Complexity of polynomial systems representations: triangulation, modular methods, dynamic evaluation.

Xavier Dahan

ÉCOLE POLYTECHNIQUE

THÈSE

présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE

Spécialité :

INFORMATIQUE

par

Xavier DAHAN

Sujet de la thèse :

Sur la représentation des systèmes polynomiaux : triangulation, méthodes modulaires, évaluation dynamique

Soutenue le 24 novembre 2006 devant le jury composé de :

Président :	Maurice MIGNOTTE
Rapporteurs :	Ferdinando MORA
	Kazuhiro YOKOYAMA *(absent)*
Examinateur :	Luis Miguel PARDO
Directeurs :	Marc GIUSTI
	Éric SCHOST

LABORATOIRE D'INFORMATIQUE
Unité Mixte CNRS n°7161
École Polytechnique 91128 Palaiseau Cedex FRANCE

Sur la représentation des systèmes polynomiaux : triangulation, méthodes modulaires, évaluation dynamique

Thèse présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE

par Xavier DAHAN

Rédigée en anglais sous le titre :

On the representation of polynomial systems: triangulation, modular methods, dynamic evaluation

Soutenue le 24 novembre 2006 devant le jury composé de :

Président :	Maurice MIGNOTTE	*Université Louis Pasteur*, Strasbourg
Rapporteurs :	Ferdinando MORA	*Università di Genova*, Italie
	Kazuhiro YOKOYAMA *(absent)*	*Rikkyo university*, Japon
Examinateur :	Luis Miguel PARDO	*Universidad de Cantabria*, Espagne
Directeurs :	Marc GIUSTI	*École Polytechnique*
	Éric SCHOST	*University of Western Ontario*, Canada.

2006

ÉCOLE POLYTECHNIQUE
PARIS

Remerciements

Beaucoup d'aspirants thésards ont dans l'idée de travailler avec un grand chercheur expérimenté, une "ponte" dans le jargon. On entend dire aussi que travailler avec un jeune chercheur permet une collaboration plus égalitaire et fructeuse. Pour ma part, j'ai eu l'impression de bénéficier des avantages de ces deux types de directeur en étant encadré par Éric Schost. Son large savoir, allant de l'informatique appliquée jusqu'à des mathématiques abstraites pas évidentes m'a particulièrement bien servi. À tous points de vue, il s'est avéré un directeur exemplaire, et je souhaite à chacun un encadrement si haut de gamme.

Je dois beaucoup aussi à Marc Moreno Maza. Il m'a invité à plusieurs reprises dans le chaleureux laboratoire "Orcca" et j'ai pu commencer à tirer profit d'une fructueuse collaboration internationale avec ses étudiants Yuzhen Xie, Wenyuan Wu et Xin Jin. Je conçois aujourd'hui l'intérêt que je peux tirer des implantations Maple qu'il a dirigé, et le travail que ça a demandé.

Je voudrais remercier vivement Kazuhiro Yokoyama et Teo Mora d'avoir accepté de rapporter ma thèse, en de si brefs délais. Ainsi que les autres membres du jury, en particulier Marucie Mignotte qui m'a fait l'honneur de présider la soutenance, et Luis-Miguel Pardo.

J'adresse de vifs remerciements à Marc Giusti : en étant le directeur du feu laboratoire stix, où j'ai commencé ma thèse, sa stature au sein de l'École a permis à toute son équipe de travailler dans de bonnes conditions. Il a joué le même rôle ensuite en tant que chef de l'équipe max, pour l'imposer dans le laboratoire d'informatique.

Durant plus de trois ans, j'ai pu côtyer les chercheurs Bruno Salvy, Alin Bostan, François Ollivier, Michel Fliess, Jean Moulin Ollagnier, Frédéric Chyzak, Mohab Safey El Din et Grégoire Lecerf et m'enrichir de leurs contacts. Les ingénieurs système m'ont beaucoup aidé à sortir de mon ignorance totale en bureautique. Merci à Matthieu Guionnet notamment que j'ai certainement (le pauvre) le plus sollicité. Également à Bernd Wiebelt, Pierre Lafon et James Regis. L'assistante que j'ai fréquentée durant ces années était compétente, agréable, et avait le mot facile pendant les pauses, c'est Nicole Dubois. Un grand merci à elle.

Mes collègues thésards, que ce soit ceux du stix, Thomas Cluzeau et Anne Fredet, du lix Simon Bliudze, Behshad Behzadi, Dimitri Lebedev, Claus Gwiggner, Mathilde Hurand, Jérôme Waldispuhl, ceux de orcca (Elena entre autres), où ceux de conférence (Clément, Guénaël etc.), ont gaiement accompagnés ces années de labeur. *Thank you, guys.*

Enfin, je pense aux amis que je n'ai pas cités et à ma famille aussi, surtout à mes parents.

Contents

Introduction

Avec les nouvelles possibilités de calcul apportées par les ordinateurs, un regain d'intérêt pour les questions effectives en algèbre a emergé après, puis à côté (plus que côte-à-côte) de son abstraction croissante tout au long du XXème siècle. Cela s'est fait parallèlement avec la forte demande industrielle en modélisation, et l'émergence dans l'industrie de la branche des mathématiques appliquées, l'*analyse numérique*, qui malgré les inévitables problèmes liés à la décision pour un nombre d'être nul ou pas, continue à y jouer un rôle dominant. Pourtant, loin de se cantonner à des algorithmes algébriques qui seraient utilisés comme des calculs "expérimentaux" par les mathématiciens, le Calcul Formel a permis des applications industrielles dont un des exemples les plus spectaculaires est certainement la cryptographie.

Le travail en Calcul Formel peut s'effectuer à plusieurs échelons : architecture et arithmétique des ordinateurs, algorithmique de base (opérations de base), algorithmique impliquant des structures évoluées, création de logiciels spécialisés (*Computer Algebra System*) ; analyse de complexité. Les sujets abordés ici se situent dans la conception d'algorithmes avec des structures évoluées, ainsi que de leur analyse de complexité.

Les thèmes mathématiques en Calul Formel concernent surtout la théorie algébrique des nombres, l'algèbre commutative et géometrie algébrique, l'algèbre différentielle. Les polynômes y jouent ainsi un des rôles principaux, et c'est sur eux que portent les résultats de cette thèse. Avant d'en détailler les énoncés, introduisons le contexte et les concepts permettant d'en comprendre les enjeux.

Calcul avec les systèmes polynomiaux

Il est question de transformer un système d'équations polynomiales donné en un ou plusieurs autres ayant les propriétés adéquates pour lire les informations que l'on souhaite acquérir. Cette transformation d'un système à un autre est appelée *résolution* (d'un système de polynômes), comme souligné par Daniel Lazard dans [74]. Ces propriétés pourront être la capacité à représenter l'idéal engendré par le système et le calcul dans l'algèbre quotient, la lecture efficace des singularités, la précision des approximations numériques, entre autres.

Bases de Gröbner

La méthode la plus utilisée est le calcul de bases de Gröbner, qui permettent de résoudre un grand nombre de problèmes, et pouvant être calculées par un algorithme simple (d'après Buchberger). Elles ne constituent souvent qu'une étape intermédiaire mais permettent une large palette d'applications. De nettes améliorations ont été produites dans la conception et l'implémentation de l'algorithme de Buchberger depuis sa création. Malgré une complexité dans le cas le pire doublement exponentielle en le nombre de variables ou le degré des

polynômes, il n'en demeure pas moins que leurs calculs sont assez efficaces. Récemment, Bardet *et al.* [11] ont d'ailleurs montré qu'en moyenne cette complexité était simplement exponentielle, en un degré de "semi-régularité".

Représentation à la Kronecker

Une autre structure de données majeure est la représentation par *élément primitif* (Cf. § 1.1.2). C'est sous la forme rationnelle que cette représentation est la plus économe en place mémoire. Bien que déjà mentionnée dans l'œuvre de Kronecker, qui justifie le terme de *représentation de Kronecker* [75] pour désigner cette représentation, aujourd'hui la terminologie de *Représentation Univariée Rationnelle* (en abrégée RUR) est également largement utilisée. Trois grandes écoles existent pour calculer ce genre de système, celle que représente désormais Rouillier *et al.*, obtenue à partir d'une base de Gröbner [101], celle du groupe TERA, dont l'algorithme finalisé porte le nom de *résolution géometrique* [52], et celle de l'algèbre linéaire des matrices bezoutiennes [39]. Cette structure de données est bien adaptée pour les calculs numériques, car permet l'utilisation du savoir-faire du cas univarié (malgré une grande précision nécessaire pour ces approximations). Toutefois ne permet pas la représentation des singularités, seulement une information sur les multiplicités peut être fournie ; certaines informations géometriques, comme d'éventuelles symétries par exemple, sont généralement perdues par le choix de la forme linéaire séparante.

Décomposition triangulaire

Dans cette thèse on s'intéressera aux décompositions triangulaires (ou *triangulations*, en s'inspirant du terme anglais "triangulation-decomposition") d'un système polynomial. Il n'y a plus un mais plusieurs systèmes en sortie, à l'instar de la décomposition primaire, qui en est en fait un cas particulier. Les variables sont ordonnées par un ordre lexicographique, le i-ème polynôme fait intervenir au moins une nouvelle variable plus grande que toutes celles du $i-1$-ème. Ce type de système, très structuré, a été largement étudié du côté de l'algèbre différentielle comme commutative, son utilisation pour la résolution bien reconnu aujourd'hui (voir l'article de Lazard sur *un* état de l'art de la résolution des systèmes polynomiaux en 2000 [73]). Cette structure triangulaire permet d'avoir un point de vue univarié, *la* variable considérée étant la plus grande — pour l'ordre lexicographique considéré, les autres sont placées dans le corps ou l'anneau de base. Selon les hypothèses que l'on rajoute à ces ensembles triangulaires, notamment en ce qui concerne les polynômes formant les coefficients dominants en cette plus grande variable, de nombreuses définitions ont été introduites : ensembles caractéristiques, chaînes régulières, ensembles triangulaires de Lazard (se réferrer au § 1.1.3 pour plus de détails). On ne s'intéressera qu'aux ensembles triangulaires de Lazard et aux chaînes régulières dans ce manuscrit. Ces deux derniers systèmes offrent en effet des propriétés intéressantes pour la résolution des systèmes polynomiaux, tant du point de vue conceptuel qu'algorithmique. Les améliorations apportées ici découlent essentiellement des deux faits suivants :

1. l'existence d'un opérateur de Newton-Hensel pour les ensembles triangulaires de Lazard zéro-dimensionnels.

2. le point de vue univarié des ces polynômes.

Le point 1 autorise les "calculs modulaires" (Cf. Figure 3.1, p. 94) et sera au centre des chapitres 4 et 3, et dans une moindre mesure dans le chapitre 2. Bien que sous-jacente même à l'intérêt des ensembles triangulaires, l'utilité du point 2 se ressent particulièrement dans le chapitre 5 où l'on étendra des algorithmes rapides dédiés aux polynômes univariés tels que le calcul de pgcd, le calcul d'une base sans facteurs communs d'une famille de polyômes, à ces ensembles triangulaires ; cela dans le contexte de l'estimation de la complexité de l'évaluation dynamique.

Résultats

Chapitre 2 Souvent les algorithmes de résolution font intervenir des polynômes intermédiaires avec des coefficients très grands, comparés à ceux qui sont en entrée. Il est bien connu que les calculs modulaires peuvent porter remède à ce grossissement, que ce soit avec des "restes chinois" ou avec la remontée de Hensel. Un problème de même nature est posé pour les systèmes de dimension positive où cette fois-ci, c'est le degré des variables libres qui peut être excessivement élevé. Dans ce cas il est notoirement connu que l'opérateur de Newton peut permettre de réduire les calculs à la dimension zéro sous certaines hypothèses, qui avec les progrès deviennent de moins en moins restrictives. Les algorithmes de résolution sur lesquels porteront nos résultats concernent les méthodes de *triangulation* d'un système en plusieurs ensembles triangulaires.

L'opérateur de Newton est un outil omniprésent dans le calcul numérique approché. L'extension au cadre formel de l'approximation —numérique— des zéros d'un système polynomial s'est avérée efficace après les travaux de nombreux auteurs aboutissant à l'algorithme de résolution présenté dans [75] pour le calcul de représentation de Kronecker, puis dans [102] pour les ensembles triangulaires de Lazard zéro-dimensionnels et radicaux.

Le lien avec la remontée de Hensel (calcul modulo les puissances d'un nombre premier) est bien mis en valeur dans [75], où la terminologie de topologie $\mathfrak{m}$-adique permet d'unifier les deux approches (topologie archimédienne dans le cas de l'opérateur de Newton, non-archimédienne dans le cas de la remontée de Hensel). Le terme de remontée de Newton-Hensel a ainsi bien un sens, et sera utilisé par la suite.

Le principe est l'approximation successive des zéros avec convergence quadratique. Le nombre d'étapes permettant d'assurer une approximation suffisante requiert une borne sur la taille des coefficients ou le degré des variables libres pour la représentation de sortie (les ensembles triangulaires donc). Cela fait l'objet du second chapitre, où une amélioration substantielle des précédentes bornes pour les ensembles triangulaires de Lazard est prouvée. L'outil de mesure adéquat est la *hauteur*, provenant de l'approximation diophantienne et permettant d'unifier le cas archimédien et non-archimédien. Différentes définitions existent pour les hauteurs des variétés, celle que nous utiliserons est due à Philippon [95], reposant sur la forme de Chow.

Theorem 2.7 *Soit $T \subset K[X_1, \ldots, X_n]$ un ensemble triangulaire de Lazard, radical et de dimension zéro défini sur un corps K, extension finie de $\mathbb{Q}$ ou du corps des fractions rationnelles en m variables, $k(p_1, \ldots, p_m)$. On note V l'ensemble (fini) des zéros de T dans une clôture algébrique de k, et pour $1 \leq \ell \leq n$, $\pi_\ell^n(V)$ la projection de V sur les axes $X_1, \ldots, X_\ell$.*

La hauteur $h(T_\ell)$ de T_ℓ est bornée par une quantitée en

$$O\Big(\deg\big(\pi_\ell^n(V)\big)\cdot h\big(\pi_\ell^n(V)\big)+\deg\big(\pi_\ell^n(V)\big)^2\Big).$$

Nous en avons profité pour clarifier le même type de résultat (déjà connu mais dans une formulation moins générale) pour les polynômes formant une représentation de Kronecker. Soit V une variété de dimension zéro définie sur un corps K, extension finie de $\mathbb{Q}$ ou de $k(p_1,\ldots,p_m)$. On considère la représentation de Kronecker de V de forme linéaire séprante U, et d'élément primitif χ_u :

$$(\chi_u(T)\,,\ w_1(T)\,,\ w_2(T)\,,\ldots,\ w_n(T)).$$

Theorem 2.2 *La hauteur des coefficients de $\chi_u'(T)$ et de $w_i(T)$ est bornée par :*

$$\begin{array}{c} h(V)+\deg(V)h(U)+\deg(V)\log(n+2)+(n+1)\log\deg(V) \quad \textit{(K est un corps de nombres)}\\ h(V)+\deg(V)h(U) \quad \textit{(K est un corps de fonctions).}\end{array}$$

Ces bornes ont la particularité d'être intrinsèques, c'est-à-dire ne dépendent pas d'un système polynomial particulier représentant V. Il est toutefois facile de déduire des bornes extrinsèques grâce aux théorèmes de Bézout géometrique et arithmétique. D'autres bornes sont données pour des systèmes triangulaires avec introduction de différents coefficients dominants, avec des preuves totalement différentes, mais non dénuées d'intérêt, car font apparaitre des formules non triviales de dérivations de la forme de Chow. L'Introduction à ce chapitre propose un résumé des méthodes, résultats, et comparaisons expérimentales.

Chapitre 4 Nous nous intéressons dans ce chapitre à la résolution des systèmes polynomiaux par triangulation-decomposition, la nouveauté étant dans la méthode, puisque cela y est fait *modulairement* : les principaux calculs, en général les plus gourmands en taille mémoire, sont faits modulo un nombre premier p, donc sans croissance excessive des coefficients. Les algorithmes de triangulation ne renvoient pas un résultat canonique, et savoir si le nombre premier p de réduction donne lieu à une réduction stable, c'est-à-dire compatibilité entre les ensembles triangulaires obtenus par exécution de l'algorithme de triangulation modulo p et sur $\mathbb{Q}$, n'est pas évident.

Nous avons ainsi introduit une nouvelle triangulation canonique des systèmes polynomiaux dans le cas radical et zéro-dimensionnel, la *décomposition équiprojetable* ; le choix du nombre premier de réduction est quantifiable numériquement. On peut même utiliser un nombre premier plus petit au détriment du déterminisme, mais en contrôlant alors complètement la probabilité de succès. Ce type d'analyse probabiliste est relativement standard une fois que le critère numérique est prouvé (sinon il serait souhaitable qu'elle le devienne).

Theorem 4.1 *(Vulgarisé). Soit n polynômes multivariés $f_1,\ldots,f_n$ dans $\mathbb{Q}[X_1,\ldots,X_n]$ de degré au plus d, et de hauteur au plus h. Il existe un entier A, dont le nombre de chiffres est essentiellement borné par une quantitée en $\tilde{O}(n^2hd^{2n+1})$, tel que tout nombre premier*

p ne divisant pas A, rend compatible la décomposition équiprojetable des zéros simples *de* $V(f \mod p)$ *et la réduction modulo p de la décomposition équiprojetable des zéros* simples *de* $V(f)$.

Ceci résout le défaut de compatibilité entre les ensembles triangulaires calculés modulairement sur $\mathbb{F}_p$, et ceux calculés sur $\mathbb{Q}$. L'entier A n'est pas explicité ; pour s'assurer de choisir un bon premier p de réduction, il faudrait donc le choisir plus grand que la borne donnée, soit un nombre de chiffres de l'ordre de d^{2n+1}, ce qui n'est pas une amélioration substantielle pour l'utilisation d'une méthode modulaire. En revanche, il est possible de déduire un algorithme probabiliste avec contrôle de la probabilité de succès :

Theorem 4.2*(Vulgarisé).* *Pour tout* $\varepsilon > 1$ *assez grand, le choix d'un nombre premier p avec un nombre de chiffres de l'ordre de* $\log \varepsilon + \log \theta(n, d, h)$ *donne lieu, avec probabilité* $1 - \frac{1}{\varepsilon}$*, à une compatibilité semblable à celle énoncée dans le Theorem 4.1, ainsi qu'au calcul de la décomposition équiprojetable des zéros simples de* $V(f)$.

L'algorithme probabiliste utilisé requiert au plus un nombre polynomial en des données "standards" du problème : degré et hauteur de $V(f)$*, complexité d'évaluation de* f *et taille des constantes dans ce schéma d'évaluation de* f*, notamment. La fonction* θ *est dominée par un terme en* $O(n^2 h d^{2n+1})$.

Chapitre 3 Une autre transformation des systèmes polynomiaux intéressante, un peu à côté de ce que nous avons appelé résolution, est le changement d'ordre des variables. Bien sûr pour les bases de Gröbner où le calcul sous un ordre peut être bien plus efficace que sous un autre, ceci peut présenter un intérêt, mais nous nous restreindrons aux ordres lexicograhiques, donc à un périmètre bien délimité, mais toutefois non dépourvu d'applications : on peut citer l'implicitisation et toutes ces utilisations [27], mais aussi la réecriture des invariants dans une base d'invariants fondementaux. De plus, l'intérêt réside également dans l'approche théorique nouvelle. En dimension zéro cela se fait déjà efficacement (le célèbre FGLM [42] de complexité cubique en le degré de la variété) et l'on en tire parti pour le cas de la dimension positive, grâce à des spécialisations judicieuses des variables libres pour se retrouver en dimension zéro. Mais le changement d'ordre et la spécialisation de variables libres, impliquant leur disparition, ne sont pas à priori compatibles. Ce problème est résolu par l'adaptation non évidente des principales étapes de l'algorithme de résolution géometrique au contexte du changement d'ordre.

Le changement d'ordre s'opère en effet étape par étape par échanges successifs de deux variables (voir la figure 3.3). Le système est remonté par l'opérateur de Newton en dimension un, puis respécialisé en dimension zéro en une autre variable ; un changement d'ordre y est alors effectué. On répète ces trois opérations sur plusieurs couples de variables à remonter / spécialiser, et l'on parvient à avoir en sortie une fibre de remontée de la sortie souhaitée (cette dernière pouvant être obtenue par application de l'opérateur de Newton multivarié pour faire apparaitre les variables libres, mais alors le coût est exponentiel).

Cela ne fonctionne pas pour tous les systèmes, bien sûr. Seules les "chaînes régulières" (Definition 1.3, page 18), jouant un rôle important dans le cas de la dimenion positive, sont envisagées. On considère une variété irréductible W de dimension positive, on dispose d'une chaîne régiulière la décrivant, et l'on souhaite changer d'ordre des variables de cette chaîne régulière. Pour décider des couples de variables que nous aurons à remonter / spécialiser, nous utilisons l'analogie entre les variables libres de notre variété et celles d'un de ses espaces

tangents en un point générique.

Theorem 3.1 *Soit* $\mathbf{F} = (F_1, \ldots, F_s)$ *une chaîne régulière dans* $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_n]$ *pour un ordre d'entrée* $<$*, dont le saturé* $\mathsf{Sat}(\mathbf{F})$ *est un idéal premier. Supposons connus un straight-line program de taille* L *qui calcule* $\mathbf{F}$*, la variable principale de chacun des polynômes de* $\mathbf{F}$ *ainsi que leur degré en cette variable.*

Étant donnné un ordre cible $<'$ *sur* $\mathbf{X}$*, on peut calculer par un algorithme probabiliste une fibre de remontée pour l'ordre cible* $<'$*. En cas de succès, l'algorithme utilise*

$$(nL \deg V(\mathsf{Sat}(\mathbf{F}))^{O(1)}$$

opérations dans $\mathbb{K}$*. L'algorithme choisit* $n + s$ *paramètres dans* $\mathbb{K}$*. Si ces paramèters sont choisis aléatoirement uniformément dans un ensemble fini* S *de* $\mathbb{K}$*, alors, notant* $m = \max(n, d)$*, la probabilité d'échec est d'au plus* $g(n, m, d)/|S|$*, où* $g \in O(nm^2d^{3n})$*.*

La complexité est polynomiale puisqu'en "grand O(1)" en les données naturelles d'entrée L, n et $\deg V(\mathsf{Sat}(\mathbf{F}))$, ce qui constitue une nouveauté pour ce type de problème en dimension positive. De même la probabilité d'échec de cet algorithme croit polynomialement en le nombre de Bézout (égal ci-dessus à d^n). En choisissant S suffisamment vaste, cette probabilité peut être arbitrairement minorée.

Chapitre 5 Ce chapitre est un peu à part des trois autres dans la mesure où le lien avec l'opérateur de Newton-Hensel est caché, et que les applications dépassent le cadre de cet opérateur. Dans les annés 80, un article d'environ une page de *D*icrescenzo, *D*ominique *D*uval et *D*ella-*D*ora, jette les bases d'une nouvelle méthode pour calculer avec les nombres algébriques, en suivant une idée suggérée par Lazard. Depuis, le "principe D5" fait l'objet de nombreux travaux toujours en chantier au sujet des applications, les liens avec d'autres domaines de l'informatique et de l'algorithmique. Toutefois, du côté de la complexité, à part l'étude de la complexité parallèle dans [46], aucun résultat général n'a été donné, à notre connaissance. Nous comblons ce vide dans ce chapitre.

Comment calculer avec les nombres algébriques ? L'approche standard est de calculer dans le quotient $\mathbb{Q}[X]/(p)$ où p est le polynôme minimal du nombre algébrique. Lorsqu'il y en a plusieurs, par exemple solutions d'un polynôme f (que l'on supposera sans facteur carré), on peut factoriser et retrouver le polynôme minimal de chaque nombre algébrique.

Plus généralement, soit I un idéal radical de dimension 0 et Z l'ensemble des points algébriques associés à I, et f une fonction algébrique définie sur Z. L'ensemble des points où f est inversible se note $D(f)$ en général. Ainsi $D(f)$ et son complémentaire $V(f)$ forment une partition de Z, qui se traduit en terme d'idéaux en un scindage de l'anneau de fonctions sur Z:

$$k[X_1, \ldots, X_n]/I \simeq k[X_1, \ldots, X_n]/(I : f) \times k[X_1, \ldots, X_n]/I + (f), \tag{1}$$

et ce, sans recours à la décomposition primaire. On peut commencer par décomposer I en ensembles triangulaires et se ramener à une situation où I est lui-même engendré par un ensemble triangulaire de Lazard T. D'un point de vue effectif, l'intérêt est la possiblité de se réferrer, par induction sur le nombre de variables, au cas bien compris d'une seule variable. En particulier, l'inversion sera donnée par un calcul de pgcd étendu. Le nombre de divisions euclidiennes nécessaires à son calcul conditionne inévitablement le nombre de scindages

ayant lieu, puisque qu'il faut alors ne considérer que des restes *unitaires*, nécessitant un calcul d'inverse. Ainsi, ça ne sera qu'un raffinement de chacune des deux branches de 1 que l'on ne pourra calculer.

Lors d'une étude de complexité d'un algorithme reposant sur ce principe, on est amené à estimer le coût de l'opération d'évaluation après un scindage ; plus précisemment, soit $T \in k[X_1, \ldots, X_n]$ un ensemble triangulaire (de Lazard, zéro dimensionel et radical) et $T^1, \ldots, T^e$ une famille de e ensembles triangulaires tels que $V(T^i) \cap V(T^j) = \emptyset$, et $V(T) = V(T^1) \cup \ldots \cup V(T^e)$ (on dira que $T^1, \ldots, T^e$ est une *décomposition triangulaire* de T).

$$\begin{array}{rcl} k[X_1, \ldots, X_n]/(T) & \to & k[X_1, \ldots, X_n]/(T^1) \times \cdots \times k[X_1, \ldots, X_n]/(T^e) \\ \alpha \bmod T & \mapsto & (\alpha_1 \bmod T^1, \ldots, \alpha_e \bmod T^e). \end{array} \tag{2}$$

La complexité dans le cas univarié est bien connue (c'est la multi-évaluation, Proposition 1.7 4., p. 32). Elle se généralise aux ensembles triangulaires par induction. Toutefois, les *hypothèses* nécessaires au cas multivarié ne se généralisent pas elles, aussi facilement, et nécessitent un raffinement de la définition de décomposition triangulaire, appelée *non-critical triangular decomposition* (voir Definition 5.5, page 147 et l'exemple qui la précède et surtout qui la suit). Dans ces conditions, l'opération d'évaluation discutée ci-dessus peut être calculée avec une complexité raisonnable, comparable au cas univaré. Soit $T = T_1, \ldots, T_n$ les polynômes de l'ensemble triangulaire T, et soit d_i le degré en X_i de T_i.

