots, \alpha_k$ qui sont les coefficients du polynôme minimal de α sur $\mathbb{K}$. De ce fait on vient de montrer que $Q(X, Y) \in \mathbb{K}[X, Y]$.

Nous avons donc Q divise P, $Q \in \mathbb{K}[X, Y]$ et $\deg_Y Q < \deg_Y P$ (car $k < s$). Cela nie le fait que P est irréductible. On obtient donc $k = s$ ainsi ev_{P_1} est bijective et $P = P_1 \ldots P_s = \prod_{i=1}^{k} \hat{\sigma_i}(P_1)$. □

Remarque :
Nous avons $P = \prod_{i=1}^{s} \hat{\sigma_i}(P_1)$ dans ce qui précède car les polynômes P_i sont unitaires. En effet a priori nous obtenons : $\prod_{i=1}^{s} \hat{\sigma_i}(P_1)$ divise P. Donc nous avons une égalité à un inversible prés, puis en identifiant les termes de tête nous en déduisons l'égalité. Si P n'était pas unitaire nous aurions $P = u \prod_{i=1}^{s} \hat{\sigma_i}(P_1)$ où $u \in \mathbb{K} - \{0\}$.

Exemples :
$P(X, Y) = Y^4 + 2Y^2 - 2X^2 + 1 \in \mathbb{Q}[X, Y]$ est irréductible dans $\mathbb{Q}[X, Y]$ et nous avons : $P(X, Y) = (Y^2 - \sqrt{2}X + 1)(Y^2 + \sqrt{2}X + 1) \in \overline{\mathbb{Q}}[X, Y]$.
Nous obtenons deux facteurs, les coefficients $\sqrt{2}$ et $-\sqrt{2}$ sont conjugués sur $\mathbb{Q}$, $\mathbb{Q}[\alpha] = \mathbb{Q}[\sqrt{2}]$ et $[\mathbb{Q}[\sqrt{2}] : \mathbb{Q}] = 2$.
Dans le cas non-unitaire nous pouvons rencontrer le type de situation suivant :
$P(X, Y) = 3Y^4 + 6Y^2 - 6X^2 + 3 = (\sqrt{3}Y^2 - \sqrt{6}X + \sqrt{3})(\sqrt{3}Y^2 + \sqrt{6}X + \sqrt{3})$.
Ici nous avons deux facteurs mais $\mathbb{Q}[\alpha] = \mathbb{Q}[\sqrt{2}, \sqrt{3}]$ et $[\mathbb{Q}[\sqrt{2}, \sqrt{3}] : \mathbb{Q}] = 4 \neq 2$.

1.2 Les théorèmes classiques et leurs versions effectives

Dans cette section nous allons présenter les théorèmes de Noether, Ostrowski et Bertini qui sont fondamentaux dans l'étude de la factorisation absolue. Le quatrième résultat, la "remontée de Hensel", est plus général, c'est un outil puissant à la base de nombreux algorithmes de factorisation.

Nous commencerons par l'étude du théorème de Noether. Ce théorème montre que nous pouvons tester l'irréductibilité absolue d'un polynôme en effectuant uniquement des opérations dans le corps de base. Ce théorème montre aussi que si l'on choisit un polynôme au "hasard" dans $\mathbb{R}[X_1, \ldots, X_n]$ (avec $n \geq 2$) alors il sera absolument irréductible. Dans la partie 1.2.2 nous déduirons du théorème de Noether le théorème d'Ostrowski qui étudie le comportement de la factorisation absolue modulo p. Puis dans la partie 1.2.3 toujours à partir du théorème de Noether nous obtiendrons une version du théorème de Bertini. Le théorème de Bertini permet de ramener l'étude de la factorisation absolue des polynômes en n variables à celle des polynômes en 2 variables. Tous ces théorèmes seront accompagnés de leurs versions effectives.

1.2.1 Le théorème de Noether

Dans cette section nous supposons $n \geq 2$.

Théorème 2 (Noether). *Soit P un polynôme de $\mathbb{K}[X_1, \ldots, X_n]$ de degré au plus d, donné par :*

$$P(X_1, \ldots, X_n) = \sum_{i_1+\cdots+i_n \leq d} c_{i_1,\ldots,i_n} X_1^{i_1} \cdots X^{i_n}.$$

Considérons des variables $C_{i_1,\ldots,i_n}$ avec $i_1 + \cdots + i_n \leq d$. Il existe des polynômes $\phi_1, \ldots, \phi_T$ en les $C_{i_1,\ldots,i_n}$ tels que :
P est réductible sur $\overline{\mathbb{K}}$ ou de degré $< d$ si et seulement si pour tout t tels que $1 \leq t \leq T$ nous avons $\phi_t\{c_{i_1,\ldots,i_n}\} = 0$.

De plus ces polynômes dépendent uniquement de d et de n et sont indépendants du corps $\mathbb{K}$, plus précisément :
Si $\mathbb{K}$ est de caractéristique nulle alors ils sont à coefficients dans $\mathbb{Z}$, et si $\mathbb{K}$ est de caractéristique $p > 0$ alors ils sont obtenus par réduction modulo p de ces mêmes coefficients.

Une preuve simple de ce théorème se trouve dans [Sch76], [Sch00]. La preuve n'utilise que des systèmes linéaires et des résultants.

Comme annoncé dans l'introduction de cette partie nous avons donc un moyen de tester si un polynôme est absolument irréductible en effectuant uniquement des opérations arithmétiques dans le corps $\mathbb{K}$ (i.e. évaluation de ϕ_t en les coefficients de P). Cette méthode n'est pas réaliste en pratique car la valeur de T peut être trés grande.

En identifiant l'ensemble des polynômes à n variables de degré au plus d à $\mathbb{K}^{\binom{n+d}{n}}$, nous voyons que les polynômes absolument réductibles ou de degré strictement inférieur à d forment un ensemble algébrique $V(\phi_1, \ldots, \phi_T) \subsetneq \mathbb{K}^{\binom{n+d}{n}}$. Cela fournit l'énoncé probabiliste suivant :

Si $\mathbb{K} = \mathbb{R}$ alors la probabilité (pour la mesure de Lebesgue) qu'un polynôme à n variables pris au hasard soit absolument irréductible est égale à 1.

Un autre énoncé probabiliste sera donné à la fin de cette section et il sera la conséquence du théorème de Noether effectif suivant :

Théorème 3 (Noether effectif [Rup86]). *Si $\mathbb{K}$ est de caractéristique 0 alors pour $1 \leq t \leq T$ nous avons :*

$$\deg(\phi_t) \leq d^2 - 1 \ \ et \ \ \|\phi_t\|_1 \leq d^{3d^2-3} \left[\binom{n+d}{n} 3^d \right]^{d^2-1}.$$

Ce résultat est remarquable en deux points. Tout d'abord la borne sur le degré de ϕ_t est polynomiale et indépendante du nombre de variables. En effet dans [Sch76] la borne était $\deg(\phi_t) \leq 2^{n-1} d^{2^{n-1}-1}$. La preuve de la borne de Schmidt utilise des systèmes résultants. L. Busé a remarqué qu'en reprenant la preuve de [Sch76] avec les techniques actuelles sur les résultants la borne devient $\deg(\phi_t) \leq \binom{nd}{n-1} \sim 2^{nd} d^{n-1}$. Ce résultat améliore la borne de Schmidt mais reste moins fin que la borne de Ruppert. Cela nous amène donc au deuxième point remarquable du théorème précédent : la démonstration. Dans sa démonstration W. Ruppert reformule le problème de la factorisation à l'aide d'une équation aux dérivées partielles. Cette reformulation se retrouve dans l'algorithme de S. Gao (voir section 1.4.5, et, [Gao03]), dans les travaux de E. Kaltofen et J. May (voir [KM03], [GKM+04]), et, dans l'algorithme de G. Lecerf (voir section 1.4.6, [Lec04b]).

En 2000 dans son livre [Sch00] Schinzel reprend le travail de Ruppert et obtient une borne sur les $\|\phi_t\|_1$ plus fine. Il remplace 3^d par 2^d.

En 2003, E. Kaltofen et J. May se basent sur un article de Ruppert de 1999 [Rup99], et donnent le résultat suivant sur les rayons d'irréductibilité comme corollaire d'un théorème de Noether effectif, (voir [KM03]) :

Corollaire 1. *Soient $P \in \mathbb{Z}[X_1, \dots, X_n]$ un polynôme absolument irréductible et $\tilde{P} \in \mathbb{C}[X_1, \dots, X_n]$ tels que $\deg(P) = \deg(\tilde{P})$.*
Si l'inégalité suivante est vérifiée :

$$\|P - \tilde{P}\|_\infty \leq (2d)^{(-12d^7+29nd^6)}(\|P\|_\infty + 1)^{(-12d^6)},$$

alors $\tilde{P}$ est irréductible dans $\mathbb{C}[X_1, \dots, X_n]$.

Pour finir nous allons donner un corollaire probabiliste au théorème de Noether effectif. Contrairement à ce que nous avons vu précédemment l'énoncé sera valable pour tout type de corps. Pour cela nous allons avoir besoin du lemme suivant :

Lemme 2 (Zippel-Schwartz, [Zip93], [Sch80]). *Soit $P \in A[X_1, \dots, X_n]$ un polynôme de degré total d, où A est un anneau intègre. Soit S un sous ensemble fini* *de A contenant $|S|$ éléments. Nous avons alors la probabilité suivante en prenant x_i au hasard de manière uniforme dans S :*

$$\mathcal{P}(P(x_1,\ldots,x_n)=0 \,|\, x_i\in S)\le \frac{d}{|S|}.$$

En appliquant ce lemme au théorème de Noether effectif nous obtenons :

Proposition 1. *Soit $P(X_1,\ldots,X_n)=\sum_{i_1+\ldots+i_n\le d} c_{i_1,\ldots,i_n}X_1^{i_1}\ldots X^{i_n}\in\mathbb{K}[X_1,\ldots,X_n]$. Si nous prenons au hasard et de manière uniforme les coefficients $c_{i_1,\ldots,i_n}$ dans un sous-ensemble fini S de $\mathbb{K}$ alors la probabilité que P soit absolument irréductible et de degré total égal à d, est minorée par :*

$$\mathcal{P}(P \textit{ est absolument irréductible et de degré } d)\ge 1-\frac{d^2-1}{|S|}.$$

Démonstration. L'évènement contraire de P est absolument irréductible et de degré d correspond à $\cap_{j=1}^{T}(\psi_j\{c_{i_1,\ldots,i_n}\}=0)$. Or $\cap_{j=1}^{T}(\psi_j\{c_{i_1,\ldots,i_n}\}=0)\subset(\psi_1\{c_{i_1,\ldots,i_n}\}=0)$. En appliquant le lemme de Zippel/Schwartz à ψ_1, nous obtenons le résultat souhaité. □

1.2.2 Le théorème d'Ostrowski

Dans cette partie nous allons déduire du théorème de Noether le théorème d'Ostrowski. Nous conservons donc l'hypothèse de la section précédente $n\ge 2$.

Théorème 4 (Ostrowski). *Soit $P\in\mathbb{Z}[X_1,\ldots,X_n]$ un polynôme absolument irréductible de degré total d. Alors pour tous les nombres premiers sauf un nombre fini nous avons :*
$P \mod p$ est de degré total d et absolument irréductible (c'est à dire $P \mod p$ est irréductible dans $\overline{\mathbb{F}_p}[X_1,\ldots,X_n]$).

Démonstration. P est absolument irréductible donc d'après le théorème de Noether il existe un polynôme ϕ_t tel que : $\phi_t\{c_{i_1,\ldots,i_n}\}=N\in\mathbb{Z}$ et $N\neq 0$. Donc pour tous les nombres premiers, exceptés les diviseurs de N, nous avons $\overline{N}\neq\overline{0} \mod p$. En appliquant une nouvelle fois le théorème de Noether nous obtenons le résultat. □

Remarque :
Ce théorème n'est plus vrai si l'on enlève l'adverbe "absolument". En effet le polynôme $P(X,Y)=X^4+Y^4\in\mathbb{Q}[X,Y]$ est irréductible dans $\mathbb{Q}[X,Y]$ mais réductible dans $\mathbb{F}_p[X,Y]$ pour tous les p premiers. Pour voir cela il suffit de se rappeler que $g(X)=X^4+1$ est irréductible dans $\mathbb{Q}[X]$ mais réductible dans $\mathbb{F}_p[X]$ pour tous les p premiers (voir par exemple [Chi95]).
Ce résultat montre en particulier qu'il y a une différence profonde entre factorisation rationnelle et absolue.

Plusieurs versions effectives de ce théorème ont été données, mais avant de les énoncer nous devons rappeler une définition :

Définition 4. *Soit $P(X,Y) \in \mathbb{Z}[X,Y]$ tel que $P(X,Y) = \sum_{i,j} c_{i,j} X^i Y^j$. La hauteur de P, notée $\|P\|_\infty$, est l'entier $\max_{i,j} |c_{i,j}|$.*

Nous pouvons à présent énoncer une des versions effectives du théorème d'Ostrowski (voir [Rup86]) :

Théorème 5 (Ostrowski effectif, [Rup86]). *Soit $P(X,Y) \in \mathbb{Z}[X,Y]$ un polynôme absolument irréductible de degré total d. Soit p un nombre premier tel que $p > d^{3d^2-3}\|P\|_\infty^{d^2-1}$ alors $P \mod p$ est absolument irréductible.*

D'autres énoncés effectifs ont été donnés depuis : W. Ruppert en 1999 dans [Rup99] a amélioré ce résultat en prenant en compte le degré en X et en Y de P. S. Gao en 2002 (voir [GR03]) en a donné un énoncé à l'aide du nombre de points entiers se trouvant dans le polytope de Newton de P.
De tels énoncés peuvent servir à construire des tests d'irréductibilité absolue. En effet nous avons le théorème suivant :

Théorème 6. *Soient P un polynôme de $\mathbb{Z}[X_1,\dots,X_n]$ et p un nombre premier tels que : $\deg(P \mod p) = \deg(P)$, et, $P \mod p$ est absolument irréductible (c'est à dire irréductible dans $\overline{\mathbb{F}_p}[X_1,\dots,X_n]$). On a alors : P est absolument irréductible.*

Démonstration. D'après le théorème de Noether il existe un polynôme ϕ_t tel que $\phi_t\{\overline{c_{i_1,\dots,i_n}}\} \neq \overline{0}$ dans $\mathbb{F}_p$. Donc $\phi_t\{c_{i_1,\dots,i_n}\} \neq 0$ dans $\mathbb{Z}$, et P est absolument irréductible d'après le théorème de Noether. □

Une démonstration de ce théorème n'utilisant pas le théorème de Noether se trouve dans [Rag97].
Ainsi à l'aide du théorème précédent et d'une version effective du théorème d'Ostrowski nous pouvons ramener l'étude de l'irréductibilité absolue d'un polynôme de $\mathbb{Z}[X,Y]$ dans $\mathbb{F}_p[X,Y]$ pour p assez grand. Des tests d'irréductibilité absolue sont basés sur cette idée (voir [Rag97] ou section 1.6).

1.2.3 Le théorème de Bertini

Le théorème de Bertini que nous allons étudier à présent nous permet de ramener l'étude des polynômes en $n \geq 2$ variables à l'étude des polynômes en deux variables. Dans ce qui suit nous supposerons donc encore $n \geq 2$.
D'un point de vue géométrique le théorème de Bertini nous dit : *l'intersection d'une variété algébrique irréductible de dimension ≥ 2 avec un hyperplan est presque toujours une sous-variété irréductible* ; (pour un énoncé précis voir [Har77] page 179). Par exemple : l'intersection d'une sphère et d'un plan donne un cercle qui est une sous-variété irréductible. Dans notre cas cela signifie que si $P(X_1,\dots,X_n) \in \mathbb{Q}[X_1,\dots,X_n]$ est absolument irréductible alors $P(a_1X+b_1Y+c_1,\dots,a_nX+b_nY+c_n)$ est un polynôme de $\mathbb{Q}[X,Y]$ absolument irréductible où $a_i, b_i, c_i \in \mathbb{Q}$.
A présent nous allons voir comment obtenir une version probabiliste du théorème de Bertini. Mais tout d'abord donnons un énoncé précis du théorème de Bertini (voir [Kal95]).

Théorème 7. *Soient $\mathbb{K}$ un corps, $P \in \mathbb{K}[X_1, \dots, X_n]$, C_1, A_2, B_2, C_2, $\dots$, A_n, B_n, C_n des variables indépendantes sur $\mathbb{K}$, $\mathbb{L} = \mathbb{K}(C_1, A_2, B_2, C_2, \dots, A_n, B_n, C_n)$ et $P_0 = P(X + C_1, A_2X + B_2Y + C_2, \dots, A_nX + B_nY + C_n) \in \mathbb{L}[X, Y]$.*
On a alors : P_0 est absolument irréductible sur $\mathbb{L}$ si et seulement si P est absolument irréductible sur $\mathbb{K}$.

A partir du théorème de Noether, du lemme de Zippel/Schwartz et du résultat ci dessus nous obtenons le résultat original suivant :

Théorème 8 (Théorème de Bertini probabiliste).
Soient $\mathbb{K}$ un corps et S un sous-ensemble fini de $\mathbb{K}$. Soient $P \in \mathbb{K}[X_1, \dots, X_n]$ de degré total d et $p_0 = P(X + c_1, \dots, a_nX + b_nY + c_n)$ où $a_i, b_i, c_i \in \mathbb{K}$.
Si $\mathbb{K}$ est de caractéristique 0, alors en prenant les a_i, b_i, c_i au hasard de manière uniforme dans S la probabilité que tous les facteurs absolument irréductibles de P deviennent des facteurs absolument irréductibles de p_0 et qu'ils conservent leur degré est supérieure à $1 - \dfrac{d^3 - d}{|S|}$.

Démonstration. Soit $P = P_1 \cdots P_s$ la factorisation de P dans $\overline{\mathbb{K}}[X_1, \dots, X_n]$. On note $\mathbb{K}_i$ l'extension de $\mathbb{K}$ engendrée par les coefficients de P_i, et d_i le degré total de P_i. Notons $p_i(X, Y) = P_i(X + C_1, A_2X + B_2Y + C_2, \dots, A_nX + B_nY + C_n) \in \mathbb{L}_i[X, Y]$ où $\mathbb{L}_i = \mathbb{K}_i(\underline{A}, \underline{B}, \underline{C})$.
D'après le théorème 7, p_i est absolument irréductible dans $\mathbb{L}_i[X, Y]$. De plus nous avons $p_i(X, Y) = \sum_{i,j} c_{i,j}(\underline{A}, \underline{B}, \underline{C})X^iY^j$ où $c_{i,j} \in \mathbb{K}_i[\underline{A}, \underline{B}, \underline{C}]$, donc d'après le théorème de Noether effectif il existe un polynôme ϕ_i tel que : $\deg(\phi_i) \leq d_i^2 - 1$, et, $\phi_i\{c_{i,j}(\underline{A}, \underline{B}, \underline{C})\} \neq 0$ dans $\mathbb{L}_i$.
On obtient alors un polynôme ψ_i tel que : $\deg(\psi_i) \leq (d_i^2 - 1)d_i$, et, $\psi_i\{\underline{A}, \underline{B}, \underline{C}\} \neq 0$ dans $\mathbb{K}_i[\underline{A}, \underline{B}, \underline{C}]$.
A présent on remarque que si $a_2, \dots, a_n, b_2, \dots, b_n, c_1, \dots, c_n \in \mathbb{K}$ sont tels que : $\psi_i\{\underline{a}, \underline{b}, \underline{c}\} \neq 0$ dans $\mathbb{K}_i$ alors d'après le théorème de Noether $\overline{p_i}(X, Y) = P_i(X + c_1, a_2X + b_2Y + c_2, \dots, a_nX + b_nY + c_n) \in \mathbb{K}_i[X, Y]$ est absolument irréductible. Ainsi pour que p_0 ait le même nombre de facteurs absolument irréductibles que P et que le degré de chacun d'eux soit conservé il faut que ; $\psi_i\{\underline{a}, \underline{b}, \underline{c}\} \neq 0$ pour $1 \leq i \leq s$. L'évènement contraire est : $\cup_{i=1}^s(\psi_i\{\underline{a}, \underline{b}, \underline{c}\} = 0)$. De ce fait la probabilité $\mathcal{P}$ recherchée vérifie :
$\mathcal{P} \geq 1 - \sum_{i=1}^s \dfrac{d_i^3 - d_i}{|S|} \geq 1 - \dfrac{d^3 - d}{|S|}$. □

Il existe de nombreux énoncés probabilistes du théorème de Bertini. En effet différents auteurs ont étudié ce problème :

- Il semblerait que les premiers auteurs à avoir étudié le théorème de Bertini pour obtenir des résultats de complexité soient J. Heintz et M. Sieveking en 1981 dans [HS81].

- E. Kaltofen a donné de nombreux énoncés de ce théorème (voir [Kal85a], [Kal85c], [Kal95], toutes ces preuves sont algébriques). La probabilité obtenue est du type : $1 - O(d^4)/|S|$.
- C. Bajaj et ses coauteurs ont donné eux aussi un énoncé probabiliste du théorème de Bertini mais à l'aide d'une preuve géométrique (voir [BCGW93]). Là aussi on obtient une probabilité du type : $1 - O(d^4)/|S|$.
- S. Gao (voir [Gao03]) a donné en 2000 un énoncé de ce théorème à la suite de son algorithme de factorisation. Celui-ci obtient alors une probabilité du type : $1 - 2d^3/|S|$.

Certaines des probabilités données ci-dessus sont valables aussi pour des corps quelconques. Ici nous avons montré comment à l'aide du résultat de W. Ruppert datant de 1986 [Rup86] nous pouvions améliorer des résultats plus récents dans le cadre $\mathbb{K}$ de caractéristique 0· Néanmoins cet énoncé peut se généraliser au cas où $\mathbb{K}$ a une caractéristique suffisamment grande car nous connaissons la “longueur” des polynômes ϕ_t (i.e. $\|\phi_t\|_1$).

Remarques : Nous pouvons considérer à la place d'une substitution du type $X_i = a_iX + b_iY + c_i$ une évaluation $X_i = t_i$, et dans ce cas nous avons les résultats suivants :

Théorème 9 (Hilbert). *Soit $P(X, T_1, \dots, T_n) \in \mathbb{Q}[X, T_1, \dots, T_n]$ un polynôme irréductible. Il existe une infinité de $t_i \in \mathbb{Q}$ tels que : $P(X, t_1, \dots, t_n) \in \mathbb{Q}[X]$ soit irréductible.*

Pour des preuves de ce théorème voir [Lan83], [Völ96], [Sch00].

Ce théorème reste vrai pour une extension finie de $\mathbb{Q}$, mais n'a aucun sens pour les corps finis. Des énoncés plus forts ont été donnés (voir [Sch00]). Certaines estimations sur le nombre de points conservant l'irréductibilité (points hilbertiens) ont été obtenus (certaines utilisent l'hypothèse de Riemann). Cependant un énoncé probabiliste explicite n'est pas connu à ce jour. Le premier ensemble explicite de points hilbertiens est du à Sprindžuk et date de 1983. Depuis d'autres résultats ont été obtenus dans ce sens (voir [Sch00]).

Proposition 2 (Sprindžuk). *Soit $P(X, Y)$ un polynôme irréductible dans $\mathbb{Q}[X, Y]$ et H l'ensemble d'entiers suivants :*

$$H = \{\lfloor e^{\sqrt{loglogm}} \rfloor + 2^{m^2} m! \mid m \in \mathbb{N} \setminus \{0\}\}.$$

On a alors : $P(h, Y)$ est réductible pour un nombre fini de points $h \in H$.

1.2.4 Le lemme de Hensel

L'idée de la méthode de Hensel est de reproduire dans un cadre algébrique la méthode de Newton. Cette dernière nous permet d'obtenir une solution exacte d'une équation à partir d'une solution approchée. Ici l'approximation sera I-adique où I est un idéal d'un anneau A et la solution se trouvera dans A_I le complété I-adique de l'anneau A (voir [Eis95], [Gou97], [Zip93] pour la définition du complété I-adique d'un anneau et les premières propriétés).

Théorème 10. *Soit I un idéal d'un anneau intègre A. Soient $\vec{F} = (F_1(X_1, \ldots, X_n), \ldots, F_n(X_1, \ldots, X_n))$ où les F_i sont des polynômes à coefficients dans A, et, $Jac_{\vec{F}}$ le Jacobien de $\vec{F}$. Soit $(x_1, \ldots, x_n)$ une racine de $\vec{F}$ modulo I, c'est-à-dire : $\vec{F}(x_1, \ldots, x_n) = 0 \mod I$ et telle que $Jac_{\vec{F}}(x_1, \ldots, x_n)$ ait un inverse dans $\dfrac{A}{I}$. Alors il existe un unique élément $(\hat{x_1}, \ldots, \hat{x_n})$ de A_I tel que :*

$$\hat{x_i} = x_i \mod I, \text{ et, } \vec{F}(\hat{x_1}, \ldots, \hat{x_n}) = 0.$$

On en déduit voir [Zip93].

Théorème 11 (Remontée de Hensel).

Soient A un anneau intègre unitaire, $f(X)$ un polynôme unitaire de $A[X]$, et I un idéal de A. S'il existe des polynômes unitaires $g_1(X), h_1(X)$ dans $\dfrac{A}{I}[X]$ premiers entre eux tels que :

$$f(X) = g_1(X)h_1(X) \mod I$$

alors il existe un unique couple de polynômes unitaires $g_n(X), h_n(X)$ dans $\dfrac{A}{I^n}[X]$ tels que :

$$g_n(X) = g_1(X) \mod I,$$

$$h_n(X) = h_1(X) \mod I,$$

$$f(X) = g_n(X)h_n(X) \mod I^n.$$

De plus il existe dans $A_I[X]$ un unique couple de polynômes $\hat{g}(X)$ et $\hat{h}(X)$ tel que :

$$\hat{g}(X) = g_1(X) \mod I,$$

$$\hat{h}(X) = h_1(X) \mod I,$$

$$f(X) = \hat{g}(X)\hat{h}(X) \mod I^n.$$

Remarque : Ces deux théorèmes sont “constructifs” : la preuve de ceux-ci nous donne un algorithme permettant d'obtenir une factorisation modulo I^n à partir d'une factorisation modulo I.

Description de la remontée de Hensel dans un cas particulier

A présent nous allons expliquer comment effectuer une remontée de Hensel dans le cas suivant : $A = \mathbb{K}[Y]$ et $I = (X)$. La situation décrite ci dessous sera celle du chapitre 3.
Nous cherchons la factorisation suivante :
$P(X,Y) = P_1(X,Y)Q(X,Y)$ où P, P_1, $Q \in \mathbb{K}[X,Y]$ sont unitaires, $\deg(P) = \deg_Y(P) = n$, $\deg(P_1) = \deg_y(P_1) = m$, $\deg(Q) = \deg_Y(Q) = n - m$, et, $P_1(0,Y)$ et $Q(0,Y)$ sont premiers entre eux.

L'objectif de ce qui suit est de montrer comment obtenir la factorisation $P = P_1 Q$ lorsque nous connaissons $P(X,Y)$, $P_1(0,Y)$ et $Q(0,Y)$.
Nous allons tout d'abord poser quelques notations afin d'expliciter le procédé de remontée de Hensel. On pose :

- $A[X]_d$ est l'anneau des polynômes à coefficients dans A de degré $\leq d$,
- $P_1^{(k)}$ est la troncature en X^{k+1} de P_1, c'est à dire $P_1^{(k)} = P_1 \mod X^{k+1}$ et $P_1^{(k)} \in (\mathbb{K}[Y])[X]_k$,
- $P_1^{(0)}(X,Y) = P_1(0,Y)$ et $Q^{(0)}(X,Y) = Q(0,Y)$,
- $P_1^{(k+1)}(X,Y) = P_1^{(k)}(X,Y) + \Delta_1^{(k+1)}(Y)X^{k+1}$, $\deg \Delta_1^{(k+1)} \leq m - (k+1)$,
- $Q^{(k+1)}(X,Y) = Q^{(k)}(X,Y) + \Delta_2^{(k+1)}(Y)X^{k+1}$, $\deg \Delta_2^{(k+1)} \leq n - (m+k+1)$,
- $P_1^{(k)}Q^{(k)} = P(X,Y) + X^{k+1}R_{k+1}(X,Y)$.

Pour expliciter la remontée de Hensel nous devons montrer comment nous obtenons successivement les polynômes $\Delta_i^{(k)}(Y)$.

Nous avons $P_1^{(k+1)}Q^{(k+1)} = P + X^{k+2}R_{k+2}$, et
$P_1^{(k+1)}Q^{(k+1)} = P_1^{(k)}Q^{(k)} + X^{k+1}(P_1^{(k)}\Delta_2^{(k+1)} + Q^{(k)}\Delta_1^{(k+1)}) \mod X^{2k+2}$.
D'où :
$P + X^{k+2}R_{k+2} = P_1^{(k)}Q^{(k)} + X^{k+1}(P_1^{(k)}\Delta_2^{(k+1)} + Q^{(k)}\Delta_1^{(k+1)}) \mod X^{2k+2}$
et $P - P_1^{(k)}Q^{(k)} = X^{k+1}(P_1^{(k)}\Delta_2^{(k+1)} + Q^{(k)}\Delta_1^{(k+1)}) \mod X^{k+2}$.
Cela donne :
$R_{k+1} = (P_1^{(k)}\Delta_2^{(k+1)} + Q^{(k)}\Delta_1^{(k+1)}) \mod X$.
Donc :
$R_{k+1}(0,Y) = P_1^{(k)}(0,Y)\Delta_2^{(k+1)}(Y) + Q^{(k)}(0,Y)\Delta_1^{(k+1)}(Y)$.
Or $P_1^{(k)}(0,Y) = P_1^{(0)}(X,Y)$ et $Q^{(k)}(0,Y) = Q^{(0)}(X,Y)$.
Il vient alors :

$$R_{k+1}(0,Y) = P_1^{(0)}(0,Y)\Delta_2^{(k+1)}(Y) + Q^{(0)}(0,Y)\Delta_1^{(k+1)}(Y).$$

Comme $P_1^{(0)}(0,Y)$ et $Q^{(0)}(0,Y)$ sont premiers entre eux nous pouvons résoudre cette relation de Bézout. Nous pouvons donc obtenir les $\Delta_i^{(k+1)}$ connaissant $R_{k+1}(0,Y)$. Il nous reste donc à expliciter comment calculer $R_{k+1}(0,Y)$.

Comme : $P(X,Y) - P_1^{(k)}(X,Y)Q^{(k)}(X,Y) = X^{k+1}R_{k+1}(X,Y)$ nous pouvons faire le produit "naïf" de $P_1^{(k)}(X,Y)$ par $Q^{(k)}(X,Y)$ cela donnerait $(k+1)^2$ produits dans $\mathbb{K}[Y]$. Ici nous allons suivre la démarche explicitée dans [Ber98].

Nous cherchons $R_{k+1}(0,Y)$, c'est la différence du terme de degré $k+1$ en X de $P(X,Y)$ avec le terme de degré $k+1$ en X de $P_1^{(k)}(X,Y)Q^{(k)}(X,Y)$. Nous notons alors $P_{[X^k]}$ le terme de $\mathbb{K}[Y]$ correspondant au terme en X^k de P. Nous devons donc calculer :

$$P_{[X^k]} - (P_1^{(k)}(X,Y)Q^{(k)}(X,Y))_{[X^k]}.$$

Or $P_1^{(k)}(X,Y) = \sum_{i=0}^k \Delta_1^{(i)}(Y)X^i$ et $Q^{(k)}(X,Y) = \sum_{i=0}^k \Delta_2^{(i)}(Y)X^i$.
D'où : $(P_1^{(k)}(X,Y)Q^{(k)}(X,Y))_{[X^k]} = \sum_{i=0}^k \Delta_1^{(i)}(Y)\Delta_2^{(k-i)}(Y)$.
Donc

$$R_{k+1}(0,Y) = P_{[X^k]} - \sum_{i=0}^{k} \Delta_1^{(i)}(Y)\Delta_2^{(k-i)}(Y).$$

Ici seulement $k+1$ multiplications dans $\mathbb{K}[Y]$ sont utilisées. De plus nous remarquons que lorsque $k \geq m+1$, il n'est plus nécessaire de calculer $\Delta_1^{(k)}(Y)$.

D'autres algorithmes permettant d'effectuer une remontée de Hensel existent (voir [vzGG03] [BLS+04]). Nous avons présenté cette méthode car c'est celle qui a été choisie pour programmer l'algorithme du chapitre 3 pour des raisons pratiques liées au logiciel MagmaV2.8-1.

1.3 L'algorithme LLL

Dans cette section nous allons présenter l'algorithme LLL du à A.K. Lenstra, H.W. Lenstra et L. Lovász [LLL82]. Nous donnerons tout d'abord quelques définitions et propriétés, puis nous verrons comment utiliser cet algorithme pour factoriser des polynômes dans $\mathbb{Z}[X]$. De nombreuses autres applications existent. En effet depuis son apparition en 1982 cet algorithme s'est révélé très efficace dans différentes situations : recherche de relations entre des nombres donnés, "simplification" d'une extension de $\mathbb{Q}$, cryptographie ... (voir [vzGG03], [Coh93]).
Dans le chapitre 3 nous utiliserons cet algorithme en adaptant les idées de M. van Hoeij afin d'obtenir un algorithme de factorisation absolue pour les polynômes de $\mathbb{Q}[X,Y]$.

1.3.1 Définitions et Propriétés

Dans de nombreux textes [LLL82], [Coh93], [Zip93], [vzGG03], [MŞ99] nous pouvons trouver une description complète de l'algorithme LLL. Nous allons donc ici nous contenter de donner les grandes lignes conduisant à cet algorithme.
Tout d'abord cet algorithme a pour but de trouver un petit vecteur dans un réseau. Ici, petit signifie "petite norme" pour la norme provenant du produit scalaire $<;>$ usuel.

Définition 5. *Soit* $\{e_1, \ldots, e_k\}$ *une famille libre de vecteurs de* $\mathbb{R}^n$*. Le* $\mathbb{Z}$ *module libre engendré par ces vecteurs est le réseau* $\mathcal{L}$ *de base* $\{e_1, \ldots, e_k\}$*. C'est à dire :* $\mathcal{L} = \{\sum_{i=1}^{k} \lambda_i e_i \mid \lambda_i \in \mathbb{Z}\}$.

Nous remarquons que dans le cas où nous avons une base orthogonale pour le réseau $\mathcal{L}$ alors un des vecteurs de base est le plus petit vecteur de $\mathcal{L}$. (Il suffit de penser à 2 vecteurs orthogonaux et au théorème de Pythagore). Nous allons donc avoir besoin de bases orthogonales.

Définition 6. *On note* $\{e_1^*, \ldots, e_k^*\}$ *la base orthogonale de* $Vect_{\mathbb{R}}(e_1, \ldots, e_k)$ *obtenue par le procédé de Gram-Schmidt appliqué à* $\{e_1, \ldots, e_k\}$*. De plus on pose :* $e_i^* = e_i - \sum_{j=1}^{i-1} \mu_{i,j} e_j^*$*, avec* $\mu_{i,j} \in \mathbb{R}$.

A présent nous pouvons énoncer plus précisément la remarque faite plus haut.

Proposition 3. *Soient* $\{e_1, \ldots, e_k\}$ *une base de* $\mathcal{L}$ *et* $\{e_1^*, \ldots, e_k^*\}$ *son orthogonalisée de Gram-Schmidt. Pour tout vecteur* v *non nul de* $\mathcal{L}$ *nous avons :* $\|v\| \geq \min_{i=1,\ldots,k}(\|e_i^*\|)$.

Ainsi, pour obtenir un petit vecteur dans le réseau $\mathcal{L}$ nous allons fabriquer une base pour $\mathcal{L}$ "la plus orthogonale" possible. Pour cela les coordonnées $\mu_{i,j}$ des vecteurs e_j^* donnent une indication : plus les $\mu_{i,j}$ seront petits, plus la base $\{e_1, \ldots, e_k\}$ sera "proche" d'une base orthogonale.

Définition 7. *Une base* $\{e_1, \ldots, e_k\}$ *de* $\mathcal{L}$ *est dite faiblement réduite si* $|\mu_{i,j}| \leq 1/2$.

Il existe un algorithme simple (à base d'orthogonalisation de Gram-Schmidt, et de calculs de l'entier le plus proche d'un nombre réel) permettant d'obtenir une base faiblement réduite.
Comme son nom l'indique une telle base ne convient pas complètement, pour obtenir de bons résultats nous devons introduire la notion suivante :

Définition 8. *Soit* $\{e_1, \ldots, e_k\}$ *une base d'un réseau* $\mathcal{L}$*. On appelle déterminant du réseau* $\mathcal{L}$ *la valeur suivante :*

$$det(\mathcal{L}) = \sqrt{det(< e_i; e_j >)} = \prod_{i=1}^{k} \|e_i^*\|.$$

On appelle défaut d'orthogonalité de la base $\{e_1, \ldots, e_k\}$ *le rapport :*

$$\rho(e_1, \ldots, e_k) = \frac{\prod_{i=1}^{k} \|e_i\|}{\prod_{i=1}^{k} \|e_i^*\|}.$$

D'après le théorème d'Hadamard nous avons $\rho(e_1, \ldots, e_k) \geq 1$ et $\rho(e_1, \ldots, e_k) = 1$ lorsque la base $\{e_1, \ldots, e_k\}$ est orthogonale.
A présent nous cherchons une condition sur la base $\{e_1, \ldots, e_k\}$ nous permettant

d'obtenir un petit défaut d'orthogonalité. Pour cela, nous cherchons une condition permettant d'avoir $\prod_{i=1}^{k} \|e_i^*\|$ le plus grand possible. Nous allons donc poser une condition pour que la suite $(\|e_i^*\|)_i$ ne décroisse pas trop vite. Considérons par exemple les vecteurs de $\mathbb{R}^3$: $e_1 = (1, 2, 1000)$, $e_2 = (3, 1, 1000)$, et $e_3 = (5, 7, 1000)$. Nous voyons que $\|e_2^*\| \ll \|e_1^*\|$. Tandis que pour les vecteurs $f_1 = (2, -1, 0)$, $f_2 = (2, 6, 0)$ et $f_3 = (1, 2, 1000)$ qui engendrent le même réseau, nous n'avons pas de décroissance brutale des $\|f_i^*\|$. Le défaut d'orthogonalité est donc meilleur dans le deuxième cas.
Pour obliger les $\|e_i^*\|$ à ne pas décroître trop vite nous allons imposer la condition : $\|e_i^*\|^2 \leq 2\|e_{i+1}^*\|^2$.

Définition 9. *Soit $\mathcal{B} = \{e_1, \ldots, e_k\}$ une base du réseau $\mathcal{L}$. On dit que $\mathcal{B}$ est LLL réduite lorsque :*

1. *$\mathcal{B}$ est faiblement réduite, et*
2. *$\|e_h^*\|^2 \leq 2^i \|e_{h+i}^*\|^2$, pour $h \leq i \leq k - h$.*

Remarque :
On peut considérer une notion plus générale de base LLL ν-réduite dépendant d'un paramètre ν. Dans ce cas, la dernière condition peut être remplacée par :
$\|e_i^* + \mu_{i,i-1} e_{i-1}^*\|^2 > \nu \|e_{i-1}^*\|^2$ avec $1/4 < \nu < 1$.
La définition ci-dessus correspond au cas $\nu = 3/4$.

A partir de cette définition nous obtenons les propriétés suivantes :

Proposition 4. *Soit $\{e_1, \ldots, e_n\}$ une base LLL réduite d'un réseau $\mathcal{L} \subset \mathbb{R}^n$. On a :*

1. *$\|e_j\|^2 \leq 2^{i-1} \|e_i^*\|^2$, pour $1 \leq j \leq i \leq n$.*
2. *$det(\mathcal{L}) \leq \prod_{i=1}^{n} \|e_i\| \leq 2^{n(n-1)/4} det(\mathcal{L})$.*
3. *$\|e_1\| \leq 2^{(n-1)/4} det(\mathcal{L})^{1/n}$.*
4. *$\|e_1\|^2 \leq 2^{n-1} \min_{v \in \mathcal{L} \setminus \{0\}} (\|v\|^2)$.*

Le dernier point de cette proposition nous montre que nous avons obtenu un vecteur $e_1 \in \mathcal{L}$ relativement petit.

L'algorithme LLL se présente ainsi :

Algorithme LLL
Entrée : Une base $\{e_1, \ldots, e_k\}$ d'un réseau $\mathcal{L}$.
Sortie : Une base LLL réduite du réseau $\mathcal{L}$.

1. Calculer une base $\{b_1, \ldots, b_k\}$ faiblement réduite.
2. Tant qu'il existe un indice j vérifiant $2\|b_j^*\| < \|b_{j-1}^*\|$ faire :
   (a) Échanger b_j et b_{j-1}.
   (b) Calculer une base faiblement réduite de ce nouveau réseau.

3. Retourner la base obtenue.

L'inconvénient des bases LLL réduites est que l'on n'obtient pas "le" plus petit vecteur du réseau mais l'avantage est que l'on peut les calculer rapidement.

Proposition 5. *Soient* $\{e_1, \ldots, e_n\}$ *une base d'un réseau* $\mathcal{L} \subset \mathbb{Z}^n$*, et* A *tel que* $\|e_i\| \leq A$ *pour* $1 \leq i \leq n$*. L'algorithme LLL appliqué au réseau* $\mathcal{L}$ *avec pour base* $\{e_1, \ldots, e_n\}$ *nécessite* $O(n^4 \log A)$ *opérations arithmétiques sur des entiers de taille* $O(n \log A)$*.*

1.3.2 Applications à la factorisation dans $\mathbb{Z}[X]$

Pour obtenir une factorisation de $P(X)$ dans $\mathbb{Z}[X]$ l'algorithme de Zassenhauss fonctionne ainsi :
Tout d'abord nous calculons la factorisation dans $\mathbb{Q}_p[X]$ de $P(X) = \prod_{i=1}^k P_i(X)$. Cela peut se faire à l'aide de l'algorithme de Berlekamp suivi d'une remontée de Hensel. Ensuite nous testons pour tous les sous ensembles $I \subset \{1, \ldots, k\}$, si $\prod_{i \in I} P_i(X)$ appartient à $\mathbb{Z}[X]$. Cette dernière étape demande un nombre exponentiel de tests. La méthode factorisation proposée par A.K. Lenstra, H.W. Lenstra et L. Lovász améliore ce point.

Factorisation dans $\mathbb{Z}[X]$ en utilisant la méthode LLL

Dans ce qui suit on identifie un polynôme $P(X) \in \mathbb{Z}[X]$ de degré n avec ses coefficients dans $\mathbb{Z}^{n+1}$, et on note $\|P\|$ la norme provenant de la norme euclidienne sur les coefficients. La méthode utilisant LLL repose alors sur la propriété suivante :

Lemme 3. *Soient* $P(X), Q(X) \in \mathbb{Z}[X]$ *de degrés respectifs* n *et* k*, et* $M \in \mathbb{Z} \setminus \{0\}$*. Soit* $U(X) \in \mathbb{Z}[X]$ *unitaire,* $\deg(U) \geq 1$ *tel que* U *divise* P *modulo* M*, et,* U *divise* Q *modulo* M*, avec* $\|P\|^k \|Q\|^n < M$*. On a alors : pgcd*$(P, Q) \in \mathbb{Z}[X] \setminus \mathbb{Z}$*.*

L'idée de l'algorithme est alors de trouver un polynôme $Q(X) \in \mathbb{Z}[X]$ tel que :
- $\|Q\|^n < M \|P\|^{-\deg(Q)}$
- Q est divisible par P_1 modulo M, où M est une constante du type p^m avec p premier, et P_1 est un facteur de P dans $\mathbb{Q}_p[X]$.

Le lemme ci-dessus nous donne alors un moyen de trouver un facteur non trivial de P dans $\mathbb{Z}[X]$. Pour obtenir le "petit vecteur " Q nous utilisons un réseau adapté à cette situation.

Factorisation dans $\mathbb{Z}[X]$ en utilisant l'approche de van Hoeij

L'idée de M. van Hoeij pour la factorisation dans $\mathbb{Z}[X]$ est de rechercher des exposants plutôt que des coefficients. En effet, nous considérons toujours la factorisation de P dans $\mathbb{Q}_p[X]$, $P = \prod_{i=1}^k P_i$, et nous remarquons qu'un facteur $Q(X) \in \mathbb{Z}[X]$

peut s'écrire $Q(X) = \prod_{i=1}^{k} P_i^{e_i}$ avec $e_i \in \{0,1\}$. Cela ramène donc le problème de la factorisation à trouver des vecteurs de $\{0,1\}^k$. Pour identifier ces 0-1 vecteurs nous allons utiliser la notion de matrice sous forme échelonnée réduite.

Définition 10. *Une matrice est dite sous forme échelonnée réduite si elle possède les trois propriétés suivantes :*

1. *Si une ligne est non nulle alors son premier coefficient non nul (en partant de la gauche) est un 1.*
2. *Le premier coefficient non nul "1" d'une ligne se trouve dans une colonne où tous les autres coefficients sont* 0*.*
3. *Si ni la ligne i ni la ligne $i+1$ ne sont nulles alors le premier* 1 *de la ligne $i+1$ se trouve plus à droite que celui de la ligne i.*

Nous remarquons qu'il existe des algorithmes pour obtenir la forme échelonnée réduite d'une matrice. De plus la forme échelonnée d'une matrice A ne change pas si l'on multiplie à gauche A par une matrice inversible. Nous utiliserons à nouveau cette notion et cette propriété dans le chapitre 3.

Nous pouvons à présent donner les points clefs de l'algorithme de van Hoeij :

- $V = \{(v_1, \ldots, v_k) \in \mathbb{Z}^k \mid \prod_{i=1}^{k} P_i^{v_i} \in \mathbb{Z}[X]\}$ est un sous réseau de $\mathbb{Z}^k$.(Comme précédemment les polynômes $(P_i)_{i=1,\ldots,k}$ sont les facteurs de P dans $\mathbb{Q}_p[X]$ obtenus à l'aide de l'algorithme de Berlekamp et d'une remontée de Hensel).
- V est un réseau engendré par des 0-1 vecteurs (c'est à dire $v \in \{0,1\}^k$) formant une base échelonnée réduite. C'est à dire la matrice où l'on écrit en ligne les 0-1 vecteurs de la base V est échelonnée réduite. Les 0-1 vecteurs de la base donnent les facteurs irréductibles de P.

Donc une fois que nous avons une base échelonnée réduite de V nous avons la factorisation de P. Pour obtenir V l'idée est de construire une suite décroissante de réseaux $(\mathcal{L}_i)_i$ telle que :

$$\begin{cases} \mathcal{L}_0 = \mathbb{Z}^k, \\ \text{Il existe un indice } i_0 \text{ tel que } \mathcal{L}_{i_0} = V. \end{cases}$$

M. van Hoeij utilise alors l'algorithme LLL pour obtenir le réseau $\mathcal{L}_{i+1}$ à partir du réseau $\mathcal{L}_i$ et il calcule une base échelonnée réduite afin d'obtenir les exposants recherchés. Avec cette démarche les réseaux se trouvent dans $\mathbb{Z}^k$ et non plus dans $\mathbb{Z}^{n+1}$ (on rappelle que $k \leq n$), et les vecteurs que l'on recherche ont des coefficients dans $\{0,1\}$. Ces deux points permettent d'obtenir des performances bien meilleures qu'avec l'algorithme de factorisation de A.K. Lenstra, H.W. Lenstra et L. Lovász.

1.4 Des méthodes symboliques pour la factorisation

Dans cette section nous allons présenter trois méthodes symboliques "modernes" pour la factorisation absolue. La première méthode (l'algorithme TKTD) nous montre comment nous pouvons passer du problème de la factorisation absolue à celui de la factorisation rationnelle. La deuxième méthode (l'algorithme Duval) étudie le quotient $K = \mathbb{K}(X)[Y]/(P(X,Y))$ et ramène le problème à calculer une base d'entiers. La troisième (l'algorithme Gao) proviendra d'une étude sur une équation aux dérivées partielles. Pour finir nous présenterons un algorithme de factorisation rationnelle (l'algorithme Lecerf) qui combine les idées de S. Gao avec un schéma type "remonter-recombiner". Mais tout d'abord nous allons présenter un algorithme simple pour la factorisation absolue des polynômes de $\mathbb{Q}[X_1, \dots, X_n]$. Ensuite nous donnerons deux méthodes "classiques" qui nous permettrons de réduire l'étude de la factorisation en n variables à celle des polynômes en une ou deux variables.

1.4.1 Existence d'un algorithme de factorisation absolue pour les polynômes à coefficients rationnels

Dans cette partie nous allons introduire certains résultats sur la hauteur d'un polynôme de $\mathbb{C}[X_1, \dots, X_n]$. Ces résultats nous permettrons d'obtenir simplement un algorithme de factorisation absolue pour les polynômes de $\mathbb{Q}[X_1, \dots, X_n]$.

Définition 11. *Soit $P(X_1, \dots, X_n) = \sum a_{i_1,\dots,i_n} X_1^{i_1} \dots X_n^{i_n}$ un polynôme de $\mathbb{C}[X_1, \dots, X_n]$. On appelle hauteur de P (notée $\|P\|_\infty$) la valeur suivante :*

$$\|P\|_\infty = \max |a_{i_1,\dots,i_n}|.$$

Nous avons la propriété suivante (voir [Sch00]) :

Proposition 6. *Soient $P_1, \dots, P_k$ des polynômes de $\mathbb{C}[X_1, \dots, X_n]$. Nous avons alors :*

$$\prod_{i=1}^{k} \|P_i\|_\infty \leq 2^\nu \| \prod_{i=1}^{k} P_i \|_\infty$$

où $\nu = \displaystyle\sum_{i=1}^{k} \sum_{j=1}^{n} \deg_{X_j}(P_i)$.

Soit $P(X_1, \dots, X_n)$ un polynôme de $\mathbb{Q}[X_1, \dots, X_n]$, unitaire en la variable X_n et de degré total d. Soient $P = Q_1 \dots Q_t$ la factorisation en irréductibles de P dans $\mathbb{Q}[X_1, \dots, X_n]$, et $Q_i = P_{i,1} \dots P_{i,s_i}$ la factorisation absolue de Q_i. Nous pouvons supposer les facteurs $P_{i,j}$ unitaires en X^n. Nous avons alors d'aprés la proposition 6 :

$$(*) \quad \|P_{i,j}\|_\infty \leq 2^{nd^2} \|P\|_\infty, \text{ pour } j = 1, \dots, s_i.$$

En effet, nous avons au plus d facteurs en n variables de degré total d. A présent comme $\|P_{i,j}\|_\infty$ est borné nous allons en déduire qu'il n'y a qu'un nombre fini de facteurs possibles pour P.

Définition 12. *Soient α un nombre agébrique, et $f_\alpha(T)$ son polynôme minimal. On note*

$$f_\alpha(T) = T^d + a_{d-1}T^{d-1} + \ldots + a_0.$$

On pose alors

$$H_{alg}(\alpha) = \max_{i=0,\ldots,d-1} |a_i|.$$

Définition 13. *Soit $P(X_1, \ldots, X_n) = \sum a_{i_1,\ldots,i_n} X_1^{i_1} \ldots X_n^{i_n} \in \overline{\mathbb{Q}}[X_1, \ldots, X_n]$. On pose :*

$$H_{alg}(P) = \max\big(H_{alg}(a_{i_1,\ldots,i_n})\big).$$

Les polynômes $P_{i,j}$ étant conjugués sur $\mathbb{Q}$, d'aprés $(*)$ nous obtenons :

$$(**) \quad H_{alg}(P_{i,j}) \leq \binom{d}{\lfloor d/2 \rfloor} 2^{nd^3} \|P\|_\infty^d.$$

Nous avons donc obtenu une condition nécessaire sur les facteurs de P. De plus, il n'y a qu'un nombre fini de polynômes de $\overline{\mathbb{Q}}[X_1, \ldots, X_n]$ qui vérifient la condition nécessaire $(**)$. Ainsi nous obtenons un algorithme de factorisation absolue :
Nous considérons tous les polynômes de $\overline{\mathbb{Q}}[X_1, \ldots, X_n]$ dont les coefficients $a_{i_1,\ldots,i_n}$ sont des nombres algébriques de degré inférieurs à d, et vérifiant la condition $(**)$. Ensuite, pour chaque polynôme obtenu nous regardons s'il divise P.

Cet algorithme nous permet donc de ramener le problème de la factorisation absolue à un nombre fini de tests de divisibilités.
Nous avons exhiber un algorithme de factorisation absolue. Cependant cet algorithme n'est pas réalisable en pratique, car il y a un nombre exponentiel de divisions à effectuer.

1.4.2 Comment se ramener à une ou deux variables

Ici nous présentons deux méthodes classiques et différentes pour se ramener à un polynôme en une seule ou deux variables :

La méthode de Hensel

La première méthode consiste à évaluer le polynôme $P(X_1, \ldots, X_n)$ appartenant à $\mathbb{K}[X_1, \ldots, X_n]$ en des points $x_2, \ldots, x_n$. Nous obtenons alors le polynôme $\bar{P}(X_1) = P(X_1, x_2, \ldots, x_n)$ que nous factorisons. Ensuite à l'aide d'une remontée de Hensel $(X_2 - x_2, \ldots, X_n - x_n)$-adique nous obtenons des polynômes $P_i(X_1, \ldots, X_n) \in \mathbb{K}[X_1, \ldots, X_n]$ suceptibles d'être des facteurs de $P(X_1, \ldots, X_n)$.

Il faut alors tester si $P_i(X_1,\ldots,X_n)$ divise $P(X_1,\ldots,X_n)$.

En effet la situation suivante peut se produire :

$$\begin{aligned} P(X,Y) &= X^4+(12Y^3-18Y^2-18Y+12)X^3+(30Y^3-72Y^2+42Y-36)X^2 \\ &\quad +(-432Y^3+648Y^2+648Y-432)X-432Y^3+2592Y^2-2160Y \end{aligned}$$

est un polynôme irréductible dans $\mathbb{Q}[X,Y]$, et lorsque l'on prend $Y=0$ on obtient :

$$\bar{f}(X)=(X-6)X(X+6)(X+12)$$

qui est réductible.

Ainsi le facteur $(X-6)$ ne donnera pas de facteur de P, seul $(X-6)X(X+6)(X+12)$ donnera aprés remontée le "facteur" de P. Il nous faut donc tester 7 facteurs candidats avant de trouver que P est irrédutible. (Cet exemple est tiré de [Kal85c]).

Remarque : Nous pouvons aussi à la place d'une évaluation $X_i=x_i$ effectuer une substitution du type Bertini : $X_i=a_iX+b_iY+c_i$.

La méthode de Kronecker

Voici une autre approche connue sous le nom d'algorithme de Kronecker. Soit $P(X,Y)$ un polynôme de $\mathbb{K}[X,Y]$ tel que $deg_X P=m$, $deg_Y P=n$ nous pouvons considérer $\bar{P}(Z)=P(Z^d,Z)$ où $d=\max(n,m)+1$.

Dans ce cas si $P(X,Y)=P_1(X,Y)P_2(X,Y)$ dans $\mathbb{K}[X,Y]$ alors $\bar{P}(Z)=P(Z^d,Z)=P_1(Z^d,Z)P_2(Z^d,Z)$.

D'autre part si $P_1(X,Y)=\sum_{\substack{0\le i\le m\\0\le j\le n}} a_{i,j}X^iY^j$

alors $P_1(Z^d,Z)=\sum_{\substack{0\le i\le m\\0\le j\le n}} a_{i,j}Z^{di}Z^j=\sum_{\substack{0\le i\le m\\0\le j\le n}} a_{i,j}Z^{di+j}=\sum_{\substack{0\le i\le m\\0\le j\le n}} a_{i,j}Z^{e(i,j)}$.

Comme l'écriture $e(i,j)=di+j$ est la division euclidienne de $e(i,j)$ par d nous retrouvons aisément $P_1(X,Y)$ à l'aide de $P_1(Z^d,Z)$. Ainsi comme dans la méthode précédente nous devons tester tous les diviseurs de $\bar{P}(Z)$ les transformer en un polynôme $P_1(X,Y)$ de $\mathbb{K}[X,Y]$, et, vérifier si $P_1(X,Y)$ divise $P(X,Y)$.

En effet, ici aussi, nous pouvons obtenir des facteurs de $\bar{P}(Z)$ qui ne donnent pas de facteurs de $P(X,Y)$. En voici un exemple (tiré de [Kal85c]) :

$P(X,Y)=Y^2X^2-30Y^2+273XY-820X^2+576$ est irréductible dans $\mathbb{Q}[X,Y]$. Dans ce cas $d=3$ et $\bar{P}(Z)=(Z-4)(Z-3)(Z-2)(Z-1)(Z+1)(Z+2)(Z+3)(Z+4)$.

Le facteur $(Z - 4)$ ne donnera donc pas de facteurs irréductibles pour $P(X, Y)$. Avec cette méthode il faut tester 127 facteurs candidats avant de trouver que P est irréductible.

Remarque :
La complexité de tels algorithmes est exponentielle en le degré de P (car P peut être irréductible et $\bar{P}$ peut se décomposer en un produit de facteurs linéaires).

1.4.3 L'algorithme TKTD

Nous allons décrire ici l'algorithme TKTD, où TKTD est un acronyme pour Trager/Kaltofen/Traverso/Dvornicich car cet algorithme a été proposé dans [Tra85], [Kal85b], [DT89]. Pour en faciliter l'exposé nous allons nous restreindre au cas des polynômes en 2 variables. L'algorithme comporte deux étapes : Tout d'abord calculer un corps K contenant le corps $\mathbb{K}[\alpha]$ engendré par les coefficients d'un facteur absolu ; puis factoriser $P(X, Y)$ dans $K[X, Y]$.
En supposant que nous ayons un algorithme efficace pour la factorisation rationnelle. Le problème ici est donc d'obtenir le corps K. Nous allons montrer comment obtenir un tel corps.

Définition 14. *Soit $(\alpha, \beta) \in \overline{\mathbb{K}}^2$. On dit que (α, β) est une solution simple de $P(X, Y) \in \mathbb{K}[X, Y]$ si $P(\alpha, \beta) = 0$ et l'une au moins des dérivées $\dfrac{\partial P}{\partial X}, \dfrac{\partial P}{\partial Y}$ est non nulle en (α, β).*

On remarque qu'une solution simple est racine d'un et d'un seul facteur absolument irréductible de P.

Théorème 12. *Soit (α, β) une solution simple de $P(X, Y) \in \mathbb{K}[X, Y]$. On a alors : un des facteurs absolument irréductible de P est à coefficients dans $\mathbb{K}[\alpha, \beta]$.*

Démonstration. Soit $P = P_1 \dots P_s$ la factorisation en irréductibles de P dans $\mathbb{K}[\alpha, \beta][X, Y]$. On note P_1 le facteur de P tel que $P_1(\alpha, \beta) = 0$. Supposons que P_1 soit réductible dans $\overline{\mathbb{K}}[X, Y]$. Nous allons montrer que cela est impossible. Si P_1 est réductible dans $\overline{\mathbb{K}}[X, Y]$ alors il s'écrit $P_1 = \prod_{i=1}^{k} F_i$ où F_i sont des facteurs absolus de P_1. On note F_1 le facteur tel que $F_1(\alpha, \beta) = 0$. D'après le lemme fondamental il existe un $\mathbb{K}[\alpha, \beta]$-homomorphisme tel que $\sigma(F_1) = F_2$. D'où $F_2(\alpha, \beta) = 0$. Cela signifie que (α, β) n'est pas un point simple, P_1 est donc irréductible dans $\overline{\mathbb{K}}[X, Y]$. □

Nous prendrons comme hypothèse, pour l'algorithme TKTD, P sans facteur carré. Cette hypothèse n'est pas trop forte car nous pouvons toujours nous ramener à cette situation.

Algorithme TKTD

ENTRÉE : $P(X,Y) \in \mathbb{K}[X,Y]$ sans carré.
SORTIE : Un facteur absolument irréductible de P.

1. Prendre $\alpha \in \mathbb{K}$ tel que $P(\alpha, Y)$ soit sans carré
2. Prendre β une racine $P(\alpha, Y)$.
3. Factoriser $P(X,Y)$ sur $\mathbb{K}[\beta][X,Y]$
4. Retourner le facteur P_1 tel que $P_1(\alpha, \beta) = 0$

Exemple (tiré de [Rag97]) :
Soit $P(X,Y) = Y^{10} - 2X^2Y^4 + 4X^6Y^2 - 2X^{10}$. Alors
$P(1,Y) = Y^{10} - 2Y^4 + 4Y^2 - 2$ est irréductible dans $\mathbb{Q}[Y]$. On pose :
$K = \frac{\mathbb{Q}[T]}{(P(1,T))}$. A présent nous allons utiliser Maple pour factoriser $P(X,Y)$ dans $K[X,Y]$.

```
>alias(beta=RootOf(x^{10}-2x^4+4x^2-2));
>factor(P,beta);
```

$(Y^5 - 2Y^2X\beta + 2Y^2X\beta^3 - Y^2X\beta^5 - Y^2X\beta^7 - Y^2X\beta^9 + 2X^5\beta - 2X^5\beta^3 + X^5\beta^5 + X^5\beta^7 + X^5\beta^9)(Y^5 + 2Y^2X\beta - 2Y^2X\beta^3 + Y^2X\beta^5 + Y^2X\beta^7 + Y^2X\beta^9 - 2X^5\beta + 2X^5\beta^3 - X^5\beta^5 - X^5\beta^7 - X^5\beta^9)$

Le temps mis pour effectuer ce calcul est de 691.5 secondes. Ici $P_1(X,Y) = Y^5 + (\beta^9 + \beta^7 + \beta^5 - 2\beta^3 + 2\beta)Y^2X + (-\beta^9 - \beta^7 - \beta^5 + 2\beta^3 - 2\beta)X^5$ est un facteur absolu, il vérifie $P_1(\alpha, \beta) = 0$.
Cet algorithme est simple mais l'inconvénient est que le corps K obtenu est trop gros. Dans l'exemple ci-dessus nous aurions pu prendre $K = \mathbb{Q}[\sqrt{2}]$. Le calcul $factor(P, sqrt(2))$ ne prend alors que 0.27 secondes.

REMARQUE : Dans sa thèse J.-F. Ragot a utilisé le théorème 12 pour donner une méthode permettant d'obtenir les facteurs linéaires d'un polynôme en n variables.

1.4.4 L'algorithme de Duval

Dans cette partie nous allons juste donner les grandes lignes de cet algorithme (pour plus de précision se reporter à [Duv91], [Duv83] et [Rag97]). Cet algorithme s'applique aux polynômes sans carrés à coefficients dans un corps de caractéristique nulle ou suffisamment grande. Dans ce qui suit nous étudions le cas où P est irréductible dans $\mathbb{K}[X,Y]$.

Définition 15. *Soit $P(X,Y) \in \mathbb{K}[X,Y]$ un polynôme irréductible. $K = \mathbb{K}(X)[Y]/(P(X,Y))$ est une extension algébrique de degré fini de $\mathbb{K}(X)$. On appelle corps des constantes (noté $\mathbb{K}_c$) l'ensemble des éléments de K algébriques*

sur $\mathbb{K}$.
On appelle anneau des entiers (noté $\mathcal{O}$) l'ensemble des éléments de K qui sont racines d'un polynôme unitaire à coefficients dans $\mathbb{K}[X]$.

Proposition 7. *$\mathcal{O}$ est un $\mathbb{K}[X]$ module libre de rang égal au degré total de P. Une base de ce module s'appelle une base entière. De plus $\mathbb{K}_c$ est un sous corps de $\mathcal{O}$.*

La connaissance de $\mathbb{K}_c$ nous donne des informations sur la factorisation absolue de P. En effet, nous avons :

Théorème 13. *Soit $P(X,Y)$ un polynôme irréductible dans $\mathbb{K}[X,Y]$. On a alors : $P(X,Y)$ est absolument irréductible $\iff \mathbb{K} = \mathbb{K}_c$.*

Corollaire 2. *Soient $P(X,Y)$ un polynôme irréductible dans $\mathbb{K}[X,Y]$, $s = [\mathbb{K}_c : \mathbb{K}]$ et x (respectivement y) l'image de X (respectivement Y) dans $K = \frac{\mathbb{K}(X)[Y]}{(P(X,Y))}$. Dans ce cas :*
La factorisation absolue de $P(X,Y)$ est $P(X,Y) = \prod_{i=1}^{s} \sigma_i(G(X,Y))$ où $G(x,Y)$ est le polynôme minimal de y sur $\mathbb{K}_c(x)$ et les σ_i sont les $\mathbb{K}$-homomorphismes de $\mathbb{K}_c$ dans $\overline{\mathbb{K}}$.

Nous pouvons alors donner un aperçu de la méthode :

Algorithme Duval
Entrée : $P(X,Y)$ un polynôme irréductible de $\mathbb{K}[X,Y]$.
Sortie : Un facteur absolument irréductible de P.

1. Calculer une base d'entiers de $\mathcal{O}$ sur $\mathbb{K}[X]$.
2. En déduire une base de $\mathbb{K}_c$ sur $\mathbb{K}$.
3. Calculer un facteur absolument irréductible.

Remarques :
- Le corps $\mathbb{K}_c$ est le corps engendré par les coefficients d'un facteur absolument irréductible. Considérons, par exemple le polynôme $P(X,Y) = Y^4 - 2X^2$, qui est irréductible dans $\mathbb{Q}[X,Y]$. Sa factorisation absolue est

$$P(X,Y) = (Y^2 - \sqrt{2}X)(Y^2 + \sqrt{2}X).$$

Nous avons $\sqrt{2} = y^2/x \in \mathbb{K}_c$, où y (respectivement x) désigne l'image de Y (respectivement l'image de X) dans K. En effet, nous avons dans K l'égalité suivante :

$$\left(\frac{y^2}{x}\right)^2 - 2 = \frac{y^4}{x^2} - 2 = y^4 - 2x^2 = 0.$$

- L'étape 1 est l'étape dominante de cet algorithme. L'étape 3 s'obtient à l'aide d'un calcul de *pgcd*.

1.4.5 L'algorithme de Gao

L'algorithme de S. Gao [Gao03] est inspiré de l'étude de W. Ruppert sur la factorisation absolue (voir [Rup86], [Rup99]), et, de l'algorithme de Niederreiter (voir [Nie93], [MŞ99]) pour la factorisation des polynômes univariés à coefficients dans un corps fini.
En effet, soit F un polynôme de $\mathbb{F}_p[X]$ se factorisant dans $\mathbb{F}_p[X]$ de la façon suivante : $F = G_1 \dots G_s$. Dans l'algorithme de Niederreiter, nous considérons une équation différentielle ordinaire dont l'espace des solutions est engendré par les polynômes FG_i'/G_i. Ensuite à l'aide d'un calcul de pgcd nous en déduisons la factorisation de F. Dans son algorithme S. Gao reprend l'équation aux dérivées partielles de W. Ruppert, et montre que l'espace des solutions de cette équation est engendré par les couples $(\dfrac{P}{P_i}\dfrac{\partial P_i}{\partial X}, \dfrac{P}{P_i}\dfrac{\partial P_i}{\partial Y})$. Ensuite, S. Gao obtient les facteurs absolument irréductibles à partir de cet espace de solutions et d'un calcul de pgcd.

Dans ce qui suit nous supposerons $P(X,Y)$ irréductible dans $\mathbb{K}[X,Y]$ et $\mathbb{K}$ de caractéristique 0. Nous faisons ces hypothèses pour obtenir plus de simplicité dans l'exposé de la méthode ; mais tout ce qui suit peut être généralisé au cas où $pgcd(P, \frac{\partial P}{\partial X}) = 1$ et $\mathbb{K}$ de caractéristique suffisamment grande. Cela signifie que l'algorithme de Gao nous permet d'obtenir à la fois la factorisation rationnelle et la factorisation absolue de P.

A présent nous allons présenter les grandes lignes de cet algorithme :
Soit $P(X,Y) = P_1(X,Y) \dots P_s(X,Y)$ la factorisation absolue de P. Nous obtenons alors :

$$\frac{\partial P}{\partial X} = \sum_{i=1}^{s} \frac{\partial P_i}{\partial X} \prod_{j \neq i} P_j, \quad \frac{\partial P}{\partial Y} = \sum_{i=1}^{s} \frac{\partial P_i}{\partial Y} \prod_{j \neq i} P_j.$$

On pose :

$$g_i(X,Y) = \frac{\partial P_i}{\partial X} \prod_{j \neq i} P_j \text{ et } h_i(X,Y) = \frac{\partial P_i}{\partial Y} \prod_{j \neq i} P_j.$$

Nous remarquons alors que nous avons :

$$\begin{cases} \dfrac{\partial}{\partial X}(\log P_i) = \dfrac{1}{P_i}\dfrac{\partial P_i}{\partial X} = \dfrac{g_i}{P}, \\ \dfrac{\partial}{\partial Y}(\log P_i) = \dfrac{1}{P_i}\dfrac{\partial P_i}{\partial Y} = \dfrac{h_i}{P}. \end{cases}$$

Le résultat classique de Schwarz sur les dérivées partielles donne :

$$\frac{\partial}{\partial Y}(\frac{g_i}{P}) = \frac{\partial}{\partial X}(\frac{h_i}{P}).$$

Ainsi les couples (g_i, h_i) forment des solutions à l'équation :

$$(\star) \quad \frac{\partial}{\partial Y}(\frac{g}{P}) = \frac{\partial}{\partial X}(\frac{h}{P}).$$

avec les conditions

$$(\star\star) \begin{cases} \deg_X(g) \leq \deg_X(P) - 1 \,,\, \deg_Y(g) \leq \deg_Y(P) \\ \deg_X(h) \leq \deg_X(P) \,,\, \deg_Y(h) \leq \deg_Y(P) - 1. \end{cases}$$

Définition 16. *On pose* $E = \{(g,h) \in (\mathbb{K}[X,Y])^2 \mid (\star)$ *et* $(\star\star)$ *sont vérifiées*$\}$.

E est un $\mathbb{K}$ espace vectoriel $E \otimes \overline{\mathbb{K}}$ contient les couples (g_i, h_i). De plus ces couples sont linéairement indépendants.
En effet, notons $\mathbb{K}[\alpha]$ le corps engendré par les coefficients de P_1, et $\sigma_1, \ldots, \sigma_s$ les $\mathbb{K}$-homomorphismes de $\mathbb{K}[\alpha]$ dans $\overline{\mathbb{K}}$. L'égalité : $\frac{g_1}{P} = \frac{\partial P_1}{\partial X}\frac{1}{P_1}$ nous montre que le corps des coefficients de g_1 est $\mathbb{K}[\alpha]$. Supposons à présent que nous ayons une relation : $\sum_{i=1}^s \lambda_i g_i = 0$, avec $\lambda_i \in \overline{\mathbb{K}}$. Cela signifie $\sum_{i=1}^s \lambda_i \sigma_i(g_1) = 0$. Cela implique alors que les $\mathbb{K}$-homorphismes σ_i sont liés. Ceci est impossible d'après le lemme de Dedekind (voir [Sam67] page 47).
Le théorème de Gao nous dit qu'en plus la famille des (g_i, h_i) est génératrice.

Théorème 14. *En conservant les notations précédentes nous avons :*

$$dim_{\mathbb{K}}(E) = dim_{\overline{\mathbb{K}}}(E \otimes \overline{\mathbb{K}}) = s.$$

De plus $\{(g_1,h_1), \ldots, (g_s,h_s)\}$ *est une base de* $E \otimes \overline{\mathbb{K}}$.

La démonstration de ce théorème repose sur une décomposition en éléments simples et sur un calcul de résidu.

Nous remarquons alors que le système $(\star)$ peut se réécrire :

$$(\star')\, P(\frac{\partial g}{\partial Y} - \frac{\partial h}{\partial X}) + h\frac{\partial P}{\partial X} - g\frac{\partial P}{\partial Y} = 0.$$

L'espace vectoriel E peut donc être vu comme le noyau d'un système linéaire dépendant des coefficients de P. Le théorème de Gao nous dit alors que la dimension de ce noyau est égal au nombre de facteurs absolument irréductible de P.
Le corollaire suivant nous montre que certains éléments de E nous permettent d'obtenir un facteur absolument irréductible.

Corollaire 3. *Soit* $g = \sum_{i=1}^s \lambda_i g_i$*, avec* $\lambda_i \neq \lambda_j$ *lorsque* $i \neq j$*. Nous avons alors :*

$$P_1 = pgcd(P, g - \lambda_1\frac{\partial P}{\partial X}).$$

Démonstration. Nous avons $\frac{\partial P}{\partial X} = \sum_{i=1}^s g_i$. Donc P_1 divise $g - \lambda_1\frac{\partial P}{\partial X}$ car $g - \lambda_1\frac{\partial P}{\partial X} = \sum_{i=2}^s(\lambda_i - \lambda_1)g_i$. De plus si $i > 1$, P_i ne divise pas $g - \lambda_1\frac{\partial P}{\partial X}$. Cela donne donc le résultat souhaité. □

Afin de vérifier qu'un élément g vérifie l'hypothèse du corollaire, nous avons besoin d'une méthode. Celle ci est donnée par le théorème suivant :

Théorème 15. *Soit $G_1, \ldots, G_s$ une base de $pr_1(E)$ sur $\mathbb{K}$. Pour chaque $g \in pr_1(E)$ il existe une unique matrice $A = (a_{i,j})$ à coefficients dans $\mathbb{K}$, de dimension $s \times s$ telle que :*

$$gG_i \equiv \sum_{i=1}^{s} a_{i,j} G_j \frac{\partial P}{\partial X} \mod P.$$

Soit $E_g(T) = det(Id.T - A) = \prod_{i=1}^{s}(T - \lambda_i)$, le polynôme caractéristique de A où $\lambda_i \in \overline{\mathbb{K}}$. Nous avons alors $g = \sum_{i=1}^{s} \lambda_i g_i$.