Proposition. *Soit $\mathsf{M}(d)$ une borne supérieure pour le coût de la multiplication univariée de degré d. L'opération d'évaluation (2) peut être calculée en moins de $nC^n \prod_{i \leq n} \mathsf{M}(d_i) \log d_i$ opérations sur k.*

Il faut pouvoir raffiner une décomposition triangulaire en une décomposition *sans paire critique* (voir Definition 5.4 p. 147), dans le temps requis. Cela nécessite un calcul de pgcd rapide au dessus d'un produit de corps. L'algorithme du Half-gcd y est adapté dans le § 5.3 ; or cet algorithme crée lui même des paires crtiques, dues aux inversions produites pour rendre unitaire les restes des divisions euclidiennes. Cependant, ces nouvelles paires critiques sont en $n-1$ variables, ce qui est rend possible un schéma de double récurrence"croisée" (Cf. Figure 5.1). Par ce biais on parvient à :

Theorem 5.1. *Il existe une constante C indépendante de T et du degré des polynômes de T, telle que l'addition, la multiplication et la quasi-inversion dans $k[X_1, \ldots, X_n]/(T)$ peuvent être calculées en au plus $C^n \prod_{1 \leq i \leq n} \mathsf{M}(d_i) \log(d_i)^3$ opérations sur k.*

Ce résultat est de même ordre que l'inversion modulaire univariée, à des facteurs carrés logarithmiques, et aux mesures de complexité naturelles liées à l'ensemble triangulaire T près. En ce sens, on peut considérer que cette complexité, et par voie de conséquences celles des autres algorithmes présentés dans ce chapitre, sont certainement optimales : les grandeurs apparaissent en croissance linéaire, si l'on omet les facteurs logarithmiques.

Chapter 1

Preliminaries

1.1 Polynomial systems representations

This section presents two representations of polynomial systems which are of practical interest. The *primitive element* and *triangular* representations of a polynomial system. The first one has a good behavior under numerical approximations. In real geometry it allows to reduce problems to the univariate situation, where powerful methods of isolation of real roots exist. It is a central object in this topic, and we refer to the book [12] for the details. More precisions are given in § 1.1.2.

The second one, triangular systems, is used in manipulation of algebraic numbers, Galois theory [2, 4, 97], differential algebra [59], dynamic evaluation [35] and in the CAD algorithm of real geometry (Cf. [12] and the references therein), where the management of the lifting step is handled by triangular systems [98]. In § 1.1.3 more details are added.

1.1.1 Hypotheses - Presentation

In the sequel, K is assumed to be any commutative field (but we will only work with number fields, finite fields and function fields in m variables). In particular K is not necessary supposed to be *perfect*, possibly inducing a lack of correspondence between the geometry and the algebraic equations: we have in mind the classic example of non-perfect field $K = \mathbb{F}_p(T)$ and the irreducible polynomial $X^p - T$ in $K[X]$. The number of solutions in the algebraic closure $\bar{K}$ of K is one, but with multiplicity p, whereas the ideal $(X^p - T)$ is radical. As we want to discard this kind of bad situation we need to add some *separability* assumptions on the ideal generated by our algebraic equations.

Separability assumption. *In the sequel, given a zero-dimensional radical ideal I of polynomials lying in $K[X_1, \ldots, X_n]$, we assume that the extension $K \to K[X_1, \ldots, X_n]/I$ is separable, that is to say: If $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ are the primary ideals of I, then each field extension $K \to K[X_1, \ldots, X_n]/\mathfrak{p}_i$ is separable.*

Under this assumption, the number of solutions in $\mathbb{A}^n_{\bar{K}}$ of the polynomials in I is equal to the dimension of the K-vector space $K[X_1, \ldots, X_n]/I$. All these solutions are *simple*. As usual, $V(I)$ denotes this set of solutions. We have a satisfactory correspondence between the number of solutions and the degree of the defining polynomials. Throughout this chapter of preliminaries, we will use the following Lemma:

Lemma 1.1. *Consider a polynomial U in $K[X_1,\ldots,X_n]$, and its multiplication map M_U:*

$$\begin{array}{rccc} M_U : & K[X_1,\ldots,X_n]/I & \xrightarrow{\times U} & K[X_1,\ldots,X_n]/I \\ & P \bmod I & \longmapsto & U\cdot P \bmod I. \end{array}$$

Under the separability assumption on I, the characteristic polynomial χ_U of U verifies:

$$\chi_U(T) = \prod_{\alpha\in V(I)} T - U(\alpha),$$

where $V(I)$ is the set of solutions in $\bar{K}$ of the polynomials in I.

Proof: Let us consider the dual endomorphism $\widehat{M_U}$ of M_U, and for $\alpha \in V = V(I)$, the evaluation map Eval_α from $\bar{K}[X_1,\ldots,X_n]/I$ to $\bar{K}$, defined by

$$\mathsf{Eval}_\alpha(p(X_1,\ldots,X_n) \bmod I) = p(\alpha).$$

Then,

$$\begin{array}{rcl} \widehat{M_U}(\mathsf{Eval}_\alpha)(p) & = & (\mathsf{Eval}_\alpha \circ M_U)(p) \\ & = & \mathsf{Eval}_\alpha(U\,.\,p) \\ & = & \left(\sum_{i=1}^{n} U_i\alpha_i\right) p(\alpha) \\ & = & \mathsf{Eval}_\alpha(p)U(\alpha). \end{array}$$

This implies that $U(\alpha)$ is an eigenvalue of eigenvector Eval_α. Since $\alpha = \mathsf{Eval}_\alpha(1_K \bmod I)$, these eigenvectors are pairwise distinct and of cardinal $\#V(I)$. From the separability assumption, $\#V(I) = \dim_K(K[X_1,\ldots,X_n]/I)$ hence all the eigenvectors are of the form Eval_α. It follows that all the eigenvalues of M_U (which are the same as $\widehat{M_U}$) are $U(\alpha)$ for $\alpha \in V(I)$.

1.1.2 Primitive element representation

This kind of representation is commonly attributed to Kronecker, Macaulay in [82] calls *Kronecker substitution*, the specialization by a separating linear form as done in Lemma 1.2. The *Shape Lemma representation* (1.3) was first considered in computer algebra, actually coming from numerical analysis work of Auzinger-Stetter [10]. Then after the remark on the size of coefficients made in Alonso-Becker-Roy-Wörmann [3], the alternative equivalent representation of Definition 1.2 is nowadays preferred.

Main algorithms to compute it are implemented by Rouillier [101] (the RUR, following the ideas presented in [3]), relying on a Gröbner basis pre-computation, and to Lecerf [76]. This last implementation follows the *Geometric Resolution* algorithm resulting on a long and exacting task initiated by Giusti and Heintz [48] in the aim to have a subexponential solver of polynomial systems. With their collaborators Krick, Morgenstern, Montaña, Morais and Pardo, they carry on in the 90's this work in the series of articles [50, 51, 49]. In [52], Giusti, Lecerf and Salvy removed some constraining assumptions of regularity. For more details we refer to the thesis of Schost [102], Ch.1 §1.1, and Lecerf [75] Ch.1 §I.5.

Both algorithm can take into account multiplicity numbers. The algorithm [76] treats the positive dimension in the equidimensional situation. The Rouillier's approach is improved by Noro-Yokoyama in [92] by using the Chinese Remaindering Theorem. The algorithm of Lecerf relies him on lifting techniques with the use of a formal Newton-Hensel operator.

Primitive element representations are not unique, rely on the choice of a *separating linear form*:

Definition 1.1. *A linear form (i.e. homogeneous polynomial of degree 1) $\Delta(X_1, \ldots, X_n)$ is a separating linear form for V if and only if $\Delta(\alpha) \neq \Delta(\beta)$ for all $\alpha \neq \beta \in V$.*

So let Δ be such a form for V and consider $\chi_\Delta(T)$ the characteristic polynomial of the endomorphism of multiplication by Δ:

$$\begin{array}{rcl} M_\Delta \,:\, K[X_1,\ldots,X_n]/I(V) & \stackrel{\times\Delta}{\longrightarrow} & K[X_1,\ldots,X_n]/I(V) \\ P \bmod I(V) & \longmapsto & \Delta \cdot P \bmod I(V). \end{array}$$

The definition of a separating linear form implies a one-one correspondence between the points of V and the roots of χ_Δ. In fact Lemma 1.1 says that the roots of χ_Δ are the $\{\Delta(\alpha)\}_{\alpha\in V}$. More precisely, we have the following isomorphism:

Proposition 1.1. *With the notation above, the following map is an isomorphism of K-algebras:*

$$\begin{array}{rcl} K[T]/(\chi_\Delta(T)) & \longrightarrow & K[X_1,\ldots,X_n]/I(V), \\ T \bmod \chi_\Delta & \longmapsto & \Delta \bmod I(V). \end{array} \tag{1.1}$$

Proof: This map is clearly an homomorphism of K-algebras. Let $P(T) \in K[T]$ such that $P(\Delta(X_1,\ldots,X_n)) \in I(V)$. This implies that for every polynomial $Q \in K[X_1,\ldots,X_n]$, $P(\Delta)\cdot Q$ belongs to $I(V)$, or

$$\forall Q \in K[X_1,\ldots,X_n]\,, \quad P(M_\Delta)\,(Q \bmod I(V)) = 0 \qquad \text{in } K[X_1,\ldots,X_n]/I(V)$$

The endomorphism $P(M_\Delta)$ is the null endomorphism. It follows that $P \in (\chi_\Delta)$ and that the map (1.1) is injective. The separability assumption implies the following equality of the dimensions:

$$\dim_K \; K[T]/(\chi_\Delta) = \deg \chi_\Delta = \dim_K \; K[X_1,\ldots,X_n]/I(V),$$

permitting to prove that the map (1.1) is also onto. □

We can go further, by describing the roots of χ_Δ in function of the solutions of any system generating I. Let us explain the geometry behind this correspondence, in the case of a real number field $K \subset \mathbb{R}$, and where $V \subset \mathbb{A}^n_{\mathbb{R}}$. Denote by $(\,.\,,\,.\,)$ the usual Euclidean scalar product on $\mathbb{R}^n \simeq \mathbb{A}^n_{\mathbb{R}}$. Let us denote by L the line orthogonal for $(\,.\,,\,.\,)$ to the hyperplane H defined by the linear form Δ and going through the origin. Figure 1.1 hereunder shows the geometric meaning of χ_Δ: its roots parametrize the projection of V on L.

Proposition 1.2. *Let $\alpha \in V$. Then the value $\Delta(\alpha)$ is the coordinate, on the axis held by the line L, of the orthogonal projection on L along H of α (as drawn on the Figure (1.1)).*

PROOF: Let us write $\Delta = \delta_1 X_1 + \cdots + \delta_n X_n$. Then the vector $\vec{\delta} = (\delta_1, \ldots, \delta_n)$ is by definition orthogonal to the hyperplane H:

$$\alpha \in H \Leftrightarrow \Delta(\alpha) = 0 \Leftrightarrow (\vec{\alpha}\,,\, \vec{\delta}) = 0.$$

Consider now $\alpha \in V$. The orthogonal projection of α over L along H is then the extremity of the vector $(\vec{\alpha}\,,\, \vec{\delta})\vec{\delta}$. Hence, this vector is $\Delta(\alpha)\vec{\delta}$. By definition of $\vec{\delta}$, this means that $\Delta(\alpha)$ is the coordinate on the line L of the orthogonal projection of α on L along H. □

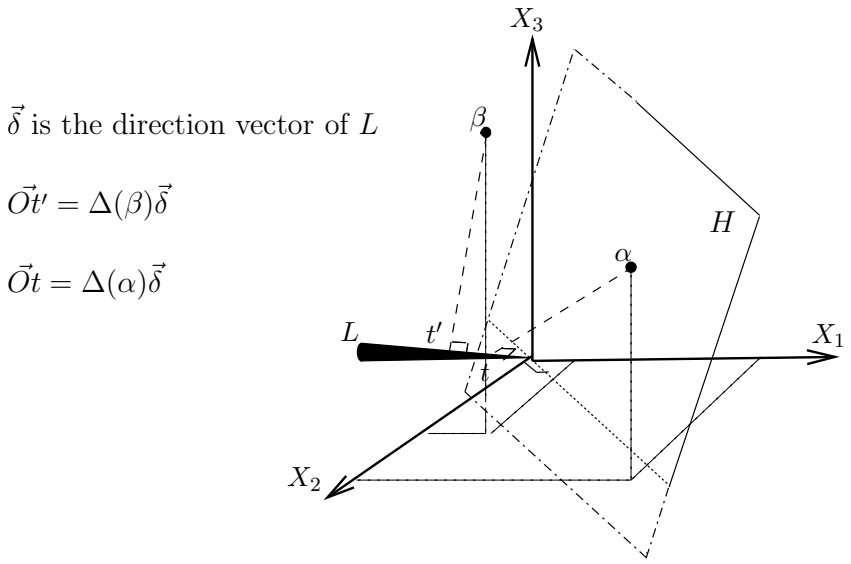

Figure 1.1: The orthogonal projection along H of two points α and β over L

We go back to the general situation where K is not necessarily contained in $\mathbb{R}$. Let $(\alpha_1, \ldots, \alpha_n)$ be the coordinates of α. We denote by W_i the following Lagrange interpolation polynomial for each value $1 \leq i \leq n$,

$$W_i(T) = \sum_{\alpha \in V} \alpha_i \prod_{\substack{\beta \in V \\ \beta \neq \alpha}} \frac{(T - \Delta(\beta))}{\Delta(\alpha) - \Delta(\beta)}. \tag{1.2}$$

So that, for every point $\alpha \in V$ and for $1 \leq i \leq n$, we have: $\alpha_i = W_i(\Delta(\alpha))$, yielding:

$$X_i \equiv W_i(\Delta) \bmod I(V).$$

The following representation due to Auzinger-Stetter [10] and called *Shape Lemma* representation of V by Lakshman, is the data of:

$$\chi(T), \quad \begin{cases} X_n - W_n(T) \\ \vdots \\ X_1 - W_1(T). \end{cases} \tag{1.3}$$

This Isomorphism (1.1) shows that $\{1, \Delta, \Delta^2, \ldots, \Delta^{\deg \chi - 1}\}$ is a basis of the K-vector space $K[X_1, \ldots, X_n]/I(V)$. Since $\deg W_i \leq \deg \chi - 1$, it follows that $W_i(\Delta)$ is the expression of X_i in this basis. Thus the polynomials W_i have coefficients in K.

Definition 1.2. *Denote by $w_i(T)$ the polynomial $W_i(T) \cdot \chi'_\Delta \bmod \chi_\Delta$. Then the following representation of V is called the Rational Univariate Representation, or the Kronecker representation:*

$$\chi_\Delta(T), \quad \begin{cases} \chi'_\Delta X_n - w_n(T) \\ \vdots \\ \chi'_\Delta X_1 - w_1(T). \end{cases}$$

Corollary 1.1. *For $1 \leq i \leq n$, the polynomials $w_i(T)$ defined above verify:*

$$w_i(T) = \sum_{\alpha \in V} \alpha_i \prod_{\substack{\beta \in V \\ \beta \neq \alpha}} (T - \Delta(\beta))$$

Proof: Let us denote $t_\alpha = \Delta(\alpha)$ for $\alpha \in V$. Since $\chi'_\Delta(T) = \sum_{\alpha \in V} \prod_{\beta \neq \alpha}(T - \Delta(\beta))$, it follows that $\chi'_\Delta(t_\alpha) = \prod_{\beta \neq \alpha}(t_\alpha - \Delta(\beta))$. Using interpolation formula (1.2), it follows that $\chi'_\Delta(t_\alpha) W_i(t_\alpha) = \alpha_i \prod_{\beta \neq \alpha}(T - \Delta(\beta))$. Hence, Definition 1.2 implies that $w_i(t_\alpha) = \alpha_i \prod_{\beta \neq \alpha}(t_\alpha - \Delta(\beta))$. It follows that both side of the equality we want to prove agree modulo χ_Δ. As both polynomials have the same degree and are monic, this implies that they are equal. □

1.1.3 Triangular systems

The notion of *characteristic sets*, close to the one of triangular sets, is commonly attributed to J. F. Ritt [100, 99], who introduced it in the differential algebra context. Since, many authors have proposed similar approaches, aiming at describing the zeros of an algebraic system through a finite family of *triangular sets*: Wu Wen-Tsun [120], D. Lazard [72, 71], M. Kalkbrener [62], D. Wang [118], M. Moreno Maza [88] as well as the dynamic evaluation school, notably D. Duval, T. Gomez-Diàs and S. Deillère. [37, 53, 34]. The algorithm proposed by Lazard in [71] is not proved, and is actually not correct in this article. The one of Gomez-Diàs [53] is implemented but not proved. Concerning regular chains (defined hereafter), the only proved algorithm and describing all the zeros of the input algebraic system is the one of Moreno Maza [80], implemented in the computer algebra systems Maple (`RegularChains` library) and in Axiom and Aldor (Triade algorithm).

The articles of Aubry, Lazard and Moreno-Maza [7, 8], and the thesis of Deillère [34], classify and compare the existing different approaches of "triangular systems". The notes of Hubert [60, 59] are emphasized on the parallel between the algebraic and differential cases. In dimension zero, where this work only deals with, different notions are coinciding. In positive dimension, these notions extends, and differences take place: the "Kalkbrener" decompositions [62, 6] only describe a dense open set of the variety considered, whereas the "Lazard decompositions" [71, 85] describe the whole variety.

The *dynamic evaluation* paradigm [36, 53, 34] permits to handle computations with algebraic numbers, eventually depending on parameters, by automatically managing the splittings which are occurring, and carrying on the computations in the different branches. This method comes from in questions of treatment of algebraic numbers [36]. For applications to triangularization of algebraic systems, we refer to [53, 34], and for a didactic approach to [37].

We begin by defining the more general notion of *regular chains* following [20]. Consider a polynomial ring $A[X_1, \ldots, X_n]$ over an unitary commutative ring A as well as a lexicographic order $\prec$ on the variables.

- The greatest variable of a polynomial P is called the *main variable* and is denoted by $\mathsf{mvar}(P)$.
- The coefficient of P with respect to its main variable is a polynomial involving smaller variables, called the *initial* and denoted by $\mathsf{init}(P)$.
- For $s \leq n$, consider the family of polynomial, $\mathbf{C} = C_1, \ldots, C_s \in A[X_1, \ldots, X_n]$ with $\mathsf{mvar}(C_i) = X_{\ell_i}$ and $X_{\ell_1} \prec X_{\ell_2} \prec \cdots \prec X_{\ell_s}$.
- Let h_i be the initial of C_i.
- The *i-th saturated ideal* of $\mathbf{C}$ denoted $\mathsf{Sat}_i(\mathbf{C})$ is the ideal $(C_1, \ldots, C_i) : (h_1 \cdots h_i)^\infty$. The n-th saturated ideal is simply denoted by $\mathsf{Sat}(\mathbf{C})$.

Definition 1.3 (Regular chain)**.** *The family of polynomials* $\mathbf{C}$ *above is a regular chain, if for all* $2 \leq i \leq s$, h_i *is a non-zero divisor in* $(A[X_1, \ldots, X_n]/\mathsf{Sat}_{i-1}(\mathbf{C}))$. *The set* $W(\mathbf{C}) = V(\mathbf{C}) \setminus V(h_1 \cdots h_n)$ *is called the* quasi-component *of* $\mathbf{C}$. *It verifies* $\overline{W(\mathbf{C})} = V(\mathsf{Sat}(\mathbf{C}))$.

Example: Assume that the ring A is a field K and consider the system in $K[X_1, X_2, X_3]$ for the order $X_1 < X_2 < X_3$:

$$\left| \begin{array}{l} C_2 = (X_1 + X_2)X_3^2 + X_3 + 1 \\ C_1 = X_1^2 + 1 \end{array} \right. \qquad \begin{array}{l} \mathsf{mvar}(C_1) = X_1\,, \quad \mathsf{mvar}(C_2) = X_3 \\ h_1 = \mathsf{init}(C_1) = 1\,, \quad h_2 = \mathsf{init}(C_2) = X_1 + X_2 \\ \mathsf{Sat}_1(\mathbf{C}) = (C_1) : h_1 = (C_1) \\ \mathsf{Sat}_2(\mathbf{C}) = (C_1, C_2) : (X_1 + X_2)^\infty \end{array}$$

The system above is a regular chain since $h_2 = X_1 + X_2$ is a non-zero divisor of the algebra $K[X_1, X_2]/(X_1^2 + 1)$. □

Given an ideal $I \subset K[X_1, \ldots, X_n]$, a subset of variables $Y_1, \ldots, Y_s \subset \{X_1, \ldots, X_n\}$ is *free* for I if $I \cap K[Y_1, \ldots, Y_s] = (0)$.

Theorem 1.1. *The ideal generated by* $\mathsf{Sat}(\mathbf{C})$ *is equidimensional, and if* $\mathfrak{p}$ *is an associated prime of* $\mathsf{Sat}(\mathbf{C})$*, then* $\dim \mathfrak{p} = n - \#\mathbf{C}$.

The variables X_i *which are not main variables of* $\mathbf{C}$ *are free variables. We call them the* canonical set of free variables *associated to* $\mathbf{C}$.

Proof: It comes from [20], Theorem 1. □

In the previous example, $\{X_2\}$ is the set of canonical free variables for $\mathbf{C}$.

Theorem 1.2. *Let* $\mathfrak{p}$ *a prime ideal of codimension* $n - d$. *A subset* $\mathbf{Y} = \{Y_1, \ldots, Y_d\} \subset \{X_1, \ldots, X_n\}$ *is a maximal set of free variables for* $\mathfrak{p}$ *if and only if there exists a regular chain* $\mathbf{R} = R_1, \ldots, R_s$ *with* $\mathfrak{p}$ *as saturated ideal in* $K[X_1, \ldots, X_n]$ *and with* $\mathbf{Y}$ *as canonical set of free variables.*

PROOF: Assume first that $\mathbf{Y}$ is a maximal set of free variables for $\mathfrak{p}$. Let us order the variables of $\mathbf{X}$ such that every variable of $\mathbf{Y}$ is smaller than every variable of $\mathbf{X} - \mathbf{Y}$. Let G be the reduced lexicographical Gröbner basis of $\mathfrak{p}$ w.r.t this order. By hypothesis, no polynomials of G lies in $K[\mathbf{Y}]$. By virtue of Theorem 3.2 in [7] one can extract from G a Ritt characteristic set $\mathbf{C}$ of $\mathfrak{p}$. Moreover, Theorems 3.3 and 6.1 in [7] show that $\mathbf{C}$ is a regular chain. Clearly, no variables in $\mathbf{Y}$ is the main variable of a polynomial in $\mathbf{C}$. Moreover, from Theorem 3.1 in [63] we have $d = n - \#\mathbf{C}$. Hence, $\mathbf{Y}$ is the canonical set of free variables of $\mathbf{C}$.

Conversely, let us assume now that there exists a regular chain $\mathbf{R} = R_1, \ldots, R_s$ with $\mathfrak{p}$ as saturated ideal and $\mathbf{Y}$ as canonical set of free variables. We can order the variables such that every variable of $\mathbf{Y}$ is smaller than every variable of $\mathbf{X} - \mathbf{Y}$ while preserving the fact that $\mathbf{R}$ is a regular chain for this new variable order. Then, it follows from Theorem 1 in [20] that $K[\mathbf{Y}] \cap \mathfrak{p}$ equals the trivial ideal, which shows that $\mathbf{Y}$ is a maximal set of free variables for $\mathfrak{p}$, concluding the proof. □

The definition of triangular set we give is specific to this thesis, and is a particular case of Lazard triangular sets defined for example in [7]. It is restrictive to the dimension zero.

Definition 1.4 (Triangular set)**.** *A triangular set* $\mathbf{T} = (T_1, \ldots, T_n)$ *is a regular chain for the order* $X_1 < \cdots < X_n$ *verifying* $\mathsf{init}(T_i) = 1$*. We ask that the ideal generated by* $\mathbf{T}$ *verifies the Separability Assumption. It is then a lexicographic Gröbner basis, that we assume to be reduced.*

REMARK: According to the definition of regular chains 1.3, triangular sets can be defined over a ring A. Such triangular sets are useful for defining the Newton operator over rings such as $\mathbb{Z}/p^{2^\kappa}$. Else, all triangular sets considered will be defined over a field K (and as usual, finite extension of $\mathbb{Q}$ or of $k(p_1, \ldots, p_m)$).

Positive dimension We suppose that we are given a triangular set $\mathbf{T}$ set with coefficients in $k(p_1, \ldots, p_m)$. The zero set of $\mathbf{T}$ in $\overline{k(p_1, \ldots, p_m)}$ is denoted by V. Dividing out the denominators, yields a regular chain $\mathbf{t}$ lying in $k[p_1, \ldots, p_m, X_1, \ldots, X_n]$. The variables $p_1, \ldots, p_m$ form the canonical set of free variables for $\mathbf{t}$. Let $\mathfrak{V} \subset \mathbb{A}^{n+m}_{\bar{k}}$ be the Zariski closure of the quasi-component of $\mathbf{t}$, i.e. $\mathfrak{V} = \overline{W(\mathbf{t})}$. Theorem 1.1 states that $\mathfrak{V}$ is equidimensional of dimension m. We have moreover:

$$\deg(V) \leq \deg(\mathfrak{V}) \tag{1.4}$$

There is a non-trivial relation between regular chains and triangular sets that we describe here:

Theorem 1.3. *Let* $\mathbf{C} = C_1, \ldots, C_n$ *a regular chain in* $k[p_1, \ldots, p_m, X_1, \ldots, X_n]$ *such that* $\mathsf{mvar}(C_i) = X_i$ *for* $1 \leq i \leq n$*, so that the canonical set of free variables for* $\mathbf{C}$ *is* $p_1, \ldots, p_m$*. Then, for* $2 \leq \ell \leq n$*,* $\mathsf{init}(C_\ell)$ *is invertible in the ring:*

$$R_{\ell-1} = k(p_1, \ldots, p_m)[X_1, \ldots, X_{\ell-1}]/\mathsf{Sat}_{\ell-1}(\mathbf{C}).$$

This is Theorem 3 of [20]. In particular, it is possible to construct a triangular set with coefficients in the function field with variables the canonical set of free variables associated to $\mathbf{C}$, by a modular inversion process.

Under the Separability Assumption and radicality, the variety described by $\mathbf{T}$ verifies a nice geometric property, called *equiprojectable*, introduced by Aubry-Valibouze in [9]. Before, let us define some projectors that will be used all along this thesis.

Definition 1.5. *Given some integers $1 \leq j \leq \imath \leq n$, let π_i^n be the projection:*

$$\begin{array}{rcl} \pi_i^n \ : \ \mathbb{A}_{\bar{K}}^n & \longrightarrow & \mathbb{A}_{\bar{K}}^i \\ (\alpha_1, \ldots, \alpha_n) & \longmapsto & (\alpha_1, \ldots, \alpha_i). \end{array}$$

We note that $\pi_j^i \circ \pi_i^n = \pi_j^n$.

Definition 1.6 (Equiprojectable variety)**.** *A finite set of points $V \subset \mathbb{A}_{\bar{K}}^i$ is said to be i-equiprojectable if either $i = 1$, or $i > 1$ and $\pi_{i-1}^i(V)$ is $i-1$ equiprojectable and*

$$\#(\pi_{i-1}^i)^{-1}(\{\alpha\}) = \#(\pi_{i-1}^i)^{-1}(\{\beta\}), \quad \textit{for each} \quad \alpha, \beta \in \pi_{i-1}^i(V).$$

Finally, a finite set of points $V \in \mathbb{A}_{\bar{K}}^n$ is said to be equiprojectable if it is n-equiprojectable.

The main result for triangular set is the following result due to Aubry and Valibouze [9], Theorem 4.5.

Theorem 1.4. *Let $\mathbf{T}$ be a radical triangular set defined over K with $d_i := \deg_{X_i}(T_i)$. The zero set $V = V(\mathbf{T}) \in \mathbb{A}_{\bar{K}}^n$ is equiprojectable. Moreover, $d_1 = \#\pi_1^n(V)$ and, for $i \geq 2$ the cardinality $\#(\pi_{i-1}^i)^{-1}(\pi_i^n(\{\alpha\}))$ for each $\alpha \in V$ is equal to d_i, for $i \geq 2$.*

Here are some pictures illustrating those definitions.