Ainsi si $E_g(T)$ a toutes ses racines distinctes nous pouvons appliquer le corollaire 3. De plus le calcul de *pgcd* s'effectue dans $\mathbb{K}[\lambda_1][X,Y]$, et $\mathbb{K}[\lambda_1]$ est donc un corps contenant celui engendré par les coefficients de P_1, et vérifiant $[\mathbb{K}[\lambda_1] : \mathbb{K}] \leq s$. Nous en déduisons : $E_g(T)$ est le polynôme minimal d'un élément primitif sur $\mathbb{K}$ engendrant le corps des coefficients de P_1. Nous venons donc d'obtenir l'algorithme suivant :

Algorithme Gao

ENTRÉE : $P(X,Y) \in \mathbb{K}[X,Y]$, irréductible dans $\mathbb{K}[X,Y]$.
SORTIE : $P_1(X,Y) \in \mathbb{K}[\alpha][X,Y]$ un facteur absolument irréductible, et f_α le polynôme minimal de α.

1. Trouver $G_1, \ldots, G_s$ une base de $pr_1(E)$. Si $s = 1$ alors $P_1 = P$.
2. Prendre au hasard des éléments a_i dans S. Poser $g = \sum_{i=1}^{s} a_i G_i$.
3. Calculer $E_g(T)$. Si $E_g(T)$ n'a pas toutes ses racines distinctes alors retourner en 2.
4. Poser $f_\alpha(T) = E_g(T)$, et $P_1(X,Y) = pgcd(P(X,Y), g(X,Y) - \alpha \frac{\partial P}{\partial X})$.

REMARQUE :
La condition ($\star\star$) implique que g possède $m(n+1)$ coefficients, et, h en possède $(m+1)n$. De plus le système linéaire ($\star'$) donne $4mn$ équations. Ainsi lorsque $n = \deg(P) = \deg_Y(P) = \deg_X(P)$, nous obtenons un système linéaire avec $O(n^2)$ inconnues et $O(n^2)$ équations. Nous verrons en 1.4.6 et dans le chapitre 4 qu'en utilisant l'équation ($\star$), et, en suivant une approche "remonter-recombiner", nous pouvons nous ramener à l'étude d'un système de taille plus petite. Nous verrons aussi comment obtenir $g = \sum_{i=1}^{s} \lambda_i G_i$ avec $\lambda_i \neq \lambda_j$ lorsque $i \neq j$, de manière déterministe.

1.4.6 Un algorithme de factorisation rationnelle : L'algorithme de Lecerf

Nous allons présenter ici un algorithme de factorisation rationnelle. Il est légitime de présenter cet algorithme dans un état de l'art sur la factorisation absolue pour trois raisons. La première est que la plupart des algorithmes de factorisation absolue prennent en entrée un polynôme irréductible dans $\mathbb{K}[X,Y]$.

Cela suppose donc que pour obtenir un algorithme de factorisation absolue complet (i.e. sans hypothèse sur le polynôme d'entrée) nous avons besoin d'un algorithme de factorisation rationnelle. La deuxième raison est que cet algorithme de factorisation rationnelle est une adaptation de l'algorithme de Gao qui lui est un algorithme de factorisation absolue. Enfin nous verrons dans le chapitre 4 qu'à l'aide des idées présentées ici nous pouvons obtenir un algorithme de factorisation absolue.

ATTENTION :
Dans cette sous-section nous allons étudier la factorisation rationnelle de $P \in \mathbb{K}[X,Y]$: $P = P_1 \dots P_s$ où $P_i \in \mathbb{K}[X,Y]$ est irréductible dans $\mathbb{K}[X,Y]$.

L'algorithme de G. Lecerf, [Lec04b], repose sur le principe "remonter-recombiner". C'est à dire nous factorisons $P(X,Y) \in \mathbb{K}[X,Y]$ dans $\mathbb{K}[[X]][Y]$, $P(X,Y) = \prod_{i=1}^t \mathfrak{F}_i(X,Y)$, où $\mathfrak{F}_i(X,Y) \in \mathbb{K}[[X]][Y]$. Puis nous cherchons à recombiner les facteurs $\mathfrak{F}_i(X,Y)$ afin d'obtenir $P_i(X,Y) = \prod_{j=i_1}^{i_k} \mathfrak{F}_j(X,Y)$, un facteur irréductible dans $\mathbb{K}[X,Y]$ de $P(X,Y)$.
Nous pouvons aussi représenter les facteurs irréductibles de la façon suivante :
$P_i = \prod_{j=1}^t \mathfrak{F}_j^{\mu_{P_i,j}}$, où $\mu_{P_i} = (\mu_{P_i,1}, \dots, \mu_{P_i,t}) \in \{0,1\}^t$. L'étape de recombinaison revient alors (comme nous l'avons vu pour l'algorithme de van Hoeij) à trouver les 0-1 vecteurs μ_{P_i}. Nous remarquons de plus que si nous obtenons l'espace vectoriel $Vect(\mu_{P_1}, \dots, \mu_{P_s})$, alors à l'aide d'une mise sous forme échelonnée réduite d'une base nous obtiendrons les vecteurs μ_{P_i}.

A présent nous allons donner les grandes lignes de l'algorithme de Lecerf. Cet algorithme propose un moyen efficace pour l'étape de recombinaison. Nous avons vu dans l'algorithme de Gao que l'espace engendré par les $(g_i = \frac{\partial P_i}{\partial X}\frac{P}{P_i}, h_i = \frac{\partial P_i}{\partial Y}\frac{P}{P_i})$ joue un rôle privilégié. Nous allons voir qu'ici avec les P_i irréductibles dans $\mathbb{K}[X,Y]$ les polynômes g_i et h_i vont aussi jouer un rôle.
Puisque nous étudions le problème de la recombinaison nous allons exprimer les polynômes g_i et h_i en fonction des polynômes $\mathfrak{F}_i \in \mathbb{K}[[X]][Y]$.
Nous avons $\dfrac{\partial P_i}{\partial Y} = \sum_{j=1}^t \mu_{P_i,j} \dfrac{\partial \mathfrak{F}_j}{\partial Y} \prod_{k=1,k\neq j}^t \mathfrak{F}_k^{\mu_{P_i,k}}$, ce qui donne :

$$g_i = \frac{P}{P_i}\frac{\partial P_i}{\partial Y} = \sum_{j=1}^t \mu_{P_i,j}\frac{\partial \mathfrak{F}_j}{\partial Y} \prod_{k=1,k\neq j}^t \mathfrak{F}_k,$$

car $\dfrac{P}{P_i} = \prod_{k=1}^t \mathfrak{F}_k^{1-\mu_{P_i,k}}$. De même nous avons :

$$h_i = \frac{P}{P_i}\frac{\partial P_i}{\partial X} = \sum_{j=1}^t \mu_{P_i,j}\frac{\partial \mathfrak{F}_j}{\partial X} \prod_{k=1,k\neq j}^t \mathfrak{F}_k.$$

Nous voyons donc qu'à chaque facteur P_i nous pouvons associer un triplet (μ_{P_i}, g_i, h_i). De plus d'après ce qui précède ce triplet appartient à l'espace vectoriel

L_σ défini comme suit :

Définition 17. *Soit $P(X,Y) \in \mathbb{K}[X,Y]$, et $P(X,Y) = \prod_{i=1}^{t} \mathfrak{F}_i(X,Y)$ sa factorisation dans $\mathbb{K}[[X]][Y]$. On pose :*

$$\hat{\mathfrak{F}}_i = \prod_{j=1, j\neq i}^{t} \mathfrak{F}_j,$$

et

$$\begin{aligned} L_\sigma = \; & \{(\mu, g, h) \in \mathbb{K}^t \times \mathbb{K}[X,Y]_{n-1} \times \mathbb{K}[X,Y]_{n-1} \mid \\ & g = \textstyle\sum_{j=1}^{t} \mu_j \frac{\partial \mathfrak{F}_j}{\partial Y} \hat{\mathfrak{F}}_j \mod (X,Y)^\sigma, \\ & h = \textstyle\sum_{j=1}^{t} \mu_j \frac{\partial \mathfrak{F}_j}{\partial X} \hat{\mathfrak{F}}_j \mod (X,Y)^\sigma + (X)^{\sigma-1}\}. \end{aligned}$$

Cet espace vectoriel met en évidence la relation existant entre g_i, h_i, l'exposant μ_{P_i}, et, les $\mathfrak{F}_j \in \mathbb{K}[[X]][Y]$. Ici au lieu d'étudier g et h pour obtenir une factorisation, nous allons regarder la projection sur $\mathbb{K}^t$, $\pi_{\mathbb{K}^t}(L_\sigma)$. Les calculs effectués ci-dessus montrent que nous avons $Vect(\mu_{P_1}, \ldots, \mu_{P_s}) \subset L_\sigma$ pour $\sigma \geq 1$. G. Lecerf a montré le résultat suivant :

Théorème 16. *Soit $P(X,Y) \in \mathbb{K}[X,Y]$ un polynôme tel que $\deg_Y(P) = \deg(P) = n$, et, $Res_Y(P, \dfrac{\partial P}{\partial Y}(0)) \neq 0$. Si $\mathbb{K}$ est un corps de caractéristique 0 ou supérieur à $n(n-1)+1$, et $\sigma \geq 2n$ alors nous avons :*

$$L_\sigma = Vect_{\mathbb{K}}\Big((\mu_{P_1}, \hat{P}_1 \frac{\partial P_1}{\partial Y}, \hat{P}_1 \frac{\partial P_1}{\partial X}), \ldots, (\mu_{P_s}, \hat{P}_s \frac{\partial P_s}{\partial Y}, \hat{P}_s \frac{\partial P_s}{\partial X})\Big)$$

$$\pi_{\mathbb{K}^t}(L_\sigma) = Vect(\mu_1, \ldots, \mu_s).$$

Ce théorème montre que si nous effectuons une remontée de Hensel à un ordre $\sigma \geq 2n$ des facteurs $\mathfrak{F}_i \in \mathbb{K}[[X]][Y]$ de P, alors nous pouvons les recombiner en étudiant $\pi_{\mathbb{K}^t}(L_\sigma)$. Le lemme suivant nous montre comment obtenir effectivement $\pi_{\mathbb{K}^t}(L_\sigma)$. Nous notons $coef(\mathfrak{F}, X^i Y^j)$ le coefficient en $X^i Y^j$ d'un polynôme $\mathfrak{F}$ de $\mathbb{K}[[X]][Y]$.

Lemme 4. *Sous les hypothèses du théorème, le système D_σ suivant en les inconnues $l_1, \ldots, l_s$ a pour noyau $\pi_{\mathbb{K}^t}(L_\sigma)$.*

$$D_\sigma = \begin{cases} \sum_{i=1}^{t} l_i \, coef(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial Y}, X^i y^j) = 0, \; k \leq n-1, \; n \leq j+k \leq \sigma - 1, \\ \sum_{i=1}^{t} l_i \, coef(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial X}, X^i y^j) = 0, \; k \leq n-1, \; j \leq \sigma - 2, \; n \leq j+k \leq \sigma - 1. \end{cases}$$

Grâce aux résultats précédents le problème de recombinaison est ramené à de l'algèbre linéaire. Le système D_σ possède t inconnues, et $O(\sigma n)$ équations. Nous rappelons que $t \leq n$ et que nous pouvons prendre $\sigma = 2n$. Le nombre d'inconnues du système D_σ est donc inférieur à celui du système obtenue dans la méthode de

S. Gao, et, l'ordre de grandeur du nombre d'équations est le même. Nous pouvons alors en déduire que cet algorithme de factorisation rationnelle est plus efficace que celui découlant de l'algorithme Gao.

Remarque :
Cet algorithme est le prolongement du travail [BLS+04] effectué par A. Bostan, G. Lecerf, B. Salvy, E. Schost et B. Wiebelt présenté à ISSAC'04. Dans l'article [BLS+04], seule la condition : $g = \sum_{i=1}^{t} l_i \frac{\mathfrak{F}_i}{\partial Y} \hat{\mathfrak{F}}_i \mod X^\sigma$ est considérée. Les auteurs montrent alors que cette approche est équivalente à celle de Sasaki (voir section 1.5.2).

1.5 Des méthodes numériques pour la factorisation absolue

1.5.1 L'algorithme BCGW

Ce que nous appelons ici algorithme BCGW est l'algorithme du à C. Bajaj, J. Canny, T. Garrity, J. Warren (voir [BCGW93]). Cette méthode a été proposée en 1989. Celle-ci donne le nombre de facteurs absolument irréductibles, leur degré et ensuite une approximation numérique de ces facteurs.
Afin de présenter l'idée directrice de cette méthode nous rappelons le résultat suivant dont une preuve est donnée en annexe (voir annexe A).

Théorème 17. *Soient $P(X,Y) \in \mathbb{C}[X,Y]$ un polynôme sans facteur carré, $\Delta = \{x \mid Disc_Y(P)(x) = 0\}$, $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 \mid P(x,y) = 0\}$, pr_1 la projection qui envoie un couple $(x,y) \in \mathcal{C}$ sur $x \in \mathbb{C}$. On a alors :*
P est irréductible dans $\mathbb{C}[X,Y] \iff \mathcal{C} - pr_1^{-1}(\Delta)$ est connexe par arcs.

Donc nous voyons que dans le cas d'un polynôme réductible $P(X,Y) \in \mathbb{C}[X,Y]$ les composantes connexes de $\mathcal{C} - pr_1^{-1}(\Delta)$ sont les zéros des facteurs de P. L'algorithme BCGW propose alors de fabriquer une grille G dans $\mathbb{C}$ telle que chaque cellule contienne au plus un point de Δ, puis de "relever" cette grille pour obtenir un graphe $K = \mathcal{C} \cap pr_1^{-1}(G)$. Ce graphe K a alors le même nombre de composantes connexes que $\mathcal{C}$. Cela permet d'obtenir le nombre de facteurs irréductibles et leur degré. D'autre part chaque sommet du graphe correspond à un point sur une composante connexe de P. Donc si nous avons assez de points sur une composante connexe nous pouvons par interpolation en déduire les coefficients du facteur correspondant. Il faut remarquer que les points des composantes connexes ne sont connus que numériquement. En effet nous les obtenons en résolvant $P(x_0, Y) = 0$ où $x_0 \in \mathbb{C}$ est fixé. Cela explique pourquoi nous obtenons avec cette méthode une approximation des coefficients des facteurs.

1.5.2 L'algorithme de Sasaki

Dans ce qui suit nous considérons $P(X,Y) \in \mathbb{C}[X,Y]$ de degré n et nous supposons $P(X,0)$ sans facteurs carrés. Nous pouvons donc à l'aide d'une remontée de Hensel obtenir n "racines" $\varphi_i(X) \in \mathbb{C}[[X]]$ telles que :

$$P(X, \varphi_i(X)) = 0.$$

Un facteur P_1 de P s'écrit donc $\prod_{i\in I}(Y-\varphi_i(X))$. Ainsi les fonctions symétriques élémentaires en les $\varphi_i(X)$, où $i \in I$, sont des polynômes. Les sommes de Newton $\sum_{i\in I} \varphi_i^k(X)$ sont donc aussi des polynômes.

Ainsi en notant $\lceil \sum_{i=0}^{+\infty} a_i X^i \rceil^d = \sum_{i=0}^{d} a_i X^i$ nous obtenons pour d suffisamment grand :

$$\begin{cases} \lceil \varphi_1(X) + \ldots + \varphi_m(X) \rceil^d = h_1(X), \ \deg(h_1(X)) \leq 1 \\ \vdots \\ \lceil \varphi_1^m(X) + \ldots + \varphi_m^m(X) \rceil^d = h_m(X), \ \deg(h_m(X)) \leq m \end{cases}$$

si et seulement si $P_1(X,Y) = \prod_{i=1}^{m}(Y - \varphi_i(X))$ est un polynôme divisant P.

Cette équivalence est la base de l'algorithme.

Sasaki et ses coauteurs ont conjecturé dans [SSH92] que $d \geq n+1$ suffisait pour obtenir l'équivalence désirée. Or dans [BLS$^+$04] un contre exemple a été trouvé, et dans ce même article les auteurs démontre que nous avons l'équivalence pour $d \geq 3n-2$. Nous verrons dans l'algorithme Galligo/Rupprecht qu'avec des conditions de généricité l'égalité $\lceil \varphi_1(X) + \ldots + \varphi_m(X) \rceil^3 = h_1(x)$ suffit pour obtenir les facteurs de P.

Nous avons placé l'algorithme de Sasaki avec les méthodes numériques car les $\varphi_i(X)$ sont obtenus numériquement. En effet $\varphi_i(0)$ est obtenu en résolvant $P(0,Y)$, nous avons donc une valeur approchée de ce nombre, et il en est de même pour les autres coefficients de $\varphi_i(X)$. Grâce à différentes études d'erreurs T. Sasaki et ses coauteurs ont donné des algorithmes de factorisation approchée, et des tests d'irréductibilité pour des polynômes perturbés, voir [Nag02], [Sas01],[SSH92],[SSKS91].

1.5.3 L'algorithme Galligo/Rupprecht

Ici nous allons rappeler le fonctionnement de l'algorithme Galligo/Rupprecht. Celui-ci repose sur une propriété géométrique plus forte que celle vue pour l'algorithme BCGW.

Théorème 18 (Harris affine). *Soit $P(X,Y)$ un polynôme irréductible de $\mathbb{Q}[X,Y]$, unitaire en Y, qui admet la factorisation absolue suivante : $P = P_1 \cdots P_s$ avec $\deg(P_i) = m$. Nous considérons $\mathcal{C} = \{(x,y) \,|\, P(x,y) = 0\}$ et ses s composantes irréductibles $\mathcal{C}_i$ que nous projetons sur l'axe des x après un changement linéaire générique de coordonnées. Alors $\pi_1(\mathbb{C}-\Delta)$ où π_1 désigne le groupe fondamental et Δ le lieu discriminant, agit sur une fibre lisse comme le produit de groupes symétriques $\mathfrak{S}_m \times \cdots \times \mathfrak{S}_m$ (avec s facteurs).*

Des éléments pour la démonstration de ce théorème se trouvent en annexe (voir annexe A).

Notations : Dans cette partie $P(X,Y)$ désignera un polynôme irréductible de $\mathbb{Q}[X,Y]$ de degré total n, unitaire en Y et sans facteurs linéaires. Nous appellerons (H) cette hypothèse.

Nous remarquons que nous pouvons nous ramener au cas unitaire en effectuant un changement linéaire de coordonnées. D'autre part nous avons vu qu'il existe des algorithmes permettant d'obtenir les facteurs linéaires d'un polynôme [Hoh97], [Rag97]. Cette hypothèse n'est donc pas trop restrictive en pratique.

Présentation de l'algorithme

A présent nous allons donner l'idée générale se trouvant derrière cet algorithme. Nous détaillerons certains points par la suite.

Soient $x_0 \notin \Delta = \{x \in \mathbb{C} \,|\, Disc_Y(P)(x) = 0\}$, et $y_i(x_0)$ les racines de $P(x_0, Y)$. D'après le lemme de Hensel nous avons des séries formelles $\varphi_i(X) = y_i(x_0) + a_i(x_0)(X - x_0) + b_i(x_0)(X - x_0)^2 + \cdots$ vérifiant :

$$\begin{cases} \varphi_i(x_0) = y_i(x_0) \\ P(X, \varphi_i(X)) = 0. \end{cases}$$

Cela signifie en particulier : $P(X,Y) = \prod_{i=1}^{n}(Y - \varphi_i(X))$. Donc un facteur unitaire P_1 de P s'écrit $P_1(X,Y) = \prod_{i\in I}(Y - \varphi_i(X))$ où I est un sous-ensemble de $\{1, \ldots, n\}$. D'autre part nous pouvons aussi écrire P_1 sous cette forme : $P_1(X,Y) = Y^m + (AX + B)Y^{m-1} + \cdots + C$, où A, B, et C appartiennent à $\mathbb{C}$. Ainsi les relations racines-coefficients donnent en particulier :

$$\begin{aligned} AX + B &= \sum_{i\in I} \varphi_i(X) \\ &= \sum_{i\in I} y_i(x_0) + (\sum_{i\in I} a_i(x_0))(X - x_0) + (\sum_{i\in I} b_i(x_0))(X - x_0)^2 + \cdots \end{aligned}$$

Il vient alors $\sum_{i\in I} b_i(x_0) = 0$.

Nous voyons alors qu'une condition nécessaire pour que P_1 soit un facteur de P est : $\sum_{i \in I} b_i(x_0) = 0$. Or nous allons voir que cette condition est aussi suffisante ! Cela donne donc l'algorithme suivant :

Algorithme Galligo/Rupprecht
ENTRÉE : $P(X,Y) \in \mathbb{Q}[X,Y]$ un polynôme irréductible dans $\mathbb{Q}[X,Y]$ de degré n et sans facteurs linéaires.
SORTIE : $P_1(X,Y) \in \mathbb{C}[X,Y]$ un facteur absolument irréductible de P.

1. Calculer $\varphi_i(X) = y_i(x_0) + a_i(x_0)(X - x_0) + b_i(x_0)(X - x_0)^2 \mod (X - x_0)^3$ pour $1 \leq i \leq n$.
2. Trouver un ensemble I tel que $\sum_{i \in I} b_i(x_0) = 0$ et $\sum_{J \subsetneq I} b_i(x_0) \neq 0$.
3. Calculer $P_1(X,Y) = \prod_{i \in I} \varphi_i(X) \mod (X - x_0)^3$.
4. Retourner le facteur obtenu par une remontée de Hensel.

La partie la plus délicate de cet algorithme est la partie 2. En effet une méthode naïve consisterait à calculer toutes les sommes possibles des $b_i(x_0)$ pour $i \in \{1, \ldots, n\}$. Cela donnerait au moins 2^n sommes à calculer. Une telle démarche ne permet pas d'obtenir une factorisation absolue dans le cas où $n = 200$. Un petit calcul amusant nous explique pourquoi.
Nous avons $2^{200} = (2^{10})^{20} \geq 10^{60}$.
Supposons que nous possédions un ordinateur capable de faire 10^{10} opérations en une seconde (c'est à dire un ordinateur "idéal" à 10 Ghz). Dans ce cas il nous faudrait plus de 10^{50} secondes pour effectuer le calcul de toutes les sommes. Or une année correspond à environ 3.10^7 secondes. Donc 10^{50} secondes correspondent à environ 3.10^{42} années. Pour donner un ordre de grandeur on estime l'âge de l'univers à environ 15.10^9 années...

A l'aide d'une astuce David Rupprecht a réduit ce nombre de calculs à environ $2^{n/4}$, voir [Rup00], [Rup04]. Nous verrons dans le chapitre 3 comment obtenir la factorisation absolue d'un polynôme de degré 200.

REMARQUE :
Les séries $\varphi_i(X)$ sont calculées jusqu'à l'ordre 2 de manière numérique. Les coefficients du facteur $P_1 \mod (x - x_0)^3$ sont donc connus de manière approchée. Après une remontée de Hensel nous obtenons donc un facteur approché. D. Rupprecht a proposé dans [Rup00] une méthode pour obtenir la factorisation exacte à partir de cette factorisation approchée. Cette méthode repose sur une utilisation des fractions continues. Nous en verrons une amélioration (sans fraction continue) dans le chapitre 2.

Propriétés des nombres b_i

Ici nous allons donner des formules permettant de calculer les nombres $b_i(x_0)$. Ensuite nous démontrerons certains résultats afin d'obtenir l'équivalence entre les sommes de $b_i(x_0)$ nulles et les facteurs de P. Nous détaillons ici certaines preuves afin de pouvoir aller plus vite dans le chapitre 3 où nous aurons un résultat plus général.

On pose :

$$\alpha(x,y) = \frac{\partial P}{\partial x}(x,y), \;\; \beta(x,y) = \frac{\partial P}{\partial y}(x,y),$$

$$\gamma(x,y) = \frac{\partial^2 P}{\partial x^2}(x,y), \;\; \delta(x,y) = \frac{\partial^2 P}{\partial y^2}(x,y), \;\; \varepsilon(x,y) = \frac{\partial^2 P}{\partial x \partial y}(x,y),$$

On a alors :

$$a_i = -\frac{\alpha(x_0, y_i(x_0))}{\beta(x_0, y_i(x_0))}, \text{ et}$$

$$b_i = -\frac{1}{2\beta(x_0, y_i(x_0))}\left(\gamma(x_0, y_i(x_0)) + 2\varepsilon(x_0, y_i(x_0))a_i + \delta(x_0, y_i(x_0))a_i^2\right).$$

Lemme 5. *Soit $P(X,Y)$ un polynôme vérifiant (H). Alors pour presque tous les $x_0 \in \mathbb{C}$, les nombres $b_i(x_0)$ sont tous non nuls.*

Démonstration. On pose : $a(X,Y) = -\dfrac{\alpha(X,Y)}{\beta(X,Y)}$, et

$$b(X,Y) = -\frac{1}{2\beta(X,Y)}\left(\gamma(X,Y) + 2\varepsilon(X,Y)a(X,Y) + \delta(X,Y)a(X,Y)\right) \in \mathbb{C}(X,Y).$$

Dans cette formule $a(X,Y)$ et $b(X,Y)$ sont des éléments de $\mathbb{C}(X,Y)$. On considère alors $\mathcal{P} = \prod_{i=1}^n b(X, \varphi(X)) \in \mathbb{C}(X, \varphi_1(X), \dots, \varphi_n(X))$. $\mathcal{P}$ est une fonction rationnelle symétrique en les $\varphi_i(X)$. Donc nous pouvons exprimer $\mathcal{P}$ comme une fonction rationnelle en les coefficients de P. D'où $\mathcal{P} \in \mathbb{C}(X)$.
Si $\mathcal{P} \neq 0$ dans $\mathbb{C}(X)$ alors pour presque tous les x_0 de $\mathbb{C}$, nous avons $\mathcal{P}(x_0) \neq 0$ donc pour $1 \leq i \leq n$, il vient : $b(x_0, \varphi_i(x_0)) = b_i(x_0) \neq 0$.
Nous allons supposer $\mathcal{P} = 0$ dans $\mathbb{C}(X)$, et aboutir à une contradiction. Cela montrera alors le lemme.
A présent nous regardons les $\varphi_i(X)$ comme des fonctions analytiques sur un voisinage U de x_0. (Cela est possible d'après la version analytique du théorème des fonctions implicites). Les $b(x, \varphi_i(x))$ sont donc des fonctions analytiques sur U.
Comme $\mathcal{P} = 0$ dans $\mathbb{C}(X)$, on obtient $\mathcal{P}(x) = \prod_{i=1}^n b(x, \varphi_i(x)) = 0$ sur U. L'anneau des fonctions analytiques étant intègre, il existe donc un indice i_0 tel que $b(x, \varphi_{i_0}(x)) = 0$ sur U. Cela signifie $\varphi_{i_0}''(x) = 0$ sur U donc $\varphi_{i_0}^{(r)}(x) = 0$ sur U pour $r \geq 2$.
Donc $\varphi_{i_0}(x) = y_{i_0} + \varphi_{i_0}'(x_0)(x - x_0)$ sur U.
A présent nous effectuons la division euclidienne de $P(X,Y)$ par

$F(X,Y) = Y - y_{i_0} - \varphi'_{i_0}(x_0)(X - x_0)$ dans $\mathbb{C}(X)[Y]$.
On obtient : $P(X,Y) = A(X,Y)F(X,Y) + R(X)$ avec $A(X,Y) \in \mathbb{C}[X,Y]$ et $R(X) \in \mathbb{C}[X]$.
Comme $F(x, \varphi_{i_0}(x)) = 0$ pour tout $x \in U$, il vient $R(X) = 0$ dans $\mathbb{C}[X]$. Cela implique donc que P a un facteur linéaire ce qui est impossible. □

La clef de l'algorithme

Ici nous allons démontrer le théorème suivant :

Théorème 19 (Galligo/Rupprecht). *Soit P un polynôme vérifiant (H) et absolument irréductible. Soit $f_\lambda(X,Y,\lambda) = P(X + \lambda Y, Y)$. Alors pour presque toutes les spécialisations (x_0, λ_0) de (x, λ) dans $\mathbb{C} \times \mathbb{Q}$ aucune des sommes $\sum_{i \in J} b_i$, où $J \subsetneq \{1, \ldots, n\}$, ne s'annule.*

De ce théorème on peut en déduire la version réductible.

Théorème 20. *Soit P un polynôme vérifiant (H). Soit $f_\lambda(X,Y,\lambda) = P(X+\lambda Y, Y)$. Alors pour presque toutes les spécialisations (x_0, λ_0) de (x, λ) dans $\mathbb{C} \times \mathbb{Q}$ les sommes $\sum_{i \in J} b_i$, où $J \subsetneq \{1, \ldots, n\}$, s'annulent si et seulement si $\prod_{i \in J}(Y - \varphi_i(X))$ est un polynôme qui divise P.*

Pour démontrer ce théorème nous allons utiliser le théorème de Harris affine, ainsi que les lemmes suivants (une preuve se trouve dans l'annexe A).

Lemme 6. *Soit γ un lacet dans $\mathbb{C} \setminus \{p_1, \ldots, p_d\}$ alors γ est homotope à un lacet analytique.*

Lemme 7. *Le relèvement $(\tilde{\gamma})$ d'un lacet analytique γ est analytique.*

Pour les définitions d'homotopie et de relèvement voir l'annexe A.

Démonstration. Théorème Galligo/Rupprecht.
Nous conservons les notations introduites dans la preuve du lemme 5.
Nous choisissons λ_0 et nous effectuons un changement de coordonnées tel que le théorème de Harris affine s'applique. A présent les nombres $y_i(x_0)$, $b_i(x_0)$ et $\varphi_i(x_0)$ correspondent au polynôme $f_{\lambda_0} = P(X + \lambda_0 Y, Y)$.
On pose $r < n$.
Nous voulons montrer que pour presque tous les x_0 nous avons $\sum_{i \in I} b_i(x_0) \neq 0$ où $I \subsetneq \{1, \ldots, n\}$ et $|I| = r$.
Nous considérons les fonctions $\mathcal{B}_I = \sum_{i \in I} b(X, \varphi_i(X))$ et $\mathcal{B} = \prod_{\mathcal{E}_r} \mathcal{B}_I$ où $\mathcal{E}_r = \{\{\sigma(1), \ldots, \sigma(r)\} \mid \sigma \in \mathfrak{S}_n\}$.
$\mathcal{B}$ est une fonction rationnelle en $X, \varphi_1(X), \ldots, \varphi_n(X)$ et symétrique par rapport à ses n derniers arguments. Donc comme dans le lemme 5, on en déduit : $\mathcal{B} \in \mathbb{C}(X)$.
Nous voulons montrer : $\mathcal{B} \neq 0$ dans $\mathbb{C}(X)$. Pour cela nous allons supposer $\mathcal{B} = 0$

dans $\mathbb{C}(X)$, et nous allons aboutir à une contradiction.
Tout d'abord nous prenons

$$x_0 \notin \Delta = Disc_Y(f_{\lambda_0}(X,Y)) \text{ tel que pour } 1 \leq i \leq n,\ b_i(x_0) \neq 0\ (\star).$$

Cela est possible d'après le lemme 5. Donc pour tout x dans un voisinage U de x_0 où les fonctions analytiques φ_i sont définies nous avons $\beta(x,\varphi_i(x)) = \frac{\partial P}{\partial Y}(x,\varphi_i(x)) \neq 0$, où $1 \leq i \leq n$.
De ce fait, $b(x,\varphi_i(x))$ est bien définie et analytique sur U. D'où $\mathcal{B}_I(x)$ est analytique.
Comme $\mathcal{B} = 0$, il existe un ensemble I_0 tel que $\mathcal{B}_{I_0}(x) = 0$ sur U.
Nous pouvons supposer $I_0 = \{1,\ldots,r\}$ sans perte de généralité, et nous posons $J_0 = \{2,3,,\ldots,r-1,r,r+1\}$.
D'après le théorème d'Harris affine il existe un lacet γ tel que $[\gamma]$ agit sur $\{y_1(x_0),\ldots,y_n(x_0)\}$ comme la transposition $(1 \quad r+1)$. Cela signifie : Soit $\tilde{\gamma_{y_i}} = (\gamma, \delta_{y_i})$ le relevé de γ au dessus de y_i, où γ et δ_{y_i} sont analytiques (voir lemme 42), nous avons alors :

$$\begin{cases} \delta_{y_i}(0) = y_i \text{ pour } i = 1,\ldots,n \\ \delta_{y_i}(1) = y_i \text{ si } i \neq 1 \text{ et } i \neq r+1 \\ \delta_{y_1}(1) = y_{r+1} \text{ et } \delta_{y_{r+1}}(1) = y_1. \end{cases}$$

A présent on pose : $H(t) = \sum_{i=1}^{r} b(\gamma(t),\delta_{y_i}(t))$, c'est une fonction analytique sur $]0,1[$, telle que $H(0) = \mathcal{B}_{I_0}(x_0)$ et $H(1) = \mathcal{B}_{J_0}(x_0)$. Comme $H(t) = \mathcal{B}_{I_0}(\gamma(t))$ pour $t \in \gamma^{-1}(U)$ et $\mathcal{B}_{I_0} = 0$ sur U, on obtient $H = 0$ car H est analytique. D'où $\mathcal{B}_{I_0}(x_0) = \mathcal{B}_{J_0}(x_0)$. Cela implique $b_1(x_0) = b_{r+1}(x_0)$. De même on obtient $b_1(x_0) = \ldots = b_n(x_0)$.
Finalement comme $\sum_{i=1}^{n} b_i(x_0) = 0$, on en déduit pour $1 \leq i \leq n$, $b_i(x_0) = 0$ ce qui contredit $(\star)$. □

Avec la même technique de preuve que ci-dessus (il suffit de considérer le produit $\mathcal{P} = \prod_{i\neq j}(b_i - b_j))$ nous pouvons aussi montrer le résultat suivant :

Lemme 8. *Soit P un polynôme vérifiant (H). Soit $f_\lambda(X,Y,\lambda) = P(X+\lambda Y, Y)$. Alors pour presque toutes les spécialisations (x_0,λ_0) de (x,λ) dans $\mathbb{C}\times\mathbb{Q}$ les nombres b_i sont distincts deux à deux.*

Cette propriété est utilisée dans l'algorithme Galligo/Rupprecht et sera aussi utilisée dans le chapitre 3.

1.5.4 L'algorithme SVW

L'algorithme proposé et développé par A. Sommese, J. Verschelde et C. Wampler dans [SVW01a], [SVW01c], [SVW02c], [SVW03b], est lui aussi basé sur le théorème d' Harris affine et sur la propriété "$\sum b_i = 0$". Cependant dans cette méthode nous devons remonter et suivre des lacets de manière effective.
Voici le schéma de cet algorithme : Nous rappelons que

$\Delta = \{x \in \mathbb{C} \,|\, Disc_Y(P(X,Y))(x) = 0\}$. Pour faciliter la présentation de cette méthode nous supposons $0 \notin \Delta$.

Algorithme SVW
Entrée : $P(X,Y) \in \mathbb{C}[X,Y]$, $0 \notin \Delta$.
Sortie : Une approximation $\tilde{P_1}(X,Y)$ d'un facteur absolument irréductible de $P(X,Y)$.

1. Calculer les racines y_i du polynôme $P(0,Y)$.
2. Construire un lacet γ dans $\mathbb{C} - \Delta$ partant de 0, et ses n relevés $\tilde{\gamma_{y_i}}$ à partir des y_i.
3. En déduire une partition de la fibre de $P(0,Y)$ au dessus de 0.
4. Vérifier que cette partition correspond à un facteur à l'aide de la propriété "$\sum b_i = 0$".
5. Retourner un facteur approché par interpolation.

Dans l'étape 3 nous regroupons les y_i appartenant à la même composante connexe par arcs. En effet les $\tilde{\gamma_{y_i}}$ sont des chemins reliant 2 points dans la fibre. Ensuite afin de s'assurer que la partition obtenue correspond bien à un facteur, dans l'étape 4 nous vérifions cette partition à l'aide de la propriété sur les b_i. En pratique cette vérification s'effectue ainsi :
A l'aide des lacets $\tilde{\gamma_{y_i}}$ nous avons une partition pour la fibre de P au dessus de différents points a, b, $c \in \mathbb{C} - \Delta$, trois points distincts tels que : $P(a,\alpha_i) = 0$, $P(b,\beta_i) = 0$, $P(c,\gamma_i) = 0$. Soit $I \subset \{1,\dots,n\}$ l'ensemble d'indice correspondant à la partition obtenue. Nous calculons alors A, $B \in \mathbb{C}$ tels que :
$$\left\{ \begin{array}{l} Aa + B = \sum_{i\in I} \alpha_i, \\ Ab + B = \sum_{i\in I} \beta_i. \end{array} \right.$$

Ensuite nous vérifions que : $Ac + B = \sum_{i\in I} \gamma_i$.
Cet algorithme provient de l'étude des systèmes polynomiaux $f : \mathbb{C}^n \to \mathbb{C}^m$ effectuée par A. Sommese, J. Verschelde et C. Wampler. Ici nous sommes dans le cas particulier où $m = 1$.

1.6 Des critères d'irréductibilité absolue

Nous avons vu en 1.2.1 que presque tous les polynômes étaient absolument irréductibles. Ainsi lorsque nous fabriquons un algorithme de factorisation absolu nous pouvons utiliser cette information. C'est à dire, nous pouvons tester à l'étape 1 de l'algorithme si celui-ci est absolument irréductible ou pas. Les tests d'irréductibilité étant en général plus rapides que les méthodes de factorisation, cela permet d'optimiser un algorithme de factorisation.

Dans cette section nous allons rappeler certaines méthodes classiques. Puis nous présenterons l'algorithme de J.F. Ragot qui utilise des calculs modulaires. Pour finir nous donnerons le critère de S. Gao qui repose sur l'étude du polytope de Newton.

1.6.1 Des critères classiques

Dans cette partie nous allons donner sans démonstration quelques tests d'irréductibilité. Ces tests seront accompagnés d'exemples.
Tout d'abord nous remarquons que certains tests utilisés dans le cas univarié s'adaptent parfaitement au cas multivarié.

Proposition 8 (Réduction). *Soient A un anneau factoriel et K son corps de fractions. Soient I un idéal premier de A, $B = A/I$ et L le corps de fractions de B. Soit $P(X) = a_nX^n + \ldots + a_0$ un polynôme de $A[X]$ et $\overline{P}$ sa réduction modulo I. Si $\overline{a_n} \neq 0$ et $\overline{P}$ est irréductible sur B ou L, alors le polynôme P est irréductible sur K.*

Exemple : $P(X,Y) = Y^2 + (48X^7 + 20X^2 - 9)Y + 123X^2 + 3$ est irréductible dans $\mathbb{Q}[X,Y]$. Prendre $A = \mathbb{Z}[X]$ et $I = (X)$, puis $A = \mathbb{Z}$ et $I = (2)$.

Proposition 9 (Eisenstein). *Soient A un anneau factoriel, p un irréductible de A, et $P(X) = a_nX^n + \ldots + a_0 \in A[X]$. Si $a_n \neq 0 \mod p$, $a_i = 0 \mod p$ pour $0 \leq i \leq n-1$, et $a_0 \neq 0 \mod p^2$ alors $P(X)$ est irréductible dans $A[X]$.*

Exemple : $P(X,Y) = Y^5 + 2XY^4 + (X^2+3X)Y^3 + (5X^3 - 9X^2)Y + X \in \mathbb{C}[X,Y]$ est irréductible car il vérifie le test pour $A = \mathbb{C}[X]$ et $p = X$.

De nombreuses généralisation du critère d'Eisenstein ont été données en utilisant les polygones de Newton. Pour un aperçu de ces résultats nous pouvons consulter [Gao01].
A présent donnons un énoncé conçu spécialement pour le cas des polynômes bivariés.

Proposition 10. *Soient A un anneau intègre, $P_1(X) \in A[X]$, $P_2(Y) \in A[Y]$. Si pgcd$(\deg(P_1), \deg(P_2)) = 1$ alors $P(X,Y) = P_1(X) + P_2(Y)$ est irréductible dans $A[X,Y]$.*

Pour une preuve nous pouvons consulter [MŞ99] page 64.

Exemple : $P(X,Y) = Y^5 + 3Y^4 + 10 + 9X^4 + 11X - 2$ est irréductible dans $\mathbb{C}[X,Y]$.

1.6.2 L'algorithme de Ragot

Dans cette partie $P(X,Y)$ désignera un polynôme de $\mathbb{Z}[X,Y]$ et p un nombre premier. Nous avons vu en 1.2.2 que si un polynôme $P(X,Y) \in \mathbb{Z}[X,Y]$ est absolument irréductible modulo p alors il est absolument irréductible. D'autre part si un polynôme irréductible a une solution rationnelle simple alors ce polynôme est absolument irréductible d'après le théorème 12. De ces deux résultats nous pouvons en déduire un test d'irréductibilité absolu.

Définition 18. *Nous dirons que P vérifie le critère Ragot modulo p si $P \mod p$ est irréductible, admet une solution simple dans $\mathbb{F}_p^2$, et, si* $\deg(P \mod p) = \deg(P)$.

Nous avons donc le critère :

Proposition 11. *Si P vérifie le critère Ragot modulo p alors P est absolument irréductible.*

Nous pouvons à partir de cette proposition obtenir un algorithme déterministe. En effet nous avons le résultat suivant (voir [Rag97]) :

Proposition 12. *Soit $P(X,Y) \in \mathbb{F}_q[X,Y]$ un polynôme absolument irréductible de degré d. Si $q \geq (d-1)^4$ alors P a des solutions $\mathbb{F}_q$ rationnelles simples.*

Ainsi de cette proposition et du théorème d'Ostrowski effectif nous pouvons obtenir un test déterministe. L'inconvénient de cette démarche est qu'il faut prendre un nombre premier très grand alors que d'autres plus petits auraient pu convenir. J.F. Ragot propose alors l'algorithme suivant :

Algorithme Ragot
ENTRÉE : $P(X,Y) \in \mathbb{Z}[X,Y]$.
SORTIE : " P est absolument irréductible" ou " ?".
Pour p allant de 2 à 101 (par exemple) faire :
Si $P \mod p$ vérifie le critère Ragot modulo p alors retourner "P est absolument irréductible"
Fin boucle
Retourner " ?".

Dans sa thèse J.F. Ragot [Rag97] fait une étude détaillée de ce test probabiliste.

1.6.3 Les critères de Gao

Dans cette partie nous allons voir que dans certains cas nous pouvons affirmer qu'un polynôme est absolument irréductible en ne connaissant que son support. Pour cela nous devons définir le polytope de Newton associé à un polynôme.

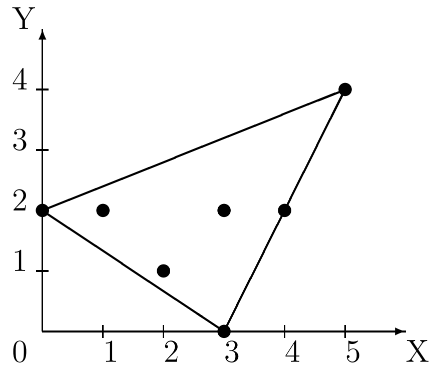
Définition 19. *Soit $P(X,Y) = \sum_{i,j} c_{i,j} X^i Y^j \in \mathbb{K}[X,Y]$. Le polytope de Newton $Newt(P)$ de P est l'enveloppe convexe des points $(i,j) \in \mathbb{R}^2$ tels que $c_{i,j} \neq 0$.*

Après une étude géométrique basée sur les sommes de Minkowski de 2 convexes S. Gao donne dans [Gao01] les tests suivants :

Proposition 13. *Soit* $P(X,Y) = aX^n + bY^m + cX^uY^v + \sum_{i,j} c_{i,j}X^iY^j \in \mathbb{K}[X,Y]$ *avec* a*,* b*, et* c *non nuls. Si* $Newt(P)$ *est le triangle de sommets* $(n,0)$*,* $(0,m)$ *et* (u,v)*, et* $pgcd(m,n,u,v) = 1$ *alors* P *est absolument irréductible.*

Proposition 14. *Soit* $P(X,Y) = aX^n + bY^m + \sum_{i,j} X^iY^j \in \mathbb{K}[X,Y]$ *avec* a*,* b *non nuls et* (i,j) *différents de* $(n,0)$ *et* $(0,m)$*. Si* $Newt(P)$ *est contenu dans un triangle de sommets* $(m,0)$*,* $(0,n)$ *et* $(u,v) \in \mathbb{R}^2$*, et,* $pgcd(m,n) = 1$ *alors* P *est absolument irréductible.*

Exemple : $P(X,Y) = X^3+Y^2+X^5Y^4+X^4Y^2+X^3Y^2+X^2Y+XY^2 \in \mathbb{Q}[X,Y]$ et $Q(X,Y) = 2X^3+7Y^2+X^5Y^4+10X^4Y^2+X^3Y^2+X^2Y+3XY^2 \in \mathbb{Q}[X,Y]$ sont absolument irréductibles. Ici $n=3$, $m=2$, $u=5$, $v=4$. Le polytope de Newton de ces deux polynômes est représenté ci-dessous :



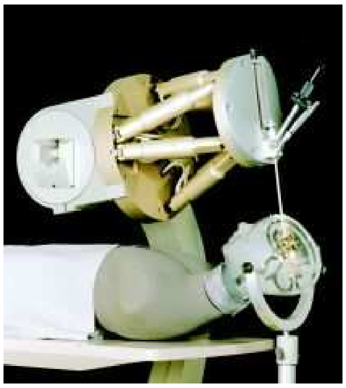
Au chapitre 5 nous donnerons un test très ressemblant où l'on enlèvera l'hypothèse sur le triangle mais où l'on ajoutera l'hypothèse P est irréductible dans $\mathbb{K}[X,Y]$.

1.7 Applications

Pour finir ce chapitre d'introduction à la factorisation absolue nous allons donner un exemple d'application : les plates-formes de Stewart.
Une plate-forme de Stewart (inventée en réalité par V.E. Gough en 1947 et redécouverte en 1965 par S. Stewart) est un mécanisme constitué d'une plate-forme mobile à laquelle sont attachés 6 bras en des points P_1 à P_6, les autres extrémités des bras sont attachées en des points immobiles Q_1 à Q_6. Les bras peuvent pivoter librement sur leur point d'ancrage. On déplace la plate-forme en faisant varier par un mécanisme approprié la longueur des bras. Les points peuvent être groupés par paires sur la plate-forme ou sur la base. La plate-forme peut être remplacée par un solide quelconque et les points d'ancrage ne sont pas nécessairement dans un même plan. En voici quelques exemples :

Une plate-forme de Stewart “classique” :

Une utilisation de la plate-forme de Stewart en chirurgie :



Une utilisation de la plate-forme de Stewart comme perceuse :

Une utilisation de la plate-forme de Stewart comme simulateur de vol :

Dans [CLO97]et [SVW03b] nous voyons comment décrire la position et l'orientation d'un robot en fonction des paramètres : longueur des bras et angles de ceux-ci.

Nous avons alors une application polynomiale :

$$\begin{array}{rcl} f : \mathbf{P} := \mathbb{C}^m & \longrightarrow & \mathbb{C}^n =: \mathbf{T} \\ x_1, \ldots, x_n & \longmapsto & f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m) \end{array}$$

où $\mathbf{P}$ est l'espace des paramètres (longueur des bras et angles) et $\mathbf{T}$ est l'espace de travail (position et orientation du robot). Le calcul du jacobien J_f de f nous permet alors de déterminer certains déplacements impossibles pour ce robot ou certains lieux à éviter. Plus précisément :

Définition 20. *Une singularité pour un robot est un point $p \in \mathbf{P}$ tel que le rang de $J_f(p)$ soit strictement inférieur à* $\min(m, n)$.

A présent voyons pourquoi ces points sont "singuliers" :
Supposons que nous voulions obtenir un certain déplacement "lisse" du robot par rapport au temps : $c(t)$, où t est le temps et $c(t) \in \mathbf{T}$ désigne la position du robot à l'instant t. Pour obtenir un tel déplacement il faut avoir une courbe paramétrée $p(t) \in \mathbf{P}$ telle que : $c(t) = f(p(t))$.
Nous dérivons cette égalité et nous obtenons : $c'(t) = J_f(p(t)).p'(t)$.
A présent si $p(t_0)$ est une singularité alors $J_f(p(t_0))$ n'est pas surjective donc il est possible qu'aucun vecteur v de $\mathbb{C}^m$ ne donne l'égalité : $c'(t) = J_f(p(t)).v$. Ainsi le déplacement $c(t)$ peut ne pas être obtenu avec ce robot pour le paramètre $p(t_0)$. Cela est un premier type d'ennui que l'on peut obtenir lorsque nous rencontrons un point singulier.

Une autre situation est délicate :
En conservant les notations précédentes supposons que $p(t_0)$ se trouve "au voisinage" d'un point singulier, et pour simplifier supposons aussi que $m = n$.
Au voisinage de $p(t_0)$ nous avons : $f(p(t_0) + \Delta_p) = f(p(t_0)) + J_f(p(t_0)).\Delta_p + O(\Delta_p^2)$.

D'où : $c(t_0) + \Delta_c = f(p(t_0) + \Delta_p) = f(p(t_0)) + J_f(p(t_0)).\Delta_p + O(\Delta_p^2)$.
Il vient : $\Delta_c \approx J_f(p(t_0)).\Delta_p$.

Nous constatons alors que pour un petit déplacement Δ_c dans l'espace de travail **T**, il se peut que nous ayons besoin d'avoir un grand déplacement dans l'espace des paramètres **P**. En effet si nous exprimons Δ_p à l'aide des formules de Cramer alors nous avons au dénominateur de chaque coordonnées de Δ_p le déterminant $det(J_f(p(t_0))$. Or nous avons supposé $p(t_0)$ proche d'un point singulier donc par continuité $det(J_f(p(t_0))$ est très petit, donc Δ_p peut être très grand.
Ainsi lorsque nous sommes très proches d'un point singulier, pour obtenir un petit déplacement du robot, il se peut que nous devions effectuer de grandes modifications dans les paramètres.
De même la relation $c'(t_0) = J_f(p(t_0)).p'(t_0)$ montre que même si nous voulons déplacer lentement le robot au voisinage d'un point singulier il faudra changer rapidement les paramètres (car $\|p'(t_0)\|$ est grand). Tout cela entraînerait une usure du robot si une telle tâche était demandée de manière répétitive.

La bonne connaissance du lieu singulier est donc très importante en robotique. De plus la factorisation du déterminant $det(J_f(x_1, \ldots, x_m)) \in \mathbb{C}[x_1, \ldots, x_m]$ donne un moyen d'étudier séparément chaque composante connexe du lieu singulier. En effet les composantes connexes correspondent aux zéros des facteurs de $det(J_f(x_1, \ldots, x_m))$ (voir annexe A). L'étude séparée de chaque composante connexe permet donc de diminuer la complexité du problème.

Une étude de la plate-forme de Stewart, à partir d'un algorithme de factorisation absolu a été mené par Sommese Verschelde et Wampler. Leurs résultats exposés dans [SVW03b] montrent qu'ils ont retrouvé avec leur méthode des résultats récents (2002) étudiés en robotique (voir [SVW03b] et [MSOG00] , [PH02]).

Première partie

Méthodes symboliques-numériques

Chapitre 2

D'une factorisation approchée à une factorisation exacte

Ce chapitre est une version étendue de l'article [CG03].

La factorisation approchée de polynômes en plusieurs variables est le sujet de nombreuses études (voir [CGKW02],[Sas01],[SVW03b],[GR02],[Nag02]). Ici nous allons voir le "lien"qui existe entre de telles factorisations et la factorisation exacte.

Dans ce chapitre nous allons étudier P un polynôme de $\mathbb{Q}[X,Y]$ unitaire en Y et irréductible dans $\mathbb{Q}[X,Y]$ ainsi que sa factorisation $P_1 \dots P_s$ dans $\mathbb{C}[X,Y]$. Le corps $K = \mathbb{Q}[\alpha]$ désignera la plus petite extension de $\mathbb{Q}$ dans laquelle se trouvent tous les coefficients de P_1.

Le problème étudié est le suivant :
Supposons que nous ayons un algorithme de factorisation approchée. Dans ce cas nous pouvons obtenir les polynômes $\tilde{P}_i$ tels que $\|P_i - \tilde{P}_i\|_\infty < \epsilon < 1$ où $\|P\|_\infty$ désigne le maximum des valeurs absolues de tous les coefficients de P.
A l'aide de cet algorithme nous voulons obtenir une factorisation exacte. Cela signifie que nous voulons trouver l'extension $K = \mathbb{Q}[\alpha]$ (c'est-à-dire donner le polynôme minimal de α), et, les coefficients des monômes X^iY^j de P_1 sous la forme $q_0 + q_1\alpha + \dots + q_{s-1}\alpha^{s-1}$ où $q_i \in \mathbb{Q}$.

Les coefficients que nous avons à notre disposition sont connus à ϵ prés, donc pour obtenir le polynôme minimal de α il nous faut "reconnaître" les coefficients exacts de celui-ci. Or a priori les coefficients du polynôme minimal de α sont dans $\mathbb{Q}$, il est donc difficile de retrouver les coefficients exacts de ce polynôme. Dans ce chapitre nous verrons qu'en nous ramenant au cas $P(X,Y) \in \mathbb{Z}[X,Y]$, nous n'aurons qu'à "reconnaître" des nombres entiers. Ce travail de reconnaissance correspondra à prendre l'entier le plus proche d'un nombre réel. Nous pourrons alors obtenir un algorithme de factorisation absolue exacte à partir d'un algorithme de factorisation absolue approchée.

Exemple : Pour le polynôme

$$P(X,Y) = Y^4 + 2Y^2X + 14Y^2 - 7X^2 + 6X + 47 \in \mathbb{Q}[X,Y]$$

un algorithme de factorisation approchée donne :

$$\tilde{P}_1(X,Y) = Y^2 + 3.828X + 8.414\,,\ \tilde{P}_2(X,Y) = Y^2 - 1.828X + 5.585.$$

Nous allons présenter un algorithme qui nous permettra d'en déduire l'expression exacte d'un facteur :

$P_1(X,Y) = Y^2 + (2\alpha+1)X + \alpha + 7$ où α est racine du polynôme $f_\alpha(T) = T^2 - 2$.

2.1 Des résultats théoriques

Dans cette section nous allons montrer que nous pouvons nous ramener à une situation où les P_i ont tous leurs coefficients entiers sur $\mathbb{Z}$; cela nous permettra d'obtenir un élément primitif α entier sur $\mathbb{Z}$. L'intérêt de cette démarche est qu'un élément entier sur $\mathbb{Z}$ a un polynôme minimal à coefficients dans $\mathbb{Z}$, donc pour obtenir le polynôme minimal de α il nous faut reconnaître un nombre entier à partir d'une approximation de celui-ci. Dans ce cas le travail de reconnaissance des coefficients exacts est donc beaucoup plus facile.

2.1.1 Restriction effective au cas $\mathbb{Z}[X,Y]$

Dans ce paragraphe nous allons voir comment nous pouvons nous ramener au cas d'un polynôme $P(X,Y) \in \mathbb{Z}[X,Y]$ irréductible et unitaire en Y.

Le polynôme Q est supposé vérifier les hypothèses du lemme fondamental ; il est connu, donc nous connaissons le dénominateur commun des coefficients de Q. On note ce dénominateur d et on pose :

$$P(X,Y) = d^n Q\left(X, \frac{Y}{d}\right).$$

Alors $P(X,Y)$ est unitaire en Y, irréductible dans $\mathbb{Q}[X,Y]$ et appartient à $\mathbb{Z}[X,Y]$.

Cela signifie donc que l'on peut ramener le problème à l'étude de la factorisation de $P(X,Y) \in \mathbb{Z}[X,Y]$ et non plus de $Q \in \mathbb{Q}[X,Y]$. Le lemme suivant nous dit comment faire :

Lemme 9. *Soit $Q(X,Y)$ un polynôme de $\mathbb{Q}[X,Y]$ unitaire et irréductible dans $\mathbb{Q}[X,Y]$. Soient d le dénominateur commun des coefficients de Q et*

$Q(X,Y) = Q_1(X,Y)...Q_s(X,Y)$ sa factorisation en polynômes irréductibles dans $\mathbb{C}[X,Y]$. Nous avons alors :

$$P(X,Y) = d^n Q\left(X, \frac{Y}{d}\right) = d^m Q_1\left(X, \frac{Y}{d}\right) ...d^m Q_s\left(X, \frac{Y}{d}\right).$$

avec $P(X,Y) \in \mathbb{Z}[X,Y]$ qui est irréductible dans $\mathbb{Q}[X,Y]$ et unitaire en Y.
De plus les polynômes $P_i(X,Y) = d^m Q_i\left(X, \frac{Y}{d}\right) \in \mathbb{C}[X,Y]$ sont les facteurs irréductibles de P dans $\mathbb{C}[X,Y]$.

D'autre part nous avons le lemme suivant :

Lemme 10. *En conservant les notations précédentes et en notant K' le plus petit corps engendré par les coefficients de Q_1 et K le plus petit corps engendré par les coefficients de P_1, nous avons : $K' = K$.*

Remarques :
Le lemme 10 nous montre que si on obtient un élément primitif de K avec son polynôme minimal, alors ce sera aussi un élément primitif de K'.
Nous avons $Q(X,Y) = \frac{1}{d^n} P(X, dY)$ et $Q_i(X,Y) = \frac{1}{d^m} P_i(X, dY)$.
Pratiquement nous allons passer de Q à P, appliquer un algorithme de factorisation approchée à P, puis nous en déduirons une factorisation exacte pour P. Nous obtiendrons alors une factorisation exacte de Q grâce aux formules ci-dessus.

2.1.2 Les coefficients des P_i sont des entiers algébriques sur $\mathbb{Z}$

L'objectif de cette section est de montrer le théorème suivant :

Théorème 21. *Soit $P \in \mathbb{Z}[X,Y]$ unitaire en Y irréductible sur $\mathbb{Q}[X,Y]$. Alors sa factorisation en irréductibles dans $\mathbb{C}[X,Y]$: $P_1 \dots P_s$ est constituée de polynômes P_i à coefficients entiers algébriques sur $\mathbb{Z}$.*

Nous allons tout d'abord démontrer le lemme suivant :

Lemme 11. *Soient α un nombre algébrique sur $\mathbb{Q}$, et, $p(X) \in \mathbb{Q}[\alpha][X]$ un entier sur $\mathbb{Z}[X]$. Dans ce cas tous les coefficients de $p(X)$ sont entiers sur $\mathbb{Z}$.*

Démonstration. On note s le degré de α sur $\mathbb{Q}$ et $l = \deg_X(p)$. Alors $\mathbb{Q}(X)[\alpha]$ est une extension de $\mathbb{Q}(X)$ de degré s. De plus :
($\star$) Tous les conjugués de $p(X)$ appartiennent à $\mathbb{C}[X]$ et ont le même degré l.
Comme $\mathbb{Z}[X]$ est un anneau intégralement clos, il vient (voir [Sam67] page 45) :
($\star\star$) Les coefficients du polynôme caractéristique $P_{char}(p(X))$ de $p(X)$ sur $\mathbb{Q}(X)$ sont dans $\mathbb{Z}[X]$.

Soit $k = [\mathbb{Q}(X)[\alpha] : \mathbb{Q}(X)[p(X)]]$, nous notons q_i les conjugués de $p(X)$ sur $\mathbb{Q}(X)$ et $q_1 = p(X)$. Nous obtenons alors $P_{char}(p(X))(Z) = \prod_{i=1}^{s/k}(Z - q_i)^k$.
A présent nous allons montrer que tous les coefficients de $p(X)$ sont entiers sur $\mathbb{Z}$. Nous allons commencer par $\mathrm{lc}(p(X))$ le terme de tête de $p(X)$. Nous avons :

$$P_{char}(p(X))(Z) = Z^s + (\sum_i q_i)Z^{s-1} + ... + \prod_i q_i = Z^s + c_{s-1}(X)Z^{s-1} + ... + c_0(X)$$

avec $c_i(X) \in \mathbb{Z}[X]$ d'après $(\star\star)$, et $\deg(c_{s-i}(X)) \leq il$, d'après $(\star)$. Donc $\deg(c_{s-i}(X)p(X)^{s-i}) \leq ls$. Comme $P_{char}(p(X))(p(X)) = 0$ dans $\mathbb{C}[X]$, le terme de degré ls donne :

$$\lambda_l^s + \sum_{i \in I} \mathrm{lc}(c_{s-i})\lambda_l^{s-i} = 0$$

où $\lambda_l = \mathrm{lc}(p(X))$ et I est l'ensemble $I = \{i / \deg(c_{s-i}(X)p(X)^{s-i}) = ls\}$.
Du fait que tous les coefficients $\mathrm{lc}(c_i)$ sont entiers, il vient λ_l est un entier algébrique sur $\mathbb{Z}$.
Pour finir nous remarquons que $\lambda_l X^l$ est un entier algébrique sur $\mathbb{Z}[X]$, et donc $p(X) - \lambda_l X^l \in \mathbb{Q}[\alpha][X]$ est entier sur $\mathbb{Z}[X]$. Nous pouvons alors répéter ce qui précède avec $p(X) - \lambda_l X^l$ au lieu de $p(X)$ et cela nous donne le résultat souhaité. □

Démonstration. Théorème 21.
D'après le théorème de Steinitz il existe un corps algébriquement clos tel que $\mathcal{K} \supset \mathbb{Q}(X) \supset \mathbb{Z}[X]$. Nous avons alors :

$$P(X,Y) = Y^n + a_{n-1}(X)Y^{n-1} + ... + a_0(X) = \prod_{i=1}^{n}(Y - r_i(X))$$

où $r_i(X) \in \mathcal{K}$ sont des entiers algébriques sur $\mathbb{Z}[X]$. De plus $P_1(X,Y)$ est un facteur de $P(X,Y)$ dans $\mathbb{Q}[\alpha][X,Y]$, nous avons donc :

$$P_1(X,Y) = \prod_{i=1}^{m}(Y - r_i(X)) = Y^m + p_{m-1}(X)Y^{m-1} + ... + p_0(X).$$

Ainsi $p_i(X)$ sont des entiers sur $\mathbb{Z}[X]$ car ce sont des polynômes en $r_i(X)$. Pour conclure il nous suffit donc d'appliquer le lemme précédent à chaque $p_i(X)$. □

Remarque : Dans leur étude du déterminant de Cayley-Menger (voir [DS04]) M. Sombra et C. D'Andrea ont généralisé ce résultat en remplaçant l'hypothèse unitaire, par P de contenu égal à 1.

2.2 Reconnaissance du polynôme minimal d'un élément primitif

Nous avons vu précédemment que tous les coefficients de P_1 engendrent une extension K de $\mathbb{Q}$. Le théorème de l'élément primitif nous donne l'existence d'un élément α tel que $K = \mathbb{Q}[\alpha]$. A présent nous allons voir comment obtenir le polynôme minimal d'un élément primitif de K *entier sur* $\mathbb{Z}$.

Dans un premier temps nous allons montrer comment reconnaître un élément primitif parmi les coefficients de P_1. Ensuite nous donnerons une autre méthode permettant de fabriquer un élément primitif.

La deuxième méthode est là pour venir en aide à la première dans le cas où aucun coefficients de P_1 n'est primitif (ce cas est possible théoriquement mais rare "en pratique"). Le fait de privilégier la reconnaissance à la construction d'un élément primitif est expliqué en 2.5.1.

Pour finir nous étudierons la précision nécessaire à cette étape de l'algorithme.

2.2.1 Reconnaître un élément primitif

Dans cette partie nous allons voir comment obtenir le polynôme caractéristique de chacun des coefficients de P_1. Dans le cas d'un élément primitif cela correspondra à son polynôme minimal. Ensuite nous verrons comment caractériser les polynômes correspondant à un élément primitif .

Notations : Notons σ_i les s $\mathbb{Q}$-homomorphismes de K dans $\mathbb{C}$, et $P_i(X,Y) = \sum_u \sum_v a_i^{(u,v)} X^u Y^v$.

D'après le lemme fondamental on a :

Lemme 12. *Avec les notations précédentes on obtient le polynôme caractéristique de $a_1^{(u,v)}$ ainsi :*

$$P_{char}(a_1^{(u,v)})(T) = \prod_{i=1}^{s} (T - a_i^{(u,v)}).$$

Nous en déduisons la caractérisation suivante :

Lemme 13. *Avec les notations précédentes, nous avons :*

$$pgcd(P_{char}(a_1^{(u,v)}), \frac{\partial}{\partial T} P_{char}(a_1^{(u,v)})) = 1 \iff a_1^{(u,v)} \text{ est un élément primitif de } \mathbb{K}.$$

Dans ce cas $P_{char}(a_1^{(u,v)})$ est le polynôme minimal $f_{a_1^{(u,v)}}$ de $a_1^{(u,v)}$ sur $\mathbb{Q}$.

Nous verrons en 2.2.3 comment obtenir de façon exacte $f_{a_1^{(u,v)}}$. Ainsi le lemme précédent permettra de reconnaître des éléments primitifs de manière certifiée.

Il peut arriver dans de rares cas qu'aucun coefficients de P_1 ne soient primitif. Dans ce cas il faut construire un élément primitif de K. C'est ce que nous allons faire.

2.2.2 Construire un élément primitif

Commençons par remarquer que pour $i \neq j$ il existe toujours un coefficient $a_1^{(u,v)}$ de P_1 tel que $\sigma_i(a_1^{(u,v)}) \neq \sigma_j(a_1^{(u,v)})$. Si ce n'était pas le cas alors il existerait i et j tels que $\sigma_i(a_1^{(u,v)}) = \sigma_j(a_1^{(u,v)})$ pour tous les coefficients $a_1^{(u,v)}$ de P_1, dans ce cas nous aurions $\sigma_i = \sigma_j$ et cela est absurde.

Nous considérons alors le polynôme :

$$\begin{aligned} H(\lambda_{(1,0)},...,\lambda_{(2,n-1)}) &= \prod_{i<j}((\sigma_i - \sigma_j)(a_1^{(0,0)}) + \lambda_{(1,0)}(\sigma_i - \sigma_j)(a_1^{(1,0)}) + \dots \\ &\quad + \lambda_{(1,n-1)}(\sigma_i - \sigma_j)(a_1^{(1,n-1)})) \in \mathbb{C}[\lambda_{(i,j)}] \end{aligned}$$

Nous avons alors $H(\underline{\lambda_{i,j}}) \neq 0$. Nous pouvons donc trouver $(\lambda_{(1,0)}, \dots, \lambda_{(1,n-1)})$ avec $\lambda_{i,j} \in \mathbb{Z}$ tels que :

$$\sigma_i(a_1^{(0,0)} + \lambda_{(1,0)}a_1^{(1,0)} + \dots + \lambda_{(1,n-1)}a_1^{(1,n-1)}) \neq \sigma_j(a_1^{(0,0)} + \lambda_{(1,0)}a_1^{(1,0)} + \dots + \lambda_{(1,n-1)}a_1^{(1,n-1)}).$$

Cela signifie : $a_1^{(0,0)} + \lambda_{(1,0)}a_1^{(1,0)} + \dots + \lambda_{(1,n-1)}a_1^{(1,n-1)}$ est un élément primitif.

En appliquant le lemme de Zippel-Schwartz au polynôme $H(\underline{\lambda_{(i,j)}}) \in \mathbb{C}[\underline{\lambda_{i,j}}]$ on obtient :
$\mathcal{P}(a_1^{(0,0)} + s_{(1,0)}a_1^{(1,0)} + \dots + s_{(1,n-1)}a_1^{(1,n-1)}$ non primitif $\mid s_{i,j} \in S) \leq \dfrac{\binom{s}{2}}{|S|}$.

Nous avons démontré le lemme suivant :

Lemme 14. *Soit P un polynôme de $\mathbb{Z}[X,Y]$ unitaire en Y, et irréductible dans $\mathbb{Q}[X,Y]$. Soient $a_1^{(u,v)}$ les coefficients de P_1 et $\mathbb{Q}[\alpha]$ l'extension de $\mathbb{Q}$ engendrée par tous les coefficients de P_1. Soit S un sous ensemble de $\mathbb{Z}$. Dans cette situation nous avons :*

$$\mathcal{P}\big(a_1^{(0,0)} + s_{(1,0)}a_1^{(1,0)} + \dots + s_{(1,n-1)}a_1^{(1,n-1)} \textit{ non primitif } \mid s_{i,j} \in S\big) \leq \frac{\binom{s}{2}}{|S|}.$$

Ce lemme nous donne donc une démarche pour trouver un élément primitif de $\mathbb{Q}[\alpha]$ entier sur $\mathbb{Z}$, ainsi que son polynôme minimal.

En effet, les $a_i^{(u,v)}$ sont des entiers algébriques sur $\mathbb{Z}$ d'après le théorème 21 et les s_i sont dans $\mathbb{Z}$ donc : $a_1^{(0,0)} + s_{(1,0)}a_1^{(1,0)} + \dots + s_{(2,n-1)}a_1^{(2,n-1)}$ est un entier algébrique sur $\mathbb{Z}$ et la probabilité pour que cet élément ne soit pas primitif est inférieure à $\dfrac{\binom{s}{2}}{|S|}$ (probabilité que l'on peut donc prendre aussi petite que l'on veut). De plus nous pouvons vérifier à l'aide du lemme 13 si cet élément est primitif ou non. Si ce n'est pas le cas nous recommençons un nouveau tirage. Nous avons donc un moyen rapide pour obtenir un élément de $\mathbb{Q}[\alpha]$ primitif et entier sur $\mathbb{Z}$.

Remarque : Nous pouvons obtenir un élément primitif de manière déterministe, mais dans ce cas les coefficients $s_{i,j} \in \mathbb{Z}$ sont plus grands que ceux utilisés dans la méthode probabiliste.

2.2.3 Choix de la précision

Nous avons vu que le polynôme minimal de α est du type : $f_\alpha(T) = \prod_{k=1}^{s}(T-\alpha_k)$ où les α_k sont les conjugués de α. De plus grâce au théorème 21 et aux résultats de la partie précédente nous savons que α est un entier algébrique et donc que son polynôme minimal $f_\alpha(T)$ appartient à $\mathbb{Z}[T]$. Cependant nous n'avons à notre disposition que des valeurs approchées $\alpha_k + \epsilon_k$ des α_k. De ce fait nous n'avons pas à notre disposition le polynôme $f_\alpha(T)$ mais le polynôme $f_{\alpha+\epsilon}(T) = \prod_{k=1}^{s}(T-(\alpha_k+\epsilon_k))$.

Comme nous souhaitons avoir l'expression développée de f_α nous allons regarder ici quelle est l'influence sur les coefficients d'une perturbation sur les racines. Du fait que $f_\alpha(T) \in \mathbb{Z}[T]$ nous pourrons reconnaître f_α à partir des données approchées si la perturbation sur les coefficients est inférieure à 0.5.

Il faut donc étudier l'application φ de $\mathbb{C}^s$ dans $\mathbb{C}^s$ décrite ci dessous :

$$\begin{array}{cccc} \varphi : & \mathbb{C}^s & \longrightarrow & \mathbb{C}^s \\ & \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \vdots \\ \alpha_s \end{pmatrix} & \longmapsto & \begin{pmatrix} S_1(\alpha_1,\dots,\alpha_s) & = & \alpha_1+\alpha_2+\dots\alpha_s \\ & \vdots & \\ S_k(\alpha_1,\dots,\alpha_s) & = & \displaystyle\sum_{1\le i_1<..<i_k\le s} \alpha_{i_1}\dots\alpha_{i_k} \\ & \vdots & \\ S_s(\alpha_1,\dots,\alpha_s) & = & \alpha_1\times\dots\times\alpha_s \end{pmatrix} \end{array}$$

φ est une application polynomiale dont chaque composante est de degré total au plus s. Ainsi dans la formule de Taylor avec reste intégral le reste est nul lorsque l'on considère un développement à l'ordre s.

Avant de passer aux calculs donnons une notation :

$$\left[\sum_{i=1}^{s} \epsilon_i \frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[k]} = \sum_{\substack{i_1+\dots+i_s=k \\ i_j\in\{0,1\}}} \frac{k!}{i_1!\dots i_s!}\epsilon_1^{i_1}\dots\epsilon_s^{i_s}\frac{\partial^k\varphi}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha).$$

Le développement de Taylor s'écrit donc

$$\varphi(\alpha+\epsilon)-\varphi(\alpha) = \left[\sum_{i=1}^{s} \epsilon_i \frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right] + \frac{1}{2!}\left[\sum_{i=1}^{s} \epsilon_i \frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[2]} + \dots + \frac{1}{s!}\left[\sum_{i=1}^{s} \epsilon_i \frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[s]}.$$

Nous allons majorer $\left[\sum_{i=1}^{s} \epsilon_i \frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[k]}$ et nous aurons alors une estimation de la variation des coefficients en fonction de celle des racines.

Pour réaliser cette majoration nous introduisons les constantes ϵ et M ayant les propriétés suivantes :
- $|\alpha_i| \leq M$ pour tout $1 \leq i \leq s$,
- $|\epsilon_i| < \epsilon < 1$.

Nous allons à présent montrer le lemme suivant :

Lemme 15. *Avec les notations établies précédemment nous avons :*

$$\|\varphi(\alpha+\epsilon)-\varphi(\alpha)\|_\infty \leq \left(1+\sum_{k=1}^{s-1}\binom{s}{k}\max\left(1, \max_{j=k+1,\dots,s}(\binom{s-k}{j-k}M^{j-k})\right)\right)\epsilon$$

Démonstration. Comme les polynômes S_j sont de degré total j nous en déduisons :
- Si $k > j$ alors $\dfrac{\partial^k S_j}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha) = 0$.
- Si $k = j$ alors $\dfrac{\partial^k S_j}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha) = 1$.