Example: The picture on the right shows an equiprojectable variety, described by a triangular set T_1, T_2, T_3 of degrees $(d_1, d_2, d_3) = (1, 1, 3)$.

The picture in the middle shows a non equiprojectable variety since $\#(\pi_1^2)^{-1}(\{A\}) = 2$, whereas $\#(\pi_1^2)^{-1}(\{B\}) = 1$ (so it is not 2-equiprojectable).

The variety V drawn on the left is also not equiprojectable: in fact the fiber $(\pi_2^3)^{-1}(\{D\})$ over D has cardinality 2, but the other fibers over A, B and C have cardinality 3. However, the projection $\pi_2^3(V)$ on the X_1, X_2-axes is equiprojectable, hence V is 2-equiprojectable.

1.2 Chow form and height

Height theory is a long time studied mathematical subject. For the height of varieties, different notions exist; we refer to the introductory slides of Silverman [108] for a survey of the subject. In computer algebra and effective algebra, it appears that the height of Philippon relying on the Chow form reveals to be the most used [66, 95]. We perpetuate this "tradition" in this work.

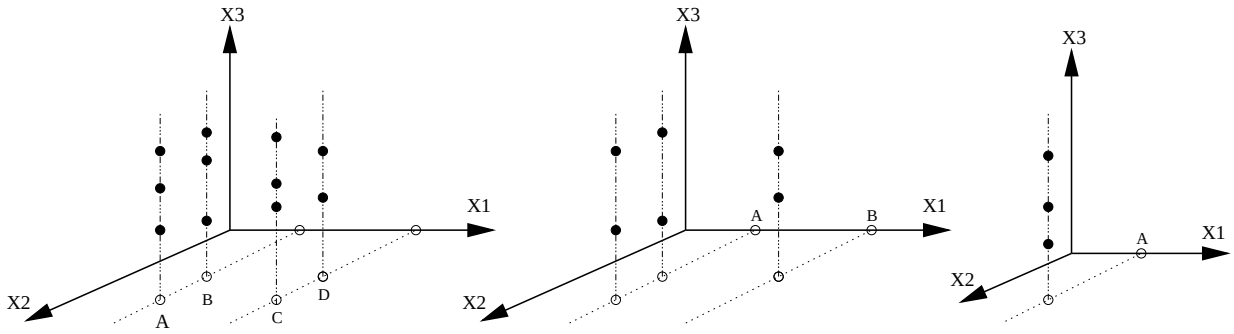

Figure 1.2: Example of equiprojectable and non equiprojectable varieties

1.2.1 Chow form

The notion of Chow form can exist for positive equidimensional ideals. However, we only deal with the 0-dimensional case, where it is much easier to define. For a general treatment see [66, 94].

So suppose we are given a 0-dimensional variety V defined over K. Let $I = I(V)$ be the ideal of polynomials vanishing on V. Let us introduce some new indeterminates $U_1, \ldots, U_n$; the notation $K[\mathbf{U}]$ may be used instead of $K[U_1, \ldots, U_n]$ and $K[\mathbf{X}]$ instead of $K[X_1, \ldots, X_n]$. This following scalar extension is useful

$$(K[\mathbf{X}] \otimes K[\mathbf{U}])/(I \otimes K[\mathbf{U}]) \simeq (K[\mathbf{X}]/I) \otimes K[\mathbf{U}] \simeq K[\mathbf{X}, \mathbf{U}]/IK[\mathbf{X}, \mathbf{U}].$$

If $\mathfrak{p}$ is an associated (respectively minimal) prime of I, then so is it of $\mathfrak{p} \otimes K(\mathbf{U})$ for $I \otimes K(\mathbf{U})$. For each such prime, the extension $K[\mathbf{X}]/\mathfrak{p}$ is then generated over K by a separable element $a_{\mathfrak{p}}$. Hence the extension $K[\mathbf{X}]/\mathfrak{p} \otimes K(\mathbf{U})$ is also generated by $a_{\mathfrak{p}}$ over $K(\mathbf{U})$, meaning that this extension is separable. Consequently, if I verifies the Separability Assumption, $I \otimes K(\mathbf{U})$ verifies it also.

Proposition 1.3. *$K[\mathbf{U}] \otimes (K[\mathbf{X}]/I)$ is a free $K[\mathbf{U}]$ -module of rank the dimension of the K vector space $K[\mathbf{X}]$. Moreover if $p_1, \ldots, p_D$ is basis of $K[\mathbf{X}]/I$, then $p_1 \otimes 1_K, \ldots, p_D \otimes 1_K$ is a basis of $(K[\mathbf{X}]/I) \otimes K[\mathbf{U}]$.*

Proof: For $a \in K[\mathbf{X}]/I$, $\langle a \rangle$ denotes the sub-vector space of $K[\mathbf{X}]/I$ generated by a. Let us prove the following isomorphism, from which the claim is deduced immediately, since $\oplus_{i=1}^{D} \langle p_i \rangle = K[\mathbf{X}]/I$:

$$\left(\oplus_{i=1}^{D} \langle p_i \rangle\right) \otimes_K K[\mathbf{U}] \simeq \oplus_{i=1}^{D} (\langle p_i \rangle \otimes_K K[\mathbf{U}]).$$

In fact there is a K-bilinear map:

$$\begin{aligned} (\oplus_i \langle p_i \rangle) \times K[\mathbf{U}] &\longrightarrow \oplus_i (\langle p_i \rangle \otimes_K K[\mathbf{U}]) \\ ((a_1 p_1, \ldots, a_D p_D), P) &\longmapsto \sum_{i=1}^{D} (a_i p_i \otimes_K P) \end{aligned}$$

permitting to define the map $(\oplus_i \langle p_i \rangle) \otimes K[\mathbf{U}] \to \oplus_i (\langle p_i \rangle \otimes K[\mathbf{U}])$, which admits the reciprocal map:

$$\begin{array}{rcl} \oplus_{i=1}^{D} (\langle p_i \rangle \otimes_K K[\mathbf{U}]) & \longrightarrow & \left(\oplus_{i=1}^{D} \langle p_i \rangle\right) \otimes_K K[\mathbf{U}] \\ (p_1 \otimes R_1, \ldots, p_D \otimes R_D) & \longmapsto & \displaystyle\sum_{i=1}^{D} (0, \ldots, p_i, \ldots, 0) \otimes R_i \end{array}$$

□

Definition 1.7 (Chow form)**.** *Let $U := U_1 X_1 + \cdots + U_n X_n$. Consider the endomorphism M_U of multiplication by U in the $K[\mathbf{U}]$-module $(K[\mathbf{X}, \mathbf{U}]/IK[\mathbf{X}, \mathbf{U}])$. The characteristic polynomial $\det(T1 - M_U)$ is called the Chow form of V and is denoted $\mathcal{C}_V(U_1, \ldots, U_n, T)$.*

It is used in the following form:

Proposition 1.4. *The Chow form of V verifies the following identity:*

$$\mathcal{C}_V(U_1, \ldots, U_n, T) = \prod_{\alpha \in V} (T - \sum_{i=1}^{n} \alpha_i U_i).$$

Proof: From Separability Assumption, $\sum_{i=1}^{n} \alpha_i U_i \neq \sum_{i=1}^{n} \beta_i U_i$ as soon as $\alpha \neq \beta$. From Lemma 1.1 all the $\{\sum_{i=1}^{n} \alpha_i U_i\}_{\alpha \in V}$ are eigenvalues, which are pairwise distinct from the Separability Assumption. Moreover $\#V = \dim_{K[\mathbf{U}]}(K[\mathbf{U}, \mathbf{X}]/IK[\mathbf{U}, \mathbf{X}])$ by Proposition 1.3, concluding the proof. □

It follows immediately the *multiplicative property* of the Chow form. If V_1 and V_2 are disjoint varieties then:

$$\mathcal{C}_{V_1 \cup V_2} = \mathcal{C}_{V_1} \mathcal{C}_{V_2}. \tag{1.5}$$

Let us mention the easy but important cancellation identity:

Lemma 1.2. *If $U = U_1 X_1 + \cdots + U_n X_n$, then*

$$\mathcal{C}_V(U_1, \ldots, U_n, U) \equiv 0 \bmod K[\mathbf{X}, \mathbf{U}]/IK[\mathbf{X}, \mathbf{U}].$$

Proof: Let us reuse the notation M_U of Definition 1.7.

$$M_U(1_K \bmod IK[\mathbf{U}, \mathbf{X}]) = U \bmod IK[\mathbf{U}, \mathbf{X}],$$

so that

$$\mathcal{C}_V(U_1, \ldots, U_n, U) \equiv \mathcal{C}_V(U_1, \ldots, U_n, M_U(1_K)) \equiv \mathcal{C}_V(U_1, \ldots, U_n, M_U)(1_K) \bmod IK[\mathbf{U}, \mathbf{X}].$$

Finally the last term above is null due to Cayley-Hamilton's theorem: $\mathcal{C}_V(U_1, \ldots, U_n, M_U)$ is the null endomorphism in the $K[\mathbf{U}]$-module $K[\mathbf{X}, \mathbf{U}]/IK[\mathbf{X}, \mathbf{U}]$. □

Finally, using the fact that the notion of triangular set and equiprojectable variety is stable under projection discarding the last variables, we have:

Lemma 1.3. *Let V be an equiprojectable variety defined by a triangular set $T_1, \ldots, T_n$ over K of degrees $d_1, \ldots, d_n$. Let $\mathcal{C}_V$ be the Chow form of V, and denote by $\mathcal{C}_i$ the Chow form of $\pi_i^n(V)$ instead of $\mathcal{C}_{\pi_i^n(V)}$. Then for all $1 \leq i \leq n-1$, the following equality holds:*

$$\mathcal{C}_{i+1}(U_1, \ldots, U_i, 0, T) = \mathcal{C}_i(U_1, \ldots, U_i, T)^{d_{i+1}}.$$

Proof: From Proposition 1.4,

$$\mathcal{C}_{i+1}(U_1, \ldots, U_i, 0, T) = \prod_{\alpha \in \pi_{i+1}^n(V)} (T - U_1\alpha_1 - \cdots - U_i\alpha_i).$$

The factor $(T - U_1\alpha_1 - \cdots - U_i\alpha_i)$ appears $\#(\pi_i^{i+1})^{-1}(\{\alpha_1, \ldots, \alpha_i\})$ times, which is d_{i+1} from Theorem 1.4, independently of α. The proposition follows. □

1.2.2 Height theory

The literature is vast on this topic, and we refer to one of the numerous books of Diophantine Approximation or Geometry for a more general treatment (for example [67, 68, 58]) The notion of height relies on *absolute values* existing over a field. When the field presents no *Archimedean absolute* value, then the notion is easy and all the different approaches are essentially the same. Problems arise when the field presents such absolute values. An absolute value over a field K is an application

$$\begin{aligned} |.|_v : K &\longrightarrow \mathbb{R}+ \\ x &\longmapsto |x|_v, \end{aligned}$$

satisfying the standard properties:

(i) $|x|_v = 0$ if and only if $x = 0$.

(ii) $|x \cdot y|_v = |x|_v \cdot |y|_v$.

(iii) $|x + y|_v \leq |x|_v + |y|_v$.

If moreover the *ultrametric inequality* holds:

(iii)' $|x + y|_v \leq \max\{|x|_v, |y|_v\}$,

then $|.|_v$ is said *non-Archimedean*. Else it is said *Archimedean*.

In this work we will be interested in two families of fields: the number fields, i.e. finite extensions of $\mathbb{Q}$, and function fields, i.e. finite extensions of $k(p_1, \ldots, p_m)$, where p_i are parameters. The first family presents Archimedean absolute values, and the second one does not.

Definition 1.8. *In the sequel and all along this thesis, K_0 will refer either to the base field $\mathbb{Q}$, or to the base field $k(p_1, \ldots, p_m)$; the finite extension of K_0 considered will be denoted by K.*

The results presented in Chapter 2 are easier and nicer in the function field case. The height measures the *arithmetic complexity* of a rational number, the primitive roots of unity being yardsticks: their height is in fact null. For the functional case, height measures the degree of divisors. These measures are particularly clear for rational numbers and rational functions: number of digits, and degree in the parameters respectively.

Absolute values over the rational number and function fields. Let $x = a/b \in \mathbb{Q}$, a and b being relatively prime integers. We denote by $|.|_\infty$ the usual absolute value, i.e. $|x|_\infty = \max\{x, -x\}$; it is Archimedean. Let p be a prime number. Denote by $v_p(a)$ the exponent of p in the decomposition of the integer a in prime numbers (i.e. $p^{v_p(a)}|a$ but $p^{v_p(a)+1} \nmid a$). We define by $|.|_p$ the application:

$$\begin{array}{rcl} |.|_p : \mathbb{Q} & \longrightarrow & \mathbb{R}+ \\ x & \longmapsto & p^{v_p(b)-v_p(a)} \end{array}$$

This defines a non-Archimedean absolute value over $\mathbb{Q}$.

In the same way, consider a rational function $F = A/B$, with A, B relatively prime polynomials in $k[p_1, \ldots, p_m]$. There is a natural absolute value,

$$|F|_\infty = e^{\deg A - \deg B},$$

which is *not* Archimedean. Additionally, for an irreducible polynomial $P \in k[p_1, \ldots, p_m]$, let $v_P(A)$ be the exponent of P appearing in the factorization of A. The following application $|.|_P$ is a non-Archimedean absolute value:

$$\begin{array}{rcl} |.|_P : k(p_1, \ldots, p_m) & \longrightarrow & \mathbb{R}+ \\ F & \longmapsto & e^{\deg P(v_P(B)-v_P(A))} \end{array}$$

In the sequel, when we speak of the set of absolute values over $K_0 = \mathbb{Q}$ or $K_0 = k(p_1, \ldots, p_m)$, we always mean the set of absolute values as above. We denote this set by $M_{K_0} = (M^0_{K_0}, M^\infty_{K_0})$, where $M^0_{K_0}$ are the non-Archimedean ones, and $M^\infty_{K_0}$ are the Archimedean ones (when $K_0 = k(p_1, \ldots, p_m)$ then $M^\infty_{K_0} = \emptyset$). Consider a field L with a set of absolute values $M_L = (M^0_L, M^\infty_L)$. We say that M_L satisfies the *product formula with multiplicity* m_v if we have:

$$\prod_{v \in M_L} |x|_v^{m_v} = 1 \qquad \text{for all} \quad x \in L, \quad x \neq 0.$$

For fields L endowed with such a family of absolute values, it is possible to define the *height* of an element x of F:

$$h(x) = \sum_{v \in M_L} m_v \log \max\{1, |x|_v\}.$$

When $L = \mathbb{Q}$, the set of absolute values defined previously verifies the product formula with multiplicity one; if $x = a/b$ then the height of x is nothing else that:

$$h(x) = \log \max\{|a|_\infty, |b|_\infty\}.$$

Hence $h(x)$ bounds the number of digits of its numerator and denominator. When $L = k(p_1, \ldots, p_m)$, the set of absolute values defined previously also verifies the product formula with multiplicity one; if $F = A/B$ is:

$$h(F) = \max\{\deg A, \deg B\}.$$

Fields extension. Consider now a finite extension K of the base field K_0. An absolute value v of M_{K_0} defines a metric space on K_0, so the concept of Cauchy sequences and completion make sense. We denote by K_{0v} the completion of K_0 for the metric induced by v. Let w be an absolute value over K extending $v \in M_{K_0}$. Then the fields extension $K_w|K_{0v}$ is finite. Let $\mathbb{C}_v$ be the completion of the algebraic closure of K_{0v} ($\mathbb{C}_v$ is also algebraically closed). There is an embedding:

$$\sigma_w : K \to \mathbb{C}_v, \tag{1.6}$$

of K as a subfield of $\mathbb{C}_v$. Then $w \in M_K$ is defined by $|x|_w = |\sigma_w(x)|_v$, for all $x \in K$.

For each absolute value w of K, we have what we call the *local degree* $N_w = [K_w : K_{0v}]$, and the *degree formula* [58, Proposition B.1.1], holds:

$$\sum_{w \in M_K \,, w|v} [K_w : K_{0v}] = [K : K_0], \tag{1.7}$$

where the symbol $w|v$ means that the restriction of w to K_0 is v. As a result, it follows that the set of absolute values M_K satisfies the product formula with multiplicity N_w, namely:

$$\prod_{w \in M_K} |x|_w^{N_w} = 1 \qquad \text{for all} \quad x \in K, \quad x \neq 0.$$

It is therefore possible to define the height of an element of K:

$$h(x) := \frac{1}{[K : K_0]} \sum_{w \in M_K} N_w \log \max\{1, |x|_w\}$$

Height of polynomials. Let f be a polynomial in $K[X_1, \ldots, X_n]$, where K is a field endowed with a family of absolute values satisfying the product formula with multiplicity N_v. Denote by $\mathbf{X^a}$ the monomial $X_1^{a_1} \cdots X_n^{a_n}$ for a n-uple $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$. Write the polynomial f in the following way:

$$f = \sum_{\mathbf{a} \in \mathbb{N}^n} f_{\mathbf{a}} \mathbf{X^a}, \qquad \text{where the } f_{\mathbf{a}} \in K \text{ are almost all zero.}$$

Let v be an absolute value in M_K. The following notation is convenient in the sequel:

$$\log |f|_v := \log\{\max_{\mathbf{a} \in \mathbb{N}^n}\{|f_{\mathbf{a}}|_v\}\}. \tag{1.8}$$

We define the *local height* of f:

$$h_v(f) = \max\{0, \log |f|_v\}.$$

Then the height of a polynomial f is the sum of its local heights:

$$h(f) := \frac{1}{[K : K_0]} \sum_{v \in M_K} N_v h_v(f).$$

We note that if f has its coefficients in K_0, then the height of f defined over K and defined over K_0 coincide. Let v be an Archimedean absolute value. Then in the embedding (1.6)

σ_v, the image $\mathbb{C}_v$ is the complex field $\mathbb{C}$, endowed with its usual norm. Extending σ_v to the polynomial rings over K, we define the *Mahler measure* of $f \in K[X_1, \ldots, X_n]$:

$$m(\sigma_v(f)) := \int_0^1 \ldots \int_0^1 \log |\sigma_v(f)(e^{2i\pi t_1}, \ldots, e^{2i\pi t_n})| dt_1 \ldots dt_n,$$

and the S_n-Mahler measure of f (integration is made on the sphere and no more on the torus):

$$m(\sigma_v(f); S_n) := \int_{S_n} \log |\sigma_v(f)| \mu_n,$$

where S_n is the complex sphere of dimension n, μ_n is the Haar measure over S_n. It is immediately seen that both quantities are additive.

We conclude this paragraph by giving useful inequalities for the height of polynomials over $K_0 := \mathbb{Q}$ and over $K_0 := k(p_1, \ldots, p_m)$, showing that this notion is relevant to space complexity. The ring of integers of a field K endowed with a family of absolute value M_K is the ring R equal to:

$$R = \bigcap_{v \in M_K} \{x \in K, \text{ such that } |x|_v \leq 1\}$$

Proposition 1.5. *Let $P \in K_0[X_1, \ldots, X_n]$, c the lcm of the denominators of its coefficients. Then cP has its coefficients in the ring of integers of K_0 (so $\mathbb{Z}$ or $k[p_1, \ldots, p_m]$). We denote by C the set of the coefficient of cP. Then,*

$$h(P) = \log \max \left(\{|c|_\infty\} \cup \{|x|_\infty \,,\ x \in C\} \right),$$

where $|x|_\infty$ is $\max\{x, -x\}$ when $K_0 = \mathbb{Q}$, and is $\deg(x)$ when $K_0 = k(p_1, \ldots, p_m)$.

Proof: By definition, if $v \neq \infty$ then $h_v(cP) = 0$. Hence

$$h(cP) = h_\infty(cP) = \log \max\{|x|_\infty \,,\ x \in C\},$$

since the coefficients $x \in C$ are in the ring of integers, hence $|x|_\infty \geq 1$. Moreover, taking the maximum in the equality $|y|_v = |cy|_v \cdot |\frac{1}{c}|_v$, yields:

$$\log \max\{|y|_v \,,\ y \text{ coefficient of } P\} = \begin{cases} \log |\frac{1}{c}|_v + h(cP) & \text{if } v = \infty, \\ \log |\frac{1}{c}|_v & \text{if } v \neq \infty. \end{cases} \tag{1.9}$$

Let $v \neq \infty$. Since c is the lcm of the denominators of the coefficients of P, we have:

$$h_v(P) > 0 \Leftrightarrow \log |\frac{1}{c}|_v > 0.$$

Let $M^0_{K_0}(P) := \{v \in M^0_{K_0} \,,\ h_v(P) > 0\}$. We then have

$$h_v(P) = \log \max\{|x|_v \,,\ x \text{ coefficient of } P\},$$

for such a v, and with Equality (1.9):

$$\sum_{v \in M^0_{K_0}} h_v(P) = \sum_{v \in M^0_{K_0}(P)} h_v(P) = \sum_{v \in M^0_{K_0}(P)} \log |\frac{1}{c}|_v = \sum_{v \in M^0_{K_0}} \log |\frac{1}{c}|_v = \log |c|_\infty,$$

the last equality coming from the product formula. It follows that $h(P) = \log|c|_\infty + h_\infty(P)$. If $h_\infty(P) = 0$, then each coefficient x of P verifies $|x|_\infty \leq 1$, so $|cx|_\infty \leq |c|_\infty$ and the maximum in $\{|c|_\infty\} \cup \{|y|_\infty\,,\ y \in C\}$ is $|c|_\infty$. Since $h(P) = \log\{|c|_\infty\}$, this proves the proposition when $h_\infty(P) = 0$. Else $h_\infty(P) = \log\max\{|x|_\infty\,,\ x \text{ coefficient of } P\}$, yielding $h_\infty(P) = \log|\frac{1}{c}|_\infty + h_\infty(cP)$, and by Equality (1.9):

$$h(P) = \log|c|_\infty + \log|\frac{1}{c}|_\infty + h_\infty(cP) = h_\infty(cP).$$

Now $h_\infty(P) > 0$ implies that there exists a coefficient x of P, such that $|x|_\infty > 1$, so that $cx \in C$ gives $|cx|_\infty > |x|_\infty$; hence $|c|_\infty$ is not the maximum in the set of the proposition Consequently, in this case, we have $h(P) = h_\infty(cP) = \log(\{|c|_\infty\} \cup \{|x|_\infty\,,\ x \in C\})$. □

As an immediate corollary, with the notations of Paragraph "Positive dimension" after Definition 1.4, we have:

$$h(V) \leq \deg(\mathfrak{V}) \tag{1.10}$$

Height of varieties. We define here the height of a zero-set of a polynomial system over K (the same as above, a number field or a function field in m variables). In the case of dimension zero, the *Weil* height is commonly used, but we prefer to use the height of Philippon relying on the Chow form, as it appears quite naturally in our problem. Moreover, an extension to the positive dimensional case, where the Weil height is no more available, is foreseen. Let us mention the height of Bost-Gillet-Soulé [18], widely used in the mathematics's community. These two definitions of height coincide, as soon as the metric for the Archimedean absolute values is well chosen, as shown in [111, théorème 3] .

Let $V \subseteq \mathbb{A}^n_{\bar{K}}$ be a variety of dimension 0, $\mathcal{C}_V$ its Chow form. The height of the variety V, in the functional case (no Archimedean absolute values) is:

$$h(V) := \frac{1}{[K : K_0]} \sum_{v \in M_K^0} N_v h_v(\mathcal{C}_V), \tag{1.11}$$

and when K is a number field (with Archimedean absolute values) the height of V is:

$$h(V) := \frac{1}{[K : K_0]} \sum_{v \in M_K^0} N_v h_v(\mathcal{C}_V) + \frac{1}{[K : K_0]} \sum_{v \in M_K^\infty} N_v m(\sigma_v(\mathcal{C}_V); S_{n+1}) + \deg(V) \left(\sum_{i=1}^{n} \frac{1}{2i} \right). \tag{1.12}$$

See [66] for an explanation of the corrective term at the end, and for a discussion of the inequalities hereunder:

$$m(f) - \deg(f)\log(n+1) \leq \log|f| \leq m(f) + \deg(f)\log(n+1) \tag{1.13}$$

$$0 \leq m(f) - m(f; S_n) \leq \deg(f) \left(\sum_{i=1}^{n-1} \frac{1}{2i} \right) \tag{1.14}$$

The following straightforward corollary of these inequalities is useful in many situations.

Corollary 1.2. *Suppose that K is a number field, and $v = \infty$ is the Archimedean absolute value of $\mathbb{Q}$. We consider a variety V defined over K with Chow form $\mathcal{C}_V \in K[X_1, \ldots, X_n, T]$. Then, if $h_\infty(\,.\,) := \frac{1}{[K:\mathbb{Q}]} \sum_{w|\infty} [K_w : \mathbb{R}] h_w(\,.\,)$ we have:*

$$h_\infty(\mathcal{C}_V) \leq h_\infty(V) + \deg(V)\log(n+2).$$

PROOF: From Equality (1.13), $h_w(\mathcal{C}_V) \leq m(\sigma_w(\mathcal{C}_V)) + \deg(\mathcal{C}_V)\log(n+2)$. From Equality (1.14),

$$h_w(\mathcal{C}_V) \leq m(\sigma_w(\mathcal{C}_V); S_{n+1}) + \deg(\mathcal{C}_V)\left(\sum_{i=1}^{n}\frac{1}{2i} + \log(n+2)\right).$$

The degree formula (1.7) implies then:

$$h_\infty(\mathcal{C}_V) \leq \deg(\mathcal{C}_V)\left(\sum_{i=1}^{n}\frac{1}{2i} + \log(n+2)\right) + \frac{1}{[K:\mathbb{Q}]}\sum_{w|\infty}[K_w:\mathbb{R}]m(\sigma_w(\mathcal{C}_V); S_{n+1}).$$

We recognize the definition of the height of the variety:

$$\begin{aligned} h_\infty(\mathcal{C}_V) &\leq \deg(\mathcal{C}_V)\log(n+2) + \frac{1}{[K:\mathbb{Q}]}\sum_{w|v}[K_w:\mathbb{R}]h_w(V) \\ &\leq \deg(\mathcal{C}_V)\log(n+2) + h_\infty(V). \end{aligned}$$

We conclude with $\deg(\mathcal{C}_V) = \deg(V)$. □

A nice property of the height is that if V_1 and V_2 are disjoint varieties, following from equality (1.5):

$$h(V_1 \cup V_2) = h(V_1) + h(V_2).$$

The well known geometric Bézout theorem, bounding the degree of the intersection of two varieties has an arithmetic counterpart, due to Philippon, but we refer to § 2.2.2 of [66], closer to our notations.

Theorem 1.5. *Let $f_1, \ldots, f_s$ be a family of n-variate polynomials defined over K. We define:*

$$d := \max_{1\leq i\leq s}\{\deg(f_i)\} \qquad \text{and} \qquad h := \max_{1\leq i\leq s}\{h(f_i)\}.$$

The degree of $V = V(f_1, \ldots, f_s)$ is bounded by (geometric Bézout theorem):

$$\deg(V) \leq d^s.$$

Its height verifies the following inequality (arithmetic Bézout theorem):

$$h(V) \leq d^s\big(sh + (n+s)\log(n+1)\big)$$

These results will be useful to get extrinsic bounds in Table 2.1 p. 50.

Useful inequalities. We conclude by giving basic inequalities for local heights and Mahler measures. All the results of this section are taken from [66], §1.1. Except the two inequalities $\mathbf{A_2}$ and $\mathbf{A_6}$, coming from inequalities (1.13) and (1.14)discussed above, all the proofs are not difficult.

Let $f_1, \ldots, f_s$ be in $K[X_0, \ldots, X_n]$, f in $K[X_1]$, $g \in K[Y_1, \ldots, Y_s]$, and assume that each f_i has *at least one coefficient equal to 1* (this simplifying assumption is satisfied in the sequel). If v is an Archimedean absolute value on K, we have:

$\mathbf{A_1}$ $m(f_i) \geq 0$ if $\deg(f_i) = 1$.

A₂ $h_v(f_i) \leq m_v(f_i) + \log(n+2)\deg(f_i)$.

A₃ $h_v(f_1 \cdots f_s) \leq \sum_{i=1}^{s} h_v(f_i) + \log(n+2)\sum_{i=1}^{s}\deg(f_i)$.

A₄ $\sum_{i=1}^{s} h_v(f_i) \leq h_v(f_1 \cdots f_s) + 2\log(n+2)\sum_{i=1}^{s}\deg(f_i)$.

A₅ $h_v(f_1 + \cdots + f_s) \leq \max_{i\leq s} h_v(f_i) + \log s$.

A₆ $m_v(f_i) \leq m_v(f_i; S_{n+1}) + \deg(f_i)\left(\sum_{i=1}^{n} \frac{1}{2i}\right)$.

A₇ $h_v(f(x)) \leq h_v(f) + \deg(f)(h_v(x) + \log(2))$ for $x \in K$.

A₈ $m_v(f_i(X_0, \ldots, X_{n-1}, 0)) \leq m_v(f_i)$.

A₉ $h_v(g(f_1, \ldots, f_s)) \leq h_v(g) + \deg g(\max_{i\leq s} h_v(f_i) + \log(s+1) + \max_{i\leq s}\{\deg(f_i)\}\log(n+1))$

If v is a non-Archimedean absolute value on K, we have:

N₁ $h_v(f_1 \cdots f_s) = h_v(f_1) + \cdots + h_v(f_s)$.

N₂ $h_v(f_1 + \cdots + f_s) \leq \max_{i\leq s} h_v(f_i)$.

N₃ $h_v(f(x)) \leq h_v(f) + \deg(f)h_v(x)$ for $x \in k$.

If we drop the assumption that each f_i has one coefficient equal to 1, we still have, for any absolute value v:

E $h_v(xf_i) \leq h_v(x) + h_v(f_i)$ for $x \in K$.

Corollary 1.3. *Let M be a $s\times s$ matrice of polynomials $(f_{i.j})_{1\leq i,j\leq s}$, with $d = \max_{i,j}\{\deg(f_{i,j})\}$ and $h_\infty = \max_{1\leq i,j\leq s} h_\infty(f_{i,j})$. Then,*

$$h_\infty(\det(M)) \leq s(h_\infty + \log s + d\log(n+1)).$$

1.3 Basic algorithmic

This section is devoted to some well-known definitions and statements concerning basic algorithmic that is of use all along this work. A good reference is the book of Gathen-Gerhard [117]. The first subsection "Generalities" defines some notions of elementary algorithmic such that super-additive functions, and the subproduct tree, useful for stating the results in Chapter "On the complexity of D5 principle". The second subsection recalls the complexity of basic operations, multiplication, division, extended GCD, simultaneous remainders, multivariate multiplication.

1.3.1 Generalities

Here no big results appear, just some recalls and properties useful for handling fast algorithms. They are relevant to the Chapter "On the complexity of the D5 principle".

Super-additive functions. We start by introducing a notion of super-additivity for functions of several variables.

Definition 1.9. *Let n be a positive integer. A function $\mathsf{A} : \mathbb{N}^n \to \mathbb{R}$ is* super-additive *if for all $s \geq 1$, for all integer n-uples $(d_1, \ldots, d_n)$ and $(d_{i,1}, \ldots, d_{i,n})$, with $1 \leq i \leq s$, satisfying*

$$\sum_{i \leq s} d_{i,1} \cdots d_{i,n} = d_1 \cdots d_n \quad \text{and} \quad \text{for all } j, \quad d_{i,j} \leq d_j,$$

the inequality

$$\sum_{i \leq s} \mathsf{A}(d_{i,1}, \ldots, d_{i,n}) \leq \mathsf{A}(d_1, \ldots, d_n)$$

holds.

The following lemma helps to prove that a function is super-additive.

Lemma 1.4. *Suppose that for all $(d_1, \ldots, d_n)$ and $(d'_1, \ldots, d'_n)$ in $\mathbb{N}^n$, with $d_i \leq d'_i$ for all i, the inequality*

$$\frac{\mathsf{A}(d_1, \ldots, d_n)}{d_1 \cdots d_n} \leq \frac{\mathsf{A}(d'_1, \ldots, d'_n)}{d'_1 \cdots d'_n}$$

holds. Then A is super-additive.

Proof: Let s, $(d_1, \ldots, d_n)$ and $(d_{i,1}, \ldots, d_{i,n})$ be as in Definition 1.9. For any $i \leq s$, our assumption yield the inequalities

$$\frac{\mathsf{A}(d_{i,1}, \ldots, d_{i,n})}{d_{i,1} \cdots d_{i,n}} \leq \frac{\mathsf{A}(d_1, \ldots, d_n)}{d_1 \cdots d_n},$$

whence

$$d_1 \cdots d_n \ \mathsf{A}(d_{i,1}, \ldots, d_{i,n}) \leq d_{i,1} \cdots d_{i,n} \ \mathsf{A}(d_1, \ldots, d_n).$$

Summing over all i leads to

$$\begin{aligned} d_1 \cdots d_n \ \sum_{i \leq s} \mathsf{A}(d_{i,1}, \ldots, d_{i,n}) &\leq \left(\sum_{i \leq s} d_{i,1} \cdots d_{i,n} \right) \mathsf{A}(d_1, \ldots, d_n) \\ &\leq d_1 \cdots d_n \ \mathsf{A}(d_1, \ldots, d_n). \end{aligned}$$

Canceling $d_1 \cdots d_n$ gives the result. □

Corollary 1.4. *Suppose that $\mathsf{U} : \mathbb{N} \to \mathbb{N}$ satisfies $\frac{\mathsf{U}(d)}{d} \leq \frac{\mathsf{U}(d')}{d'}$ for all $d \leq d'$. Then the function*

$$(d_1, \ldots, d_n) \mapsto \prod_{i \leq n} \mathsf{U}(d_i)$$

is super-additive.

Logarithmic functions. For our complexity estimates, we need to state inequalities involving logarithmic functions. In order to obtain explicit results that hold for all values of the arguments, we are led to the following definition.

Definition 1.10. *The function* logp *is defined by* $\mathrm{logp}(x) = 2\log_2(\max\{2, x\})$ *for any positive integer* x*.*

This definition is motivated by the following lemma.

Lemma 1.5. *For all* n *and all positive integers* $d_1, \ldots, d_n$*, we have the inequalities*

$$2 \leq \mathrm{logp}(d_1 \cdots d_n) \leq \mathrm{logp}(d_1) \cdots \mathrm{logp}(d_n).$$

Proof: Let $d_1, \ldots, d_n$ be positive integers. The inequality $2 \leq \mathrm{logp}(d_1 \cdots d_n)$ is obvious. To prove the right-hand inequality, we can freely suppose that the d_i are not all equal to 1, this last case being trivial. Suppose further that $d_1, \ldots, d_k$ are all at least 2, whereas $d_{k+1}, \ldots, d_n$ are all 1. Then $d_1 \cdots d_n = d_1 \cdots d_k$, so we get

$$\mathrm{logp}(d_1 \cdots d_n) = \mathrm{logp}(d_1 \cdots d_k).$$

We then have the equalities

$$\mathrm{logp}(d_1 \cdots d_k) = 2\log_2(d_1 \cdots d_k) = 2\sum_{1 \leq j \leq k} \log_2(d_j) = 2\sum_{1 \leq j \leq k} \frac{\mathrm{logp}(d_j)}{2}.$$

This estimate admits the upper bounds

$$\sum_{1 \leq j \leq k} \mathrm{logp}(d_j) \leq \prod_{1 \leq j \leq k} \mathrm{logp}(d_j),$$

the last inequality following from the lower bound $\mathrm{logp}(d_j) \geq 2$. □

The subproduct tree. The subproduct tree is a useful construction to devise fast algorithms for univariate polynomials. It is a binary tree, all of whose nodes are labeled by univariate polynomials.

Definition 1.11. *Let* R *be a ring and* $m_1, \ldots, m_r$ *be monic, non-constant, polynomials in* $R[y]$*. The subproduct tree* Sub *associated to* $m_1, \ldots, m_r$ *is defined as follows:*

- *If* $r = 1$*, then* Sub *is a single node, labeled by the polynomial* m_1*.*
- *Else, let* $r' = \lceil r/2 \rceil$*, and let* Sub_1 *and* Sub_2 *be the trees associated to* $m_1, \ldots, m_{r'}$ *and* $m_{r'+1}, \ldots, m_r$ *respectively. Let* p_1 *and* p_2 *be the polynomials at the roots of* Sub_1 *and* Sub_2*. Then* Sub *is the tree whose root is labeled by the product* $p_1 p_2$ *and has children* Sub_1 *and* Sub_2*.*

A row *of the tree consists in all nodes lying at some given distance from the root. The* depth *of the tree is the number of its non-empty rows.*

Lemma 1.6. *Let* $d = \sum_{i=1}^{r} \deg(m_i)$*. Then the following holds:*

1. *The sum of the degrees of the polynomials on any row of* Sub *is at most* d.

2. *The depth of* Sub *is at most* $\mathrm{logp}(d)$.

Proof: Point (1) comes by an immediate structural induction.

We next prove that for all $r \geq 1$, the depth admits the upper bound $1 + \lceil \log_2(r) \rceil$. This is proved by induction: the result clearly holds for $r = 1$, and the induction step follows from the identity $\lceil \log_2(r) \rceil = 1 + \lceil \log_2(\lceil r/2 \rceil) \rceil$, which holds for all $r \geq 2$. Point (2) now comes from the inequality $r \leq d$, which holds since all m_i are non-constant, and from the definition of the function logp. □

1.3.2 Basic operations

We deal here with fast algorithms for multiplication, GCD computation, multivariate multiplication and rational reconstruction. It is strongly inspired by Chapters 10 and 11 of Gathen-Gerhard [117].

Operations on univariate polynomials. We now define multiplication time for univariate polynomials.

Definition 1.12. *A* multiplication time *is a map* $\mathsf{M} : \mathbb{N} \to \mathbb{R}$ *such that:*

- *For any ring* R*, polynomials of degree less than* d *in* $R[X]$ *can be multiplied in at most* $\mathsf{M}(d)$ *operations* $(+, \times)$ *in* R.
- *For any* $d \leq d'$*, the inequality* $\frac{\mathsf{M}(d)}{d} \leq \frac{\mathsf{M}(d')}{d'}$ *holds.*

Note that in particular, the inequality $\mathsf{M}(d) \geq d$ holds for all d. The following result is due to [26], following work of Schönhage and Strassen.

Proposition 1.6. *There exists* $c \in \mathbb{R}$ *such that the function*

$$d \mapsto \mathsf{M}(d) = c\, d\, \mathrm{logp}(d) \mathrm{logplogp}(d)$$

is a multiplication time.

Fast polynomial multiplication is the basis of many other fast algorithms: Euclidean division, computation of the subproduct tree, and multiple remaindering. We give two sorts of statements: one, as is usual, in terms of the M function, involving $O(\ )$ terms, and another with more explicit estimates.

Proposition 1.7. *Let* M *be a multiplication time. There exists a constant* $C_{\mathsf{M}} \geq 1$ *such that the following holds over any ring* R*:*

1. *Dividing in* $R[X]$ *a polynomial of degree less than* $2d$ *by a monic polynomial of degree at most* d *can be done using at most*

$$5\mathsf{M}(d) + O(d) \leq C_{\mathsf{M}}\, \mathsf{M}(d)$$

operations $(+, \times)$ *in* R.

2. *Let F be a monic polynomial of degree d in $R[X]$. Then additions and multiplications in $R[X]/F$ can be done using at most*

$$6\,\mathsf{M}(d) + O(d) \leq C_{\mathsf{M}}\,\mathsf{M}(d)$$

operations $(+, \times)$ in R.

3. *Let $F_1, \ldots, F_s$ be non-constant monic polynomials in $R[X]$, with sum of degrees d. Then one can compute the subproduct tree associated to $F_1, \ldots, F_s$ using at most*

$$\mathsf{M}(d)\mathrm{logp}(d)$$

operations $(+, \times)$ in R.

4. *Let $F_1, \ldots, F_s$ be non-constant monic polynomials in $R[X]$, with sum of degrees d. Then given A in $R[X]$ of degree less than d, one can compute $A \bmod F_1, \ldots, A \bmod F_s$ using at most*

$$11\,\mathsf{M}(d)\mathrm{logp}(d) + O(d\mathrm{logp}(d)) \leq C_{\mathsf{M}}\,\mathsf{M}(d)\mathrm{logp}(d)$$

operations $(+, \times)$ in R.

5. *Assume that R is a field. Then, given two polynomials in $R[X]$ of degree at most d, computing their monic GCD and their Bézout coefficients can be done in no more than*

$$33\,\mathsf{M}(d)\mathrm{logp}(d) + O(d\mathrm{logp}(d)) \leq C_{\mathsf{M}}\,\mathsf{M}(d)\mathrm{logp}(d)$$

multiplications, additions and inversions in R.

Proof: The first point is proved in [117, Theorem 9.6, Ch. 9] and implies the second one [117, Corollary 9.7]. The third and fourth points are proved in Lemma10.4 and Corollary 10.7 of Chapter 10 of the same reference. The fifth point is reported in its Chapter 11 with constant of 24 instead of 33. In Chapter "On the complexity of the D5 principle", using this constant is more convenient and simplifies the calculations. □

Multivariate multiplication. This paragraph deals with multivariate multiplication of polynomials, power series and the related question of multiplication modulo a triangular set. Here are the notations used:

- $\mathsf{M}(m, d)$: multiplication of two multivariate polynomials of degree d with m variables.
- $\mathsf{M}_s(m, d)$: multiplication of two power series at precision d with m variables.
- $\mathsf{M}_{\mathrm{trig}}(d_1, \ldots, d_n)$ or $\mathsf{M}_{\mathrm{trig}}(T)$: multiplication of a polynomial $P \in K[X_1, \ldots, X_n]$ modulo a triangular set T of multi-degree $(d_1, \ldots, d_n)$.

Using a Kronecker substitution [117, § 8.4] to a polynomial $f \in K[X_1, \ldots, X_n]$ of partial degree in X_i at most d_i leads to univariate multiplications of degree $\prod_{i=1}^{n}(2d_i + 1)$. Over a field of characteristic zero or greater than $2d_i + 1$ for each i, Pan [93] gives a complexity in $O\big(\prod_{i=1}^{n}(2d_i + 1)\sum_{i=1}^{n}\frac{\mathsf{M}(d_i)\log(d_i)}{d_i}\big)$. In 2004, van der Hoeven [116] with some improvements of the complexity analysis of the truncated Fourier transform, reachs a

$O\big(\prod_{i=1}^{n}(2d_i+1)\sum_i \log(d_i)\big)$, when there exists enough primitive roots of unity in the base field.

Concerning truncated series, two types of truncation are of interest. First, the *partial* degree truncation, where monomials in the ideal $(p_1^{d_1},\ldots,p_m^{d_m})$ are discarded. The best result in that direction is due to Schost [104, Corollary 2]. The complexity reached is $\mathsf{MS}(m, d_1\cdots d_m) \in O\big((d_1\cdots d_m)^{1+\epsilon}\big)$, for every ϵ. Second, the *total* degree truncation. It is of use notably for the Newton-Hensel algorithm (see next section). Here, they are the monomials in the ideal $(p_1,\ldots,p_m)^d$ which are discarded. The more satisfactory result seems to be the work of Lecerf and Schost [78, Theorem 1]. Van der Hoeven announces a better result in [116], but the proof was not correct. A proof in an *addendum* [115] is in progress of validation.

$$\text{Lecerf-Schost}\quad \mathsf{MS}(m,d) \in O\big(D\log(D)^3\log(\log(d))\big),\quad \text{where}\quad D=\binom{m+d}{m}. \tag{1.15}$$

Modulo triangular sets. The problem is now to evaluate the complexity of multiplication modulo a triangular set. As required for the analysis of a lifting step of the Newton-Hensel lifting in Algorithm 1.1, we state this result here. Let $T = T_1,\ldots,T_n$ be a triangular set over K, of degree $d_1,\ldots,d_n$. Given two polynomials f and g, both reduced modulo T, we want to evaluate the number of operations $\mathsf{M}_{\text{trig}}(T) = \mathsf{M}_{\text{trig}}(\deg_{X_1}(T_1),\ldots,\deg_{X_n}(T_n))$ over K required for computing $fg \bmod (T)$. An easy induction, using point (1) of the Proposition 1.7 gives:

$$\mathsf{M}_{\text{trig}}(T) \le C_{\mathsf{M}}^n \mathsf{M}(\deg_{X_1}(T_1))\cdots\mathsf{M}(\deg_{X_n}(T_n)). \tag{1.16}$$

In Chapter 3, this function is denoted MT. There, since MT take additionally care of modular inversions, appear cubic logarithmic terms.

1.4 Lifting techniques

This section provides all the material required for using the Newton-Hensel operator: construction of the triangular operator, rational reconstruction, and the stop criterion. These three steps are sometimes known as the *specialize and lift paradigm*, and are enclosed in the generic term of *lifting techniques*.

The Newton operator is used in this thesis in its *triangular* form: We only give here the main lines of its description, in the next subsection and refer to the thesis of Schost [102, Chapitre 6 and Annexe C]. Subsection 1.4.2 tackles the "Rational reconstruction" problem. More details on the multivariate rational reconstruction are given in Paragraph 4.3 of [103], which strongly inspired these lines. The presentation of Lecerf in his thesis [75, § II.4] is also of interest. Both of these works are in the continuity of the articles of Giusti, Heintz, Pardo *et al.* [49, 50] and also in [56]. Inside their work, appears an use of the Newton operator with complexity considerations. The complexities of the algorithms for changing of order (Chapter 3) and for the equiprojectable decomposition (Chapter 4) both rely on a study in the same vein of the Newton operator.

1.4.1 Triangular Newton-Hensel operator

Originated from numerical analysis, the Newton operator is used for solving polynomial systems symbolically since the 80's. The presentation and the spirit of the complexity results given here are mostly inspired by the work of the TERA group of Giusti, Heintz, Pardo *et al.* [49, 50, 56]. It is only a sketch of presentation, for proofs and further details, the references are [102, Chapitre 6], [105, § 7], [103, § 4 and 5], and [75, § II.4].

Triangular Newton-Hensel operator: *Let A be an unitary commutative ring, $\mathfrak{m}$ an ideal of A, $\mathbf{f} = (f_1, \ldots, f_n)$ a square polynomial system in $A[X_1, \ldots, X_n]$ and $\mathbf{t}$ a triangular set in $A[X_1, \ldots, X_n]$ such that:*

- the system $\mathbf{f}$ is reduced to zero modulo $\mathbf{t}$.

- the jacobian matrix $\mathrm{Jac}\,(\mathbf{t})$ is invertible in $(A/\mathfrak{m})[X_1, \ldots, X_n]/(\mathbf{t})$.

The Newton-Hensel operator considered here computes iteratively the sequence $(\mathbf{t}^\kappa \equiv \mathbf{t} \bmod \mathfrak{m}^{2^\kappa})$, from the datum of $\mathbf{t} \bmod \mathfrak{m}$, by a succession of matrix products and one inversion.

In this thesis, the ring A and the ideal $\mathfrak{m}$ considered above will be either $\mathbb{Z}$ and the maximal ideal (p) for a prime p, either $K[p_1, \ldots, p_m]$ and the maximal ideal $(p_1 - a_1, \ldots, p_m - a_m)$ for an m-uple in K^m.

The effective algorithm is given by Schost [103, Proposition 4]; we recall how it is devised in Algorithm 1.1, and we introduce some notations used therein:

- $\mathbf{t}^\kappa$ denotes the system $\mathbf{t} \bmod \mathfrak{m}^{2^\kappa}$
- $\mathbf{T}^\kappa = (T_1^\kappa, \ldots, T_n^\kappa)$ is a triangular set such that

$$T_j^\kappa \equiv t_j^\kappa \bmod \mathfrak{m}^{2^\kappa}(A/\mathfrak{m}^{2^{\kappa+1}})[X_1, \ldots, X_n]\,, \qquad j = 1, \ldots, n.$$

- $\mathbf{f}^\kappa$ is the image of $\mathbf{f}$ through $A[X_1, \ldots, X_n] \to (A/\mathfrak{m}^{2^{\kappa+1}})[X_1, \ldots, X_n]/(\mathbf{T}^\kappa)$.
- $\mathrm{Jac}\,(\mathbf{f}^\kappa)$ denotes the jacobian matrix of $\mathbf{f}^\kappa$.

Let us turn out to the complexity of the lifting step. It is convenient to introduce the *complexity of evaluation* of a polynomial. Numerous references treat of algorithmic topics by evaluation computation [65, 24, 49]. In our context, we follow the ideas present in the works of Giusti *et al.* [49, 50, 51] where it is shown that the use of such data structures permits to obtain a better complexity than the representation of polynomials in the monomial basis.

It is a natural assumption that $\mathbf{f}$ is given by a straight-line program since the algorithm required to evaluate the system and its jacobian modulo a triangular set. More precisely, in the multivariate situation where $\mathbf{f} \subset k[p_1, \ldots, p_m, X_1, \ldots, X_n]$ is given by s straight-line program of size L, then we have [105, Proposition 11]:

Proposition 1.8. *Computing $\mathbf{t}^{\kappa+1}$ from $\mathbf{t}^\kappa$ using Algorithm 1.1 requires a number of operations over A in the order of:*

$$O\big((nL + n^3)\mathsf{M}_{\mathrm{trig}}(\deg_{X_1}(t_1^\kappa), \ldots, \deg_{X_n}(t_n^\kappa))\mathsf{MS}(2^{\kappa+1}, m)\big).$$

As in the introduction, K_0 denotes either $\mathbb{Q}$ or $k(p_1, \ldots, p_m)$. We are interested in only these two situations and no more on finite extensions of those, so that $K = K_0$ in this section. Moreover the ring of integers of K is denoted $\mathcal{O}_K$ (so that $\mathcal{O}_K = \mathbb{Z}$ or $\mathcal{O}_K = k[p_1, \ldots, p_m]$).

`Lift`$(\mathbf{f}\,,\,\mathbf{t}^\kappa\,,\,\mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1}\,)$

#`Inputs:` $\mathbf{f}$ the input polynomial system.
$\mathbf{t}^\kappa$ such that $\mathbf{f} \bmod \mathfrak{m}^{2^\kappa} \subset \mathbf{t}^\kappa$.
$\mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1}$ the inverse of the jacobian matrix of $\mathbf{f}^\kappa$.
#`Outputs:` the triangular set $\mathbf{t}^{\kappa+1} \equiv \mathbf{t}^\kappa \bmod \mathfrak{m}^{2^{\kappa+1}} A[X_1, \ldots, X_n]$.
the inverse matrix $\mathrm{Jac}\,(\mathbf{f}^{\kappa+1})^{-1}$.

The computations in Steps 1., 2. and 3. are done over $A/\mathfrak{m}^{2^{\kappa+1}}[X_1, \ldots, X_n]/(T_1^\kappa, \ldots, T_n^\kappa)$

1. Compute $\mathrm{Jac}\,(\mathbf{T}^\kappa)$.
2. Let $\delta^\kappa = \mathrm{Jac}\,(\mathbf{T}^\kappa)\mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1}\mathbf{f}^\kappa$.
3. Let $\tilde{\delta}_j^\kappa$ be the preimage of δ_j^κ through
$$A/\mathfrak{m}^{2^{\kappa+1}}[X_1, \ldots, X_n] \to A/\mathfrak{m}^{2^{\kappa+1}}[X_1, \ldots, X_n]/(T_1^\kappa, \ldots, T_n^\kappa),$$
expressed in the monomial basis $\{X_1^{a_1} \cdots X_n^{a_n}\,,\ 0 \le a_j < \deg_{X_j}(T_j^\kappa)\}$.
4. Let $\mathbf{t}^{\kappa+1} = (t_1^{\kappa+1}, \ldots, t_n^{\kappa+1}) = (T_1^\kappa + \tilde{\delta}_1^\kappa, \ldots, T_n^\kappa + \tilde{\delta}_n^\kappa)$.
5. $\mathrm{Jac}\,(\mathbf{f}^{\kappa+1})^{-1} = 2\mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1} - \mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1} \cdot \mathrm{Jac}\,(\mathbf{f}) \cdot \mathrm{Jac}\,(\mathbf{f}^\kappa)^{-1}$.
This permits to perform only one inversion, explicitly $\mathrm{Jac}\,(\mathbf{f}^0)^{-1}$*, to get* $\mathrm{Jac}\,(\mathbf{f}^{\kappa+1})^{-1}$ *only with matrix multiplication*
6. `return` $\mathrm{Jac}\,(\mathbf{f}^{\kappa+1})^{-1}$ `and` $\mathbf{t}^{\kappa+1}$.

Algo **1.1:** One iteration of the lifting procedure: from 2^κ to $2^{\kappa+1}$

To control the number of steps of Newton iterations, it is necessary to have at hands some bounds on the height of the element x of K we aim at reconstruct. An easy case in when this element x belongs to the ring of integers $\mathcal{O}_K$, since there is no rational reconstruction necessary. Then the number of steps κ should verify at least

$$\kappa > \begin{cases} \lceil \log_2 \left(\frac{h_p(x)+1}{\log(p)} \right) \rceil & \text{if } x \in \mathbb{Z} \text{ and } \mathfrak{m} = (p), \\ \lceil \log_2(\deg(x)+1) \rceil & \text{if } x \in k[p_1, \ldots, p_m]. \end{cases} \tag{1.17}$$

When $x \in K - \mathcal{O}_K$, then an additional procedure is required, the *rational reconstruction*. It is the object of the next subsection. In this case, κ should verify

$$\kappa > \begin{cases} \lceil \log_2 \left(\frac{2h_p(x)+1}{\log(p)} \right) \rceil & \text{if } x \in \mathbb{Q} \text{ and } \mathfrak{m} = (p), \\ \lceil \log_2(2\deg(x)+1) \rceil & \text{if } x \in k(p_1, \ldots, p_m). \end{cases} \tag{1.18}$$

In our context, we aim at reconstruct coefficients in $\mathbb{Q}$ or $k(p_1, \ldots, p_m)$ of the polynomials of a triangular set. In Chapter 2 general bounds are given for triangular systems, including sharp bounds for triangular sets. To control the number of steps, we make use of those. They are intrinsic, i.e. depends on the degree and the height of the variety described by

the input system. In practice, Bézout theorem permits to obtain bounds readable on the system; but they are in d^n if there are n polynomials of maximal degree d.

To avoid lifing until the Bézout bound d^n, a probabilistic version of the Stop Criterion permits to minimize costly Newton iterations (Subsection 1.4.3). The choice of that criterion is discussed at the end of the next subsection, after the presentation of the rational reconstruction principle. Whatever is decided, probabilistic criterion or not, let us call `StopCriterion`($\mathbf{t}^\kappa$) the procedure taking as input a triangular set in $\mathcal{O}_K/\mathfrak{m}^{2^\kappa}$ and returning a boolean and eventually a triangular set over K, deciding if the lifting process should be stopped or continued. We get Algorithm 1.2.