De plus nous obtenons facilement la majoration suivante pour $k < j$:

$$\left|\frac{\partial^k S_j}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha)\right| \leq \binom{s-k}{j-k}M^{j-k}.$$

De ce fait nous obtenons :

$$\left\|\frac{\partial^k \varphi}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha)\right\|_\infty \leq \max\left(1, \max_{j=k+1,\dots,s}(\binom{s-k}{j-k}M^{j-k})\right)$$

Comme nous avons :

$$\left\|\left[\sum_{i=1}^{s}\epsilon_i\frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[k]}\right\|_\infty \leq \sum_{\substack{i_1+\dots+i_s=k\\ i_j\in\{0,1\}}}\frac{k!}{i_1!\dots i_s!}|\epsilon_1|^{i_1}\dots|\epsilon_s|^{i_s}\left\|\frac{\partial^k\varphi}{\partial\alpha_1^{i_1}\dots\partial\alpha_s^{i_s}}(\alpha)\right\|_\infty.$$

Il vient

$$\left\|\left[\sum_{i=1}^{s}\epsilon_i\frac{\partial\varphi}{\partial\alpha_i}(\alpha)\right]^{[k]}\right\|_\infty \leq \binom{s}{k}k!\epsilon^k\max\left(1, \max_{j=k+1,\dots,s}(\binom{s-k}{j-k}M^{j-k})\right).$$

Nous en déduisons :

$$\|\varphi(\alpha+\epsilon)-\varphi(\alpha)\|_\infty \leq \sum_{k=1}^{s-1}\binom{s}{k}\epsilon^k\max\left(1, \max_{j=k+1,\dots,s}(\binom{s-k}{j-k}M^{j-k})\right)+\epsilon^s.$$

Comme $\epsilon < 1$, cela donne le résultat souhaité. □

Il en découle alors :

Corollaire 4. *Avec les notations précédentes, si l'erreur ϵ sur les racines est majorée de la façon suivante :*

$$(*) \quad \epsilon \leq 0.5 \left(1 + \sum_{k=1}^{s-1} \binom{s}{k} \max\big(1, \max_{j=k+1,\dots,s} (\binom{s-k}{j-k} M^{j-k})\big)\right)^{-1}$$

alors l'erreur sur les coefficients est inférieure à 0.5.

Corollaire 5. *Si l'erreur ϵ sur les racines est majorée comme en $(*)$ alors nous pouvons reconnaître les coefficients de $f_\alpha(T)$ à partir de ceux de $f_{\alpha+\epsilon}(T)$.*

Il faut à présent remarquer que la connaissance de la précision demandée aux calculs ne permet pas de les certifier. En effet pour cela il faut donner le nombre de chiffres significatifs nécessaires lors des calculs.

Il nous reste donc à majorer les coefficients et nous pourrons alors en déduire le nombre de chiffres significatifs demandé pour les calculs.

Or nous avons posé $|\alpha_i| < M$ pour tout $1 \leq i \leq s$, il en découle $|S_k(\alpha_1,\dots,\alpha_s)| \leq \binom{s}{k} M^k$.

Nous en déduisons la proposition :

Proposition 15. *Avec les notations précédentes et en notant $Digits1$ le nombre de chiffres significatifs utilisés pour représenter les racines :*

Si

$$Digits1 \geq \left\lfloor \left| \log_{10}\left(0.5\left(1 + \sum_{k=1}^{s-1}\binom{s}{k}\max(1, \max_{j=k+1,\dots,s}(\binom{s-k}{j-k}M^{j-k}))\right)^{-1}\right)\right| + \log_{10}(\max_{k=1,\dots,s}(\binom{s}{k}M^k))\right\rfloor$$

alors nous pouvons reconnaître les coefficients de $f_\alpha(T)$ à partir de ceux de $f_{\alpha+\epsilon}(T)$.

Afin de donner un idée des ordres de grandeur qu'entraîne une telle formule nous donnons le tableau de valeurs suivants :

s	M	$Digits1$
2	10	3
2	10^5	15
2	10^{10}	30
2	10^{20}	60
5	10	10
5	10^5	46
5	10^{10}	90
5	10^{20}	181

s	M	$Digits1$
10	10	20
10	10^5	96
10	10^{10}	191
10	10^{20}	381
15	10	31
15	10^5	146
15	10^{10}	291
15	10^{20}	581

Nous voyons donc que la "précision" demandée est acceptable par un ordinateur.

REMARQUE :
Nous pouvons affiner le corolaire 4 en utilisant la mesure de Mahler (voir [MŞ99] page 79, [Mig89] page 158).

Définition 21. *Soit P un polynôme de $\mathbb{C}[T]$ qui s'écrit*

$$\begin{aligned} P(T) &= a_d T^d + a_{d-1}T^{d-1} + \ldots + a_0 \\ &= a_d(T - z_1)\cdots(T - z_d), \end{aligned}$$

avec $a_d \neq 0$.
La mesure (de Mahler) de P est définie par la formule

$$M(P) = |a_d| \prod_{j=1}^{d} \max(1, |z_i|).$$

En utilisant la mesure de Mahler du polynôme minimal de f_α dans la preuve du lemme 15 nous obtenons :

$$\left| \frac{\partial^k S_j}{\partial\alpha_1^{i_1} \ldots \partial\alpha_s^{i_s}}(\alpha) \right| \leq \binom{s-k}{j-k} M(f_\alpha).$$

Ainsi dans le corolaire 4, nous pouvons remplacer M^{j-k} par $M(f_\alpha)$. Cela présente l'avantage suivant si nous avons, par exemple, un seul α_i vérifiant $|\alpha_i| > 1$ alors dans ce cas $M(f_\alpha) < M^{j-k}$. La majoration sera donc plus fine dans ce cas. Cependant si tous les α_i vérifient $|\alpha_i| > 1$ alors nous pouvons obtenir, pour certaines valeurs de j et de k, $M(f_\alpha) > M^{j-k}$. Ainsi afin d'obtenir une formule "optimale", nous pouvons considérer le résultat suivant :

Corollaire 6. *Avec les notations précédentes et en notant D_1 le nombre de chiffres significatifs utilisés pour représenter les racines :*
Si

$$\begin{aligned} D_1 \geq & \left\lfloor \left| \log_{10} \left(0.5 \left(1 + \sum_{k=1}^{s-1} \binom{s}{k} \max\left(1, \max_{j=k+1,\ldots,s} \left(\binom{s-k}{j-k} \min(M(f_\alpha), M^{j-k})\right)\right)\right)^{-1}\right) \right| \right. \\ & \left. + \log_{10}\left(\max_{k=1,\ldots,s} \left(\binom{s}{k} \min(M(f_\alpha), M^k)\right)\right) \right\rfloor \end{aligned}$$

alors nous pouvons reconnaître les coefficients de $f_\alpha(T)$ à partir de ceux de $f_{\alpha+\epsilon}(T)$.

Une fois de plus afin de donner une idée des ordres de grandeur qu'entraine une telle formule nous allons donner un tableau de valeurs. Dans ce tableau nous avons supposé que le minimum était atteint par $M(f_\alpha)$. Nous rappelons que cette situation existe, par exemple, lorsqu'un seul des α_i vérifie $|\alpha_i| \geq 1$.

s	$M(f_\alpha)$	D_1
2	10	2
2	10^5	10
2	10^{10}	20
2	10^{20}	40
5	10	5
5	10^5	13
5	10^{10}	23
5	10^{20}	43

s	$M(f_\alpha)$	D_1
10	10	8
10	10^5	16
10	10^{10}	26
10	10^{20}	46
15	10	12
15	10^5	20
15	10^{10}	30
15	10^{20}	50

Ainsi, lorsque le minimum est atteint pour la mesure de Mahler, dans la formule du corollaire 6, nous diminuons trés sensiblement notre exigence sur la précision. Cependant il ne faut pas oublier que dans certains cas nous pouvons avoir $M(f_\alpha) = M^s$. Dans ce cas, dans la formule du corollaire 6, le minimum ne sera pas atteint par $M(f_\alpha)$.

2.3 Une première approche pour obtenir une factorisation exacte

A présent nous nous trouvons dans une situation où l'on connaît un élément primitif α de $K = \mathbb{Q}[\alpha]$ qui est un entier algébrique sur $\mathbb{Z}$. Nous voulons exprimer tous les coefficients de P_1 comme polynômes en α à coefficients dans $\mathbb{Q}$.

De plus les coefficients de P_1 sont connus avec une certaine précision, et nous voulons donner l'expression exacte de ses coefficients dans $\mathbb{Q}[\alpha]$. En 2.3.2 nous verrons comment exprimer un coefficient $a \in K$ de P_1 sous la forme :

$a \approx \tilde{q}_0 + \tilde{q}_1\alpha + ... + \tilde{q}_{s-1}\alpha^{s-1}$ où les $\tilde{q}_i$ sont des valeurs approchées des fractions q_i qui vérifient $a = q_0 + q_1\alpha + ... + q_{s-1}\alpha^{s-1}$.

Nous allons donc devoir reconnaître les q_i à partir des $\tilde{q}_i$. Cela pourrait se faire à l'aide de techniques du type fraction continue, mais ici nous allons procéder différemment et nous verrons (voir la section 2.5) que notre approche est plus satisfaisante.

Une nouvelle fois l'idée est de se ramener à reconnaître des éléments de $\mathbb{Z}$. Nous allons tout d'abord trouver le dénominateur des q_i, puis nous reconnaîtrons leur numérateur.

2.3.1 Discriminant et dénominateur commun

Le lemme suivant (voir [Rib01]) nous montre comment trouver le dénominateur des q_i.

Lemme 16. *Si α est un entier algébrique sur $\mathbb{Z}$, et $\mathcal{O}_K$ la fermeture intégrale de $\mathbb{Z}$*

dans $\mathbb{Q}[\alpha]$ *alors tout élément de* $\mathcal{O}_K$ *s'écrit :*

$$a = \frac{z_0}{\Delta} + \frac{z_1}{\Delta}\alpha + ... + \frac{z_{s-1}}{\Delta}\alpha^{s-1}$$

où Δ *est le discriminant de* α *sur* $\mathbb{Q}$ *et* $z_i \in \mathbb{Z}$.

NOTATION : Dans la suite nous conserverons la notation $\mathcal{O}_K$ pour la fermeture intégrale de $\mathbb{Q}[\alpha]$ sur $\mathbb{Z}$.

Ce lemme est suffisant d'un point de vue théorique, mais d'un point de vue pratique nous allons en demander plus. En effet on peut demander que le dénominateur commun Δ soit petit pour faciliter la lecture du résultat. De plus nous verrons dans la partie 2.3.2 que plus le dénominateur commun sera petit plus la précision demandée dans les calculs sera faible.

A présent nous allons montrer comment réduire le discriminant pour obtenir un dénominateur commun petit (c'est-à-dire un élément $d \in \mathbb{Z}$ ayant la même propriété que Δ dans le lemme précédent et tel que d divise Δ). Ce problème a déjà été étudié dans [All01], à l'aide d'une factorisation partielle du discriminant. Ici nous proposons une autre approche qui ne demande pas de factorisation mais uniquement des calculs de pgcd.

Réduction du discriminant

Un théorème classique de théorie algébrique des nombres (voir [Rib01], [Sam67]) nous dit que :

Théorème 22. *Soit* K *une extension de degré finie* s *de* $\mathbb{Q}$, $\mathcal{O}_K$ *la fermeture intégrale de* K *sur* $\mathbb{Z}$. *Alors* $\mathcal{O}_K$ *est un* $\mathbb{Z}$ *module libre de rang* s.

Définition 22. *Soit* $\{x_1, ..., x_s\}$ *une* $\mathbb{Z}$*-base de* $\mathcal{O}_K$ *alors :*

$$disc_{K/\mathbb{Q}}(x_1, ..., x_s) = \delta_K$$

s'appelle le discriminant absolu de l'extension.

On a donc $\delta_K \in \mathbb{Z}$ et $\delta_K \neq 0$. De plus nous avons le résultat suivant :(voir [Sam67]), et le rappel sur le discriminant effectué page vii.

Lemme 17. *Si* $(y_1, ..., y_s) \in \mathcal{O}_K^s$ *alors on a :* $y_i = \sum_{j=1}^{s} a_{ij}x_j$ *avec* $a_{ij} \in \mathbb{Z}$, *et,* $disc_{K/\mathbb{Q}}(y_1, ..., y_s) = det(a_{ij})^2\delta_K$.

Ainsi en prenant $(y_1, ..., y_s) = (1, \alpha, ..., \alpha^{s-1})$ où α est un élément primitif de K on obtient :

$$M_\alpha \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha \\ \vdots \\ \alpha^{s-1} \end{pmatrix},$$

où $M_\alpha = (a_{ij})_{i,j} \in \mathcal{M}_s(\mathbb{Z})$.
Cela donne d'après le lemme précédent : $disc_{K/\mathbb{Q}}(\alpha) = (det(M_\alpha))^2\delta_K$.
Comme α est primitif on a $disc_{K/\mathbb{Q}}(\alpha) \neq 0 \Rightarrow M_\alpha \in \mathcal{GL}(\mathbb{Q})$.

M_α étant inversible on obtient avec les formules de Cramer : $x_i = \dfrac{p_i(\alpha)}{det(M_\alpha)}$ où $p_i(T) \in \mathbb{Z}[T]$. Nous venons donc de démontrer le résultat suivant :

Lemme 18. *Soit α un entier algébrique primitif de $K(= \mathbb{Q}[\alpha])$. Nous avons l'égalité $disc_{K/\mathbb{Q}}(\alpha) = D_\alpha^2\delta_K$ où $D_\alpha \in \mathbb{Z}$. De plus D_α est tel que tout élément a de $\mathcal{O}_K$ s'écrit :*

$$a = \frac{n_0}{D_\alpha} + \frac{n_1}{D_\alpha}\alpha + ... + \frac{n_{s-1}}{D_\alpha}\alpha^{s-1} \text{ avec } n_i \in \mathbb{Z}.$$

Une heuristique pour obtenir D_α

A présent nous allons présenter une heuristique qui nous permettra de calculer un dénominateur commun d en n'utilisant que des calculs de pgcd.

Pourquoi utiliser une heuristique ?

Le calcul du dénominateur commun d est un problème difficile. En effet calculer d revient soit à calculer le plus grand facteur carré de $disc_{\mathbb{K}/\mathbb{Q}}(\alpha)$ soit à calculer δ_K. D'autre part un résultat dû à Chistov (voir [Len87]) nous dit que cela est aussi difficile que de calculer une base entière de $\mathbb{Q}[\alpha]$.

L'algorithme Zassenhauss'Round 2 permet d'obtenir une base entière de $\mathcal{O}_K$. Cependant cet algorithme nécessite la factorisation dans $\mathbb{Z}$ du discriminant de α et ici les discriminants peuvent être de l'ordre de 10^{70} comme nous le verrons en 2.5.2.

Nous proposons donc ici une heuristique afin d'éviter la factorisation dans $\mathbb{Z}$ de nombres pouvant être supérieurs à 10^{70}.

Nous signalons ici qu'une autre heuristique a été proposée par B. Allombert dans sa thèse (voir [All01]). Celui-ci utilise une factorisation partielle et le polynôme minimal d'un seul élément primitif.

Passons à présent à la description de notre heuristique.
Soient α un élément primitif de K et $a_1^{(u,v)}$ les coefficients de P_1. Nous avons

$$disc_{\mathbb{K}/\mathbb{Q}}(\alpha) = D_\alpha^2\delta_{\mathbb{K}}, \text{ et } disc_{\mathbb{K}/\mathbb{Q}}(a_1^{(i_k,j_k)}) = D^2_{a_1^{(i_k,j_k)}}\delta_K, \text{ pour } k = 1, \dots, l.$$

Nous notons par p et w les deux nombres entiers suivants :

$$p = \text{pgcd}(disc_{\mathbb{K}/\mathbb{Q}}(\alpha), disc_{\mathbb{K}/\mathbb{Q}}(a_1^{(i_1,j_1)}), \dots, disc_{\mathbb{K}/\mathbb{Q}}(a_1^{(i_l,j_l)})) = w^2\delta_K,$$

$$\text{où } w = \text{pgcd}(D_\alpha, D_{a_1^{(i_1,j_1)}}, \dots, D_{a_1^{(i_l,j_l)}}).$$

On pose alors :

$$d' = \sqrt{\frac{disc_{\mathbb{K}/\mathbb{Q}}(\alpha)}{p}} = \frac{D_\alpha}{w}.$$

Si $w = 1$ alors nous obtenons D_α qui est bien un dénominateur commun.

Si $w \neq 1$ alors nous obtenons un diviseur de D_α. Cependant en pratique d' est tel que $d'a_1^{(u,v)} \in \mathbb{Z}[\alpha]$. C'est en cela que cette méthode est une heuristique.

Remarques :
Des exemples de calculs des discriminants et de leurs réductions se trouvent dans les parties 2.5.1 et 2.5.2 .
Ces méthodes (calculs de pgcd de plusieurs éléments) sous entendent que nous avons à notre disposition plusieurs éléments primitifs. C'est le cas pratiquement.
Comme nous voulons obtenir le plus petit "dénominateur commun" possible, nous prendrons en pratique comme générateur de l'extension un élément primitif qui a un petit discriminant.

2.3.2 Reconnaissance des coefficients de P_1

Dans cette partie on se donne : un polynôme P de $\mathbb{Z}[X,Y]$ unitaire en Y et irréductible dans $\mathbb{Q}[X,Y]$, un élément primitif α de K (l'extension de $\mathbb{Q}$ engendrée par les coefficients de P_1) ainsi que tous ses conjugués, un dénominateur commun d pour les entiers de $\mathbb{Q}[\alpha]$ (voir 2.3.1) et le polynôme minimal de α. Nous avons vu comment obtenir toutes ces informations dans les parties précédentes.

Nous allons à présent décrire une méthode permettant d'obtenir une expression exacte des coefficients de P_1, pour cela nous allons utiliser les résultats de la partie 2.3.1.

Soit $a_1^{(u,v)}$ un coefficient de P_1, $a_1^{(u,v)}$ est donc un élément de $\mathcal{O}_K$. En conservant les notations de la partie 2.3.1 nous avons :

$$a_1^{(u,v)} = \frac{z_0}{d} + \frac{z_1}{d}\alpha + ... + \frac{z_{s-1}}{d}\alpha^{s-1}.$$

Cela donne avec les σ_i qui ont été définis en 2.2.1 :

$$a_i^{(u,v)} = \sigma_i(a_1^{(u,v)}) = \frac{z_0}{d} + \frac{z_1}{d}\sigma_i(\alpha) + ... + \frac{z_{s-1}}{d}\sigma_i(\alpha)^{s-1}.$$

Par conséquent, si nous posons $q_i = \frac{z_i}{d}$ nous obtenons :

$$(*) \begin{pmatrix} 1 & \sigma_1(\alpha) & \sigma_1(\alpha)^2 & \cdots & \sigma_1(\alpha)^{s-1} \\ 1 & \sigma_2(\alpha) & \sigma_2(\alpha)^2 & \cdots & \sigma_2(\alpha)^{s-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \sigma_s(\alpha) & \sigma_s(\alpha)^2 & \cdots & \sigma_s(\alpha)^{s-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{s-1} \end{pmatrix} = \begin{pmatrix} a_1^{(u,v)} \\ a_2^{(u,v)} \\ \vdots \\ a_s^{(u,v)} \end{pmatrix}.$$

La matrice obtenue est une matrice de Vandermonde.

Ainsi si nous résolvons le système $(*)$ on obtient les q_i, puis en multipliant par d on en déduit les z_i. Un problème se pose alors : les α_i et les a_i ne sont pas connus de manière exacte. Si nous résolvons le système $(*)$ nous n'obtenons donc pas les q_i mais des $\tilde{q}_i$ proches de ceux-ci. C'est-à-dire : $\tilde{q}_i = q_i + e$, où e désigne l'erreur.

De ce fait : $d\tilde{q}_i = z_i + de \quad (**)$.

Une nouvelle fois si $| de |< 0,5$ nous pourrons reconnaître z_i car se sera l'entier le plus proche de $d\tilde{q}_i$.

Remarques :
Dans les lignes ci-dessus $\tilde{q}_i$ est annoncé comme étant proche de q_i. Cela sous-entend que les erreurs sur α_i et a_i ne perturbent pas trop la solution du système $(*)$. L'influence des perturbations sur les données d'un système est mesurée par le conditionnement de celui-ci. L'étude du conditionnement des systèmes de type Vandermonde est complexe (voir [Gau90]). Cependant il existe une méthode qui semble "relativement stable"pour résoudre de tels systèmes, et celle-ci a une complexité en $O(s^2)$. Cette méthode utilise le lien qui existe entre les matrices de Vandermonde et l'interpolation polynomiale, elle consiste à effectuer un changement de base dans l'espace des polynômes (voir [Hig02]). Björk et Pereyra ont dit à ce propos (voir [BP70]) : " It seems as if at least some problems connected with Vandermonde systems, which traditionally have been considered too ill-conditioned to be attacked, actually can be solved with good precision."

Ainsi, afin de permettre une bonne reconnaissance de z_i, *il faut prendre d le plus petit possible*. En effet nous limiterons alors l'effet de l'erreur de dans $(**)$.

2.3.3 Description de l'algorithme

Algorithme Appexact1
Entrée : $Q \in \mathbb{Q}[X,Y]$ irréductible dans $\mathbb{Q}[X,Y]$, unitaire en Y, *facto-app* un algorithme de factorisation approchée.
Sortie : $Q_1(X,Y) \in \mathbb{Q}[\alpha][X,Y]$ un facteur exact de Q, $f_\alpha(T) \in \mathbb{Z}[T]$ le polynôme minimal de α.

1. Se ramener au cas $P \in \mathbb{Z}[X,Y]$.
2. Effectuer *facto-app*(P), la factorisation approchée de P avec un nombre de chiffres significatifs $= Digits$.
3. Calculer $Digits1$ (voir 2.2.3) si $Digits1 \geq Digits$ alors recommencer l'étape précédente avec $Digits = Digits1$, sinon continuer.
4. Reconnaître tous les coefficients de P_1 qui sont des éléments primitifs, leur polynôme minimal, leur discriminant.
   (Si aucun des coefficients n'est primitif alors : en fabriquer 11 (par exemple), reconnaître leur polynôme minimal puis leur discriminant.)
5. Trouver l'élément primitif α ayant le plus petit discriminant, et noter f_α son polynôme minimal.
6. Calculer p le pgcd des discriminants obtenus en 4.

7. $d' = \sqrt{\dfrac{disc_{\mathbb{K}/\mathbb{Q}}(\alpha)}{p}}$.
8. Reconnaissance des coefficients exacts de P_1 à l'aide du système de Vandermonde.
9. Vérifier que P_1 divise P. Si ce n'est pas le cas alors : recommencer à partir de l'étape 2 en augmentant *Digits*.
10. Retourner le facteur de Q correspondant.

2.4 Une deuxième approche pour obtenir la factorisation exacte

Dans la méthode proposée ci-dessus une des difficultés est le calcul d'un dénominateur commun. Cela nous a conduit à donner une démarche heuristique. Dans cette partie nous allons exposer comment obtenir une expression exacte de P_1, mais ici les coefficients de P_1 ne seront plus exprimés sous la forme $\frac{z_0}{d} + \frac{z_1}{d}\alpha + ... + \frac{z_{s-1}}{d}\alpha^{s-1}$ mais sous la forme $\frac{z_0}{f'_\alpha(\alpha)} + \frac{z_1}{f'_\alpha(\alpha)}\alpha + ... + \frac{z_{s-1}}{f'_\alpha(\alpha)}\alpha^{s-1}$.

Cette représentation semble moins naturelle mais elle présente certains avantages comme nous le verrons en 2.4.3.

2.4.1 $f'_\alpha(\alpha)$ est un dénominateur commun

Dans cette partie nous rappelons certaines définitions et propriétés de théorie algébrique des nombres. Le but est de montrer brièvement que $f'_\alpha(\alpha)$ est un dénominateur commun pour tous les éléments de $\mathcal{O}_K$.

Définition 23. *Soit K une extension finie de $\mathbb{Q}$.*
Soit M une sous partie de K.
On pose $M^* = \{x \in K \mid \forall y \in M, Tr_{K/\mathbb{Q}}(xy) \in \mathbb{Z}\}$.

Soit α un entier algébrique de K, on a alors : $\mathcal{O}_K \subset \mathbb{Z}[\alpha]^*$. D'autre part nous avons la propriété suivante (voir [Rib01] page 242) :

Proposition 16. *Soient K une extension finie de $\mathbb{Q}$, α un entier algébrique sur $\mathbb{Z}$ qui est primitif et f_α son polynôme minimal. Nous avons alors :*

$$\mathbb{Z}[\alpha]^* = \frac{1}{f'_\alpha(\alpha)}\mathbb{Z}[\alpha].$$

Nous en déduisons donc :

Corollaire 7. *Avec les notations précédentes nous avons :*

$$\mathcal{O}_K \subset \frac{1}{f'_\alpha(\alpha)}\mathbb{Z}[\alpha].$$

Cela signifie que tout $a \in \mathcal{O}_K$ s'écrit sous la forme :

$$a = \frac{z_0}{f'_\alpha(\alpha)} + \frac{z_1}{f'_\alpha(\alpha)}\alpha + ... + \frac{z_{s-1}}{f'_\alpha(\alpha)}\alpha^{s-1} \text{ où } z_i \in \mathbb{Z}.$$

2.4.2 Reconnaissance des coefficients de P_1

A présent nous allons expliquer comment obtenir les coefficients de P_1 sous la forme $\frac{z_0}{f'_\alpha(\alpha)} + \frac{z_1}{f'_\alpha(\alpha)}\alpha + ... + \frac{z_{s-1}}{f'_\alpha(\alpha)}\alpha^{s-1}$. La démarche est la même que celle vue dans la section précédente.

Soit $a_1^{(u,v)}$ un coefficient de P_1, $a_1^{(u,v)}$ est donc un élément de $\mathcal{O}_K$. En conservant les notations de la partie 2.3.2 on a :

$$a_1^{(u,v)} = \frac{z_0}{f'_\alpha(\alpha)} + \frac{z_1}{f'_\alpha(\alpha)}\alpha + ... + \frac{z_{s-1}}{f'_\alpha(\alpha)}\alpha^{s-1} \text{ où } z_i \in \mathbb{Z}.$$

Donc on a :

$$a_i^{(u,v)} = \frac{z_0}{f'_\alpha(\alpha_i)} + \frac{z_1}{f'_\alpha(\alpha_i)}\sigma_i(\alpha) + ... + \frac{z_{s-1}}{f'_\alpha(\alpha_i)}\sigma_i(\alpha)^{s-1}.$$

On en déduit :

$$(\star) \begin{pmatrix} 1 & \sigma_1(\alpha) & \sigma_1(\alpha)^2 & \cdots & \sigma_1(\alpha)^{s-1} \\ 1 & \sigma_2(\alpha) & \sigma_2(\alpha)^2 & \cdots & \sigma_2(\alpha)^{s-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \sigma_s(\alpha) & \sigma_s(\alpha)^2 & \cdots & \sigma_s(\alpha)^{s-1} \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_{s-1} \end{pmatrix} = \begin{pmatrix} f'_\alpha(\sigma_1(\alpha))a_1^{(u,v)} \\ f'_\alpha(\sigma_2(\alpha))a_2^{(u,v)} \\ \vdots \\ f'_\alpha(\sigma_s(\alpha))a_s^{(u,v)} \end{pmatrix}$$

Comme précédemment, nous remarquons qu'en pratique nous n'avons pas $a_i^{(u,v)}$ mais $a_i^{(u,v)} + \nu_i$, et nous n'avons pas $\sigma_i(\alpha)$ mais $\sigma_i(\alpha) + \epsilon_i$. Donc nous devons résoudre ce système de Vandermonde et prendre l'entier le plus proche pour chaque coefficient de la solution. Nous allons voir qu'avec cette démarche nous pouvons certifier nos résultats.

2.4.3 Choix de la précision

Dans cette sous-section nous allons répondre à la question suivante : avec quelle précision faut il faire les calculs pour que ceux ci soient certifiés ?

Avant de détailler les calculs nous allons poser quelques notations :

NOTATIONS :

On rappelle que $\mathcal{M}_{s,s}(\mathbb{C})$ désigne l'anneau des matrices à s lignes et s colonnes à coefficients dans $\mathbb{C}$. Soit $\mathcal{M} = (m_{i,j})_{i,j=0}^{s-1}$ une matrice de $\mathcal{M}_{s,s}(\mathbb{C})$, on pose $\|\mathcal{M}\|_\infty = \max_{i=0,\dots,s-1} \sum_{j=0}^{s-1} |m_{i,j}|$, et pour $\vec{v} = (v_1, \dots, v_s) \in \mathbb{C}^s$, $\|\vec{v}\|_\infty = \max_{i=0,\dots,s-1} |v_i|$.

Nous avons alors $\|\mathcal{M}\vec{v}\|_\infty \leq \|\mathcal{M}\|_\infty \|\vec{v}\|_\infty$.
On pose $\alpha_i = \sigma_i(\alpha)$, ϵ_i est l'erreur sur α_i, ν_i est l'erreur sur $a_i^{(u,v)}$, et e_i est l'erreur sur z_i. ϵ est un nombre réel vérifiant :

$$\begin{cases} \forall 1 \leq i \leq s & |\epsilon_i| < \epsilon < 1, \\ \forall 1 \leq i \leq s & |\nu_i| < \epsilon < 1. \end{cases}$$

M est un nombre réel tel que $\max_{i,u,v} |a_i^{(u,v)}| \leq M$. $\mathcal{M}(\alpha)$ est la matrice suivante :

$$\mathcal{M}(\alpha) = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{s-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{s-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_s & \alpha_s^2 & \cdots & \alpha_s^{s-1} \end{pmatrix}^{-1} \begin{pmatrix} f'_\alpha(\alpha_1) & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \\ 0 & \dots & f'_\alpha(\alpha_k) & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & f'_\alpha(\alpha_s) \end{pmatrix},$$

$\vec{z}$, $\vec{a}^{(u,v)}$, $\vec{e}$, $\vec{\epsilon}$, et $\vec{\nu}$ sont les vecteurs suivants :

$$\vec{z} = \begin{pmatrix} z_0 \\ \vdots \\ z_{s-1} \end{pmatrix}, \quad \vec{a}^{(u,v)} = \begin{pmatrix} a_1^{(u,v)} \\ \vdots \\ a_s^{(u,v)} \end{pmatrix}, \quad \vec{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_s \end{pmatrix}, \quad \vec{\epsilon} = \begin{pmatrix} \epsilon_0 \\ \vdots \\ \epsilon_{s-1} \end{pmatrix}, \quad \vec{\nu} = \begin{pmatrix} \nu_0 \\ \vdots \\ \nu_{s-1} \end{pmatrix}.$$

Nous avons alors : $\vec{z} + \vec{e} = \mathcal{M}(\alpha + \epsilon)(\vec{a} + \vec{\epsilon})$.

Nous allons donc dans un premier temps donner une expression des coefficients de $\mathcal{M}(\alpha)$ en fonction des α_i. Nous pourrons alors donner une expression des coefficients de $\mathcal{M}(\alpha + \epsilon)$ en fonction des α_i et des ϵ_i. Nous en déduirons alors que :
$\mathcal{M}(\alpha + \epsilon) = \mathcal{M}(\alpha) + \epsilon\mathcal{N}$ où $\mathcal{N}$ est une matrice dont nous bornerons les coefficients.
Il en découlera alors :

$$\|\vec{e}\|_\infty \leq \big(\|\mathcal{M}(\alpha)\|_\infty + \|\mathcal{N}\|_\infty (1 + M)\big)\|\vec{\epsilon}\|_\infty$$

Expression des coefficients de $\mathcal{M}(\alpha)$ et de $\mathcal{M}(\alpha + \epsilon)$

L'objectif de cette partie est de montrer le lemme suivant et d'en donner des corollaires.

Lemme 19. *Soit $\mathcal{M}(\alpha) = (m_{i,j})_{i,j=0}^{s-1}$ nous avons :*

$$m_{i,j} = (-1)^{s-i-1} S_{s-i-1}(\alpha_1, \dots, \alpha_j, \alpha_{j+2}, \dots, \alpha_s)$$

Démonstration. Tout d'abord nous allons exprimer à l'aide des α_i les coefficients de l'inverse de la matrice $V(\alpha)$ où $V(\alpha)$ est la matrice de Vandermonde.

Soit $V(\alpha)^{-1} = (w_{i,j})_{i,j=0}^{s-1}$ l'inverse de $V(\alpha)$.

La relation $V(\alpha)V(\alpha)^{-1} = Id$ dans $\mathcal{M}_{s,s}(\mathbb{C})$ fournit les égalités suivantes :

$$\sum_{j=0}^{s-1} \alpha_{i+1}^j w_{j,k} = \delta_{i,k},$$

où $\delta_{i,k}$ est le symbole de Kronecker.

Ainsi le polynôme

$$l_k(x) = \sum_{j=0}^{s-1} w_{j,k} x^j$$

prend la valeur 1 si $x = \alpha_{k+1}$ et 0 si $x \in \{\alpha_1, \ldots, \alpha_s\} \setminus \{\alpha_{k+1}\}$.

$l_k(x)$ est donc un polynôme de Lagrange et nous obtenons alors :

$$l_k(x) = \prod_{\substack{i=1\\ i\neq k+1}}^{s} \left(\frac{x-\alpha_i}{\alpha_{k+1}-\alpha_i}\right) = \prod_{\substack{i=1\\ i\neq k+1}}^{s} (x-\alpha_i) \times \frac{1}{f'_\alpha(\alpha_{k+1})}.$$

Il vient alors :

$$w_{j,k} = \frac{(-1)^{s-1-j} S_{s-j-1}(\alpha_1, \ldots, \alpha_k, \alpha_{k+2}, \ldots, \alpha_s)}{f'_\alpha(\alpha_{k+1})},$$

où les S_k sont les polynômes symétriques élémentaires rencontrés en 2.2.3 et on pose $S_0 = 1$.

A présent passons aux coefficients de $\mathcal{M}(\alpha)$.

Par définition de $\mathcal{M}(\alpha) = (m_{i,j})_{i,j=0}^{s-1}$, nous avons : $m_{i,j} = w_{i,j} f'_\alpha(\alpha_{j+1})$. Ce qui donne le résultat souhaité. □

Nous obtenons alors comme conséquence de ce lemme :

Corollaire 8. *Soit $\mathcal{M}(\alpha+\epsilon) = (m_{i,j}(\epsilon))_{i,j=0}^{s-1}$. Nous avons*

$$m_{i,j}(\epsilon) = m_{i,j} + c_{i,j}\epsilon,$$

où : $|c_{i,j}| \leq 1 + \sum_{k=1}^{s-2} \binom{s-1}{k} \max\big(1, \max_{j=k+1,\ldots,s-1} \big(\binom{s-1-k}{j-k} M^{j-k}\big)\big).$

Démonstration. D'après le lemme précédent nous avons :

$$m_{i,j}(\epsilon) = (-1)^{s-i-1} S_{s-i-1}(\alpha_1+\epsilon_1, \ldots, \alpha_j+\epsilon_j, \alpha_{j+2}+\epsilon_{j+2}, \ldots, \alpha_s+\epsilon_s).$$

D'où :

$$\begin{aligned} m_{i,j}(\epsilon) - m_{i,j} = \quad (-1)^{s-i-1} \quad & \left(S_{s-i-1}(\alpha_1+\epsilon_1, .., \alpha_j+\epsilon_j, \alpha_{j+2}+\epsilon_{j+2}, .., \alpha_s+\epsilon_s)\right. \\ & \left. -S_{s-i-1}(\alpha_1, .., \alpha_j, \alpha_{j+2}, .., \alpha_s)\right). \end{aligned}$$

En appliquant le lemme 15 nous obtenons :

$$|m_{i,j}(\epsilon) - m_{i,j}| \leq \Big(1 + \sum_{k=1}^{s-2} \binom{s-1}{k} \max\big(1, \max_{j=k+1,\ldots,s-1} \big(\binom{s-1-k}{j-k} M^{j-k}\big)\big)\Big)\epsilon$$

Ce qui donne le résultat annoncé. □

Ce résultat se réécrit ainsi :

Corollaire 9. *Il existe une matrice $\mathcal{N} \in \mathcal{M}_{s,s}(\mathbb{C})$ telle que :*

$$\mathcal{M}(\alpha+\epsilon) = \mathcal{M}(\alpha) + \epsilon\mathcal{N}$$

avec $\|\mathcal{N}\|_\infty \leq s\left(1+\sum_{k=1}^{s-2}\binom{s-1}{k}\max\left(1, \max_{j=k+1,\ldots,s-1}\left(\binom{s-1-k}{j-k}M^{j-k}\right)\right)\right).$

Majoration de $\|\vec{e}\|_\infty$

Nous venons de montrer l'égalité $\mathcal{M}(\alpha+\epsilon) = \mathcal{M}(\alpha)+\epsilon\mathcal{N}$. Il en découle alors le lemme suivant :

Lemme 20. *Avec les notations précédentes nous avons :*

$$\|\vec{e}\|_\infty \leq (\|\mathcal{M}(\alpha)\|_\infty + \|\mathcal{N}\|_\infty(1+M))\epsilon.$$

Démonstration. L'égalité $\vec{z}+\vec{e} = \mathcal{M}(\alpha+\epsilon)(\vec{a}+\vec{\epsilon})$ devient : $\vec{z}+\vec{e} = (\mathcal{M}(\alpha)+\epsilon\mathcal{N})(\vec{a}+\vec{\epsilon})$. On en déduit : $\vec{e} = \mathcal{M}(\alpha)\vec{\epsilon} + \epsilon\mathcal{N}\vec{a} + \epsilon\mathcal{N}\vec{\epsilon}$. D'où :

$$\|\vec{e}\|_\infty \leq \|\mathcal{M}(\alpha)\|_\infty\epsilon + \|\mathcal{N}\|_\infty\epsilon^2 + \|\mathcal{N}\vec{a}\|_\infty\epsilon \leq (\|\mathcal{M}(\alpha)\|_\infty + \|\mathcal{N}\|_\infty + \|\mathcal{N}\|_\infty M)\epsilon.$$

□

Conclusion

Les solutions du système sont entières nous pourrons donc les "reconnaître" si $\|\vec{e}\|_\infty < 0.5$.

En réunissant tous les résultats obtenus dans cette partie nous obtenons la proposition suivante :

Proposition 17. *Si l'erreur ϵ commise sur les coefficients des facteurs P_i vérifie :*

$$\begin{aligned}\epsilon \quad \leq 0.5 \quad & \left[\max_{i=0,\ldots,s-1}\left(s\binom{s-1}{s-i-1}M^{s-i-1}\right)\right. \\ & \left. + s\left(1+\sum_{k=1}^{s-2}\binom{s-1}{k}\max\left(1, \max_{j=k+1,\ldots,s-1}\left(\binom{s-1-k}{j-k}M^{j-k}\right)\right)\right)(1+M)\right]^{-1},\end{aligned}$$

alors nous pouvons à l'aide du système ($\star$) présenté en 2.4.2 "reconnaître" les coefficients exacts de P_1.

Démonstration. Le lemme 19 nous donne : $|m_{i,j}| \leq S_{s-i-1}(|\alpha_1|,\ldots,|\alpha_j|,|\alpha_{j+2}|,\ldots,|\alpha_s|)$. Donc : $|m_{i,j}| \leq \sum_{1\leq k_1<\ldots<k_{s-i-1}\leq s-1} M^{s-i-1} \leq \binom{s-1}{s-i-1}M^{s-i-1}$. Il en découle :

$$\sum_{j=0}^{s-1}|m_{i,j}| \leq \sum_{j=0}^{s-1}\binom{s-1}{s-i-1}M^{s-i-1} = s\binom{s-1}{s-i-1}M^{s-i-1}.$$

D'où :

$$\|\mathcal{M}(\alpha)\|_\infty \leq \max_{i=0,\ldots,s-1}(s\binom{s-1}{s-i-1}M^{s-i-1}).$$

D'autre part nous avons :

$$\|\mathcal{N}\|_\infty \leq s\left(1+\sum_{k=1}^{s-2}\binom{s-1}{k}\max\left(1,\max_{j=k+1,\ldots,s-1}(\binom{s-1-k}{j-k}M^{j-k})\right)\right).$$

Ainsi en remplaçant par ces valeurs dans la majoration du lemme 20 on obtient :

$$\begin{aligned}\|\vec{e}\|_\infty \quad \leq \quad & \Bigg[\max_{i=0,\ldots,s-1}(s\binom{s-1}{s-i-1}M^{s-i-1}) \\ & + s\left(1+\sum_{k=1}^{s-2}\binom{s-1}{k}\max\left(1,\max_{j=k+1,\ldots,s-1}(\binom{s-1-k}{j-k}M^{j-k})\right)\right)(1+M)\Bigg]\epsilon.\end{aligned}$$

Nous en déduisons alors le résultat désiré. □

Comme nous l'avons remarqué en 2.2.3 : donner la précision n'est pas suffisant pour certifier les calculs. Il nous faut une fois de plus le nombre de chiffres significatifs. Il ne nous reste donc plus qu'à majorer $\|\vec{z}\|_\infty$. Or $\vec{z} = \mathcal{M}(\alpha)\vec{a}$, il vient donc

$$\|\vec{z}\|_\infty \leq \|\mathcal{M}(\alpha)\|_\infty\|\vec{a}\|_\infty \leq \|\mathcal{M}(\alpha)\|_\infty M.$$

Nous venons donc de montrer :

Proposition 18. *Avec les notations précédentes et en notant $Digits2$ le nombre de chiffres significatifs utilisé pour représenter les coefficients des facteurs $\tilde{P}_i$, et les racines α_i de f_α ; si*

$$\begin{aligned}Digits2 \quad \geq \quad & \Bigg\lfloor \Bigg|\log_{10}\Bigg(0.5\Bigg[\max_{i=0,\ldots,s-1}(s\binom{s-1}{s-i-1}M^{s-i-1}) \\ & +s\left(1+\sum_{k=1}^{s-2}\binom{s-1}{k}\max\left(1,\max_{j=k+1,\ldots,s-1}(\binom{s-1-k}{j-k}M^{j-k})\right)\right)(1+M)\Bigg]^{-1}\Bigg)\Bigg| \\ & +\log_{10}\left(\max_{i=0,\ldots,s-1}(s\binom{s-1}{s-i-1}M^{s-i-1})M\right)\Bigg\rfloor\end{aligned}$$

alors nous pourrons reconnaître les coefficients de P_1 à partir de la résolution du système $(\star)$.

Dans la proposition 18, $Digits2$ est minoré par une expression ne dépendant que de s et de M. Le nombre s est connu car c'est le nombre de facteurs approchés et M s'obtient pratiquement, étant donné que l'erreur commise sur les coefficients est supposée inférieure à 1. Ainsi la proposition 18 nous donne un moyen pratique de connaître la précision nécessaire aux calculs pour la reconnaissance des coefficients de P_1.

Afin de donner une idée sur l'ordre de grandeur qu'implique une telle formule nous donnons les tableaux suivants :

s	M	$Digits2$
2	10	3
2	10^5	15
2	10^{10}	30
2	10^{20}	60
5	10	10
5	10^5	46
5	10^{10}	91
5	10^{20}	181

s	M	$Digits2$
10	10	21
10	10^5	97
10	10^{10}	192
10	10^{20}	382
15	10	32
15	10^5	147
15	10^{10}	292
15	10^{20}	582

Remarques :
En effectuant la reconnaissance du polynôme minimal comme expliqué en 2.2.3, nous obtenons un polynôme minimal certifié. Si nous effectuons alors les calculs pour reconnaître les coefficients de P_1 avec la précision donnée par la proposition 18, il ne sera plus nécessaire de vérifier notre résultat. L'étape qui consistait à diviser P par le facteur obtenu n'a donc plus de raisons d'exister.
Nous pouvons obtenir une estimation plus fine en ne majorant pas les $|\alpha_i|$ par M mais par $\max_i(|\alpha_i|)$. La précision demandée aux calculs dépendrait alors de l'élément primitif choisi.
Dans la deuxième méthode nous avons pu mener à son terme notre analyse d'erreur car nous pouvons effectuer une simplification par $f'_{\alpha_{k+1}}$. Cette simplification n'apparaît pas dans la première méthode.

2.4.4 Retour à la représentation canonique

Nous avons vu dans cette partie comment exprimer les coefficients de P_1 sous la forme : $\frac{z_0}{f'_\alpha(\alpha)} + \frac{z_1}{f'_\alpha(\alpha)}\alpha + ... + \frac{z_{s-1}}{f'_\alpha(\alpha)}\alpha^{s-1}$.

Or la représentation canonique d'un élément de $K = \mathbb{Q}[\alpha]$ est du type : $q_0 + q_1\alpha + ... + q_{s-1}\alpha^{s-1}$ avec $q_i \in \mathbb{Q}$. Nous allons voir à présent comment revenir à cette représentation.

Soit β un élément de $\mathcal{O}_K$. Nous avons alors les deux égalités suivantes :

$$\beta = \sum_{j=0}^{s-1} \frac{z_j}{f'_\alpha(\alpha)}\alpha^j = \sum_{i=0}^{s-1} q_i\alpha^i \text{ où } z_j \in \mathbb{Z} \text{ et } q_i \in \mathbb{Q}.$$

De plus notons $B(\alpha)$ l'inverse de $f'_\alpha(\alpha)$, et posons : $\alpha^j B(\alpha) = \sum_{i=0}^{s-1} b_{i,j}\alpha^i$ où $b_{i,j} \in \mathbb{Q}$.

Il en découle alors : $\beta = \sum_{j=0}^{s-1} z_j \sum_{i=0}^{s-1} b_{i,j}\alpha^i = \sum_{i=0}^{s-1}\left(\sum_{j=0}^{s-1} b_{i,j}z_j\right)\alpha^i$. Nous en déduisons alors : $q_i = \sum_{j=0}^{s-1} b_{i,j}z_j$.

Nous venons donc de montrer le lemme :

Lemme 21. *Avec les notations précédentes et en posant :*

$$\vec{q} = \begin{pmatrix} q_0 \\ \vdots \\ q_{s-1} \end{pmatrix}, \quad \vec{z} = \begin{pmatrix} z_0 \\ \vdots \\ z_{s-1} \end{pmatrix}, \quad \mathcal{M}_B = (b_{i,j})_{i,j=0}^{s-1} \in \mathcal{M}_{s,s}(\mathbb{Q}),$$

nous avons $\vec{q} = \mathcal{M}_B(\vec{z})$

Le polynôme minimal $f_\alpha(T)$ étant connu nous obtenons facilement $B(\alpha)$, puis les coefficients $b_{i,j}$ à l'aide de division euclidienne dans $\mathbb{Q}[T]$.

2.4.5 Description de l'algorithme

Algorithme Appexact2

ENTRÉES : $Q \in \mathbb{Q}[X,Y]$ irréducible dans $\mathbb{Q}[X,Y]$, unitaire en Y, *facto-app* un algorithme de factorisation approchée.

SORTIES : $Q_1(X,Y) \in \mathbb{Q}[\alpha][X,Y]$ un facteur exact de Q, et $f_\alpha(T) \in \mathbb{Z}[T]$ le polynôme minimal de α.

1. Se ramener au cas $P(X,Y) \in \mathbb{Z}[X,Y]$.
2. Effectuer *facto-app*(P) la factorisation approchée de P avec un nombre de chiffres significatifs $= Digits$.
3. Calculer $Digits1$ et $Digits2$ (voir les propositions 15 et 18) si $\max(Digits1, Digits2) \geq Digits$ alors recommencer l'étape précédente avec $Digits = \max(Digits1, Digits2)$, sinon continuer.
4. Reconnaître tous les coefficients de P_1 qui sont des éléments primitifs, leur polynôme minimal, leur discriminant.
   (Si aucun des coefficients n'est primitif alors : en fabriquer 11 (par exemple), reconnaître leur polynôme minimal puis leur discriminant.)
5. Trouver l'élément primitif α ayant le plus petit discriminant et noter f_α son polynôme minimal.
6. Reconnaissance des coefficients exacts de P_1 à l'aide du système de Vandermonde $(\star)$, voir page 61.

7. Exprimer grâce au produit matriciel vu dans le lemme 21 les coefficients de P_1 sous la forme $q_0 + q_1\alpha + \ldots + q_{s-1}\alpha^{s-1}$ où $q_i \in \mathbb{Q}$.
8. Retourner le facteur de Q correspondant.

Remarque :
Dans cet algorithme il n'est pas nécessaire de vérifier que le polynôme obtenu à l'étape 6 divise le polynôme P.

2.5 Commentaires

2.5.1 Choix de l'élément primitif

Dans les algorithmes présentés nous avons privilégié la reconnaissance d'un coefficient comme élément primitif plutôt que la construction d'un élément primitif. Nous allons ici justifier notre choix à l'aide d'exemples.

Nous allons donc présenter quelques exemples de calculs de "dénominateur commun" .

Chaque exemple a été construit de la façon suivante : Nous nous donnons un polynôme $P \in \mathbb{Z}[X,Y]$ irréductible dans $\mathbb{Q}[X,Y]$ ainsi qu'une expression approchée de ses facteurs P_i irréductibles dans $\mathbb{C}[X,Y]$. Nous noterons s le nombre des facteurs P_i et m leur degré en Y.

A partir de cela nous calculons les discriminants de tous les coefficients de P_1 qui sont primitifs. Nous noterons $disc(\alpha_1)$ le plus petit discriminant obtenu par ce procédé, et, $[disc(\alpha_1)]_h$ le "dénominateur commun" obtenu avec l'heuristique.

Ensuite nous fabriquons 11 éléments primitifs à l'aide de la méthode décrite en 2.2.2. Puis nous calculons les discriminants de ces éléments. Nous noterons $disc(\alpha_2)$ le plus petit discriminant obtenu par ce procédé, et, $[disc(\alpha_2)]_h$ le "dénominateur commun" obtenu avec l'heuristique.

Les résultats obtenus étant des nombres entiers pouvant avoir plus de 100 chiffres dans leur écriture décimale, nous ne donnerons pas dans le tableau suivant les entiers obtenus mais leur "longueur". C'est-à-dire ce que nous notons $\sharp\ disc(\alpha_1)$ représente le nombre de chiffres de $disc(\alpha_1)$ dans son écriture décimale.

Remarques :
La réduction à l'aide de l'heuristique est très efficace.
Il est préférable de choisir comme générateur de l'extension un coefficient qui soit un élément primitif plutôt que de fabriquer un élément primitif. En effet nous voyons dans le tableau que $\sharp\ disc(\alpha_1) < \sharp\ disc(\alpha_2)$. Cela entraîne $\sharp\ [disc(\alpha_1)]_h < \sharp\ [disc(\alpha_2)]_h$. La différence entre $\sharp\ [disc(\alpha_1)]_h <$ et $\sharp\ [disc(\alpha_2)]_h$ peut être énorme (voir l'exemple $m = 2$, $s = 10$). Nous avons même $\sharp\ [disc(\alpha_2)]_h$ qui est plus grand que $disc(\alpha_1)$. Cela signifie que même après des simplifications le "dénominateur commun" d'un élément primitif fabriqué est plus grand que le discriminant d'un

(m, s)	$\sharp disc(\alpha_1)$	$\sharp[disc(\alpha_1)]_h$	$\sharp disc(\alpha_2)$	$\sharp[disc(\alpha_2)]_h$
(3,9)	36	8	134	57
(3,7)	14	1	99	43
(3,5)	11	2	30	12
(3,9)	42	11	149	65
(3,9)	51	16	182	82
(2,10)	23	1	138	58
(5,5)	14	4	53	23
(4,9)	45	15	183	84
(10,2)	3	1	12	5
(10,4)	18	6	76	35

TAB. 2.1 – Reconnaître ou construire un élément primitif ?

coefficient.

Ainsi comme nous voulons avoir le plus petit “dénominateur commun” possible nous préférons prendre comme élément primitif un coefficient plutôt qu'un élément primitif fabriqué au hasard.

2.5.2 Comparaison de l'heuristique avec l'algorithme Zassenhauss'Round 4

Nous allons ici comparer notre heuristique pour le calcul de D_α avec l'algorithme de Zassenhauss'round 4. Nous avons vu que l'algorithme de Zassenhauss nous donne une base entière, c'est ce qui nous permet d'obtenir le dénominateur D_α. Nous avons programmé nos différents algorithmes en Maple, et les tableaux suivants présentent certains résultats relatifs à la première méthode. Tous les exemples qui suivent ont été traités par le même ordinateur.

Nous avons construit plusieurs exemples, m est le degré de P_1 et s est le nombre de facteurs obtenus. Nous avons alors calculé le discriminant pour chaque coefficient de P_1. Les valeurs obtenues étant trop grandes pour figurer dans un tableau, nous avons noté $\sharp disc(\alpha)$ le nombre de chiffres dans l'écriture décimale du plus petit discriminant non nul trouvé. Ensuite nous calculons le dénominateur d calculé en utilisant notre heuristique. Nous notons $\sharp d$ et le nombre de chiffres dans l'écriture décimale de d.

Ici t est le temps en secondes nécessaire pour calculer tous les discriminants et le dénominateur d.

T est le temps en secondes nécessaire pour le calcul d'une base entière avec l'algorithme Zassenhauss'Round 4 (toujours avec Maple). La donnée pour l'algorithme Zassenhauss'Round 4 est le polynôme minimal du coefficient de P_1 qui a le plus petit discriminant non nul. A partir de cette base entière nous calculons D_α, et $\sharp D_\alpha$

désigne le nombre de chiffre significatif dans l'écriture décimale de D_α.

Dans le tableau suivant, la première colonne indique le numéro de l'exemple.

EX	(m,s)	$\sharp disc(\alpha)$	$\sharp D_\alpha$	$\sharp d$	t	T
1	(2,10)	38	1	1	0.281	0.320
2	(2,10)	39	2	1	0.160	10.401
3	(3,9)	72	19	17	0.420	112.720
4	(3,9)	66	19	17	0.331	1.840
5	(3,9)	66	18	17	0.399	8.341
6	(3,8)	55	14	14	0.461	2.269
7	(4,5)	28	7	6	0.281	0.939
8	(4,7)	49	14	13	0.411	2.170
9	(5,4)	18	5	5	0.340	0.421
10	(5,5)	30	7	7	0.340	0.920
11	(5,5)	28	7	7	0.400	0.330
12	(6,4)	23	6	3	0.451	1.430
13	(7,3)	13	3	3	0.290	0.150

TAB. 2.2 – Comparaison de l'heuristique avec Zassenhauss'Round 4

REMARQUES

Dans aucun exemple l'heuristique n'est mise en défaut, c'est-à-dire l'algorithme APPEXACT1 rend le facteur P_1 dès le premier essai.

Notre méthode ne fournit pas le dénominateur le plus petit possible. En effet après simplification des fractions q_i dans les expressions des coefficients de P_1 : $q_0 + q_1\alpha + \ldots + q_{s-1}\alpha^{s-1}$ nous obtenons des dénominateurs encore plus petits que celui donné par l'heuristique. C'est-à-dire que sur tous les exemples nous pouvons prendre comme dénominateur un diviseur strict de d.

Dans l'algorithme de Zassenhauss, T dépend du temps nécessaire pour factoriser $disc(\alpha)$. Comme $\mid \delta_K \mid \geq \frac{\pi}{3}\left(\frac{3\pi}{4}\right)^{s-1}$ pour $2 \leq s$ (voir [Sam67] p.70), lorsque s est grand l'algorithme de Zassenhauss peut être lent, voir les exemples 2 et 3. Le temps nécessaire pour obtenir d avec notre heuristique est plus régulier que le temps nécessaire pour factoriser $disc(\alpha)$.

2.5.3 Comparaison des deux méthodes

Nous avons donné deux méthodes pour obtenir une factorisation exacte. La différence fondamentale qui existe entre celles-ci est la suivante : la première méthode nécessite une étape finale de vérification (P_1 divise P), ce qui n'est pas le cas dans la seconde où les calculs sont effectués avec une précision suffisante. D'autre part dans la seconde méthode nous devons réexprimer les coefficients de P_1 de manière canonique.

Donc le choix d'une méthode dépend de la rapidité à faire la vérification : P_1 divise P, ou à faire des multiplications exactes du type matrice-vecteurs.

Afin de donner une idée des temps de calculs en Maple nous donnons le tableau suivant. La première colonne nous donne les paramètres m et s, la deuxième le temps nécessaire à la première méthode pour rendre P_1, la troisième le temps nécessaire à la vérification P_1 divise P, la quatrième le temps nécessaire à la seconde méthode pour rendre P_1.

TAB. 2.3 – Comparaison des temps des deux méthodes

(m, s)	temps méthode 1	Temps vérif.	temps méthode 2
(8,8)	507.7	502.5	23.2
(8,8)	422.4	417.2	12.6
(8,9)	908.6	901.6	14.5
(8,9)	1227.8	1220.3	21.0
(4,5)	5.55	4.68	1.57
(3,9)	62.2	60.8	3.18
(3,9)	39.04	37.65	3.17
(6,4)	7.00	5.98	1.68

REMARQUES :
Dans tous ces exemples la première méthode à donné le bon résultat dès le premier essai. Cela signifie donc que notre heuristique est vérifiée en pratique.
La première méthode est beaucoup plus lente que la seconde mais cela est dû uniquement à l'étape de vérification. En effet si l'on regarde la première ligne on constate que P_1 a été obtenu en environ 5 secondes.

2.5.4 Une étude théorique donnant a priori la précision nécessaire pour l'algorithme

Les minorations obtenues dans les propositions 15 (page 53) et 18 (page 65) ne dépendent que de s qui est le nombre de facteurs obtenus et de M qui est un majorant de la valeur absolue des coefficients de P_i. Comme s est majoré par n (le degré du polynôme P), si nous exprimons M à partir des coefficients du polynôme à factoriser P, alors nous pourrons dire quelle est la précision nécessaire dans les calculs. Cela éviterait de faire une première factorisation approchée avec un nombre de chiffres significatifs trop faible, et d'être ensuite obligé de recommencer la factorisation approchée avec un nombre de chiffres significatifs suffisant.

A présent nous allons montrer comment majorer la norme des coefficients de P_i en fonction des coefficients de P.

On se donne $P(X,Y) \in \mathbb{Z}[X,Y]$ et $P_i(X,Y) \in \mathbb{C}[X,Y]$ tels que :

$$\begin{aligned}
P(X,Y) &= Y^n + (a_{1,n-1}X + a_{0,n-1})Y^{n-1} + \ldots + a_{n,0}X^n + \ldots + a_{0,0},\\
P_i(X,Y) &= Y^m + a_i^{(1,m-1)}XY^{m-1} + a_i^{(0,m-1)}Y^{m-1} + \ldots + a_i^{(m,0)}X^m + \ldots + a_i^{(0,0)}.
\end{aligned}$$

Nous avons défini en 1.4.1 page 18 la hauteur d'un polynôme, et certaines de ses propriétés. Dans notre cas en utilisant la proposition 6, nous avons

$$\prod_{i=1}^{s} \|P_i\|_\infty \leq 2^{2sm}\|P\|_\infty.$$

Du fait que les P_i sont unitaires nous obtenons

$$(*) \quad \|P_i\|_\infty \leq 2^{2sm}\|P\|_\infty.$$

Il existe d'autres majorations de $H(P_i)$. En effet en utilisant les formules données dans [Sch00], et dans [MŞ99] page 92, nous obtenons :

$$(**) \quad \|P_i\|_\infty \leq 2^{2m}\|P\|_2,$$

où $\|P\|_2 = \sqrt{\sum a_{i,j}^2}$.

Ainsi nous pouvons remplacer M, dans les propositions 15 et 18, par $2^{2sm}\|P\|_\infty$, ou bien, par $2^{2m}\|P\|_2$.

Nous avons alors l'expression de la précision nécessaire pour obtenir une factorisation exacte à partir d'une factorisation approchée en fonction des coefficients et du degré de P.

2.5.5 Conclusion

L'algorithme présenté ici est une amélioration d'une méthode existante (voir [Rup00]).

Celle-ci reconnaissait le polynôme minimal de l'extension par une méthode de fractions continues. Le théorème suivant (voir [HW79]) nous dit quelle précision sur les coefficients était nécessaire pour réussir.

Théorème 23. *Soit x un réel positif non nul et soit $\frac{n}{d}$ une fraction irréductible telle que :*

$$\mid x - \frac{n}{d} \mid < \frac{1}{2d^2}$$

Alors $\frac{n}{d}$ est une des réduites du développement en fraction continue de x.

Or deux problèmes se posaient la réduction au cas $\mathbb{Z}[X,Y]$ n'ayant pas été réalisée, le dénominateur d était inconnu. De plus ce dénominateur pouvait être supérieur à 1, contrairement à la méthode exposée ici. Donc nous sommes passés d'une exigence sur la précision des coefficients du polynôme minimal de $1/2d^2$ avec d inconnu à $0,5$ près.
D'autre part, pour reconnaître le polynôme minimal nous devons pour chaque coefficient de $f_{\alpha+\epsilon}$ prendre l'entier le plus proche. Cela évite de calculer le développement d'une fraction continue et donc des calculs de pgcd.

Une autre étape de l'algorithme de Rupprecht consistait à trouver une racine d'un polynôme dans une extension, et cela est à présent remplacé par la résolution d'un système linéaire. Cette amélioration était déjà suggérée dans [Rup00], mais une nouvelle fois la reconnaissance des coefficients devait être effectuée avec des fractions continues. Pour mener à bien cette étape il aurait fallu obtenir une erreur $\mid e \mid < 1/2d^2$ sur la solution (avec d inconnu). Tandis qu'à présent nous avons calculé dans la première méthode d et nous nous sommes à une situation où une erreur $\mid e \mid < 1/2d$ est suffisante pour reconnaître les coefficients.

D'autre part nous avons donné une autre méthode qui passe par une représentation où le dénominateur n'est pas entier. Cependant le numérateur est un nombre entier que nous pouvons reconnaître et cela de manière certifiée. Cette démarche évite l'étape de vérification : P_1 divise P, et est très efficace en pratique.

Chapitre 3

Un algorithme symbolique-numérique de factorisation absolue

Ce chapitre est une version étendue de l'article [Chè04b].

Nous avons présenté dans la section 1.5.3 l'algorithme Galligo/Rupprecht. Dans cet algorithme nous devons trouver une somme nulle minimale entre des nombres connus de manière approchée. D. Rupprecht a montré que nous pouvions effectuer cette tâche avec une complexité en $O(2^{n/4})$ où n est le degré du polynôme. Avec cette méthode nous pouvons obtenir la factorisation absolue de polynômes de degré inférieur à 60. Afin de dépasser cette limite nous allons voir comment obtenir les sommes nulles en utilisant l'algorithme LLL. La stratégie utilisée est inspirée de celle développée par M. van Hoeij pour la factorisation dans $\mathbb{Z}[X]$. Nous utiliserons donc l'algorithme LLL ce qui nous permettra de factoriser des polynômes de degré 200.
Dans la première section de ce chapitre nous donnerons un théorème généralisant le théorème 20 page 34. Cela nous permettra de rechercher des relations entières entre les b_i et non plus des sommes nulles. Ensuite dans la section 2 nous verrons comment utiliser LLL dans notre situation. Puis dans la section 3 nous décrirons l'algorithme. Les trois dernières sections traiteront de la précision utilisée dans les calculs, des améliorations possibles pour notre algorithme, et des temps de calculs obtenus avec cette nouvelle méthode.

Notations : Comme en 1.5.3 nous supposons $P(X,Y)$ irréductible dans $\mathbb{Q}[X,Y]$ et sans facteurs linéaires. De plus les notations $y_i(x_0)$, $a_i(x_0)$, $b_i(x_0)$, $\varphi_i(X)$ et $b(X,Y)$ sont conservées. Nous rappelons que les facteurs absolument irréductibles de P sont notés $P_1,\ldots,P_s$ et qu'ils sont conjugués sur $\mathbb{Q}$.

3.1 Relations entières entre les b_i

Nous avons vu que les sommes nulles entre les $b_i(x_0)$ permettent de caractériser les facteurs de P. Nous allons voir que les relations bornées le permettent également.

Définition 24. *Soit M une constante positive. Une relation M-bornée entre les nombres $b_i(x_0)$ est une combinaison linéaire à coefficients dans $\mathbb{Z}$ telle que :*

$$\sum_{i\in I} \lambda_i b_i(x_0) = 0 \text{ où } |\lambda_i| \leq M \text{ pour tout } i.$$

Nous pouvons à présent généraliser aux relations M-bornées le théorème 20.

Théorème 24. *Soient M une constante positive, P un polynôme irréductible de $\mathbb{Q}[X,Y]$ de degré total n et tel que $\deg(P) = \deg_Y(P)$. Soit $f_\lambda(X,Y) = P(X+\lambda Y, Y)$. Nous avons alors :*
Pour presque toutes les spécialisations (x_0, λ_0) de (X, λ) les seules relations possibles M-bornées existant entre les $b_i(x_0)$ sont du type :

$$\sum_{i\in I_1} c_1 b_i(x_0) + \ldots + \sum_{i\in I_s} c_s b_i(x_0) = 0,$$

où I_k est un ensemble d'indices tel que $\prod_{i\in I_k}(Y - \varphi_i(X))$ soit un polynôme divisant P, et, où $c_k \in \mathbb{Z}$.

Démonstration. Tout d'abord nous choisissons un paramètre λ_0 tel que le théorème d'Harris affine soit vérifié et tel que $b_i(x_0) \neq b_j(x_0)$ pour $i \neq j$. Cela est possible d'après le lemme 8 page 35.
Nous considérons donc à présent le polynôme f_{λ_0}, mais nous continuerons pour plus de simplicité dans les notations à écrire $y_i(x_0)$, $b_i(x_0)$, $\varphi_i(X)$, $b(X,Y)$ et P_i. De plus nous pouvons supposer sans perte de généralité que les séries $\varphi_i(X)$ sont ordonnées de telle sorte que $P_k(X,Y) = \prod_{i\in\{m(k-1)+1,\ldots,mk\}}(Y - \varphi_i(X))$. Cela signifie que $I_k = \{m(k-1)+1, \ldots, mk\}$.
Ensuite nous considérons pour une combinaison linéaire fixée $\sum_i \lambda_i b_i(x_0)$ le produit :

$$\mathcal{B}_{(\lambda_1,\ldots,\lambda_n)}(X) = \prod_{(\sigma_1,\ldots,\sigma_s)\in\mathfrak{S}_m^s} \left[\sum_{i=1}^{m} \lambda_i b(x_0, \sigma_1(\varphi_i(X)) + \ldots + \sum_{i=n-m+1}^{n} \lambda_i b(x_0, \sigma_s(\varphi_i(X))\right].$$

On note que $\mathcal{B}_{(\lambda_1,\ldots,\lambda_n)} \in \mathbb{C}(X)$ car c'est une fonction symétrique par rapport à $(\varphi_1(X), \ldots, \varphi_m(X))$,...,$(\varphi_{n-m+1}(X), \ldots, \varphi_i(X))$, qui sont les racines des facteurs $P_i(X,Y) \in \mathbb{C}[X,Y]$.
A présent nous pouvons montrer le théorème, c'est-à-dire : pour presque tous les $x_0 \in \mathbb{C}$ si nous avons une relation linéaire $\sum_i \lambda_i b_i(x_0) = 0$ alors $\lambda_i = \lambda_j$ lorsque i et j appartiennent au même ensemble d'indice I_k.
Supposons alors que pour une relation M-bornée nous ayons $\lambda_1 \neq \lambda_2$ avec

$\{1, 2\} \subset I_1$. Nous allons montrer que $\mathcal{B}_{(\lambda_1,\dots,\lambda_n)}(X) \neq 0$ dans $\mathbb{C}(X)$. Donc pour presque tous les $x_0 \in \mathbb{C}$ une telle combinaison linéaire ne donne pas 0.
Nous allons effectuer un raisonnement par l'absurde :
Supposons $\mathcal{B}_{(\lambda_1,\dots,\lambda_n)} = 0$ dans $\mathbb{C}(X)$ alors à l'aide du théorème d'Harris affine avec $\sigma_1 = (1 \quad 2)$ nous obtenons (comme dans la preuve du théorème 19) : $\lambda_1(b_1(x_0) - b_2(x_0)) + \lambda_2(b_2(x_0) - b_1(x_0)) = 0$.
D'où

$$\frac{\lambda_1}{\lambda_2} = \frac{b_1(x_0) - b_2(x_0)}{b_1(x_0) - b_2(x_0)} = 1$$

car $b_i(x_0) \neq b_j(x_0)$. Ce qui contredit $\lambda_1 \neq \lambda_2$, donc $\mathcal{B}_{(\lambda_1,\dots,\lambda_n)} \neq 0$. □

Comme nous considérons des relations avec $|\lambda_i| \leq M$ nous avons un nombre fini de "mauvaises" relations à éviter. Les mauvais choix pour x_0 et λ_0 se trouvant sur des courbes de $\mathbb{C}^2$, une union finie de tels mauvais choix reste négligeable dans $\mathbb{C}^2$. De plus l'ensemble de ces mauvais choix n'est pas dense. En effet nous voulons éviter une situation du type $(\mathbb{Q} + i\mathbb{Q}) \times (\mathbb{Q} + i\mathbb{Q})$ qui est négligeable mais dense dans $\mathbb{C}^2$.

3.2 Comment obtenir les sommes nulles avec LLL

3.2.1 Premières remarques

Dans tout le reste de ce chapitre nous considérerons x_0 réel. En effet lorsque x_0 est réel nous savons que si β est l'un des b_i alors $\overline{\beta}$ (le conjugué de β) sera aussi un des b_i.

Notre stratégie est de trouver dans un premier temps toutes les sommes nulles entre les $\Re(b_i)$, puis dans un deuxième temps d'en déduire les sommes nulles entre les b_i. (Cela correspond à d'abord trouver les facteurs réels puis à les "découper" pour obtenir les facteurs complexes.) Pour cela nous introduisons les notations suivantes :

- $\{1, \dots, n\} = \sqcup_{i=1}^{s} I_i$ est la partition correspondant aux sommes minimales, c'est-à-dire : $\prod_{j \in I_j}(Y - \varphi_j(X))$ est un facteur absolument irréductible de P.
- $I_i^{\mathbb{R}} = \{j \in I_i \,|\, b_j \in \mathbb{R}\}$,
- $I_i^{\mathbb{C}+} = \{j \in I_i \,|\, \Im(b_j) > 0\}$,
- $I_i^{\mathbb{C}-} = \{j \in I_i \,|\, \Im(b_j) < 0\}$, où $\Im(b_j)$ est la partie imaginaire de b_j.

On remarque que : $\sum_{I_k} b_j = 0$ implique $\sum_{I_k} \overline{b_j} = 0$. Donc $\sum_{I_k} \overline{b_j} = \sum_{I_{k'}} b_j$ où $I_{k'} = \{i \in \{1, \dots, n\} \,|\, b_i = \overline{b_j} \text{ où } j \in I_k\}$.

Comme nous avons une partition soit $I_k = I_{k'}$, soit $I_k \cap I_{k'} = \emptyset$.
- Si $I_k = I_{k'}$ alors $\sum_{I_k} b_j = \sum_{I_k^{\mathbb{R}}} b_j + \sum_{I_k^{\mathbb{C}+}} 2\Re(b_j)$.

– Si $I_k \cap I_{k'} = \emptyset$ alors b_j (pour $j \in I_k$) n'est pas réel et $\sum_{I_k} b_j + \sum_{I_{k'}} b_j = \sum_{I_k^{\mathbb{C}+} \sqcup I_{k'}^{\mathbb{C}+}} 2\Re(b_j) = 0$.

Ainsi si nous avons une somme nulle entre les b_i nous avons une somme nulle entre les $\Re(b_i)$. Cette "astuce" nous permet de rechercher, dans un premier temps, des sommes nulles parmi $\frac{n+l}{2}$ nombres réels où l est le nombre de $b_i \in \mathbb{R}$. Ce nombre l est, en général, très petit comparé à n ($l << n$), voir théorème 25 page 89.

Nous rappelons qu'en pratique nous n'avons qu'une approximation $\tilde{y}_i$ de y_i, donc nous ne pouvons calculer qu'une approximation $\tilde{b}_i$ de b_i. Nous avons donc $\tilde{b}_i = b_i + \eta_i$, où η_i est l'erreur. Nous posons alors : $\eta = \max_{i=1,\ldots,n} |\eta_i|$. Il vient donc $|\tilde{b}_i - b_i| < \eta$. On rappelle que $\lfloor x \rceil$ désigne l'entier le plus proche du nombre réel x. Ainsi pour $C \in \mathbb{Z}$ nous avons : $|\lfloor C\Re(\tilde{b}_i) \rceil - C\Re(b_i)| \leq \frac{1}{2} + C\eta$

3.2.2 Sommes réelles nulles

Dans cette partie nous allons expliquer comment obtenir les sommes de $\Re(b_i)$ nulles et minimales. Ici minimale signifie que la somme ne peut être décomposée en somme de 2 termes nuls.