`LiftingProcess`($\mathbf{f}$, $\mathfrak{m}$, $\mathbf{t}^0$, `StopCriterion` , Bound)

#`Inputs:` $\mathbf{f}$ input polynomial system.
$\mathbf{t}_0$ a triangular set modulo $\mathfrak{m}$ such that $\mathbf{f}$ mod $\mathfrak{m} \subset \mathbf{t}^0$.
Bound is an *a priori* bound on the number of steps.
#`Output:` a triangular set $\mathbf{t} \supset \mathbf{f}$ such that $\mathbf{t}$ mod $\mathfrak{m} \equiv \mathbf{t}^0$.

1. `bool = false ;` $\kappa = 1$;
2. Compute the inverse Jac $(\mathbf{t})^{-1}$ in $(\mathcal{O}_K/\mathfrak{m})[X_1, \ldots, X_n]/(\mathbf{t}_0)$.
3. `while` ($\kappa \leq$ Bound) `do`
   (a) ($\mathbf{t}^\kappa$, Jac $(\mathbf{f}^\kappa)^{-1}$) =`Lift`$(\mathbf{f}, \mathbf{t}^{\kappa-1}$, Jac $(\mathbf{f}^{\kappa-1})^{-1}$)
   (b) (`bool` , $\mathbf{t}'$) =`StopCriterion`$(\mathbf{t}^\kappa)$
   (c) `if bool then return t' ; end if`
   (d) $\kappa = \kappa + 1$
4. `end while`
5. `return` fail

Algo **1.2:** The Newton-Hensel lifting process

1.4.2 Rational reconstruction

This constitutes the last step of the lifting procedure. A good reference is Paragraph 4.3 of [103]. Let us start with the univariate situation. The problem is the following:

- Let n be an integer and $f = f_0 + f_1 X + \cdots \in k[[X]]$ a univariate power series known at precision n. Given an integer $m \leq n$, is there exist (and how to compute) polynomials U and V with $\deg(U) \leq m-1$ and $\deg(V) \leq n-m$ such that:

$$V(0) \neq 0 \quad \text{and} \quad f = \frac{U}{V} \bmod X^n. \tag{1.19}$$

  It is the problem of *Padé* approximation; U and V are approximates of order $(m, n-m)$ of f.

- Let n be a positive integer, p a prime number, and f an *integer* ranged between 0 and $p^n - 1$. Given $m \leq n$, is there exist (and how to compute) integers U and V, with $|U| < p^m$ and $0 \leq V \leq p^{n-m}$, such that:

$$p \nmid V \quad \text{and} \quad f = \frac{U}{V} \bmod p^n \tag{1.20}$$

Both problems can be solved efficiently by the extended Euclidean algorithm, leading to a satisfactory complexity: with the notations above, deciding if (1.19) has a solution, and in the affirmative, computes such a solution, can be done in

$$O\big(\mathsf{M}(n)\log(n)\big) \tag{1.21}$$

operations over the base field k. The same properties are valid for the problem (1.20) with the complexity:

$$O\big(\mathsf{M}_{\mathbb{Z}}(n\log(p))\log(n\log(p))\big) \tag{1.22}$$

bit operations, where $\mathsf{M}_{\mathbb{Z}}(d)$ is an upper bound on the number of bit operations required to perform the multiplication of two integers with d digits at most. These two problems are of the same nature under the point of view of power series: Problem (1.20) is also a rational reconstruction from a power series, according to the embedding $\mathbb{Z} \hookrightarrow \mathbb{Z}_p$ and the bijection $\mathbb{Z}_p \simeq \mathbb{F}_p[[X]]$, $\sum a_i p^i \mapsto \sum a_i X^i$ (this is the *Hensel representation* of p-adic integers; there are others).

In the multivariate case the probabilistic algorithm of Schost [103, page 27] reduces to the univariate situation by a generic linear change of variable and by putting the variables in the coefficients ring. The complexity then makes appear the cost of the multiplication of multivariate power series. Moreover some choices are made for the linear change of variable, making the algorithm *probabilistic*. The way how it is designed is outlined in Algorithm 1.3.

The complexity is no more polynomial in d, the degree precision reached, because of the cost of the multivariate power series multiplication. However the results are similar to the univariate case. Proposition 83 of [102] proves that the algorithm above requires:

$$O\big(m^2\,\mathsf{M}(d)\,\mathsf{MS}(2d,m)\big), \tag{1.23}$$

operations over k. The probabilistic aspect can be quantified thanks to Proposition 81 of [102], that we recopy *in extenso* here:

Proposition 81 *Let p and q be two polynomials in $k[p_1,\ldots,p_m]$ of degree at most d, with $q(0) \neq 0$, and r the Taylor expansion of p/q at the origin at the total degree precision $d' \geq 2d$. There exists a polynomial $P \in k[\gamma_2,\ldots,\gamma_m]$ of degree at most*

$$d'^2(d'+1),$$

such that for all γ not canceling P, Algorithm 1.3 applied to (r,γ) computes p/q within the complexity (1.23).

Let us go back to the probabilistic `StopCriterion` discussed above. At each iteration of the Newton process, this criterion is performing:

- a rational reconstruction of the current Taylor approximation.

`MutlivariateRationalReconstruction`$(s\,,\,\gamma)$

#`Inputs`: $s \in k[[p_1, \ldots, p_m]]$, known at precision d.
γ is an element of k^{m-1}.
#`Output`: `true` and (p, q) if there exists $p, q \in k[X_1, \ldots, X_n]$ of degrees lower than $d/2$,
and such that $p = qs$ at precision d holds, and with $q(0) \neq 0$.
`false` otherwise.

1. $(p_1, \ldots, p_m) = (p_1, p_2 + \gamma_2 p_1, \ldots, p_m + \gamma p_1)$.
2. $s = s(t, tp_2, \ldots, tp_m)$.
   It is not necessary to keep the variable p_1 since the monomials coefficients in $s(tp_1, \ldots, tp_m)$ are all homogeneous; hence we do $p_1 = 1$.
3. $(\texttt{bool}\,,\,(P, Q)) =$ `UnivariateRationalReconstruction`(P, Q).
   It is performed in $k[[p_1, \ldots, p_m]][t]$.
4. $(p_1, \ldots, p_m) = (p_1, p_2 - \gamma_2 p_1, \ldots, p_m - \gamma_m p_1)$.
5. `if (not bool) return false ; end if`
6. Homogenize the monomials by reintroducing the variable p_1 in P and Q.
7. $p = P|_{t=1}$ and $q = Q|_{t=1}$.
8. `return` p/q.

Algo **1.3:** The multivariate rational reconstruction

- if this succeeds, a probabilistic test that the output is the good one (Figure 1.3).

This permits to reduce the number of steps of the lifting process. The probabilistic test we use in this thesis is to reduce the input system $\mathbf{f}$ on the ideal generated by the output triangular set(s). For the change of order, in Chapter 3, there is only one output triangular set, and for the modular equiprojectable decomposition described in Chapter 4, there are several. We describe in Algorithm 1.4 the test for one triangular set, and refer to the adequate chapter for more details.

Complexity analysis. Here we sketch a complexity analysis of the lifitng step of the algorithm. Let $K = \mathbb{Q}$ or $K = k(p_1, \ldots, p_m)$, $\mathcal{O}_K$ its ring of integers and $\mathbf{t}^\kappa$ a triangular set in $(\mathcal{O}_K/\mathfrak{m}^{2^\kappa})[X_1, \ldots, X_n]$ as defined in the `Input` of the algorithm `StopCriterion`. Let C_{M} be the universal constant of Proposition 1.7.

- $\mathbf{f}$ is given by a straight-line program of size L.
- $\delta_1, \ldots, \delta_n$ is the multi-degree of the triangular set $\mathbf{t}^\kappa$.
- $\mathfrak{m}'$ is a maximal ideal in $\mathcal{O}_K$, equal to $(p_1 - y'_1, \ldots, p_m - y'_m)$ for a point $y' = (y'_1, \ldots, y'_m) \in K^m$ when $\mathcal{O}_K = k[p_1, \ldots, p_m]$, and equal to (p') for a prime $p' \in \mathbb{Z}$ when $K = \mathbb{Q}$.
- For any system $\mathbf{F}$ in $\mathcal{O}_K[X_1, \ldots, X_n]$ and any maximal ideal $\mathfrak{m}'$, the notation $\mathbf{F}_{\mathfrak{m}'}$ refers to the system $\mathbf{F}$ mod $\mathfrak{m}' \in (\mathcal{O}_K/\mathfrak{m}')[X_1, \ldots X_n]$.

In Steps 3 and 4.(a), the computation of $\mathbf{f}_{\mathfrak{m}'}$ and of its normal form with respect to the Gröbner basis $\mathbf{f}_{\mathfrak{m}'}$ can be executed within

$$\begin{cases} L \cdot \mathsf{M}_{\text{trig}}(\delta_1, \ldots, \delta_n) & \text{if } K = k(p_1, \ldots, p_m) \\ L \cdot O\big(\mathsf{M}_{\mathbb{Z}}(p')\big) & \text{if } K = \mathbb{Q} \end{cases} \tag{1.24}$$

We follow each step of the straight-line program, and do the following:

- expand systematically the product or the sum corresponding to the step.
- reduce it modulo $\mathbf{t}_{\mathfrak{m}'}$.

Hence each step requires $\mathsf{M}_{\text{trig}}(\delta_1, \ldots, \delta_n)$ or $O\big(\mathsf{M}_{\mathbb{Z}}(p')\big)$ operations over k or bit operations over $\mathbb{Z}$, leading to $L \cdot \mathsf{M}_{\text{trig}}(\delta_1, \ldots, \delta_n)$ or to $L \cdot O\big(\mathsf{M}_{\mathbb{Z}}(p')\big)$.
As for Step 1.(a), the cost of all the rational recontructions is less than

$$\begin{cases} O\big(\kappa \mathsf{M}\big(2^\kappa\big)\big) & \text{if } K = k(p_1) \\ O\big(\mathsf{M}_{\mathbb{Z}}(2^\kappa \log(p')) \log(2^\kappa \log(p'))\big) & \text{if } K = \mathbb{Q} \\ O\big(m^2 \mathsf{M}(2^\kappa) \mathsf{MS}(2^{\kappa+1}, m)\big) & \text{if } K = k(p_1, \ldots, p_m) \text{ with } m \geq 2, \end{cases} \tag{1.25}$$

We have general estimates of the `StopCriterion` algorithm that we will precise depending on the situation. In Chapter "Equiprojectable decomposition", it will be used with $K = \mathbb{Q}$, for the Chapter "Changing of Order", with $K = k(p_1, \ldots, p_m)$. Now we turn on probabilistic considerations.

1.4.3 Probabilistic aspects

During the whole process in Algorithm 1.2, the random choices made come from:

- the multivariate rational reconstruction (Algorithm 1.3), only in the case where $\mathcal{O}_K = k[p_1, \ldots, p_m]$ with $m \geq 2$.
- the choices of the ideal $\mathfrak{m}'$. It comes from the `StopCriterion`, presented in Algorithm 1.4 in order to limit the number of iteration of the Newton operator, which is getting more and more costly.

It is important to remark that while the execution of a modular algorithm, there are other random choices, which are not included in the lifing process. Therefore, we suppose in this subsection that all those others random choices are lucky, i.e. answer the correct ouptut. We will quantify them in the concerned chapters.

The probabilistic quantification of the multivariate rational reconstruction has already been studied by Schost [103, 105, 102]. Let us evaluate the probability of success of the `StopCriterion` described above. When $K = \mathbb{Q}$, we want to specialize modulo a prime number p', and when $K = k(p_1, \ldots, p_m)$ specialize at a relevant point $y \in K^m$. We give here the general scheme for the probabilistic stop criterion:

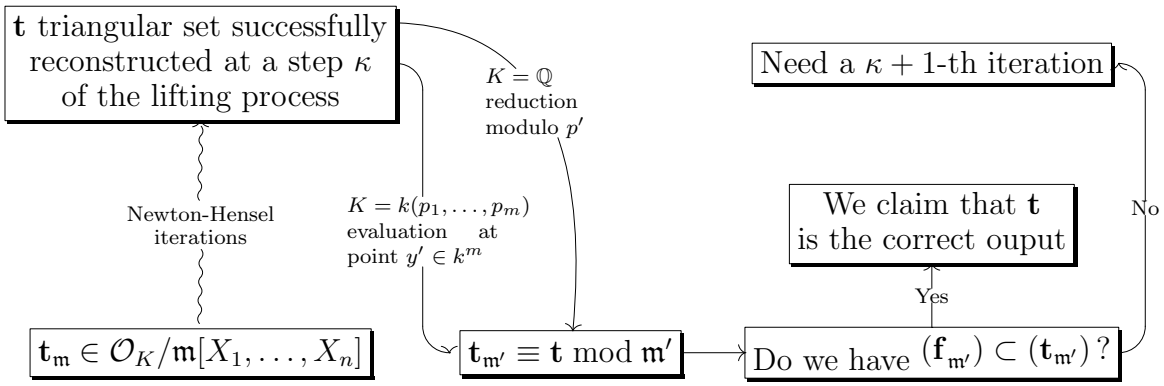

Figure 1.3: The probabilistic test in Step 4 of Algorithm 1.4 `StopCriterion`

In Chapter 3, the input system $\mathbf{f}$ is completely generated by *one* triangular set. So the test $(\mathbf{f}_{\mathfrak{m}'}) \subset (\mathbf{t}_{\mathfrak{m}'})$ is equivalent to the more classical $(\mathbf{f}_{\mathfrak{m}'}) = (\mathbf{t}_{\mathfrak{m}'})$. But in the chapter "Equiprojectable Decomposition", several triangular sets are required to generate $(\mathbf{f})$, hence the test is relevant: stating it in this way permits to treat both cases.

The ideal $\mathfrak{m}'$ is randomely chosen. A bad choice succeeds to the test but gives a wrong answer:

$$\text{bad choice of ideal } \mathfrak{m}': \qquad (\mathbf{f}_{\mathfrak{m}'}) \subset (\mathbf{t}_{\mathfrak{m}'}) \quad \text{but} \quad (\mathbf{f}) \not\subset (\mathbf{t}).$$

In order to give a probability of success of our lifting process, we need to quantify the locus of bad choices. The following proposition gives a result in that direction in the case of a function in m variables.

Proposition 1.9. *Let us consider the notations $\mathbf{t}^\kappa$, $\mathbf{f}$ like in the inputs of Algorithm 1.4:*

- $\mathbf{t}^\kappa \subset \left(k[p_1, \ldots, p_m]/\mathfrak{m}'^{2^\kappa}\right)[X_1, \ldots, X_m]$, is successfully reconstructed into a triangular set $\mathbf{t} \in k[p_1, \ldots, p_m, X_1, \ldots, X_n]$.

- the multi-degree of the triangular set $\mathbf{t}$ is denoted $d_1, \ldots, d_n$.

- d is the maximal total degree of the polynomials in the system $\mathbf{f}$

There exists a polynomial $\Theta \in k[p_1, \ldots, p_m]$ of degree $n\,d\,d_1 d_2 \cdots d_n$ such that if $y = (y_1, \ldots, y_m)$ does not vanish Θ, and if $\mathfrak{m}' = (p_1 - y_1, \ldots, p_m - y_m)$ then,

$$(\mathbf{f}_{\mathfrak{m}'}) \subset (\mathbf{t}_{\mathfrak{m}'}) \quad \Rightarrow \quad (\mathbf{f}) \subset (\mathbf{t}),$$

with the notations of Step 3 of Algorithm 1.4, $\mathbf{f}_{\mathfrak{m}'} = \mathbf{f} \bmod \mathfrak{m}'$, and $\mathbf{t}_{\mathfrak{m}'} = \mathbf{t} \bmod \mathfrak{m}'$.

Proof: Let us suppose that the test at Step 4 returns `true`, but that the output is not correct: $(\mathbf{f}) \not\subset (\mathbf{t})$. This means that there exists a polynomial $f_i \in \mathbf{f}$ such that $V(\mathbf{t}) \not\subset V(f_i)$, but $V(\mathbf{t}) \cap \{\mathbf{p} = y\} \subset V(\mathbf{f})$, where $\mathbf{p}$ denotes the set of variables $p_1, \ldots, p_m$. We want to enclose such points y in an hypersurface of the parameters space $\mathbb{A}^m_K$ of degree $dd_1 \cdots d_n$.

Let W be an irreducible component of $V(\mathbf{t})$ not completely contained in $V(f_i)$. Then $\dim(W \cap V(f_i)) = \dim(W) - 1$ and by Bézout theorem 1.5, $\deg(W \cap V(f_i)) \leq \deg(f_i)\deg(W)$. Let $\pi_{\mathbf{p}}$ be the projection on the variables $p_1, \ldots, p_m$. The Zariski closure $\overline{\pi_{\mathbf{p}}(W \cap V(f_i))}$ of the projection of $W \cap V(f_i)$ on the parameters space is of dimension $m-1$ and has degree at most $\deg(f_i)\deg(W)$. We note that $\pi_{\mathbf{p}}^{-1}(\{y\}) \cap W = W \cap \{\mathbf{p} = y\}$, implying:

$$W \cap \{\mathbf{p} = y\} \subset V(f_i) \;\Leftrightarrow \forall\, y \in \pi_{\mathbf{p}}(W \cap V(f_i)), \quad \pi_{\mathbf{p}}(\{y\}) \cap W \subset V(f_i).$$

It suffices to discard all these points y for each irreducible components of $V(\mathbf{t})$ not contained in $V(f_i)$, for an $f_i \in \mathbf{f}$. Hence, y should be outside the hypersurface defined by:

$$\bigcup_{f_i \in \mathbf{f}} \bigcup_{\substack{W \text{ irred. comp.} \\ \text{of } V(\mathbf{t}),\ W \not\subset V(f_i)}} \overline{\pi_{\mathbf{p}}\big(V(f_i) \cap W\big)}.$$

The degree of that hypersurface is bounded by:

$$n \left(\sum_{\substack{W \text{ irred. comp.} \\ \text{of } V(\mathbf{t}),\ W \not\subset V(f_i)}} \deg(W) \max_{1 \leq i \leq n} \{\deg(f_i)\} \right) \leq n\, d\, d_1 \cdots d_n,$$

since $\deg(f_i) \leq d$ and $\sum_{\substack{W \text{ irred. comp.} \\ \text{of } V(\mathbf{t})}} \deg(W) = \deg(V(\mathbf{t})) = d_1 \cdots d_n$. □

From Zippel-Schwartz Lemma [123, 106], if the choice of $\mathfrak{m}' = (p_1 - y'_1, \ldots, p_m - y'_m)$ is made inside a finite set $\Gamma^m \subset k^m$, then the bound above discriminates at most $D d_1 \cdots d_n |\Gamma|^{m-1}$ values among Γ^{m-1}.

Multivariate rational reconstruction. As stated in Subsection 1.4.2, this algorithm makes $m-1$ random choices (Cf. Algorithm 1.3) to reconstruct a rational function in $k(p_1, \ldots, p_m)$ from a power series in $k[[p_1, \ldots, p_m]]$, for $m \geq 2$. The $m-1$ choices constitute a point $\gamma \in k^{n-1}$. We want to quantify here the points γ leading to a failure. Let us assume that:

- there are N power series to reconstruct.
- the numerator and denominator are bounded in degree by D.

Proposition 81 page 38 proves that each random choice should be outside an hypersurface of $\mathbb{A}^{m-1}_{\bar{k}}$ of degree $4D(2D+1)^2$. Let Γ be a finite subset of k. If the $m-1$ choices are made inside Γ^{m-1}, then by Zippel-Schwartz Lemma, this discriminates at most:

$$4\, N\, D\, (2D+1)^2 |\Gamma|^{m-2}$$

values of γ among Γ^{m-1}.

`StopCriterion`($\mathbf{t}^\kappa$)

#`Inputs:` $\mathbf{t}^\kappa$ a triangular set in $\mathcal{O}_K/\mathfrak{m}^{2^\kappa}[X_1,\ldots,X_n]$, as given after a Newton iteration.
Remark: the input system $\mathbf{f}$ *is implicitly known also.*
#`Output:` $(\texttt{true},\mathbf{t})$ or `false`,
where $\mathbf{t}$ is a triangular set in $K[X_1,\ldots,X_n]$.

1. `For` all the coefficients $r \in \mathcal{O}_K/\mathfrak{m}^{2^\kappa}$ of all the polynomials t_i^κ of $\mathbf{t}^\kappa$ `do`:

   (a) $(\texttt{bool},\tilde{r}) = \texttt{RationalReconstruction}(r)$
   When $\mathcal{O}_K = k[p_1]$*, i.e.* $m = 1$*, or when* $K = \mathbb{Q}$ *(so* $\mathfrak{m} = (p)$*, for a prime* p*), then it is a* `UnivariateRationalReconstruction`(r)*. Else, when* $\mathcal{O}_K = k[p_1,\ldots,p_m]$ *for* $m \geq 2$*, it is the probabilistic* `MultivariateRationalReconstruction`(r,γ) *of Algorithm 1.3.*

   (b) `If (non bool) then return false; end if`

   `end for`
   Here, the rational reconstruction of all the coefficients of the system $\mathbf{t}^\kappa$ *has succeeded.*

2. reconstruct the system $\mathbf{t} \in K[X_1,\ldots,X_n]$ with all the coefficients reconstructed in the `For` loop of Step 1.

3. Choose a maximal ideal $\mathfrak{m}'$ (i.e. a prime p or a point $y_1,\ldots,y_m$) satisfying the same conditions as $\mathfrak{m}$. Compute $\mathbf{f}_{\mathfrak{m}'} \equiv \mathbf{f} \bmod \mathfrak{m}'$ and $\mathbf{t}_{\mathfrak{m}'} \equiv \mathbf{t} \bmod \mathfrak{m}'$.

4. `For` all polynomials $f_i \in \mathbf{f}_{\mathfrak{m}'}$ `do`

   (a) Compute the normal form $f_i \bmod (\mathbf{t}_{\mathfrak{m}'})$ of f_i modulo the Gröbner basis $\mathbf{t}_{\mathfrak{m}'}$.

   (b) `if` $(f_i \bmod (\mathbf{t}) \not\equiv 0)$ `return false; end if`
   If $(\mathbf{f}) \not\subset (\mathbf{t})$ *then more precisions may be required: another iteration is performed.*

   `end for`

5. `return (true,t)`

Algo **1.4:** The probabilistic stop criterion in Step 2.(b) of Algorithm 1.2

Chapter 2

Height bounds for polynomial representations

Introduction. In this chapter are given space complexity results concerning the Kronecker representation and triangular representations. For the first one, this kind of result is not really new, but nowhere clearly stated in such a generality in my knowledge, and for the second one the results are new. Usually this kind of results is stated in term of *bit size*, useful for quantifying the bit complexity of some algorithms, but the notion of height introduced in § 1.2.2 gives similar results in term of degrees over functional fields, unifying the cases of numbers and functions, and involving *intrinsic* quantities. The height of a variety has in fact been introduced in this aim, as a universal yardstick among all possible algebraic systems describing this variety.

This space complexity is of importance for a polynomial system candidate pretending to represent general algebraic varieties: The more it is compact, the best it is (even if, of course, the bit size is a feature among others to take into account). Hence such candidates are expected to have coefficients growing at most *polynomially* with natural quantities attached to the input system. Experiments and previous results show that these data are dominated by the *Bézout number*, i.e. d^n for a polynomial system of maximal degree d with n variables. Thus any complexity bounds should take care of that quantity, and hopefully be polynomial with respect to it. A look at the Bézout Theorem 1.5 shows that the degree and the height of a variety are both bounded essentially by d^n, hence intrinsic bounds should be polynomial in these two quantities. This is the case for the bounds given in this chapter. As said before, for the Kronecker representation, such bounds are not new. For triangular representations, and more precisely for triangular sets, previous bounds were *exponential* in the Bézout number, and we provide here a *quadratic* bound. It dramatically improves the previous upper bounds given in [45, 112, 105] in the function field case. Those in [105] are intrinsic; they show a bound for a polynomial of triangular set in $n^{O(n)} \deg(\mathfrak{V})^{O(n)}$, which is exponential in n (here $\mathfrak{V}$ denotes the variety described by the polynomials) The bounds of Gallo-Mishra in [45, 112] are not intrinsic, and lie in $n^{O(n)} d^{O(n^2)}$, which is exponential in the Bézout number d^n. Applying Bézout theorems to intrinsic bounds of Schost [105] gives slightly better result still exponential. These results are available in the function field case, and I am not aware of similar bound in the number field case.

The importance of having sharp bounds also holds for algorithms involving *modular methods*. To avoid expression swell during an algorithm dealing with rationals number for

example, it is performed modulo a prime number, and then the output is *lifted* to the required output. Of course, random choices are made, and controlling them permit to evaluate a probability of success. Bounds like the ones presented in this chapter help to sharpen this probability.

In this context, getting intrinsic bounds requires to link the Chow form to the polynomial system representation considered, because of the definition of the height of a variety relying on Chow forms. For the Kronecker representation, this link appears in Macaulay [82] among others, and exists since probably before. We recall this technique of differentiating and specializing the Chow form to get the parameterizing polynomials of this representation in § 2.1.2. This is a warming-up to the generalization of this technique to triangular representations. The Chow form in the triangular context needs to be differentiated carefully. Entire Paragraph 2.1.1 is devoted to technical derivation formulas. The link with Chow forms required is proved in Paragraph 2.1.3. The polynomials obtained (denoted M_i hereunder) are triangular but are not triangular sets. It is however possible to derive formulas for others interesting triangular representations. These representations (denoted NF_i and N_i in the sequel) appears in the PhD thesis of Schost [102] (in two variables, and in an experimental form, in table p. 165, Ch. 18), where he noticed that introducing initials to the polynomials of a triangular set permits to reduce the coefficients.

Before defining these polynomials, let us mention the second technique presented in Section 2.2 to make a link with the Chow form. The bounds obtained therein are the best and use interpolation formulas (Corollary 2.3 and Equation (2.26)). In this form, they seem to be new, even if the Lagrange interpolation formula (Gröbner bases version) as stated in [84, Lemma 1.5] should produce such formulas. The specificities of our work is indeed a *simplicity* and the key partition of the variety into the V_α^i (see Figure 2.3). These do not seem to appear in previous works.

Main results. Let us state now the main results of this chapter. Assume that K_0 is one of the field $\mathbb{Q}$ or $k(p_1, \ldots, p_m)$, for a field k, and that K is a finite extension of K_0. Let V be an algebraic zero-dimensional variety defined over K, of degree D whose vanishing ideal verifies the Separability Assumption. We denote by $\chi_u,\ w_1, \ldots, w_n$ the polynomials of the Kronecker representation (Cf. Definition 1.2) associated with a separating linear form U. Is proved in Theorem 2.2:

Theorem. *The height of the coefficients of $\chi_u'(T)$ and $w_i(T)$ of the primitive element representation of V is bounded by:*

$$\begin{array}{c} h(V) + Dh(U) + D\log(n+2) + (n+1)\log D \quad \textit{(number field case)} \\ h(V) + Dh(U) \quad \textit{(function field case).} \end{array}$$

This result is not a brand-new one, but it has the advantage to be written in both cases of the number and function fields. In [102, Théorèmes 8 , 15], similar bounds are given.

For triangular representations, the generalization of the differentiation of the Chow form made for the primitive element in Theorem 2.1 makes appear new polynomials. Suppose now that we are given an equiprojectable variety V (Cf. Definition 1.6) defined by a family of triangular set $\mathbf{T} = (T_1, \ldots, T_n)$ over K, of degree $(d_1, \ldots d_n)$, hence, whose generating radical ideal verifies the Separability Assumption (conditions contained in our definition

of triangular set 1.4). The use of the projectors π_i^n of Definition 1.5 and of the following convenient notation is made:

$$\begin{aligned} \mathbf{G}_\ell &= 1 + 2\sum_{i\le \ell-1}(d_i - 1) \\ \mathsf{H}_\ell &= 5\log(\ell+3)\sum_{i\le\ell} d_i \\ \mathsf{I}_\ell &= \mathsf{H}_\ell + 3\log(2)\sum_{i\le\ell-1} d_i(d_i-1). \end{aligned} \tag{2.1}$$

Since $\prod_i d_i > \sum_i (d_i - 1)$, $\mathbf{G}_\ell$ and H_ℓ are in $O(\log(\ell)(\ell + \deg\big(\pi_\ell^n(V)\big)$: we think of them as linear in $\deg(\pi_\ell^n(V))$, overlooking the dependence in ℓ. Since $d_i^2 \ge d_i(d_i-1)+1$ we get $\prod_i d_i^2 > \sum_i d_i(d_i-1)$, so:

$$\mathsf{I}_\ell \in O\Big(\big(\log(\ell)(\deg(\pi_\ell^n(V)) + \ell\big) + \big(\deg(\pi_\ell^n(V)) + \ell\big)^2\Big) = O\big((\deg(\pi_\ell^n(V)) + \ell)^2\big)$$

we see it as a quadratic quantity. We prove in Theorem 2.7:

Theorem. *For $0 \le \ell \le n-1$ the height of the polynomial $T_{\ell+1}$ is bounded by:*

$$h(T_{\ell+1}) \le \begin{cases} \mathbf{G}_{\ell+1} h(\pi_{\ell+1}^n(V)) + \mathsf{I}_{\ell+1}, & \textit{(number field case)} \\ \mathsf{G}_{\ell+1} h\big(\pi_{\ell+1}^n(V)\big) \le 2\deg(\mathfrak{V})^2, & \textit{(function field case)} \end{cases}$$

The last equality comes from Equation (1.10). The comments made about the bounds $\mathbf{G}_\ell$, H_ℓ and I_ℓ show that in the number field case the height is essentially bounded by

$$O\Big(\deg\big(\pi_{\ell+1}^n(V)\big)\cdot h\big(\pi_{\ell+1}^n(V)\big) + \deg\big(\pi_{\ell+1}^n(V)\big)^2\Big),$$

where the "big O" hides logarithmic terms in the height and in the degree. We can say that this bound, as well as the one for the function field case, is *quadratic* in the data of the problem, since only the product of the height by the degree and the square of the degree appears.

As for the Shape Lemma representation (Cf. Equations (1.3)), where introducing derivatives leads to the Kronecker representation and its diminution of the size of the coefficients, the same can be expected for triangular sets: introducing suitable initials can reduce the size of the coefficients. Schost in [102, p. 165, Ch. 18], made some experiments in two variables. Here we define these initials and give bounds to the two resulting families of triangular polynomials:

$$\begin{aligned} N_{\ell+1} &\equiv \Big(\prod_{i=1}^{\ell} \partial_{X_i}(T_i)\Big)\cdot T_{\ell+1} \bmod (T_1,\dots,T_\ell) \\ NF_{\ell+1} &\equiv d_{\ell+1}!\prod_{i=1}^{\ell}\Big((d_i-1)d_{i+1}\cdots d_{\ell+1}\Big)!\Big(\prod_{i=1}^{\ell}\partial_{X_i}(T_i)^{d_{i+1}\cdots d_{\ell+1}}\Big)\cdot T_{\ell+1} \bmod (T_1,\dots,T_\ell). \end{aligned}$$

We prove the following bounds in Section 2.2, using the interpolation formula of Corollary 2.3.

Theorem. *If $N_1 = T_1$ and $N_{\ell+1}$ defined above for $0 \leq \ell \leq n-1$ then we have*

$$h(N_{\ell+1}) \leq \begin{cases} h(\pi^n_{\ell+1}(V)) + \mathsf{H}_{\ell+1}, & \textit{(number field case)} \\ h(\pi^n_{\ell+1}(V)) \leq \deg(\mathfrak{V}). & \textit{(functional case)} \end{cases}$$

Using the comments after Definition of H_ℓ, in the number field case, the height of $h(N_{\ell+1})$ is bounded by:

$$O\big(h(\pi^n_{\ell+1}(V)) + \deg(\pi^n_{\ell+1}(V))\big),$$

where again the "big O" hides logarithmic terms in $\deg(\pi^n_{\ell+1}(V))$ or $h(\pi^n_{\ell+1}(V))$. Here the bound is *linear* in the degree and the height. It is better than the bound obtained for the coefficients of the polynomials $T_1, \ldots, T_n$. Experiments reported in Table 2.2 confirm this remark. Among all the triangular systems proposed in this Chapter, the family $N_!, \ldots, N_n$ present the best result (Cf. Table 2.1. The last inequality involving $\mathfrak{V}$ comes from Equation (1.10).

Using the formula of derivations of Section 2.1 we prove in Theorem 2.6 the following bound for the polynomials NF_i:

Theorem. *Let $NF_1 = T_1$ and $NF_{\ell+1}$ defined as above for $1 \leq \ell \leq n-1$, then the following bounds hold:*

$$h(NF_{\ell+1}) \leq \begin{cases} h\big(\pi^n_{\ell+1}(V)\big) + \deg(\pi^n_{\ell+1}(V))\Big(\log\big((n+2)\deg(\pi^n_{\ell+1}(V))\big) \\ +(n-1)h(\pi^n_{\ell+1}(V))\deg(\pi^n_{\ell+1}(V)) + 5n^2 + 4n^3\deg(\pi^n_{\ell+1}(V))\Big) & \textit{(number field)} \\ h(NF_{\ell+1}) \leq h(\pi^n_{\ell+1}(V))\left(1 + (n-1)\deg(\pi^n_{\ell+1}(V))^2\right) & \textit{(function field)} \end{cases}$$

This bound is cubic: In fact it is bounded by:

$$O\big(\deg(\pi^n_{\ell+1}(V))^2 h(\pi^n_{\ell+1}(V)\big),$$

However, the simplifications made to get this bound may appear brutal in some situations. Even if these simplifications are nearly optimal in some specific examples, I would say that these bounds are nearly quadratic. It is polynomial in any case, which is already a good point. Even it would be quadratic, that is to say inn the same class than the bounds obtained for polynomials T_i, experiments show that the coefficients are usually smaller than the coefficients of the polynomial $T_{\ell+1}$.

Denote by $\mathcal{C}_V$ the Chow form of V and for simplicity $\mathcal{C}_i$ the Chow form of $\pi^n_i(V)$ instead of $\mathcal{C}_{\pi^n_i(V)}$. We prove that (Theorem 2.5):

Theorem. *Let $M_1 = T_1$ and for $1 \leq \ell \leq n-1$, define the derivation $\partial \in A_{\ell+1}(K) \subset \mathrm{Der}_K(K[X_1, \ldots, X_{\ell+1}])$ as follows:*

$$\partial := \partial_2^{(d_2-1)d_3\cdots d_{\ell+1}} \partial_3^{(d_3-1)d_4\cdots d_{\ell+1}} \cdots \partial_\ell^{(d_\ell-1)d_{\ell+1}}.$$

Define $M_{\ell+1}(X_1, \ldots, X_{\ell+1}) = \partial\partial_{\ell+1}^{d_{\ell+1}}(\mathcal{C}_{\ell+1})(1, 0, \ldots, 0, X_1)$. Then:

$$h(M_{\ell+1}) \leq \begin{cases} h\big(\pi^n_{\ell+1}(V)\big) + \deg\big(\pi^n_{\ell+1}(V)\big)\Big(\log(\ell+3) + \log\deg(\pi^n_{\ell+1}(V))\Big), & \textit{(number field)} \\ h(\pi^n_{\ell+1}(V)) \leq \deg(\mathfrak{V}) & \textit{(function field)} \end{cases}$$

It is almost linear in the degree and the height, but the bound is a bit less better than the bound obtained for the polynomial $N_{\ell+1}$. Moreover the degrees in $X_1, \ldots, X_\ell$ are much higher, since $\deg_{X_i}(M_{\ell+1}) = (d_i - 1)d_{i+1} \cdots d_{\ell+1}$. And last but not least, except by calculating the Chow form, which is not an easy task (and rely on Gröbner basis computation), I do not know how to compute these polynomials. However, the theoretical bound is useful for getting the bounds on the polynomials NF_i, since (Theorem 2.4):

$$NF_i \equiv M_i \bmod (T_1, \ldots, T_{i-1}).$$

Comments. The bounds given in this Chapter are intrinsic, that is to say, only depend on quantities attached to the underlying variety: the height and the degree. It is more general to state them in this way, since it does not depend on the datum of a polynomial system. However, the height of a variety is tedious to compute, since it was not introduced at all in this aim. Hence having at hand bounds involving quantities attached to an input polynomial system is of interest.

For example, a polynomial system over K whose zero-set is equiprojectable, has a lexicographic Gröbner basis which is a triangular set. If d and h are the maximal degree and height of the polynomials of the system, then it is natural to want to estimate the height of the polynomials of the output Gröbner basis in function of d and h. Using the geometric and arithmetic Bézout theorem 1.5 permits to derive such bounds form our intrinsic ones: this is how we fill the column "Extrinsic bounds" in Table 2.1.

We can compare the bounds for the triangular and primitive element representation. In fact, if $(\chi_u, W_1, \ldots, W_n)$ is a Shape Lemma representation defined by Equation (1.2) then the polynomials $T_1, \ldots, T_{n+1}$ hereunder form a triangular set.

$$\left|\begin{array}{l} T_1(X_1) = \chi_u(X_1) \\ T_2(X_1, X_2) = X_2 - W_2(X_1) \\ \quad \vdots \\ T_n(X_1, \ldots, X_n) = X_n - W_{n-1}(X_1) \\ T_{n+1}(X_1, \ldots, X_{n+1}) = X_{n+1} - W_n(X_1) \end{array}\right. . \tag{2.2}$$

Then the polynomials $N_1, \ldots, N_{n+1}$ verify:

$$\left|\begin{array}{l} N_1(X_1) = \chi_u(X_1) \\ N_2(X_1, X_2) = \chi'_u X_2 - w_2(X_1) \\ \quad \vdots \\ N_n(X_1, \ldots, X_n) = \chi'_u X_n - w_{n-1}(X_1) \\ N_{n+1}(X_1, \ldots, X_{n+1}) = \chi'_u X_{n+1} - w_n(X_1) \end{array}\right. , \tag{2.3}$$

where $\chi_u, w_1, \ldots, w_n$ is a Kronecker representation (or RUR) defined in Definition 1.2. Let $V \subset \mathbb{A}^n_{\bar{K}}$ be the variety parametrized by this representation, and $V' \subset \mathbb{A}^{n+1}_{\bar{K}}$ the variety defined by $T_1, \ldots, T_{n+1}$. If $\deg(V) = D$, then $\deg(\pi_i^{n+1}(V')) = D$ for $1 \leq i \leq n+1$. In Corollary 2.4, we obtain resuls summarized in Figure 2.1, restricted to the most interesting case of a number field.
Conclusion bounds for the regular chain $N_1, \ldots, N_n$ are of the same order than the bounds for the Kronecker representation.

Polynomials	Definition	Kind of fields	Intrinsic bound	Extrinsic bound	Theorem Section
$(T_1, \ldots, T_n)$	Def. 1.4	Number	$\deg(V)\Big(h(V) + 5\log(n+3)\deg(V) + 3\log(2)\deg(V)\Big)$	$O(nhd^{2n})$	Th. 2.7, § 2.2.3
		Function	$2\deg(\mathfrak{V})^2$	$2d^{2n}$	
$(N_1, \ldots, N_n)$	Def. 2.3	Number	$h(V) + 5\log(n+3)\deg(V)$	$O(nhd^n)$	Th. 2.7, § 2.2
		Function	$\deg(\mathfrak{V})$	d^n	
$(M_1, \ldots, M_n)$	Def. 2.2	Number	$h(V)+\deg(V)\Big(\log(n+3)+\log\deg(V)\Big)$	$O(nhd^n)$	Th. 2.5, § 2.1.4
		Function	$\deg(\mathfrak{V})$	d^n	
$(NF_1, \ldots, NF_n)$	§ 2.1.5	Number	$h(V) + \deg(V)\Big(\log\big((n+2)\deg(V)\big) + (n-1)\deg(V)h(V) + 5n^2 + 4n^3\deg(V)\Big)$	$O(n^3hd^{3n})$	Th. 2.6, § 2.1.5
		Function	$n\deg(\mathfrak{V}))^3$	nd^{3n}	
$(\chi_u, w_1, \ldots, w_n)$	Def. 1.2	Number	$h(V) + \deg(V)h(U) + \deg(V)\log(n+2) + (n+1)\log(D)$	$O\big(n^2hd^n\big)$	Th. 2.2 § 2.1.2
		Function	$\deg(\mathfrak{V})(1+\deg(\mathfrak{u})$	$d^n\deg(\mathfrak{u})$	

Table 2.1: Summary of the results and extrinsic bounds

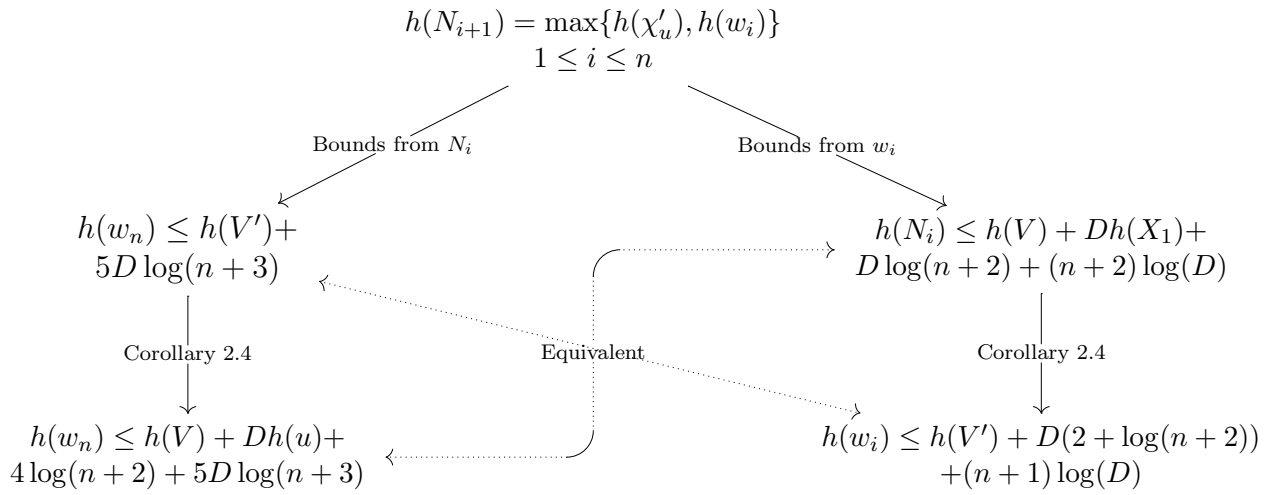

Figure 2.1: Comparison of bounds for N_i and for Kronecker representation

In this last section, we compare the representations by the polynomials T_ℓ, N_ℓ, and NF_ℓ from practical viewpoint. We do not mention the polynomials M_i, because as stated before they are not easily computable. We set $K = K_0 = \mathbb{Q}$ and compare the bit-size of the coefficients of these polynomials for various systems coming from applications. In our

DATA	Syst.	P19	Bersh.	Hawes	J1J2J3
	Var.	5	4	7	4
	Deg.	31 , 1 , 2 , 1 , 1	12 , 2 , 1 , 1	30 , 4 , 1 , 1 , 1 , 1 , 1	5 , 2 , 3 , 1
$T_1, \ldots, T_n$	h_{num}	90 , 1444 , 1029 ,	15 , 58 ,	77 , 1560 , 1558 , 1563 ,	13 , 25 ,
	T_ℓ	1444 , 1467	57 , 72	1564 , 1561 , 1560	24 , 39
	h_{den}	30 , 1448 , 1031 ,	5 , 57 ,	46 , 1560 , 1557 , 1561	19 , 24 ,
	T_ℓ	1450 , 1483	57 , 70	1561 , 1563 , 1560	25 , 39
$N_1, \ldots, N_n$	h_{num}	90 , 94 , 117 ,	15 , 17 ,	77 , 80 , 78 , 78 ,	13 , 17 ,
	N_ℓ	117 , 117	17 , 29	79 , 118 , 80	21 , 17
	h_{den}	30 , 28 , 44 ,	5 , 5 ,	46 , 48 , 47 ,	19 , 2 ,
	N_ℓ	44 , 62	5 , 18	46 , 46 , 85 , 47	8 , 5
$_{NF_1, \ldots, NF_n}$	h_{num}	92 , 85 , 489 ,	15 , 29 ,	77 , 661 , 661 ,	13 , 26 ,
	NF_ℓ	490 , 400	29 , 38	661 , 694 , 694	96 , 95
	h_{den}	28 , 60 , 342 ,	5 , 20 ,	46 , 558 , 558 ,	0 , 9 ,
	NF_ℓ	342, 230	20 , 34	551 , 554 , 591 ,561	33 , 31

Table 2.2: Number of digits of coefficients for 4 systems

experiments, the representation N_ℓ always leads to smaller coefficients, sometimes by an important factor. These systems, called Bershenko, P19, Hawes and J1J2J3, together with background information, are given in [102, Annexe E].

How to read Table 2.2 ? The second line gives the number n of variables of the system the third returns the lists of degrees $[d_1, \ldots, d_n]$ of the polynomials $[T_1, \ldots, T_n]$; for example, the system called Bershenko has 4 variables, and the triangular set T_1, T_2, T_3, T_4 representing it has degree 12, 2, 1, 1. Then the reminding rows are divided in three parts: $T_1, \ldots, T_n$, $N_1, \ldots, N_n$ and $_{NF_1, \ldots, NF_n}$. Each of these three parts is composed of two rows. The first one, denoted h_{num} returns the maximal number of digits of the integer at the numerator, among all the rational coefficients of the polynomial considered. The second line gives the same series of numbers, but for the denominator.

We observe a systematic diminution of the size of the coefficients for the polynomials N_ℓ, which is sometimes quite important: our conclusion is that using the polynomials N_ℓ is a good choice in practice. For the polynomials $NF_1, \ldots, NF_n$, it appears that coefficients are often smaller than the ones of polynomial $T_1, \ldots, T_n$, whereas their bounds are similar.

The first section is devoted to prove the bounds on the Kronecker representation and on the polynomials $M_1, \ldots, M_n$ and $NF_1, \ldots, NF_n$. It relies on technical results on the behavior of polynomials under derivations, proved in Subsection 2.1.1, and may be skipped for a first reading. The second section mostly presents the results of the article [32] written with the collaboration of É. Schost. The bounds for $T_{\ell+1}$ and $N_{\ell+1}$ proved therein use new interpolation formula exploiting the very simple shape of a triangular set.

2.1 Bounds from derivation of the Chow form

The first application of this technique is to obtain bounds for primitive element representation (see Subsection 2.1.2) using the analogy of the Chow form and the primitive element, as seen in Subsection 1.2.1. Then we extend the result to triangular sets; as it is more tedious, all the technical results are gathered in the following subsection.

2.1.1 Formulas of derivations

We aim at proving the following result (Proposition 2.3) in this subsection. It will be central in the proof of our main result (Theorem 2.4) about the invertibilty of the leading coefficient of polynomial $M_{\ell+1}$ in Section 2.1.3 Let V be an equiprojectable variety defined by a triangular set $T_1, \ldots, T_n$ over K, with degrees $d_1, \ldots, d_n$. The Chow form of $\pi_i^n(V)$ is denoted by $\mathcal{C}_i$ instead of $\mathcal{C}_{\pi_i^n(V)}$.

Proposition. *Let $1 \leq \ell \leq s$ be two integers and set $d_{\geq \ell} := d_\ell d_{\ell+1} \ldots d_s$, and $d_{<\ell} = d_1 \ldots d_{\ell-1}$. Consider also some integers $n_\ell, \ldots, n_s, n_T$, satisfying $\sum_{i=\ell}^{s} n_i + n_T = S \leq d_{\geq \ell} - 1$. Then the derivation*

$$\partial := \partial_{U_\ell}^{n_\ell} \partial_{U_{\ell+1}}^{n_\ell+1} \ldots \partial_{U_s}^{n_s} \partial_T^{n_T},$$

verifies the following property:

$$\mathcal{C}_{\ell-1}(U_1, \ldots, U_{\ell-1}, T) \mid \partial(\mathcal{C}_s)(U_1, \ldots, U_{\ell-1}, 0, \ldots, 0, T).$$

We prove a series of results involving derivations. We recall their definition:

Definition 2.1 (Derivation)**.** *Let K be a field and R a commutative K-algebra. A K-linear map $d : R \to R$ is a derivation of order 1 if: for all x, $y \in R$, $d(xy) = xd(y) + d(x)y$. Recursively, we define a derivation of order n by saying that it is a K-linear map $D : R \to R$ such that $\forall\, x$, $y \in R$, $y \mapsto D(xy) - xD(y) - yD(x)$ is a derivation of order $n-1$. The set of K-derivations over R is denoted by $\mathrm{Der}_K(R)$, and it is a (non-commutative) K-algebra.*

Example 2.1: If $R = K[X_1, \ldots, X_n]$ we define derivations of order 1:

$$\text{for } i = 1, \ldots, n \qquad \partial_{X_i}(X_1^{\alpha_1} \cdots X_n^{\alpha_n}) := \alpha_i X_1^{\alpha_1} \cdots X_i^{\alpha_i - 1} \cdots X_n^{\alpha_n}.$$

They commute each other. The sub-K-algebra $A_n(K)$ of $\mathrm{Der}_K(K[X_1, \ldots, X_n])$ they generate is called the *n-th Weyl algebra*. It is isomorphic to $K[X_1, \ldots, X_n, Y_1, \ldots, Y_n]/\mathcal{R}$, where $\mathcal{R}$ is the ideal of relations generated by $[X_i, X_j] = [X_i, Y_j] = [Y_i, Y_j] = 0$ if $i \neq j$ and $[X_i, Y_i] = 1$ ($[.,.]$ is the usual Lie product). ∂_{X_i} is then identified to Y_i.

The following lemma is not of use for this paragraph, but as it deals with derivations, we state it here:

Lemma 2.1. *Let $A \in K[U_1, \ldots, U_i][T, X_1, \ldots, X_{i-1}]$ and ∂_i be the derivation of order 1 equal to $\partial_{U_i} + X_i \partial_T$. Then:*

$$\forall\, k \in \mathbb{N} \qquad \partial_{X_i} \partial_i^k(A) = k\, \partial_T \partial_i^{k-1}(A).$$

Proof. Since ∂_{U_i}, X_i and ∂_T commute, $\partial_i^k = \sum_{j=0}^k \binom{k}{j} \partial_{U_i}^j X_i^{k-j} \partial_T^{k-j}$. It follows:

$$\begin{aligned}
\partial_{X_i}\partial_i^k(A) &= \partial_{X_i}\left(\sum_{j=0}^{k}\binom{k}{j}\partial_{U_i}^j X_i^{k-j}\partial_T^{k-j}\right)(A) \\
&= \sum_{j=0}^{k}\binom{k}{j}\partial_{U_i}^j\partial_T^{k-j}\partial_{X_i}(X_i^{k-j}\cdot A) \\
&= \partial_T\left(\sum_{j=0}^{k-1}(k-j)\binom{k}{j}\partial_{U_i}^j X_i^{k-j-1}\partial_T^{k-j-1}\right)(A).
\end{aligned}$$

The last equality follows from the fact that $\partial_{X_i}(X_i^\alpha.A) = \alpha X_i^{\alpha-1}.A$ since by hypothesis A does not involve X_i. The equality $(k-j)\binom{k}{j} = k\binom{k-1}{j}$ permits then to conclude. □

We want now some formulas for the derivative of a product of polynomials. First with one derivative (Proposition 2.1), and then for several (Corollary 2.1). At last, we apply these formulas in the special case of products of linear forms, with specific derivations to get the fundamental Proposition 2.2. It will be applied to Chow forms, which are product of linear forms: this is how the Proposition 2.3 is obtained.