Quelques notations

Nous ordonnons les nombres b_i de la manière suivante : les premiers $b_1, \ldots, b_l$ sont réels, $b_{l+1}, \ldots, b_{\frac{n+l}{2}}$ sont tels que $\Im(b_i) > 0$ et $b_{\frac{l+n}{2}+1}, \ldots, b_n$ sont tels que $\Im(b_i) < 0$ et vérifient $b_{\frac{n+l}{2}+i} = \overline{b_{l+i}}$ $(1 \leq i \leq \frac{n+l}{2} - l)$.
A présent nous considérons $\Re(b_1), \ldots, \Re(b_l)$, $2\Re(b_{l+1}), \ldots, 2\Re(b_{\frac{l+n}{2}})$, et nous posons $\epsilon_i\Re(b_i)$ $(1 \leq i \leq \frac{n+l}{2})$ pour les nombres réels vérifiant : $\epsilon_i\Re(b_i) = \Re(b_i)$ si $1 \leq i \leq l$, et $\epsilon_i\Re(b_i) = 2\Re(b_i)$ si $l+1 \leq i \leq \frac{n+l}{2}$.
Une somme nulle minimale pour les $\epsilon_i\Re(b_i)$ est une somme $\sum_{i\in I} \epsilon_i\Re(b_i)$ où $\sum_{i \in J \subsetneq I} \epsilon_i\Re(b_i) \neq 0$.

Une telle somme est de la forme suivante : $\sum_{i=1}^{\frac{n+l}{2}} x_i\epsilon_i\Re(b_i)$ où $x_i = 0$ ou 1. On note alors $\overrightarrow{\epsilon\Re(b)}$ le vecteur de $\mathbb{R}^{\frac{n+l}{2}}$ qui a pour $i^{\text{ème}}$ coordonnées $\epsilon_i\Re(b_i)$. Donc une somme nulle correspond à un 0-1 vecteur $v_i \in \mathbb{Z}^{\frac{n+l}{2}}$ tel que $< v_i, \overrightarrow{\epsilon\Re(b)} >= 0$. Comme en 1.3.2 page 16, nous appelons 0-1 vecteur un vecteur appartenant à $\{0,1\}^{\frac{n+l}{2}}$. On a alors la définition :

Définition 25. *On appelle V le sous-réseau de $\mathbb{Z}^{\frac{n+l}{2}}$ engendré par les 0-1 vecteurs linéairement indépendants $v_1, \ldots, v_t$ correspondant aux sommes nulles minimales pour les $\epsilon_i\Re(b_i)$.*

La stratégie

Nous voulons calculer une base $v_1, \ldots, v_t$ de V. Nous allons suivre la stratégie développée dans [vH02] : Nous allons construire une suite de réseaux $\mathcal{L}_i$ vérifiant :

$$\begin{cases} V \subset \mathcal{L}_i \subset \mathcal{L}_{i+1} \subset \mathbb{Z}^{\frac{n+l}{2}}, \\ \text{Il existe un entier } k \text{ tel que pour tout } i \geq k \text{ nous avons : } \mathcal{L}_i = \mathcal{L}_k = V. \end{cases}$$

Nous initialisons notre suite avec $\mathcal{L}_0 = \mathbb{Z}^{\frac{n+l}{2}}$. A présent nous allons expliquer comment obtenir $\mathcal{L}_{i+1}$ à partir de $\mathcal{L}_i$.
Soit $\mathcal{L}$ un réseau engendré par $w_1, \ldots, w_k$ où $w_i \in \mathbb{Z}^{\frac{n+l}{2}}$ tel que $V \subset \mathcal{L}$. Soit $p_{\mathcal{L}}$ le morphisme de $\mathbb{Z}$-module suivant :

$$\begin{array}{rcl} p_{\mathcal{L}} : \mathcal{L} & \longrightarrow & \mathcal{L}' \\ w_i = (w_{i,1}, \ldots, w_{i,\frac{n+l}{2}}) & \longmapsto & (w_{i,1}, \ldots, w_{i,\frac{n+l}{2}}, < w_i, \overrightarrow{\lfloor C\epsilon\Re(\tilde{b})\rceil} >) \end{array}$$

où $\overrightarrow{\lfloor C\epsilon\Re(\tilde{b})\rceil}$ est le vecteur qui a pour $i^{\text{ème}}$ coordonnées $\lfloor C\epsilon_i\Re(\tilde{b}_i)\rceil$.

Lemme 22. *Soient $\mathcal{L}$ un réseau contenant V, $\mathcal{L}' = p_{\mathcal{L}}(\mathcal{L})$, $\nu_1, \ldots, \nu_k$ une base de $\mathcal{L}'$ et k_0 un entier tel que :*
$\forall i > k_0$, $\|\nu_i^*\| > M$ où $M = \sqrt{\frac{n+l}{2} + [(\frac{1}{2} + 2C\eta).\frac{n+l}{2}]^2}$.
Alors le réseau engendré par $p_{\mathcal{L}}^{-1}(\nu_1), \ldots, p_{\mathcal{L}}^{-1}(\nu_{k_0})$ contient V.

Pour pouvoir appliquer ce lemme il faut que les premiers vecteurs de base aient une petite norme. Nous utiliserons donc en pratique des bases LLL réduites afin de réaliser les hypothèses de ce lemme. Pour démontrer ce lemme nous allons avoir besoin du résultat suivant :

Lemme 23 (Nancy's lemma). *Soient $b_1, \ldots, b_n$ une base d'un réseau $\mathcal{L} \subset \mathbb{R}^n$ et $b_1^*, \ldots, b_n^*$ la base correspondante obtenue par le procédé d'orthogonalisation de Gram-Schmidt. Soit k_0 un entier vérifiant : $\forall i > k_0$, $\|b_i^*\| > B$. On a alors :*

$$\{x \in \mathcal{L} \,|\, \|x\| \leq B\} \subset Vect_{\mathbb{Z}}(b_1, \ldots, b_{k_0}).$$

Démonstration. Nous allons effectuer un raisonnement par l'absurde. Supposons $\{x \in \mathcal{L} \,|\, \|x\| \leq B\} \not\subset Vect_{\mathbb{Z}}(b_1, \ldots, b_{k_0})$. Dans ce cas il existe $x \in \mathcal{L}$ tel que $\|x\| \leq B$, et, $x \notin Vect_{\mathbb{Z}}(b_1, \ldots, b_{k_0})$. On peut alors écrire $x = \sum_{i=1}^{l} r_i b_i$, avec $r_i \in \mathbb{Z}$, $n \geq l > k_0$ et $r_l \neq 0$. D'où $x = r_l b_l^* + \sum_{i=1}^{l-1} \lambda_i b_i^*$. (Nous utilisons ici le fait que la matrice de passage de $b_1, \ldots, b_n$ à $b_1^*, \ldots, b_n^*$ est triangulaire avec des 1 sur la diagonale.) Cela donne : $\|x\|^2 \geq r_l^2 \|b_l^*\|^2 > B^2$, ce qui est absurde. □

Démonstration. Lemme 22.
Nous savons que les 0-1 générateurs v_i de V ont au plus $n + l/2$ coordonnées égales à 1. D'où

$$| < v_i, \overrightarrow{\lfloor C\epsilon\Re(\tilde{b})\rceil} > | \leq (\frac{1}{2} + \epsilon C\eta)\frac{n+l}{2}.$$

Donc $\|p_{\mathcal{L}}(v_i)\| \leq M$. Par conséquent :

$$\{p_{\mathcal{L}}(v_1), \ldots, p_{\mathcal{L}}(v_t)\} \subset \{x \in \mathcal{L} \,|\, \|x\| \leq M\}.$$

Le lemme 23 implique :

$$\{p_{\mathcal{L}}(v_1), \ldots, p_{\mathcal{L}}(v_t)\} \subset Vect_{\mathbb{Z}}(\nu_1, \ldots, \nu_{k_0}).$$

En prenant l'image inverse par $p_{\mathcal{L}}$ nous obtenons le résultat désiré. □

Le lemme suivant donne une condition nécessaire et suffisante sur $\mathcal{L}$ pour arrêter le calcul de la suite $(\mathcal{L}_i)_i$. Ce lemme est très proche du lemme 2.8 dans [vH02]. Pour l'énoncer nous avons besoin de quelques notations :

Notations :
Si $\mathcal{L}$ est un réseau alors nous notons $\mathcal{B}_{\mathcal{L}}$ une base de ce réseau. La matrice dont les lignes sont les vecteurs de $\mathcal{B}_{\mathcal{L}}$ est notée $(\mathcal{B}_{\mathcal{L}})$, et la forme échelonnée réduite de cette matrice est notée $RREF(\mathcal{B}_{\mathcal{L}})$.

Lemme 24 (Test d'arret). *Soit $V \subset \mathcal{L}$ on a alors l'équivalence suivante : $RREF(\mathcal{B}_V) = RREF(\mathcal{B}_{\mathcal{L}}) \iff RREF(\mathcal{B}_{\mathcal{L}})$ est telle que chaque colonne contient un 1 et que des 0 ailleurs, et, chaque ligne permet d'obtenir un facteur de P.*

Démonstration. $\Rightarrow$) Nous considérons les 0-1 vecteurs $v_1, \ldots, v_t$ engendrant V. Ces vecteurs forment une base $\mathcal{B}_V = \{v_1, \ldots, v_t\}$. Cette base est déjà sous forme échelonnée réduite et chaque colonne de $(\mathcal{B}_V)$ contient précisément un 1 et que des 0. En effet les 0-1 vecteurs v_i proviennent de la partition correspondant aux sommes minimales sur les $\epsilon_i b_i$. De plus chaque ligne de $RREF(\mathcal{B}_V)$ correspond à un vecteur v_i, et nous permet donc d'obtenir un facteur de P. Ainsi si $RREF(\mathcal{B}_V) = RREF(\mathcal{B}_{\mathcal{L}})$ alors le membre de droite de l'équivalence est vérifié.

$\Leftarrow$) Si $RREF(\mathcal{B}_V) \neq RREF(\mathcal{B}_{\mathcal{L}})$ alors nous devons montrer que soit $RREF(\mathcal{B}_{\mathcal{L}})$ n'a pas un 1 par colonne et que des 0 ailleurs, soit les polynômes obtenus à partir de $RREF(\mathcal{B}_{\mathcal{L}})$ ne divisent pas P. Supposons que $RREF(\mathcal{B}_{\mathcal{L}})$ possède un 1 par colonne et que des 0 ailleurs, et $V \subset \mathcal{L}$. Nous devons alors montrer qu'une ligne de $RREF(\mathcal{B}_{\mathcal{L}}$ ne correspond pas à un facteur de P.
Nous notons alors $w_1, \ldots w_k$ une base de $\mathcal{L}$ et $v_1, \ldots v_t$ les 0-1 vecteurs de la base engendrant V. Nous avons alors

$$RREF(\mathcal{B}_{\mathcal{L}}) = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} = \mathcal{M} \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix}$$

où $\mathcal{M} \in Gl_k(\mathbb{Q})$ et les z_i sont des 0-1 vecteurs linéairement indépendants. Nous devons donc montrer qu'un vecteur z_i ne fournit pas un facteur de P.
Comme $V \subset \mathcal{L}$ nous avons $v_i = \sum_{j=1}^{k} \lambda_j w_j = \sum_{j=1}^{k} \mu_j z_j$ où $\lambda_j \in \mathbb{Z}$, et $\mu_j \in \mathbb{Q}$. Du

fait que v_i et z_j sont des 0-1 vecteurs sous forme échelonnée nous obtenons $\mu_j = 1$ ou $\mu_j = 0$.
Nous pouvons donc écrire : $v_i = \sum_{j=j_1}^{j_{k'}} z_j$. Si pour chaque i nous avons l'égalité $v_i = z_{j_i}$ (i.e. $k' = 1$) alors il vient $RREF(\mathcal{B}_V) = RREF(\mathcal{B}_\mathcal{L})$ ce qui contredit notre hypothèse. D'où $t < k$. Donc les 0-1 vecteurs z_i donnent plus de facteurs réels qu'il n'y a de facteurs réels irréductibles. Nous aboutissons donc à une contradiction, ce qui prouve le résultat annoncé. $\square$

La condition "chaque ligne permet d'obtenir un facteur de P" n'est pas difficile à vérifier. En effet, en pratique nous serons dans une situation où tous les facteurs seront conjugués il suffira donc de tester si un polynôme correspondant à une ligne divise P.
En pratique nous allons donc fabriquer une suite de réseaux. Nous obtiendrons $\mathcal{L}_{i+1}$ en supprimant les vecteurs qui ont une trop grande norme comme expliqué dans le lemme 22. Pour chaque nouveau réseau construit, nous calculons sa forme échelonnée réduite. Si celle-ci possède un 1 par colonne et que des 0 ailleurs alors nous essayons de retrouver les facteurs. En pratique nous pourrons tester si cette décomposition est valable si tous les polynômes obtenus ont le même degré et en calculant $\sum_I b_i$. En effet à l'aide d'une transformation approchée-exacte nous pourrons tester si ce nombre $\sum_I b_i$ est égal à 0 ou non. Nous pouvons donc tester si $RREF(\mathcal{B}_V) = RREF(\mathcal{B}_\mathcal{L})$ sans avoir à construire complètement un facteur.

3.2.3 Passage des sommes réelles aux sommes nulles complexes

Ici nous supposons que nous avons trouvé le réseau V. Notre but à présent est de trouver le réseau W engendré par les 0-1 vecteurs $r_1, \ldots, r_s$ correspondant aux sommes nulles minimales entre les b_i.
Nous considérons alors l'application suivante :

$$f : \begin{array}{ccc} \mathbb{Z}^{\frac{n+l}{2}} & \longrightarrow & \mathbb{Z}^n \\ (x_1, \ldots, x_{\frac{n+l}{2}}) & \longmapsto & (x_1, \ldots, x_{\frac{n+l}{2}}, x_l, \ldots, x_{\frac{n+l}{2}}) \end{array}$$

Pour chaque 0-1 vecteur de V nous avons $< v_i, \overrightarrow{\epsilon\Re(b)} >= 0$. Donc $< f(v_i), \overrightarrow{b} >= 0$, où $\overrightarrow{b}$ est le vecteur de $\mathbb{C}^n$ qui a pour $i^{\text{ème}}$ coordonnées b_i.

Lemme 25. *Soit v un 0-1 générateur de V, c'est-à-dire v correspond à une somme nulle minimale sur les $\epsilon_i\Re(b_i)$. S'il existe un indice $1 \leq j_0 \leq l$ tel que $v_{j_0} \neq 0$, alors $f(v)$ correspond à une somme nulle minimale pour les b_i.*

Démonstration. Il est clair que $f(v)$ est un 0-1 vecteur correspondant à une somme nulle pour les b_i. Nous devons donc montrer que cette somme est minimale. Si $f(v)$ n'est pas minimale alors $f(v) = u_1 + u_2$ où u_i sont des 0-1 vecteurs. u_1 correspond à une somme minimale indexée par I_1 et $u_{1,j_0} \neq 0$.

Nous avons $< u_1, \overrightarrow{b} >= \sum_{i\in I_1} b_i = 0$, donc il existe un sous-ensemble I_2 tel que $\sum_{i\in I_1} \overline{b_i} = \sum_{i\in I_2} b_i = 0$.
Comme les ensemble d'indices correspondant aux sommes nulles minimales forment une partition de $\{1,\ldots,n\}$, nous avons $I_1 = I_2$ ou $I_1 \cap I_2 = \emptyset$. Mais $j_0 \in I_1$ et $b_{j_0} \in \mathbb{R}$, donc $I_1 = I_2$. D'où $u_{1,l+j} = u_{1,\frac{n+l}{2}+j}$ $(1 \leq j \leq \frac{n+l}{2} - l)$, et il existe un élément a dans V tel que $f(a) = u_1$. A présent nous considérons $u_2 = f(v) - u_1$ et nous obtenons $u_{2,l+j} = u_{2,\frac{n+l}{2}+j}$ $(1 \leq j \leq \frac{n+l}{2} - l)$, donc il existe un élément b dans V tel que $f(b) = u_2$. Donc $f(v) = f(a) + f(b) = f(a + b)$, et $v = a + b$. Ce qui contredit la minimalité de v. $\square$

Lemme 26. *Soit v un 0-1 générateur de V, c'est-à-dire v correspond à une somme nulle minimale sur les $\epsilon_i\Re(b_i)$. Si $v_j = 0$ (pour $1 \leq j \leq l$), alors soit $f(v)$ correspond à une somme minimale nulle soit $f(v) = u_1 + u_2$ où $u_{1,j} = u_{2,j} = 0$ pour $1 \leq j \leq l$, $u_{1,l+j} = u_{2,\frac{n+l}{2}+j}$ pour $1 \leq j \leq \frac{n+l}{2} - l$, $u_{1,\frac{n+l}{2}+j} = u_{2,l+j}$ pour $1 \leq j \leq \frac{n+l}{2} - l$, et u_i correspond à une somme minimale pour les b_i.*

Démonstration. Tout d'abord nous rappelons la notation suivante : Si E est une ensemble alors $|E|$ est le cardinal de cet ensemble.
Il est clair que $f(v)$ correspond à une somme nulle minimale pour les b_i. Nous avons $< f(v), \overrightarrow{b} >= \sum_{j\in I} b_j + \sum_{j\in J} b_j = \sum_{j\in I} b_j + \overline{b_j} = \sum_{j\in I} 2\Re(b_j) =< v, \overrightarrow{\epsilon\Re(b)} >= 0$, où $I \subset \{l+1,\ldots,\frac{n+l}{2}\}$, $J = \{j \,|\, \overline{b_j} = b_i$ où $i \in I\}$, et $|I| = |J| = \|v\|$.
A présent nous supposons qu'il existe une somme nulle minimale pour les b_j où j appartient à $I \sqcup J$. Dans ce cas il existe un sous-ensemble $H_1 \subset I \sqcup J$ tel que $\sum_{j\in H_1} b_j = 0$. Comme $\sum_{j\in H_1} \overline{b_j} = \sum_{j\in H_2} b_j = 0$ et que les indices des sommes nulles minimales forment une partition de $\{1,\ldots,n\}$, nous avons $H_1 = H_2$ ou $H_1 \cap H_2 = \emptyset$. Maintenant nous considérons $H_1 \cup H_2 = H$, et nous obtenons

$$\{\Re(b_i) \,|\, i \in H \cap I\} = \{\Re(b_i) \,|\, i \in H \cap J\}.$$

Donc $\sum_{j\in H} b_j = 0$ implique :

$$\sum_{j\in H} \Re(b_j) = \sum_{j\in H\cap I} \Re(b_j) + \sum_{j\in H\cap J} \Re(b_j) = \sum_{j\in H\cap I} 2\Re(b_j) = 0.$$

D'où $|H \cap I| \geq \|v\|$. En effet v correspond à une somme nulle minimale pour les $2\Re(b_j)$ lorsque $j \in I$. Comme $H \cap I \subset I$ nous avons $|H \cap I| \leq |I| = \|v\|$, donc $|H \cap I| = \|v\|$. Avec le même type d'arguments nous obtenons $|H \cap J| = \|v\|$. D'où $|H| = 2\|v\|$.

Si $H_1 = H$ (i.e. $H_1 = H_2$) alors $|H_1| = 2\|v\|$, et $H_1 = I \sqcup J$. Dans ce cas, nous avons $f(v)$ qui correspond à une somme nulle minimale, et, cela prouve la première partie du lemme.

Si $H = H_1 \sqcup H_2$ (i.e. $H_1 \cap H_2 = \emptyset$) alors $|H_1| = |H_2| = \|v\|$. Dans ce cas nous avons $I \sqcup J = H_1 \sqcup H_2$, et, $\sum_{j\in H_i} b_j = 0$ sont deux sommes nulles minimales. Ces résultats avec la définition de H_2 prouvent la deuxième partie du lemme. $\square$

A l'aide des deux lemmes précédents nous sommes capables de détecter les sommes nulles minimales pour b_i.

Comment décomposer $f(v)$

Nous avons vu que dans certains cas nous devons décomposer $f(v) = u_1 + u_2$. Ici, nous allons expliquer comment obtenir u_1 et u_2.
Soit v un 0-1 générateur de V tel que $\|f(v)\| = 2m$. Soient $e_1, \dots, e_n$ la base canonique de $\mathbb{R}^n$, nous avons $f(v) = \sum_{i=1}^n x_i e_i = \sum_{j=1}^{2m} x_{i_j} e_{i_j}$, où $x_i = 0$ ou 1.
Nous devons trouver deux sommes nulles minimales pour les $b_{i_1}, \dots, b_{i_{2m}}$. Nous allons procéder de la même manière qu'en 3.2.2.

Les vecteurs u'_1 et u'_2 sont des 0-1 vecteurs de $\mathbb{Z}^{2m}$ et correspondent à deux sommes nulles minimales différentes pour les $b_{i_1}, \dots, b_{i_{2m}}$. (On remarque que nous pouvons aisément obtenir u_1 et u_2 à partir de u'_1 et u'_2). Soit U_v le réseau engendré par u'_1 et u'_2.
Comme précédemment nous allons construire une suite de réseaux $\mathcal{L}_{v,i}$ telle que :

$$\begin{cases} U_v \subset \mathcal{L}_{v,i+1} \subset \mathcal{L}_{v,i} \subset \mathbb{Z}^{2m} \\ \text{Il existe un entier } k \text{ tel que pour tout } i \geq k \text{ nous avons } \mathcal{L}_{v,i} = \mathcal{L}_{v,k} = U_v. \end{cases}$$

Nous initialisons la suite avec : $\mathcal{L}_{v,0} = \mathbb{Z}^{2m}$, et maintenant nous expliquons comment obtenir $\mathcal{L}_{v,i+1}$ à partir de $\mathcal{L}_{v,i}$.

Soit $\mathcal{L}_v$ un réseau engendré par $w_1, \dots, w_k$ où $w_i = (w_{i,1}, \dots, w_{i,2m}) \in \mathbb{Z}^{2m}$, tel que $U_v \subset \mathcal{L}_v$. Soit $q_{\mathcal{L}_v}$ le morphisme de $\mathbb{Z}$-modules suivant :

$$\begin{array}{rccl} q_{\mathcal{L}_v} : & \mathcal{L}_v & \longrightarrow & \mathcal{L}'_v \\ & w_i & \longmapsto & (w_{i,1}, \dots, w_{i,2m}, \sum_{j=1}^{2m} w_{i,j} \lfloor C\Re(\tilde{b}_{i_j}) \rceil, \sum_{j=1}^{2m} w_{i,j} \lfloor C\Im(\tilde{b}_{i_j}) \rceil) \end{array}$$

Lemme 27. *Soient $\mathcal{L}_v$ un réseau tel que $U_v \subset \mathcal{L}_v$, et $\mathcal{L}'_v = q_{\mathcal{L}_v}(\mathcal{L}_v)$. Soient $\nu_1, \dots, \nu_k$ une base de $\mathcal{L}'_v$ et k_0 un entier tel que : pour tout $i > k_0$, $\|\nu_i^*\| > M$ où $M = \sqrt{m + [(\frac{1}{2} + C\eta)m]^2}$. On a alors : Le réseau engendré par $q_{\mathcal{L}_v}^{-1}(\nu_1), \dots, q_{\mathcal{L}_v}^{-1}(\nu_{k_0})$ contient U_v.*

Démonstration. Utiliser le même type d'arguments que pour le lemme 22. □

Ici la condition d'arrêt est plus facile car nous savons combien de vecteurs nous voulons obtenir.

Nous arrêtons donc le calcul de la suite de réseaux lorsque nous obtenons un réseau engendré par deux 0-1 vecteurs orthogonaux, ou bien, lorsque le réseau obtenu est engendré par un vecteur dont toutes les coordonnées sont égales à 1.

Calcul des sommes nulles complexes

Un b_i est réel

Nous supposons que b_{j_0} est réel. Soit v_1 le 0-1 vecteur correspondant à la somme nulle minimale pour les $\epsilon_j\Re(b_j)$ où b_{j_0} apparaît. D'après le lemme 25, $f(v_1)$ correspond à une somme nulle minimale pour les b_j. Donc $\|f(v_1)\|^2 = m$ est le degré d'un facteur absolu. Maintenant nous considérons un vecteur v_2 qui correspond à une somme nulle minimale pour les $\epsilon_j\Re(b_j)$, mais où $v_{2_k} = 0$ $(1 \leq k \leq l)$. Cela signifie que dans cette somme nulle il n'y a pas de b_j réel. Le lemme 26 nous dit alors que $f(v_2)$ correspond à une somme nulle entre les b_j telle que :
Si $\|f(v_2)\|^2 = m$ alors $f(v_2)$ correspond a une somme nulle minimale, sinon $\|f(v_2)\|^2 = \|u_1 + u_2\|^2 = \|u_1\|^2 + \|u_2\|^2 = 2m$ et dans ce cas nous devons calculer u_1 et u_2 (voir le lemme 27).

Tous les b_i appartiennent à $\mathbb{C} \setminus \mathbb{R}$

Ici nous avons deux possibilités : soit tous les $f(v_i)$ ont la même norme, soit il existe deux vecteurs v_1 et v_2 dans V tels que $\|f(v_1)\| < \|f(v_2)\|$. D'après le lemme 26, nous avons $\|f(v)\|^2 = m$ ou $2m$ pour chaque 0-1 générateur de V. Donc dans le premier cas, $\|f(v_1)\|^2 = \ldots = \|f(v_t)\|^2 = l$, et $l = m$ ou $l = 2m$. Pour savoir si $l = m$ ou $l = 2m$ nous essayons de décomposer $f(v_1)$ comme expliqué ci-dessus. Dans le deuxième cas $\|f(v_1)\|^2 = m$, et $f(v_1)$ correspond à une somme nulle minimale. Donc si $f(v)$ est tel que $\|f(v)\| = \|f(v_1)\|$ alors $f(v)$ correspond à une somme nulle minimale, sinon $f(v) = u_1 + u_2$ où u_1 et u_2 donnent des sommes minimales nulles.

3.3 L'algorithme

Ici nous allons décrire notre algorithme, ensuite nous prouverons qu'il termine et qu'il retourne un facteur absolument irréductible.

3.3.1 Description de l'algorithme

Algorithme Fac-knap

Entrée : $P(X,Y) \in \mathbb{Q}[X,Y]$ un polynôme irréductible dans $\mathbb{Q}[X,Y]$ tel que $\deg(P) = \deg_Y(P) = n$.
Sortie : $P_1(X,Y) \in \mathbb{Q}[\alpha][X,Y]$ un facteur absolument irréductible de P, $Q = P_2 \ldots P_s \in \mathbb{Q}[\alpha][X,Y]$ le quotient de P par P_1, et, $f_\alpha(T) \in \mathbb{Z}[T]$ le polynôme minimal de α, (α est entier sur $\mathbb{Z}$).

1. $C := 1$.
2. Changement générique de coordonnées : Choisir $\lambda \in \mathbb{Z}$, et poser $P(X,Y) := P(X + \lambda Y, Y)$.
3. Choix de la fibre : Choisir $x_0 \in \mathbb{Z}$.

4. Calculer l le nombre de $b_i \in \mathbb{R}$.
   Poser
$$C := \max\left(\frac{\sqrt{2}}{\sqrt{n+l}}\sqrt{\left(\frac{n+l}{2} + [\frac{5}{4}(n+l)]^2\right)^{\frac{n+l}{2}} - 1}, C\right) \text{ et } \eta := \frac{1}{C}.$$
   Calculer y_i, a_i, b_i avec la précision η.
5. Trier les b_i comme expliqué en 3.2.2 page 78.
6. Poser $\mathcal{L} := \mathbb{Z}^{\frac{n+l}{2}}$ (avec base canonique).
7. Calculer la base LLL réduite de $\mathcal{L}'$, supprimer les vecteurs de grandes normes (comme dans le lemme 22) et garder k_0 vecteurs w_i tels que $\mathcal{L} := Vect_{\mathbb{Z}}(w_1, \ldots, w_{k_0}) \supset V$.
8. Calculer $RREF(\mathcal{B}_{\mathcal{L}})$.
9. Si $RREF(\mathcal{B}_{\mathcal{L}})$ ne vérifie pas le test d'arrêt du lemme 24, alors :
   Soit il existe un vecteur $v = (v_1, \ldots, v_{\frac{n+l}{2}})$ tel que v n'est pas un 0-1 vecteur, et, $| < v, \overrightarrow{C\lfloor \Re(\tilde{b}) \rceil} > | \leq (\frac{1}{2} + C\eta) \sum_{i=1}^{\frac{n+l}{2}} |v_i|$. Dans ce cas poser $C := C^2$, $\eta := \frac{1}{C}$ et retourner à l'étape 2.
   Soit il n'existe pas de tels vecteurs, dans ce cas poser $C := C^2$, $\eta := \frac{1}{C}$ recalculer b_i avec la précision η et retourner à l'étape 7.
10. Si $RREF(\mathcal{B}_{\mathcal{L}})$ vérifie le test d'arrêt du lemme 24, alors :
    (a) Obtenir les sommes nulles comme expliqué dans les lemmes 25 et 26.
    (b) Décomposer, lorsque cela est nécessaire, le vecteur $f(v) = u_1 + u_2$ de la même manière qu'en $7, 8$ et 9, (mais avec le lemme 27).
11. $r_1, \ldots, r_s$ sont les 0-1 vecteurs correspondant aux sommes nulles minimales. Vérifier $\|r_1\|^2 = \|r_i\|^2$, pour $2 \leq i \leq s$, et, $s\|r_1\|^2 = n$. Si ce n'est pas le cas alors poser $C := C^2$, $\eta := \frac{1}{C}$ et retourner en 2.
12. Construire $\tilde{P}_i \mod (X - x_0)^3$ pour $i = 1, \ldots, s$.
13. Reconnaître les facteurs exacts $P_1 \mod (X - x_0)^3$ et $Q \mod (X - x_0)^3$ à partir des facteurs approchés $\tilde{P}_i$.
14. Vérifier $\sum_{i \in I} b_i = 0$ où I est l'ensemble d'indices correspondant au facteur P_1 :
    Si $\sum_{i \in I} b_i \neq 0$ alors poser $C := C^2$, $\eta := \frac{1}{C}$ et retourner en 7, sinon relever cette factorisation.
15. Vérifier $P_1 Q = P$: Si ce n'est pas le cas alors poser $C := C^2$, $\eta := \frac{1}{C}$ et retourner à l'étape 2, sinon rendre après changement de variables P_1, Q, et f_α.

REMARQUES : L'étape 13 est effectuée en utilisant les techniques développées au chapitre 2.
La formule utilisée pour la constante C sera expliquée en 3.4.1 page 91.

3.3.2 Terminaison et correction de l'algorithme

Lemme 28. *Si l'algorithme termine alors l'algorithme rend un facteur absolument irréductible.*

Démonstration. Notre algorithme rend un facteur de P nous devons donc montrer que ce facteur est absolument irréductible. Si l'algorithme termine cela signifie que le test d'arrêt est vérifié (voir lemme 24). Donc $RREF(\mathcal{B}_{\mathcal{L}}) = RREF(\mathcal{B}_V)$, les 0-1 vecteurs utilisés pour construire les facteurs P_i sont donc bien les 0-1 vecteurs correspondant aux sommes nulles minimales. ☐

Lemme 29. *L'algorithme termine.*

Démonstration. Afin de montrer que l'algorithme termine nous allons montrer qu'il n'y a qu'un nombre fini de retours aux étapes 2 et 7.
► Il y a un nombre fini de retours à l'étape 2 :
Les retours à l'étape 2 proviennent des mauvais cas des étapes 9, 11 et 15. Le retour à l'étape 2 correspond à une situation non générique. Il est clair que nous ne sommes pas dans une situation générique lorsque nous rencontrons les mauvais cas des étapes 11, et 15.
Dans le premier cas de l'étape 9 nous avons $| < v, \overrightarrow{\lfloor C\Re(\tilde{b})\rceil} > | \leq (\frac{1}{2} + C\eta)\sum_{i=1}^{\frac{n+l}{2}} |v_i|$, donc il est possible que v donne une somme nulle pour les $\epsilon_i \Re(b_i)$, et ensuite pour les b_i. Mais dans ce cas v n'est pas un 0-1 vecteur, donc nous ne sommes pas dans une situation générique. De plus les lemmes 22 et 27 montrent que nous considérons des relations bornées. Donc d'après le théorème 24 page 76, nous n'avons qu'à modifier un nombre fini de fois x_0 et λ_0 pour être dans une situation générique.
Maintenant nous remarquons que dans une situation générique, si C est suffisamment grand le premier cas de l'étape 9 ne peut pas apparaître. En effet si $| < v, \overrightarrow{\Re(b)} > | = e > 0$, donc $| < v, \overrightarrow{C\Re(b)} > | = Ce > 0$.
Donc si C est suffisamment grand nous ne pouvons pas obtenir

$$| < v, \overrightarrow{\lfloor C\Re(\tilde{b})\rceil} > | < (\frac{1}{2} + C\eta)\sum_{i=1}^{\frac{n+l}{2}} |v_i|.$$

Il nous reste à voir alors que dans une situation générique nous ne pouvons pas avoir de vecteur v provenant de $RREF(\mathcal{B}_{\mathcal{L}})$ donnant $< v, \overrightarrow{\Re(b)} >= 0$ et $v \notin \{0,1\}^{\frac{n+l}{2}}$. La seule possibilité pour v est d'être de la forme (d'après le théorème 24), $v = \sum_{i=1}^{t} \lambda_i v_i$, où $v_1, \ldots, v_t$ désigne la base de 0-1 vecteurs engendrant V, et $\lambda_i \in \mathbb{Z}$. Nous supposons donc avoir v une ligne de $RREF(\mathcal{B}_{\mathcal{L}})$ telle que $< v, \overrightarrow{\Re(b)} >= 0$, $v \notin \{0,1\}^{\frac{n+l}{2}}$ et $v = v_1 + \lambda v_2$ avec $\lambda \in \mathbb{Z} - \{0,1\}$. Nous allons montrer que cela est impossible.
Nous notons $l_1, \ldots, l_k$ les vecteurs correspondant aux lignes de $RREF(\mathcal{B}_{\mathcal{L}})$, ce sont des vecteurs de base de $\mathcal{L}\otimes\mathbb{Q}$. De plus il existe un indice i_0 tel que $l_{i_0} = v = v_1 + \lambda v_2$.

Comme $v_i \in V \subset \mathcal{L}$, on a $v_i \in \mathcal{L} \otimes \mathbb{Q}$, donc $v_i = \sum_{i=1}^{k} \mu_i l_i$. Du fait que $\{l_1, \dots, l_k\}$ est une base échelonnée réduite, et, que les vecteurs v_i soient des 0-1 vecteurs il vient $\mu_i = 0$ ou 1. D'où $v_1 = \sum_{i=j_1}^{j_{k'}} l_i$ où $j_1 \geq i_0$, et, $v_2 = \sum_{c_1}^{c_{k''}} l_i$ où $c_1 > i_0$ (car la base est échelonnée réduite et $l_{i_0} = v_1 + \lambda v_2$). Les vecteurs l_i étant linéairement indépendants, il vient $\lambda = 0$ et $l_{i_0} = v = v_1$, ce qui contredit notre hypothèse. Ainsi dans une situation générique les seules relations possibles sur les $\epsilon_i \Re(b_i)$ provenant d'une matrice sous forme échelonnée réduite sont des 0-1 vecteurs.
Dans une situation générique, avec C suffisamment grand, nous ne rencontrons donc pas le mauvais cas de l'étape 9. Cela prouve notre première affirmation.

► Il y a un nombre fini de retours à l'étape 7.
Nous supposons que nous sommes dans une situation générique. Nous retournons en 7 dans le deuxième cas de l'étape 9 et dans le premier cas de l'étape 14.
Nous retournons en 7 afin d'obtenir un nouveau réseau de dimension inférieure. En effet nous allons montrer que si C est suffisamment grand alors dans l'étape 7 $dim_{\mathbb{Z}}(\mathcal{L})$ diminue. Ainsi comme $V \subset \mathcal{L}$, nous allons obtenir au bout d'un temps fini $V \subset \mathcal{L}$ et $dim_{\mathbb{Z}}(\mathcal{L}) = dim_{\mathbb{Z}}(V)$. Donc $V \otimes \mathbb{Q} \subset \mathcal{L} \otimes \mathbb{Q}$ et $dim_{\mathbb{Q}}(\mathcal{L}) = dim_{\mathbb{Q}}(V)$. D'où $V \otimes \mathbb{Q} = \mathcal{L} \otimes \mathbb{Q}$, et donc $RREF(\mathcal{B}_{\mathcal{L}}) = RREF(\mathcal{B}_V)$. Ainsi au bout d'un temps fini nous allons obtenir l'égalité $RREF(\mathcal{B}_{\mathcal{L}}) = RREF(\mathcal{B}_V)$ qui correspond au test d'arrêt (voir lemme 24).
Supposons donc que $\mathcal{L}' = Vect_{\mathbb{Z}}(p_{\mathcal{L}}(w_1), \dots, p_{\mathcal{L}}(w_k))$, alors nous avons $\det(\mathcal{L}') = \sqrt{\det(< p_{\mathcal{L}}(w_i), p_{\mathcal{L}}(w_j) >)}$. La définition de $p_{\mathcal{L}}(w_i)$ entraîne : $\det(\mathcal{L}')$ est un "polynôme" en C. Soit $\nu_1, \dots, \nu_k$ une base LLL réduite de $\mathcal{L}'$, nous avons alors $\det(\mathcal{L}') = \prod_{i=1}^{k} \|\nu_i^*\|$. Donc si C est suffisamment grand il existe un indice k_0 tel que

$$\|\nu_{k_0}^*\| > \sqrt{\frac{n+l}{2} + \left[(\frac{1}{2} + 2C\eta) + \frac{n+l}{2} \right]^2 .2^{\frac{n+l}{4}}} = \mathcal{M}.$$

Comme $C\eta \leq 1$, $\mathcal{M}$ ne dépend que de n et l. D'où $\mathcal{M} \leq \|\nu_{k_0}^*\| \leq 2^{\frac{i}{2}} \|\nu_{k_0+i}^*\|$, pour $1 \leq i \leq \frac{n+l}{2} - k_0$, voir la proposition 4 page 15, et donc

$$\sqrt{\frac{n+l}{2} + \left[(\frac{1}{2} + 2C\eta) + \frac{n+l}{2} \right]^2} \leq \mathcal{M}.2^{\frac{-i}{2}} \leq \|\nu_{k_0+i}^*\|, \text{ pour } 1 \leq i \leq \frac{n+l}{2} - k_0.$$

Donc la dimension de $\mathcal{L}'$ décroît, et donc la dimension de $\mathcal{L}$ aussi. Cela prouve notre deuxième affirmation. □

3.3.3 Étude des éléments primitifs

Dans l'étape 13 de notre algorithme de factorisation absolue nous effectuons une reconnaissance "approchée-exacte". Celle-ci diffère un peu de celle effectuée au chapitre 2. En effet ici nous avons simplement l'expression d'une factorisation approchée modulo $(X-x_0)^3$. Il se peut donc que les coefficients de $P_1 \mod (X-x_0)^3$ n'engendrent pas le corps $\mathbb{Q}[\alpha]$ (corps engendré par tous les coefficients de P_1). Ici nous allons voir qu'après un changement générique de coordonnées les coefficients de $P_1 \mod (X - x_0)$ suffisent pour obtenir $\mathbb{Q}[\alpha]$. Cela rend donc possible le passage approché-exacte de l'algorithme.

Lemme 30. *Soient $F(X,Y) = P(X + x_0 + \lambda Y, Y)$ avec $x_0 \in \mathbb{Q}$ et $\lambda \in \mathbb{Q}$, $P(X,Y) = P_1(X,Y) \cdots P_s(X,Y)$ la factorisation absolue de P, et, $F(X,Y) = F_1(X,Y) \dots F_s(X,Y)$ la factorisation absolue de F.*
Nous désignons par $\mathbb{Q}[\alpha]$ le corps engendré par les coefficients du polynôme P_1, et par $\{coeff(F_1(0,Y))\}$ l'ensemble des coefficients de $F_1(0,Y)$.
Dans ce cas pour tous les x_0 et tous les λ sauf un nombre fini nous avons :

$$\mathbb{Q}[\alpha] = \mathbb{Q}[\{coeff(F_1(0,Y))\}]$$

Démonstration. On pose $H(X,Y) = P(X + x_0, Y)$, et $H_i(X,Y) = P_i(X + x_0, Y)$.
On a $P_1(X,Y) = \sum_{i,j} a_{i,j} X^i Y^j$ et $H_1(X,Y) = \sum_{i,j} a_{i,j}(X + x_0)^i Y^j$.
D'où

$$\mathbb{Q}[\{\text{coeff}(H_1)\}] \subset \mathbb{Q}[\alpha]$$

où $\{\text{coeff}(H_1)\}$ désigne l'ensemble des coefficients de P_1.
Or $[\mathbb{Q}[\{\text{coeff}(H_1)\}] : \mathbb{Q}] = s$ d'après le lemme fondamental. D'où

$$\mathbb{Q}[\{\text{coeff}(H_1)\}] = \mathbb{Q}[\alpha] \quad (\star).$$

A présent on pose $G(Y) = H(\lambda Y, Y)$, $G_i(Y) = H_i(\lambda Y, Y)$. On obtient alors :

$$G_1(Y) = H_1(\lambda Y, Y) = P_1(\lambda Y + x_0, Y) = F_1(0, Y).$$

On pose $H_1(X,Y) = \sum_{k=0}^{m} \sum_{i=0}^{k} h_{i,k-i} X^i Y^{k-i}$. D'où

$$F_1(0,Y) = H_1(\lambda Y, Y) = \sum_{k=0}^{m} (\sum_{i=0}^{k} h_{i,k-i} \lambda^i) Y^k.$$

On note :

$$h_k(\lambda) = \sum_{i=0}^{k} h_{i,k-i} \lambda^i.$$

Nous allons montrer que : $\mathbb{Q}[\{h_{i,k-i} \,|\, i = 0, \dots, k\}] = \mathbb{Q}[h_k(\lambda)]$ pour tous les λ sauf un nombre fini $(\star\star)$.
Ainsi nous obtiendrons $\mathbb{Q}[\{\text{coeff}(H_1)\}] = \mathbb{Q}[\{h_k(\lambda) \,|\, k = 0, \dots, m\}]$. Comme $\mathbb{Q}[\{h_k(\lambda) \,|\, k = 0, \dots, m\}] = \mathbb{Q}[\{\text{coeff}(F_1(0,Y))\}]$, en utilisant $(\star)$ nous en déduirons le résultat souhaité.
On pose alors $d_k = [\mathbb{Q}[\{h_{i,k-i} \,|\, i = 0, \dots, k\}] : \mathbb{Q}]$. On note $\sigma_1, \dots, \sigma_{d_k}$ les d_k

$\mathbb{Q}$-homomorphismes de $\mathbb{Q}[\{h_{i,k-i} \,|\, i = 0, \ldots, k\}]$ dans $\mathbb{C}$. Soit

$$\begin{aligned} f_k(T) &= \prod_{u \neq v} (\sigma_u(h_k(T)) - \sigma_v(h_k(T))) \\ &= \prod_{u \neq v} [(\sigma_u - \sigma_v)(\sum_{i=0}^{k} h_{i,k-i} T^i)] \\ &= \prod_{u \neq v} [\sum_{i=0}^{k} (\sigma_u - \sigma_v)(h_{i,k-i}) T^i] \in \mathbb{C}[T]. \end{aligned}$$

Nous avons $f_k(T) \neq 0$ dans $\mathbb{C}[T]$, car sinon nous aurions deux indices u_0 et v_0 tels que $u_0 \neq v_0$ et $\sum_{i=0}^{k} (\sigma_u - \sigma_v)(h_{i,k-i}) T^i = 0$. Nous aurions alors : $\sigma_{u_0}(h_{i,k-i}) = \sigma_{v_0}(h_{i,k-i})$, pour $0 \leq i \leq k$, cela signifierait $\sigma_{u_0} = \sigma_{v_0}$ ce qui est absurde car $u_0 \neq v_0$.
Donc si λ n'est pas racine de $f_k(T)$ alors $h_k(\lambda)$ a au moins d_k conjugués. Cela signifie que $h_k(\lambda)$ est un élément primitif de $\mathbb{Q}[\{h_{i,k-i} \,|\, i = 0, \ldots, k\}]$, ce qui prouve $(\star\star)$. $\square$

Exemple :
Soit $P(X, Y) = Y^{20} + 2Y^{10} - 2X^8Y^6 + 1 = (Y^{10} - \sqrt{2}X^4Y^3 + 1)(Y^{10} + \sqrt{2}X^4Y^3 + 1)$. Nous obtenons $\mathbb{Q}[\alpha] = \mathbb{Q}[\sqrt{2}]$ uniquement pour $P_1 \mod X^5$. Après changement de variables nous avons :

$$\begin{aligned} F(X,Y) \; = P(X+Y,Y) = \; & Y^{20} + 2Y^{10} - 2X^8Y^6 - 16Y^7X^7 - 56Y^8X^6 - 112Y^9X^5 \\ & -140Y^{10}X^4 - 112Y^{11}X^3 - 56Y^{12}X^2 - 16Y^{13}X \\ & -2Y^{14} + 1, \end{aligned}$$

$$F_1(X,Y) = Y^{10} - \sqrt{2}X^4Y^3 - 4\sqrt{2}Y^4X^3 - 6\sqrt{2}Y^5X^2 - 4\sqrt{2}Y^6X - \sqrt{2}Y^7 + 1.$$

Nous obtenons donc bien $\mathbb{Q}[\{\mathrm{coeff}(F_1(0, Y)\}] = \mathbb{Q}[\sqrt{2}]$.

3.3.4 Nombre de b_i réel

La stratégie utilisée, consistant à trouver tout d'abord les sommes nulles minimales sur les $\Re(b_i)$, nous a amené à ne considérer que $\frac{n+l}{2}$ nombres réels au lieu de n nombres complexes, où l désigne le nombre de $b_i \in \mathbb{R}$. Afin de comparer les deux nombres $\frac{n+l}{2}$ et n, nous allons rappeler un résultat sur le nombre de racines réelles d'un polynôme (voir [Kac43], [EK95]).

Théorème 25. *Soient $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \in \mathbb{R}[x]$ des polynômes aléatoires où les a_i proviennent de variables aléatoires indépendantes suivant une loi normale. Le nombre moyen E_n de racines réelles pour ces polynômes admet le développement asymptotique suivant (quand $n \to \infty$) :*

$$E_n = \frac{2}{\pi} \log(n) + 0.62573\ldots + \frac{2}{n\pi} + O(\frac{1}{n^2}).$$

Si les variables aléatoires sont indépendantes, et suivent des lois normales de moyenne nulle mais de variance égale à $\binom{n}{i}$, *alors nous avons*

$$E_n = \sqrt{n}.$$

Comme les b_i sont des nombres conjugués sur $\mathbb{Q}$, d'après ce théorème nous pouvons supposer qu'en général nous avons $\frac{n+l}{2} < n$. Ainsi il est préférable d'appliquer l'algorithme LLL à un réseau de dimension $\frac{n+l}{2}$ qu'à un réseau de dimension n. Cela explique notre choix d'étudier dans un premier temps les sommes nulles minimales pour les $\Re(b_i)$.

3.3.5 Complexité

Nous avons montré que notre algorithme termine, mais nous ne savons pas dire a priori combien de fois celui-ci va utiliser LLL. En pratique l'algorithme ne passe qu'une seule fois par l'étape 7, c'est-à-dire nous ne calculons qu'une seule fois une base LLL réduite. Cette étape remplaçant le calcul des 2^n sommes possibles entre les b_i, nous avons donc en pratique un algorithme avec une complexité polynomiale. En effet nous pouvons supposer que les k vecteurs v_i auxquels nous appliquons LLL sont tels que $\|v_i\| \approx C$. Or nous savons (voir [LLL82], [vzGG03], Propriété 5 page 16) que le nombre d'opérations arithmétiques nécessaires pour obtenir une base LLL réduite est de l'ordre de $O(k^4 \log(C))$ et ces opérations sont effectuées sur des nombres entiers de taille $O(klogC)$. Comme ici $k = \frac{n+l}{2}$ et $C \approx (\frac{n+l}{2})^{\frac{n+l}{2}}$, nous devons donc effectuer $O((\frac{n+l}{2})^5 \log(\frac{n+l}{2}))$ opérations arithmétiques sur des entiers de taille $O((\frac{n+l}{2})^2 \log(\frac{n+l}{2}))$.

Dans ce qui précède nous affirmons obtenir une complexité polynomiale en pratique car une seule utilisation de LLL s'avère suffisante. On peut alors se demander si avec un autre choix pour la constante C nous pouvons garantir le nombre de passages par l'étape 7. C'est ce que nous ferons dans la section suivante. Nous verrons alors que pour être sûr de n'utiliser qu'une seule fois LLL nous devons prendre dans certains cas une constante C supérieure à 2^{2^n}. Avec une telle constante nous obtenons alors un algorithme exponentiel.

3.4 Remarques sur la précision à utiliser dans les calculs

Un point important dans notre algorithme est le choix de la précision dans les calculs, ou ce qui est équivalent le choix de la constante C. Dans cette section nous allons expliquer le choix fait pour la constante C dans l'étape 4. Ce choix provient d'une heuristique et donne de bons résultats en pratique (voir section 3.6). Nous utilisons cette heuristique afin d'obtenir une constante C suffisamment grande pour permettre à la dimension du réseau $\mathcal{L}$ de décroître lors de l'étape 7. Nous verrons ensuite qu'il existe une formule théorique nous permettant d'obtenir une constante C suffisamment grande pour obtenir le réseau V en utilisant une seule fois LLL. Pour finir nous donnerons une méthode permettant de certifier l'erreur commise sur les b_i lors des calculs.

3.4.1 Comment choisir η

Tout d'abord nous allons expliquer pourquoi nous posons $\eta = \frac{1}{C}$.
Nous avons $|\tilde{b}_i - b_i| < \eta$, donc $|C\Re(\tilde{b}_i) - C\Re(b_i)| < C\eta$. Comme nous étudions la "partie entière" de ces nombres, nous voulons $C\eta < 1$. Mais si $C\eta << 1$, cela signifie que lorsque nous prenons la partie entière nous n'utilisons pas toute l'information exacte se trouvant dans la partie décimale. Donc le choix optimal est $C\eta = 1$. Comme nous voulons que C appartienne à $\mathbb{Z}$ nous choisissons d'abord C puis nous posons $\eta = \frac{1}{C}$.

A présent nous voulons choisir η (donc C), de telle sorte que nous puissions supprimer certains vecteurs durant l'étape 7. Donc soit $\nu_1, \ldots, \nu_{\frac{n+l}{2}}$ une base LLL réduite de $\mathcal{L}'$, où $\mathcal{L} = \mathbb{Z}^{\frac{n+l}{2}}$ avec sa base canonique.
Nous avons

$$\prod_{i=1}^{\frac{n+l}{2}} \|\nu_i^*\| = \det(\mathcal{L}') = \sqrt{\det(A)}$$

où A est la matrice symétrique de taille $\frac{n+l}{2} \times \frac{n+l}{2}$ telle que $A_{i,i} = 1 + [C\lfloor \epsilon_i \Re(\tilde{b}_i)\rceil]^2$, et $A_{i,j} = C^2 \epsilon_i \epsilon_j \lfloor \Re(\tilde{b}_i)\rceil \lfloor \Re(\tilde{b}_j)\rceil$ si $i \neq j$. Pour obtenir une estimation de C nous remplaçons $\lfloor \epsilon_i \Re(\tilde{b}_i)\rceil$ par 1. Cela signifie que nous n'étudions pas le déterminant de la matrice A, mais le déterminant de la matrice B, où B est la matrice symétrique de taille $\frac{n+l}{2} \times \frac{n+l}{2}$ telle que $B_{i,i} = 1 + C^2$, et $B_{i,j} = C^2$ si $i \neq j$.
De plus $\det B = 1 + (\frac{n+l}{2})C^2$, et nous voulons qu'un saut se produise, c'est-à-dire soit

$$\|\nu_i^*\| \approx \sqrt{\frac{n+l}{2} + \left[(\frac{1}{2} + 2C\eta)\frac{n+l}{2}\right]^2}$$

pour les vecteurs que l'on veut conserver, soit

$$\|\nu_i^*\| > \sqrt{\frac{n+l}{2} + \left[(\frac{1}{2} + 2C\eta)\frac{n+l}{2}\right]^2}$$

pour les vecteurs que nous voulons supprimer. On obtient donc :

$$1 + \frac{n+l}{2}C^2 \geq (\frac{n+l}{2} + \left[(\frac{1}{2} + 2C\eta)\frac{n+l}{2}\right]^2)^{\frac{n+l}{2}},$$

il en découle :

$$C \geq \frac{\sqrt{2}}{\sqrt{n+l}} \sqrt{\left(\frac{n+l}{2} + \left[\frac{5}{4}(n+l)\right]^2\right)^{\frac{n+l}{2}} - 1}.$$

3.4.2 Étude théorique de la précision

Ici nous allons voir que si on prend une précision suffisante, c'est-à-dire une constante C suffisamment grande alors en une exécution de LLL nous obtenons le réseau V. Bien évidemment, nous supposons que nous sommes dans une situation générique, en particulier V est engendré par des 0-1 vecteurs. Posons quelques notations :

Notations : $M = \sqrt{\frac{n+l}{2} + [(\frac{5}{2}).\frac{n+l}{2}]^2}$, $\nu_1, \dots, \nu_{\frac{n+l}{2}}$ est une base LLL réduite de $\mathcal{L}'$, où $\nu_i = (\nu_{i,1}, \dots, \nu_{i,\frac{n+l}{2}}, \sum_{k=0}^{\frac{n+l}{2}} \nu_{i,k} \lfloor C\epsilon_k \Re(\tilde{b_k}) \rceil)$.
On introduit la quantité suivante :

$$\epsilon_{\mathfrak{M}} = \inf \Big\{ |\sum_k \lambda_k \epsilon_k \Re(b_k)| \ \mid \ \lambda_k \in \mathbb{Z},\, |\lambda_k| \leq \mathfrak{M},\, \sum_k \lambda_k \epsilon_k \Re(b_k) \neq 0 \Big\}.$$

On remarque que $\epsilon_{\mathfrak{M}} \neq 0$.

Nous souhaitons trouver une constante C telle que $\|\nu_1^*\|, \dots, \|\nu_{k_0}^*\|$ soient inférieurs à M, et $\|\nu_{k_0+1}\|, \dots, \|\nu_{\frac{n+l}{2}}^*\|$ soient supérieurs à M, avec $V = Vect_{\mathbb{Z}}(p_{\mathcal{L}}^{-1}(\nu_1), \dots, p_{\mathcal{L}}^{-1}(\nu_{k_0}))$. Pour obtenir cela nous allons :

1. Trouver une constante $N > M$ telle que : $\|\nu_i^*\| \geq N \Longrightarrow \forall j \geq 0,\, \|\nu_{i+j}^*\| \geq M$.
2. Trouver une condition sur C afin d'obtenir :

$$|\sum_k \nu_{i,k} \epsilon_k \Re(b_k)| \neq 0 \Longrightarrow \|\nu_i^*\| > N > M.$$

Avec de telles conditions nous obtenons : $\|\nu_i^*\| \leq M \Longrightarrow |\sum_k \nu_{i,k}\epsilon_k \Re(b_k)| = 0$. Donc si $\|\nu_i^*\| \leq M$ pour $i = 1, \dots, k_0$, alors $\nu_1, \dots, \nu_{k_0}$ appartiennent à V. Ainsi comme $V \subset Vect_{\mathbb{Z}}(\nu_1, \dots, \nu_{k_0})$ d'après le lemme 22 et que $\nu_1, \dots, \nu_{k_0} \in V$, nous en déduisons $Vect_{\mathbb{Z}}(\nu_1, \dots, \nu_{k_0}) = V$.

A présent trouvons la constante N.

Lemme 31. *Soit* $N = 2^{\frac{n+l}{4}} M$. *Si* $\|\nu_i^*\| \geq N$ *alors pour tout* $0 \leq j \leq \frac{n+l}{2}$ *nous avons* $\|\nu_{i+j}^*\| \geq M$.

Démonstration. Nous avons $\|\nu_i^*\| \leq 2^{\frac{j}{2}} \|\nu_{i+j}^*\|$ car la base est LLL réduite. D'où :

$$2^{\frac{n+l}{4}} M \leq \|\nu_i^*\| \leq 2^{\frac{j}{2}} \|\nu_{i+j}^*\| \Longrightarrow 2^{(\frac{n+l}{2}-j)/2} M \leq \|\nu_{i+j}^*\| \Longrightarrow M \leq \|\nu_{i+j}^*\|.$$

□

Le lemme suivant donne une condition suffisante sur C pour obtenir le deuxième point qui nous permettra de conclure.

Lemme 32. *Avec les notations précédentes, $N = 2^{\frac{n+l}{4}}M$, et sous l'hypothèse :*

$$C > \frac{2^{(n+l-1)/2}M + (\frac{n+l}{2})\frac{5}{2}2^{(n+l-1)/2}}{\epsilon_{2^{(n+l-1)/2}M}} \quad (\star).$$

Nous avons la propriété suivante : $|\sum_k \nu_{i,k}\epsilon_k \Re(b_k)| \neq 0 \Longrightarrow \|\nu_i^*\| > N > M\cdot$

Démonstration. Considérons ν_i tel que $|\sum_k \nu_{i,k}\epsilon_k \Re(b_k)| \neq 0$.
Soit il existe i_0 tel que $|\nu_{i,i_0}| > N2^{(\frac{n+l}{2}-1)/2}$ et dans ce cas $\|\nu_i\| > N2^{(\frac{n+l}{2}-1)/2}$. Or $\|\nu_i\| \leq 2^{\frac{i-1}{2}}\|\nu_i^*\|$ (d'après les propriétés des bases LLL réduites). D'où $N2^{(\frac{n+l}{2}-1)/2}2^{\frac{-i+1}{2}} < \|\nu_i^*\|$ et donc $N \leq N2^{(\frac{n+l}{2}-i)/2} < \|\nu_i^*\|$.
Soit pour tout indice k nous avons $|\nu_{i,k}| < N2^{(\frac{n+l}{2}-1)/2}$. Dans ce cas $|\sum_k \nu_{i,k}\epsilon_k \Re(b_k)| \geq \epsilon_{N2^{(\frac{n+l}{2}-1)/2}}$. D'où :

$$|\sum_k \nu_{i,k}C\epsilon_k \Re(b_k)| \geq C\epsilon_{N2^{(\frac{n+l}{2}-1)/2}} \quad (\sharp).$$

De plus $|C\epsilon_k\Re(b_k) - \lfloor C\epsilon_k \Re(\tilde{b_k})\rceil| \leq \frac{5}{2}$ car $C\eta \leq 1$ et $\epsilon_k \leq 2$. Donc :

$$|\sum_k \nu_{i,k}C\epsilon_k \Re(b_k) - \sum_k \nu_{i,k}\lfloor C\epsilon_k\Re(\tilde{b_k})\rceil| \leq \frac{n+l}{2}N2^{(\frac{n+l}{2}-1)/2}\,\frac{5}{2} \quad (\sharp\sharp).$$

De plus

$$|\sum_k \nu_{i,k}\lfloor C\epsilon_k\Re(b_k)\rceil| \geq \left| |\sum_k \nu_{i,k}\lfloor C\epsilon_k\Re(b_k)\rceil - \sum_k \nu_{i,k}C\epsilon_k\Re(b_k)| - |\sum_k \nu_{i,k}C\epsilon_k\Re(b_k)| \right|.$$

Le second membre de l'inégalité ci-dessus est du type $||A| - |B||$ où $|A| \leq \frac{n+l}{2}N2^{(\frac{n+l}{2}-1)/2}\frac{5}{2}$ d'après ($\sharp\sharp$) et $|B| \geq C\epsilon_{N2^{(\frac{n+l}{2}-1)/2}}$ d'après ($\sharp$). La condition ($\star$) implique alors $||A| - |B|| = |B| - |A| \geq N2^{(\frac{n+l}{2}-1)/2}$. Donc nous obtenons :

$$|\sum_k \nu_{i,k}\lfloor C\epsilon_k\Re(b_k)\rceil| \geq N2^{(\frac{n+l}{2}-1)/2}.$$

Comme précédemment nous en déduisons que $\|\nu_i^*\| \geq N$. □

Les lemmes précédents nous permettent de conclure :

Lemme 33. *Avec les notations précédentes, si ($\star$) est vérifiée alors une exécution de LLL nous permet d'obtenir le réseau V.*

La constante C obtenue peut être énorme. En effet supposons $b_1 = 1$, $b_2 = 2$, $b_3 = 3 + \frac{1}{2^{2^n}}$. Comme $|b_3 - b_1 - b_2| = \frac{1}{2^{2^n}} \geq \epsilon_1$, il vient $\epsilon_{2^{(n+l-1)/2}M} \leq \epsilon_1 \leq \frac{1}{2^{2^n}}$, d'où $C \geq 2^{2^n}$. Le nombre d'opérations arithmétiques à effectuer pour obtenir une base LLL réduite est dans ce cas exponentiel en n.
Ce résultat n'est donc pas utilisable en pratique car il demande de connaître

$\epsilon_{2^{(n+l-1)/2}M}$, et, d'autre part la précision demandée n'est pas envisageable car beaucoup trop grande.
Nous pouvons alors être moins exigeant et souhaiter obtenir une constante C suffisamment grande pour faire décroître la dimension du réseau après une seule utilisation de LLL. Pour cela nous allons considérer le réseau $\mathcal{L}''$ définit par les vecteurs correspondant aux lignes de la matrice carrée suivante :

$$\mathcal{M}_{\mathcal{L}''} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \lfloor C\epsilon_1 \Re(\tilde{b_1}) \rceil \\ 0 & 1 & \cdots & 0 & \lfloor C\epsilon_2 \Re(\tilde{b_2}) \rceil \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & \lfloor C\epsilon_{\frac{n+l}{2}-1} \Re(\tilde{b}_{\frac{n+l}{2}-1}) \rceil \\ 0 & 0 & \cdots & 0 & \lfloor C\epsilon_{\frac{n+l}{2}} \Re(\tilde{b}_{\frac{n+l}{2}}) \rceil \end{pmatrix}.$$

Nous remarquons que nous pouvons passer facilement du réseau $\mathcal{L}''$ au réseau $\mathcal{L}'$. L'intérêt du réseau $\mathcal{L}''$ est que l'on peut facilement calculer son volume car la matrice $\mathcal{M}_{\mathcal{L}''}$ est carrée. Nous pouvons alors dans ce cadre obtenir le résultat suivant :

Lemme 34. *Avec les notations précédentes et en considérant le réseau $\mathcal{L}''$ nous avons : Si $C \geq \dfrac{(M2^{\frac{n+l}{4}})^{\frac{n+l}{2}} + 1}{|\epsilon_{\frac{n+l}{2}} \Re(\tilde{b}_{\frac{n+l}{2}})|}$ alors il suffit d'une utilisation de LLL pour faire décroître la dimension du réseau $\mathcal{L}$.*

Démonstration. Soit $\nu_1, \ldots, \nu_{\frac{n+l}{2}}$ une base LLL réduite de $\mathcal{L}''$. Nous avons

$$det(\mathcal{L}'') = |det(\mathcal{M}_{\mathcal{L}''})| = \prod_{k=1}^{\frac{n+l}{2}} \|\nu_i^*\|.$$

On remarque alors que si $det(\mathcal{L}'') \geq (M.2^{\frac{n+l}{4}})^{\frac{n+l}{2}}$ alors il existe un indice $\|\nu_{i_0}^*\| \geq M.2^{\frac{n+l}{4}}$. En reprenant les calculs de la preuve du lemme 29 page 86 (terminaison de l'algorithme), cela nous permet d'affirmer que dans ce cas certains vecteurs seront supprimés. La dimension du réseau $\mathcal{L}$ va donc diminuer.
Or si C vérifie la condition du lemme nous obtenons $|\lfloor C\epsilon_{\frac{n+l}{2}} \Re(\tilde{b}_{\frac{n+l}{2}}) \rceil| \geq (M.2^{\frac{n+l}{4}})^{\frac{n+l}{2}}$. Comme $|\lfloor C\epsilon_{\frac{n+l}{2}} \Re(\tilde{b}_{\frac{n+l}{2}}) \rceil| = |det(\mathcal{M}_{\mathcal{L}''})| = det(\mathcal{L}'')$, cela nous permet de conclure. $\square$

Dans le cas d'un polynôme de degré 120, avec quatre b_i réels et $b_i \approx 1$, cela donne : $C \geq M^{62}.2^{1922} + 1$. Une fois de plus la constante obtenue est trop grande pour être utilisée en pratique.

3.4.3 Comment certifier les résultats

Dans notre algorithme ou tout autre se basant sur l'étude des nombres b_i, la connaissance de l'erreur commise sur ces nombres est cruciale. Dans sa thèse

D. Rupprecht a commencé ce travail d'analyse d'erreur et celui-ci fait l'hypothèse suivante : $|f(x+\epsilon) - f(x)| = |f'(x)||\epsilon|$, où f est une fonction rationnelle. Les termes d'ordre supérieur à 2 sont donc négliger mais cela suffit pour obtenir un ordre de grandeur convenable pour l'erreur.

Ici nous allons donner un moyen de certifier l'erreur sur les b_i. Pour cela nous allons voir comment calculer P_b le polynôme minimal des b_i. (P_b est le polynôme unitaire de $\mathbb{Q}[T]$ ayant les b_i pour racines). Une fois que nous avons le polynôme P_b nous pouvons calculer les nombres b_i à l'aide des formules vues en 1.5.3 page 33 puis évaluer l'erreur sur ces nombres à l'aide d'un résultat type α-théorème (voir théorème 26 page 96). (Nous ne calculons pas directement les racines de P_b car nous devons pouvoir associer à chaque y_i le nombre b_i correspondant).
Nous verrons que cette démarche possède un avantage et un inconvénient. L'inconvénient est que le temps nécessaire pour le calcul de P_b peut être grand comparé au reste de l'algorithme. L'avantage est que l'estimation de l'erreur est très fine.
Nous pourrions aussi envisager d'utiliser l'arithmétique d'intervalles afin de certifier nos calculs. La méthode proposée ici n'est donc peut être pas optimale mais des améliorations peuvent être envisagées comme nous le verrons en 3.5.3.

Calcul de P_b

Tout d'abord nous rappelons certaines notations :
$P(X,Y) = Y^n + A_{n-1}(X)Y^{n-1} + \ldots + A_0(X)$, $Disc(X) = Res_Y(P(X,Y), \frac{\partial P}{\partial Y}(X,Y))$, $Disc(0) \neq 0$. Nous avons donc n points distincts y_i dans la fibre au dessus de 0.
Nous avons deux polynômes $H(X,Y)$ et $G(X,Y)$ appartenant à $\mathbb{Q}[X,Y]$ tels que :

$$Disc(X) = H(X,Y)P(X,Y) + G(X,Y)\frac{\partial P}{\partial Y}(X,Y)\,(\star).$$

Nous considérons la fraction rationnelle (vue page 33) $b(X,Y)$ telle que $b(0,y_i) = b_i$, et on pose

$$b(X,Y) = \frac{N(X,Y)}{(\frac{\partial P}{\partial Y}(X,Y))^3},\ \text{où } N(X,Y) \in \mathbb{Q}[X,Y].$$

Nous nous plaçons dans une situation générique, nous pouvons donc supposer que tous les b_i sont distincts ($\star\star$).
Enfin E désigne la $\mathbb{Q}$-algèbre suivante : $E = \frac{\mathbb{Q}[y]}{P(0,Y)}$.

Proposition 19. *Soient $b_k^{(i)} \in \mathbb{Q}$ tels que : $\overline{b(0,Y)}^i = \sum_{k=0}^{n-1} b_k^{(i)}\overline{Y}^k$. On obtient les coefficients de $P_b(T) = T^n + \mu_{n-1}T^{n-1} + \ldots + \mu_0 \in \mathbb{Q}[T]$ en résolvant le système :*

$$\begin{pmatrix} b_0^{(0)} & \cdots & b_0^{(n-1)} \\ b_1^{(0)} & \cdots & b_1^{(n-1)} \\ \vdots & \cdots & \vdots \\ b_{n-1}^{(0)} & \cdots & b_{n-1}^{(n-1)} \end{pmatrix} \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{n-1} \end{pmatrix} = \begin{pmatrix} b_0^{(n)} \\ b_1^{(n)} \\ \vdots \\ b_{n-1}^{(n)} \end{pmatrix}.$$

Démonstration. E est un $\mathbb{Q}$ espace vectoriel de dimension n, de base $\overline{1}$, $\overline{Y}$,..., $\overline{Y}^{n-1}$.
L'égalité ($\star$) nous montre que $\overline{\frac{\partial P}{\partial Y}(0,Y)}$ est inversible dans E d'inverse $\frac{\overline{G(0,Y)}}{Disc(0)}$.
Ainsi $\overline{b(0,Y)}$ est bien définie dans E et

$$\overline{b(0,Y)} = \frac{\overline{N(0,Y)G(0,Y)}}{Disc(0)}.$$

Comme $\overline{1}$, $\overline{b(0,Y)}$, ..., $\overline{b(0,Y)}^{n-1}$ sont liés, il existe une combinaison linéaire telle que :

$$\sum_{i=0}^{n} \lambda_i \overline{b(0,Y)}^i = \overline{0}.$$

Considérons alors le polynôme $p(T) = \sum_{i=0}^{n} \lambda_i T^i \in \mathbb{Q}[T]$. Ce polynôme vérifie $p(b_j) = \sum_{i=0}^{n} \lambda_i b(0,y_j)^i = 0$ pour $j = 1,\ldots,n$, car y_j est une racine de $P(0,Y)$. $p(T)$ est donc un polynôme de degré n non nul dont les racines sont les b_j. On pose alors :

$$P_b(T) = T^n + \sum_{i=0}^{n-1} \mu_i T^i, \text{ où } \mu_i = \frac{\lambda_i}{\lambda_n}.$$

Nous recherchons donc à présent les μ_i. Ces nombres vérifient $\sum_{i=0}^{n-1} \mu_i \overline{b(0,Y)}^i = \overline{b(0,Y)}^n$.
On pose alors $\overline{b(0,Y)}^i = \sum_{k=0}^{n-1} b_k^{(i)} \overline{Y}^k$.
On obtient donc : $\sum_{i=0}^{n-1} \mu_i \sum_{k=0}^{n-1} b_k^{(i)} \overline{Y}^k = \sum_{k=0}^{n-1} b_k^{(n)} \overline{Y}^k$.
D'où $\sum_{i=0}^{n-1} \mu_i b_k^{(i)} = b_k^{(n)}$, ce qui donne le système annoncé. □

Remarque : Nous pouvons supprimer la dernière colonne de la matrice car nous savons que $\mu_{n-1} = 0$, puisque $\sum_{i=1}^{n} b_i = 0$.

Estimation de l'erreur

Ici nous rappelons un théorème tiré de [Yak],[WH90].

Théorème 26. *Soient E, F deux espaces de Banach réels ou complexes et $f : E \to F$ une fonction analytique. Soit $x_0 \in E$ tel que $D(f(x_0))$ est inversible.*
On pose : $\beta = \|Df(x_0)^{-1}f(x_0)\|$, $\gamma = \sup_{k\geq 2} \|\frac{Df(x_0)^{-1}D^k f(x_0)}{k!}\|^{\frac{1}{k-1}}$, $\alpha = \beta.\gamma$, $h(t) = \beta - 2t + \frac{t}{1-\gamma t}$.
On définit la suite de Newton : $t_0 = 0$, $t_{k+1} = t_k - \frac{h(t_k)}{h'(t_k)}$.
Si $\alpha < 3 - 2\sqrt{2}$ alors

1. *La fonction $h(t)$ a deux racines positives $t_- < t_+$, et la suite $(t_k)_{k\geq 0}$ converge quadratiquement vers t_-.*
2. *Il existe un zéro x^* de la fonction analytique $f(x)$ dans la boule ouverte $B(x_0, t_-)$.*

3. *La suite de Newton* $x_{k+1} = x_k - Df(x_k)^{-1}f(x_k)$, $k \geq 0$ *est bien définie, et nous avons pour* $k \geq 0$: $\|x_{k+1} - x_k\| \leq t_{k+1} - t_k$ *et* $\|x_k - x^*\| \leq t_- - t_k$.

Ce théorème appliqué au polynôme P_b nous permet donc de voir qu'à l'aide de la méthode de Newton nous pouvons obtenir une "approximation certifiée" des b_i. Comme nous sommes obligés de calculer les y_i afin de pouvoir entamer la remontée de Hensel, nous pouvons calculer les b_i à partir des y_i. Dans ce cas nous avons une approximation $\tilde{b}_i$, et, l'erreur commise est majorée de la façon suivante :

$$|\tilde{b}_i - b_i| < t_-.$$

Nous remarquons que dans notre cas $f := P_b$, $x_0 := \tilde{b}_i$ et donc γ est calculable. C'est-à-dire, ici la borne supérieure est un maximum puisque f est un polynôme.

Remarque :

La méthode présentée ci-dessus nous permet d'estimer de manière rigoureuse l'erreur commise sur les nombres b_i. Cependant cette certification demande de calculer P_b. Or nous allons voir sur un exemple que ce calcul peut s'avérer très coûteux.

Nous avons appliqué notre algorithme *Fac-knap* à un polynôme de degré 30 possédant 5 facteurs absolument irréductibles.
Afin d'obtenir les $\tilde{b}_i$ à une précision de l'ordre de $\eta = 10^{-26}$, nous avons calculé les $\tilde{y}_i$ à la précision $10^{-46} = 10^{-20}\eta$. A l'aide du théorème 26 nous avons alors estimé l'erreur sur les nombres $\tilde{b}_i$. Dans ce cas les nombres $\tilde{b}_i$ étaient connus à 10^{-56} prés alors qu'une précision de $\eta = 10^{-46}$ était suffisante pour certifier les calculs. Cela signifie deux choses : la précision sur les $\tilde{y}_i$ étaient supérieure à 10^{-46}, et les calculs n'ont pas dégradé cette précision. Cependant le temps nécessaire pour calculer le polynôme P_b est de 8 secondes. Or sans le calcul de P_b l'algorithme ne prend que 0.9 secondes.
Le même type de phénomène est apparu sur tous les exemples testés. C'est-à-dire : si nous effectuons les calculs à la précision $10^{-20}\eta$ alors nous obtenons les b_i avec une précision de l'ordre de η, mais le calcul de P_b est très long. C'est pourquoi dans l'implémentation actuelle nous effectuons les calculs à la précision $10^{-20}\eta$, mais nous ne calculons pas P_b afin de certifier l'erreur sur les b_i. Nous proposons en 3.5.3 une démarche pouvant permettre de certifier les calculs sans calculer P_b.