Proposition 2.1. *Let F be a commutative field, $f_1, \ldots, f_s$ some polynomials in $F[X_1, \ldots, X_n]$ and ∂ a derivation of order 1 of $A_n(F)$. We have the following (generalized Leibniz) formula:*

$$\partial^k(f_1 \cdots f_s) = \sum_{\substack{\mathbf{j}=(j_1,\ldots,j_s)\in\mathbb{N}^s \\ |\mathbf{j}|=k}} \binom{k}{\mathbf{j}} \partial^{j_1}(f_1)\cdot\partial^{j_2}(f_2)\cdots\partial^{j_s}(f_s)$$

where $\binom{k}{\mathbf{j}} = \frac{k!}{j_1!\ldots j_s!}$ *and* $|\mathbf{j}| = \sum_{t=1}^s j_t$

Proof. By induction on k. For $k = 1$, it is the well-known Leibniz formula. Suppose the formula is true at rank k, and let us show it at rank $k+1$. We are led to consider $\partial\left(\prod_{t=1}^s \partial^{j_t}(f_t)\right)$, which is equal to $\sum_{t=1}^s \partial^{j_t+1}\prod_{\alpha\neq t}\partial^{j_\alpha} f_\alpha$ (due to the Leibniz formula). So,

$$\partial^{k+1}(f_1\ldots f_s) = \sum_{\substack{\mathbf{j}=(j_1,\ldots,j_s) \\ |\mathbf{j}|=k}} \binom{k}{\mathbf{j}} \left(\sum_{t=1}^s \partial^{j_t+1}\prod_{\alpha\neq t}\partial^{j_\alpha}(f_\alpha)\right). \tag{2.4}$$

Denote by $\mathcal{S}_{s,k}$ the set of s-uples whose sum is k, $\mathcal{S}_{s,k} := \{(j_1,\ldots,j_s)\in\mathbb{N}^s \mid \sum_{i=1}^s j_i = k\}$. Consider the application:

$$\begin{aligned}
\phi\ :\ \mathcal{S}_{s,k} &\longrightarrow \text{Set of sets of cardinality } s \text{ of elements in } \mathcal{S}_{s,k+1} \\
\mathbf{j} &\longmapsto \{(j_1+1, j_2, \ldots, j_s), (j_1, j_2+1, j_3, \ldots, j_s), \ldots, (j_1, \ldots, j_{s-1}, j_s+1)\}
\end{aligned}$$

Then the equation 2.4 is rewritten:

$$\partial^{k+1}(f_1\ldots f_s) = \sum_{\substack{\mathbf{j}=(j_1,\ldots,j_s) \\ |\mathbf{j}|=k}} \binom{k}{\mathbf{j}} \sum_{(\ell_1,\ldots,\ell_s)\in\phi(\mathbf{j})} \partial^{\ell_1}(f_1)\ldots\partial^{\ell_s}(f_s) \tag{2.5}$$

It is easy to see that

$$\bigcup_{\mathbf{j}\in\mathcal{S}_{s,k}} \cup_{\mathbf{l}\in\phi(\mathbf{j})}\mathbf{l} = \mathcal{S}_{s,k+1}.$$

Hence the formula (2.5) is indeed of the shape:

$$\sum_{\substack{\mathbf{l}=(\ell_1,\ldots,\ell_s)\\ |\mathbf{l}|=k+1}} C(\mathbf{l})\,.\,\prod_{i=1}^{s}\partial^{\ell_t}(f_t),$$

where $C(\mathbf{l})\in F$ is only dependent on $\mathbf{l}$. It remains to determine the coefficient $C(\mathbf{l})$. Let us fix $\mathbf{l}=(\ell_1,\ldots,\ell_s)\in\mathcal{S}_{s,k+1}$, and define $\mathcal{E}_{\mathbf{l}} := \{\beta\in\{1,\ldots,s\}\,|\,\ell_\beta>0\}$. For each $\beta\in\mathcal{E}_{\mathbf{l}}$, we can associate to $\mathbf{l}$ an unique element $\mathbf{j}\in\mathcal{S}_{s,k}$: it suffices to take $j_\beta=\ell_\beta-1$ and $j_\alpha=\ell_\alpha, \forall\alpha\neq\beta$. We get:

$$C(\mathbf{l}) = \sum_{\substack{\mathbf{j}\in\mathcal{S}_{s,k}\\ \mathbf{l}\in\phi(\mathbf{j})}} \binom{k}{\mathbf{j}} = \sum_{\beta\in\mathcal{E}_{\mathbf{l}}} \frac{k!}{\prod_{\alpha\neq\beta}\ell_\alpha!(\ell_\beta-1)!} = k!\left(\frac{\sum_{\beta\in\mathcal{E}_{\mathbf{l}}}\ell_\beta}{\prod_{\alpha=1}^{s}\ell_\alpha!}\right) = \binom{k+1}{\mathbf{l}}.$$

This concludes the proof. □

Corollary 2.1. *Let $\delta_1,\ldots,\delta_i$ be some derivations of order 1, and $k_1,\ldots,k_i$ some non-negative integers. With the same polynomials f_i of the previous proposition we have:*

$$\delta_1^{k_1}\delta_2^{k_2}\cdots\delta_i^{k_i}(f_1\cdots f_s) = \sum_{\substack{\mathbf{j}^{(1)},\ldots,\mathbf{j}^{(i)}\\ \mathbf{j}^{(r)}\in\mathbb{N}^s\,,\,|\mathbf{j}^{(r)}|=k_r}} \binom{k_1}{\mathbf{j}^{(1)}}\cdots\binom{k_i}{\mathbf{j}^{(i)}}\prod_{t=1}^{s}\delta_1^{j_t^{(1)}}\delta_2^{j_t^{(2)}}\cdots\delta_i^{j_t^{(i)}}(f_t).$$

Proof. By induction on i, the previous proposition giving the case $i=1$. We suppose that the corollary is true for a product of $i-1$ derivations indexed from 2 to i; so that:

$$\delta_2^{k_2}\ldots\delta_i^{k_i}(f_1\ldots f_s) = \sum_{\substack{\mathbf{j}^{(2)},\ldots\mathbf{j}^{(i)}\\ \mathbf{j}^{(r)}\in\mathbb{N}^s\,,\,|\mathbf{j}^{(r)}|=k_r}} \binom{k_2}{\mathbf{j}^{(2)}}\cdots\binom{k_i}{\mathbf{j}^{(i)}}\prod_{t=2}^{s}\delta_2^{j_t^{(2)}}\delta_3^{j_t^{(3)}}\ldots\delta_i^{j_t^{(i)}}(f_t). \tag{2.6}$$

Let δ_1 be a derivation of order 1, and $k_1\in\mathbb{N}$. Due to the generalized Leibniz formula from the previous Proposition:

$$\delta_1^{k_1}\left(\prod_{t=1}^{s}\delta_2^{j_t^{(2)}}\delta_3^{j_t^{(3)}}\ldots\delta_i^{j_t^{(i)}}(f_t)\right) = \sum_{\substack{\mathbf{j}^{(1)}=(j_1^{(1)},\ldots,j_s^{(1)})\\ |\mathbf{j}^{(1)}|=k_1}} \binom{k_1}{\mathbf{j}^{(1)}}\prod_{t=1}^{s}\delta_1^{j_t^{(1)}}\ldots\delta_i^{j_t^{(i)}}(f_t).$$

If we report this sum in equation (2.6), we obtain the required formula. □

Proposition 2.2. *Let ℓ and s be two integers such that $2\leq\ell<s$. Let $(a_{i,j})_{\substack{1\leq i\leq s\\ 1\leq j\leq\ell}}$ and $(b_{i,j,k})_{\substack{1\leq i\leq s\\ 1\leq j\leq r\\ \ell\leq k\leq n}}$ be elements of $\bar{K}$. We set for all $i=1,\ldots,s$:*

$$f_i := \prod_{j=1}^{r}(a_{i,1}X_1+\cdots+a_{i,\ell-1}X_{\ell-1}+b_{i,j,\ell}X_\ell+\cdots+b_{i,j,n}X_n),$$

that will be written more simply: $f_i := \prod_{j=1}^{r}(X'(a_i)+X''(b_{i,j}))$. *Let* $k_\ell, \dots, k_n$ *be some integers such that* $\sum_{\alpha=\ell}^{n} k_\alpha = S$. *Then,*

$$\partial_{X_\ell}^{k_\ell}\partial_{X_{\ell+1}}^{k_{\ell+1}}\cdots\partial_{X_n}^{k_n}(f_1\cdots f_s) = \sum_{\substack{\mathbf{J}=(J_1,\dots,J_s)\\ |\mathbf{J}|=S\\ J_i\subseteq\{1,\dots,r\}\cup\{\emptyset\}}} C(\mathbf{J})\prod_{i=1}^{s}C(J_i)\prod_{j\notin J_i}(X'(a_i)+X''(b_{i,j})),$$

where $C(\mathbf{J})$ *and* $C(J_i)$ *are constants only dependent on* $\mathbf{J}$ *and* J_i *respectively.*

Proof. Let us set $\partial := \partial_{X_\ell}^{k_\ell}\partial_{X_{\ell+1}}^{k_{\ell+1}}\cdots\partial_{X_n}^{k_n}$. From Corollary 2.1:

$$\partial(f_1\cdots f_s) = \sum_{\substack{\mathbf{j}^{(\ell)},\dots,\mathbf{j}^{(n)}\in\mathbb{N}^s\\ |\mathbf{j}^{(\alpha)}|=k_\alpha,\ \ell\le\alpha\le n}} C(\mathbf{j}^{(\ell)},\dots,\mathbf{j}^{(n)})\prod_{t=1}^{s}\partial_{X_\ell}^{j_t^{(\ell)}}\cdots\partial_{X_n}^{j_t^{(n)}}(f_t), \tag{2.7}$$

Let us fix some s-uples $(\mathbf{j}^{(\alpha)})_{\alpha=\ell,\dots,n}$ as well as t in $\{1,\dots,s\}$, and denote $\partial_0 := \partial_{X_\ell}^{j_t^{(\ell)}}\cdots\partial_{X_n}^{j_t^{(n)}}$. We focus now on $\partial_0(f_t)$. We aim at proving:

$$\partial_0(f_t) := \partial_{X_\ell}^{j_t^{(\ell)}}\dots\partial_{X_n}^{j_t^{(n)}}(f_t) = \sum_{\substack{J\subseteq\{1,\dots,r\}\\ |J|=j_t^{(\ell)}+\cdots+j_t^{(n)}}} C(J)\prod_{j\notin J}(X'(a_i)+X''(b_{i,j})). \tag{2.8}$$

We need the following for that purpose:

Lemma 2.2. *Let* $g=\prod_{i=1}^{p}(a_iX+b_i)$ *where* a_i, b_i *are in a field* L *extension of a field* L_0, *and an integer* $1\le d<p$. *The following formula holds:*

$$\partial_X^d(g) = \sum_{\substack{\{i_1,\dots,i_d\}\subseteq\\ \{1,\dots,p\}}} C(\{i_1,\dots,i_d\})\prod_{i\notin\{i_1,\dots,i_d\}}(a_iX+b_i),$$

where $C(\{i_1,\dots,i_d\})\in L$ *is only dependent on the set* $\{i_1,\dots,i_d\}$.

Proof. By induction on d, the case $d=1$ being easy. Let us show the formula at rank $d+1$, supposing it is true at rank d:

$$\begin{aligned}
\partial_X^{d+1}(g) &= \partial_X\left(\sum_{\substack{\{i_1,\dots,i_d\}\\ \subseteq\{1,\dots,p\}}} C(\{i_1,\dots,i_d\})\prod_{i\notin\{i_1,\dots,i_d\}}(a_iX+b_i)\right)\\
&= \sum_{\substack{\{i_1,\dots,i_d\}\\ \subseteq\{1,\dots,p\}}} C(\{i_1,\dots,i_d\})\partial_X\left(\prod_{i\notin\{i_1,\dots,i_d\}}(a_iX+b_i)\right)\\
&= \sum_{\substack{\{i_1,\dots,i_d\}\\ \subseteq\{1,\dots,p\}}} C(\{i_1,\dots,i_d\})\left(\sum_{\substack{i_{d+1}=1\\ i_{d+1}\notin\{i_1,\dots,i_d\}}}^{p} C(i_{d+1})\prod_{i\notin\{i_1,\dots,i_{d+1}\}}(a_iX+b_i)\right)\\
&= \sum_{\substack{\{i_1,\dots,i_{d+1}\}\\ \subseteq\{1,\dots,p\}}} C(\{i_1,\dots,i_{d+1}\})\prod_{i\notin\{i_1,\dots,i_{d+1}\}}(a_iX+b_i),
\end{aligned}$$

where $C(\{i_1,\dots,i_{d+1}\}) = C(\{i_1,\dots,i_d\})C(i_{d+1})$. □

Back to the proposition : The proof of Equation (2.8) is done by decreasing induction on ℓ. For $\ell = n$ (which corresponds to the initialization of our induction), we have, thanks to the previous lemma applied with $X = X_n$, $L = K(X_1,\dots,X_{n-1})$ and $L_0 = K$:

$$\partial_{X_n}^{j_t^{(n)}}(f_t) = \sum_{\substack{J\subseteq\{1,\dots,r\}\\ |J|=j_t^{(n)}}} C(J)\prod_{i\notin J}(X'(a_i)+X''(b_{i,j})).$$

Suppose the formula exact at rank $\ell+1$, and let us show it at rank ℓ:

$$\partial_0(f_t) = \partial_{X_\ell}^{j_t^{(\ell)}}\left(\sum_{\substack{J\subseteq\{1,\dots,r\}\\ |J|=j_t^{(\ell+1)}+\cdots+j_t^{(n)}}} C(J)\prod_{j\notin J}(X'(a_i)+X''(b_{i,j}))\right). \tag{2.9}$$

So we look at $\partial_{X_\ell}^{j_t^{(\ell)}}\left(\prod_{j\notin J}(X'(a_i)+X''(b_{i,j}))\right)$. Again from the previous Lemma applied with $X = X_\ell$, $L = K(X_1,\dots,X_{\ell-1},X_{\ell+1},\dots,X_n)$, $d = j_t^{(\ell)}$ and $p = r-|J|$

$$\partial_{X_\ell}^{j_t^{(\ell)}}\left(\prod_{j\notin J}(X'(a_i)+X''(b_{i,j}))\right) = \sum_{\substack{J'\subseteq\{1,\dots,r\}\setminus J\\ |J'|=j_t^{(\ell)}}} C(J,J')\prod_{j\notin J'\cup J}(X'(a_i)+X''(b_{i,j})),$$

where $C(J,J')$ is only dependent on J and J'. This sum is to be added to each terms indexed by the J in (2.9). But,

$$\left\{J\cup J' \text{ such that } J\,,\ J'\subseteq\{1,\dots,r\} \text{ and } J\cap J'=\emptyset \text{ and } |J| = j_t^{(\ell+1)}+\cdots+j_t^{(n)} \right.$$
$$\left. \text{and } |J'| = j_t^{(\ell)}\right\} = \left\{J''\subseteq\{1,\dots,r\},\ |J''| = j_t^{(\ell)}+\cdots+j_t^{(n)}\right\}.$$

Hence taking into account this new way of indexing, Equation (2.9) becomes:

$$\partial_0(f_t) = \sum_{\substack{J''\subseteq\{1,\dots,r\}\\ |J''|=j_t^{(\ell)}+\cdots+j_t^{(n)}}} C(J'')\prod_{j\notin J''}(X'(a_i)+X''(b_{i,j})),$$

where $C(J'')\in K$ is only dependent on the set J''. This is Equation (2.8). So, Equation (2.7) is rewritten:

$$\partial(f_1\dots f_s) = \sum_{\substack{\mathbf{j}^{(\ell)},\dots,\mathbf{j}^{(n)}\\ |\mathbf{j}^{(r)}|=d_\alpha,\forall\alpha}} C(\mathbf{j}^{(\ell)},\dots,\mathbf{j}^{(n)})\prod_{t=1}^{s}\left(\sum_{\substack{J\subseteq\{1,\dots,r\}\\ |J|=j_t^{(\ell)}+\cdots+j_t^{(n)}}} C(J)\prod_{j\notin J}(X'(a_i)+X''(b_{i,j}))\right).$$

A classical formula, of type $\prod_{i=1}^n(\sum_{j=1}^m a_{i,j}) = \sum_{1\le\ell_1,\dots,\ell_n\le m}\prod_{i=1}^n a_{i,\ell_i}$, gives in our context:

$$\prod_{t=1}^{s}\left(\sum_{\substack{J\subseteq\{1,\dots,r\}\\ |J|=j_t^{(\ell)}+\cdots+j_t^{(n)}}} C(J)\prod_{j\notin J}(X'(a_i)+X''(b_{i,j}))\right) = \sum_{\substack{\mathbf{J}=(J_1,\dots,J_s)\\ |J_t|=j_t^{(\ell)}+\cdots+j_t^{(n)}}} C(\mathbf{J})\prod_{t=1}^{s}C(J_t)\prod_{j\notin J_t}(X'(a_i)+X''(b_{i,j})).$$

Again, here $C(\mathbf{J})$ and $C(J_t)$ only depends of the s-uple of sets $\mathbf{J}$ and on the set J_t respectively. At last, to arrive at the required formula, it suffices to see that:

$$\bigcup_{\substack{\mathbf{j}^{(\ell)},\dots,\mathbf{j}^{(n)} \\ |\mathbf{j}^{(\alpha)}|=k_\alpha,\ \ell\leq\alpha\leq n}} \{(J_1,\dots,J_s),\ |J_t| = j_t^{(\ell)}+\cdots+j_t^{(n)}\} \ = \ \{(J_1,\dots,J_s),\ J_i\subset\{1,\dots r\}\cup\{\emptyset\}, \text{ and } |J_1|+\cdots+|J_s|=S\}$$

This is precisely the set on which holds the sum of the proposition. Note that the union is not disjoint, which possibly makes appear constants $C(\mathbf{J})$, but as stated before, computing them is not useful for our purpose. □

Finally, we are able to prove the important divisibility result announced in the beginning of this subsection. This result will be used intensively in Paragraph 2.1.3 and is an outcome of the previous results of this subsection. Given an equiprojectable variety $V\subset A^n_{\bar{K}}$ defined over K, with d_i as cardinal of the fibers of π^i_{i-1}, for $2\leq i\leq n$ and d_1 as cardinal of the points of $\pi^n_1(V)$, we have:

Proposition 2.3. *Let $1\leq\ell\leq s$ be two integers and set $d_{\geq\ell}:=d_\ell d_{\ell+1}\dots d_s$, and $d_{<\ell}=d_1\dots d_{\ell-1}$. Consider also some integers $n_\ell,\dots,n_s,n_T$, satisfying $\sum_{i=\ell}^s n_i+n_T=S\leq d_{\geq\ell}-1$. Then the derivation*

$$\partial := \partial_{U_\ell}^{n_\ell}\,\partial_{U_{\ell+1}}^{n_\ell+1}\,\dots\,\partial_{U_s}^{n_s}\,\partial_T^{n_T},$$

verifies the following property:

$$\mathcal{C}_{\ell-1}(U_1,\dots,U_{\ell-1},T)\mid\partial(\mathcal{C}_s)(U_1,\dots,U_{\ell-1},0,\dots,0,T).$$

Proof. From Proposition 1.4, and by definition of an equiprojectable variety, we have:

$$\mathcal{C}_s := \prod_{\alpha\in\pi^n_{\ell-1}(V)}\ \prod_{\beta\in(\pi^s_{\ell-1})^{-1}(\alpha)} (T-U_1\alpha_1-\cdots-U_{\ell-1}\alpha_{\ell-1}-U_\ell\beta_\ell-\cdots-U_s\beta_s).$$

Before applying Proposition 2.2 with the linear forms f_i equal to $\prod_{\beta\in(\pi^s_{\ell-1})^{-1}(\alpha)}(T-U_1\alpha_1-\cdots-U_{\ell-1}\alpha_{\ell-1}-U_\ell\beta_\ell-\cdots-U_s\beta_s)$, a homogenization between the notations of Proposition 2.2 and the ones here is necessary. For each $\alpha\in\pi^n_{\ell-1}(V)$, consider a bijection ϕ_α:

$$\phi_\alpha\ :\ \{1,\dots,d_{\geq\ell}\}\longrightarrow(\pi^s_{\ell-1})^{-1}(\{\alpha\}).$$

Then the summation of Proposition 2.2 is rewritten as:

$$\partial(\mathcal{C}_s)(U_1,\dots,U_s,T) := \sum_{\substack{\mathbf{J}=(J_1,\dots,J_{d_{<\ell}}) \\ |\mathbf{J}|=S \\ J_i\subset\{1,\dots,d_{\geq\ell}\}\cup\{\emptyset\}}} C(\mathbf{J})\prod_{\alpha\in\pi^n_{\ell-1}(V)} C(J_i)\prod_{\beta\notin\phi_\alpha(J_i)}(T-U_1\alpha_1-\cdots \cdots-U_{\ell-1}\alpha_{\ell-1}-U_\ell\beta_\ell-\cdots-U_s\beta_s).$$

Evaluating $U_\ell=U_{\ell+1}=\cdots=U_s=0$, yields:

$$\partial(\mathcal{C}_s)(U_1,\dots,U_{\ell-1},0,\dots,0,T) = \sum_{\substack{\mathbf{j}=(j_1,\dots,j_{d_{<\ell}}) \\ |\mathbf{j}|=S}} C(\mathbf{J})\prod_{\alpha\in\pi^n_{\ell-1}(V)}(T-U_1\alpha_1-\cdots-U_{\ell-1}\alpha_{\ell-1})^{d_{\geq\ell}-j_i}.$$

Since for all i, $j_i\leq S\leq d_{\geq\ell}-1$, all the exponents above are non-zero. So $\prod_{\alpha\in\pi^n_{\ell-1}(V)}(T-U_1\alpha_1-\cdots-U_{\ell-1}\alpha_{\ell-1}):=\mathcal{C}_\ell$, divides $\partial(\mathcal{C}_s)(U_1,\dots,U_{\ell-1},0,\dots,0,T)$. □

2.1.2 Bounds for primitive element representations

In this paragraph, we are interested in proving the estimate in Theorem 2.2. Previous estimates are given in terms of a suitable multiplication tensor in [3, 101], and polynomial-type bounds are also given for such representations in [47, 109, 102].

We consider a zero-dimensional variety V of a polynomial system over K. As usual, its vanishing ideal verifies the Separability Assumption. We are interested in recovering its primitive element representation by differentiating and specializing its Chow form (Theorem 2.1). It does not use the previous technical results, but deals also with derivations.

In this section, D is the degree $\deg(V)$ of V. The following lemma shows an important cancellation identity of the Chow form when specialized at a generic linear form.

Lemma 2.3. *For an integer $i \geq 1$, consider a derivation ∂ in a Weyl algebra $A_{i+1}(K[X_1, \ldots, X_n]) \subset \mathrm{Der}_{K[X_1,\ldots,X_n]}(K[X_1, \ldots, X_n][U_1, \ldots, U_i, T])$, such that ∂ is product of N derivations among $\{\partial_1, \ldots, \partial_i\}$. Consider also a polynomial A in $K[U_1, \ldots, U_i, T]$. Then,*

$$\partial(A \cdot \mathcal{C}_i)(U_1, \ldots, U_i, \sum_{1 \leq k \leq i} U_k X_k) \equiv 0 \bmod I(\pi_i^n(V)) \otimes K[U_1, \ldots, U_i].$$

Proof. Let us prove it by induction on N. The case $N = 0$ corresponds to Lemma 1.2 applied for each projection $\pi_i^n(V)$. Assume that the result is true for every derivation of $A_{i+1}(K)$ of order $N - 1$. By definition, the K-linear map:

$$\begin{array}{rccl} L & : & K[X_1, \ldots, X_n, U_1, \ldots, U_i, T] & \longrightarrow & K[X_1, \ldots, X_n, U_1, \ldots, U_i, T] \\ & & y & \longmapsto & \partial(A \cdot y) - A \cdot \partial(y) - y \cdot \partial(A). \end{array}$$

is a derivation of order $N - 1$. So that $\partial(\mathcal{C}_i \,.\, A) = A \,.\, \partial(\mathcal{C}_i) + \mathcal{C}_i \,.\, \partial(A) + L(\mathcal{C}_i)$. By the induction hypothesis,

$$L(\mathcal{C}_i)(U_1, \ldots, U_i, \sum_{k=1}^{i} U_k X_k) \equiv 0 \bmod I(\pi_i^n(V)) \otimes K[U_1, \ldots, U_i]$$

and Lemma 1.2 gives:

$$\mathcal{C}_i(U_1, \ldots, U_i, U_1 X_1 + \cdots + U_i X_i) \equiv 0 \bmod I(\pi_i^n(V)) \otimes K[U_1, \ldots, U_i]$$

Let us prove that $\partial(\mathcal{C}_i)(U_1, \ldots, U_i, U_1 X_1 + \cdots + U_i X_i)$ is also null modulo that ideal. This will conclude the proof. Let us see $\mathcal{C}_i(U_1, \ldots, U_i, \sum_{1 \leq \ell \leq i} U_\ell X_\ell)$ as a polynomial in $U_1, \ldots, U_i$ and with coefficients in $K[X_1, \ldots, X_i]$. Lemma 1.2 means that all the coefficients of this polynomial lie in $I(\pi_i^n(V))$. To prove that $\partial(\mathcal{C}_i)(U_1, \ldots, U_i, U_1 X_1 + \cdots + U_i X_i)$ is also null modulo $I(\pi_i^n(V)) \otimes K[U_1, \ldots, U_i]$, we prove the following lemma:

Lemma 2.4. *Let R be a ring, $f \in R[U_1, \ldots, U_\ell]$ and ∂ a derivation in $A_\ell(R)$, of order N. Suppose that the coefficients of f lie in an ideal I of R. Then the coefficients of $\partial(f)$ also belong to I.*

Proof. Let us write f as $\sum_{\mathbf{a} \in \mathbb{N}^\ell} f_{\mathbf{a}} U_1^{a_1} \cdots U_\ell^{a_\ell}$, with $f_{\mathbf{a}} \in I$. For $1 \leq i \leq \ell$, and an integer b_i we have:

$$\partial_{U_i}^{b_i}(f) = 0,$$

if $b_i > \max\{a_i | \mathbf{a} \in \mathbb{N}^s \text{and } f_{\mathbf{a}} \neq 0\}$. Else,

$$\partial_{U_i}^{b_i}(f) = \sum_{\substack{\mathbf{a}\in\mathbb{N}^s \text{ such} \\ \text{that } a_i \geq b_i}} f_{\mathbf{a}} \frac{a_i!}{(a_i - b_i)!} U_1^{a_1} \cdots U_i^{a_i - b_i} \cdots U_\ell^{a_\ell}.$$

As $f_{\mathbf{a}} \in I$, it follows that $\frac{a_i!}{(a_i-b_i)!} f_{\mathbf{a}} \in I$. Hence for every monomials $U = U_1^{a_1} \cdots U_\ell^{a_\ell}$, the polynomial $U(\partial_{U_1}, \ldots, \partial_{U_\ell})(f)$ has all its coefficients in I. As the derivatives ∂_{U_i} generates the Weyl algebra, so it is for $\partial(f)$. □

Back to the proof: We define the derivation δ from ∂ by replacing each factor ∂_j by ∂_{U_j}. If f denotes $\mathcal{C}_i(U_1, \ldots, U_\ell, U_1X_1 + \cdots + U_iX_i)$, then an easy induction gives:

$$\delta(f) = \partial(\mathcal{C}_i)(U_1, \ldots, U_i, U_1X_1 + \cdots + U_iX_i).$$

The previous lemma applied with $R = K[X_1, \ldots, X_i]$, with $I = I(\pi_i^n(V))$ and with $\partial = \delta$, leads to $\partial(\mathcal{C}_i)(U_1, \ldots, U_i, U_1X_1 + \cdots + U_iX_i)$ is null modulo $I(\pi_i^n(V)) \otimes K[U_1, \ldots, U_i]$. □

Now that we have at hand this cancellation identity that we use partially (with $A = 1$) in this section (but fully exploited in the next section), we start to investigate link between primitive element representations and Chow forms.

$$F(U_1, \ldots, U_n) = \mathcal{C}_V(U_1, \ldots, U_n, U_1X_1 + \cdots + U_nX_n),$$

where $F \in K[X_1, \ldots, X_n][U_1, \ldots, U_n]$, we get:

$$\partial_{U_i}(F)(U_1, \ldots, U_n) = (\partial_{U_i}(\mathcal{C}_V) + X_i\partial_T(\mathcal{C}_V))(U_1, \ldots, U_n, U_1X_1 + \cdots + X_nU_n). \tag{2.10}$$

Thanks to the previous Lemma applied with the derivation $\partial_i = \partial_{U_i} + X_i\partial_T$, it is null modulo $I(V) \otimes K[U_1, \ldots, U_n]$. Let $(u_1, \ldots, u_n) \in \bar{K}^n$ such that $u := u_1X_1 + \cdots + u_nX_n$ is a separating element in $K[X_1, \ldots, X_n]/I(V)$. In other words, the map $V \to \bar{K},\ \alpha \mapsto u(\alpha)$ is injective. We consider the specialization map $\varphi : K[U_1, \ldots, U_n] \to \bar{K},\ (U_i \mapsto u_i)_i$ and denote by $\chi_u(T)$ the characteristic polynomial of the endomorphism M_u of multiplication by u in $K[X_1, \ldots, X_n]/I(V)$. We have seen in Proposition 1.3 that $K[\mathbf{X}]/I(V) \otimes K[\mathsf{U}]$ is a $K[\mathsf{U}]$-free module of the same dimension as the K-vector space $K[\mathbf{X}]/I(V)$. So the following diagram

$$\begin{array}{ccccc}
K[\mathbf{X}]/I(V) \otimes K[\mathsf{U}] & \xrightarrow{\quad M_U \quad} & K[\mathbf{X}]/I(V) \otimes K[\mathsf{U}] & & \mathcal{C}_V(U_1, \ldots, U_n, T) \\
\Big\downarrow{\scriptstyle \varphi} & & \Big\downarrow{\scriptstyle \varphi} & & \Big\downarrow{\scriptstyle \varphi} \\
K[\mathbf{X}]/I(V) & \xrightarrow{\quad M_u \quad} & K[\mathbf{X}]/I(V) & & \chi_u(T)
\end{array}$$

(with $\det(T.\mathbf{I_d} - M_U)$ above the top row and $\det(T.\mathbf{I_d} - M_u)$ below the bottom row)

commutes, as the Chow form of V is the characteristic polynomial of M_U and the operation det commutes with the specialization φ. We get,

$$\varphi(\mathcal{C}_V(U_1, \ldots, U_n, T)) = \chi_u(T). \tag{2.11}$$

Moreover φ and ∂_T commute also, leading to:

$$\phi(\partial_T(\mathcal{C}_V)(U_1,\ldots,U_n,T)) = \chi'_u(T). \tag{2.12}$$

This equation with the following theorem provide the link sought between the Chow form and the Kronecker representation of Definition 1.2. As said before, this result is absolutely not new, nut is a warming-up for the generalization of this technique to triangular representation.