3.4.4 A propos des nombres p-adiques

Dans notre algorithme nous effectuons dans un premier temps les calculs de manière approchée dans $\mathbb{C}$. Cela nous permet d'obtenir rapidement des valeurs approchées $\tilde{b}_i$. Le problème est qu'ensuite nous devons retrouver l'expression exacte d'un facteur. Pour cela nous devons gérer l'erreur commise sur les $\tilde{b}_i$. Nous avons vu que cette étude de l'erreur est assez complexe. Une question se pose alors pourquoi ne pas utiliser les nombres p-adiques ?

Voyons ce qu'il se passe lorsque $P(0,Y)$ n'a que des racines simples.
Nous devons trouver un nombre premier p tel que toutes les racines de $P(0,Y)$ appartiennent à $\mathbb{Q}_p$ ou à une extension finie de $\mathbb{Q}_p$. Pour obtenir un tel nombre p nous pouvons calculer le corps de décomposition de $P(0,Y)$ et ensuite calculer f le polynôme minimal d'un élément primitif de ce corps. Ensuite nous considérons un nombre premier p ne divisant pas le discriminant de f. Toutes les racines de $P(0,Y)$ se trouvent alors dans une extension finie de $\mathbb{Q}_p$. Cette démarche n'est pas réaliste car le calcul du corps de décomposition s'avérera impraticable lorsque le degré de P sera élevé (le degré de f pouvant être égal à $n!$).

Nous pouvons essayer alors une approche probabiliste. Nous allons chercher p "au hasard" tel que $P(0,Y)$ ait n racines distinctes dans $\mathbb{F}_p$. Ainsi après remontée de Hensel nous aurons n racines de $P(0,Y)$ dans $\mathbb{Q}_p$. Ces racines sont tout simplement une expression p-adique des y_i. Donc à partir de ces nombres nous pouvons obtenir les b_i. Le problème est donc de trouver un nombre premier p tel que $P(0,Y)$ ait n racines distinctes dans $\mathbb{F}_p$. Or de tels nombres sont rares. Nous allons à présent énoncer le théorème de Čebotarev qui nous dit que la probabilité d'obtenir un tel p en le prenant au hasard est très faible (voir [Lan94], [Neu99], [vzGG03]) .

Théorème 27 (Čebotarev). *Soient $f(X) \in \mathbb{Z}[X]$ un polynôme de degré n, $\mathcal{D}(f)$ le corps de décomposition de f sur $\mathbb{Q}$, $\mathbb{P}$ l'ensemble des nombres premiers, $\mathcal{P}(f)$ l'ensemble des p premiers tels que $f \mod p$ ait n racines distinctes dans $\mathbb{F}_p$. De plus on note :*

$$d(\mathcal{P}(f)) = \lim_{x \to +\infty} \frac{|\{p \in \mathcal{P}(f) \,|\, p \leq x\}|}{|\{p \in \mathbb{P} \,|\, p \leq x\}|}.$$

Sous ces hypothèses nous avons :

$$d(\mathcal{P}(f)) = \frac{1}{[\mathcal{D}(f) : \mathbb{Q}]}.$$

Ici $d(\mathcal{P}(f))$ représente la densité des $p \in \mathcal{P}(f)$, autrement dit la "probabilité" pour un nombre premier d'être dans $\mathcal{P}(f)$. Lorsque f est irréductible nous avons $n \leq [\mathcal{D}(f) : \mathbb{Q}] \leq n!$ nous voyons donc qu'une approche probabiliste est elle aussi vouée à l'échec. Cela explique pourquoi nous ne pouvons pas utiliser les nombres p-adiques dans notre situation.

3.5 Améliorations possibles

3.5.1 Décroissance des $\|b_i^*\|$

Il se peut que dans certains cas nous ayons $\|w_1^*\| \leq M, \ldots, \|w_{k_1}^*\| \leq M$, $\|w_{k_1+1}^*\| > M$, et, qu'il existe un indice i_0 tel que $\|w_{k_1+1+i_0}^*\| \leq M$ pour $i_0 > 0$. Dans ce cas l'indice k_0 du lemme 22 est tel que $k_0 \geq k_1+1+i_0$, mais nous pouvons suspecter que $V \subset Vect_{\mathbb{Z}}(w_1, \ldots, w_{k_1})$. Nous pouvons alors essayer de poursuivre l'algorithme avec le réseau $Vect_{\mathbb{Z}}(w_1, \ldots, w_{k_1})$ au lieu du réseau $Vect_{\mathbb{Z}}(w_1, \ldots, w_{k_1}, \ldots, w_{k_0})$.

3.5.2 Simplification d'une extension

Après l'étape de reconnaissance exacte d'un facteur P_1 absolument irréductible, nous avons l'extension $\mathbb{Q}[\alpha]$ de $\mathbb{Q}$, où $\mathbb{Q}[\alpha]$ est le corps engendré par tous les coefficients de P_1. Ce corps est donné par f_α le polynôme minimal de α.
La représentation de $\mathbb{Q}[\alpha]$ a son importance dans le reste de l'algorithme. En effet plus les coefficients de f_α seront grands, plus les calculs effectués dans $\mathbb{Q}[\alpha]$ seront lents. Donc afin d'obtenir une remontée de Hensel efficace dans notre algorithme, nous pouvons essayer de simplifier f_α.
Dans son livre [Coh93] H. Cohen propose l'algorithme POLRED pour obtenir un polynôme minimal f_β plus "simple" permettant de représenter $\mathbb{Q}[\alpha]$. Nous allons brièvement présenter cet algorithme.

Définition 26. *Soient $f \in \mathbb{C}[T]$, et α_i ses racines complexes comptées avec multiplicitées. On appelle Taille de f le nombre suivant :*

$$\textit{Taille}(f) = \sum_i |\alpha_i|^2.$$

L'algorithme POLRED consiste à trouver un polynôme de petite Taille. L'idée est qu'un polynôme avec une Taille petite aura de "petites" racines donc de petits coefficients.
Soient $\omega_1, \dots, \omega_s$ une base entière de $\mathcal{O}_{\mathbb{Q}[\alpha]}$ (voir théorème 22 page 56), nous avons donc pour tout $x \in \mathcal{O}_{\mathbb{Q}[\alpha]}$, $x = \sum_{i=1}^{s} z_i \omega_i$ avec $z_i \in \mathbb{Z}$. Nous notons $\sigma_1, \dots, \sigma_s$ les s $\mathbb{Q}$-homomorphismes de $\mathbb{Q}$ dans $\mathbb{C}$. Si x est un élément primitif, son polynôme minimal est :

$$f_x = \prod_{k=1}^{s} (T - \sum_{i=1}^{s} z_i \sigma_k(\omega_i)).$$

D'où :

$$\text{Taille}(f_x) = \sum_{k=1}^{s} |\sum_{i=1}^{s} z_i \sigma_k(\omega_i))|^2.$$

A présent nous rappelons une définition alternative d'un réseau : Un réseau est un $\mathbb{Z}$ module libre $\mathcal{L}$ de rang fini avec une forme quadratique définie positive q sur $\mathcal{L} \otimes \mathbb{R}$. Ici nous posons $\mathcal{L} = \mathbb{Z}^s$ et $q(z_1, \dots, z_s) = \sum_{k=1}^{s} |\sum_{i=1}^{s} z_i \sigma_k(\omega_i)|^2$. Nous pouvons alors en appliquant LLL à ce réseau, obtenir un polynôme avec une petite Taille.

Le problème dans cette démarche est que nous devons calculer une base entière de $\mathcal{O}_{\mathbb{Q}[\alpha]}$. Nous avons vu au chapitre 2 que cela est coûteux car il faut factoriser un grand nombre entier. M van Hoeij et A. Novocin ont proposé, à ISSAC'04 dans un poster, une approche s'adaptant parfaitement à notre situation. Ici nous prenons en entrée α un entier algébrique primitif définissant $\mathbb{Q}[\alpha]$, mais aussi d'autres éléments $\beta_1, \dots, \beta_n$ de $\mathcal{O}_{\mathbb{Q}[\alpha]}$. M van Hoeij et A. Novocin calculent alors le $\mathbb{Z}$-module libre $\mathbb{Z}[\alpha, \beta_1, \dots, \beta_n] \subset \mathcal{O}_{\mathbb{Q}[\alpha]}$, et appliquent la méthode précédente à une $\mathbb{Z}$-base de $\mathbb{Z}[\alpha, \beta_1, \dots, \beta_n]$. Nous obtenons donc des résultats peut être moins bons qu'avec une base entière mais en un temps plus raisonnable. Nous remarquons que plus $\mathbb{Z}[\alpha, \beta_1, \dots, \beta_n]$ sera gros, donc proche de $\mathcal{O}_{\mathbb{Q}[\alpha]}$, meilleur seront les résultats.

Pour le moment nous utilisons la fonction OptimizedReprezentation qui est la version de POLRED dans Magma. Avec cette fonction les résultats sont variables : parfois, le temps gagné dans une remontée de Hensel avec une bonne représentation de $\mathbb{Q}[\alpha]$ est perdu lors de la simplification de $\mathbb{Q}[\alpha]$ (voir exemple suivant). La méthode van Hoeij/Novocin permettrait donc d'améliorer nos résultats.

Exemples :
Ici, nous allons présenter deux exemples pour mettre en évidence le rôle et les limites de la fonction *OptimizedRepresentation*. Les temps de calculs présentés ici ont été obtenus en utilisant MagmaV2.8-1 sur un ordinateur portable (mobile AMD Athlon XP 2000+ (1656 Mhz), 200 Mo).

• Nous avons essayé d'obtenir la factorisation absolue du polynôme $n150s5$ (se trouvant sur la page http ://math.unice.fr/∼cheze/) la première fois en simplifiant l'extension $\mathbb{Q}[\alpha]$ à l'aide de la fonction *OptimizedRepresentation*, la seconde sans effectuer de simplification. Le polynôme $n150s5$ est de degré 150 et possède 5 facteurs absolument irréductibles.
Le polynôme minimal obtenu pour α est :

$$\begin{aligned} f_\alpha(t) \;=\;& t^5 + 4243377149907973450 4t^4 - 483778029422062355273240 8t^3 \\ & +345902383544536601790542321536t^2 + 2770985685407406002243342586 72t \\ & +577138790619657436414868527 68. \end{aligned}$$

Après simplification nous obtenons le polynôme minimal d'un élément β tel que $\mathbb{Q}[\alpha] = \mathbb{Q}[\beta]$, et, le polynôme minimal de β est :

$$f_{beta}(t) = t^5 + t^4 - 10t^3 + 4t^2 + 10t + 3.$$

Cette simplification a nécessité 1164.141 secondes, et le temps total pour obtenir les facteurs absolument irréductibles en effectuant cette simplification a été de 1840 secondes. La remontée de Hensel a pris 51.78 secondes.
Si nous ne simplifions pas l'extension nous obtenons la factorisation absolue en 1397 secondes, et la remontée de Hensel a pris 562 secondes.

• Nous avons répété cette expérience avec le polynôme $n90s10$, de degré 90, et, avec 10 facteurs absolument irréductibles.

Le polynôme obtenu pour α est :

$$\begin{aligned} f_\alpha \;=\;& t^{10} - 51604t^9 + 2848072861t^8 + 10293297234t^7 \\ & -123081797465t^6 - 1089968029562t^5 + 3347520167456t^4 \\ & +24296013545928t^3 + 5996817995306t^2 - 236381437784097t \\ & +40300591439925. \end{aligned}$$

Après simplification nous obtenons :
$f_\beta(t) = t^{10} + 3t^9 - 8t^8 - 44t^7 - 37t^6 + 131t^5 + 395t^4 + 491t^3 + 328t^2 + 118t + 25.$
Cette simplification a nécessité 13.731 secondes, et le temps total pour obtenir les facteurs absolument irréductibles en effectuant cette simplification a été de 62.359 secondes. La remontée de Hensel a pris 4.87 secondes.
Si nous ne simplifions pas l'extension nous obtenons la factorisation absolue en 330.759 secondes, et la remontée de Hensel a pris 250.7 secondes.

3.5.3 Etude de l'erreur

En 3.4.3 nous avons donné un moyen permettant de certifier l'erreur sur les nombres $\tilde{b}_i$. Néanmoins cette méthode nécessite le calcul de P_b : le polynôme minimal des b_i sur $\mathbb{Q}$, et ce calcul peut s'avérer coûteux. Nous allons donner ici une piste conduisant à une certification de l'erreur $|b_i - \tilde{b}_i|$ sans passer par le calcul de P_b. L'idée que nous allons présenter repose sur l'utilisation du théorème 28 qui est un théorème d'inversion locale effectif (voir le théorème 2.1 dans [GLSY]). Avant d'énoncer ce théorème nous devons poser quelques définitions :

Définition 27. *On note $\mathbb{R}\{t\}$ l'anneau des séries convergentes. Nous avons un ordre partiel sur $\mathbb{R}\{t\}$ noté $\leq$. Cet ordre est défini ainsi :*

Soient $F, G \in \mathbb{R}\{t\}$, si $F^{(k)}(0) \leq G^{(k)}(0)$ pour tout $k \geq 0$ alors $F \leq G$.

Théorème 28. *Soit U un ouvert de $\mathbb{C}^n$, $f :\longrightarrow \mathbb{C}^n$ une application analytique et $\zeta \in U$ un point tel que $Df(\zeta)$ soit inversible. Soient $\sigma \geq \|Df(\zeta)^{-1}\|$, $\gamma \geq \sup_{k\geq 2} \|\dfrac{Df(x_0)^{-1}D^kf(x_0)}{k!}\|^{\frac{1}{k-1}}$, B_f la boule centrée en ζ de rayon $\dfrac{1-\sqrt{2}/2}{\gamma}$. On suppose $B_f \subset U$. Alors il existe une unique application g vérifiant les propriétés suivantes :*

1. *g est définie et analytique dans la boule de centre $f(\zeta)$ et de rayon $\dfrac{1}{(3+2\sqrt{2})\sigma\gamma}$,*
2. *$g(B_g) \subseteq B_f$,*
3. *$f \circ g(b) = b$, pour tout $b \in B_g$,*
4. *$\sum_{k\geq 0}^{\infty} \|D^k(g-\zeta)(f(\zeta))\| \dfrac{t^k}{k!} \leq \dfrac{\sigma t}{1-(3+2\sqrt{2})\sigma\gamma t}$.*

A présent nous allons voir comment nous ramenons notre problème à un problème d'inversion locale effectif.

On pose $z_i = \dfrac{P(x_0, \tilde{y}_i)}{\frac{\partial P}{\partial Y}(x_0, \tilde{y}_i)}$, et on suppose que z_i est bien défini. Autrement dit, on suppose que le dénominateur ne s'annule pas, ce qui est le cas si $\tilde{y}_i$ est suffisamment proche de y_i. Afin d'effectuer le lien avec le problème de l'inversion locale nous introduisons la fonction Σ_i suivante :

$$\begin{array}{rcl} \Sigma_i : U_i & \longrightarrow & V_i \\ (x,y) & \longmapsto & \Big(x, \Big(\dfrac{\partial P}{\partial Y}(x_0, \tilde{y}_i)\Big)^{-1} P(x,y)\Big), \end{array}$$

où U_i est un voisinage de $(x_0, \tilde{y}_i)$ et V_i est un voisinage de (x_0, z_i).
Cette application est inversible localement et nous notons $\phi_i = (\phi_{i,1}, \phi_{i,2})$ cet inverse. Nous avons alors $P \circ \phi_i \circ \Sigma_i = P$ d'où : $P \circ \phi_i \circ \Sigma(x, \tilde{y}_i) = P(x_0, \tilde{y}_i)$. Il en découle :

$$P(x, \phi_{i,2}(x, z_0)) = P(x_0, \tilde{y}_i).$$

Or la série que nous calculons en pratique est :

$$\tilde{\varphi}_i(X) = \tilde{y}_i + \tilde{a}_i(X - x_0) + \tilde{b}_i(X - x_0)^2 + \dots .$$

Cette série vérifie l'équation $P(X, .) = P(x_0, \tilde{y}_i)$. Nous en déduisons donc :

$$\tilde{\varphi}_i(x) = \phi_{i,2}(x, z_0).$$

De même il vient :

$$\varphi_i(x) = \phi_{i,2}(x, 0).$$

Nous avons donc ramené notre problème de la majoration de $|b_i - \tilde{b}_i|$ à celui de la majoration de

$$(*) \qquad \Big|\frac{\partial^2 \phi_{i,2}}{\partial y^2}(x_0, z_0) - \frac{\partial^2 \phi_{i,2}}{\partial y^2}(x_0, 0)\Big|.$$

Or en appliquant le théorème 28 à notre situation cela donne :

$$\sum_{k \geq 0}^{\infty} \|D^k(\phi_i - (x_0, \tilde{y}_i))(x_0, z_i)\| \frac{t^k}{k!} \leq \frac{\sigma t}{1 - (3 + 2\sqrt{2})\sigma\gamma t}.$$

A l'aide de l'inégalité précédente et des propriétés sur les séries majorantes nous pouvons envisager d'obtenir une majoration de l'expression $(*)$ à l'aide de σ, γ, et z_i.

Un tel procédé nous permettrait alors de certifier l'erreur commise sur les $\tilde{b}_i$ en n'évaluant que des dérivées de P en $(x_0, \tilde{y}_i)$, et surtout sans avoir besoin de calculer P_b. Cette stratégie pourrait donc nous permettre de certifier rapidement nos calculs.

3.6 Temps de calculs et exemples

Dans cette section nous allons comparer différents algorithmes de factorisation absolue. Les tests ont été effectués sur une machine du projet GALAAD à l'INRIA Sophia Antipolis (Intel Pentium 4, 3.40 Ghz, 1 Go).

Dans un premier temps nous allons comparer l'algorithme *Fac-Knap*, avec l'algorithme Galligo/Rupprecht, l'algorithme de S. Gao, l'algorithme SVW, et l'algorithme de Maple 5 sur des polynômes de petits degrés. Ensuite nous comparerons l'algorithme *Fac-Knap* à l'algorithme Galligo/Rupprecht sur des exemples de degrés élevés.
L'algorithme *Fac-Knap* a été programmé en utilisant MagmaV2.11-8. Les parties dominantes des algorithmes Galligo/Rupprecht et Gao ont elles aussi été programmées en MagmaV2.11-8.
Pour l'algorithme SVW, nous avons utilisé une des fonctionnalités de PHC (Polynomial Homotopy Continuation). PHC est un programme développé par J. Verschelde (voir http ://www2.math.uic.edu/∼jan/).

Les comparaisons ont été effectuées sur les polynômes F_1, F_2, F_3 (de degré respectif 14, 20, 30) donnés ci-dessous. Ces polynômes ont été choisis, car leurs expressions sont suffisamment petites pour pouvoir être présentées. Le polynôme F_2 est un polynôme qui a déjà servi de test dans les thèses de J.F Ragot et de D. Rupprecht.

$$\begin{aligned} F_1 &= Y^{14}+18X^3Y^4+54X^5Y^2+6Y^7+6Y^{11}X^3+18Y^9X^5+7X^6Y^8+38X^8Y^6 \\ &\quad +49X^{10}Y^4-28X^3Y^7-4X^4Y^4-112X^5Y^5-16X^6Y^2-28Y^3X-98Y^6 \\ &\quad -2X^2+9 \end{aligned}$$

$$\begin{aligned} F_2 &= Y^{20}+X^{20}+5X^{16}Y^4+Y^5X^5+9X^8Y^4-6X^{14}Y^2-18X^{10}Y^6-18X^6Y^{10} \\ &\quad +10X^{12}Y^8+10X^8Y^{12}+5Y^{16}X^4+9Y^8X^4-6Y^{14}X^2 \end{aligned}$$

$$\begin{aligned} F_3 &= Y^{30}+2Y^{29}X+126X^{13}Y^2+18XY^{14}+18Y^{15}+90Y^3+14Y^{17}X^{13}-X^2Y^{28} \\ &\quad +14X^{14}Y^{16}+6XY^{17}+49X^{26}Y^4+70X^{13}Y^5+10Y^{18}+23Y^6+81 \end{aligned}$$

Dans le tableau suivant nous avons résumé les résultats obtenus.

- ***knap*** est le temps en secondes pour obtenir la factorisation absolue avec l'algorithme *Fac-Knap* en utilisant MagmaV2.11-8.
- **$\boldsymbol{\Sigma}$-*Rup*** désigne le temps en secondes pour obtenir toutes les sommes nulles dans l'algorithme Galligo/Rupprecht en utilisant MagmaV2.11-8. Ici le temps ne correspond qu'à la partie de complexité $O(2^{n/4})$. Ce n'est pas le temps total de l'algorithme.

- ***Gao*** désigne le temps en secondes pour effectuer la première étape (la plus coûteuse) de l'algorithme de S. Gao en MagmaV2.11-8. Cette première étape correspond à la résolution d'un système linéaire.
- ***Phc-Rup*** désigne le temps en secondes pour obtenir la factorisation absolue en utilisant la méthode Galligo/Rupprecht avec PHC.
- ***Phc-SVW*** désigne le temps en secondes pour obtenir la factorisation absolue en utilisant la méthode SVW avec PHC.
- ***Maple*** désigne le temps en secondes pour obtenir la factorisation absolue avec Maple 5.
- **n** désigne le degré du polynôme.
- **s** désigne le nombre de facteurs absolument irréductibles.
- **Ex** désigne le nom de l'exemple.

Ex	n	s	*knap*	Σ-*Rup*	*Gao*	*Phc-Rup*	*Phc-SVW*	*Maple*
F_1	14	2	0.13	0	0.8	4.81	33.83	9.11
F_2	20	5	0.21	0	14.35	8.88	erreur	1713.90
F_3	30	2	0.5	0.23	68.9	42.6	erreur	-

REMARQUES :
Pour le polynôme F_2, la méthode Galligo/Rupprecht de PHC rend 16 facteurs linéaires, dont 8 d'entre eux ont une multiplicité 2. Ce résultat est donc incorrect. Aprés un autre essai nous obtenons au bout de 5.21 secondes, 11 facteurs dont un de degré 10. Ce résultat est donc lui aussi incorrect. Dans le tableau la mention *erreur* signifie que l'ordinateur nous a rendu un message d'erreur.
Pour le polynôme F_2, en utilisant Maple 8 sur le serveur du laboratoire de mathématiques de l'université de Nice (Intel Xeon 3.06 Ghz, 2Go), nous avons obtenu le résultat aprés 999 secondes.
Le temps de calculs des sommes nulles dans l'algorithme Galligo/Rupprecht est négligeable lorsque les degrés des polynômes considérés sont inférieurs à 30.

L'objectif de l'algorithme *Fac-Knap* étant d'améliorer l'algorithme Galligo/Rupprecht, nous allons à présent comparer ces deux algorithmes sur des polynômes ayant des degrés plus élevés. Nous n'avons pas poursuivi la comparaison avec l'algorithme de S. Gao car nous obtenons un message d'erreur (aprés 1100 secondes) lorsque nous l'appliquons à un polynôme de degré 50. Le tableau ci-dessous présente les temps de calculs obtenus.

Les notations précédentes sont conservées.
l désigne le nombre de b_i réels obtenus, et C la constante utilisée lors de la première réduction du réseau.

n	s	l	C	*knap*	Σ-*Rup*
30	5	2	10^{26}	0.64	0.06
48	3	2	10^{44}	4.07	48.331
50	5	6	10^{47}	3.24	345.771
100	1	4	10^{105}	22.51	erreur
100	2	4	10^{105}	162.03	erreur
100	5	2	10^{105}	362.89	erreur
120	6	4	10^{130}	61.36	-
200	10	4	10^{207}	828.95	-

L'algorithme *Fac-knap* se compare donc avantageusement à l'algorithme de D. Rupprecht.

Remarques :

– Dans tous les exemples nous avons pris $x_0 = 1$, et une seule réduction a suffi pour obtenir la factorisation absolue.
– Dans l'exemple correspondant à $n = 100$ et $s = 5$ l'étape dominante de l'algorithme est la simplification de l'extension. Nous avons décrit ce type de phénomène à la page 99.
– Le polynôme correspondant à $n = 100$ et $s = 1$ est un polynôme que nous avons construit de manière aléatoire en prenant des coefficients au hasard dans $[-10^{12}; 10^{12}]$.
– Pour le polynôme $n = 100$, $s = 2$ le temps mis pour obtenir une factorisation approchée modulo X^3 a été de 7.18 secondes. La partie "exacte" est donc largement dominante sur cet exemple. Par exemple, le temps nécessaire à la remontée de Hensel a été de 129 secondes. Il faut noter toutefois que pour ce polynôme la valeur moyenne des coefficients est de l'ordre de 10^{36}. Cela explique la "lenteur" des calculs exacts.
– Pour le polynôme de degré 120 (respectivement de degré 200) la valeur moyenne des coefficients est de l'ordre de 10^7 (respectivement de 10^{12}).
– Dans la version actuelle de *Fac-Knap* la remontée de Hensel est linéaire (voir la section 1.2.4). Dans un futur proche nous espérons remplacer cette étape par une remontée plus efficace.

Deuxième partie

Méthodes exactes

Chapitre 4

Un algorithme symbolique de factorisation absolue

Ce chapitre correspond à un travail en cours en collaboration avec G. Lecerf [CL]. Une version plus détaillée de ce travail se trouve dans l'annexe C.

Nous avons vu en 1.3.2, et 1.4.6 la stratégie de factorisation "remonter-recombiner". Nous allons présenter dans ce chapitre un algorithme symbolique de factorisation absolue utilisant cette stratégie. Cet algorithme est basé sur le travail de G. Lecerf (voir [Lec04b], et section 1.4.6), et améliore celui de S. Gao (voir [Gao03], et section 1.4.5). Les améliorations proposées sont de deux types. Tout d'abord nous allons voir que nous pouvons obtenir le même espace vectoriel que celui utilisé par S. Gao en résolvant un système linéaire en n inconnues au lieu de $2n(n+1)$ inconnues. Nous obtiendrons au passage un algorithme pour tester l'irréductibilité absolue d'un polynôme. Ensuite nous proposerons une approche déterministe pour la partie probabiliste de l'algorithme de S. Gao.

Notations :
Dans ce chapitre nous supposerons $P(X,Y) \in \mathbb{K}[X,Y]$ avec $P(X,Y)$ irréductible dans $\mathbb{K}[X,Y]$ et de degré total n. Nous noterons $P_1, \ldots, P_s$ ses facteurs absolument irréductibles. De plus nous allons travailler sous les hypothèses suivantes :

(C) $\quad \mathbb{K}$ est un corps parfait de caractéristique 0, ou bien supérieure ou égale à $n(n-1)+1$.

$$(H) \quad \begin{cases} i)\ \deg_Y(P) = \deg(P) = n, \\ ii)\ Res_Y\Big(P, \dfrac{\partial P}{\partial Y}\Big)(0) \neq 0. \end{cases}$$

Nous remarquons que nous pouvons satisfaire l'hypothèse (H), en effectuant un changement générique linéaire de coordonnées dans P. De plus nous pouvons sans perte de généralité supposer que le coefficient du terme en Y^n dans P est 1.

4.1 Une approche “remonter-recombiner” pour la factorisation absolue

Avant de décrire la stratégie utilisée dans notre algorithme, voyons ce que la démarche “remonter-recombiner” vue en 1.4.6 pour l’algorithme de G. Lecerf donnerait si nous l’appliquions à $P(X,Y) \in \overline{\mathbb{K}}[X,Y]$. Pour cela rappelons certaines notations :

L’hypothèse (H) étant satisfaite nous pouvons écrire :

$$P(X,Y) = \prod_{i=1}^{n}(Y - \phi_i(X)) \text{ avec } \phi_i(X) \in \overline{\mathbb{K}}[[X]].$$

On note alors

$$\mathfrak{F}_i = Y - \phi_i(X), \text{ et } \hat{\mathfrak{F}}_i = \prod_{j=1, j\neq 1}^{s} \mathfrak{F}_j.$$

Ensuite à chaque facteur P_i absolument irréductible nous associons un 0-1 vecteur $\mu_{P_i} \in \{0,1\}^n$ tel que $P_i = \prod_{j=1}^{n} \mathfrak{F}_j^{\mu_{P_{i,j}}}$. Le théorème 16 vu en 1.4.6 nous dit alors que l’on peut obtenir les μ_{P_i} en étudiant le $\overline{\mathbb{K}}$ espace vectoriel suivant :

Définition 28. *On appelle $\overline{L_\sigma}$ le $\overline{\mathbb{K}}$ espace vectoriel suivant :*

$$\begin{aligned}\overline{L_\sigma} =& \{(l_1,\ldots,l_n; \overline{G}; \overline{H}) \in \overline{\mathbb{K}}^n \times \overline{\mathbb{K}}[X,Y]_{n-1} \times \overline{\mathbb{K}}[X,Y]_{n-1} \mid \\ &\overline{G} = \sum_{i=1}^{s} l_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial Y} \mod (X,Y)^\sigma, \\ &\overline{H} = \sum_{i=1}^{s} l_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial X} \mod (X,Y)^\sigma + (X)^{\sigma-1}\},\end{aligned}$$

où $\overline{\mathbb{K}}[X,Y]_{n-1}$ désigne l’ensemble des polynômes de $\overline{\mathbb{K}}[X,Y]$ de degré inférieur ou égal à $n-1$.

Dans notre cas particulier le théorème 16 dit alors :

Théorème 29. *Sous les hypothèses (C) et (H) nous avons pour tout $\sigma \geq 2n$,*

$$\overline{L_\sigma} = Vect_{\overline{\mathbb{K}}}((\mu_{P_i}, \hat{P}_i \frac{\partial P_i}{\partial Y}, \hat{P}_i \frac{\partial P_i}{\partial X}) \mid i \in \{1,\ldots,s\}),$$

où $\hat{P}_i = \prod_{j=1, j\neq i}^{s} P_j$.

Nous voyons alors que l’espace $\overline{L_\sigma}$ permet théoriquement d’obtenir la factorisation absolue de P. Mais cela pose un problème pratique : Comment obtenir les $\mathfrak{F}_i$? C’est à dire, comment effectuer une remontée de Hensel dans $\overline{\mathbb{K}}$? Une solution est

de calculer le corps de décomposition de $P(0, Y)$, et d'effectuer la remontée de Hensel dans ce corps. Or cela est inenvisageable car nous devrions travailler dans une extension pouvant être de degré $n!$. Nous pourrions aussi envisager d'appliquer la méthode Kaltofen/Traverso vue en 1.4.3 page 21 afin d'obtenir un corps $\mathbb{K}[\theta]$ contenant les coefficients de P_1. Nous pourrions ensuite appliquer la méthode de G. Lecerf pour obtenir la factorisation de P dans $\mathbb{K}[\theta][X, Y]$. Dans ce cas nous serions amené à résoudre un système linéaire à coefficients dans $\mathbb{K}[\theta]$.
Dans cette section nous allons présenter une stratégie où l'on se ramène à résoudre un système linéaire en n inconnues à coefficients dans $\mathbb{K}$. Pour cela nous allons fabriquer un espace vectoriel L_σ tel que $L_\sigma \otimes \overline{\mathbb{K}}$ soit isomorphe à $\overline{L_\sigma}$, et, nous devrons effectuer une remontée de Hensel dans une $\mathbb{K}$ algèbre de dimension n. Afin de présenter cette stratégie nous allons introduire certaines notations.

4.1.1 Définitions et notations

Définition 29. *Soit* $p(Y) = P(0, Y)$*, on note* $\mathbb{A}$ *la* $\mathbb{K}$*-algèbre* $\dfrac{\mathbb{K}[Y]}{(p(Y))}$*, et,* φ *l'image de* y *dans* $\mathbb{A}$*.*
Soit $f = f_0 + f_1\varphi + \ldots + f_{n-1}\varphi^{n-1} \in \mathbb{A}$*, on note alors* $\operatorname{coef}(f, \varphi^i) = f_i$*.*

Nous avons la propriété suivante :

Proposition 20. *Soient* $y_1, \ldots, y_n$ *les* n *racines distinctes de* $p(Y)$*. Nous avons alors le morphisme de* $\overline{\mathbb{K}}$*-espace vectoriel suivant :*

$$\begin{array}{cccc} \psi : & \overline{\mathbb{K}} \otimes \mathbb{A} & \longrightarrow & \overline{\mathbb{K}}^n \\ & \sum_{i=0}^{n-1} f_i\varphi^i & \longmapsto & \left(\sum_{i=0}^{n-1} f_i y_1^i, \ldots, \sum_{i=0}^{n-1} f_i y_n^i\right). \end{array}$$

Démonstration. Le polynôme $p(Y)$ possède n racines distinctes car l'hypothèse (H) est vérifiée. L'existence de l'application ψ est alors immédiate. ☐

Définition 30. *On note* V *la matrice de Vandermonde correspondant à l'application* ψ *pour la base canonique, c'est à dire :*

$$\begin{pmatrix} 1 & y_1 & \cdots & y_1^{n-1} \\ 1 & y_2 & \cdots & y_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_n & \cdots & y_n^{n-1} \end{pmatrix}.$$

Grâce à l'hypothèse (H) nous pouvons remonter à l'aide d'une remontée de Hensel (ou bien l'application d'un opérateur de Newton) la racine φ en une série ϕ dans $\mathbb{A}[[X]]$. Cela nous conduit à la définition suivante :

Définition 31. *On note :* $\phi(X)$ *l'élément de* $\mathbb{A}[[X]]$ *vérifiant*

$$\phi(X) = \varphi \mod X, \textit{ et, } P(X, \phi(X)) = 0.$$

On pose

$$\mathfrak{F}(X,Y) = Y - \phi(X) \in \mathbb{A}[[X]][Y],$$

et,

$$\hat{\mathfrak{F}}(X,Y) = \frac{P(X,Y)}{\mathfrak{F}(X,Y)} \in \mathbb{A}[[X]][Y].$$

Pour un élément

$$\mathfrak{G}(X,Y) = \sum_{v=0}^{K}\sum_{u=0}^{\infty}\sum_{i=0}^{n-1} g_{v,u,i}\varphi^i X^u Y^v \in \mathbb{A}[[X]][Y],$$

on pose :

$$\mathrm{coef}(\mathfrak{G}, \varphi^i) = \sum_{v=0}^{K}\sum_{u=0}^{\infty} g_{v,u,i} X^u Y^v,$$

et,

$$\mathrm{coef}(\mathfrak{G}, \varphi^i, X^u, Y^v) = g_{v,u,i}.$$

L'espace L_σ que nous allons définir va “remplacer” en pratique l'espace $\overline{L_\sigma}$.

Définition 32. *On appelle L_σ le $\mathbb{K}$ espace vectoriel suivant :*

$$\begin{aligned} L_\sigma =&\{(l_1,\dots,l_n; G; H) \in \mathbb{K}^n \times \mathbb{K}[X,Y]_{n-1} \times \mathbb{K}[X,Y]_{n-1} \mid \\ &G = \sum_{i=1}^{n} l_i \,\mathrm{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{i-1}) \mod (X,Y)^\sigma, \\ &H = \sum_{i=1}^{n} l_i \,\mathrm{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial X}, \varphi^{i-1}) \mod (X,Y)^\sigma + (X)^{\sigma-1}\}, \end{aligned}$$

où $\mathbb{K}[X,Y]_{n-1}$ désigne l'ensemble des polynômes de $\mathbb{K}[X,Y]$ de degré total inférieur ou égal à $n-1$.

Nous remarquons que les espaces L_σ sont construits sur le même modèle que les espaces $\overline{L_\sigma}$. Nous avons laissé $\frac{\partial \mathfrak{F}}{\partial Y}$ et $\frac{\partial \mathfrak{F}}{\partial X}$ afin de faire apparaître cette similarité ; mais nous pouvons simplifier ces expressions car $\frac{\partial \mathfrak{F}}{\partial Y} = 1$ et $\frac{\partial \mathfrak{F}}{\partial X} = -\phi'$.

4.1.2 Le théorème de remontée

Théorème 30 (Théorème de remontée). *Pour tout $\sigma \geq 2n$ nous avons :*

$$\overline{\mathbb{K}} \otimes L_\sigma = Vect_{\overline{\mathbb{K}}}\Big((\mu_i, \hat{P}_i\frac{\partial P_i}{\partial Y}, \hat{P}_i\frac{\partial P_i}{\partial X}) \mid i \in \{1,\dots,s\}\Big),$$

où

$$\hat{P}_i = \prod_{j=1, j\neq i}^{s} P_j,$$

et

$$\mu_i = \big(1, Tr_1(P_i(0,Y)), Tr_2(P_i(0,Y)), \ldots, Tr_{n-1}(P_i(0,Y))\big),$$

avec

$$Tr_j(P_i(0,Y)) = \sum_{P_i(0,y)=0} y^j.$$

Démonstration. La clef de la preuve réside dans l'isomorphisme de $\overline{\mathbb{K}}$ espaces vectoriels suivant :

$$\begin{array}{rcl} \Sigma : \overline{L_\sigma} & \longrightarrow & L_\sigma \otimes \overline{\mathbb{K}} \\ (\bar{l}, \overline{G}, \overline{H}) & \longmapsto & (V^t \bar{l}, \overline{G}, \overline{H}) \end{array}$$

où V^t désigne la transposée de la matrice V (voir la définition 30). Autrement dit, en notant $\bar{l} = (\overline{l_1}, \ldots, \overline{l_n})$, y_i les racines distinctes de $P(0,Y)$, et $l = (l_1, \ldots, l_n) = V^t \bar{l}$, nous avons : $l_i = \sum_{j=1}^n \overline{l_j} y_j^{i-1}$.
Si l'application Σ est bien définie, alors Σ sera clairement un isomorphisme de $\overline{\mathbb{K}}$ espace vectoriel. Comme $V^t \mu_{P_i} = \big(1, Tr_1(P_i(0,Y)), \ldots, Tr_n(P_i(0,Y))\big)$ nous aurons alors clairement démontré le théorème.
Montrons donc que Σ est bien définie :

Pour cela nous allons montrer que : $\begin{cases} \displaystyle\sum_{i=1}^n \overline{l_i} \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial Y} = \sum_{k=1}^n l_k \operatorname{coef}(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{k-1}), \\ \displaystyle\sum_{i=1}^n \overline{l_i} \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial X} = \sum_{k=1}^n l_k \operatorname{coef}(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial X}, \varphi^{k-1}). \end{cases}$

Tout d'abord nous rappelons que $\hat{\mathfrak{F}}(X,Y) = \dfrac{P(X,Y)}{Y - \phi(X)}$ et $\hat{\mathfrak{F}}_i(X,Y) = \dfrac{P(X,Y)}{Y - \phi_i(X)}$.
Ainsi si nous écrivons :

$$\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial Y} = \sum_{u=0}^{\infty} \sum_{v=0}^{n-1} \sum_{k=0}^{n-1} b_k^{(u,v)} \varphi^k X^u Y^v,$$

alors nous avons :

$$\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial Y} = \sum_{u=0}^{\infty} \sum_{v=0}^{n-1} \sum_{k=0}^{n-1} b_k^{(u,v)} y_i^k X^u Y^v.$$

De ce fait :

$$\begin{aligned}
\sum_{i=1}^{n} \overline{l_i}\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial Y} &= \sum_{i=1}^{n} \overline{l_i} \sum_{u=0}^{\infty}\sum_{v=0}^{n-1}\sum_{k=0}^{n-1} b_k^{(u,v)} y_i^k X^u Y^v \\
&= \sum_{u=0}^{\infty}\sum_{v=0}^{n-1}\sum_{k=0}^{n-1} (\sum_{i=1}^{n} \overline{l_i} y_i^k) b_k^{(u,v)} X^u Y^v \\
&= \sum_{u=0}^{\infty}\sum_{v=0}^{n-1}\sum_{k=1}^{n} l_k b_{k-1}^{(u,v)} X^u Y^v \\
&= \sum_{k=1}^{n} l_k \operatorname{coef}(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{k-1}).
\end{aligned}$$

Nous procédons de la même manière pour prouver la seconde égalité, ce qui achève la démonstration. □

A présent nous allons voir certaines conséquences de ce théorème. Tout d'abord nous rappelons que les polynômes $\hat{P}_i \frac{\partial P_i}{\partial Y}$, où $i = 1, \ldots, s$, sont $\overline{\mathbb{K}}$ linéairement indépendants (voir l'utilisation du lemme de Dedekind à la page 25). De plus comme $L_\sigma \subset L_\nu$ pour $\sigma \geq \nu$ nous obtenons d'aprés le théorème 30 le corollaire suivant :

Corollaire 10. *Si $\sigma \geq 2n$ alors nous avons $L_\sigma \otimes \overline{\mathbb{K}} = L_{2n} \otimes \overline{\mathbb{K}}$ et $\dim_{\overline{\mathbb{K}}} L_\sigma \otimes \overline{\mathbb{K}} = s$.*

De plus si L_σ est un $\mathbb{K}$ espace vectoriel de dimension r alors nous avons $L_\sigma \simeq \mathbb{K}^r$, d'où $L_\sigma \otimes \overline{\mathbb{K}} \simeq \mathbb{K}^r \otimes \overline{\mathbb{K}} \simeq \overline{\mathbb{K}}^r$. Ainsi nous avons montré :

Corollaire 11. *Si $\sigma \geq 2n$ alors $\dim_{\mathbb{K}} L_\sigma = s$ et $L_\sigma = L_{2n}$.*

Nous venons donc de voir que la dimension du $\mathbb{K}$ espace vectoriel L_σ est égale au nombre de facteurs absolument irréductibles de P. A présent nous allons ramener l'étude de l'espace L_σ à la résolution d'un système linéaire avec n inconnues. Pour cela nous allons introduire quelques notations.

NOTATIONS :
Nous notons par $\pi_{\mathbb{K}^n}$ la projection qui envoie $(l_1, \ldots, l_n; G; H) \in L_\sigma$ sur $(l_1, \ldots, l_n) \in \mathbb{K}^n$. De même nous notons avec un léger abus de notations π_G (respectivement π_H) la projection qui envoie $(l_1, \ldots, l_n; G; H) \in L_\sigma$ sur G (respectivement sur H).

La propriété suivante va nous permettre de ramener à $\pi_{\mathbb{K}^n}(L_\sigma)$ l'étude de L_σ.

Proposition 21. *Pour $\sigma \geq 2n$, $\pi_{\mathbb{K}^n} : L_\sigma \longrightarrow \pi_{\mathbb{K}^n}(L_\sigma)$ est un isomorphisme de $\mathbb{K}$ espaces vectoriels.*

Démonstration. La surjection est claire. Il nous reste donc à étudier le noyau. Soit $(0, \ldots, 0; G; H)$ un élément du noyau. G doit donc vérifier $G = 0 \mod (X, Y)^\sigma$. Or $G \in \mathbb{K}[X, Y]_{n-1}$ et $\sigma \geq 2n$. Il vient donc $G = 0$ dans $\mathbb{K}[X, Y]_{n-1}$.
Nous reprenons le même raisonnement avec H, et nous obtenons l'injection. □

La proposition suivante nous explique comment obtenir $\pi_{\mathbb{K}^n}(L_\sigma)$.

Proposition 22. *Soit D_σ le système suivant en les inconnues $l_1, \dots, l_n$.*

$$D_\sigma = \begin{cases} \displaystyle\sum_{i=1}^{n} l_i \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{i-1}, X^j, Y^k) = 0,\ k \leq n-1,\ n \leq j+k \leq \sigma - 1, \\ \displaystyle\sum_{i=1}^{n} l_i \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial X}, \varphi^{i-1}, X^j, Y^k) = 0,\ k \leq n-1,\ j \leq \sigma-2,\ n \leq j+k \leq \sigma - 1. \end{cases}$$

Pour $\sigma \geq n+1$ nous avons : $\pi_{\mathbb{K}^n}(L_\sigma) = \{(l_1, \dots, l_n) \mid D_\sigma\}$.

Démonstration. Montrons $\pi_{\mathbb{K}^n}(L_\sigma) \subset \{(l_1, \dots, l_n) \mid D_\sigma\}$.
Soit $(l_1, \dots, l_n) \in \pi_{\mathbb{K}^n}(L_\sigma)$. Nous avons donc un polynôme $G \in \mathbb{K}[X,Y]_{n-1}$ tel que :

$$(\star) \quad G(X,Y) = \sum_{i=1}^{n} l_i \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{i-1}) \mod (X,Y)^\sigma.$$

Comme $\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y} = \frac{P(X,Y)}{Y - \phi(X)}$ nous avons $\deg_Y(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}) = n-1$, et nous pouvons écrire :

$$G(X,Y) = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} g_{j,k} X^j Y^k = \sum_{i=1}^{n}\sum_{j=0}^{\infty}\sum_{k=0}^{n-1} l_i \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{i-1}, X^j, Y^k) X^j Y^k \mod (X,Y)^\sigma.$$

En identifiant les termes du membre de droite et du membre de gauche correspondant aux monômes n'appartenant ni à $\mathbb{K}[X,Y]_{n-1}$ ni à $(X,Y)^\sigma$ nous obtenons un vecteur $(l_1, \dots, l_n)$ vérifiant le premier ensemble d'équations de D_σ.
En procédant de même avec :

$$(\star\star) \quad H(X,Y) = \sum_{i=1}^{n} l_i \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial X}, \varphi^{i-1}) \mod (X,Y)^\sigma + (X)^{\sigma-1},$$

nous obtenons $(l_1, \dots, l_n)$ vérifiant le deuxième ensemble d'équations. D'où $\pi_{\mathbb{K}^n}(L_\sigma) \subset \{(l_1, \dots, l_n) \mid D_\sigma\}$.
A présent, montrons $\pi_{\mathbb{K}^n}(L_\sigma) \supset \{(l_1, \dots, l_n) \mid D_\sigma\}$.
Pour cela il suffit de définir G à l'aide de l'équation $(\star)$, et H à l'aide de l'équation $(\star\star)$. □

Grâce à cette proposition nous voyons que le calcul de $\pi_{\mathbb{K}^n}(L_\sigma)$ se ramène à la résolution d'un système linéaire en n inconnues. Ainsi lorsque $\sigma \geq 2n$, à partir d'une base $\{\nu_1, \dots, \nu_s\}$ de $\pi_{\mathbb{K}^n}(L_\sigma)$ nous pouvons obtenir une base $\{G_1, \dots, G_s\}$ de $\pi_G(L_\sigma)$. Pour cela il suffit de poser :

$$G_i = \sum_{j=1}^{n} \nu_{i,j} \operatorname{coef}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{j-1}).$$

Comme nous avons obtenu une base de $\pi_G(L_\sigma)$, nous avons donc réalisé la première étape de l'algorithme de S. Gao. Nous pouvons donc déterminer de façon probabiliste la factorisation absolue de P en suivant la méthode de S. Gao avec les $G_1, \ldots, G_s$. Nous verrons dans la section 4.2, comment à partir des G_i obtenir la factorisation absolue de P de manière déterministe.
Avant de passer à la factorisation déterministe, nous donnons un algorithme déterministe pour tester l'irréductibilité absolue de P.

Algorithme Test-Deterministe-Abs-Irred
ENTRÉE : $P(X,Y) \in \mathbb{K}[X,Y]$ avec $P(X,Y)$ irréductible dans $\mathbb{K}[X,Y]$ et les hypothèses (C) et (H) vérifiées.
SORTIE : "Absolument irréductible" ou "Absolument réductible".

1. Calculer $\mathfrak{F} \mod X^{2n}$.
2. Former le système D_σ.
3. Si $\dim_{\mathbb{K}}\{(l_1, \ldots, l_n) \mid D_\sigma\} = 1$ alors rendre "Absolument irréductible", sinon rendre "Absolument réductible".

Proposition 23. *L'algorithme* Test-Deterministe-Abs-Irred *est correct.*

Démonstration. C'est une conséquence immédiate du corollaire 11 et des propriétés 21 et 22. □

4.2 Comment rendre déterministe l'algorithme de S. Gao

4.2.1 Séparation

Nous avons vu dans la section précédente, qu'en relevant $\mathfrak{F}$ à la précision $2n$ nous pouvions définir des polynômes $G_i(X,Y) \in \mathbb{K}[X,Y]_{n-1}$ ainsi :

$$G_i = \sum_{j=1}^{n} \nu_{i,j} \operatorname{coef}(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial Y}, \varphi^{j-1}).$$

D'après le théorème de remontée (voir théorème 30 page 112), il existe pour chaque G_i un unique vecteur $\rho \in \overline{\mathbb{K}}^s$ tel que :

$$G_i(X,Y) = \sum_{j=1}^{s} \rho_{j,i} \hat{P}_j \frac{\partial P_j}{\partial Y}.$$

Par construction la matrice $(\rho_{i,j})_{(i,j)}$ de taille $s \times s$ est inversible car elle correspond à un changement de base. En particulier cela signifie que l'ensemble $V = \{(\rho_{j,1}, \ldots, \rho_{j,s}) \mid j = 1, \ldots, s\}$ est de cardinal s. Dans la section suivante

nous allons présenter une méthode déterministe pour obtenir le polynôme minimal d'un élément primitif pour le corps des coefficients de P_1. Nous verrons que cela se ramène à trouver une forme linéaire $c(t_1, \ldots, t_s) = \sum_{i=1}^{s} c_i t_i$ séparant les points $(\rho_{j,1}, \ldots, \rho_{j,s})$.
Afin d'alléger l'écriture dans ce qui suit, nous introduisons de nouvelles notations :

NOTATIONS :
On pose $g_i(Y) = G_i(0, Y)$, $p_i(Y) = P_i(0, Y)$ et $\hat{p}_i(Y) = \hat{P}_i(0, Y)$. Nous avons donc $g_i = \sum_{j=1}^{s} \rho_{j,i} \hat{p}_j p_j$. Pour chaque forme $\mathbb{K}$ linéaire c on note V_c l'ensemble $\left\{c(\rho_j) = \sum_{i=1}^{s} c_i \rho_{j,i} \mid j \in \{1, \ldots, s\}\right\}$ et s_c le cardinal de V_c. On remarque que $s_c = s$ est équivalent à c sépare les ρ_j.

Afin de démontrer l'existence de formes séparantes c, nous allons démontrer un lemme qui nous permet de retrouver l'approche probabiliste de S. Gao.

Lemme 35. *Avec les notations précédentes, nous avons le résultat suivant :*
Il existe un polynôme $S \in \overline{\mathbb{K}}[C_1, \ldots, C_s]$ de degré total $s(s-1)/2$ tel que : pour toute forme $c(t_1, \ldots, t_s) = \sum_{i=1}^{s} c_i t^i$, vérifiant $S(c_1, \ldots, c_s) \neq 0$ nous avons $s_c = s$.

Démonstration. Il suffit de considérer le polynôme :

$$S(C_1, \ldots, C_s) = \prod_{i<j} \Big(\sum_{k=1}^{s} (\rho_{i,k} - \rho_{j,k}) C_k \Big).$$

□

Corollaire 12. *En prenant les coefficients c_i au hasard de manière uniforme dans $S = \{0, 1, \ldots, n(n-1)\}$, nous obtenons l'estimation de la probabilité suivante :*

$$\mathcal{P}(s_c \neq s) \leq \frac{s(s-1)}{2n(n-1)}.$$

De plus il existe des formes séparantes à coefficients dans S.

Démonstration. L'hypothèse (C) étant vérifiée nous pouvons considérer le sous-ensemble S de $\mathbb{K}$ définit dans l'énoncé. L'estimation de la probabilité provient du lemme de Zippel/Schwartz (voir lemme 2 page 6).
Ainsi le nombre de formes $\mathbb{K}$ linéaires à coefficients dans S ne séparant pas les ρ_j est inférieur à $n(n-1)^{s-1} s(s-1)/2$. Cela signifie donc qu'il existe des formes séparant les ρ_j. □

4.2.2 Utilisation du résultant

Le théorème suivant est une version avec résultant du théorème 15 vu page 26. Nous introduisons les résultants car le calcul du résultant de deux polynômes de degré inférieur à n peut s'effectuer en utilisant $(24M(n) + O(n))\log(n)$ opérations dans $\mathbb{K}$ (voir [vzGG03] corollaire 11.16 page 323). Tandis que le calcul du polynôme caractéristique d'une matrice de taille $n \times n$ à coefficients dans $\mathbb{K}$ demande $O(n^\omega \log n)$ opérations dans $\mathbb{K}$ (voir [AL04] page 244-245). L'exposant ω est l'exposant de l'algèbre linéaire, c'est-à-dire le produit de 2 matrices $n \times n$ à coefficients dans $\mathbb{K}$ s'effectue en $O(n^\omega)$ opérations dans $\mathbb{K}$. Nous rappelons que $2 \leq \omega \leq 3$ et $M(n) \in O(n \log(n) \log\log(n))$.

Théorème 31. *Soit c une forme $\mathbb{K}$ linéaire, nous avons alors :*

$$Res_Y(p(Y), Zp'(Y) - c(g_1(Y), \ldots, g_s(Y)) = \delta q_c(Z)^{\frac{n}{s_c}},$$

où $\delta = Disc_{\mathbb{K}}(p)$, et $q_c(Z) = \prod_{\gamma \in V_c}(Z - \gamma) \in \mathbb{K}[Z]$. De plus, q_c est irréductible dans $\mathbb{K}[Z]$ et pour tout $\gamma \in V_c$ nous avons :

1. $\deg_Y(pgcd(p, \gamma p' - c(g_1, \ldots, g_s))) = \dfrac{n}{s_c}$,
2. $pgcd(P, \gamma \dfrac{\partial P}{\partial Y} - c(G_1, \ldots, G_s)) \in \mathbb{K}[\gamma][X, Y]$ *est irréductible dans* $\mathbb{K}[\gamma][X, Y]$, *et, de degré* $\dfrac{n}{s_c}$.

Avant de passer à la preuve de ce théorème, remarquons que l'existence de formes séparantes vue dans la section précédente, nous permet d'obtenir un algorithme probabiliste. En effet si c est une forme séparante alors $q_c(Z)$ est un polynôme irréductible de degré $s_c = s$ ayant pour racine γ, et, $pgcd(P, \gamma\, \partial P/\partial Y - c(G1, \ldots, G_s))$ est un polynôme irréductible de degré n/s divisant P. Le calcul du *pgcd* nous donne donc un facteur absolument irréductible de P, et q_c nous donne le polynôme minimal d'un élément primitif du corps engendré par les coefficients de ce facteur.

Démonstration. Tout d'abord nous allons montrer que les $c(\rho_j)$ sont conjugués sur $\mathbb{K}$.
Soit α un élément primitif sur $\mathbb{K}$ pour le corps des coefficients de P_1, et, $\sigma_1, \ldots, \sigma_s$ les $\mathbb{K}$-homomorphismes de $\mathbb{K}[\alpha]$ dans $\overline{\mathbb{K}}$. Nous posons alors :

$$B_k = \sum_{j=1}^{s} \sigma_j(\alpha^k \hat{P}_1 \frac{\partial P_1}{\partial Y}) = \sum_{j=1}^{s} \sigma_j(\alpha^k)\hat{P}_j \frac{\partial P_j}{\partial Y} \in \mathbb{K}[X, Y], \text{ pour } k = 1, \ldots, s.$$

L'ensemble $\{B_1, \ldots, B_s\}$ est donc une base de $\pi_G(L_\sigma)$ car la matrice de passage de $Vect(\hat{P}_1\partial P_1/\partial Y, \ldots, \hat{P}_s\partial P_s/\partial Y)$ à $Vect(B_1, \ldots, B_s)$ est la matrice de Vandermonde en $\sigma_1(\alpha), \ldots, \sigma_s(\alpha)$, qui est donc inversible. Nous pouvons donc écrire les g_i de la manière suivante :

$$\begin{aligned} g_i(Y) &= \sum_{k=1}^{s} \lambda_k^{(i)} B_k(0, Y) = \sum_{k=1}^{s} \lambda_k^{(i)} \sum_{j=1}^{s} \sigma_j(\alpha^k)\hat{p}_j p'_j \\ &= \sum_{j=1}^{s} \left(\sum_{k=1}^{s} \lambda_k^{(i)} \sigma_j(\alpha^k) \right) \hat{p}_j p'_j. \end{aligned}$$

Par définition des $\rho_{j,i}$ (voir page 116), il vient :

$$\rho_j = \Big(\sum_{k=1}^{s} \lambda_k^{(1)} \sigma_j(\alpha^k), \ldots, \sum_{k=1}^{s} \lambda_k^{(s)} \sigma_j(\alpha^k)\Big).$$

Ainsi les éléments $c(\rho_j) = \sum_{i=1}^{s} c_i \rho_{j,i}$ sont conjugués sur $\mathbb{K}$, c'est à dire $\sigma_j(c(\rho_1)) = c(\rho_j)$. Par conséquent, $q_c(Z) = \prod_{\gamma \in V_c}(Z - \gamma)$ est le polynôme minimal sur $\mathbb{K}$ de $c(\rho_1)$, il est donc irréductible sur $\mathbb{K}$.

A présent calculons le résultant. Pour cela on note y_i les n racines distinctes de $p(Y)$, et $g(Y) = c(g_1(Y), \ldots, g_s(Y))$, et $R(Y) = Res_Y(p(Y), Zp'(Y) - g(Y))$. Nous avons alors :

$$R(Y) = \prod_{i=1}^{n} p'(y_i) \times \prod_{i=1}^{n} \Big(Z - \frac{g(y_i)}{p'(y_i)}\Big) = Disc_{\mathbb{K}}(p) \times \prod_{i=1}^{n} \Big(Z - \frac{g(y_i)}{p'(y_i)}\Big).$$

Nous devons donc simplifier le second membre de l'égalité ci-dessus. On note alors $y_{i,k}$ les racines de $p_k(Y)$. Nous avons alors :

$$\frac{g(y_{i,k})}{p'(y_{i,k})} = \frac{\sum_{j=1}^{s} c_j \sum_{u=1}^{s} \rho_{u,j} \hat{p}_u(y_{i,k}) p'_u(y_{i,k})}{p'(y_{i,k})} = \frac{\sum_{j=1}^{s} c_j \rho_{k,j} \hat{p}_k(y_{i,k}) p'_k(y_{i,k})}{p'(y_{i,k})},$$

car $\hat{p}_u(y_{i,k}) = 0$ si $k \neq u$. De plus comme $p'(y_{i,k}) = \hat{p}_k(y_{i,k}) p'_k(y_{i,k})$, nous avons :

$$\frac{g(y_{i,k})}{p'(y_{i,k})} = \sum_{j=1}^{s} c_j \rho_{k,j} = c(\rho_k).$$

Ainsi :

$$R(Z) = \delta \times \prod_{i=1}^{s} (Z - c(\rho_i))^{\deg(p_i)}.$$

Or, nous avons $\deg_Y(P_i(X,Y)) = \dfrac{n}{s}$ d'après le lemme fondamental (voir le lemme 1 page 3). De plus nous avons supposé P unitaire, il vient donc $\deg(p_i) = \dfrac{n}{s}$. D'où :

$$\begin{aligned} R(Z) &= \delta \times \prod_{i=1}^{s} (Z - c(\rho_i))^{n/s} = \delta \times (P_{char}(c(\rho_1))(Z))^{n/s} \\ &= \delta \times (q_c(Z))^{[\mathbb{K}[\alpha]:\mathbb{K}[c(\rho_1)]]\cdot\frac{n}{s}}. \end{aligned}$$

Comme $[\mathbb{K}[\alpha] : \mathbb{K}[c(\rho_1)]] = \dfrac{s}{s_c}$, nous obtenons :

$$R(Z) = \delta \times q_c(Z)^{n/s_c}.$$

A présent en reprenant la preuve du corollaire 3 page 25, avec

$$c(G_1,\dots,G_s)=\sum_{\gamma\in V_c}\gamma\Big(\sum_{c(\rho_i)=\gamma}\hat{P}_i\frac{\partial P_i}{\partial Y}\Big),$$

nous voyons que

$$\prod_{c(\rho_i)=\gamma}P_i=pgcd(P,\gamma\frac{\partial P}{\partial Y}-c(G1,\dots,G_s))$$

est un polynôme irréductible dans $\mathbb{K}[\gamma][X,Y]$, de degré n/s_c.
De même nous obtenons $pgcd(p,\gamma p'-c(g_1,\dots,g_s))=\prod_{c(\rho_i)=\gamma}p_i$, ce qui achève la démonstration. □

Remarques :
En caractéristique zéro, nous pouvons voir ce théorème comme un cas particulier de l'algorithme de Rothstein/Trager (voir [vzGG03] , et, [Bro97]). En effet nous sommes dans la situation suivante :

$$\frac{c(g_1,\dots,g_s)}{p}=\sum_{\gamma\in V_c}\gamma\sum_{c(\rho_j)=\gamma}\frac{p'_j}{p_j}.$$

Cela donne :

$$\int\frac{c(g_1,\dots,g_s)}{p}=\sum_{\gamma\in V_c}\gamma\log\Big(\prod_{c(\rho_j)=\gamma}p_j\Big).$$

Le théorème de Trager et Rothstein nous dit alors :

1. les racines de $R(Z)=Res_Y(p(Y),Zp'(Y)-c(g_1(Y),\dots,g_s(Y))$ sont les éléments γ de V_c,
2. $\prod_{c(\rho_i)=\gamma}p_i=pgcd(p,\gamma p'-c(g_1,\dots,g_s)))$.

Ensuite en utilisant le théorème de Lazard/Rioboo (voir [vzGG03], et, [LR90]) nous pouvons déduire que l'exposant de $q_c(Z)$ est n/s_c.

4.2.3 Un algorithme déterministe pour la factorisation absolue

L'idée de notre algorithme est la suivante : Nous allons calculer une base $\nu_1,\dots,\nu_s$ de $\pi_{\mathbb{K}^n}(L_{2n})$. Nous pouvons alors en déduire les polynômes $g_1,\dots,g_s$ définis page 117. Ensuite nous appliquons de manière itérative le théorème 31. C'est à dire : si la forme obtenue n'est pas séparante, alors nous obtenons une extension $\mathbb{K}[\gamma]$ de degré s_c sur $\mathbb{K}$, et un facteur $\mathfrak{P}$ de P irréductible dans $\mathbb{K}[\gamma]$. Dans ce cas nous appliquons de nouveau à $\mathfrak{P}$ le théorème 31.
Avant de passer à la description de notre algorithme, nous allons donner deux lemmes. Le premier nous permettra d'utiliser le théorème 31 de manière itérative. Le second facilitera la preuve de notre algorithme.

Lemme 36. *Soient* $p = \prod_{i=1}^{s} p_i$, *et* $k < s$. *Nous notons alors :*
- $\zeta = \prod_{i=1}^{k} p_i$,
- $\chi = \prod_{i=k+1}^{s} p_i$,
- $g = \sum_{j=1}^{s} c(\rho_j)\hat{p}_j p'_j$,
- $\hat{p}_j^{(\zeta)} = \prod_{i=1, i\neq j}^{k} p_i$,
- $g^{(\zeta)} = \sum_{j=1}^{k} c(\rho_j)\hat{p}_j^{(\zeta)} p'_j$.

$Res_Y(\zeta, Tp' - g)$ *est alors à une constante multiplicative près égale à* $Res_Y(\zeta, T\zeta' - g^{(\zeta)})$. *Ce que nous noterons*

$$Res_Y(\zeta, Tp' - g) \sim Res_Y(\zeta, T\zeta' - g^{(\zeta)}).$$

Démonstration. Calculons le résultant :

$$\begin{aligned} Res_Y(\zeta, Tp' - g) &= Res_Y(\zeta, T(\zeta'\chi + \zeta\chi') - \sum_{j=1}^{k} c(\rho_j)\hat{p}_j^{(\zeta)} p'_j \chi - \zeta h) \text{ où } h \in \overline{\mathbb{K}}[Y] \\ &= Res_Y(\zeta, T(\zeta'\chi) - \sum_{j=1}^{k} c(\rho_j)\hat{p}_j^{(\zeta)} p'_j \chi) \\ &= Res_Y(\zeta, \chi) Res_Y(\zeta, T\zeta' - \sum_{j=1}^{k} c(\rho_j)\hat{p}_j^{(\zeta)} p'_j) \\ &\sim Res_Y(\zeta, T\zeta' - g^{(\zeta)}). \end{aligned}$$

□

Lemme 37. *Soient* $\mathbb{E} := \mathbb{K}[\alpha] = \dfrac{\mathbb{K}[T]}{(q(T))}$ *une extension de* $\mathbb{K}$, $r(Y)$ *un polynôme irréductible dans* $\mathbb{K}[\alpha][Y]$ *et* $\mathrm{r}(T, Y) \in \mathbb{K}[T, Y]$ *tel que* $\mathrm{r}(\alpha, Y) = r(Y)$. *On note* $\alpha_1, \ldots, \alpha_k$ *les racines du polynôme irréductible* q, *avec* $\alpha_1 = \alpha$. *On a :*

$$N(Y) = \prod_{i=1}^{k} \mathrm{r}(\alpha_i, Y) = Res_T(q(T), \mathrm{r}(T, Y)) \in \mathbb{K}[T].$$

De plus $N(Y)$ *est une puissance d'un polynôme irréductible dans* $\mathbb{K}[T]$.

Démonstration. L'égalité $\prod_{i=1}^{k} \mathrm{r}(\alpha_i, Y) = Res_T(q(T), \mathrm{r}(T, Y))$ provient de la première propriété du résultant (voir [Esc97] page 34). Notre deuxième affirmation est le théorème 2.1 de l'article de Trager (voir [Tra76]) sur la norme d'un polynôme. □

A présent nous pouvons décrire notre algorithme :

Algorithme Facto-Abs-Déterministe

Entrée : $P \in \mathbb{K}[X,Y]$, tel que les hypothèses (C) et (H) soient vérifiées.
Sortie : $P_1 \in \mathbb{K}[\alpha][X,Y]$ un facteur absolument irréductible de P, et $q \in \mathbb{K}[T]$ le polynôme minimal de α.

1. Résoudre D_{2n}, et obtenir une base $\nu_1, \ldots, \nu_s$ de $\pi_{\mathbb{K}^n}(L_{2n})$.
2. En déduire $g_1, \ldots, g_s \in \mathbb{K}[Y]$.
3. $h := 0$, $k := 1$, $q := T$, $\mathbb{E} := \mathbb{K}[\alpha] = \dfrac{\mathbb{K}[T]}{(q(T))}$, $\zeta(Y) := p(Y) \in \mathbb{E}[Y]$.
4. Tant que $k < s$ faire
   (a) Trouver $i_0 \in \{1, \ldots, s\}$ tel que $g_{i_0} \mod \zeta$ ne soit pas proportionnel à ζ'.
   (b) $R(Z) := Res_Y(\zeta, Zp' - g_{i_0}) \in \mathbb{E}[Z]$.
   (c) Calculer la décomposition sans carré de R pour obtenir R proportionnel à $r^{n/d}$.
   (d) $k := k \times d$.
   (e) Calcul du polynôme minimal d'un élément de $\dfrac{\mathbb{E}[Y]}{(r(Y))}$:
   Pour $a := 0$ à $s(s-1)$ faire
      i. $f(T) := Res_W(q(W), \mathrm{r}(W, T + aW)) \in \mathbb{K}[T]$, où $\mathrm{r}(W,T) \in \mathbb{K}[W,T]$ est tel que $\mathrm{r}(\alpha, T) = r(T)$ dans $\mathbb{E}[T]$.
      ii. Si f est sans carré alors stopper la boucle "Pour".
   (f) Trouver h tel que $Res_Y(p, Tp' - h)$ soit proportionnel à $f^{n/d}$:
   $h := -ah + g_{i_0}$.
   (g) $q := f$, $\mathbb{E} := \mathbb{K}[\alpha] = \dfrac{\mathbb{K}[T]}{(q(T))}$.
   (h) $\zeta := pgcd(p, \alpha p' - h) \in \mathbb{E}[Y]$.
5. Calculer à l'aide d'une remontée de Hensel jusqu'à l'ordre $n/s+1$ le polynôme $P_1(X,Y) \in \mathbb{K}[\alpha][X,Y]$ divisant P et vérifiant $P_1(0,Y) = \zeta(Y)$.

Proposition 24. *L'algorithme* Facto-Abs-Déterministe *termine et est correct.*

Démonstration. ► L'algorithme termine.

Pour voir cela nous allons montrer que la boucle "Tant que" s'arrête. Il suffit de prouver qu'à chaque passage dans la boucle nous avons $d > 1$. D'aprés le théorème 31, r est de degré d, donc il suffit de montrer qu'à chaque passage $\deg(r) > 1$.
Comme $Res_Y(\zeta, Zp' - g_{i_0}) \sim Res_Y(\zeta, Z\zeta' - g_{i_0}^{(\zeta)})$, cela sera le cas dès que $g_{i_0}^{(\zeta)}$ ne sera pas proportionnel à ζ', (toujours d'aprés le théorème 31).
Or, lors du premier passage nous avons $\zeta = p$, donc avec un abus de notations il vient $g_i^{(\zeta)} = g_i$. De plus $p' = \sum_{j=1}^s \hat{p}_j p'_j$ et $g_i = \sum_{j=1}^s \rho_{j,i}\hat{p}_j p'j$. La matrice $(\rho_{j,i})_{i,j}$ étant inversible il existe un indice i_0 tel que g_{i_0} ne soit pas proportionnel à p'.
Lors des passage suivants, nous avons : $\zeta = \prod_{j=1}^k p_j$, avec $k < s$, et

$\zeta' = \sum_{j=1}^{k} \hat{p}_j^{(\zeta)} p_j'$. Nous rappelons que : $g_i^{(\zeta)} = \sum_{j=1}^{k} \rho_{j,i} \hat{p}_j^{(\zeta)} p_j'$ et que $V = \{\rho_j = (\rho_{j,1}, \ldots, \rho_{j,s}) \mid j = 1, \ldots, s\}$ est de cardinal s (voir page 116). Nous ne pouvons donc pas avoir tous les $g_i^{(\zeta)}$ proportionnel à ζ'. En effet, cela impliquerait : $\rho_1 = \ldots = \rho_k$ et nierait : $|V| = s$. Ainsi il existe un indice i_0 tel que $g_{i_0}^{(\zeta)}$ ne soit pas proportionnel à ζ'. Nous en déduisons alors que $g_{i_0} \mod \zeta$ n'est pas proportionnel à ζ'.

Finallement, nous venons de montrer que nous pouvons trouver un indice i_0 tel que $g_{i_0} \mod \zeta$ ne soit pas proportionnel à ζ', et que pour cet indice i_0 nous avons $g_{i_0}^{(\zeta)}$ non proportionnel à ζ'. Cela nous permet donc de conclure qu'à chaque passage dans la boucle nous avons $d > 1$, et donc que l'algorithme termine.

► L'algorithme est correct.

Nous devons avant tout remarquer que nous appliquons le théorème 31 de manière itérative.

En 4e nous cherchons un élément primitif de $\dfrac{\mathbb{E}[Y]}{(r(Y))}$ du type : $a\alpha+\beta$ où α est une racine de q et β une racine de r. On désigne par α_i les $\deg(q)$ conjugués de α et par β_j les $\deg(r)$ conjugués de β. Si nous prenons un élément a, tel que les $a\alpha_i + \beta_j$ sont différents deux à deux, alors nous aurons obtenu l'élément primitif recherché. Ainsi a ne doit pas être égal à des éléments du type : $\dfrac{\beta_k - \beta_l}{\alpha_i - \alpha_j}$ nous devons donc éviter $\deg(q)(\deg(q)-1)/2 + \deg(r)(\deg(r)-1)/2$ valeurs. Or $\deg(q) \leq s$ et $\deg(r) \leq s$. Ainsi nous devons éviter au plus $s(s-1)$ valeurs. Nous obtiendrons donc une valeur vérifiant la propriété souhaitée en prenant a dans $\{0, 1, \ldots, s(s-1)\}$.

D'après le lemme 37 en 4e nous calculons bien le polynôme minimal d'un élément primitif.

Il nous reste à voir qu'en 4f nous construisons bien un polynôme h tel que $Res_Y(\zeta, T\zeta' - h)$ soit proportionnel à $f^{n/d}$. Nous avons à la fin de l'étape 4e :

α est une racine de q, et nous notons $\sigma_1, \ldots, \sigma_{\deg(q)}$ les $\deg(q)$ $\mathbb{K}$-homorphisme de $\mathbb{K}[\alpha]$ dans $\overline{\mathbb{K}}$. Nous avons aussi $\zeta = pgcd(p, \alpha p' - h) \in \mathbb{K}[\alpha][Y]$. Donc $\prod_{i=1}^{\deg(q)} \sigma_i(\zeta) = p$ car ζ divise p et $\deg(\zeta) = n/\deg(q)$ d'après le théorème 31. Calculons à présent le résultant :

$$Res_Y(p, Tp' + ah - g_{i_0}) = \prod_{i=1}^{\deg(q)} \sigma_i(Res_Y(\zeta, Tp' + ah - g_{i_0})).$$

Comme $\zeta = pgcd(p, \alpha p' - h)$ nous avons $ah = a\alpha p' \mod \zeta$. D'où :

$$\begin{aligned} Res_Y(p, Tp' + ah - g_{i_0}) &= \prod_{i=1}^{\deg(q)} \sigma_i(Res_Y(\zeta, (T + a\alpha)p' - g_{i_0}) \\ &= \prod_{i=1}^{\deg(q)} \sigma_i(R(T + a\alpha)) \end{aligned}$$

$$\begin{aligned} Res_Y(p, Tp' + ah - g_{i_0}) &= \prod_{i=1}^{\deg(q)} \sigma_i(R(T + a\alpha)) \\ &\sim \prod_{i=1}^{\deg(q)} \sigma_i(r(T + a\alpha))^{n/d} \\ &\sim (Res_W(q(W), \mathrm{r}(T + aW))^{n/d} \\ &\sim (f)^{n/d}. \end{aligned}$$

Ainsi l'élément h construit à l'étape 4f vérifie bien la propriété annoncée. □

4.3 Conclusion

Ce chapitre correspond à un travail en collaboration avec G. Lecerf. Certains résultats présentés ici peuvent être généralisés, ou précisés.

En effet, nous venons de voir un algorithme de factorisation absolue s'appliquant à des polynômes irréductibles de $\mathbb{K}[X, Y]$. Cette hypothèse a été choisie afin de faciliter l'exposé de cette méthode. Nous pourrions généraliser cet algorithme aux polynômes sans facteurs carrés. En effet le théorème 30 reste inchangé sous cette hypothèse plus large. De même l'hypothèse "$\mathbb{K}$ est un corps parfait" est présente seulement pour nous permettre d'utiliser le lemme fondamental dans la preuve du théorème 31. Or nous pouvons grâce au lemme 31 obtenir une version plus générale du lemme fondamental. Cette version ne suppose pas $\mathbb{K}$ parfait.
De plus une étude détaillée de l'algorithme *Facto-Abs-Déterministe* nous montre que celui-ci nécessite $\tilde{O}(n^4)$ opérations arithmétiques dans $\mathbb{K}$, (voir [vzGG03] pour la définition de $\tilde{O}$). Nous pouvons aussi étudier une version probabiliste de cet algorithme, dans ce cas la complexité est de l'ordre de $\tilde{O}(n^{(\omega+3)/2})$. Nous rappelons que ω est un nombre réel tel que le produit de 2 matrices $n \times n$ à coefficients dans $\mathbb{K}$ puisse s'effectuer en $O(n^\omega)$ opérations dans $\mathbb{K}$, et, nous avons $2 \leq \omega \leq 3$ (voir [vzGG03]).
Nous voyons donc que cet algorithme apporte un gain certain, étant donné que la complexité de l'algorithme probabiliste de S. Gao est de l'ordre de $\tilde{O}(n^5)$.

Ces diverses précisions sont en cours d'écriture pour l'article [CL].

Chapitre 5

Un critère d'irréductibilité absolue

Ce chapitre correspond à un travail en cours [Chè04a].

D'après le théorème de Noether (voir page 5), génériquement un polynôme de degré total d dans $\mathbb{K}[X,Y]$ est absolument irréductible. D'autre part, dans de nombreux algorithmes de factorisation le pire cas est celui où le polynôme donné en entrée est irréductible. Ainsi il est important de savoir tester rapidement si un polynôme est absolument irréductible ou pas. En effet un test rapide d'irréductibilité absolue permet d'éviter un algorithme de factorisation pouvant être coûteux. Dans ce chapitre nous allons démontrer le test d'irréductibilité suivant. Ce test est très proche de celui donné par S. Gao dans [Gao01] (voir section 1.6.3 page 38) mais la démonstration est très différente.

Lemme 38 (Test d'irréductibilité absolue). *Soit $P(X,Y) \in \mathbb{K}[X,Y]$ un polynôme irréductible dans $\mathbb{K}[X,Y]$, où $\mathbb{K}$ est un corps parfait. Soient (i_1, j_1), …, (i_l, j_l) les sommets de $Newt(P)$, le polytope de Newton de P.*
Si $pgcd(i_1, j_1, \ldots, i_l, j_l) = 1$ alors P est absolument irréductible.

Ce lemme montre que lorsque $pgcd(i_1, j_1, \ldots, i_l, j_l) = 1$, tester l'irréductibilité de P dans $\overline{\mathbb{K}}[X,Y]$ revient à tester l'irréductibilité de P dans $\mathbb{K}[X,Y]$.

Dans ce chapitre nous allons tout d'abord démontrer ce lemme. Ensuite nous donnerons diverses manières d'utiliser ce lemme. Nous verrons alors qu'une utilisation directe de ce lemme donne un test efficace pour l'irréductibilité absolue des polynômes creux de $\mathbb{Q}[X,Y]$. Mais nous verrons aussi qu'à l'aide de ce lemme nous pouvons fabriquer des algorithmes modulaires généralisant celui de J.-F. Ragot (voir la section 1.6.2).

Afin de simplifier les notations nous avons pris $P(X,Y) \in \mathbb{K}[X,Y]$, mais ce lemme et les algorithmes qui en découlent se généralisent aisément à un polynôme $P(X_1, X_2, \ldots, X_n) \in \mathbb{K}[X_1, \ldots, X_n]$.

5.1 Polytope de Newton et ordre monômial

Nous avons vu que le polytope de Newton d'un polynôme $\sum_{i,j} c_{i,j} X^i Y^j$ est l'enveloppe convexe dans $\mathbb{R}^2$ des points (i,j) correspondant à un coefficient $c_{i,j}$ non nul. Ici nous allons montrer que chaque sommet du polytope de Newton correspond à un "terme de tête" pour un ordre monômial donné.

Définition 33. *Un ordre monômial dans $\mathbb{K}[X,Y]$ est une relation $\prec$ dans $\mathbb{N}^2$ telle que :*

1. $\prec$ est un ordre total

2. Pour tout α, β, $\gamma \in \mathbb{N}^2$: $\alpha \prec \beta \Longrightarrow \alpha + \gamma \prec \beta + \gamma$.

3. $\prec$ est un bon ordre.

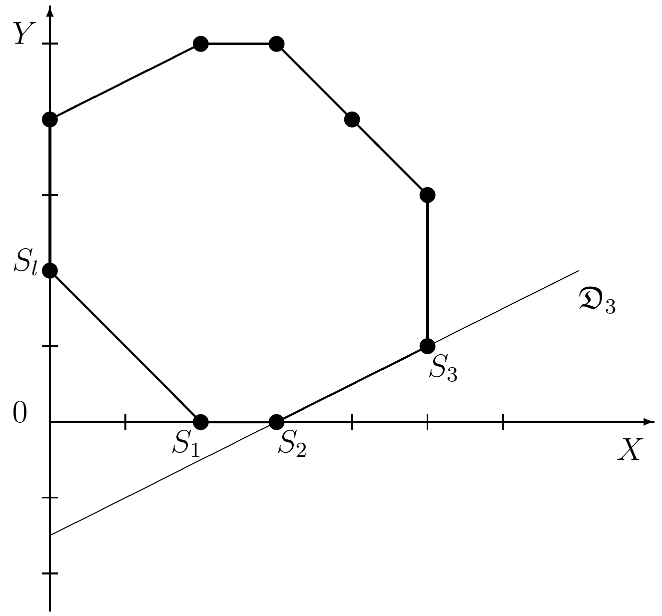
Définition 34. *Le multidegré de $P(X,Y) = \sum_{i,j} c_{i,j} X^i Y^j \in \mathbb{K}[X,Y]$ est :*
$mdeg_\prec(P) = \max_\prec\{\alpha \in \mathbb{N}^2 \,|\, c_\alpha \neq 0\}$ où $\max_\prec$ est le maximum pour l'ordre $\prec$.

Nous avons alors le lemme suivant :

Lemme 39. *Soit $\prec$ un ordre monômial sur $\mathbb{K}[X,Y]$ et $f(X,Y)$, $g(X,Y) \in \mathbb{K}[X,Y]$. On a alors : $mdeg_\prec(fg) = mdeg_\prec(f) + mdeg_\prec(g)$.*

La preuve du test d'irréductibilité absolue est basée sur le lemme suivant :

Lemme 40. *Si (i,j) est un sommet de $Newt(P)$ alors il existe un ordre monômial $\prec$ tel que : $mdeg_\prec(P) = (i,j)$.*



Démonstration. Tout d'abord nous énumérons les sommets $S_k = (i_k, j_k)$ de $Newt(P)$ en tournant dans le sens trigonométrique et en posant $S_1 = (i_1, j_1) = \text{mdeg}_{grevlex}(P)$, où *grevlex* est l'ordre gradué lexicographique inverse (voir figure ci-dessus). A présent, soit S_k un sommet de $Newt(P)$ avec $k > 1$, nous allons construire un ordre monômial $\prec$ tel que : $S_k = (i_k, j_k) = \text{mdeg}_{\prec}(P)$.
Nous considérons la droite $\mathfrak{D}_k$ passant par S_{k-1} et S_k d'équation : $a_k x + b_k y = c_k$.

Comme $Newt(P)$ est convexe alors, pour tous les points (x, y) de $Newt(P)$ n'appartenant pas à $\mathfrak{D}_k$, nous avons soit $a_k x + b_k y > c_k$, soit $a_k x + b_k y < c_k$. Si $a_k x + b_k y > c_k$ alors nous définissons $\prec$ comme suit :
Soient (α, β), $(\gamma, \delta) \in \mathbb{N}^2$. Si $a_k\alpha + b_k\beta < a_k\gamma + b_k\delta$ alors $(\gamma, \delta) \prec (\alpha, \beta)$.
(Lorsque $a_k x + b_k y < c_k$ on pose : Si $a_k\alpha + b_k\beta < a_k\gamma + b_k\delta$ alors $(\alpha, \beta) \prec (\gamma, \delta)$).
Si $a_k\alpha + b_k\beta = a_k\gamma + b_k\delta$ alors (α, β) et (γ, δ) sont sur une parallèle à $\mathfrak{D}_k$. Dans ce cas nous allons ordonner ces points à l'aide du vecteur $\overrightarrow{S_{k-1}S_k}$. On pose alors :
Si $< \overrightarrow{S_{k-1}S_k} ; (\gamma - \alpha, \delta - \beta) > \; > 0$ où $<;>$ désigne le produit scalaire usuel, alors $(\alpha, \beta) \prec (\gamma, \delta)$ sinon $(\gamma, \delta) \prec (\alpha, \beta)$.
A présent il est aisé de vérifier que l'ordre $\prec$ ainsi défini est un ordre monômial et que $\text{mdeg}_{\prec}(P) = (i_k, j_k)$. $\square$

5.2 Un critère d'irréductibilité absolue

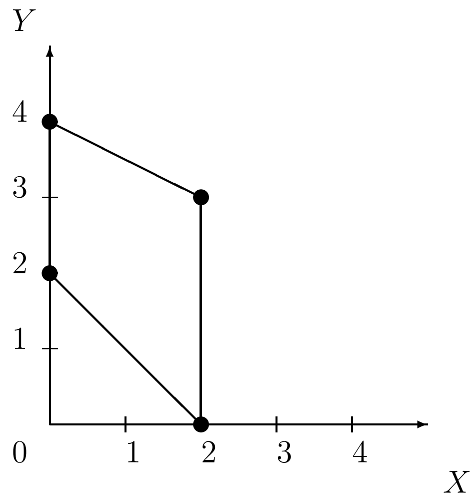
Dans cette partie nous allons donner la démonstration du test d'irréductibilité absolue et faire quelques remarques à propos de son utilisation.

Démonstration. Soit $P(X, Y) = P_1(X, Y) \cdots P_s(X, Y)$ une factorisation absolue de P, et soit (i_k, j_k) un sommet de $Newt(P)$. D'après le lemme 40 nous avons alors un ordre monômial $\prec$ tel que $\text{mdeg}_{\prec}(P) = (i_k, j_k)$. De plus le lemme 39 donne :

$$(i_k, j_k) = \text{mdeg}_{\prec}(P) = \sum_{i=1}^{s} \text{mdeg}_{\prec}(P_i).$$

D'autre part le lemme fondamental (lemme 1 page 3) donne $\text{mdeg}_{\prec}(P_1) = \ldots = \text{mdeg}_{\prec}(P_s)$ car les P_i sont conjugués. Nous posons alors $\text{mdeg}_{\prec}(P_i) = (\alpha, \beta)$. Il vient : $(i_k, j_k) = (s\alpha, s\beta)$ donc s divise i_k et j_k. Ainsi lorsque $pgcd(i_1, j_1, \ldots, i_l, j_l) = 1$ nous avons $s = 1$, ce qui signifie : P est absolument irréductible. $\square$

Exemple : Ce test d'irréductibilité s'applique au polynôme $P(X, Y) = Y^4 + X^2Y^3 + Y^2 + X^2 \in \mathbb{Q}[X, Y]$. En effet P est irréductible dans $\mathbb{Q}$ et le polytope de Newton associé à ce polynôme est :



Nous remarquons que sur cet exemple les critères de Gao ne s'appliquent pas. D'autre part nous ne pouvons pas non plus utiliser de manière immédiate le critère de Ragot car $(0,0)$ n'est pas un point simple.

Dans notre test l'hypothèse : $P(X,Y)$ irréductible, est nécessaire. En effet le polynôme $P(X,Y) = XY + X + Y + 1 = (X+1)(Y+1)$ a pour polytope de Newton le carré unité mais ce polynôme est réductible.

5.3 Diverses utilisations du critère

Dans ce qui suit nous allons présenter différentes utilisations de notre critère pour l'étude de l'irréductibilité absolue d'un polynôme $P(X,Y) \in \mathbb{Z}[X,Y]$.

5.3.1 L'utilisation directe

Nous avons vu que si le pgcd des coordonnées des sommets est égal à 1 alors tester l'irréductibilité absolue revient à tester l'irréductibilité dans $\mathbb{Q}[X,Y]$. Une question se pose alors : sommes nous souvent dans cette situation ? Il est clair que si nous prenons un polynôme dense alors son polytope de Newton sera le triangle de sommets $(0,0)$, $(n,0)$, $(0,n)$, et le test ne pourra pas s'appliquer.

Nous avons fabriqué de manière aléatoire 100 polynômes de degré 50 avec des coefficients dans $[-10^{12}, 10^{12}]$. Nous avons alors regardé combien d'entre eux vérifiaient le critère : aucun. (Cette expérience a été menée aussi sur des polynômes d'autres degrés et les résultats sont identiques).

Nous avons alors répété cette expérience pour des polynômes creux. Le tableau suivant résume les résultats obtenus. Les tests ont été effectués sur une machine du projet GALAAD de l'INRIA (Intel Pentium 4, 3.4 Ghz, 1 Go) en utilisant MagmaV2.11-8.
Nous avons fabriqué Nb polynômes aléatoires dans $\mathbb{Z}[X, Y]$ de degré total n de la manière suivante : Pour chaque monôme X^iY^j nous effectuons deux tirages, le premier dans $[0, Prop]$ et le deuxième dans $[-N, N]$. Si le nombre obtenu lors du premier tirage est 0, alors nous effectuons un deuxième tirage dans $[-N, N]$ et le nombre obtenu donne le coefficient de X^iY^j. Sinon le coefficient est 0. Ainsi lorsque $Prop = 1$ nous avons autant de coefficients nuls que non nuls, et si $Prop = 2$ nous avons deux fois plus de coefficients non nuls. $Succes$ désigne le nombre de fois où notre critère a permis d'affirmer que le polynôme était absolument irréductible. T_{moy} désigne le temps moyen en secondes nécessaire pour effectuer le test sur un polynôme.

n	N	$Prop$	Nb	$Succes$	T_{moy}
10	10^3	1	1000	852	0.0008
10	10^6	1	1000	848	0.0008
10	10^{12}	1	1000	834	0.0009
10	10^{12}	2	1000	916	0.0009
10	10^{12}	7	1000	576	0.0007
50	10^{12}	1	1000	819	0.0134
50	10^{12}	2	1000	943	0.0122
50	10^{12}	7	1000	985	0.0102
100	10^{12}	1	1000	832	0.0787
200	10^{12}	1	1000	849	0.6023
200	10^{12}	2	1000	948	0.4432

Nous voyons donc que ce test est bien adapté pour les polynômes creux.

5.3.2 L'utilisation modulaire directe

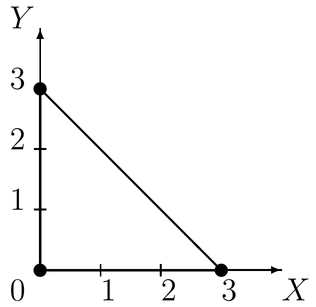
Ici nous allons présenter une utilisation modulaire de notre critère. Nous allons utiliser le théorème 6 page 8 :
Si $\deg(P \mod p) = \deg(P)$ et $P \mod p$ est absolument irréductible sur $\mathbb{F}_p$, alors P est absolument irréductible sur $\mathbb{Q}$.

Ici l'intérêt de regarder P modulo p vient du fait que l'on va pouvoir faire "disparaître" certains sommets, et en faire apparaître de nouveaux.
Soient $a_1, \ldots, a_k$ les coefficients de P correspondant aux sommets de $Newt(P)$. Soient $L := [p_1, \ldots, p_l]$ la liste des nombres premiers divisant au moins un des a_i. Nous remarquons alors que :

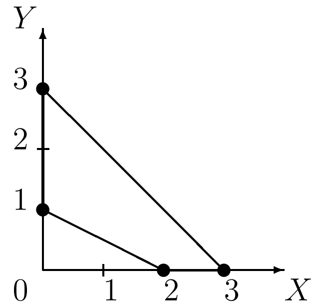
$$\forall p_i \in L, Newt(P) \neq Newt(P \mod p_i).$$

Donc si $Newt(P)$ est le triangle de sommets $(0,0)$, $(0,n)$, $(n,0)$, nous ne pouvons pas appliquer notre critère. Mais $Newt(P \mod p_i)$ est différent nous pouvons donc espérer appliquer notre critère.

EXEMPLE : $P(X,Y) = Y^3 + X^3 + 5X^2 + 3Y + 2$.



Polytope de Newton de P



Polytope de Newton de $P \mod 2$

Définition 35. *On dit que la condition (Polytope-Newton) est vérifiée lorsque le pgcd des coordonnées des sommets de $Newt(P)$ est égal à 1.*

On peut tester l'irréductibilité absolue de la manière suivante :

Algorithme Polytope-Newton-modulaire
ENTRÉE : $P(X,Y) \in \mathbb{Z}[X,Y]$, irréductible dans $\mathbb{Q}[X,Y]$.
SORTIE : "P est absolument irréductible" ou "Je ne sais pas".

1. Calculer $Newt(P)$.
2. Établir la liste L des facteurs premiers divisant un coefficient correspondant à un sommet de $Newt(P)$.
3. test :=faux ; $i := 1$;
4. Tant que (test=faux) et ($i \leq |L|$) faire
   $p := L[i]$;
   Si $\deg(P \mod p) = \deg(P)$ alors
      Calculer $Newt(P \mod p)$.
      Si $P \mod p$ vérifie la condition (Polytope-Newton) alors
         Si $P \mod p$ est irréductible dans $\mathbb{F}_p[X,Y]$ alors test :=vrai ; Fin Si ;
      Fin Si ;
   Fin Si ;
   $i := i + 1$;
   Fin Tant que ;
5. Si (test = vrai) alors rendre "P est absolument irréductible" ; sinon rendre "Je ne sais pas" ; Fin Si ;

Nous avons testé cet algorithme dans les mêmes conditions que précédemment, et les résultats obtenus se trouvent dans le tableau suivant :
Nous avons fabriqué Nb polynômes dans $\mathbb{Z}[X, Y]$ de degré total n en prenant les coefficients de manière aléatoire dans $[-N, N]$. Pour chaque polynôme nous avons appliqué l'algorithme Polytope-Newton-modulaire. *Succes* désigne le nombre de fois où l'algorithme a rendu "P est absolument irréductible". T_{moy} désigne le temps moyen en secondes mis par l'algorithme *Polytope-Newton-modulaire.* T_{max} (respectivement T_{min}) désigne la plus longue (respectivement la plus courte) durée en secondes obtenue sur les Nb essais de l'algorithme.

n	N	Nb	$Succes$	T_{moy}	T_{max}	T_{min}
10	10^3	1000	1000	0.0031	0.35	0
10	10^6	1000	1000	0.0030	0.34	0
10	10^{12}	1000	1000	0.0041	0.33	0
30	10^{12}	1000	1000	0.0113	0.56	0
50	10^{12}	1000	1000	0.0252	0.59	0.009
100	10^{12}	1000	1000	0.1552	0.66	0.081
200	10^{12}	1000	1000	1.7579	3.22	0.701

Sur des polynômes aléatoires l'algorithme *Polytope-Newton-modulaire* est efficace, même pour des polynômes denses.

Remarque : Il serait intéressant d'avoir une stratégie permettant de choisir un nombre p premier pour lequel le test d'irréductibilité dans $\mathbb{F}_p[X, Y]$ soit vérifié et ceci rapidement.
Nous avons essayé la méthode suivante : Trier les éléments de L dans l'ordre décroissant. L'objectif étant de commencer avec de grands p premiers, car nous avons le résultat suivant :

Théorème 32. *La proportion $\mathcal{P}$ de polynômes irréductibles de $\mathbb{F}_p[X, Y]$ de degré au plus n est minorée par :*

$$1 - \frac{1}{p^{n-1}}(1 + \frac{6}{p}) \leq \mathcal{P}.$$

Donc plus p est grand, plus nous avons de chances d'obtenir un polynôme irréductible modulo p, donc plus nous avons de chances de terminer rapidement l'algorithme *Polytope-Newton-modulaire.* Cependant en pratique le choix d'une liste L triée par ordre décroissant n'est pas toujours judicieux. En effet pour un polynôme de degré $n = 50$, avec ces coefficients dans $[-10^{12}, 10^{12}]$ nous avons obtenu pour L non ordonnée une réponse en 3.8 s. Pour ce même polynôme, avec L triée par ordre décroissant nous avons obtenu une réponse en 11.9 s. Nous avons aussi obtenu des résultats dans l'autre sens : Pour un polynôme de degré 50 avec ses coefficients dans $[-10^{12}, 10^{12}]$ nous avons obtenu une réponse en 6.1 s pour L non ordonnée, et en 0.2 s pour L ordonnée de façon décroissante.

5.3.3 L'utilisation modulaire avec recherche des racines

Nous avons vu dans la partie précédente que l'algorithme *Polytope-Newton-modulaire* donne de très bon résultats sur des polynômes génériques. L'inconvénient de cet algorithme est qu'il n'est pas adapté aux polynômes du type $x^n + y^n + 1$. En effet avec un tel polynôme l'algorithme n'apporte rien car modulo p le polytope de Newton reste inchangé puisque $L = \emptyset$.
Cependant nous avons vu en 1.6.2, l'algorithme Ragot, et, lui aussi passe par du calcul modulaire. Nous allons à présent voir comment compléter l'algorithme Ragot avec notre critère.

Tout d'abord nous remarquons la chose suivante :

$(0,0)$ *est racine simple de* $P \Longrightarrow$ *La condition (Polytope-Newton) est vérifiée.*

En effet si $(0,0)$ est racine simple de P alors le terme constant de P est nul et le coefficient de P en X ou en Y est non nul. Donc $Newt(P)$ a un sommet ayant pour coordonnées $(0,1)$ ou $(1,0)$.

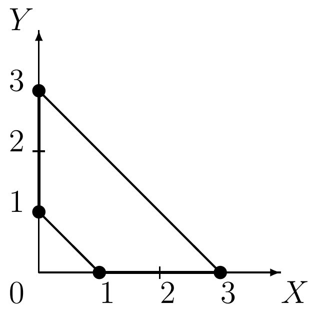
Nous allons à présent élargir la condition "$(0,0)$ est un zéro simple de P" de façon de telle manière à conserver l'implication ci-dessus.

Lemme 41. *Soient $P \in \mathbb{K}[X,Y]$, un polynôme de degré total n, et $a,b \in \mathbb{K}$. Supposons que ayons :*
- *Soit (a,b) est un zéro simple de P,*
- *soit $\frac{\partial P}{\partial X^n}(a,b) = 0$, et, $\frac{\partial P}{\partial X^{n-1}\partial Y}(a,b) \neq 0$ ou $\frac{\partial P}{\partial X^{n-1}}(a,b) \neq 0$,*
- *soit $\frac{\partial P}{\partial Y^n}(a,b) = 0$, et, $\frac{\partial P}{\partial Y^{n-1}\partial X}(a,b) \neq 0$ ou $\frac{\partial P}{\partial Y^{n-1}}(a,b) \neq 0$,*

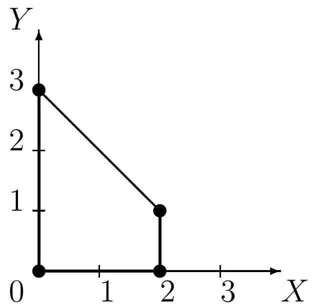
alors la condition (Polytope-Newton) est vérifiée pour $P(X-a, Y-b)$.

Démonstration. Nous pouvons nous ramener au cas $a = b = 0$. Le premier cas a été traité plus haut. Supposons $\frac{\partial P}{\partial X^n}(a,b) = 0$, et, $\frac{\partial P}{\partial X^{n-1}\partial Y}(a,b) \neq 0$, alors nécessairement $(n-1, 0)$ est un sommet de $Newt(P)$. Comme P est un polynôme de degré total n, il existe un sommet de $Newt(P)$ avec pour coordonnées (i,j) telles que $i + j = n$. Comme $pgcd(n-1, i, j) = 1$ la condition (Polytope-Newton) est vérifiée.
Supposons $\frac{\partial P}{\partial X^n}(a,b) = 0$, et, $\frac{\partial P}{\partial X^{n-1}}(a,b) \neq 0$, alors dans ce cas $(n-1, 1)$ est un sommet de $Newt(P)$. 1 étant une coordonnées d'un sommet la condition (Polytope-Newton) est vérifiée. □

Remarque : L'algorithme Ragot revenait à dire :
Si nous avons un polytope de Newton du type suivant :



alors nous pouvons conclure.
Ici nous rajoutons d'autres types de polytopes facilement identifiable à l'aide des dérivées de f. Par exemple :



Définition 36. *Si nous sommes dans une des trois conditions du lemme 41 alors nous dirons que la condition (Ragot-complétée) est vérifiée en* (a, b).

Nous avons alors l'algorithme suivant :

Algorithme Ragot-complété

ENTRÉE : $P(X,Y) \in \mathbb{Z}[X,Y]$, un polynôme irréductible dans $\mathbb{Q}[X,Y]$.
SORTIE : "P est absolument irréductible" ou "Je ne sais pas".
Pour p premier allant de 2 à 101 (par exemple) faire :
    Pour $(a,b) \in \mathbb{F}_p^2$ faire :
        $F(X,Y) = P(X,Y) \mod p$;
        Si $\deg(F) = \deg(P)$ alors
            Si F vérifie la condition (Ragot-complétée) en (a,b) alors
                Si F est irréductible dans $\mathbb{F}_p$ alors rendre "P est absolument irréductible" ; Fin Si ;
            Fin Si ;
        Fin Si ;
    Fin Pour ;
Fin Pour ;
Rendre "Je ne sais pas".

Remarque : Ici nous n'utilisons pas de polytope de Newton dans l'algorithme. Dans cet algorithme si nous remplaçons 101 par un nombre premier suffisamment grand dépendant du degré de P, alors nous pourrons reconnaître tous les polynômes absolument irréductibles. Cette borne étant trop élevée nous adoptons comme J.-F. Ragot une approche probabiliste.
La probabilité de succès de l'algorithme *Ragot-complété* est supérieure à celle de l'algorithme *Ragot*. En effet la condition utilisée est plus large.

Exemple : Le polynôme $P(X,Y) = Y^3+Y^2+X^2$ est irréductible dans $\mathbb{F}_2[X,Y]$, il n'a pas de zéros simple en $(0,0)$, mais vérifie la condition (Ragot-complétée) en $(0,0)$. Nous voyons donc que $g(X,Y) = Y^3 + 3Y^2 + 5X^2 + 6X + 2 \in \mathbb{Z}[X,Y]$ est absolument irréductible (sur $\mathbb{Q}$) dès le premier essai grâce à l'algorithme Ragot-complété, ce qui n'est pas le cas avec l'algorithme Ragot.

5.3.4 L'utilisation modulaire avec changement de variable

Dans cette partie nous allons généraliser les algorithmes précédents. Nous allons tester la condition (Polytope-Newton) sur le polynôme $P(X+a, Y+b) \mod p$.

Algorithme Polytope-Newton-modulaire-chg-var
Entrée : $P(X,Y) \in \mathbb{Z}[X,Y]$, un polynôme irréductible dans $\mathbb{Q}[X,Y]$.
Sortie : "P est absolument irréductible" ou "Je ne sais pas".

Pour p premier allant de 2 à 101 (par exemple) faire :
Pour $(a,b) \in \mathbb{F}_p^2$ faire :
$F(X,Y) = P(X+a, Y+b) \mod p$;
Si $\deg(F) = \deg(P)$ alors
Si F vérifie la condition (Polytope-Newton) alors
Si F est irréductible dans $\mathbb{F}_p$ alors rendre "P est absolument irréductible" ; Fin Si ;
Fin Si ;
Fin Si ;
Fin Pour ;
Fin Pour ;
Rendre "Je ne sais pas".

Remarque : Cet algorithme a plus de chances de succès que l'algorithme Ragot-complété (donc aussi plus de chances de succès que l'algorithme Ragot) car ici nous testons la condition (Polytope-Newton) et non pas une condition plus forte. Cependant ici nous effectuons un changement de coordonnées alors que dans l'algorithme *Ragot-complété* nous évaluons des points en des dérivées de P.

Exemple :

$$\begin{aligned} P(X,Y) &= X^{10} + X^4(Y-1)^6 + 13X^4(Y-1)^4 + 123X^3(Y-1)^2 \\ &\quad +11X^2(Y-1)^8 + X^2(Y-1)^3 + (Y-1)^{10} \in \mathbb{Z}[X,Y] \end{aligned}$$

Ce polynôme est absolument irréductible, mais nous ne pouvons pas conclure avec l'algorithme Polytope-Newton-modulaire. Avec l'algorithme Ragot nous obtenons "P est absolument irréductible" pour $p = 5$, après 16 recherches de racines simples (sans compter les recherches modulo $p = 2, 3$). Mais avec l'algorithme *Polytope-Newton-modulaire-chg-var* nous obtenons "P est absolument irréductible" pour $p = 2$, $a = 0$ et $b = 1$. (La condition(Polytope-Newton) est vérifiée mais la condition (Ragot-complétée) ne l'est pas pour $p = 2$, $a = 0$, $b = 1$.)

5.4 Conclusion

Dans ce chapitre nous venons de donner une condition suffisante d'irréductibilité absolue pour des polynômes irréductibles. Cette condition nous a permis d'obtenir différents algorithmes pour tester l'irréductibilité absolue d'un polynôme $P \in \mathbb{Z}[X,Y]$. Ces algorithmes tirent profit du calcul modulaire (comme dans l'algorithme de J.-F. Ragot, voir section 1.6), et de l'information contenue dans le polytope de Newton (comme dans les tests de S. Gao, voir section 1.6). Cette démarche nous permet d'obtenir des algorithmes généralisant celui de J.-F. Ragot. De plus ces algorithmes peuvent être étendus au cas où $P(X,Y) \in \mathbb{A}[X,Y] \subset \mathbb{K}[X,Y]$, où $\mathbb{A}$ est un anneau et $\mathbb{K}$ un corps parfait. Dans ce cas les calculs modulo un nombre premier p, seront remplacés par des calculs modulo un idéal premier I.

Nous avons vu aux pages 129 et 131, aprés une étude sur des exemples, que la condition suffisante de notre test est trés souvent, ou, presque toujours vérifiée. Une étude de la probabilité de réussite de notre test nous permettrait de quantifier plus rigoureusement ce résultat. De plus, les tests proposés ici se généralisent au cas des polynômes en n variables. Nous pourrions alors étudier cette probabilité de réussite en fonction du nombre de variables. Cette probabilité sera d'autant plus grande que n sera grand. En effet, si nous avons n variables chaque sommet du polytope aura n coordonnées. Donc plus n sera grand plus nous aurons de coordonnées, et plus la probabilité que ces entiers soient premiers entre eux sera grande. Une telle étude est en cours.

Troisième partie

Annexes

Annexe A

Éléments pour le théorème de Harris affine

Cette annexe est un extrait du chapitre rédigé par G. Chèze et A. Galligo pour le livre [CG05]. *Ici nous donnons quelques éléments de topologie et de géométrie afin de présenter le théorème de Harris affine.*

A.1 Basic definitions and classical results

Here we recall some topological classical results, for the proof we refer e.g. to [Rot88].

Definition 37. *Let X be a topological space, and x_0 a point of X.*
Let $\Gamma(X, x_0) = \{\gamma \in \mathcal{C}^0([0,1], X) \mid \gamma(0) = \gamma(1) = x_0\}$ be the loops space on X. Homotopy between loops, denoted by the symbol $\sim$, is an equivalence relation on $\Gamma(X, x_0)$. We denote by $[\gamma]$ the homotopy class of γ, and by $\pi_1(X, x_0)$ the set $\dfrac{\Gamma(X, x_0)}{\sim}$.

It can be shown that $\pi_1(X, x_0)$ equipped with the concatenation is a group, in general non commutative.

Example Take for X the complement in the real plane of a set of N points; $X = \mathbb{R}^2 \setminus \{p_1, ..., p_N\}$. Then $\pi_1(X, x_0)$ is a free group generated by N loops around the points p_i.

Definition 38. $\Pi : Y \to X$ *is a covering if* Π *is continuous and if all $x \in X$ have an open neighborhood U_x such that $\Pi^{-1}(U_x)$ is a disjoint union of open sets V_i in Y, with $\Pi_{/V_i} : V_i \to U_x$ is an homeomorphism for every i.*
We call $\Pi^{-1}_{/V_i} : U_x \to V_i$ a section of Π.

Proposition 25. *Let X be a connected topological space, and let $\Pi : Y \to X$ be a covering. If there exist $x_0 \in X$ such that the cardinal of $\Pi^{-1}(x_0)$ satisfy $|\Pi^{-1}(x_0)| = N$*

then for all x in X, we have $|\Pi^{-1}(x)| = N$. In this situation, Π (or Y) is called an $N-$fold covering.

exercise 1. *Let $P(X,Y) = Y^n + a_{n-1}(X)Y^{n-1} + \ldots + a_0(X)$ be a polynomial in $\mathbb{C}[X,Y]$, $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 \mid P(x,y) = 0\}$, $pr_1 : \mathbb{C}^2 \to \mathbb{C}$ be the projection on the first coordinate and $\Delta = \{x \in \mathbb{C} \mid Disc_Y(P(X,Y))(x) = 0\}$.*
Show that $pr_{1/\mathcal{C}-pr_1^{-1}(\Delta)} : \mathcal{C} - pr_1^{-1}(\Delta) \to \mathbb{C} - \Delta$ is an $n-$fold covering.
(Hint : use the implicit function theorem.)

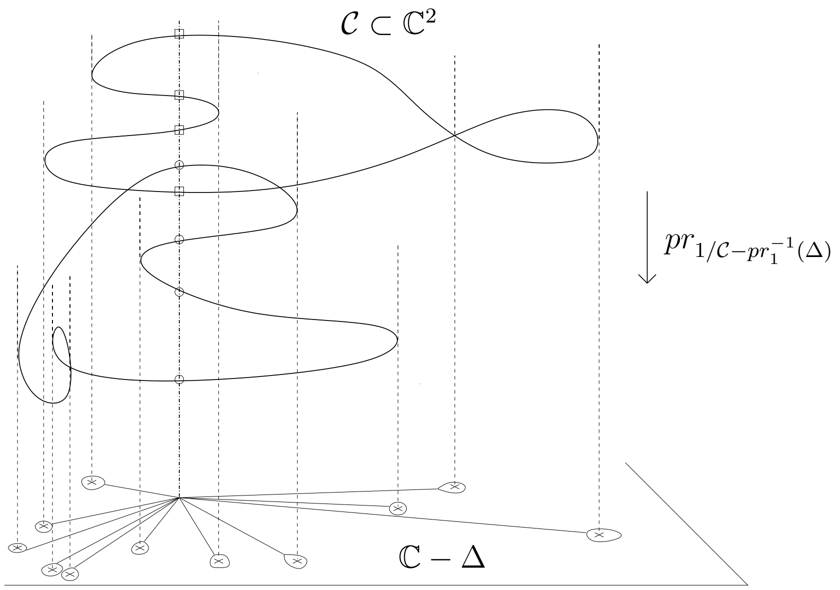


FIG. A.1 – A ramified covering with a smooth generic fiber.

Theorem 33 (Lifting lemma). *Let X be a connected topological space, $\Pi : Y \to X$ be a covering and $\gamma : [0,1] \to X$ a path such that $\gamma(0) = \gamma(1) = x_0$.*

If y_0 is in the fiber over x_0 (i.e. $\Pi(y_0) = x_0$), then there exist a unique path $\tilde{\gamma}_{y_0} : [0,1] \to Y$ such that $\tilde{\gamma}_{y_0}(0) = y_0$ and $\Pi \circ \tilde{\gamma}_{y_0} = \gamma$.

With this lifting, we can define a group action on the fiber.

Proposition 26. *Let X be a connected topological space, let $\Pi : Y \to X$ be a N-covering, and x_0 a point of X. We denote by $\mathcal{F}$ the fiber over x_0 (i.e. $\mathcal{F} = \Pi^{-1}(x_0)$). We have a group action :*

$$\pi_1(X, x_0) \times \mathcal{F} \to \mathcal{F}$$

$$([\gamma], y_0) \mapsto \tilde{\gamma}_{y_0}(1)$$

and a group morphism :

$$\pi_1(X, x_0) \to Aut(\mathcal{F}) = \mathfrak{S}_N$$

where $\mathfrak{S}_N$ is the symmetric group.

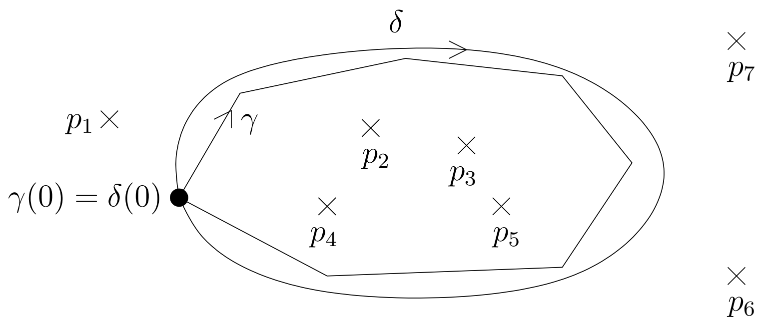


FIG. A.2 – γ is homotopic to the analytic path δ.

Now we give two useful lemmas about analytic paths.

Lemme 42. *Let γ be a closed path in $\mathbb{C} \setminus \{p_1, \dots, p_d\}$, then γ is homotopic to an analytic closed path δ.*

Proof. We have a continuous map $\gamma : [0,1] \to \mathbb{C} \setminus \{p_1, \dots, p_d\}$ such that $\gamma(0) = \gamma(1)$. We set $\mathcal{E} = \{f \in \mathcal{C}^0([0,1], \mathbb{C}) \mid f(0) = f(1)\}$. Furthermore $S = Span(\{e^{2i\pi n\theta} \mid n \in \mathbb{Z}\})$ is a dense subset of $\mathcal{E}$ for the $\|.\|_\infty$ norm ($\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|$).

Now the distance between $\gamma([0,1])$ and $\{p_1, \dots, p_d\}$ is strictly bigger than 0, because these two compact sets are such that $\gamma([0,1]) \cap \{p_1, \dots, p_d\} = \emptyset$. So we set $d(\gamma([0,1]), \{p_1, \dots, p_d\})/4 = \epsilon$ and we have $\epsilon > 0$.

Because of the density of S there exists a sequence $(f_n)_n \in S$ with the following property : there exists a number N such that for all $n \geq N$ we have $\|\gamma - f_n\|_\infty < \epsilon$. We set $\delta(t) = f_N(t) - f_N(0) + \gamma(0)$.

Then,

$$\begin{aligned} \|\delta(t) - \gamma(t)\|_\infty &\leq \|f_N(t) - \gamma(t)\|_\infty + \|f_N(0) - \gamma(0)\| \\ &< d(\gamma([0,1]), \{p_1, \dots, p_d\}). \end{aligned}$$

So δ is homotopic to γ, δ is analytic and $\delta(0) = \delta(1) = \gamma(0) = \gamma(1)$. $\square$

Lemme 43. *The lifting $(\tilde{\gamma})$ of an analytic path γ in theorem 33 is analytic.*

exercise 2. *Prove lemma 43. (Hint : Use the implicit function theorem.)*

A.2 Irreducibility and path connected space

Theorem 34. *Let $P(X,Y)$ be a square free polynomial of $\mathbb{C}[X,Y]$, $\Delta = \{x \mid Disc_Y(P(X,Y))(x) = 0\}$, and $\mathcal{C} = \{(x,y) \in \mathbb{C}^2 \mid P(x,y) = 0\}$. Then*

$$P \text{ is irreducible in } \mathbb{C}[X,Y] \iff \mathcal{C} - pr_1^{-1}(\Delta) \text{ is path connected.}$$

We need two lemmas to prove this theorem.

Lemma 44. *Let $\Pi : Y \to X$ be an $n-$fold covering, and Y_1 be a connected component of Y, then $\Pi_{/Y_1} : Y_1 \to X$ is a $d-$fold covering with $d \leq n$.*

exercise 3. *Prove lemma 44.*

Lemma 45. *Let $P(X,Y) = Y^n + a_1(X)Y^{n-1} + \ldots + a_n(X)$, $x \in \mathbb{C}$, and $y(x)$ be a root of $P(x,Y)$, then $|y(x)| \leq \max(1, \sum_{i=1}^n |a_i(x)|) \leq 1 + \sum_{i=1}^n |a_i(x)|$.*

Proof. If $|y(x)| \leq 1$ then the lemma is true.
If $|y(x)| \geq 1$ then $P(x,y(x)) = 0$ thus $y(x)^n = a_1(x)y(x)^{n-1} + \ldots + a_n(x)$. It follows : $|y(x)|^n \leq \sum_{j=1}^n |a_i(x)||y(x)|^{n-i} \leq \sum_{j=1}^n |a_i(x)||y(x)|^{n-1}$. So $|y(x)| \leq \sum_{j=1}^n |a_i(x)|$. $\square$

Proof. Theorem 34. $\Rightarrow$) We suppose P irreducible. First we remark that : $\mathcal{C} - pr_1^{-1}(\Delta)$ is locally path connected because $\mathbb{C} - \Delta$ is locally path connected. So it suffices to show that $\mathcal{C} - pr_1^{-1}(\Delta)$ is connected.

Let C_1 be a connected component of $\mathcal{C} - pr_1^{-1}(\Delta)$, lemma 44 implies $pr_{1/C_1} : C_1 \to \mathbb{C} - \Delta$ is a $d-$fold covering with $d \leq n$. Thus if we show that $d = n$ then we have $C_1 = \mathcal{C} - pr_1^{-1}(\Delta)$ and we are done.

For every $x_0 \in \mathbb{C} - \Delta$, we have $pr_{1/C_1}^{-1}(x_0) = \{y_1(x_0), \ldots, y_d(x_0)\}$ with $y_i(x_0) \neq y_j(x_0)$ when $i \neq j$. As pr_{1/C_1} is a covering we have a neighborhood V_{x_0} of x_0 such that $y_1, \ldots, y_d$ are defined on V_{x_0}. Furthermore we have for every $x \in V_{x_0}$, $pr_{1/C_1}^{-1}(x) = \{y_1(x), \ldots, y_d(x)\}$. Hence y_i is analytic on V_{x_0}, by the implicit function theorem applied to $P(x_0, y_i(x_0))$.
We consider the polynomial :

$$\mathcal{P}(x,Y) = (Y - y_1(x)) \ldots (Y - y_d(x)) = Y^d + S_1(x)Y^{d-1} + \ldots + S_d(x).$$

The $S_i(x)$ are the elementary symmetric functions in $y_i(x)$. Each $S_i(x)$ is defined on V_{x_0}, and we now see that $S_i(x)$ is a polynomial. First, we show that $S_i(x)$ is defined on $\mathbb{C} - \Delta$, and in a second time we show that $S_i(x)$ is defined on $\mathbb{C}$, and bounded by a polynomial.

Let $x_1 \neq x_0$, as before there exits a neighborhood U_{x_1} of x_1 and d analytic functions $\varphi_1, \ldots, \varphi_d$, such that $pr_{1/C_1}^{-1}(x) = \{\varphi_1(x), \ldots, \varphi_d(x)\}$ for every x in V_{x_1}. If $V_{x_1} \cap V_{x_0} \neq \emptyset$ then as pr_{1/C_1} is a covering and y_i and φ_i are sections of pr_{1/C_1} we have an element $\sigma \in \mathfrak{S}_d$ such that : $y_i = \varphi_{\sigma(i)}$ on $V_{x_0} \cap V_{x_1}$. Thereby, we have for example $s_1(x) = \sum_{i=1}^d y_i(x) = \sum_{i=1}^d \varphi_{\sigma(i)}(x) = \sum_{i=d}^d \varphi_i(x)$ on $V_{x_1} \cap V_{x_0}$, thus $S_1(x)$ is defined and analytic in $V_{x_0} \cup V_{x_1}$. So if we repeat this previous step we get d analytic functions $S_i(x)$ defined on $\mathbb{C} - \Delta$. Now we have to prove that $S_i(x)$ is defined on $\mathbb{C}$. Let us do it for $S_1(x)$.

Let x_0 be in Δ, and let $(\epsilon_n)_n$ be a sequence of element in $\mathbb{C}$ such that $\lim_{n\to\infty} \epsilon_n = 0$. Consider $\lim_{n\to\infty} S_1(x_0+\epsilon_n)$. This is equal to $\lim_{n\to\infty} \sum_{i=1}^d y_i(x_0+\epsilon_n)$,

with $y_i(x)$ well defined and continuous in x_0 (as they are the roots of the polynomial $P(x, Y)$). We get $\lim_{n\to\infty} S_1(x_0 + \epsilon_n) = \sum_{i=1}^{d} y_i(x_0)$ (here there exist i_0 and j_0 such that $y_i(x_0) = y_j(x_0)$). Thus there are no singularity on Δ, and we can extend analytically S_1 to $\mathbb{C}$. We proceed similarly for all S_i.
Let x be in $\mathbb{C}$, and $y_i(x)$ be a root of $P(x, Y)$. Lemma 45 implies $|y_i(x)| \leq 1 + \sum_{j=1}^{n} |a_j(x)|$, this means that $|S_i(x)|$ is bounded by a polynomial, then Liouville's theorem implies that $S_i(x)$ is a polynomial.

Thereby : $\mathcal{P}(X, Y) = Y^d + S_1(x)Y^{d-1} + \ldots + S_d(X)$ belongs to $\mathbb{C}[X, Y]$. Now we perform the euclidean division of P by $\mathcal{P}$, in $\mathbb{C}(X)[Y]$. As $\mathcal{P}$ is monic we get :

$$P(X, Y) = A(X, Y)\mathcal{P}(X, Y) + R(X, Y),$$

with $A(X, Y)$, $R(X, Y) \in \mathbb{C}[X, Y]$ and $R(X, Y) = r_{d-1}(X)Y^{d-1} \ldots + r_0(X)$. For every $x \notin \Delta$, we set $\{y_1(x), \ldots, y_d(x)\} = \{y \,|\, P_1(x, y) = 0\}$, where $y_i(x) \neq y_j(x)$ if $i \neq j$. As $(x, y_i(x)) \in C_1 \subset \mathcal{C} - pr_1^{-1}(\Delta)$ we have : $P(x, y_i(x)) = 0$ for $i = 1, \ldots, d$. Hence $R(x, y_i(x)) = 0$ for $i = 1, \ldots, d$, and thus $R(x, Y) = 0$ in $\mathbb{C}[Y]$ for every $x \notin \Delta$. So $r_{d-1}(X) = \ldots = r_0(X) = 0$ in $\mathbb{C}[X]$, and then : $\mathcal{P}$ divides P. Now as P is irreducible, it follows that $\mathcal{P} = P$ and then $d = n$.

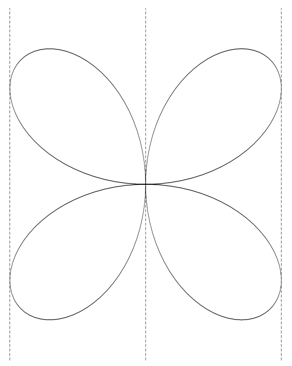
$\Leftarrow$) We suppose $P = P_1.P_2$ where P_i is irreducible in $\mathbb{C}[X, Y]$, and $P_1 \neq P_2$ because P is square free (if we have more than two factors the proof is similar). We set $\mathcal{V}(P_i) = \{(x, y) \in \mathbb{C}^2 \,|\, P_i(x, y) = 0\}$ and $C_i = \mathcal{V}(P_i) \cap (\mathcal{C} - pr_1^{-1}(\Delta))$. C_i is a closed subset of $\mathcal{C} - pr_1^{-1}(\Delta)$.
Furthermore C_1 and C_2 are distinct, because $pr_1(\mathcal{V}(P_1) \cap \mathcal{V}(P_2)) \subset Disc_Y(P)$. Indeed we have $Disc_Y(P) = Disc_Y(P_1).Disc_Y(P_2).Res_Y(P_1, P_2)^2$. So we can conclude that $\mathcal{C} - pr_1^{-1}(\Delta) = C_1 \sqcup C_2$, and then that $\mathcal{C} - pr_1^{-1}(\Delta)$ is not connected. $\square$

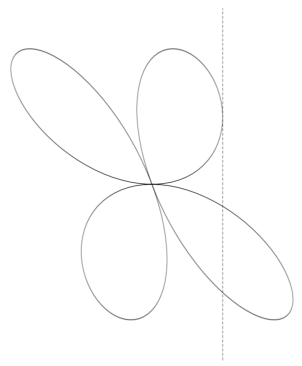
A.3 Double point set and transpositions

Lemma 46 (Change of coordinates). *Let $P(X, Y) \in \mathbb{Q}[X, Y]$ of total degree n. We consider the change of coordinate : $f_\lambda(X, Y) = P(X + \lambda Y, Y)$.*
Let U be the subset of $\mathbb{Q}$ such that for every λ in U we have : $\deg_Y(f_\lambda(X, Y)) = n$, and there exists x_0 in $\mathbb{C}$ such that the polynomial $f_\lambda(x_0, Y) \in \mathbb{C}[Y]$ has one root y_0 of multiplicity two, all the other have multiplicity one, and $(\partial f_\lambda/\partial X)(x_0, y_0) \neq 0$.
Then there exists F a finite subset of $\mathbb{Q}$ such that $U = \mathbb{Q} - F$.

Proof. First we show that $V = \{\lambda \in \mathbb{Q} \,|\, \deg_Y(f_\lambda(X, Y)) \neq n\}$ is a finite subset of $\mathbb{Q}$. We set $P(X, Y) = \sum_{k+l \leq n} a_{k,l} X^k Y^l$ then

$$\begin{aligned} f_\lambda(X, Y) &= \sum_{k+l \leq n} a_{k,l} \left(\sum_{i=0}^{k} \binom{k}{i} \lambda^i Y^i X^{k-i}\right) Y^l \\ &= a_n(\lambda)Y^n + a_{n-1}(X, \lambda)Y^{n-1} + \ldots + a_0(X, \lambda) \text{ where } a_n(\lambda) \in \mathbb{Q}[\lambda]. \end{aligned}$$





$(X^2+Y^2)^3 - 4X^2Y^2 = 0$ $\qquad$ $((X+\frac{1}{2}Y)^2+Y^2)^3 - 4(X+\frac{1}{2}Y)^2Y^2 = 0$

FIG. A.3 – Examples of a bad case $f_0(X,Y) = P(X,Y) = 0$ and of a good case $f_{1/2}(X,Y) = 0$

Thus $V = \{\lambda \,|\, a_n(\lambda) = 0\}$, hence V is finite.

Now we consider $d_1(\lambda, X) = Disc_Y(f_\lambda(X,Y)) \in \mathbb{Q}[\lambda, X]$. If (u_0, v_0) is a singular point of $P(X,Y)$ then $(u_0 - \lambda v_0, v_0)$ is a singular point of $f_\lambda(X,Y)$. So $d_1(\lambda, u_0 - \lambda v_0) = 0$ and $X - (x_0 - \lambda v_0)$ divides $d_1(\lambda, X)$. We denote by (u_i, v_i), for $i = 1, \ldots, d$ the singular points of P, then $d_1(\lambda, X) = (\prod_{i=1}^d (X - (u_i - \lambda v_i))^{e_i}) q(\lambda, X)$ and we set $d_2(\lambda) = Disc_X(q(\lambda, X))$.

Now we claim that if λ is not a root of d_2 and if the change of coordinates preserves the degree in Y of P then λ belongs to U, this will prove the lemma. Indeed, we choose (λ_0, x_0) in the following way :

$$\begin{cases} d_2(\lambda_0) \neq 0 & (1) \\ d_1(\lambda_0, x_0) = 0 & (2) \\ x_0 \neq u_i - \lambda_0 v_i & (3) \end{cases}$$

In order to satisfy (2) and (3), we choose a root x_0 of $q(\lambda_0, X)$ which is not $u_i - \lambda_0 v_i$.

Now we consider the roots $y_i(X) \in \overline{\mathbb{C}[X]}$ of the polynomial $f_{\lambda_0}(X,Y) \in \mathbb{C}[X,Y]$ and we get : $d_1(\lambda_0, X) = \prod_{i \neq j}(y_i(X) - y_j(X))$. So there exists i_0 and j_0 such that $y_i(x_0) = y_{j_0}(x_0)$ with $i_0 \neq j_0$ (because of (2)). Therefore $d_2(\lambda_0) \neq 0$ so $q(\lambda_0, X)$ do not have a multiple root, and $x_0 \neq u_i - \lambda_0 v_i$, so $\frac{\partial d_1}{\partial X}(\lambda_0, x_0) \neq 0$. Furthermore :

$$\begin{aligned} \frac{\partial d_1}{\partial X}(\lambda_0, X) &= (\frac{\partial y_{i_0}}{\partial X}(X) - \frac{\partial y_{j_0}}{\partial X}(X)) \prod_{\substack{(i,j)\neq(i_0,j_0) \\ i \neq j}} (y_i(X) - y_j(X)) \\ &\quad + (y_{i_0}(X) - y_{j_0}(X)) \frac{\partial}{\partial X}(\prod_{\substack{(i,j)\neq(i_0,j_0) \\ i \neq j}} (y_i(X) - y_j(X))). \end{aligned}$$

Thus $\frac{\partial d_1}{\partial X}(\lambda_0, x_0) \neq 0$ implies that for all $(k,l) \neq (i_0, j_0)$ and $k \neq l$ we have $y_k(x_0) \neq y_l(x_0)$.

Then we conclude that $f_{\lambda_0}(x_0, Y)$ has $n-1$ distinct roots, and one root has multiplicity two ($y_{i_0}(x_0) = y_{j_0}(x_0)$). As $\frac{\partial f_{\lambda_0}}{\partial X}(x_0, y_{i_0}) \neq 0$ (because x_0 is not the abscissa of a singular point) the claim is proved. □

Theorem 35. *Let λ_0 be as in lemma 46 , $\Delta = Disc_Y(f_{\lambda_0}(X,Y))$. Then there exists $X_0 \in \mathbb{C} - \Delta$, γ a path in $\mathbb{C} - \Delta$ such that the monodromy action relatively to X_0 of γ on the fiber $f_{\lambda_0}^{-1}(X_0, Y) = \{Y_1, \ldots, Y_n\}$ is*

$$\left\{ \begin{array}{l} [\gamma].Y_{i_0} = Y_{j_0} \\ [\gamma].Y_{j_0} = Y_{i_0} \\ [\gamma].Y_i = Y_i \textit{ if } i \neq i_0 \textit{ and } i \neq j_0 \end{array} \right. , \textit{ where } i_0 \neq j_0.$$

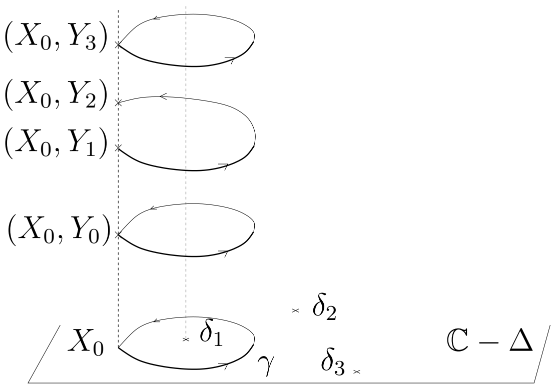


FIG. A.4 – The monodromy action of γ gives the transposition $(Y_1 \quad Y_2)$.

Proof. a) Parameterizations.
Let λ_0, x_0 and y_0 be as in lemma 46, we have :

$$f_{\lambda_0}(x_0, y_0) = \frac{\partial f_{\lambda_0}}{\partial Y}(x_0, y_0) = 0, \ \frac{\partial^2 f_{\lambda_0}}{\partial Y^2}(x_0, y_0) \neq 0 \text{ and } \frac{\partial f_{\lambda_0}}{\partial X}(x_0, y_0) \neq 0.$$

We denote by $y_3, \ldots, y_n$ the simple roots of $f_{\lambda_0}(x_0, Y)$. We have :

$$\left\{ \begin{array}{ll} \forall i \geq 3 & f_{\lambda_0}(x_0, y_i) = 0, \\ \forall i \geq 3 & \dfrac{\partial f_{\lambda_0}}{\partial Y}(x_0, y_i) \neq 0. \end{array} \right.$$

So we can apply to every (x_0, y_i) the analytic version of the implicit function theorem. Thus, there exits V_{y_0} a neighborhood of y_0, $U^0_{x_0}$ a neighborhood of x_0, and φ an analytic function such that :

$$x \in U^0_{x_0}, y \in V_{y_0} \text{ and } f_{\lambda_0}(x, y) = 0 \iff x = \varphi(y) \text{ and } y \in V_{y_0}.$$

For $i \geq 3$, there exist a neighborhood $U^i_{x_0}$ of x_0, V_{y_i} a neighborhood of y_i, and p_i an analytic function such that :

$$x \in U^i_{x_0}, y \in V_{y_i} \text{ and } f_{\lambda_0}(x,y) = 0 \iff y = p_i(x) \text{ and } x \in U^i_{x_0}.$$

Now, we consider the parametrization $x = \varphi(y)$.
We have : $\varphi(y) = x_0 + a(y - y_0) + b(y - y_0)^2 + \dots$ in a neighborhood of y_0, with

$$a = -\frac{\partial f_{\lambda_0}}{\partial Y}(x_0, y_0)(\frac{\partial f_{\lambda_0}}{\partial X}(x_0, y_0))^{-1} = 0,$$

and

$$b = -\frac{1}{2}\frac{\partial^2 f_{\lambda_0}}{\partial Y^2}(x_0, y_0)(\frac{\partial f_{\lambda_0}}{\partial X}(x_0, y_0))^{-1} \neq 0.$$

Thus in a neighborhood of (x_0, y_0) we have :

$$(x - x_0) = (y - y_0)^2[b + \sum_{k\geq 1} a_k(y - y_0)^k].$$

b) The parameterization $(x - x_0) = (\psi(y))^2$.
The equation $z^2 = b$ has two distinct nonzero roots r_1 and r_2. Near $r_1 \neq 0$ the function $c : \mathbb{C} \to \mathbb{C} : z \mapsto z^2$ has an analytic inverse because $c'(r_1) = 2r_1 \neq 0$. Let r be this inverse, $r : V_b \to W_{r_1}$. Hence, in a neighborhood $V^{(1)}_{y_0}$ of y_0 we define :

$$\begin{array}{rcl} \mathcal{R} : V^{(1)}_{y_0} & \to & W_{r_1} \\ y & \mapsto & r\big(b + \sum_{k\geq 1} a_k(y - y_0)^k\big). \end{array}$$

Thus in a neighborhood of (x_0, y_0) we have : $x - x_0 = (y - y_0)^2(\mathcal{R}(y))^2$.
We denote by ψ the following map : $\psi : V^{(2)}_{y_0} \to V_0 : y \mapsto (y - y_0)\mathcal{R}(y)$, where V_0 is a neighborhood of 0, and $V^{(2)}_{y_0}$ is a neighborhood of y_0 such that ψ is an isomorphism (this is possible because $\psi'(y_0) = \mathcal{R}(y_0) = r(b) \neq 0$). We denote then by ξ the inverse of ψ. Thus we have a neighborhood $\mathcal{V}_{y_0}$ of y_0, and a neighborhood $\mathcal{U}_{x_0}$ of x_0 such that :

$$(*) \quad x \in \mathcal{U}_{x_0}, y \in \mathcal{V}_{y_0} \text{ and } f_{\lambda_0}(x,y) = 0 \iff x - x_0 = \psi(y)^2 \text{ and } y \in \mathcal{V}_{y_0}$$

Now we remark that y_0 is a root of multiplicity two of $(\psi(y))^2 = 0$, where ψ is a non constant analytic function on $\mathcal{V}_{y_0}$. So we have established (see [Car61] p 97, Prop 4.2) :

$(**)$ There exist $\mathcal{V}'_{y_0}$ a neighborhood of y_0, $\mathcal{U}'_{x_0}$ a neighborhood of x_0 such that for all $X_0 \neq x_0$ and X_0 in $\mathcal{U}'_{x_0}$, $(\psi(y))^2 = X_0 - x_0$ has exactly two distinct simple roots in $\mathcal{V}'_{y_0}$.

We set $V = (\cap_{i\geq 3} U^i_{x_0}) \cap \mathcal{U}_{x_0} \cap \mathcal{U}'_{x_0}$, this is a neighborhood of x_0. Now we choose a real number $\rho > 0$ such that $\mathcal{B}(x_0, \rho) \subset V$ and $\mathcal{B}(0, \sqrt{\rho}) \subset V_0$.

c) Lifting paths.
We set : $X_0 = x_0 + \rho$ and $\gamma : [0,1] \to \mathbb{C} - \Delta : t \mapsto x_0 + \rho e^{2i\pi t}$. Thus $f_{\lambda_0}(X_0, Y)$ has n distinct roots : $Y_1, \ldots, Y_n$. Now we write all these roots with ξ or p_i.
If $X_0 \in V$ and $y \in \mathcal{V}'_{y_0}$, then $f_{\lambda_0}(X_0, y)$ has two distinct roots Y_1 and Y_2 in $\mathcal{V}'_{y_0}$. By $(**)$ and by $(*)$ we can set $\psi(Y_1) = \sqrt{\rho}$ and $\psi(Y_2) = -\sqrt{\rho}$. Hence $Y_1 = \xi(\sqrt{\rho})$ and $Y_2 = \xi(-\sqrt{\rho})$.
If $X_0 \in V$ and $y \in V_{y_i}$, $f_\lambda(X_0, y) = 0 \iff y = p_i(X_0)$ for $i = 3, \ldots, n$. So we set $Y_i = p_i(X_0)$.
Now we lift γ above Y_1 and Y_2. We set $\gamma_1(t) = \xi(\sqrt{\rho}e^{i\pi t})$ and $\gamma_2(t) = \xi(-\sqrt{\rho}e^{i\pi t})$. These two paths are well defined, continuous and $\gamma_i(0) = Y_i$.

Furthermore : $\forall t \in [0,1], \gamma(t) - x_0 = \rho e^{2i\pi t} = [\psi(\xi(\sqrt{\rho}e^{i\pi t}))]^2$ because $\psi \circ \xi =$ id , then $\forall t \in [0,1], \gamma(t) - x_0 = [\psi(\gamma_1(t))]^2 = [\psi(\gamma_2(t))]^2$.

By $(*)$ we get $f(\gamma(t), \gamma_i(t)) = 0, \forall t \in [0,1]$. Thus γ_1 lifts γ above Y_1 and γ_2 lifts γ above Y_2. As $\gamma_1(1) = Y_2$ and $\gamma_2(1) = Y_1$, we get : $[\gamma].Y_1 = Y_2$, $[\gamma].Y_2 = Y_1$.

Therefore we set $\gamma_i(t) = p_i(\gamma(t))$ for $i = 3, \ldots, n$. We have :
$f(\gamma(t), \gamma_i(t)) = 0, \forall t \in [0,1]$ (by definition of p_i). γ_i lifts γ above Y_i and $\gamma_i(1) = p_i(\gamma(1)) = p_i(\gamma(0)) = Y_i$. Hence for $i = 3, \ldots, n, [\gamma].Y_i = Y_i$. □

Transpositions will play an important role for the proof of Harris' lemma and its generalizations (see theorem 36). Theorem 35 will be used with the following lemma.

Définition 39. *Let $G \times X \to X$ be a group action. If for every two pairs of points x_1, x_2 and y_1, y_2 ($x_1 \neq x_2$ and $y_1 \neq y_2$) there is a group element g such that $g.x_i = y_i$, then the group action is called $2-$transitive.*

Lemme 47. *Let G be a subgroup of $\mathfrak{S}_n$ such that the action of G on $\{1, \ldots, n\}$ is $2-$transitive and such that there exists a transposition τ in G, then $G = \mathfrak{S}_n$.*

Démonstration. We can suppose $\tau = (1 \quad 2)$. Let x and y be two distinct elements of $\{1, \ldots, n\}$. There exists a permutation $\sigma \in G$ such that $\sigma(1) = x$ and $\sigma(2) = y$ (because the action is $2-$transitive).
Then $\sigma^{-1}\tau\sigma = (\sigma(1) \quad \sigma(2)) = (x \quad y) \in G$. Therefore every transposition belongs to G, this implies that $G = \mathfrak{S}_n$. □

A.4 Monodromy and genericity

We have seen in the previous subsection that when P is irreducible, the monodromy action on the fiber is transitive. That is to say, any two points y_i and y_j of the fiber $\phi^{-1}(x_0)$ can be exchanged following a continuous path on the curve on top of some loop γ. This result also expresses the connexity of the subspace formed by the curve $\mathcal{C}$ minus the ramification points. In fact there is a stronger connexity result which is a consequence of a lemma due to J. Harris (see [Har80] or [ACGH85]), which was originally used to establish his Uniform position theorem on the generic hyperplane sections of a projective curve.

We will adapt Harris lemma to our setting in order to obtain what we call an Affine Harris theorem. This theorem says that if we perform a **generic** change of coordinates before taking the projection, not only the action of the monodromy group is transitive but any permutation of the point of the fiber $\phi^{-1}(x_0)$ can be obtained following a continuous path on the curve on top of some loop γ. This key fact and its application to absolute factorization was first observed by Galligo, stated in [GW97], then in [Gal99] and in [Rup04]. However in these papers it was just indicated that this statement was a consequence of Harris' lemma and classical arguments in algebraic geometry. As Sommese,Verschelde,Wampler needed this statement to improve their algorithm, in [SVW02c] they gave a more complete proof of it and made precise references to two textbooks ([ACGH85] and [GM83]). In the next subsection, we will give a detailed exposition of this result. Let us start by reviewing Harris' lemma and its proof in the case of planar curves.

Lemma 48 (Harris' lemma). *Let $\mathcal{C}$ be an irreducible projective planar curve possibly singular, call n its degree. Let U be the Zariski open subset in $\mathbb{P}^2(\mathbb{C})^*$ of lines transverse to $\mathcal{C}$, i.e. cutting $\mathcal{C}$ in n simple points. Consider the incidence correspondence graph I and its second projection :*

$$pr_2 : I = \{(p, H) \in \mathcal{C} \times U \quad | \quad p \in H\} \to U.$$

Then, pr_2 is a n-fold topological covering. We fix $H_0 \in U$ and let Γ_0 denote the set of n intersection points $\mathcal{C} \cap H_0$. Then the monodromy map

$$\pi_1(U, H_0) \to \mathfrak{S}_n$$

is surjective.

Proof. Let G be the image of the monodromy map. We know by application of Theorem 34 that $\mathcal{C}$ minus its singular locus is path connected. As by theorem 35, U contains in its border a line H_1 which is tangent to $\mathcal{C}$ at a simple point and transverse to $\mathcal{C}$ at all the other $n-2$ intersection points, we deduce that G contains a transposition. So in order to apply lemma 47 we need only to prove that G is 2-transitive. To express this property geometrically let us set

$$I_2 = \{(p_1, p_2, H) \in \mathcal{C} \times \mathcal{C} \times U \quad | \quad p_1 \in H, p_2 \in H, p_1 \neq p_2\}$$

and similarly

$$J_2 = \{(p_1, p_2, H) \in \mathcal{C} \times \mathcal{C} \times \mathbb{P}^2(\mathbb{C})^* \quad | \quad p_1 \in H, p_2 \in H, p_1 \neq p_2\}.$$

With this definition, obviously J_2 is a line bundle over $\mathcal{C} \times \mathcal{C} - \Lambda$ where Λ is the diagonal of $\mathcal{C} \times \mathcal{C}$ (indeed two distinct points define a line). Now Λ is a complex subvariety of $\mathcal{C} \times \mathcal{C}$ of dimension strictly less, and $\mathcal{C} \times \mathcal{C}$ is path connected because $\mathcal{C}$ is path connected. Therefore J_2 is also path connected. As I_2 is obtained from J_2 by subtracting a complex subvariety of dimension strictly less, it is also path connected. This implies that G is 2-transitive. □

A.5 Affine Harris theorem

Theorem 36 ([GW97]). *Let $P \in \mathbb{Q}[X,Y]$ be a an absolutely irreducible polynomial of total degree n. Let $\mathcal{C}$ be the corresponding affine curve in $\mathbb{C}^2$. Then there exists a Zariski open set of affine change of coordinates such that, after such a change of coordinates, the projection on the first coordinate x :*

$$pr_1 : \mathcal{C} \to \mathbb{C}$$

induces on the fiber ${pr_1}^{-1}(0)$ a monodromy map

$$\pi_1(\mathbb{C} - \Delta, 0) \to \mathfrak{S}_n$$

which is surjective.

Proof. The theorem is a corollary of Harris' lemma and of a classical theorem of van Kampen recalled below as theorem 37. With the notations of the previous subsection we identify $\mathbb{C} - \Delta$ with the set of all lines in $\mathbb{C}^2$ parallel to the Oy-axis and transverse to $\mathcal{C}$. Then we include this set in the intersection of U with the line of $\mathbb{P}^2(\mathbb{C})^*$ formed by all the lines passing by the point at infinity corresponding to the Oy-axis. Moreover we suppose that O is not in Δ and choose $H_0 = Oy$. Then to prove the theorem, it suffices to show that the induced group morphism

$$\pi_1(\mathbb{C} - \Delta, 0) \to \pi_1(U, H_0)$$

is surjective.

We view U as the complementary of a reduced projective curve in the dual projective plane $\mathbb{P}^2(\mathbb{C})^*$, which is isomorphic to the usual projective plane $\mathbb{P}^2(\mathbb{C})$. Then our theorem will be a consequence of a classical theorem of E. van Kampen in 1933. A short rigorous and self-contained exposition of this last result was given in a paper by D. Cheniot in 1973. It contains a precise description (by generators and relations) of the fundamentals groups. (We can also find this theorem in [Dim92]). We summarize it as follows.

Theorem 37 (van Kampen). *Let $\mathcal{H}$ be a reduced algebraic curve of degree n in $\mathbb{P}^2(\mathbb{C})$ and A be point outside $\mathcal{H}$. Let L_i for $i = 1$ to m and L_∞ be all the line passing by A and not transverse to $\mathcal{H}$. Call λ_i for $i = 1$ to m and λ_∞ their direction in a $\mathbb{P}^1(\mathbb{C})$ complementary to A. Let L be a line passing by A and transverse to $\mathcal{H}$, call λ its direction. Then*

$$\pi_1(\mathbb{P}^1(\mathbb{C}) - \{\lambda_1, ..., \lambda_m, \lambda_\infty\}, \lambda) \to \pi_1(\mathbb{P}^2(\mathbb{C}) - \mathcal{H}, A)$$

is surjective.

With our notations, we get

$$\pi_1(\mathbb{P}^1(\mathbb{C}) - \{\lambda_1, ..., \lambda_m, \lambda_\infty\}, \lambda) = \pi_1(\mathbb{C} - \Delta, 0)$$

and we are done.

□

Remark :
It would have been more elegant to provide an algebraic proof of our Affine Harris theorem. A natural way for this purpose is to adapt the proof recalled in the last section, by using Jouanolou's version of Bertini's theorem applied to the algebraic set I_2. However this only proves a weaker version of our claim. Indeed, instead of obtaining the monodromy map associated to the lines parallel to a generic direction Oy, we get the monodromy map associated to the lines passing by a generic point of $\mathbb{P}^2(\mathbb{C})$ and we are unable to certify that we can choose such a point on the line at infinity. So we are led to rely on a topological analysis, which in this situation gives more precise informations.

Composite Monodromy

To validate Galligo-Rupprecht's algorithm, a result for the composite case is needed.

When P has several factors, then $\mathcal{C}$ has several irreducible components $\mathcal{C}_1 \cdots \mathcal{C}_s$, and each of them has a monodromy action. So we can relate the monodromy of $\mathcal{C}$ to the monodromies of the $\mathcal{C}_i$. The result claimed in [Gal99] and in [Rup04], says that after a generic change of coordinates, the following group homomorphism is surjective

$$\pi_1(\mathcal{C} - \Delta) \to \mathfrak{S}_{n_1} \times \mathfrak{S}_{n_2} \times ... \times \mathfrak{S}_{n_s}.$$

This result is a straightforward corollary of the proof of theorem 36 that we explained above.

Annexe B

Étude de l'algorithme *Fac-Knap* sur un exemple

Dans cette annexe nous allons décrire l'algorithme *Fac-Knap* sur un polynôme irréductible de $\mathbb{Q}[X, Y]$ de degré $n = 30$ possédant 5 facteurs absolument irréductibles. Ce polynôme est le polynôme $n30s5$ se trouvant sur la page http ://math.unice.fr/∼cheze/. Les calculs ont été effectués sur un ordinateur portable (mobile AMD Athlon XP 2000+ (1656 Mhz), 200 Mo).

Les paramètres utilisés sont les suivants : $\lambda = 0$, $x_0 = 0$. Autrement dit, nous supposons que nous sommes dans une situation générique. Si ce n'est pas le cas, alors nous effectuerons effectivement un changement de variables.

Tout d'abord, nous devons déterminer l le nombre de b_i réels, et, la précision utilisée dans les calculs. Pour cela nous estimons dans un premier temps, grâce au théorème 25, l de la façon suivante : $l \approx \sqrt{30}$. Puis nous calculons C grâce à la formule donnée page 85. C'est à dire que l'on pose :

$$C = \frac{\sqrt{2}}{\sqrt{n+1}} \sqrt{(\frac{n+l}{2} + [\frac{5}{4}(n+l)]^2)^{\frac{n+l}{2}} - 1}.$$

Dans notre cas nous obtenons : $C := 10^{26}$. Nous posons alors, comme expliqué à la page 97, $\eta := 10^{-46}$.

Le temps en secondes pour obtenir les 30 valeurs des b_i est de 0.531 s.

Nous constatons que sur cet exemple nous avons quatre b_i réels. L'estimation donnée par le théorème 25 est donc vérifiée, puisque $4 \leq \sqrt{30} \approx 5,47$. Nous ordonnons alors les b_i de telle sorte à avoir $b_1, \dots, b_4 \in \mathbb{R}$.

Nous commençons donc la construction de notre suite de réseau avec $\mathcal{L}_0 = \mathbb{Z}^{17}$, car ici $\dfrac{n+l}{2} = 17$. Nous en déduisons alors le réseau $\mathcal{L}'$ à l'aide des parties réelles des b_i. De plus, dans cet exemple la constante M du lemme 22 de la page 79 est $M = \sqrt{667,25}$. Ainsi pour une base orthogonale de $\mathcal{L}'$ nous pourrons supprimer tous les vecteurs consécutifs ayant une norme supérieure à $\sqrt{667}$.

Nous calculons donc une base LLL réduite de $\mathcal{L}'$, fabriquons la base orthogonalisée correspondante $\{\nu_1^*, \ldots, \nu_{17}^*\}$ et supprimons tous les vecteurs consécutifs de norme supérieure à $\sqrt{667}$. Il nous reste alors 4 vecteurs. Cela nous permet de définir le réseau $\mathcal{L}_1$. La forme échelonnée réduite de $\mathcal{L}_1$ est alors :

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Comme nous avons $b_1, \ldots, b_4 \in \mathbb{R}$, nous remarquons alors que les deux premières lignes vont donner chacune un facteur réel. Par exemple la première ligne va donner le facteur correspondant à $\{b_1, b_4, b_6, \overline{b}_6, b_{10}, \overline{b}_{10}\}$. Les facteurs recherchés sont donc de degré 6. Le facteur réel correspondant à la dernière ligne de le matrice doit donc être factorisé en deux facteurs complexes, d'après l'étude effectuée en 3.2.3 page 84. A la fin de cette étape nous avons les facteurs approchés modulo X^3.

La durée de cette étape (calcul des facteurs approchés à partir des $\tilde{y}_i$, $\tilde{a}_i$, $\tilde{b}_i$) a été de 0,12 secondes.

A l'aide des facteurs approchés, nous pouvons obtenir le polynôme minimal de l'extension $\mathbb{Q}[\alpha]$. Dans un premier temps nous obtenons une approximation de ce polynôme (voir chapitre 2) :

$$\begin{aligned} f_{\alpha+\epsilon}(T) &= T^5 + (-154949 + 1,48368246015161275800000.10^{-67} \times i)T^4 \\ &\quad +(566512,0000 - 2,91703841090202331500000.10^{-62} \times i)T^3 \\ &\quad +(7584054,9999999\ldots - 1,55575381917878985400000.10^{-61} \times i)T^2 \\ &\quad +(20850849,9999999\ldots - 6,223015276715159416000.10^{-61} \times i)T \\ &\quad +17815979,9999999\ldots + 6,223015276715159416000.10^{-61} \times i \end{aligned}$$

Nous en déduisons alors :

$$f_\alpha(T) = T^5 - 154949T^4 + 566512T^3 + 75840055T^2 + 20850850T + 17815980$$

Nous simplifions alors l'extension à l'aide de la commande *OptimizedRepresentation* de Magma, et nous obtenons le polynôme :

$$g(t) = T^5 - 48T^3 + 200T^2 - 305T + 155.$$

Nous reconnaissons alors, à l'aide de la méthode présentée au chapitre 2, les facteurs exacts modulo X^3 dans l'extension $\dfrac{\mathbb{Q}[T]}{(g(T))}$. Pour finir nous effectuons la remontée de Hensel afin d'obtenir un facteur exact dans $\dfrac{\mathbb{Q}[T]}{(g(T))}$.
La remontée de Hensel a pris 0.11 secondes.

Nous obtenons alors un facteur absolument irréductible à coefficient dans $\dfrac{\mathbb{Q}[T]}{(g(T))}$.
La durée totale de l'algorithme aura été de 0.95 secondes.

Annexe C

Lifting and Recombination Techniques for absolute factorization

Cette annexe est une version préliminaire de [CL]. *Cette annexe permet donc de compléter le chapitre* 4.

Introduction

In this text, F denotes the polynomial we want to factor : this is a polynomial in two variables x and y over a commutative field $\mathbb{K}$; its total degree is denoted by d and is assumed to be positive. Under

$$\text{Hypothesis (C)} \quad \mathbb{K} \text{ has characteristic } 0 \text{ or at least } d(d-1)+1,$$

we present new faster probabilistic and deterministic algorithms for computing the *absolute factorization* of F, that is the irreducible factorization over the algebraic closure $\bar{\mathbb{K}}$ of $\mathbb{K}$. In order to avoid confusion we say *rational factorization* for factorization in $\mathbb{K}[x, y]$.

Our new algorithms combine advantages of Gao's algorithm [Gao03] and the rational factorization algorithms introduced in [Lec04b]. In particular, we propose an efficient adaptation of the classical *(Hensel) lifting and recombination* factorization scheme to absolute factorization.

Absolute factorization is a classical problem in computer algebra and has several applications in various fields. In the beginning of the eighties, the first polynomial time algorithms originated in the field of effective algebraic geometry and were motivated by the problem of computing irreducible decompositions of solution sets of polynomial equation systems [GC84]. Such computations still remain challenging problems from both theoretical and practical points of view. Yet current algorithms already allow one to deal with real world applications. For instance, when $\mathbb{K} = \mathbb{Q}$, absolute factorization allows one to decompose the smooth locus of an equidimensional algebraic set into path-connected components : this is an important ingredient for solving problems arising from kinematics, as exemplified in [SVW02a]. In the eighties, symbolic integration was also a main motivation in designing absolute factorization algorithms [Tra85]. Nowadays, absolute factorization is used in solving linear differential equations [SU97, HRUW99, Bro01].

After the presentation of our new complexity results, we describe the main steps of our algorithms and then we conclude this introduction with discussing related work. Before all we start with some prerequisites.

Notation

The field $\mathbb{K}$ is always considered to be a subfield of $\bar{\mathbb{K}}$. The algebra of polynomials in two variables over $\mathbb{K}$ is denoted by $\mathbb{K}[x,y]$ and $\mathbb{K}[x,y]_m$ represents the vector space of polynomials of total degree at most m. The field of fractions of $\mathbb{K}[y]$ is denoted $\mathbb{K}(y)$, the power series algebra over $\mathbb{K}$ is denoted $\mathbb{K}[[x]]$ and its field of fractions $\mathbb{K}((x))$. For any polynomial $G \in \mathbb{K}[x,y]$, $\deg(G)$ represents the total degree of G and its degree with respect to the variable x (resp. y) is written $\deg_x(G)$ (resp. $\deg_y(G)$).

When defined, the greatest common divisor of u and v is denoted $\gcd(u,v)$. The remainder of u divided by v is denoted $u \bmod v$. The resultant of f and g in $\mathbb{K}[y]$ is written $\mathrm{Res}(f,g)$. By $\mathrm{Res}_y(F,G)$ of two polynomials F and G in $\mathbb{K}[x,y]$ we mean the resultant of F and G seen as polynomials in $\mathbb{K}[x][y]$.

We use Gantmacher's notation $\ell_{1:d}$ to denote the d-tuple $(\ell_1,\ldots,\ell_d)$. We also use the notation $\langle \mu_1,\ldots,\mu_r\rangle = \langle \mu_{1:r}\rangle$ to represent the $\mathbb{K}$-vector space generated by $\mu_{1:r}$.

For a real number a, $\lceil a \rceil$ (resp. $\lfloor a \rfloor$) denotes the smallest integer larger than or equal to (resp. the greatest integer less than or equal to) a.

In the pseudo-code, we use the function coeff in various contexts. For any ring R, if $G \in R[[x,y]]$ then $\mathrm{coeff}(G, x^i y^j)$ represents the coefficient of the monomial $x^i y^j$ of G. For a univariate polynomial $f \in \mathbb{K}[y]$ of degree d, if $\mathbb{A}$ represents the $\mathbb{K}$-algebra $\mathbb{K}[y]/(f(y))$ and if φ denotes the canonical image of y in $\mathbb{A}$ then any $b \in \mathbb{A}$ can uniquely be written $b = b_0 + b_1\varphi + \cdots + b_{d-1}\varphi^{d-1}$. In this case, we define $\mathrm{coeff}(b,\varphi^i) := b_i$. For readability, we write $\mathrm{coeff}(G, \varphi^i x^j y^k)$ instead of $\mathrm{coeff}(\mathrm{coeff}(G, x^j y^k), \varphi^i)$ for any $G \in \mathbb{A}[[x,y]]$.

Complexity Model

For our complexity analysis, we use the *computation tree* model [BCS97, Chapter 4] with the *total complexity* point of view. Roughly speaking, this means that complexity estimates charge a constant cost for each arithmetic operation ($+$, $-$, $\times$, $\div$) and the equality test. Yet all the constants in the base fields (or rings) of the trees are thought to be freely at our disposal.

The way how trees are constructed is often described by means of *pseudo-code*. We hope this presentation is convenient for those who wish to implement our algorithms or simply translate them into another complexity model.

We use the classical $\mathcal{O}$ and $\tilde{\mathcal{O}}$ ("soft Oh") notation in the neighborhood of infinity as defined in [vzGG03, Chapter 25.7]. Informally speaking, "soft Oh"s are used for readability in order to remove logarithmic factors in complexity estimates.

Univariate polynomials are thought to be represented by dense vectors of their coefficients in the canonical basis. For each integer n, we assume we are given a computation tree that computes the products of two polynomials of degree at most n with at most $\mathsf{M}(n)$ operations, independently of the base ring. As in [vzGG03, Chapter 8.3], for any positive integers m and n, we assume that M satisfies : $M(mn) \le m^2\mathsf{M}(n)$ and $\mathsf{M}(n)/n \ge \mathsf{M}(m)/m$ if $n \ge m$. In particular, this implies the *super-additivity* of M, that is : $\mathsf{M}(n_1+n_2) \ge \mathsf{M}(n_1)+\mathsf{M}(n_2)$ for any positive integers n_1 and n_2. This setting allows us to design algorithms that are independent of the choice of the subroutine chosen for polynomial multiplication (naive, Karatsuba or fast Fourier transform, for instances). Recall that using the fast Fourier transform allows us to take $\mathsf{M}(n) \in \mathcal{O}(n\log(n)\log\log(n)) \subseteq \tilde{\mathcal{O}}(n)$ [vzGG03, Chapter 8.3].

We recall that the resultant and the extended greatest common divisor of two univariate polynomials over $\mathbb{K}$ can be computed within $\mathcal{O}(\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$ [vzGG03, Chapter 11]. In particular, if $\mathbb{F} := \mathbb{K}[z]/(q(z))$ is an algebraic extension of $\mathbb{K}$ of degree n then each field operation in $\mathbb{E}$ can be done within $\mathcal{O}(\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.

Lastly, we use the constant ω to denote a *feasible matrix multiplication* exponent as defined in [vzGG03, Chapter 12] : two $n \times n$ matrices over $\mathbb{K}$ can be multiplied within $\mathcal{O}(n^\omega)$ field operations ($2 \le \omega \le 3$). In contrast to polynomials, we shall only deal with matrices over $\mathbb{K}$. In practice, we must keep in mind that ω is often close to 3.

Main Complexity Results

The absolute irreducible factors of F will be denoted by $F_1, \ldots, F_r$. By definition, the F_i all belong to $\bar{\mathbb{K}}[x,y]$ but, from a computational point of view, a more precise description of the algebraic extensions of $\mathbb{K}$ containing the F_i is necessary. Although it is classical, if F is irreducible then we will show in Section C.4.4 that there exist :

- A separable algebraic extension $\mathbb{F}$ of $\mathbb{K}$ of degree r ;
- A polynomial $\mathsf{F} \in \mathbb{F}[x,y]$ of total degree d/r such that :

$$\{F_1, \ldots, F_r\} = \{\chi(\mathsf{F}) \mid \chi : \mathbb{F} \hookrightarrow \bar{\mathbb{K}}\},$$

where χ runs over all the embeddings of $\mathbb{F}$ in $\bar{\mathbb{K}}$ that induce the identity on $\mathbb{K}$. For convenience, we say that F is the *generic factor* of F. In practice, the extension $\mathbb{F}$ will be represented by a quotient $\mathbb{F} := \mathbb{K}[z]/(q(z))$, where q is a monic separable irreducible polynomial in $\mathbb{K}[z]$ of degree r. Of course the representation of F is not unique since it depends of the choice of the primitive element α of $\mathbb{F}$ over $\mathbb{K}$. If F

is not irreducible then our general algorithm of Section C.4 computes the generic factor of each rational irreducible factor of F.

For efficiency reasons, we present optimized algorithms in the case when F is irreducible : because, so far, the rational factorization algorithms of [Lec04b] remain a bit faster than our absolute factorization algorithms. The complexity analysis of the general case is work in progress in [CL04]. We now present our main complexity results in terms of the *computation tree* model.

Theorem 1. *Under Hypothesis (C), if F is irreducible then its generic absolute factor can be computed within $\tilde{\mathcal{O}}(d^4)$ arithmetic operations in $\mathbb{K}$.*

Although we will not use any probabilistic complexity model, we will informally use the term *probabilistic algorithm* when speaking about the computation trees occurring in the next theorem. For the sake of precision, we preferred to express the probabilistic aspects in terms of the existence of families of computation trees so that, if the cardinality of $\mathbb{K}$ is infinite, almost all the trees are executable and compute suitable results. For convenience, we introduce $\mathcal{U}(P) := \{a \in \mathbb{K}^n \mid P(a) \neq 0\}$, for any polynomial $P \in \bar{\mathbb{K}}[x_{1:n}]$.

Theorem 2. *There exists a family of computation trees over $\mathbb{K}$ parametrized by*

$$(u, v, a, c_1, \ldots, c_d) \in \mathbb{K}^{d+3}$$

such that : for any irreducible polynomial $F \in \mathbb{K}[x, y]$ such that Hypothesis (C) holds, any executable tree of the family computes the generic factor F. The maximum of the costs of the trees of the family belongs to $\tilde{\mathcal{O}}(d^{(\omega+3)/2})$.

In addition, there exists a nonzero polynomial $P \in \mathbb{K}[U]$ of degree at most d such that, for any $u \in \mathcal{U}(P)$, there exists a nonzero polynomial $Q_u \in \mathbb{K}[V]$ of degree at most $d(d-1)$ such that, for any $v \in \mathcal{U}(Q_u)$, there exists a nonzero polynomial $R_{u,v} \in \mathbb{K}[A]$ of degree at most $d(d-1)$ such that, for any $a \in \mathcal{U}(R_{u,v})$, there exists a nonzero polynomial $S_{u,v,a} \in \bar{\mathbb{K}}[C_{1:d}]$ of total degree at most $d(d-1)/2$ such that, for any $c_{1:d} \in \mathcal{U}(S_{u,v,a})$, the tree corresponding to $(u, v, a, c_1, \ldots, c_d)$ is executable on F.

At first sight, the complexity estimates of these two theorems only make sense in characteristic 0 : for a fixed field $\mathbb{K}$ of positive characteristic, Hypothesis (C) implies that the admissible values for the degree d are bounded, so that the complexity can be bounded by a constant. However, it is important to underline that the constants hidden behind the $\mathcal{O}$ of the complexity estimates of the two theorems only depend on the characteristic of the base field $\mathbb{K}$. This comes from the fact that ω depends on the characteristic of $\mathbb{K}$. In consequence, this dependency disappears if we take $\omega := 3$.

In addition, it is important to notice that Hypothesis (C) and the degree bounds given in the second half of Theorem 2 guarantee the existence of at least one tree of the family that is executable on a given F. This immediately yields the existence of a *deterministic* tree but of higher complexity than in Theorem 1.

The probabilistic strategy is all the more efficient as the cardinality of the base field becomes large compared to d^2. In this case, one can use the classical Zippel-Schwartz zero test [Zip93, Sch80] to estimate the probability for picking up an executable tree at random in a given finite subset of values of the parameters.

Overview of the Algorithms

The different versions of our algorithm all share the same main ideas and divide into the following main stages. At the beginning, the coordinates are changed to sufficiently generic ones. This is detailed in the first section. Then we let $f(y) := F(0, y)$, $\mathbb{A} := \mathbb{K}[y]/(f(y))$ and lift the *generic* solution ϕ of $F(x, \phi) = 0$ in $\mathbb{A}[[x]]$. The lifting algorithm is presented in Section C.2. From the approximation of ϕ up to a precision that is linear in d, we then build a linear system whose basis of solution has rank r (the number of absolute irreducible factors) and contains all the information allowing to deduce the irreducible factors. This construction and the resolution of the system are studied in Section C.3. Then, it remains to construct the generic factors, this is the purpose of Section C.4. Lastly, the coordinates are changed back.

The general algorithm is completed in Section C.4. In the next sections, we assume that F is irreducible and propose optimized algorithms that lead to Theorems 1 and 2. Before entering into details, we briefly specify each of the subroutines so that the skeleton of the factorization algorithms becomes clear.

Change of Coordinates

Our algorithms start with changing the original coordinates in order to ensure the following Hypothesis (H) :

$$\text{Hypothesis (H)} \quad \begin{cases} (i) \ \deg_y(F) = \deg(F) = d; \\ (ii) \ \mathrm{Res}_y\left(F, \frac{\partial F}{\partial y}\right)(0) \neq 0. \end{cases}$$

We remark that (i) implies the monicity of F with respect to y. Without loss of generality, we also assume that the coefficient of y^d is exactly 1.

Lifting Stage

The lifting stage consists in computing an approximation of the generic power series solution ϕ of $F(x, .) = 0$. We denote by φ the canonical projection of y in $\mathbb{A}$. One can consider φ as formally representing any root of $f(y)$ and under Hypothesis (H), there exists a unique $\phi \in \mathbb{A}[[x]]$ such that $\phi - \varphi \in (x)$ and $F(x, \phi) = 0$. It is classical that ϕ can be approximated at any precision (x^σ), by means of Newton's operator.

Recombination Stage

From ϕ computed up to precision (x^σ), we construct a linear system from the coefficients of $\hat{\mathfrak{F}} := F/\mathfrak{F} \in \mathbb{A}[[x]][y]$, where $\mathfrak{F} := y - \phi$. The following definition constitutes the cornerstone of our method :

$$L_\sigma := \Big\{ (\ell_{1:d}, G, H) \in \mathbb{K}^d \times \mathbb{K}[x,y]_{d-1} \times \mathbb{K}[x,y]_{d-1} \mid$$
$$G - \sum_{i=1}^{d} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial y}, \varphi^{i-1}\right) \in (x,y)^\sigma, \tag{C.1}$$
$$H - \sum_{i=1}^{d} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial x}, \varphi^{i-1}\right) \in (x,y)^\sigma + (x^{\sigma-1}) \Big\}, \tag{C.2}$$

where the notation $\operatorname{coeff}(P, \varphi^i)$ abusively represents $P_i \in \mathbb{K}[[x]][y]$ uniquely defined by $P = P_0 + P_1\varphi + \cdots + P_{d-1}\varphi^{d-1}$.

Although the form of this construction does not look intuitive at first sight, we will see that it is directly related to the recombination technique introduced in [Lec04b]. In addition, the notation is designed to be consistent with [Lec04b] but observe that $\frac{\partial \mathfrak{F}}{\partial y} = 1$ and $\frac{\partial \mathfrak{F}}{\partial x} = -\phi'$.

For any $i \in \{1, \ldots, r\}$, we introduce the partial product $\hat{F}_i := \prod_{j=1, j\neq i}^{r} F_j$ and $\operatorname{Tr}_j f_i := \sum_{f_i(\psi)=0} \psi^j$ that denotes the sum of the jth powers of the roots of $f_i := F_i(0,y)$. Lastly, π (resp. π_G) denotes the canonical projection that maps $(\ell_{1:d}, G, H)$ to $\ell_{1:d}$ (resp. G). One of our main results concerns the properties of L_σ : for sufficiently large precisions σ, the vector space L_σ contains all the information about the absolute factorization of F. Since $\bar{\mathbb{K}}$ is faithfully flat over $\mathbb{K}$, we naturally identify the extension $\bar{\mathbb{K}} \otimes L_\sigma$ with the space of solutions $(\ell_{1:d}, G, H) \in \bar{\mathbb{K}}^d \times \bar{\mathbb{K}}[x,y]_{d-1} \times \bar{\mathbb{K}}[x,y]_{d-1}$ of Equations (C.1) and (C.2).

Theorem 3. *Under Hypotheses (C) and (H), for any $\sigma \geq 2d$, we have :*

$$\bar{\mathbb{K}} \otimes L_\sigma = \left\langle \left(\mu_i, \hat{F}_i \frac{\partial F_i}{\partial y}, \hat{F}_i \frac{\partial F_i}{\partial x}\right) \mid i \in \{1, \ldots, r\} \right\rangle,$$

where $\mu_i := (\operatorname{Tr}_0 f_i, \ldots, \operatorname{Tr}_{d-1} f_i)$.

For the sake of convenience, we introduce L_∞ defined by

$$L_\infty := L_\sigma, \quad \text{for } \sigma \geq 2d.$$

Construction of the Generic Factor

Based on the previous theorem, we will show how to compute a basis $G_{1:r}$ of $\pi_G(L_\infty)$. For efficiency reasons, we do almost all the computations with $x = 0$. This is why we introduce $g_{1:r}(y) := G_{1:r}(0,y)$. We will show that, for all $c_{1:r}$ in a certain Zariski open subset of $\mathbb{K}^r$, the monic squarefree part q of

$$\mathrm{Res}_y\left(f, zf' - \sum_{i=1}^{r} c_i g_i\right) \tag{C.3}$$

has degree r. In this case, we can decompose q into its irreducible factors $q_1, \dots, q_m$. Then, for each $i \in \{1, \dots, m\}$, we introduce $\mathbb{F}_i := \mathbb{K}[z]/(q_i(z))$ and we let α_i be the canonical image of z in $\mathbb{F}_i$. We will show that there corresponds a generic factor F_i of F to each i, that is given by the following formula :

$$\mathsf{F}_i := \gcd\left(F, \alpha_i \frac{\partial F}{\partial y} - \sum_{i=1}^{r} c_i G_i\right). \tag{C.4}$$

We shall show in Section C.4 that $c_{1:r}$ can be taken at random with a high probability of success but we will also describe a deterministic way for computing a suitable candidate in Section C.6.

Example. *Before going further, let us illustrate the computation of the absolute factorization of a small example : let* $\mathbb{K} := \mathbb{Q}$ *(the field of the rational numbers) and*

$$F := y^4 + 2xy^2 + 14y^2 - 7x^2 + 6x + 47.$$

Hypothesis (H) is satisfied, one has $f := y^4 + 14y^2 + 47$ *and, with* $\sigma := 2\deg(F) = 8$, *we obtain :*

$$\begin{aligned}\phi = \varphi &+ (-13/94\varphi^3 - 44/47\varphi)x + (39/8836\varphi^3 + 199/17672\varphi)x^2 \\ &+ (-4745/1661168\varphi^3 - 15073/830584\varphi)x^3 \\ &+ (67665/156149792\varphi^3 + 1231735/624599168\varphi)x^4 \\ &+ (-13201279/58712321792\varphi^3 - 19943203/14678080448\varphi)x^5 \\ &+ (305810505/5518958248448\varphi^3 + 3137922039/11037916496896\varphi)x^6 \\ &+ (-26241896109/1037564150708224\varphi^3 - 76656876747/518782075354112\varphi)x^7 \\ &+ O(x^8)\end{aligned}$$

A possible basis of $\pi(L_\sigma)$ *is* $(1,0,0,0)$, $(0,0,1,0)$. *Then, taking* $c = (0,1)$, *we obtain* $q(z) = z^2 - 1/2z - 47/32$. *Since* q *is irreducible, we deduce that* F *is irreducible and that its generic factor is* $\mathsf{F} := y^2 + (-16/7\alpha + 11/7)x - 8/7\alpha + 51/7$, *where* α *denotes the image of* z *in* $\mathbb{F} := \mathbb{K}[z]/(q(z))$.

Related Work

First, we relate the main contributions in the field of effective absolute factorization. Detailed comparisons are discussed next.

Although polynomial factorization has been widely studied in computer algebra, it still remains a central topic, for no softly asymptotically optimal algorithm is known. Some classical books [Zip93, vzGG03] largely cover this subject and we refer to them for classical results. For histories and bibliographies we refer to [Kal90, Kal92, Kal95, Kal03, Gao03, CG05].

First, it is worth recalling that the problem of factoring multivariate polynomials reduces to factoring bivariate polynomials : this probabilistic reduction originates in computer algebra in [HS81] and uses a quantitative version of Bertini's irreducibility theorem [Zip93, Chapter 19]. We refer to [Rup86, Kal95, Gao03, Lec04a] for more details in this direction, which is closely related to the problem of bounding the degrees of the *Noether irreducibility forms* [Sch00, Chapter V].

In the following paragraphs we present a short survey of all known absolute factorization algorithms.

Lattice Reduction

Inspired by [LLL82], many authors developed factorization algorithms on the basis of *lattice reduction techniques* and *algebraic approximant computations* including [GC84, Chi84, Kal85b, Kal95]. For example, Kaltofen's algorithm computes the minimal polynomial of a power series expansion of the curve $F(x, y) = 0$ at a smooth point.

Trager's Reduction to Factoring in Algebraic Extensions

Trager [Tra85] suggested that absolute factorization could be reduced to rational factorization over a suitable algebraic extension. This extension is built as the minimal algebraic extension $\mathbb{E}$ containing a smooth point (α, β) on the curve defined by $F(x, y) = 0$. Then, Trager's algorithm [Tra76] can be invoked to compute the factorization of F in $\mathbb{E}[x, y]$. This reduces the original problem to rational factorization, with an overhead that increases with the degree of $\mathbb{E}$. This idea has been developed in [Tra86, DT89]. Independently, considering such an extension is the basis of the absolute irreducibility test proposed in [Kal85b].

Duval's and Ragot's Algorithms

As described in [Duv91], Duval's algorithm computes a basis of a suitable vector subspace D of divisors of the curve defined by $F(x, y) = 0$. From this basis and a smooth point $(\alpha, \beta) \in \mathbb{E}^2$ on this curve, one can compute a set of polynomials whose common gcd with F produces an absolute factor of F. This method is improved in Ragot's thesis [Rag97], in order to compute generic factors faster. In [CSTU02], this algorithm is further improved : the expensive computation of a basis of D is replaced by the resolution of a system of linear differential equations.

Topology and Monodromy

In the nineties, based on topological concepts, new families of algorithms have been designed for the special case $\mathbb{K} = \mathbb{Q}$. The use of the connectedness property of the irreducible components of the curve $F(x, y) = 0$ outside the singular locus is explored in [BCGW93], in order to compute numerical approximations of the factors.

Recently, the idea of using the monodromy theory appeared in the works of Galligo and his collaborators [GR02, CGKW02, Rup04] : their algorithms perform mixed symbolic and numerical computations but the final result is always exact. In [Chè04b], these algorithms are improved thanks to using lattice reductions. On the other hand, in [SVW02c, SVW02a], this idea is turned into a purely numerical algorithm well suited to homotopy continuation. Lastly, it is worth mentioning that the computation an exact factorization from a sufficiently accurate numerical approximation is always possible, as explained in [CG03].

Differential Operators

In [CSTU02], the links between absolute factorization and linear differential operators are investigated and it is shown that the factorization can be computed from the *minimal differential operator* associated to F. Improvements of these techniques are presented in [BT03].

Closed Differential Forms

In [Rup86, Rup99], Ruppert proposes a very efficient absolute irreducibility test based on arithmetic alone (often refereed to as *Ruppert's theorem*). Roughly speaking, the test formulates in terms of the existence of a closed differential 1-form having F as denominator and a bounded polynomial numerator. This test yields sharp bounds on the degrees of the Noether irreducibility forms and on the heights of the *Ostrowski integers*, we refer to Schinzel's book [Sch00, Chapter V] for a detailed exposition.

Ruppert's irreducibility test was turned into a probabilistic factorization algorithm by Gao [Gao03] : the absolute factorization reduces to the computation of a vector space of closed differential 1-forms. It is worth mentioning that his algorithm computes both rational and absolute factorizations at the same time within $\tilde{\mathcal{O}}(d^5)$ operations in $\mathbb{K}$. At the end of [Gao03], there are briefly exposed clues that one allow to reach $\tilde{\mathcal{O}}(d^4)$. From these techniques, an approximate factorization algorithm is derived in [GKM$^+$04].

Duval's, Ragot's, Gao's and our algorithms have the following point in common to Berlekamp's and Niederreiter's algorithms [Nie93, GG94, MŞ99] for factoring univariate polynomials over finite fields : one first compute a basis of a certain vector space whose dimension equals the number of factors, then these factors are obtained by means of gcd.

Irreducibility Tests

Lastly and less connected to our present concerns, we mention the following irreducibility tests : [HS81, Kal85b, Gao01, GL01, GR03, GL, Rag02, CL04].

Contributions

Generally speaking, lifting and recombination schemes divide into three main stages. Assuming that the coordinates are sufficiently generic (Hypothesis (H) often suffices), in the first stage the first variable is specialized to a random value (0 in our case) and the univariate polynomial f obtained this way is factored. In the second stage, this factorization is *lifted* over a power series algebra and the last stage is devoted to discover how the true factors *recombine* from the lifted factors.

These schemes were designed to perform rational factorization only. Here, we present new techniques for adapting them to absolute factorization. In particular, we adapt [Lec04b], which is currently known as the fastest rational factorization algorithm. This way, we obtain new upper complexity bounds for the absolute factorization problem. Yet it is worth underlining that this adaptation is not as *direct* as expected : we had to modify the lifting device and the *reduced echelon form* computation had to be replaced by a more complicated procedure.

In the next paragraphs, we detail how our ideas compare to the most related algorithms mentioned above.

Comparison to Trager's Reduction

Lifting and recombination schemes can be directly applied to the absolute factorization problem as soon as a factorization of f over $\bar{\mathbb{K}}$ is known. Several algorithms exist for computing splitting fields ([ORV04] for instance) but, unfortunately, their complexities overwhelm the one of absolute factorization. Similarly, *dynamic evaluation* techniques [Duv95, Del01, Ste02] are also very expensive. As an exception, when $\mathbb{K} = \mathbb{Q}$ (the field of rational numbers), one can numerically perform the computations over $\mathbb{C}$ as in [SS93].

A yet more efficient technique is possible for any field, following Trager's idea. In case of the factorization algorithm presented in [Lec04b], assuming that F is monic with respect to y and that $F(0, y)$ is squarefree, this suggests the following method : take $\alpha := 0$, compute an irreducible factor $h(y)$ of $F(0, y)$ in $\mathbb{K}[y]$, let $\mathbb{E} := \mathbb{K}[y]/(h(y))$ and let β be the canonical image of y in $\mathbb{E}$, whence $F(0, \beta) = 0$. Using the algorithms of [Lec04b] then allows one to compute the absolute irreducible factor of F that vanishes at (α, β).

Let us start with discussing the worst case complexity. Neglecting the costs due to factoring $F(0, y)$ over $\mathbb{K}$ and over $\mathbb{E}$, the complexity of this process is at most the complexity of factoring in $\mathbb{E}[x, y]$. We deduce that the overhead of the absolute factorization with respect to the rational factorization is at most the degree of h, which can equal d in the worst case. Roughly speaking, since the probabilistic algorithm of [Lec04b] has a complexity within $\mathcal{O}(d^\omega)$ this strategy has a worst case complexity upper bound at least in $\mathcal{O}(d^{\omega+1})$ in terms of operations in $\mathbb{K}$, which is greater than the bound presented in Theorem 2.

On the other hand, in practice, if one finds a factor h of very small degree this strategy may become interesting. Of course, deeper comparisons quickly become tedious and we leave out this discussion for further work. In the spirit of [FGP01, GL02], average complexity analysis should be of great interest.

Anyway, one of our contributions is an alternative to factoring over an algebraic extension $\mathbb{E}$: we avoid both factorizations of $F(0, y)$ in $K[y]$ and in $\mathbb{E}[y]$. Another contribution is that we directly compute the minimal algebraic extension $\mathbb{F}$ that contains the coefficients of one absolute factor. In addition, the complexities of our algorithms are very closed to the ones given in [Lec04b] for rational factorization : the bound using d^4 (resp. $d^{(\omega+3)/2}$) of Theorem 1 (resp. 2) is to be compared to $d^{\omega+1}$ (resp. d^ω) of [Lec04b, Proposition 1 (resp. 2)]. In both cases the overhead is much less than a factor of d.

Comparison to Duval's Algorithm

In Duval's algorithm, as described in [Duv91], one first computes a $\mathbb{K}$-basis $D_{1:r}$ of $\mathbb{F}$, seen as the algebraic closure of $\mathbb{K}$ in $\mathbb{K}(x)[y]/(F(x, y))$. If I_i denotes the inverse of $\hat{F}_i$ modulo F_i then $(\hat{F}_i I_i)_{i\in\{1,\dots,r\}}$ is a $\bar{\mathbb{K}}$-basis of $\langle D_{1:r}\rangle$. Then, from a smooth point (α, β) (defined as above) on the curve $F(x, y) = 0$ and by means of elementary linear algebra operations, one deduces a $\mathbb{K}$-basis $\tilde{D}_{1,r-1}$ of the elements of $\langle D_{1:r}\rangle$ that vanish at (α, β). Lastly, the common gcd of $\tilde{D}_{1,r-1}$ with F equals an absolute irreducible factor of F. As in Trager's method, a post-treatment is necessary to deduce the generic factor, according to our definition. Dynamic evaluation helps avoiding factorization (if F is irreducible).

It turns out that $(\hat{F}_i \frac{\partial F_i}{\partial y})_{i\in\{1,\dots,r\}}$ is a $\bar{\mathbb{K}}$-basis of $\langle G_{1:r}\rangle$. In consequence $G_{1:r}$ and $D_{1:r}$ have very similar properties : in particular one can adapt Duval's method to recover one irreducible factor from $G_{1:r}$ instead of $D_{1:r}$.

The bottleneck of this algorithm is the computation of the ring of integers of $\mathbb{K}(x)[y]/(F(x, y))$ over $\mathbb{K}[x]$, from which $D_{1:r}$ is deduced.

Comparison to Ragot's Algorithm

Ragot's algorithm [Rag97] starts as Duval's algorithm with computing a basis $D_{1:r}$. But Ragot provides a direct and faster method for computing the generic factor when F is irreducible. By means of a resultant computation very similar to (C.3), he shows how to recover a primitive element representation of $\mathbb{F}$ over $\mathbb{K}$. Then he deduces the generic factor thanks to a gcd similar to (C.4).

Indeed, our Formulae (C.3) and (C.4) follow from arguments that are very closed to the ones invoked in [Rag97, Section 2.3] and which originates from Rothstein and Trager (see [vzGG03, Theorem 22.8]). For the resultant (C.3), we use the same optimization as Ragot that is based on specializing x to 0 but for the gcd (C.4), we find that it is better to keep this variable specialized and make use of Hensel lifting.

Ragot's method is probabilistic and requires F to be irreducible and $\mathbb{K}$ perfect. To our best knowledge, no (efficient) deterministic version has been proposed : we expect that our deterministic computation of a primitive element of $\mathbb{F}$ presented in Section C.6 could be adapted there.

Comparison to Gao's Algorithm

As in [Lec04b], our absolute factorization algorithm is very closed to Gao's algorithm [Gao03] and computes the same output. The conditions on the characteristic of $\mathbb{K}$ are mostly the same : we consider total degrees whereas Gao treats bi-degrees, that are degrees in x and y separately, which induces slight changes in some bounds.

We compute the same basis $G_{1:r}$ but our main gain in complexity comes from the fact that we solve almost the same linear problem but faster : our number of unknowns is only d instead of $\mathcal{O}(d^2)$. As a slight drawback, we have to ensure Hypothesis (H), but the change of variables handling this problem is negligible in practice.

Formula (C.3) yields a faster algorithm than the one related to [Gao03, Theorem 2.8], as pointed out in [Gao03, Section 4, Additional Remarks]. On the other hand, Formula (C.4) is already used in Gao's algorithm. However, Gao provides no deterministic way for finding primitive elements of the field extensions.

C.1 Change of Coordinates

In this section, we show that, under Hypothesis (C), Hypothesis (H) is not restrictive. Although the results presented here are classical, we recall them briefly for completeness.

For any squarefree polynomial $G \in \mathbb{K}[x, y]$ of total degree d, we want to characterize the values u and v in $\mathbb{K}$ such that $F := G(x+uy+v, y)$ satisfies Hypothesis (H). Let $G^{\#}$ denote the homogeneous component of $G \in \mathbb{K}[x, y]$ of highest degree d.

Lemma 1. *Under Hypothesis (C) and according to the above notation and assumptions, let $u \in \mathbb{K}$ be such that $G^{\#}(u, 1) \neq 0$, then $G_u := G(x + uy, y)$ is monic with respect to y and $\Delta_u(x) := \mathrm{Res}_y(G_u, \frac{\partial G_u}{\partial y})(x) \neq 0$.*

For any $v \in \mathbb{K}$ such that $\Delta_u(v) \neq 0$, $F := G(x + uy + v, y)$ satisfies Hypothesis (H).

Démonstration. It is straightforward to check that $G^{\#}(u, 1)$ is the coefficient of y^d in G_u. Therefore, G_u is monic with respect to y. If we had $\Delta_u(x) = 0$ then this would imply that G_u and $\frac{\partial G_u}{\partial y}$ share a common irreducible factor $H \in \mathbb{K}[x, y]$. Necessarily, one would have $\frac{\partial H}{\partial y} = 0$, which is not possible according to Hypothesis (C). □

The complexity of substituting $x + uy + v$ for x is estimated in the following lemma. This lemma will be used twice : at the beginning and at the end of the main factorization algorithms, this is why we use a new set of notation.

Lemma 2. *Let $\mathbb{E}$ be a commutative ring with unity, let $H \in \mathbb{E}[x, y]$ of total degree n and let u, v in $\mathbb{E}$. If $n!$ is invertible in $\mathbb{E}$ and if we are given its inverse then $H(x + uy + v, y)$ can be computed within $\mathcal{O}(n\mathsf{M}(n))$ arithmetic operations in $\mathbb{E}$.*

Démonstration. We first prove that $H(x + uy, y)$ can be computed within the claimed complexity. If H is homogeneous then $H(x+uy, y)$ is also homogeneous, thus it suffices to compute $H(x+u, 1)$ and homogenize the result with respect to y. The cost of this operation is dominated by the *shift* operation of the variable of a univariate polynomial, which is in $\mathcal{O}(\mathsf{M}(n))$, according to [BP94, Chapter 1, Section 2] and using the inverse of $n!$. If H is not homogeneous, we apply this process on its homogeneous components, which yields a total cost in $\mathcal{O}(\mathsf{M}(1)+\mathsf{M}(2)+\cdots+\mathsf{M}(n))$. Lastly, the super-additivity of M implies $\mathsf{M}(i) \leq M(n)$, for any $i \leq n$, which concludes the case $v = 0$.

If $v \neq 0$ we first compute $H(x+uy, y)$ and then $H(x+uy+v, y)$, therefore it now suffices to examine the case $u = 0$. This task corresponds to shifting the variable x in each coefficient of H seen as a polynomial in y. The total cost of these shifts is again in $\mathcal{O}(n\mathsf{M}(n))$. □

According to the notation of Lemma 1, for a given squarefree polynomial $G \in \mathbb{K}[x, y]$, the number of values for u (resp. v) in $\mathbb{K}$ such that $G^{\#}(u, 1) = 0$ (resp. $\Delta_u(v) = 0$) is at most d (resp. $d(d-1)$). Therefore, the existence of suitable values for u and v is guaranteed by Hypothesis (C). In practice, it is interesting to test values for u (resp. v) in increasing order in the range $[0, \ldots, d]$ (resp. $[0, \ldots, d(d-1)]$). Using fast multi-point evaluation, these tests can be performed faster, as described in the proof of the following proposition :

Proposition 1. *For any squarefree polynomial $G \in \mathbb{K}[x, y]$ of total degree d such that Hypothesis (C) is satisfied, one can compute u and v in $\mathbb{K}$, such that $F := G(x + uy + v, y)$ satisfies Hypothesis (H), within $\mathcal{O}(d^2\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$.*

Démonstration. Following Lemma 1, we first compute u such that $G^{\#}(u, 1) \neq 0$: using a fast multi-point evaluation algorithm, one can compute all the $G^{\#}(i, 1)$, for $i \in \{0, \ldots, d\}$, within $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$ [vzGG03, Chapter 10] (see also [BLS03], for recent advances). Necessarily, one of the $G^{\#}(i, 1)$ is not zero, which determines a suitable value for u.

In order to find a suitable value for v, we partition the set $Z := \{0, \ldots, d(d-1)\}$ into subsets $Z_j := \{j(d+1), \ldots, \min((j+1)(d+1) - 1, d(d-1))\}$, for $j \in \{0, \ldots, \lceil (d(d-1)+1)/(d+1) \rceil - 1\}$. This partition contains at most d subsets. For each of these subsets Z_j, one can compute $\{G_u(i, y) \mid i \in Z_j\}$ within $\mathcal{O}(d\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$. Then one can deduce $\{\Delta_u(i) \mid i \in Z_j\}$ within $\mathcal{O}(d\mathsf{M}(d)\log(d))$. Therefore, the total cost for computing $\{\Delta_u(i) \mid i \in Z\}$ belongs to $\mathcal{O}(d^2\mathsf{M}(d)\log(d))$.

From Lemma 1, one of these values must be nonzero, which provides a suitable value for v. $\square$

In the algorithm described in this proof, the most consuming part is the computation of a suitable value for v. In average, very few tests are necessary : one can expect that the only computation related to Z_0 is sufficient, thus the cost reduces to $\mathcal{O}(d\mathsf{M}(d)\log(d))$. We will not focus on such an average complexity here but we state a probabilistic version relying on finding suitable values for u and v at random. Recall that the notation $\mathcal{U}(P)$ is defined just before Theorem 2.

Proposition 2. *There exists a family of computation trees parametrized by $(u,v) \in \mathbb{K}^2$ such that : for any squarefree polynomial $G \in \mathbb{K}[x,y]$ of degree d such that Hypothesis (C) holds, any executable tree of the family computes a polynomial $F := G(x+uy+v,y)$ satisfying Hypothesis (H). The maximum of the costs of the trees of the family belongs to $\mathcal{O}(d\mathsf{M}(d))$.*

In addition, there exists a nonzero polynomial $P \in \mathbb{K}[U]$ of degree at most d such that, for any $u \in \mathcal{U}(P)$, there exists a nonzero polynomial $Q_u \in \mathbb{K}[V]$ of degree at most $d(d-1)$ such that, for any $v \in \mathcal{U}(Q_u)$, the tree corresponding to (u,v) is executable on G.

Démonstration. For any $(u,v) \in \mathbb{K}^2$ we construct the following computation tree. From Lemma 2 one can compute F within $\mathcal{O}(d\mathsf{M}(d))$ operations in $\mathbb{K}$. Then, Hypothesis (H) can be checked within $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$. The polynomials P and Q_u come from Lemma 1 : we take $P(U) := G^{\#}(U,1)$ and $Q_u(V) := \Delta_u(V)$. $\square$

C.2 Lifting

From now and until the absolute factorization of F is computed, we assume that F satisfies Hypothesis (H). Following the notation of the introduction, we detail the computation of the approximations of ϕ : it is classical to handle this computation by means of iterating Newton's operator $N(y) := y - F(x,y)/\frac{\partial F}{\partial y}(x,y)$, so that $\phi = N^{\kappa}(\varphi) + \mathcal{O}(x^{2^\kappa})$.

A direct complexity analysis of this algorithm proceeds as follows : at the beginning, one needs to compute the inverse of $\frac{\partial F}{\partial y}(0,\varphi)$ in $\mathbb{A}$, this costs $\mathcal{O}(\mathsf{M}(d)\log(d))$. Then, using Hörner's polynomial evaluation scheme, computing the κth iterate from the previous one requires $\mathcal{O}(d)$ ring operations in $\mathbb{A}[[x]]/(x^{2^\kappa})$: the computation of the inverse of $\frac{\partial F}{\partial y}(x,\phi)$ is also handled by means of the classical fast Newton based method.

From the super-additivity of M we deduce

$$\mathsf{M}(1) + \mathsf{M}(2) + \mathsf{M}(4) + \cdots + \mathsf{M}(2^{\lceil \log_2(\sigma) \rceil}) \in \mathcal{O}(\mathsf{M}(\sigma)), \tag{C.5}$$

thus the total cost of this process belongs to :

$$\mathcal{O}(d\mathsf{M}(d)\mathsf{M}(\sigma)). \tag{C.6}$$

We refer to [vzGG03, Chapter 9] for more details about this algorithm. Because this lifting stage is part of the theoretical bottleneck of our absolute factorization algorithm, optimizations are crucial. In order to reach the complexity stated in Theorem 2, we need to replace Hörner's rule by Paterson and Stockmeyer's evaluation scheme [PS73].

C.2.1 Polynomial Evaluation

Let $\mathbb{R}$ denote a commutative ring with unity and $\mathbb{E}$ be a ring extension of $\mathbb{R}$ which is a free $\mathbb{R}$-module of dimension d. We assume we know a basis $E_{1:d}$ of $\mathbb{E}$ such that $E_1 = 1$ and we denote by $E^*_{1:d}$ the canonical dual basis. From a computational point of view, we assume that the elements of $\mathbb{E}$ are represented by their coordinate vectors in the basis $E_{1:d}$.

Paterson and Stockmeyer's evaluation scheme is summarized in the following procedure. For the only computation of ϕ, we could have directly invoked the version described in [vzGG03, Chapter 12.2] but for proving Corollary 2 below (used in Section C.5), we need the following slightly stronger version :

Algorithm Evaluation

Input : a polynomial $P \in \mathbb{R}[y]$ of degree at most d, $e \in \mathbb{E}$.

Output : $P(e) \in \mathbb{E}$.

1. $k := \lfloor \sqrt{d+1} \rfloor$; $k' := \lceil (d+1)/k \rceil$;
2. Compute $1, e, \ldots, e^{k-1}$;
3. Construct the $d \times k$ matrix M with entries in $\mathbb{R}$ defined by $M_{i,j} := E^*_i(e^{j-1})$;
4. Construct the $k \times k'$ matrix N with entries in $\mathbb{R}$ defined by $N_{i,j} := \mathrm{coeff}(P, y^{k(j-1)+i-1})$;
5. Compute the $d \times k'$ matrix $C := MN$;
6. Let $D_i := \sum_{j=1}^{d} C_{j,i} E_j$, for $i \in \{1, \ldots, k'\}$;
7. Compute $1, e^k, \ldots, e^{k(k'-1)}$;
8. Return $\sum_{i=1}^{k'} D_i e^{k(i-1)}$.

In the following proposition we consider operations in $\mathbb{E}$ and matrix multiplication over $\mathbb{R}$ as black-box subroutines that will be specified below, in each case of use.

Proposition 3. *Algorithm Evaluation is correct and costs $\mathcal{O}(\sqrt{d})$ arithmetic operations in $\mathbb{E}$ and one matrix multiplication in size $d \times k$ times $k \times k'$ over $\mathbb{R}$.*

Démonstration. The correction of the algorithm follows from the following identities :

$$P(e) = \sum_{i=1}^{k'} D_i e^{k(i-1)} \text{ and } D_i = \sum_{j=1}^{k} \mathrm{coeff}(P, y^{k(i-1)+j-1}) e^{j-1} \text{ for } i \in \{1, \ldots, k'\}.$$

Concerning the complexity estimate, one obviously has $k \in \mathcal{O}(\sqrt{d})$ and $k' \in \mathcal{O}(\sqrt{d})$. Then Steps (2), (7) and (8) require $\mathcal{O}(\sqrt{d})$ operations in $\mathbb{E}$. The matrix multiplication of the statement is the one of Step (5) and other computations are negligible. □

The following corollary is to be used in the next subsection. We carry on using the notation of Algorithm Evaluation.

Corollary 1. *Under Hypothesis (C), let $\sigma \in \mathcal{O}(d)$, $\mathbb{R} := \mathbb{K}[[x]]/(x^\sigma)$ and $\mathbb{E} := \mathbb{R}[y]/(f(y)) = \mathbb{A}[[x]]/(x^\sigma)$. The cost of Evaluation$(P, \varphi)$ belongs to $\mathcal{O}(\sigma d^{(\omega+1)/2} + \sqrt{d}\mathsf{M}(\sigma)(\mathsf{M}(d) + d\log(d)))$.*

Démonstration. Each ring operation in $\mathbb{E}$ can be done within $\mathcal{O}(\mathsf{M}(\sigma)\mathsf{M}(d))$, thus the conclusion follows from the previous proposition and Lemma 3 below. □

The second corollary is used in Section C.5, in order to test whether a candidate generic factor actually divides F or not.

Corollary 2. *Under Hypotheses (C) and (H), let $\sigma \in \mathcal{O}(d)$, $\mathbb{R} := \mathbb{K}[[x]]/(x^\sigma)$, $q \in \mathbb{K}[z]$, $r := \deg(q)$, $\mathsf{F} \in \mathbb{K}[z, x, y]$ and $\mathbb{E} := \mathbb{R}[z, y]/(q(z), \mathsf{F}(z, x, y))$ such that r divides d, $\deg_z(\mathsf{F}) \leq r - 1$, $\deg_x(\mathsf{F}) \leq \sigma$, $\deg_y(\mathsf{F}) = d/r$ and F is monic with respect to y when seen in $\mathbb{K}[x, z][y]$. Let ψ denote the canonical image of y in $\mathbb{E}$. The cost of Evaluation(F, ψ) belongs to $\mathcal{O}(\sigma d^{(\omega+1)/2} + \sqrt{d}\mathsf{M}(\sigma)(\mathsf{M}(r)\mathsf{M}(d/r) + d\log(d)))$.*

Démonstration. The basis we consider for $\mathbb{E}$ is the canonical basis constituted by the monomials $(z^i y^j)_{0 \leq i \leq r-1,\ 0 \leq j \leq d/r-1}$. Each ring operation in $\mathbb{E}$ can be done within $\mathcal{O}(\mathsf{M}(\sigma)\mathsf{M}(r)\mathsf{M}(d/r))$. Again, the conclusion follows from the previous proposition and Lemma 3 below. □

The following lemma, that is used in these two corollaries, only relies on classical arguments, that we recall for completeness :

Lemma 3. *Under Hypothesis (C) and for any $\sigma \in \mathcal{O}(d)$, one matrix multiplication in sizes $d \times k$ times $k \times k'$ with entries in $\mathbb{K}[[x]]/(x^\sigma)$ can be computed within $\mathcal{O}(\sigma d^{(\omega+1)/2} + \sqrt{d}\mathsf{M}(\sigma) d \log(d))$ operations in $\mathbb{K}$.*

Démonstration. According to the definitions of k and k', this matrix multiplication reduces to multiplying $\mathcal{O}(\sqrt{d})$ matrices in sizes $\mathcal{O}(\sqrt{d} \times \sqrt{d})$ with entries in $\mathbb{K}[[x]]/(x^\sigma)$. Using fast multi-point evaluation and interpolation devices [vzGG03, Chapter 10], each of these matrix products can be handled within $\mathcal{O}(\sigma d^{\omega/2} + d\mathsf{M}(\sigma)\log(\sigma))$ operations in $\mathbb{K}$ this way : first we evaluate the entries of the matrices on $2\sigma - 1$ points taken in $\mathbb{K}$ then we perform the multiplications of the evaluated matrices and last we interpolate the results. Thanks to Hypothesis (C), if $\sigma \leq d(d-1)/2 + 1$ then one can use the set $\{0, \ldots, 2\sigma - 1\}$ for evaluation and interpolation. In consequence, if $\sigma \in \mathcal{O}(d)$ then this algorithm applies except for a finite number of values of d. The claimed complexity then follows by replacing $\log(\sigma)$ by $\log(d)$. □

In the factorization algorithm, we will at most have $\sigma \leq 2d+1$, hence the algorithm of the previous proof applies if $d \geq 5$. For the sake of optimization, it is worth mentioning that one can take advantage of the above special set of interpolating points thanks to the optimized techniques of [BS04].

C.2.2 Lifting of ϕ

We are now able to study the complexity over $\mathbb{K}$ of Newton's operator with call to the above $\mathsf{Evaluation}$ procedure. Recall that our goal is the computation of ϕ up to a given precision (x^σ). If the requested precision σ is not a power of two then one can use a better strategy than doubling the precision at each step, this is the role played by the *precision sequence* $(\sigma_i)_i$ defined by induction : $\sigma_0 := \sigma$ and $\sigma_{i+1} := \lceil \sigma_i/2 \rceil$, for $i \geq 0$. Let ν denote the first integer such that $\sigma_{\nu+1} = 1$.

Algorithm $\mathsf{Lifting}$

Input : $F \in \mathbb{K}[x,y]$ satisfying Hypothesis (H), $\sigma \geq 1$.

Output : ϕ at precision (x^σ).

1. Compute I as the inverse of $\frac{\partial F}{\partial y}(0,\varphi)$ in $\mathbb{A}$;
2. $\psi \leftarrow \varphi$;
3. For i from ν down to 0 do
   (a) Compute $I := I + I(1 - I\mathsf{Evaluation}(\frac{\partial F}{\partial y}, \psi))$ at precision $(x^{\sigma_{i+1}})$;
   (b) Compute $\psi := \psi - I\mathsf{Evaluation}(F, \psi)$ at precision (x^{σ_i}) ;
4. Return ψ.

In the calls to the procedure $\mathsf{Evaluation}$, F and $\frac{\partial F}{\partial y}$ are interpreted as polynomials in $\mathbb{R}[y]$ with $\mathbb{R} := \mathbb{K}[[x]]/(x^k)$ and $\mathbb{E}$ corresponds to $\mathbb{R}[y]/(f(y)) = \mathbb{A}[[x]]/(x^k)$, for k running over the precision sequence.

Proposition 4. *Under Hypotheses (C) and (H), for any $\sigma \in \mathcal{O}(d)$, ϕ can be computed at any precision (x^σ) by Algorithm $\mathsf{Lifting}$ within $\mathcal{O}(\sigma d^{(\omega+1)/2} + \sqrt{d}\mathsf{M}(\sigma)(\mathsf{M}(d) + d\log(d)))$ arithmetic operations in $\mathbb{K}$.*

Démonstration. By induction, we assume that, when entering step i of the loop, we have : I represents the inverse of $\frac{\partial F}{\partial y}(x,\phi)$ at precision $(x^{\sigma_{i+2}})$ and ψ equals ϕ up to precision $(x^{\sigma_{i+1}})$. When $i = \nu$, we have $\sigma_{i+1} = \sigma_{i+2} = 1$, thus the induction hypothesis holds. At Step (3a) the call to $\mathsf{Evaluation}$ returns an approximation of $\frac{\partial F}{\partial y}(x,\phi)$ at precision $(x^{\sigma_{i+1}})$. On the other hand, by the induction hypothesis, we have $1 - I\frac{\partial F}{\partial y}(x,\psi) \in (x^{\sigma_{i+2}})$. Since $\sigma_{i+2} \geq \sigma_{i+1}/2$, it follows that I actually contains the inverse of $\frac{\partial F}{\partial y}(x,\phi)$ at precision $(x^{\sigma_{i+1}})$ at the end of this step. In a similar way, we prove that the variable ψ equals ϕ at precision (x^{σ_i}) at the end of Step (3b).

The cost of Step (1) belongs to $\mathcal{O}(\mathsf{M}(d)\log(d))$, hence is negligible. In step i of the loop, the calls to $\mathsf{Evaluation}$ dominate the complexity and cost $\mathcal{O}(\sigma_i d^{(\omega+1)/2} +$

$\sqrt{d}\mathsf{M}(\sigma_i)(\mathsf{M}(d) + d\log(d)))$, according to Corollary 1. In order to sum these complexities over $i \leq \nu$, we remark that $\sigma_i \leq 2^{\lceil \log_2(\sigma) \rceil - i}$ and we conclude by means of Formula (C.5). □

Using fast polynomial multiplication, that is $\mathsf{M}(d) \in \tilde{\mathcal{O}}(d)$, the cost of this lifting drops to $\mathcal{O}(d^{(\omega+3)/2})$, if $\omega > 2$ and $\sigma \in \mathcal{O}(d)$. Notice that, even with $\omega = 3$, this lifting algorithm gains in the logarithmic factors when compared to the cost (C.6), obtained from the use of Hörner's rule. On the other hand, using slow polynomial multiplication, that is $\mathsf{M}(d) \in \mathcal{O}(d^2)$, we reach $\mathcal{O}(d^{4.5})$, whereas the cost (C.6) is in $\mathcal{O}(d^5)$.

C.3 Recombination

Still following the notation of the introduction and assuming that F satisfies Hypothesis (H), we now address the recombination problem : that is computing the absolute irreducible factors from an approximation of ϕ (obtained from the algorithm of the previous section).

In this section, we prove Theorem 3 and describe algorithms for computing bases of L_σ. The techniques presented here are adapted from [Lec04b] : up to a linear change over $\bar{\mathbb{K}}$ of the variables $\ell_{1:d}$, the system defining L_σ coincide with the one introduced in [Lec04b]. The computation the generic factors from a basis of L_σ is treated in the next section.

C.3.1 Proof of Theorem 3

Let $\phi_{1:d}$ represent the roots of F in $\bar{\mathbb{K}}[[x]]$, so that $F = \prod_{i=1}^{d}(y - \phi_i)$ holds. In order to stay close to the notation of [Lec04b], for $i \in \{1, \ldots, d\}$, we introduce $\mathfrak{F}_i := y - \phi_i$ and the partial products

$$\hat{\mathfrak{F}}_i := \prod_{j=1, j\neq i}^{d} \mathfrak{F}_j.$$

To each $i \in \{1, \ldots, r\}$, we associate the vector $\bar{\mu}_i \in \{0,1\}^d$, defined by

$$F_i = \prod_{j=1}^{d} \mathfrak{F}_j^{\bar{\mu}_{i,j}}. \tag{C.7}$$

If one knows all the ϕ_i up to a sufficient precision, then the factorization of F reduces to computing the $\bar{\mu}_i$. This problem is efficiently solved in [Lec04b] by means of introducing the following linear system over $\bar{\mathbb{K}}$:

$$\begin{aligned}\bar{L}_\sigma := \Big\{ (\bar{\ell}_{1:d}, \bar{G}, \bar{H}) \in \bar{\mathbb{K}}^d \times \bar{\mathbb{K}}[x,y]_{d-1} \times \bar{\mathbb{K}}[x,y]_{d-1} \mid \\ \bar{G} - \sum_{i=1}^{d} \bar{\ell}_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y} \in (x,y)^\sigma, \\ \bar{H} - \sum_{i=1}^{d} \bar{\ell}_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x} \in (x,y)^\sigma + (x^{\sigma-1}) \Big\}.\end{aligned}$$

Differentiating (C.7) with respect to x and y respectively gives :

$$\hat{F}_i \frac{\partial F_i}{\partial x} = \sum_{j=1}^{d} \bar{\mu}_{i,j} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial x} \quad \text{and} \quad \hat{F}_i \frac{\partial F_i}{\partial y} = \sum_{j=1}^{d} \bar{\mu}_{i,j} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial y},$$

whence the inclusion $\langle \bar{\mu}_{1:r} \rangle \subseteq \pi(\bar{L}_\sigma)$. If σ is sufficiently large then this inclusion becomes an equality, as stated in :

Theorem 4. *[Lec04b, Theorem 1] Under Hypotheses (C) and (H), for any $\sigma \geq 2d$ we have :*

$$\bar{L}_\sigma = \left\langle \left(\bar{\mu}_i, \hat{F}_i \frac{\partial F_i}{\partial y}, \hat{F}_i \frac{\partial F_i}{\partial x} \right) \mid i \in \{1, \ldots, r\} \right\rangle.$$

The next proposition tells us that $\bar{L}_\sigma$ and the extension $\bar{\mathbb{K}} \otimes L_\sigma$ of L_σ are isomorphic. For short, we write $\varphi_i := \phi_i(0)$ and we introduce the following isomorphism that sends φ to $(\varphi_1, \ldots, \varphi_d)$:

$$\begin{aligned}\bar{\mathbb{K}} \otimes \mathbb{A} &\to \bar{\mathbb{K}}^d \\ b &\mapsto (b(\varphi_1), \ldots, b(\varphi_d)).\end{aligned}$$

With respect to the canonical bases of $\bar{\mathbb{K}}[y]/(f(y)) = \bar{\mathbb{K}} \otimes \mathbb{A}$ and $\bar{\mathbb{K}}^d$, the matrix of this map is the Vandermonde matrix V of $(\varphi_1, \ldots, \varphi_d)$.

Proposition 5. *Under Hypothesis (H), for any $\sigma \geq 1$, the map*

$$\begin{array}{rcl}\Sigma : \quad \bar{L}_\sigma & \to & \bar{\mathbb{K}} \otimes L_\sigma \\ (\bar{\ell}, \bar{G}, \bar{H}) & \mapsto & (V^t \bar{\ell}, \bar{G}, \bar{H})\end{array} \tag{C.8}$$

is an isomorphism.

Démonstration. For any $b \in \bar{\mathbb{K}} \otimes \mathbb{A}$, $\ell \in \bar{\mathbb{K}}^d$ and $\bar{\ell} \in \bar{\mathbb{K}}^d$ such that $\ell = V^t \bar{\ell}$, one has :

$$\begin{aligned}\sum_{i=1}^{d} \ell_i \operatorname{coeff}(b, \varphi^{i-1}) &= \sum_{i=1}^{d} \operatorname{coeff}(b, \varphi^{i-1}) \sum_{j=1}^{d} \bar{\ell}_j \varphi_j^{i-1} \\ &= \sum_{j=1}^{d} \bar{\ell}_j \sum_{i=1}^{d} \operatorname{coeff}(b, \varphi^{i-1}) \varphi_j^{i-1} \\ &= \sum_{j=1}^{d} \bar{\ell}_j b(\varphi_j).\end{aligned}$$

One the other hand, it is straightforward to verify that substituting φ_i for φ in $\mathfrak{F}$ gives $\mathfrak{F}_i$. Thus the map Σ is well defined and is clearly an isomorphism. □

Now, the proof of Theorem 3 directly follows from combining this proposition with Theorem 4, since $\mu_i = V^t\bar{\mu}_i$.

In order to compute a basis of L_σ it suffices to compute a basis of $\pi(L_\sigma)$, which leads to considering a linear system in d unknowns only. We next study the complexity of this linear system solving. We first detail the natural deterministic method and then we adapt the probabilistic speed-up presented in [BLS+04, Lec04b], which gains in reducing the number of equations.

C.3.2 Deterministic Method

From the approximation of ϕ at precision (x^σ) and the definition of L_σ it is straightforward to compute a basis of $\pi(L_\sigma)$. For this purpose, we introduce the following linear system D_σ, with d unknowns $\ell_{1:d}$:

$$D_\sigma \begin{cases} \sum_{i=1}^{d} \ell_i \operatorname{coeff}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial y}, \varphi^{i-1}x^j y^k) = 0, \ k \leq d-1, \ d \leq j+k \leq \sigma - 1, \\ \sum_{i=1}^{d} \ell_i \operatorname{coeff}(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial x}, \varphi^{i-1}x^j y^k) = 0, \ k \leq d-1, \ j \leq \sigma - 2, \ d \leq j+k \leq \sigma - 1. \end{cases}$$

Lemma 4. *For all $\sigma \geq 1$, we have $\pi(L_\sigma) = \{\ell_{1:d} \in \mathbb{K}^d \mid D_\sigma\}$.*

Démonstration. The proof is straightforward from the definition of $\pi(L_\sigma)$. □

The deterministic algorithm for computing a basis of $\pi(L_\sigma)$ proceeds as follows :

Algorithm DeterministicSolve

Input :
- A polynomial F satisfying Hypothesis (H) ;
- ϕ at precision (x^σ).

Output : a basis of $\pi(L_\sigma)$.

1. Compute $\hat{\mathfrak{F}} = F/(y-\phi)$ at precision (x^σ), by means of a polynomial division algorithm and build the linear system D_σ ;
2. Compute and return a basis of solutions of the linear system D_σ.

Let us recall here that computing a solution basis of a linear system of m equations in $d \leq m$ unknowns over $\mathbb{K}$ can be done within

$$\mathcal{O}(md^{\omega-1}) \tag{C.9}$$

operations in $\mathbb{K}$ [BP94, Chapter 2]. We deduce the following complexity estimate :

Proposition 6. *For any $\sigma \geq d$, Algorithm DeterministicSolve is correct and performs $\mathcal{O}(\sigma d^\omega + d\mathsf{M}(\sigma)\mathsf{M}(d))$ arithmetic operations in $\mathbb{K}$.*

Démonstration. The computation of $\hat{\mathfrak{F}}$ can be handled by means of a naive division algorithm, which performs $\mathcal{O}(d)$ ring operations in $\mathbb{A}[[x]]/(x^\sigma)$, whence the second term of the complexity. The cost of the construction of D_σ is negligible. Then the first term comes from the bound (C.9), applied with $m \in \mathcal{O}(\sigma d)$, which upper bounds the number of equations of D_σ. □

Note that, when using fast polynomial multiplication, i.e. $\mathsf{M}(d) \in \tilde{\mathcal{O}}(d)$, $\omega > 2$ and $\sigma = 2d$, the complexity estimate of this proposition drops into $\mathcal{O}(d^{\omega+1})$.

C.3.3 Probabilistic Method

According to Theorem 3, we shall take $\sigma = 2d$ in order to be able to recover the absolute factorization. At this precision, D_σ involves about d^2 equations for d unknowns. A classical trick for reducing the complexity of the resolution of such overdetermined systems consists in replacing the original set of equations by fewer random linear combinations of them. For this purpose, we adapt the fast probabilistic strategy of [Lec04b, Section 3] in our context. As in [Lec04b, Section 3], this strategy requires a slightly larger precision in the computation of ϕ. This precision is denoted by τ instead of σ, in order to avoid confusion.

For any $u \in \mathbb{K}$, we introduce the following linear system P_τ^u :

$$P_\tau^u \left\{ \begin{array}{l} \sum_{i=1}^d \ell_i \operatorname{coeff}(\hat{\mathfrak{F}}(x,ux)\frac{\partial \mathfrak{F}}{\partial x}(x,ux), \varphi^{i-1}x^j) = 0,\ d \le j \le \tau - 2, \\ \sum_{i=1}^d \ell_i \operatorname{coeff}(\hat{\mathfrak{F}}(x,ux)\frac{\partial \mathfrak{F}}{\partial y}(x,ux), \varphi^{i-1}x^j) = 0,\ d \le j \le \tau - 2. \end{array} \right.$$

Of course, one has : $\frac{\partial \mathfrak{F}}{\partial x}(x,ux) = -\phi'(x)$ and $\frac{\partial \mathfrak{F}}{\partial y}(x,ux) = 1$. For any $(a,b) \in \mathbb{K}^2$, $P_\tau^{a,b}$ represents the union of the two systems P_τ^a and P_τ^b. The following lemma relates the solution set of $P_\tau^{a,b}$ to our recombination problem :

Lemma 5. *For any $(a,b) \in \mathbb{K}^2$, one has $\pi(L_\tau) \subseteq \{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\}$. In addition, for any $b \in \mathbb{K}$, there exists a nonzero polynomial $R_b \in \mathbb{K}[A]$ of degree at most $d(d-1)$ such that, for all $a \in \mathcal{U}(R_b)$, one has $\{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\} \subseteq \pi(L_{\tau-1})$.*

Démonstration. The proof closely follows the one of [Lec04b, Lemma 5]. We introduce :

$$\begin{aligned} \Lambda_\tau := \Big\{ & \ell_{1:d} \in \mathbb{K}^d \mid \\ & \sum_{i=1}^d \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial x}, \varphi^{i-1}x^j y^k\right) = 0,\ k \le d-1,\ d \le j+k \le \tau - 2, \\ & \sum_{i=1}^d \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}\frac{\partial \mathfrak{F}}{\partial y}, \varphi^{i-1}x^j y^k\right) = 0,\ k \le d-1,\ d \le j+k \le \tau - 2\Big\}. \end{aligned}$$

From definitions, it is straightforward to check that

$$\pi(L_\tau) \subseteq \Lambda_\tau \subseteq \pi(L_{\tau-1}). \tag{C.10}$$

Let A denote a new variable, substituting Ax for y in the above definition of Λ_τ, we get :

$$\begin{aligned}\Lambda_\tau =\Big\{&\ell_{1:d} \in \mathbb{K}^d \mid \\ &\sum_{i=1}^{d} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}(x,Ax)\frac{\partial \mathfrak{F}}{\partial x}(x,Ax), \varphi^{i-1}x^j A^k\right) = 0,\ k \le d-1,\ d \le j \le \tau-2,\\ &\sum_{i=1}^{d} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}(x,Ax)\frac{\partial \mathfrak{F}}{\partial y}(x,Ax), \varphi^{i-1}x^j A^k\right) = 0,\ k \le d-1,\ d \le j \le \tau-2\Big\}.\end{aligned}$$

For any $u \in K$, we define $\Lambda_\tau^u := \{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^u\}$. Obviously, we have $\Lambda_\tau \subseteq \Lambda_\tau^a \cap \Lambda_\tau^b = \{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\}$, for any a and b. Let $\lambda_{1:\dim(\Lambda_\tau^b)}$ be a basis of Λ_τ^b such that the $\dim(\Lambda_\tau)$ first vectors form a basis of Λ_τ. For any $\lambda_k \notin \Lambda_\tau$ there exists $j \in \{d, \ldots, \tau-2\}$ such that one polynomial among

$$\sum_{i=1}^{d} \lambda_{k,i} \operatorname{coeff}\left(\hat{\mathfrak{F}}(x,Ax)\frac{\partial \mathfrak{F}}{\partial y}(x,Ax), \varphi^{i-1}x^j\right) \text{ and}$$
$$\sum_{i=1}^{d} \lambda_{k,i} \operatorname{coeff}\left(\hat{\mathfrak{F}}(x,Ax)\frac{\partial \mathfrak{F}}{\partial x}(x,Ax), \varphi^{i-1}x^j\right)$$

is not zero : let $r_k(A)$ represent one of them which is not zero. We take R_b as the product of all such r_k. By construction, each r_k has degree at most $d-1$. Lastly, for any $a \in \mathbb{K}$ such that $R_b(a) \neq 0$, we have $\lambda_k \notin \Lambda_\tau^a$, whence $\Lambda_\tau = \Lambda_\tau^a \cap \Lambda_\tau^b$ and the conclusion follows from (C.10). $\square$

From an algorithmic point of view, the difficulty now relies in constructing this new linear system $P_\tau^{a,b}$ in an efficient way. We start with the same idea as in [Lec04b] : we attempt to compute $\hat{\mathfrak{F}}(x,ux)$ for $u \in \{a,b\}$ without performing the exact division of $F(x,y)$ by $y-\phi(x)$. Our strategy relies on inverting $ux-\phi(x)$: if it is not invertible then we split the computation. For this purpose, we introduce $f_y := \gcd(f,y)$, $\hat{f}_y := f/f_y$, $\mathbb{A}_y := \mathbb{K}[y]/(f_y)$, $\hat{\mathbb{A}}_y := \mathbb{K}[y]/(\hat{f}_y)$. By construction, there exists an isomorphism $\mathfrak{A} : \mathbb{A} \to \mathbb{A}_y \times \hat{\mathbb{A}}_y$. If f_y is constant then $\mathbb{A}_y = \{0\}$ and $\mathbb{A} = \hat{\mathbb{A}}_y$. This corresponds to the case when φ is invertible in $\mathbb{A}$ or, equivalently ϕ invertible in $\mathbb{A}[[x]]$. Otherwise $f_y = y$ and one has $\mathbb{A}_y = \mathbb{K}$.

For complexity estimates, we shall explicit the isomorphism $\mathfrak{A}$ and its inverse. For any $\bar{\beta} \in \mathbb{A}$, $\mathfrak{A}(\bar{\beta}) = (\beta \bmod f_y, \beta \bmod \hat{f}_y)$, where $\beta \in \mathbb{K}[y]$ represents the canonical preimage of $\bar{\beta}$ of degree $d-1$. This computation of $\mathfrak{A}(\bar{\beta})$ requires $\mathcal{O}(d)$ operations in $\mathbb{K}$: if f_y is constant then $\mathfrak{A}$ is the identity map, otherwise $f_y = y$ thus $\beta \bmod f_y$ only consists in taking the constant coefficient of β and $\beta \bmod \hat{f}_y$ corresponds to the computation of the remainder in the division of a polynomial of degree at most $d-1$ by a polynomial of degree $d-1$, which can be done with the naive division algorithm within $\mathcal{O}(d)$ operations in $\mathbb{K}$.

Concerning the inverse $\mathfrak{A}^{-1}$, if $\mathfrak{A}$ is not the identity, for any $(\bar{\beta}_1, \bar{\beta}_2) \in \mathbb{A}_y \times \hat{\mathbb{A}}_y$, we write β_1 and β_2 respectively for the canonical preimages of $\bar{\beta}_1$ and $\bar{\beta}_2$ in $\mathbb{K}[y]$. The inverse $\bar{\beta} := \mathfrak{A}^{-1}(\bar{\beta}_1, \bar{\beta}_2)$ is given by the following formulas, where β represents again the canonical preimage of $\bar{\beta}$ in $\mathbb{K}[y]$: $\beta = \beta_1 + y\beta_3$, where β_3 is the canonical preimage in $\mathbb{K}[y]$ of $(\beta_2 - \beta_1)/y$ in $\hat{\mathbb{A}}_y$. Notice that the inverse of y in $\hat{\mathbb{A}}_y$ can be easily obtained from the Euclidean division of f_y by y, which only requires $\mathcal{O}(d)$ operations in $\mathbb{K}$. In all cases, computing $\mathfrak{A}^{-1}(\bar{\beta}_1, \bar{\beta}_2)$ requires $\mathcal{O}(\mathsf{M}(d))$ operations in $\mathbb{K}$.

With a slight abuse of notation, we still write $\mathfrak{A}$ for the natural extension of $\mathfrak{A}$ to $\mathbb{A}[[x]] \to \mathbb{A}_y[[x]] \times \hat{\mathbb{A}}_y[[x]]$ that maps $\mathfrak{A}$ coefficients by coefficients. Lastly, we write $\mathfrak{A}_y$ (resp. $(\hat{\mathfrak{A}}_y)$) for the first (resp. second) projection of $\mathfrak{A}$, so that $\mathfrak{A} = (\mathfrak{A}_y, \hat{\mathfrak{A}}_y)$. We are ready to present our probabilistic algorithm for computing bases of $\pi(L_\sigma)$.

Algorithm ProbabilisticSolve

Input :
- A polynomial F satisfying Hypothesis (H) ;
- ϕ at precision (x^τ) ;
- $(a, b) \in \mathbb{K}^2$.

Output : a basis of $\{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\}$.

1. For all $u \in \{a, b\}$ do
   (a) In $\hat{\mathbb{A}}_y[[x]]$ compute $\hat{\mathfrak{f}}_y := F(x, ux)/\hat{\mathfrak{A}}_y(ux - \phi)$ at precision (x^τ) ;
   (b) In $\mathbb{A}_y[[x]][y]$ compute the quotient $\hat{\mathfrak{F}}_y$ of $F(x, y)$ by $y - \mathfrak{A}_y(\phi)$ at precision (x^τ) and let $\mathfrak{f}_y := \hat{\mathfrak{F}}_y(x, ux)$;
   (c) Compute $\hat{\mathfrak{F}}(x, ux)$ as $\mathfrak{A}^{-1}(\mathfrak{f}_y, \hat{\mathfrak{f}}_y)$ at precision (x^τ) ;
2. Compute and return a basis of $\{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\}$.

We now show that this algorithm is correct and analyze its complexity.

Proposition 7. *For any $\tau \geq d$, Algorithm ProbabilisticSolve is correct and performs $\mathcal{O}(\tau d^{\omega-1} + \mathsf{M}(\tau)\mathsf{M}(d))$ operations in $\mathbb{K}$.*

Démonstration. If $\mathbb{A}_y = \{0\}$ then Steps (1b) and (1c) are useless so that we can focus on the case $\mathbb{A}_y \neq \{0\}$. Since $\mathfrak{F}$ exactly divides F and $\hat{\mathfrak{A}}_y(ux - \phi)$ is invertible, $\hat{\mathfrak{f}}_y$ equals $\hat{\mathfrak{A}}_y(\hat{\mathfrak{F}}(x, ux))$ at precision (x^τ). On the other hand $y - \mathfrak{A}_y(\phi)$ divides F hence $\mathfrak{f}_y$ equals $\mathfrak{A}_y(\hat{\mathfrak{F}}(x, ux))$ at precision (x^τ).

Concerning the complexity estimates, Step (1a) can make use of Newton's iteration [vzGG03, Chapter 9.1] in order to perform the series inversion within $\mathcal{O}(\mathsf{M}(d)(\log(d) + \mathsf{M}(\tau)))$. The cost of Step (1b) belongs to $\mathcal{O}(d\mathsf{M}(\tau))$ and Step (1c) requires $\mathcal{O}(\tau\mathsf{M}(d))$. According to Bound (C.9), the resolution of the linear system in Step 2 requires $\mathcal{O}(\tau d^{\omega-1})$, since the system contains $\mathcal{O}(\tau)$ equations. □

Note that, when using fast polynomial multiplication, i.e. $\mathsf{M}(d) \in \tilde{\mathcal{O}}(d)$, $\omega > 2$ and $\tau = 2d + 1$, the complexity estimate of this proposition drops into $\mathcal{O}(d^\omega)$.

C.4 Recovering the Absolute Factors

In this section, we assume that we are given a basis $\nu_{1:r}$ of $\pi(L_\infty)$ and we explain how to compute the generic factors.

C.4.1 Residue Basis

According to the definitions, there correspond to each ν_i unique polynomials $G_i \in \mathbb{K}[x,y]_{d-1}$ and $H_i \in \mathbb{K}[x,y]_{d-1}$ such that $(\nu_i, G_i, H_i) \in L_\infty$. These polynomials can be obtained by means of the following formulas, in which the series can be truncated at precision (x^d) :

$$G_i = \sum_{j=1}^{d} \nu_{i,j} \operatorname{coeff}\left(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial y}, \varphi^{i-1}\right), \tag{C.11}$$

$$H_i = \sum_{j=1}^{d} \nu_{i,j} \operatorname{coeff}\left(\hat{\mathfrak{F}} \frac{\partial \mathfrak{F}}{\partial x}, \varphi^{i-1}\right).$$

Since $((\nu_i, G_i, H_i))_{i \in \{1,\ldots,r\}}$ is a basis of L_∞, we deduce from Theorem 3 that their exists an invertible $r \times r$ matrix $(\rho_{i,j})_{(i,j)}$ with entries in $\bar{\mathbb{K}}$ such that :

$$(\nu_i, G_i, H_i) = \sum_{j=1}^{r} \rho_{j,i} \left(\mu_j, \hat{F}_j \frac{\partial F_j}{\partial y}, \hat{F}_j \frac{\partial F_j}{\partial x}\right). \tag{C.12}$$

In particular, the set of vectors $\{\rho_1, \ldots, \rho_r\}$ has cardinality r and we are going to show that the computation of the generic factors reduces to find a point $c_{1:r} \in \mathbb{K}^r$ such that the corresponding linear form $c : t_{1:r} \mapsto c_1 t_1 + \cdots + c_r t_r$ satisfies $c(\rho_i) \neq c(\rho_j)$ for all $i \neq j$. In this case we say that c *separates* the ρ_i.

For the sake of efficiency, we aim to keep x specialized to 0 as far as possible in the computations. This is why we introduce $g_i := G_i(0,y)$ and $\hat{f}_i := \hat{F}_i(0,y)$. Recall that $f_i := F_i(0,y)$ has been already defined. Substituting 0 for x in Equation (C.12) one obtains

$$g_i = \sum_{j=1}^{r} \rho_{j,i} \hat{f}_j f'_j. \tag{C.13}$$

It follows that $\rho_{i,j}$ is the residue of g_i/f with respect to f_j. This is why we call $\rho_{1:r}$ the *residue basis* associated to $\nu_{1:r}$. At this point it is worth noting that $(\hat{f}_i f'_i)_{i \in \{1,\ldots,r\}}$ is a free family, therefore $(g_i)_{i \in \{1,\ldots,r\}}$ is also free.

C.4.2 Residue Separation

Let $c_{1:r} \in \mathbb{K}^r$ and let s denote the cardinality of $\{c(\rho_i) \mid i \in \{1,\ldots,r\}\}$: s equals r if and only if c separates the ρ_i. In order to prove the existence of separating forms we show the following stronger quantitative result :

Lemma 6. *According to the above notation and assumptions, there exists a nonzero polynomial $S \in \bar{\mathbb{K}}[C_{1:r}]$ of total degree $r(r-1)/2$ such that any $c_{1:r} \in \mathcal{U}(S)$ separates the ρ_i.*

Démonstration. Clearly, the following polynomial suits us :

$$S := \prod_{i<j} \left(\sum_{k=1}^{r} (\rho_{i,k} - \rho_{j,k}) C_k \right). \qquad \square$$

Under Hypothesis (C) the set $\mathcal{S} := \{0, \ldots, d(d-1)\}$ has cardinality $d(d-1)+1$. Applying Zippel-Schwartz' zero-test [Zip93, Sch80] with the polynomial S defined in this lemma, we deduce that the cardinality of $\mathcal{U}(S) \cap \mathcal{S}^r$ is bounded by $(d(d-1)+1)^{r-1} r(r-1)/2$. Since $r \leq d$, we conclude that there exist separating forms. From a quantitative point of view, the proportion of separating forms in $\mathcal{S}^r$ is a least $1 - (r(r-1)/2)/(d(d-1)+1) \geq 1/2$. By the way these arguments can be adapted to give a shorter proof of [Gao03, Theorem 2.10].

C.4.3 Main Formulas

We introduce $d_i := \deg(F_i)$, $\Delta := \mathrm{Res}_y(F, \frac{\partial F}{\partial y})$,

$$Q(z) := \prod_{i=1}^{r} (z - c(\rho_i))^{d_i} \quad \text{and} \quad q(z) := \prod_{\beta \in \{c(\rho_j) | j \in \{1,\ldots,r\}\}} (z - \beta).$$

Of course one has $\delta = \Delta(0)$. Lemmas 7 and 9 below will be used several times. They are at the heart of Rothstein's and Trager's integration algorithm [vzGG03, Theorem 22.8]. Because we have to deal with positive characteristic, we detail the proofs. The first lemma gives a formula for computing Q.

Lemma 7. *According to the above notation we have :*

$$\mathrm{Res}_y \left(F, z\frac{\partial F}{\partial y} - c(G_{1:r}) \right) = \Delta(x) Q(z) \quad \textit{and} \quad \mathrm{Res}_y \left(f, zf' - c(g_{1:r}) \right) = \delta Q(z).$$

Démonstration. From Formula (C.12), we directly obtain :

$$c(G_{1:r}) = \sum_{i=1}^{r} c(\rho_i) \hat{F}_i \frac{\partial F_i}{\partial y}.$$

Taking this equality modulo F_i yields :

$$c(G_{1:r}) \bmod F_i = c(\rho_i) \hat{F}_i \frac{\partial F_i}{\partial y} \bmod F_i = c(\rho_i) \frac{\partial F}{\partial y} \bmod F_i.$$

Using the multiplicative property of the resultant, we deduce :

$$\begin{aligned}\mathrm{Res}_y\left(F, z\frac{\partial F}{\partial y} - c(G_{1:r})\right) &= \prod_{i=1}^{r} \mathrm{Res}_y\left(F_i, z\frac{\partial F}{\partial y} - c(G_{1:r})\right)\\ &= \prod_{i=1}^{r} \mathrm{Res}_y\left(F_i, \frac{\partial F}{\partial y}(z - c(\rho_i))\right)\\ &= \prod_{i=1}^{r} \mathrm{Res}_y\left(F_i, \frac{\partial F}{\partial y}\right)\prod_{i=1}^{r} \mathrm{Res}_y\left(F_i, z - c(\rho_i)\right)\\ &= \mathrm{Res}_y\left(F, \frac{\partial F}{\partial y}\right)\prod_{i=1}^{r}(z - c(\rho_i))^{d_i}.\end{aligned}$$

The second formula follows the same way when substituting 0 for x. □

In particular, we deduce that $Q \in \mathbb{K}[z]$. Since Q has degree d, Hypothesis (C) implies that q can be computed as the squarefree part of Q :

Lemma 8. *According to the above notation, q is separable and satisfies*

$$q = \frac{Q}{\gcd(Q, Q')}.$$

The last lemma of this section gives us formulas for computing the absolute factors. We adapt to proof of [vzGG03, Lemma 14.22] in order to use Hypothesis (H) in case of positive characteristic.

Lemma 9. *According to the above notation, for all $i \in \{1, \ldots, r\}$, one has :*

$$\gcd\left(F, c(\rho_i)\frac{\partial F}{\partial y} - c(G_{1:r})\right) = \prod_{\{j | c(\rho_j) = c(\rho_i)\}} F_j$$

and also

$$\gcd\left(f, c(\rho_i)f' - c(g_{1:r})\right) = \prod_{\{j | c(\rho_j) = c(\rho_i)\}} f_j.$$

Démonstration. For any $j \in \{1, \ldots, r\}$, taking both sides of the equality

$$c(\rho_i)\frac{\partial F}{\partial y} - c(G_{1:r}) = \sum_{j=1}^{r}(c(\rho_i) - c(\rho_j))\hat{F}_j\frac{\partial F_j}{\partial y}$$

modulo F_j, we obtain :

$$c(\rho_i)\frac{\partial F}{\partial y} - c(G_{1:r}) \bmod F_j = (c(\rho_i) - c(\rho_j))\hat{F}_j\frac{\partial F_j}{\partial y} \bmod F_j.$$

Now substituting 0 for $x = 0$, this equality reads as :

$$c(\rho_i)f' - c(g_{1:r}) \bmod f_j = (c(\rho_i) - c(\rho_j))\hat{f}_j f_j' \bmod f_j.$$

Thanks to Hypothesis (H), $\hat{f}_j f_j'$ is invertible modulo f_j, and hence : F_j divides $c(\rho_i)\frac{\partial F}{\partial y} - c(G_{1:r})$ if and only if $c(\rho_i) - c(\rho_j) = 0$, which concludes the proof. □

C.4.4 General Factorization Algorithm

We are now ready to present the general factorization algorithm :

Algorithm AbsoluteFactorization

Input : a squarefree polynomial F in $\mathbb{K}[x, y]$ of degree d satisfying Hypothesis (C).

Output : the generic absolute factors $\mathsf{F}_1, \ldots, \mathsf{F}_m$ of F.

1. Use the algorithm described in the proof of Proposition 1 with F in order to compute $(u, v) \in \mathbb{K}^2$ such that $F(x+uy+v, y)$ satisfies Hypothesis (H). Now replace F by $F(x + uy + v, y)$.
2. Let $\sigma = 2d$ and compute ϕ at precision (x^σ) as $\mathsf{Lifting}(F, \sigma)$;
3. Compute $\nu_{1:r} := \mathsf{DeterministicSolve}(F, \phi)$;
4. Compute $G_{1:r}$ by means of Formula (C.11) ;
5. Find a point $c_{1:r} \in \mathbb{K}^r$ such that c separates the residue basis $\rho_{1:r}$ corresponding to $\nu_{1:r}$, as defined in Section C.4.1 ;
6. Compute $Q := \mathrm{Res}_y\left(f, zf' - c(g_{1:r})\right)/\delta$;
7. Compute $q := Q/\gcd(Q, Q')$;
8. Decompose q into irreducible factors $q_1, \ldots, q_m$;
9. For all $i \in \{1, \ldots, m\}$, let $\mathbb{F}_i := \mathbb{K}[z]/(q_i(z))$, let α_i denote the canonical image of z in $\mathbb{F}_i$, and compute
$$\mathsf{F}_i := \gcd\left(F, \alpha_i \frac{\partial F}{\partial y} - c(G_{1:r})\right) ; \tag{C.14}$$
10. Return $\mathsf{F}_{1:m}(x - uy - v, y)$.

The complexity analysis of this algorithm is not studied in this article but is work in progress in [CL04]. In particular, we do not explain how to find a separating form c in Step (5). Nevertheless we can prove :

Proposition 8. *Algorithm AbsoluteFactorization is correct.*

Démonstration. Thanks to Propositions 1, 4 and 6 we are sure that $\nu_{1:r}$ is a basis of $\pi(L_\infty)$ at the end of Step (3). In Step (6) the computation of q uses Lemma 7, and in Step (7) the computation of q uses Lemma 8.

For each $i \in \{1, \ldots, m\}$ we introduce the subset I_i of $\{1, \ldots, r\}$ defined by :
$$q_i = \prod_{j \in I_i} (z - c(\rho_j)).$$
For any $j \in I_i$ we introduce the embedding $\chi_{i,j}$ of $\mathbb{F}_i$ in $\bar{\mathbb{K}}$ that maps α_i to $c(\phi_j)$. Mapping $\chi_{i,j}$ to both sides of Formula (C.14) and thanks to Lemma 9, we obtain :
$$\chi_{i,j}(\mathsf{F}_i) = \gcd\left(F, c(\phi_j)\frac{\partial F}{\partial y} - c(G_{1:r})\right) = F_j.$$

It follows that $\bar{F}_i := \prod_{j\in I_i} F_j$ belongs to $\mathbb{K}[x,y]$. We now claim that $\bar{F}_i$ is irreducible. If this were not the case then we would have a strict subset I_i' of I_i such that $\prod_{j\in I_i'} F_j$ would be a proper factor of $\bar{F}_i$. Using the same calculations as in the proof of Lemma 7, we would deduce that the squarefree part of

$$\mathrm{Res}_y\left(\prod_{j\in I_i'} F_j, z\frac{\partial F}{\partial y} - c(G_{1:r})\right) = \mathrm{Res}_y\left(\prod_{j\in I_i'} F_j, \frac{\partial F}{\partial y}\right)\prod_{j\in I_i'}(z - c(\rho_j))^{d_i}$$

would be a proper factor of q_i. At the end we have proved that F_i is the generic absolute factor of the rational factor $\bar{F}_i$, which concludes the proof. □

In the rest of this article, we restrict ourselves to the useful case when F is irreducible over $\mathbb{K}$, which means that F has only one generic factor F (i.e. $m = 1$). In particular, we propose optimized algorithms, which deeply use the following property :

Corollary 3. *Under Hypotheses (C) and (H), if F is irreducible then q is irreducible and $Q = q^{d/s}$. In particular, all the absolute irreducible factors of F have the same degree d/r.*

Démonstration. If q had a non-trivial factor p then

$$\prod_{p(\beta)=0} \gcd\left(F, \beta\frac{\partial F}{\partial y} - c(G_{1:r})\right)$$

would be a non-trivial rational factor of F, according to Lemma 9. From Lemma 7, we deduce that all the roots of Q have the same multiplicities, which concludes the proof. □

In the next section, we study the complexity of computing F, assuming we are given a separating form c. The problem of computing such a form in a deterministic way is studied in the last section.

C.5 Probabilistic Algorithm

From now on we assume that F is irreducible and we suppose suppose that we are given a free family of vectors $\nu_{1:\tilde{r}}$ of $\mathbb{K}^d$ such that $\pi(L_\infty) \subseteq \langle \nu_{1:\tilde{r}}\rangle$, hence $\tilde{r} \geq r$. These family is supposed to have been calculated by the probabilistic algorithm of Section C.3.3, thus it may appear that we have the strict inequality $\tilde{r} > r$.

C.5.1 Probabilistic Generic Factor

We compute the generic factor as if we had $\tilde{r} = r$, but we stop the execution as soon as the computation is not possible. Formally speaking, the execution is stopped by means of computing $1/0$. At the end, if the candidate absolute factor actually divides F, then we show that it is necessarily correct. In particular, this allows one to prove that $\tilde{r} = r$ *a posteriori*.

Algorithm $\mathsf{ProbabilisticGenericFactor}$

Input :
- an irreducible polynomial F of degree d such that Hypotheses (C) and (H) hold ;
- a free family $\nu_{1:\tilde{r}}$ of vectors of $\mathbb{K}^d$ such that $\pi(L_\infty) \subseteq \langle \nu_{1:\tilde{r}} \rangle$;
- $\tilde{c}_{1:\tilde{r}} \in \mathbb{K}^{\tilde{r}}$.

Output : the generic factor F of F.

1. Let $\mathfrak{f}'(y) := y - \varphi$ and compute $\hat{\mathfrak{f}}(y)$ as the quotient of f by $\mathfrak{f}'$ in $\mathbb{A}[y]$.
2. Compute
$$\tilde{\nu} := \left(\sum_{j=1}^{\tilde{r}} \tilde{c}_j \nu_{j,i} \right)_{i \in (1,\ldots,d)}$$
and then
$$h := \sum_{i=1}^{d} \tilde{\nu}_i \operatorname{coeff}(\hat{\mathfrak{f}}(y)\mathfrak{f}'(y), \varphi^{i-1}).$$
3. Compute $\tilde{Q}(z)$ as the monic polynomial proportional to $\operatorname{Res}_y(f(y), zf'(y) - h(y))$.
4. Compute $\tilde{q}$ as the squarefree part of $\tilde{Q}$; If $\tilde{q}$ has not degree $\tilde{r}$ then stop the execution.
5. Let $\tilde{\mathbb{F}} := \mathbb{K}[z]/(\tilde{q}(z))$, let $\tilde{\alpha}$ represent the canonical image of z in $\mathbb{F}$ and compute $\tilde{\mathsf{f}} := \gcd(f, \tilde{\alpha} f' - h)$, as if $\tilde{\mathbb{F}}$ were a field : but stop the execution when running into the inversion of a zero divisor in $\tilde{\mathbb{F}}$; Stop the execution if $\deg(\tilde{\mathsf{f}}) \neq d/\tilde{r}$.
6. Compute the unique polynomial $\tilde{\mathsf{F}} \in \tilde{\mathbb{F}}[x, y]$ such that $\tilde{\mathsf{F}}(0, y) = \tilde{\mathsf{f}}(y)$ and $\tilde{\mathsf{F}}$ divides F modulo $(x^{d/\tilde{r}+1})$. If $\deg(\tilde{\mathsf{F}}) > d/\tilde{r}$ then stop the computation.
7. Let ψ denote the canonical image of y in $\tilde{\mathbb{F}}[[x]]/(x^{d+1})[y]/(\tilde{\mathsf{F}}(x, y))$ and call $\mathsf{Evaluation}(F, \psi)$; Stop the execution if the result is not zero. Otherwise, return $\mathsf{F} := \tilde{\mathsf{F}}$.

Proposition 9. *Algorithm* $\mathsf{ProbabilisticGenericFactor}$ *either stops prematurely or returns a correct answer. Its complexity belongs to*

$$\mathcal{O}\left(d^{(\omega+3)/2} + \sqrt{d}\mathsf{M}(d)(\mathsf{M}(\tilde{r})\mathsf{M}(d/\tilde{r}) + d\log(d)) \right).$$

In addition, if $\tilde{r} = r$ then there exists a nonzero polynomial $S \in \bar{\mathbb{K}}[C_{1:d}]$ of total degree at most $d(d-1)/2$ such that, for all $\tilde{c}_{1:d} \in \mathcal{U}(S)$, this algorithm called with $\tilde{c}_{1:r}$ returns a correct answer.

Démonstration. If $\tilde{r} = r$ then we take the polynomial S of Lemma 6 but we canonically embed it in $\mathbb{K}[C_{1:d}]$. By construction, if $\tilde{c}_{1:d} \in \mathcal{U}(S)$ then $\tilde{c}_{1:r}$ separates the residue basis $\rho_{1:r}$ associated to $\nu_{1:r}$. In this case, we are going to show that the algorithm is correct.

We let $c := \tilde{c}$ and, since $\hat{\mathfrak{f}}(y) = \hat{\mathfrak{F}}(0, y)$, one has :

$$h(y) = c_1 g_1 + \cdots + c_r g_r.$$

In step (3), one has $\tilde{Q} = Q$, from Lemma 7. The correctness of Step (4) follows from Corollary 3, in particular $\tilde{\mathbb{F}}$ is a field. In Step (6), one necessarily has $\tilde{\mathsf{F}} = \gcd(F, \tilde{\alpha}\frac{\partial F}{\partial y} - c(G_{1:r}))$, thanks to Lemma 9.

In Step (7), the test is equivalent to computing the remainder in the division of F by $\tilde{\mathsf{F}}$ at precision (x^{d+1}) : this remainder is in (x^{d+1}) if and only if $\tilde{\mathsf{F}}$ divides F. Of course, since $\tilde{r} = r$, this remainder is zero.

Let us now examine what happens if $\tilde{r} > r$. Steps (1) to (4) are well defined and if $\tilde{q}$ has not degree $\tilde{r}$ this contradicts Corollary 3 hence we stop the computation. If we reach Step (5) then we can not be sure that $\tilde{\mathbb{F}}$ is actually a field, thus the execution may be stopped. Step (6) is well defined and produces and candidate $\tilde{\mathsf{F}}$ with $\deg_y(\mathsf{F}) = d/\tilde{r} < d/r$. In other words, the degree in y of $\tilde{\mathsf{F}}$ is striclty less than the degree of the true factor. Thus the call to $\mathsf{Evaluation}$ can not return 0.

The complexity analysis of the first steps is almost straightforward : Step (1) costs $\mathcal{O}(d\mathsf{M}(d))$, Step (2) costs $\mathcal{O}(d^2)$. Step (3) costs $\mathcal{O}(d\mathsf{M}(d)\log(d))$ if we use fast interpolation of $\tilde{Q}$: that is we compute $Q(0), \ldots, Q(d)$ and then interpolate Q. Step (4) costs $\mathcal{O}(\mathsf{M}(d)\log(d))$. Each field operation in $\tilde{\mathbb{F}}$ performs $\mathcal{O}(\mathsf{M}(\tilde{r})\log(\tilde{r}))$ operations in $\mathbb{K}$: the inversion of an element is handled by an extended gcd computation and the execution is stopped if one encounters a nonzero element which is not invertible. This way, Step (5) can be done within $\mathcal{O}(\mathsf{M}(\tilde{r})\log(\tilde{r})\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$. Using classical Hensel lifting [vzGG03, Theorem 15.11] allows us to perform Step (6) within $\mathcal{O}(\mathsf{M}(\tilde{r})\log(\tilde{r})\mathsf{M}(d)\mathsf{M}(d/\tilde{r}))$. The complexity of the last step has already been given in Proposition 3. The total cost directly comes from summing up these intermediate costs. □

Note that using the classical fast division algorithm to test the divisibility of F by $\tilde{\mathsf{F}}$ costs $\mathcal{O}(\mathsf{M}(\tilde{r})\mathsf{M}(d)\mathsf{M}(d))$ operations in $\mathbb{K}$. In consequence, the test of Step (7) yields a sensible gain but is part of the bottleneck of the whole factorization algorithm at the same level as the lifting of ϕ.

C.5.2 Main Probabilistic Algorithm

Putting together all the sub-algorithms described so far, the complete probabilistic absolute factorization proceeds as follows :

Algorithm ProbabilisticAbsoluteFactorization

Input : an irreducible polynomial F in $\mathbb{K}[x,y]$ of degree d such that Hypotheses (C) and (H) hold, $(u,v,a,b,c_{1:d}) \in \mathbb{K}^{d+4}$.

Output : the generic factor F of F.

1. Replace F by $F(x+uy+v,y)$ by means of the algorithm underlying the proof of Proposition 2. If this new F does not satisfy Hypothesis (H) then stop the execution.
2. Call the procedure Lifting in order to compute ϕ at precision (x^τ), where $\tau := 2d+1$;
3. Let $b := 0$ and call the procedure ProbabilisticSolve using (a,b) in order to get a basis $\nu_{1:\tilde{r}}$ of $\{\ell_{1:d} \in \mathbb{K}^d \mid P_\tau^{a,b}\}$;
4. Compute and return F by calling the ProbabilisticGenericFactor with arguments F, $\nu_{1:\tilde{r}}$ and $c_{1:\tilde{r}}$;
5. Return $\mathsf{F}(x-uy-v,y)$.

Proposition 10. *Algorithm ProbabilisticAbsoluteFactorization either stops prematurely or returns a correct answer. Its complexity belongs to*

$$\mathcal{O}\left(d^{(\omega+3)/2} + \sqrt{d}\mathsf{M}(d)(\mathsf{M}(d)^2/d + d\log(d))\right).$$

In addition, there exists a nonzero polynomial $P \in \mathbb{K}[U]$ of degree at most d such that, for any $u \in \mathcal{U}(P)$, there exists a nonzero polynomial $Q_u \in \mathbb{K}[V]$ of degree at most $d(d-1)$ such that, for any $v \in \mathcal{U}(Q_u)$, there exists a nonzero polynomial $R_{u,v} \in \mathbb{K}[A]$ of degree at most $d(d-1)$ such that, for any $a \in \mathcal{U}(R_{u,v})$, there exists a nonzero polynomial $S_{u,v,a} \in \bar{\mathbb{K}}[C_{1:d}]$ of total degree at most $d(d-1)/2$ such that, for any $c_{1:d} \in \mathcal{U}(S_{u,v,a})$, this algorithm called with $(u,v,a,c_1,\dots,c_d)$ returns a correct answer.

Démonstration. If the computations in ProbabilisticAbsoluteFactorization do not stop prematurely then the result is necessarily correct according to successively using Propositions 2, 4 and 7. According to our assumption on M one has $\mathsf{M}(\tilde{r})/\tilde{r} \le \mathsf{M}(d)/d$ and $\mathsf{M}(d/\tilde{r})/(d/\tilde{r}) \le \mathsf{M}(d)/d$, from which we deduce

$$\mathsf{M}(\tilde{r})\mathsf{M}(d/\tilde{r}) \le \mathsf{M}(d)^2/d. \tag{C.15}$$

The claimed complexity follows from these propositions and this inequality. □

Using fast polynomial multiplication, we are now ready to complete :

Proof of Theorem 2. The proof follows from the previous proposition when taking $\mathsf{M}(n) \in \tilde{\mathcal{O}}(n)$. □

Note that the complexity obtained this way for Theorem 2 drops into $\mathcal{O}(d^{(\omega+3)/2})$ when $\omega > 2$.

C.6 Deterministic Algorithm

In order to complete a deterministic factorization algorithm, it remains to provide a deterministic process for computing a separating form c. The method we propose below actually computes the factorization in the same time, so that the algorithm of this section is independent of the one of the previous section. We carry on with the notation of Section C.4. Since we follow a deterministic strategy, we can assume that we are given a basis $\nu_{1:r}$ of $\pi(L_\infty)$ and we still write $\rho_{1:r}$ for the associated residue basis and $g_{1:r}$ for the polynomials introduced in Section C.4.1.

The main idea of our strategy relies on computing a tower of monogen algebraic extensions between $\mathbb{K}$ and $\mathbb{F}$. At each level of this tower we perform an elementary effective primitive element theorem in two variables in order to finish with a primitive element of $\mathbb{F}$. Remark that, in case when r is a prime number, this computation is simplified for there cannot exist a subfield between $\mathbb{K}$ and $\mathbb{F}$.

C.6.1 Improving the Separation

In this subsection we assume that we are given two linear forms c_1 and c_2 defined on $\mathbb{K}^r$ and we let

$$q_1(z_1) := \prod_{\beta \in \{c_1(\rho_i) | i \in \{1,\ldots,r\}\}} (z_1 - \beta), \quad s_1 := \deg(q_1), \quad \mathsf{f}_1 := \gcd(f, \alpha_1 f' - c_1(g_{1:r})),$$

where α_1 represents the canonical image of z_1 in $\mathbb{F}_1 := \mathbb{K}[z_1]/(q_1(z_1))$. From Corollary 3, we know that $\mathbb{F}_1$ is a field. We denote by s_2 the cardinality of $\{c_2(\rho_i) \mid i \in \{1,\ldots,r\}\}$ and by s_3 the cardinality of the set of values $\{(c_1(\rho_i), c_2(\rho_i)) \mid i \in \{1,\ldots,r\}\}$. If s_3 is strictly greater that $\max(s_1, s_2)$ then we say that c_1 and c_2 are *independent*. In this case we aim to compute a third form $c_3 := \gamma c_1 + c_2$, with $\gamma \in \mathbb{K}$, such that

$$q_3(z_3) := \prod_{\beta \in \{c_3(\rho_i) | i \in \{1,\ldots,r\}\}} (z_3 - \beta)$$

has degree s_3. The next subsection will be devoted to the computation an independent forms. In a similar way as in Section C.4.3 we introduce

$$Q_3(z_3) := \left(\prod_{i=1}^{r} (z - c_3(\rho_i)) \right)^{d/r}.$$

Let $P(z_2) := \mathrm{Res}(\mathsf{f}_1, z_2 f' - c_2(g_{1:r})) \in \mathbb{F}_1[z_2]$ and let p denote the monic squarefree part of P. With a slight abuse of notation, we still write $P(z_1, z_2)$ (resp. $p(z_1, z_2)$) for the canonical preimage of P (resp. p) in $\mathbb{K}[z_1, z_2]$. The following lemma relates the computation of a suitable value for γ to finding a primitive element $\gamma z_1 + z_2$ of the algebra $\mathbb{K}[z_1, z_2]/(q_1(z_1), p(z_1, z_2))$.

Lemma 10. *According to the above notation and assumption one has*

$$\delta Q_3(z_3) = \mathrm{Res}_{z_1}(q_1(z_1), P(z_1, z_3 - \gamma z_1))$$

and $s_3 = s_1 \deg(p)$. *Thus the polynomial* q_3 *equals the monic squarefree part of* $\tilde{q}_3 := \mathrm{Res}_{z_1}(q_1(z_1), p(z_1, z_3 - \gamma z_1))$. *In addition, there exists a nonzero polynomial* $T \in \bar{\mathbb{K}}[\Gamma]$ *of degree* $s_3(s_3-1)/2$ *such that, for any* $\gamma \in \mathcal{U}(T)$, *the polynomial* $\tilde{q}_3$ *is squarefree.*

Démonstration. By construction, one has $\alpha_1 f' - c_1(g_{1:r}) \bmod \mathsf{f}_1 = 0$. On the other hand, we have that $f = \prod_\chi \chi(\mathsf{f}_1)$, where χ runs over all the $\mathbb{K}$-embeddings of $\mathbb{F}_1$ into $\bar{\mathbb{K}}$. Therefore, using Lemma 7 and the multiplicative property of the resultant, we deduce the following equalities :

$$\begin{aligned}
\delta Q_3 := \mathrm{Res}(f, z_3 f' - c_3(g_{1:r})) &= \prod_\chi \chi(\mathrm{Res}(\mathsf{f}_1, z_3 f' - c_3(g_{1:r}))) \\
&= \prod_\chi \chi(\mathrm{Res}(\mathsf{f}_1, (z_3 - \gamma\alpha_1) f' - c_2(g_{1:r}))) \\
&= \prod_\chi \chi(P(z_3 - \gamma\alpha_1)) \\
&= \mathrm{Res}_{z_1}(q_1(z_1), P(z_1, z_3 - \gamma z_1)),
\end{aligned}$$

Thanks to Lemma 8, it follows that q_3 is the squarefree part of $\tilde{q}_3$.

Let $\beta_{1:s_1}$ denote the roots of q_1 in $\bar{\mathbb{K}}$. Since each i, $p(\beta_i, z_2)$ is squarefree, we can write $\beta_{i,1}, \ldots, \beta_{i,\deg(p)}$ its distinct roots in $\bar{\mathbb{K}}$. By definition of s_3 we directly deduce $s_3 = s_1 \deg(p)$ and naturally introduce

$$T(\Gamma) := \prod_{(i,j)\neq(i',j')} ((\beta_i - \beta_{i'})\Gamma + \beta_{i,j} - \beta_{i',j'}) \,.$$

Of course T has the claimed degree and if $\gamma \in \mathcal{U}(T)$ then $\tilde{q}_3$ has degree s_3 and is squarefree since :

$$\tilde{q}_3(z_3) = \prod_{i=1}^{s_1} \prod_{j=1}^{\deg(p)} (z_3 - (\beta_i \gamma + \beta_{i,j})).$$

□

If $\gamma z_1 + z_2$ is a primitive element of the algebra $\mathbb{K}[z_1, z_2]/(q_1(z_1), p(z_1, z_2))$ then we have

$$\mathbb{K}[z_1, z_2]/(q_1(z_1), p(z_1, z_2)) = \mathbb{K}[z_3]/(q_3(z_3)) =: \mathbb{F}_3$$

From Corollary 3, we deduce that $\mathbb{F}_3$ is a field generated by the image α_3 of z_3 and thus that

$$\mathsf{f}_3 := \gcd(f, \alpha_3 f' - c_3(g_{1:r}))$$

is well defined.

Algorithm $\mathsf{Separate}$

Input : f, $c_1(g_{1:r})$, q_1, f_1 and $c_2(g_{1:r})$ as introduced above.

Output : $c_3(g_{1:r})$, q_3 and f_3 such that $\deg(q_3) = s_3$, as defined above.

1. Compute $P(z_2) := \mathrm{Res}(\mathsf{f}_1, z_2 f' - c_2(g_{1:r}))$;
2. Compute the monic squarefree part p of P ;
3. $s_3 := \deg(q_1)\deg(p)$;
4. For $\gamma := 0$ to $s_3(s_3-1)/2$ do
   (a) $\tilde{q}_3(z_3) := \mathrm{Res}_{z_1}(q(z_1), p(z_1, z_3 - \gamma z_1))$;
   (b) If $\tilde{q}_3$ is squarefree then break ;
5. Let $q_3 := \tilde{q}_3$ and $c_3(g_{1:r}) := \gamma c_1(g_{1:r}) + c_2(g_{1:r})$;
6. Let $\mathbb{F}_3 := \mathbb{K}[z_3]/(q_3(z_3))$ and let α_3 denote the canonical image of z_3 in $\mathbb{F}_3$;
7. Compute $\mathsf{f}_3 := \gcd(f, \alpha_3 f' - c_3(g_{1:r}))$;
8. Return $c_3(g_{1:r})$, q_3, f_3.

Proposition 11. *Under Hypotheses (C) and (H), Algorithm* Separate *is correct and performs*

$$\mathcal{O}(\mathsf{M}(d)^2\log(d)^2 + s_3^3\mathsf{M}(s_3)\log(s_3))$$

operations in $\mathbb{K}$.

Démonstration. The correctness directly follows from the previous lemma since the polynomial T defined therein can not have more than $s_3(s_3-1)/2$ zeros.

Recall that f_1 has degree d/s_1. In Step (1) we start with computing $f' \bmod \mathsf{f}_1$ and $c_2(g_{1:r}) \bmod \mathsf{f}_1$ in time $\mathcal{O}(\mathsf{M}(s_1)\mathsf{M}(d))$. Thanks to [vzGG03, Corollary 11.18], the computation of P can be done within $\mathcal{O}(d/s_1\mathsf{M}(d/s_1)\log(d/s_1))$ operations in $\mathbb{F}_1$. We deduce that the total cost of this step belongs to

$$\mathcal{O}(\mathsf{M}(s_1)\mathsf{M}(d) + d/s_1\mathsf{M}(s_1)\log(s_1)\mathsf{M}(d/s_1)\log(d/s_1)) \subseteq \mathcal{O}(\mathsf{M}(d)^2\log(d)^2),$$

where the latter inclusion is a consequence of Inequality (C.15).

Step (2) costs $\mathcal{O}(\mathsf{M}(s_1)\log(s_1)\mathsf{M}(d/s_1)\log(d/s_1)) \subseteq \mathcal{O}(\mathsf{M}(d)^2\log(d)^2)$. Each Step (4a) can be done as follows. Let s represent $\deg(p)$, we first precompute $p_\gamma := p(z_3 + \gamma\alpha_1)$ within $\mathcal{O}(\mathsf{M}(s_1)\mathsf{M}(s))$ operations in $\mathbb{K}$. Since $s_3 = s_1 s$, we deduce from Inequality (C.15) that $\mathcal{O}(\mathsf{M}(s_1)\mathsf{M}(s)) \subseteq \mathcal{O}(\mathsf{M}(s_3)^2/s_3)$, and we obtain $\mathcal{O}(\mathsf{M}(s_3)^2/s_2) \subseteq \mathcal{O}(s_3\mathsf{M}(s_3))$. For each value $i \in \{0, \ldots, s_3\}$ the polynomial $p_\gamma(i)$ can be computed within $\mathcal{O}(ss_1) = \mathcal{O}(s_3)$ operations in $\mathbb{K}$. Then the computation of $\tilde{q}_3(i)$ costs $\mathcal{O}(\mathsf{M}(s_1)\log(s_1))$. For each γ, we deduce that the computation of $\tilde{q}_3$ costs

$$\mathcal{O}(s_3\mathsf{M}(s_3) + s_3(s_3 + \mathsf{M}(s_1)\log(s_1)) + \mathsf{M}(s_3)\log(s_3)) \subseteq \mathcal{O}(s_3\mathsf{M}(s_3)\log(s_3)).$$

Recall the squarefree part of $\tilde{q}_3$ requires $\mathcal{O}(\mathsf{M}(s_3)\log(s_3))$, so that, finally Step (4) costs $\mathcal{O}(s_3^3\mathsf{M}(s_3)\log(s_3))$.

Lastly Step (7) costs $\mathcal{O}(\mathsf{M}(s_3)\log(s_3)\mathsf{M}(d)\log(d))$ and summing up these complexities concludes the proof. □

We can avoid Step (4) in the above algorithm when $\mathbb{K}$ has caracteristic 0. In fact, it suffices to choose γ big enough (that is to say such that it is not a root of T). With Mignotte's bound we can get a lower bound for these kinds of values. The drawback of this method is that we increase the size of the coefficients of $\tilde{q}_3$.

C.6.2 Finding Independent Forms

For any given non separating form c_1, we provide a way for computing a form independent from c_1. It is easy to observe that if c_1 is separating then $g_i \bmod \mathsf{f}_1$ is proportional to $f' \bmod \mathsf{f}_1$, for all $i \in \{1, \ldots, r\}$. On the contrary, we show that if c_1 is non separating then there exists an index i such that $g_i \bmod \mathsf{f}_1$ is not proportional to $f' \bmod \mathsf{f}_1$. In this case, t_i is independent of c_1 :

Lemma 11. *If $s_1 < r$ then there exist indexes i such that g_i mod f_1 is not proportional to f' mod f_1. For such indexes, t_i is independent of c_1.*

Démonstration. Since $s_1 < r$, f_1 is reducible : let us assume that f_1 and f_2 are factors of f_1. By construction, we have $c_1(\rho_1) = c_1(\rho_2)$. Since $\rho_1 \neq \rho_2$ there exist indexes i such that $t_i(\rho_1) \neq t_i(\rho_2)$: for such i, $g_i \bmod \mathsf{f}_1$ is not proportional to $f' \bmod \mathsf{f}_1$.

If we take i such that $g_i \bmod \mathsf{f}_1$ is not proportional to $f' \bmod \mathsf{f}_1$ then there exists two factors f_{i_1} and f_{i_2} such that $t_i(\rho_{i_1}) \neq t_i(\rho_{i_2})$, hence t_i is independent of c_1. □

In terms of pseudo-code, we easily deduce the following algorithm :

Algorithm $\mathsf{FindIndependentForm}$

Input : $g_{1:r}$, f_1, with $s_1 < r$, as defined above.

Output : $c_2(g_{1:r})$ such that c_2 is independent from c_1.

1. Find $i \in \{1, \ldots, r\}$ such that $g_i \bmod \mathsf{f}_1$ is not proportional to $f' \bmod \mathsf{f}_1$;
2. Return g_i.

Proposition 12. *Algorithm $\mathsf{FindIndependentForm}$ is correct and performs $\mathcal{O}(r\mathsf{M}(s_1)\mathsf{M}(d))$ operations in $\mathbb{K}$.*

Démonstration. The correctness directly follows from the previous lemma and the complexity is immediate. □

C.6.3 Deterministic Generic Factor

Putting together the previous sub-routines, the deterministic algorithm for computing a separating form proceeds as follows.

Algorithm $\mathsf{DeterministicGenericFactor}$

Input : an irreducible polynomial F of degree d such that Hypotheses (C) and (H) hold, and a basis $\nu_{1:r}$ of $\pi(L_\infty)$.

Output : the generic factor F of F.

1. Compute $\hat{\mathfrak{f}}(y)$ as the quotient of $f(y)$ by $y - \varphi$ in $\mathbb{A}[y]$;
2. Build the $d \times d$ matrix M defined by $M_{i,j} := \operatorname{coeff}(\hat{\mathfrak{f}}(y), \varphi^{i-1} y^{j-1})$ and compute the matrix product $N := \nu M$, where ν is seen as $r \times d$ matrix $(\nu_{i,j})$. Then build all the g_i from
$$g_i := \sum_{j=1}^{d} N_{i,j} y^{j-1};$$
3. $h_3 := 0$; $q_3 := z$; $\mathsf{f}_3 := f$;
4. While $\deg(q_3) < r$ do
   (a) $h_1 := h_3$; $q_1 := q_3$; $\mathsf{f}_1 := \mathsf{f}_3$;
   (b) $h_2 := \mathsf{FindIndependentForm}(g_{1:r}, \mathsf{f}_1)$;
   (c) $h_3, q_3, \mathsf{f}_3 := \mathsf{Separate}(f, g_{1:r}, h_1, q_1, \mathsf{f}_1, h_2)$;
5. $q := q_3$; $\mathsf{f} := \mathsf{f}_3$; $\mathbb{F} := \mathbb{F}_3$; $\alpha := \alpha_3$;
6. By means of Hensel lifting, compute the unique polynomial $\mathsf{F} \in \mathbb{K}[\alpha][x, y]$ of degree d/r in x such that $\mathsf{F}(0, y) = \mathsf{f}(y)$ and F divides F modulo $(x^{d/r+1})$.

Proposition 13. *Algorithm* DeterministicGenericFactor *is correct and performs* $\mathcal{O}(\mathcal{O}(d^3 \mathsf{M}(d) \log(d))$ *operations in* $\mathbb{K}$.

Démonstration. The correctness follows from combining Proportions 11 and 12. For the complexities, Step (1) costs $\mathcal{O}(d\mathsf{M}(d))$ and Step (2) costs $\mathcal{O}(d^2 r^{\omega-2})$. According to Proposition 12, Step (4b) costs $\mathcal{O}(r\mathsf{M}(s_1)\mathsf{M}(d))$. According to Proposition 11, Step (4c) costs $\mathcal{O}(\mathsf{M}(d)^2 \log(d)^2 + s_3^3 \mathsf{M}(s_3) \log(s_3))$. Using the super-additivity of M (C.5) we deduce that the total cost of Step (4) belongs to $\mathcal{O}(\mathsf{M}(d)^2 \log(d)^3 + r^3 \mathsf{M}(r) \log(r) + r\mathsf{M}(r)\mathsf{M}(d))$. The complexity of the last step belongs to $\mathcal{O}(\mathsf{M}(r) \log(r) \mathsf{M}(d) \mathsf{M}(d/r))$. Summing up these complexities yields the claimed bound. The last inclusion follows from using the fact the M is at most quadratic. □

C.6.4 Main Algorithm

We are now coming to the main algorithm.

Algorithm DeterministicAbsoluteFactorization

Input : an irreducible polynomial F in $\mathbb{K}[x, y]$ of degree d satisfying Hypothesis (C).

Output : the generic factor F of F ;

1. Use the algorithm described in the proof of Proposition 1 with F in order to compute $(u, v) \in \mathbb{K}^2$ such that $F(x + uy + v, y)$ satisfies Hypothesis (H). Now replace F by $F(x + uy + v, y)$.

2. Let $\sigma = 2d$ and compute ϕ at precision (x^{σ}) as $\mathsf{Lifting}(F, \sigma)$;
3. Compute $\nu_{1:r} := \mathsf{DeterministicSolve}(F, \phi)$;
4. Compute $\mathsf{F} := \mathsf{DeterministicGenericFactor}(F, \nu_{1:r})$;
5. Return $\mathsf{F}(x - uy - v, y)$.

Proposition 14. *Algorithm DeterministicAbsoluteFactorization is correct and performs*

$$\mathcal{O}(d^3 \mathsf{M}(d) \log(d))$$

operations in $\mathbb{K}$.

Démonstration. The correctness and complexity follow from combining Propositions 1, 4, 6 and 13, and Lemma 2 for the last step. □

We are now able to achieve :

Proof of Theorem 1. It suffices to take fast polynomial multiplication, that is $\mathsf{M}(n) \in \tilde{\mathcal{O}}(n)$, in the previous proposition. □

Acknowledgments

We wish to thank Manuel Bronstein for pointing us out applications of absolute factorization in the field of effective linear differential algebra.

Bibliographie

[Abb88] J. A. Abbott. *Factorization of polynomials over algebraic function fields.* PhD thesis, Univ. Bath, England, 1988.

[ACGH85] E. Arbarello, M. Cornalba, P. A. Griffiths, and J. Harris. *Geometry of algebraic curves. Vol. I*, volume 267 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 1985.

[AGL04] F. Abu Salem, S. Gao, and A. G. B. Lauder. Factoring polynomials via polytopes. In *Proceedings of ISSAC 2004*, pages 4–11. ACM Press, 2004.

[AL04] Jounaïdi Abdeljaoued and Henri Lombardi. *Méthodes matricielles : introduction à la complexité algébrique*, volume 42 of *Mathématiques & Applications [Mathematics & Applications]*. Springer-Verlag, Berlin, 2004.

[All01] Bill Allombert. *Théorie de Galois effective pour les corps de nombres et les corps finis.* PhD thesis, Université de Bordeaux, 2001.

[BCGW93] Chanderjit Bajaj, John Canny, Thomas Garrity, and Joe Warren. Factoring rational polynomials over the complex numbers. *SIAM J. Comput.*, 22(2) :318–331, 1993.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory.* Springer, 1997.

[Ber98] Laurent Bernardin. On bivariate Hensel lifting and its parallelization. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (Rostock)*, pages 96–100, New York, 1998. ACM.

[BHKS04] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel. Factoring polynomials over global fields. Manuscript, October 2004.

[BLS03] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *Proceedings of ISSAC 2003*, pages 37–44. ACM, 2003.

[BLS+04] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt. Complexity Issues in Bivariate Polynomial Factorization. In *Proceedings of ISSAC 2004*, pages 42–49. ACM, 2004.

[BP70] Åke Björck and Victor Pereyra. Solution of Vandermonde systems of equations. *Math. Comp.*, 24 :893–903, 1970.

[BP94] D. Bini and V. Y. Pan. *Polynomial and matrix computations. Vol. 1*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1994. Fundamental algorithms.

[Bro97] Manuel Bronstein. *Symbolic integration. I*, volume 1 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 1997. Transcendental functions, With a foreword by B. F. Caviness.

[Bro01] M. Bronstein. Computer algebra algorithms for linear ordinary differential and difference equations. In *European Congress of Mathematics, Vol. II (Barcelona, 2000)*, volume 202 of *Progr. Math.*, pages 105–119. Birkhäuser, Basel, 2001.

[BS04] A. Bostan and É. Schost. Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity (Festschrift for Arnold Schönhage)*, 2004. To appear.

[BT03] M. Bronstein and B. M. Trager. A reduction for regular differential systems, 2003. Manuscript.

[Car61] Henri Cartan. *Théorie élémentaire des fonctions analytiques d'une ou plusieurs variables complexes*. Avec le concours de Reiji Takahashi. Enseignement des Sciences. Hermann, Paris, 1961.

[CG03] Guillaume Chèze and André Galligo. From an approximate to an exact factorization. Submitted to JSC, 2003.

[CG05] Guillaume Chèze and André Galligo. Four lessons on polynomial absolute factorization, due to appear in April 2005. To appear in Solving Polynomials Equations : Foundations, Algorithms and Applications.

[CGKW02] R.M. Corless, A. Galligo, I.S. Kotsireas, and S.M. Watt. A geometric-numeric algorithm for factoring multivariate polynomials. In T. Mora, editor, *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (ISSAC 2002)*, pages 37–45. ACM, 2002.

[Chi84] A. L. Chistov. An algorithm of polynomial complexity for factoring polynomials, and determination of the components of a variety in a subexponential time. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 137 :124–188, 1984. Theory of the complexity of computations, II.

[Chi95] Lindsay N. Childs. *A concrete introduction to higher algebra*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1995.

[Chè04a] Guillaume Chèze. Absolute irreducibility, Newton polytopes and modular computations. Manuscript in preparation, 2004.

[Chè04b] Guillaume Chèze. Absolute polynomial factorization in several variables and the knapsack problem. In *Proceedings of ISSAC 2004*, pages 87–94, 2004.

[CL] Guillaume Chèze and Grégoire Lecerf. Lifting and recombination techniques for absolute factorization. Preprint.

[CL04] G. Chèze and G. Lecerf. Partial absolute factorizations and absolute irreducibility tests. Manuscript in preparation, 2004.

[CLO97] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1997. An introduction to computational algebraic geometry and commutative algebra.

[Coh93] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[CSTU02] Olivier Cormier, Michael F. Singer, Barry M. Trager, and Felix Ulmer. Linear differential operators for polynomial equations. *J. Symbolic Comput.*, 34(5) :355–398, 2002.

[CZ81] David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Math. Comp.*, 36(154) :587–592, 1981.

[Del01] S. Dellière. On the links between triangular sets and dynamic constructible closure. *J. Pure Appl. Algebra*, 163(1) :49–68, 2001.

[Dim92] Alexandru Dimca. *Singularities and topology of hypersurfaces*. Universitext. Springer-Verlag, New York, 1992.

[DS04] C. D'Andrea and M. Sombra. The Cayley-Menger determinant is irreducible for $n \geq 3$. Submitted, 7pp., 2004.

[DT89] Roberto Dvornicich and Carlo Traverso. Newton symmetric functions and the arithmetic of algebraically closed fields. In *Applied algebra, algebraic algorithms and error-correcting codes (Menorca, 1987)*, volume 356 of *Lecture Notes in Comput. Sci.*, pages 216–224. Springer, Berlin, 1989.

[Duv83] Dominique Duval. Une méthode géométrique de factorisation des polynômes en deux indéterminées. Technical report, Institut Fourier, Université de Grenoble 1, 1983.

[Duv91] Dominique Duval. Absolute factorization of polynomials : a geometric approach. *SIAM J. Comput.*, 20(1) :1–21, 1991.

[Duv95] D. Duval. Évaluation dynamique et clôture algébrique en Axiom. *Journal of pure and Applied Algebra*, 99 :267–295, 1995.

[Eis95] David Eisenbud. *Commutative algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. With a view toward algebraic geometry.

[EK95] Alan Edelman and Eric Kostlan. How many zeros of a random polynomial are real ? *Bull. Amer. Math. Soc. (N.S.)*, 32(1) :1–37, 1995.

[Esc97] Jean-Pierre Escofier. *Théorie de Galois.* Enseignement des Mathématiques. [The Teaching of Mathematics]. Masson, Paris, 1997. Cours avec exercices corrigés. [Course with exercises and solutions].

[FGP01] P. Flajolet, X. Gourdon, and D. Panario. The complete analysis of a polynomial factorization algorithm over finite fields. *J. Algorithms*, 40(1) :37–81, 2001.

[Gal99] André Galligo. Real factorization of multivariate polynomials with integer coefficients. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI)*, 258(Teor. Predst. Din. Sist. Komb. i Algoritm. Metody. 4) :60–70, 355, 1999.

[Gao01] Shuhong Gao. Absolute irreducibility of polynomials via Newton polytopes. *J. Algebra*, 237(2) :501–520, 2001.

[Gao03] Shuhong Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72(242) :801–822, 2003.

[Gau90] Walter Gautschi. How (un)stable are Vandermonde systems ? In *Asymptotic and computational analysis (Winnipeg, MB, 1989)*, volume 124 of *Lecture Notes in Pure and Appl. Math.*, pages 193–210. Dekker, New York, 1990.

[GC84] D. Yu. Grigor′ev and A. L. Chistov. Fast factorization of polynomials into irreducible ones and the solution of systems of algebraic equations. *Dokl. Akad. Nauk SSSR*, 275(6) :1302–1306, 1984.

[GG94] S. Gao and J. von zur Gathen. Berlekamp's and Niederreiter's polynomial factorization algorithms. In *Finite fields : theory, applications, and algorithms (Las Vegas, NV, 1993)*, volume 168 of *Contemp. Math.*, pages 101–116. Amer. Math. Soc., Providence, RI, 1994.

[GKL04] S. Gao, E. Kaltofen, and A. Lauder. Deterministic distinct degree factorization for polynomials over finite fields. *J. Symbolic Comput.*, 38(6) :1461–1470, 2004.

[GKM$^+$04] Shuhong Gao, Erich Kaltofen, John P. May, Zhengfeng Yang, and Lihong Zhi. Approximate factorization of multivariate polynomials via differential equations. In *Proceedings of ISSAC 2004*, pages 167–174, 2004.

[GL] Shuhong Gao and Alan G.B. Lauder. Fast absolute irreducibility testing via Newton polytopes. Preprint 2004.

[GL01] S. Gao and A. G. B. Lauder. Decomposition of polytopes and polynomials. *Discrete Comput. Geom.*, 26(1) :89–104, 2001.

[GL02] S. Gao and A. G. B. Lauder. Hensel lifting and bivariate polynomial factorisation over finite fields. *Math. Comp.*, 71(240) :1663–1676, 2002.

[GLSY] M. Giusti, G. Lecerf, B. Salvy, and J.C. Yakoubsohn. On Location and approximation of clusters of zeroes : case of embedding dimension one. Preprint.

[GM83] Mark Goresky and Robert MacPherson. Stratified Morse theory. In *Singularities, Part 1 (Arcata, Calif., 1981)*, volume 40 of *Proc. Sympos. Pure Math.*, pages 517–533. Amer. Math. Soc., Providence, R.I., 1983.

[Gou97] Fernando Q. Gouvêa. *p-adic numbers*. Universitext. Springer-Verlag, Berlin, second edition, 1997. An introduction.

[GR02] André Galligo and David Rupprecht. Irreducible decomposition of curves. *J. Symbolic Comput.*, 33(5) :661–677, 2002. Computer algebra (London, ON, 2001).

[GR03] Shuhong Gao and Virgínia M. Rodrigues. Irreducibility of polynomials modulo p via Newton polytopes. *J. Number Theory*, 101(1) :32–47, 2003.

[GW97] André Galligo and Stephen Watt. A numerical absolute primality test for bivariate polynomials. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI)*, pages 217–224, New York, 1997. ACM.

[Har77] Robin Hartshorne. *Algebraic geometry*. Springer-Verlag, New York, 1977. Graduate Texts in Mathematics, No. 52.

[Har80] Joe Harris. The genus of a space curve. *Math. Ann.*, 249 :191–204, 1980.

[Hig02] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2002.

[Hoh97] Jean-Christophe Hohl. Massively parallel search for linear factors in polynomials with many variables. *Appl. Math. Comput.*, 85(2-3) :227–243, 1997.

[HRUW99] M. van Hoeij, J.-F. Ragot, F. Ulmer, and J.-A. Weil. Liouvillian solutions of linear differential equations of order three and higher. *J. Symbolic Comput.*, 28(4-5) :589–609, 1999. Differential algebra and differential equations.

[HS81] Joos Heintz and Malte Sieveking. Absolute primality of polynomials is decidable in random polynomial time in the number of variables. In *Automata, languages and programming (Akko, 1981)*, volume 115 of *Lecture Notes in Comput. Sci.*, pages 16–28. Springer, Berlin, 1981.

[HW79] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. The Clarendon Press Oxford University Press, New York, fifth edition, 1979.

[Jou83] Jean-Pierre Jouanolou. *Théorèmes de Bertini et applications*, volume 42 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1983.

[Kac43] M. Kac. On the average number of real roots of a random algebraic equation. *Bull. Amer. Math. Soc.*, 49 :314–320, 1943.

[Kal85a] Erich Kaltofen. Effective Hilbert irreducibility. *Inform. and Control*, 66(3) :123–137, 1985.

[Kal85b] Erich Kaltofen. Fast parallel absolute irreducibility testing. *J. Symbolic Comput.*, 1(1) :57–67, 1985.

[Kal85c] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2) :469–489, 1985.

[Kal90] Erich Kaltofen. Polynomial factorization 1982–1986. In *Computers in mathematics (Stanford, CA, 1986)*, volume 125 of *Lecture Notes in Pure and Appl. Math.*, pages 285–309. Dekker, New York, 1990.

[Kal92] E. Kaltofen. Polynomial factorization 1987-1991. In I. Simon, editor, *Proc. LATIN '92*, volume 583, pages 294–313. Springer-Verlag, 1992.

[Kal95] Erich Kaltofen. Effective Noether irreducibility forms and applications. *J. Comput. System Sci.*, 50(2) :274–295, 1995. 23rd Symposium on the Theory of Computing (New Orleans, LA, 1991).

[Kal03] E. Kaltofen. Polynomial factorization : a success story. In *ISSAC'03*, pages 3–4. ACM Press, 2003.

[Kle98] Steven L. Kleiman. Bertini and his two fundamental theorems. *Rend. Circ. Mat. Palermo (2) Suppl.*, (55) :9–37, 1998. Studies in the history of modern mathematics, III.

[KM03] Erich Kaltofen and John May. On approximate irreducibility of polynomials in several variables. In *Proceedings of ISSAC 2003*, pages 161–168, 2003.

[Kur23] J. Kurschak. Irreduzible Formen. *Journal für die Reine und Angew. Math.*, 152 :180–191, 1923.

[Lan83] Serge Lang. *Fundamentals of Diophantine geometry.* Springer-Verlag, New York, 1983.

[Lan85] Susan Landau. Factoring polynomials over algebraic number fields. *SIAM J. Comput.*, 14(1) :184–195, 1985.

[Lan94] Serge Lang. *Algebraic number theory*, volume 110 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1994.

[Lec04a] G. Lecerf. On probabilitic algorithms for multivariate polynomial factorization. Manuscript, Université de Versailles Saint-Quentin-en-Yvelines, 2004.

[Lec04b] G. Lecerf. Sharp precision in Hensel lifting for bivariate polynomial factorization. Manuscript, Université de Versailles Saint-Quentin-en-Yvelines, 2004.

[Lec05] G. Lecerf. Improved dense multivariate polynomial factorization algorithms. Manuscript, 2005.

[Len87] Arjen K. Lenstra. Factoring multivariate polynomials over algebraic number fields. *SIAM J. Comput.*, 16(3) :591–598, 1987.

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4) :515–534, 1982.

[LR90] D. Lazard and R. Rioboo. Integration of rational functions : rational computation of the logarithmic part. *J. Symbolic Comput.*, 9(2) :113–115, 1990.

[Mag] The Magma computational algebra system for algebra, number theory and geometry. http ://magma.maths.usyd.edu.au/magma/. Computational Algebra Group, School of Mathematics and Statistics, The University of Sydney, NSW 2006 Australia.

[Mig89] Maurice Mignotte. *Mathématiques pour le calcul formel.* Mathématiques. [Mathematics]. Presses Universitaires de France, Paris, 1989.

[MŞ99] Maurice Mignotte and Doru Ştefănescu. *Polynomials.* Springer Series in Discrete Mathematics and Theoretical Computer Science. Springer-Verlag Singapore, Singapore, 1999.

[MSOG00] Boris Mayer St Onge and Clement Gosselin. Singularity analysis and representation of the Gough-Stewart platform. *The International Journal of Robotics Research*, 19(3) :271–288, 2000.

[Nag02] Kosaku Nagasaka. Towards certified irreducibility testing of bivariate approximate polynomials. 2002.

[Neu99] Jürgen Neukirch. *Algebraic number theory*, volume 322 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 1999. Translated from the 1992 German original and with a note by Norbert Schappacher, with a foreword by G. Harder.

[Nie93] Harald Niederreiter. A new efficient factorization algorithm for polynomials over small finite fields. *Appl. Algebra Engrg. Comm. Comput.*, 4(2) :81–87, 1993.

[NS98] Kosaku Nagasaka and Tateaki Sasaki. Approximate factorization of multivariable polynomials and its computational complexity. *Sūrikaisekikenkyūsho Kōkyūroku*, (1038) :111–118, 1998. Research on the theory and applications of computer algebra (Japanese) (Kyoto, 1997).

[ORV04] S. Orange, G. Renault, and A. Valibouze. Calcul efficace de corps de décomposition. *Experimental mathematics*, 2004. to appear.

[PH02] F Pernkopf and M.L. Husty. Singularity analysis of spatial Stewart-Gough platforms with planar base and platform. *Proc. ASME Design Eng. Tech. Conf. Montreal, Canada,*, 2002.

[Poh04] M. E. Pohst. Factoring polynomials over global fields I. *J. Symbolic Comput.*, 2004. In Press, Corrected Proof, Available online.

[PS73] M. Paterson and L. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. on Computing*, 2(1) :60–66, 1973.

[Rag97] J.F. Ragot. *Sur la factorisation absolue des polynômes.* PhD thesis, Univ. Limoges, 1997.

[Rag02] Jean-François Ragot. Probabilistic absolute irreducibility test for polynomials. *J. Pure Appl. Algebra*, 172(1) :87–107, 2002.

[Rib01] Paulo Ribenboim. *Classical theory of algebraic numbers.* Universitext. Springer-Verlag, New York, 2001.

[Rot88] Joseph J. Rotman. *An introduction to algebraic topology*, volume 119 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1988.

[Rup86] Wolfgang Ruppert. Reduzibilität ebener Kurven. *J. Reine Angew. Math.*, 369 :167–191, 1986.

[Rup99] Wolfgang M. Ruppert. Reducibility of polynomials $f(x, y)$ modulo p. *J. Number Theory*, 77(1) :62–70, 1999.

[Rup00] David Rupprecht. *Elements de géométrie algébrique approchée : Etude du pgcd et de la factorisation.* PhD thesis, Univ. Nice Sophia Antipolis, 2000.

[Rup04] David Rupprecht. Semi-numerical absolute factorization of polynomials with integer coefficients. *J. Symbolic Comput.*, 37 :557–574, 2004.

[Sam67] Pierre Samuel. *Théorie algébrique des nombres.* Hermann, Paris, 1967.

[Sas01] T. Sasaki. Approximate multivariate polynomial factorization based on zero-sum relations. In B. Mourrain, editor, *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 284–291. ACM, 2001.

[Sch76] Wolfgang M. Schmidt. *Equations over finite fields. An elementary approach.* Springer-Verlag, Berlin, 1976. Lecture Notes in Mathematics, Vol. 536.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.*, 27(4) :701–717, 1980.

[Sch00] A. Schinzel. *Polynomials with special regard to reducibility*, volume 77 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2000. With an appendix by Umberto Zannier.

[SS93] Tateaki Sasaki and Mutsuko Sasaki. A unified method for multivariate polynomial factorizations. *Japan J. Indust. Appl. Math.*, 10(1) :21–39, 1993.

[SSH92] Tateaki Sasaki, Tomokatsu Saito, and Teruhiko Hilano. Analysis of approximate factorization algorithm. I. *Japan J. Indust. Appl. Math.*, 9(3) :351–368, 1992.

[SSKS91] Tateaki Sasaki, Masayuki Suzuki, Miroslav Kolář, and Mutsuko Sasaki. Approximate factorization of multivariate polynomials and absolute irreducibility testing. *Japan J. Indust. Appl. Math.*, 8(3) :357–375, 1991.

[Ste02] A. Steel. A new scheme for computing with algebraically closed fields. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 491–505. Springer, Berlin, 2002.

[Sto00] A. Storjohann. *Algorithms for matrix canonical forms.* PhD thesis, ETH, Zürich (Switzerland), 2000.

[SU97] M. F. Singer and F. Ulmer. Linear differential equations and products of linear forms. *J. Pure Appl. Algebra*, 117/118 :549–563, 1997. Algorithms for algebra (Eindhoven, 1996).

[SV00] A.J. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *J. of Complexity*, 16(3) :572–602, 2000.

[SVW01a] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM J. Numer. Anal.*, 38(6) :2022–2046, 2001.

[SVW01b] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using projections from points on the components. In E.L. Green, S. Hoşten, R.C. Laubenbacher, and V. Powers, editors, *Symbolic Computation : Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, pages 37–51. AMS, 2001.

[SVW01c] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In C. Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher, editors, *Application of Algebraic Geometry to Coding Theory, Physics and Computation*, pages 297–315. Kluwer Academic Publishers, 2001. Proceedings of a NATO Conference, February 25 - March 1, 2001, Eilat, Israel.

[SVW02a] A.J. Sommese, J. Verschelde, and C.W. Wampler. Advances in polynomial continuation for solving problems in kinematics. In *Proc. ASME Design Engineering Technical Conf. (CDROM), Montreal, Quebec, Sept. 29-Oct. 2, 2002.*, 2002. Paper DETC2002/MECH-34254. A revised version will appear in the *ASME Journal of Mechanical Design.*

[SVW02b] A.J. Sommese, J. Verschelde, and C.W. Wampler. A method for tracking singular paths with application to the numerical irreducible decomposition. In M.C. Beltrametti, F. Catanese, C. Ciliberto, A. Lanteri, and C. Pedrini, editors, *Algebraic Geometry, a Volume in Memory of Paolo Francia*, pages 329–345. Walter de Gruyter, 2002.

[SVW02c] A.J. Sommese, J. Verschelde, and C.W. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM J. Numer. Anal.*, 40(6) :2026–2046, 2002.

[SVW03a] A.J. Sommese, J. Verschelde, and C.W. Wampler. Homotopies for intersecting solution components of polynomial systems. Submitted for publication, 2003.

[SVW03b] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical factorization of multivariate complex polynomials. Accepted for publication in a special issue of *Theoretical Computer Science* on Algebraic and Numerical Algorithms, 2003.

[SVW03c] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using PHCpack. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 109–130. Springer–Verlag, 2003.

[SW96] A.J. Sommese and C.W. Wampler. Numerical algebraic geometry. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 749–763. AMS, 1996. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics. Park City, Utah, July 17-August 11, 1995, Park City, Utah.

[Tra76] Barry M. Trager. Algebraic factoring and rational function integration. In *Proceedings of the third ACM symposium on Symbolic and Algebraic Computation*, pages 219–226. ACM Press, 1976.

[Tra85] Barry Trager. *On the integration of algebraic functions*. PhD thesis, M.I.T., 1985.

[Tra86] Carlo Traverso. A study on algebraic algorithms : the normalization. *Rend. Sem. Mat. Univ. Politec. Torino*, (Special Issue) :111–130 (1987), 1986. Conference on algebraic varieties of small dimension (Turin, 1985).

[vH02] Mark van Hoeij. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95(2) :167–189, 2002.

[Völ96] Helmut Völklein. *Groups as Galois groups*, volume 53 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1996. An introduction.

[vzG85] Joachim von zur Gathen. Irreducibility of multivariate polynomials. *J. Comput. System Sci.*, 31(2) :225–264, 1985. Special issue : Twenty-fourth annual symposium on the foundations of computer science (Tucson, Ariz., 1983).

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[WH90] Xing Hua Wang and Dan Fu Han. On dominating sequence method in the point estimate and Smale theorem. *Sci. China Ser. A*, 33(2) :135–144, 1990.

[Yak] Jean Claude Yakoubsohn. Conditionnement, alpha-théorie, homotopie et localisation des racines des systèmes polynomiaux. Cours de DEA : Limoges Janvier 1999.

[Zas69] H. Zassenhaus. On Hensel factorization I. *J. Number Theory*, 1(1) :291–311, 1969.

[Zas78] H. Zassenhaus. A remark on the Hensel factorization method. *Math. Comp.*, 32(141) :287–292, 1978.

[Zip93] Richard Zippel. *Effective polynomial computation.* Kluwer Academic Publishers, 1993.

Cette thèse porte sur les algorithmes de factorisation absolue. Elle débute par un état de l'art (avant notre travail) puis présente nos contributions. Celles-ci sont organisées en deux parties.

La première partie correspond à l'étude symbolique-numérique. Nous donnons une méthode permettant d'obtenir une factorisation absolue exacte à partir d'une factorisation absolue approchée. Ensuite cette méthode est utilisée pour obtenir un algorithme de factorisation absolue. Cet algorithme reprend des idées développées par A. Galligo, D. Rupprecht, et, M. van Hoeij. Grâce à l'utilisation de l'algorithme LLL, notre algorithme permet d'obtenir la factorisation absolue de polynômes de grands degrés (supérieur à 100) jusqu'ici impossible.

La deuxième partie présente deux algorithmes symboliques. Le premier est l'adaptation de la technique “remonter-recombiner” à l'algorithme de S. Gao. Nous obtenons ainsi un algorithme de factorisation absolue ayant une meilleure complexité que celui de S. Gao. Le deuxième est un algorithme, de type Las Vegas, qui nous permet de tester l'irréductibilité absolue d'un polynôme à coefficients entiers. L'approche proposée tire profit du calcul modulaire et de l'information contenu dans le polytope de Newton.

Mots clefs : polynômes en plusieurs variables, factorisation absolue, algorithme symbolique-numérique, algorithme symbolique, LLL, polytope de Newton.

This thesis is about absolute factorization algorithms. The first chapter is a survey on this topic, and then we describe our contributions. They are divided in two parts.

The first part corresponds to the symbolic-numeric study. We give a method which allows us to get an exact absolute factorization from an approximate one. Next, this method is used to get an absolute polynomial factorization algorithm. This algorithm uses ideas developped by A. Galligo, D. Rupprecht, and M. van Hoeij. Thanks to the LLL algorithm, it allows us to get the absolute factorization for polynomials with big degree (bigger than 100). It was impossible before.

In the second part, we give two algorithms. The first one adapt a “lifting and recombination” scheme to the Gao's algorithm. We get then a new algorithm with a better complexity than the Gao's one. The second one is a Las Vegas algorithm. It is an absolute irreducibility test for polynomials with integer coefficients. This algorithm use modular computations, and the shape of the Newton's polytope.

Keywords : polynomials with several variables, absolute factorization, symbolic-numeric algorithm, symbolic algorithm, LLL, Newton's polytope.

Université de Nice-Sophia Antipolis-Laboratoire J.A. Dieudonné.