Theorem 2.1. *With the notation $w_i(T)$ of the Kronecker representation introduced in Definition 1.2, we have:*

$$w_i(T) = -\varphi(\partial_{U_i}(\mathcal{C}_V)(U_1,\ldots,U_n,T)).$$

Proof. We start with Identity (2.10), to which is applied the specialization φ to get:

$$\partial_{U_i}(\mathcal{C}_V)(u_1,\ldots,u_n,u_1X_1+\cdots+u_nX_n)+X_i\partial_T(\mathcal{C}_V)(u_1,\ldots,u_n,u_1X_1+\cdots+u_nX_n) \equiv 0 \bmod I(V).$$

From Equality (2.12), this is rewritten:

$$\partial_{U_i}(\mathcal{C}_V)(u_1,\ldots,u_n,u_1X_1+\cdots+u_nX_n) + X_i\cdot\chi'_u(u) \equiv 0 \bmod I(V).$$

The polynomial $W_i(T)$ of the Shape lemma representation (see Equation (1.2)), is the expression of X_i in the basis $\{1,u,u^2,\ldots,u^{\deg\chi_u-1}\}$. So $W_i(u)\equiv X_i \bmod I(V)$. Thanks to the isomorphism (1.1),which transforms u to T, this is rewritten:

$$\partial_{U_i}(\mathcal{C}_V)(u_1,\ldots,u_n,T) + W_i(T)\chi'_u(T) \equiv 0 \bmod \chi_u(T).$$

Definition 1.2 of the Kronecker representation implies that $w_i(T)\equiv -\partial_{U_i}(\mathcal{C}_V)(u_1,\ldots,u_n,T)$ modulo χ_u. Both polynomials have the same degree in T, so they are equal. □

This result creates a link between the Chow from of V and any primitive element representation of V. It makes it possible to obtain heights bounds (using the definitions 1.12 and 1.11 of height relying on Chow forms). We need to look at the behavior of the height of a polynomial through derivation and specialization. We make use of the notation $\log|f|$ of Equation (1.8). The following notations are useful:

$$\partial_T(\mathcal{C}_V)(u_1,\ldots,u_n,T) = \sum_{i=0}^{D}\left(a_iT^i\left(\sum_{\alpha\in\mathbb{N}^n}a_\alpha\mathbf{u}^\alpha\right)\right), \tag{2.13}$$

where $\mathbf{u}^\alpha = u_1^{\alpha_1}\ldots u_n^{\alpha_n}$. Then, from:

$$\log|\partial_T(\mathcal{C}_V)(U_1,\ldots,U_n,T)|_v = \log\max_{(i,\alpha)}\{|a_ia_\alpha|_v\}.$$

Let $\partial\in\{\partial_{U_1},\partial_{U_2},\ldots,\partial_{U_n},\partial_T\}$; since the degree of each monomial of $\mathcal{C}_V$ is at most D, and that all its derivatives have less monomials (not only because of the argument $\partial_X(Y)=0$, but also because the base field may be finite):

$$\max\{|\text{ coefficients of }\partial(\mathcal{C}_V)|_v\} \le \begin{cases}\max\{|\text{ coefficients of }\mathcal{C}_V|_v\}, & \text{if } K \text{ is a function field}\\ |D|_v\max\{|\text{ coefficients of }\mathcal{C}_V|_v\}, & \text{if } K \text{ is a number field.}\end{cases}$$

Since $|D|_v = 1$ if v is a non-Archimedean absolute value in the case where K is a number field, and $|D|_v = D$ if v is Archimedean, it follows:

$$\log|\partial(\mathcal{C}_V)(U_1,\dots,U_n,T)|_v \leq \begin{cases}\log(D)+\log|\mathcal{C}_V|_v & \text{if } v \text{ is Archimedean,} \quad (A)_\partial\\ \log|\mathcal{C}_V|_v & \text{if } v \text{ is non-Archimedean.} \quad (NA)_\partial\end{cases}$$

Now Equation (2.13) shows that:

$$\log|\partial_T(\mathcal{C}_V)(u_1,\dots,u_n,T)|_v = \log\max_i\{|a_i(\sum_\alpha a_\alpha \mathbf{u}^\alpha)|_v\}, \tag{2.14}$$

So, if v is non-Archimedean, we get:

$$\begin{aligned}
|a_i(\sum_\alpha a_\alpha\mathbf{u}^\alpha)|_v &= |a_i|_v|(\sum_\alpha a_\alpha\mathbf{u}^\alpha)|_v, \qquad \text{then by the ultrametric inequality,}\\
&\leq |a_i|_v \max_\alpha\{|a_\alpha\mathbf{u}^\alpha|_v\}\\
&\leq \max_{(i,\alpha)}\{|a_ia_\alpha|_v\}\max_\alpha\{|\mathbf{u}^\alpha|_v\}, \qquad \text{that implies,}\\
\log\max_i\{|a_i(\sum_\alpha a_\alpha\mathbf{u}^\alpha)|_v\} &\leq \log\max_{(i,\alpha)}\{|a_ia_\alpha|_v\}+\log\max_\alpha\{|\mathbf{u}^\alpha|_v\}, \qquad \text{and then}\\
\log|\partial_T\mathcal{C}_V(u_1,\dots,u_n,T)|_v &\leq \log|\partial_T\mathcal{C}_V(U_1,\dots,U_n,T)|_v+\log(\max\{|u_1|_v,\dots,|u_n|_v\}^D)\\
&\leq \log|\partial_T\mathcal{C}_V(U_1,\dots,U_n,T)|_v + D\log\max_i\{|u_i|_v\}\\
\text{after } (A)_{\partial_T} &\leq D\log\max_i\{|u_i|_v\}+\log|\mathcal{C}_V|_v\\
h_v(\partial_T\mathcal{C}_V(u_1,\dots,u_n,T)) &\leq h_v(V)+Dh_v(U).
\end{aligned} \tag{2.15}$$

And if v is Archimedean:

$$\begin{aligned}
|a_i(\textstyle\sum_\alpha a_\alpha\mathbf{u}^\alpha)| &\leq |a_i|_v|(\textstyle\sum_\alpha a_\alpha\mathbf{u}^\alpha)|_v\\
&\leq |a_i|_vD^n\max_\alpha\{|a_\alpha\mathbf{u}^\alpha|_v\}\\
&\leq D^n\max_{(i,\alpha)}\{|a_ia_\alpha|_v\}\max_\alpha\{|\mathbf{u}^\alpha|_v\}, \qquad \text{which implies}\\
\log\max_i\{|a_i(\textstyle\sum_\alpha a_\alpha\mathbf{u}^\alpha)|_v\} &\leq n\log D+\log\max_{(i,\alpha)}\{|a_ia_\alpha|_v\}+\log\max_\alpha\{|\mathbf{u}^\alpha|_v\}, \qquad \text{so,}\\
\log|\partial_T\mathcal{C}_V(u_1,\dots,u_n,T)|_v &\leq n\log D+\log|\partial_T\mathcal{C}_V(U_1,\dots,U_n,T)|_v+\\
&\quad +\log(\max\{|u_1|_v,\dots,|u_n|_v\}^D)\\
&\leq n\log D+\log|\partial_T\mathcal{C}_V(U_1,\dots,U_n,T)|_v+D\log\max_i\{|u_i|_v\}\\
\text{after } (NA)_{\partial_T} &\leq (n+1)\log D+D\log\max_i\{|u_i|_v\}+\log|\mathcal{C}_V|_v.
\end{aligned}$$

From inequality (1.13), we get, when v is Archimedean:

$$\log|\mathcal{C}_V|_v \leq m(\sigma_v(\mathcal{C}_V))+D\log(n+2)$$

and then, after (1.14) we get:

$$\log|\mathcal{C}_V|_v \leq m(\sigma_v(\mathcal{C}_V);S_{n+2})+D\left(\log(n+2)+\sum_{i=1}^{n+1}\frac{1}{2i}\right)$$

It follows:

$$\log|\partial_T(\mathcal{C}_V)(u_1,\ldots,u_n,T)|_v \le (n+1)\log D + m(\sigma_v(\mathcal{C}_V);S_{n+2}) + D\left(\log\max_i\{|u_i|_v\} + \log(n+2) + \sum_{i=1}^{n+1}\frac{1}{2i}\right).$$

And finally

$$h_v\big(\partial_T(\mathcal{C}_V)(u_1,\ldots,u_n,T)\big) \le h_v(V) + D\log(n+2) + (n+1)\log D + Dh_v(U). \qquad (2.16)$$

We deduce the bounds on the Kronecker representation in the theorem hereunder. As for the previous theorem, this bound is not new, but it stated in full generality, and is readable. Moreover, it will be useful in the sequel.

Theorem 2.2. *The height of the coefficients of $\chi'_u(T)$ and $w_i(T)$ of the primitive element representation of V is bounded by:*

$$\begin{array}{c} h(V) + Dh(U) + D\log(n+2) + (n+1)\log D \quad \textit{(number field case)} \\ h(V) + Dh(U) \quad \textit{(function field case).} \end{array}$$

Proof. From the definition of the height, we have

$$h(\chi'_u(T)) = \frac{1}{[K:K_0]}\sum_{v\in M_K^\infty} N_v h_v(\chi'_u(T)) + \frac{1}{[K:K_0]}\sum_{v\in M_K^0} N_v h_v(\chi'_u(T))$$

So we use Equations (2.16) and (2.15) to get the expected result. □

2.1.3 A link between Chow forms and triangular polynomials

We generalize the trick of differentiating the Chow form to get a primitive element representation, to triangular representations. We manage to get a family of polynomials $(M_1,\ldots,M_n)$ (see Definition 2.2) having the same solutions than the corresponding triangular set, and the possibility to convert this family to this triangular set (see Algorithm 2.1).

The bounds obtained here use fully the derivation formulas of Section 1. As for the primitive element representation, we need to link the Chow form of V (zero-dimensional variety, equiprojectable, vanishing ideal verifying Separability Assumption) and the polynomials of the triangular set describing V. Let us first see on an example how this link is coming across.

Introduction - Case $n = 2$ and $n = 3$

As the results are quite technical, this paragraph first introduces the problem and provides some examples. Suppose we have an equiprojectable variety $V \subset \mathbb{A}^3_{\bar{K}}$. As usual $\mathcal{C}_3 \in K[U_1,U_2,U_3,T]$, $\mathcal{C}_2 \in K[U_1,U_2,T]$ and $\mathcal{C}_1 \in K[U_1,T]$ will denote the Chow forms of V, $\pi_2^3(V)$ and $\pi_1^3(V)$ respectively. The following theorem shows how to reconstruct the polynomials T_1, T_2 and T_3 of the triangular set describing V.

Theorem 2.3. *Let ∂_i denote the derivation $\partial_{U_i}+X_i\partial_T$ for $i=2$ or 3. Define the polynomials M_1, M_2, and M_3 as follows:*

$$\left|\begin{array}{rcl} M_1(X_1) & = & \mathcal{C}_1(X_1) \\ M_2(X_1,X_2) & = & \partial_2^{d_2}(\mathcal{C}_2)(1,0,X_1) \\ M_3(X_1,X_2,X_3) & = & \partial_2^{(d_2-1)d_3}\partial_3^{d_3}(\mathcal{C}_3)(1,0,0,X_1). \end{array}\right.$$

Then (M_1,M_2,M_3) is a regular chain whose initials $h_2=\mathsf{init}(M_2)$ and $h_3=\mathsf{init}(M_3)$ verify:

$$\begin{array}{rcl} h_2 & \equiv & d_2!T_1'(X_1)^{d_2} \bmod (T_1), \\ h_3 & \equiv & d_3!(d_3(d_2-1))!\left(T_1'(X_1)^{d_2d_3}\cdot\partial_{X_2}(T_2)(X_1,X_2)^{d_3}\right) \bmod (T_1,T_2). \end{array}$$

Moreover, these initials are invertible modulo (T_1) and (T_1,T_2) respectively, and the following hold:

$$\begin{array}{rcl} T_1(X_1) & = & M_1(X_1) \\ T_2(X_1,X_2) & \equiv & (h_2^{-1} \bmod (T_1))\cdot M_2(X_1,X_2) \bmod (T_1), \\ T_3(X_1,X_2,X_3) & \equiv & (h_3^{-1} \bmod (T_1,T_2))\cdot M_3(X_1,X_2,X_3) \bmod (T_1,T_2). \end{array}$$

Example: Consider the family of 8 points of coordinates (i,j,k), i, j, $k=1$ or 2, forming an equiprojectable variety in $\mathbb{A}^3_{\mathbb{Q}}$. It is folklore to see that this variety is described by the triangular set:

$$T_1(X_1)=X_1^2-3X_1+2\,,\qquad T_2(X_1,X_2)=X_2^2-3X_2+2\,,\qquad T_3(X_1,X_2,X_3)=X_3^2-3X_3+2.$$

It is particularly simple since T_2 does not involve X_2 and T_3 does not involve X_2 neither X_1.

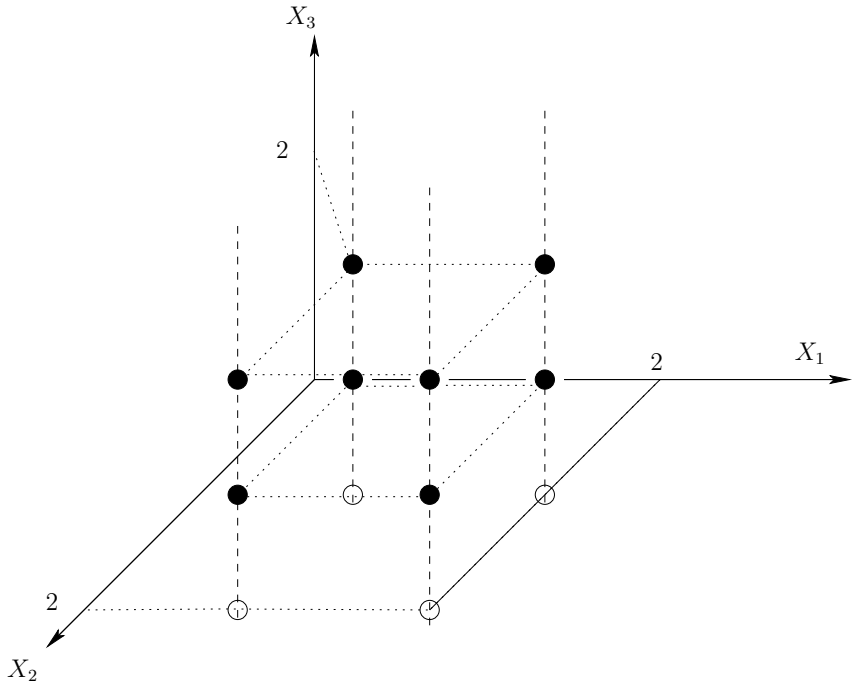

Figure 2.2: Example on an easy equiprojectable family of 8 points

Although, the Chow form $\mathcal{C}_3(U_1, U_2, U_3, T)$ is equal to:

$$
\begin{aligned}
&16\,U_1^8 + 144\,U_1^7\,U_2 + 144\,U_1^7\,U_3 - 96\,U_1^7\,T + 548\,U_1^6\,U_2^2 + 1116\,U_1^6\,U_2\,U_3 - 744\,U_1^6\,U_2\,T + 548\,U_1^6\,U_3^2 - 744\,U_1^6\,U_3\,T + 248\,U_1^6\,T^2 + 1152\,U_1^5\,U_2^3\\
&\quad + 3582\,U_1^5\,U_2^2\,U_3 - 2388\,U_1^5\,U_2^2\,T + 3582\,U_1^5\,U_2\,U_3^2 - 4860\,U_1^5\,U_2\,U_3\,T + 1620\,U_1^5\,U_2\,T^2 + 1152\,U_1^5\,U_3^3 - 2388\,U_1^5\,U_3^2\,T + 1620\,U_1^5\,U_3\,T^2\\
&\quad - 360\,U_1^5\,T^3 + 1464\,U_1^4\,U_2^4 + 6174\,U_1^4\,U_2^3\,U_3 - 4116\,U_1^4\,U_2^3\,T + 9424\,U_1^4\,U_2^2\,U_3^2 - 12783\,U_1^4\,U_2^2\,U_3\,T + 4261\,U_1^4\,U_2^2\,T^2 + 6174\,U_1^4\,U_2\,U_3^3\\
&\quad + 12783\,U_1^4\,U_2\,U_3^2\,T + 8667\,U_1^4\,U_2\,U_3\,T^2 - 1926\,U_1^4\,U_2\,T^3 + 1464\,U_1^4\,U_3^4 - 4116\,U_1^4\,U_3^3\,T + 4261\,U_1^4\,U_3^2\,T^2 - 1926\,U_1^4\,U_3\,T^3 + 321\,U_1^4\,T^4\\
&\quad + 1152\,U_1^3\,U_2^5 + 6174\,U_1^3\,U_2^4\,U_3 - 4116\,U_1^3\,U_2^4\,T + 12780\,U_1^3\,U_2^3\,U_3^2 - 17334\,U_1^3\,U_2^3\,U_3\,T + 5778\,U_1^3\,U_2^3\,T^2 + 12780\,U_1^3\,U_2^2\,U_3^3\\
&\quad - 26448\,U_1^3\,U_2^2\,U_3^2\,T + 17928\,U_1^3\,U_2^2\,U_3\,T^2 - 3984\,U_1^3\,U_2^2\,T^3 + 6174\,U_1^3\,U_2\,U_3^4 - 17334\,U_1^3\,U_2\,U_3^3\,T + 17928\,U_1^3\,U_2\,U_3^2\,T^2\\
&\quad - 8100\,U_1^3\,U_2\,U_3\,T^3 + 1350\,U_1^3\,U_2\,T^4 + 1152\,U_1^3\,U_3^5 - 4116\,U_1^3\,U_3^4\,T + 5778\,U_1^3\,U_3^3\,T^2 - 3984\,U_1^3\,U_3^2\,T^3\\
&\quad + 1350\,U_1^3\,U_3\,T^4 - 180\,U_1^3\,T^5 + 548\,U_1^2\,U_2^6 + 3582\,U_1^2\,U_2^5\,U_3 - 2388\,U_1^2\,U_2^5\,T + 9424\,U_1^2\,U_2^4\,U_3^2\\
&\quad - 12783\,U_1^2\,U_2^4\,U_3\,T + 4261\,U_1^2\,U_2^4\,T^2 + 12780\,U_1^2\,U_2^3\,U_3^3 - 26448\,U_1^2\,U_2^3\,U_3^2\,T + 17928\,U_1^2\,U_2^3\,U_3\,T^2\\
&\quad - 3984\,U_1^2\,U_2^3\,T^3 + 9424\,U_1^2\,U_2^2\,U_3^4 - 26448\,U_1^2\,U_2^2\,U_3^3\,T + 27347\,U_1^2\,U_2^2\,U_3^2\,T^2 - 12354\,U_1^2\,U_2^2\,U_3\,T^3 + 2059\,U_1^2\,U_2^2\,T^4\\
&\quad + 3582\,U_1^2\,U_2\,U_3^5 - 12783\,U_1^2\,U_2\,U_3^4\,T + 17928\,U_1^2\,U_2\,U_3^3\,T^2 - 12354\,U_1^2\,U_2\,U_3^2\,T^3 + 4185\,U_1^2\,U_2\,U_3\,T^4 - 558\,U_1^2\,U_2\,T^5\\
&\quad + 548\,U_1^2\,U_3^6 - 2388\,U_1^2\,U_3^5\,T + 4261\,U_1^2\,U_3^4\,T^2 - 3984\,U_1^2\,U_3^3\,T^3 + 2059\,U_1^2\,U_3^2\,T^4 - 558\,U_1^2\,U_3\,T^5 + 62\,U_1^2\,T^6 + 144\,U_1\,U_2^7 + 1116\,U_1\,U_2^6\,U_3\\
&\quad - 744\,U_1\,U_2^6\,T + 3582\,U_1\,U_2^5\,U_3^2 - 4860\,U_1\,U_2^5\,U_3\,T + 1620\,U_1\,U_2^5\,T^2 + 6174\,U_1\,U_2^4\,U_3^3 - 12783\,U_1\,U_2^4\,U_3^2\,T + 8667\,U_1\,U_2^4\,U_3\,T^2\\
&\quad - 1926\,U_1\,U_2^4\,T^3 + 6174\,U_1\,U_2^3\,U_3^4 - 17334\,U_1\,U_2^3\,U_3^3\,T + 17928\,U_1\,U_2^3\,U_3^2\,T^2 - 8100\,U_1\,U_2^3\,U_3\,T^3 + 1350\,U_1\,U_2^3\,T^4 + 3582\,U_1\,U_2^2\,U_3^5\\
&\quad - 12783\,U_1\,U_2^2\,U_3^4\,T + 17928\,U_1\,U_2^2\,U_3^3\,T^2 - 12354\,U_1\,U_2^2\,U_3^2\,T^3 + 4185\,U_1\,U_2^2\,U_3\,T^4 - 558\,U_1\,U_2^2\,T^5 + 1116\,U_1\,U_2\,U_3^6\\
&\quad - 4860\,U_1\,U_2\,U_3^5\,T + 8667\,U_1\,U_2\,U_3^4\,T^2 - 8100\,U_1\,U_2\,U_3^3\,T^3 + 4185\,U_1\,U_2\,U_3^2\,T^4 - 1134\,U_1\,U_2\,U_3\,T^5 + 126\,U_1\,U_2\,T^6 + 144\,U_1\,U_3^7\\
&\quad - 744\,U_1\,U_3^6\,T + 1620\,U_1\,U_3^5\,T^2 - 1926\,U_1\,U_3^4\,T^3 + 1350\,U_1\,U_3^3\,T^4 - 558\,U_1\,U_3^2\,T^5 + 126\,U_1\,U_3\,T^6 - 12\,U_1\,T^7\\
&\quad + 16\,U_2^8 + 144\,U_2^7\,U_3 - 96\,U_2^7\,T + 548\,U_2^6\,U_3^2 - 744\,U_2^6\,U_3\,T + 248\,U_2^6\,T^2 + 1152\,U_2^5\,U_3^3 - 2388\,U_2^5\,U_3^2\,T + 1620\,U_2^5\,U_3\,T^2 - 360\,U_2^5\,T^3\\
&\quad - 1464\,U_2^4\,U_3^4 - 4116\,U_2^4\,U_3^3\,T + 4261\,U_2^4\,U_3^2\,T^2 - 1926\,U_2^4\,U_3\,T^3 + 321\,U_2^4\,T^4 + 1152\,U_2^3\,U_3^5 - 4116\,U_2^3\,U_3^4\,T + 5778\,U_2^3\,U_3^3\,T^2 - 3984\,U_2^3\,U_3^2\,T^3\\
&\quad + 1350\,U_2^3\,U_3\,T^4 - 180\,U_2^3\,T^5 + 548\,U_2^2\,U_3^6 - 2388\,U_2^2\,U_3^5\,T + 4261\,U_2^2\,U_3^4\,T^2 - 3984\,U_2^2\,U_3^3\,T^3 + 2059\,U_2^2\,U_3^2\,T^4 - 558\,U_2^2\,U_3\,T^5\\
&\quad + 62\,U_2^2\,T^6 + 144\,U_2\,U_3^7 - 744\,U_2\,U_3^6\,T + 1620\,U_2\,U_3^5\,T^2 - 1926\,U_2\,U_3^4\,T^3 + 1350\,U_2\,U_3^3\,T^4 - 558\,U_2\,U_3^2\,T^5 + 126\,U_2\,U_3\,T^6\\
&\quad - 12\,U_2\,T^7 + 16\,U_3^8 - 96\,U_3^7\,T + 248\,U_3^6\,T^2 - 360\,U_3^5\,T^3 + 321\,U_3^4\,T^4 - 180\,U_3^3\,T^5 + 62\,U_3^2\,T^6 - 12\,U_3\,T^7 + T^8
\end{aligned}
$$

Lemma 1.3 shows that the Chow form $\mathcal{C}_2$ is equal to $\mathcal{C}_3(U_1, U_2, 0, T)^{1/2}$:

$$
\begin{aligned}
4U_1^4 + 18U_1^3U_2 - 12U_1^3T + 28U_1^2U_2^2 - 39U_1^2U_2T + 13U_1^2T^2 + 18U_1U_2^3 - 39U_1U_2^2T + 27U_1U_2T^2\\
- 6U_1T^3 + 4U_2^4 - 12U_2^3T + 13U_2^2T^2 - 6U_2T^3 + T^4
\end{aligned}
$$

and obviously $\mathcal{C}_1(U_1, T) = (T - U_1)\,(T - 2\,U_1)$, so that $T_1(X_1) = \mathcal{C}_1(1, X_1) = (X_1 - 1)(X_1 - 2)$. Following the formula of the theorem above, we compute:

$$\partial_2^{d_2}(\mathcal{C}_2)(1, 0, X_1) = 12X_1^2X_2^2 - 36X_1^2X_2 + 26X_1^2 - 36X_1X_2^2 + 108X_1X_2 - 78X_1 + 26X_2^2 - 78X_2 + 56.$$

Then an extended GCD computation gives $T_1'(X_1)^{-1} \bmod T_1 = 2X_1 - 3$, so that:

$$
\begin{aligned}
\frac{1}{d_2!}\left(T_1'(X_1)^{-d_2} \bmod T_1\right)\,\partial_2^{d_2}(\mathcal{C}_2)(1, 0, X_1) &= 4X_1^4X_2^3 - 72X_1^4X_2^2 + 52X_1^4X_2 - 144X_1^3X_2^3\\
&+ 432X_1^3X_2^2 - 312X_1^3X_2 + 322X_1^2X_2^3 - 966X_1^2X_2^2 + 697X_1^2X_2 - 318X_1X_2^3\\
&+ 954X_1X_2^2 - 687X_1X_2 + 117X_2^3 - 351X_2^2 + 252X_2
\end{aligned}
$$

It remains to reduce all the coefficients in X_1 (when the polynomial above is seen as a univariate polynomial in X_2):

$$
\begin{array}{l|r}
\text{degree } 0 & 52X_1^4 - 312X_1^3 + 697X_1^2 - 687X_1 + 252 \bmod T_1 = 2\\
\text{degree } 1,\ X_2 & -72X_1^4 + 432X_1^3 - 966X_1^2 + 954X_1 - 351 \bmod T_1 = -3\\
\text{degree } 2,\ X_2^2 & 24X_1^4 - 144X_1^3 + 322X_1^2 - 318X_1 + 117 \bmod T_1 = 1
\end{array}
$$

It finally gives the required polynomial $T_2(X_1, X_2) = X_2^2 - 3X_2 + 2$.

The computation for T_3 follows the same scheme.

$$\begin{aligned}\partial_2^2\partial_3^2(\mathcal{C}_3)(1,0,0,X_1) = {} & 420X_1^4X_2^2X_3^2 - 1260X_1^4X_2^2X_3 + 930X_1^4X_2^2 - 1260X_1^4X_2X_3^2 + 3780X_1^4X_2X_3 \\ & -2790X_1^4X_2 + 930X_1^4X_3^2 - 2790X_1^4X_3 + 2059X_1^4 - 2520X_1^3X_2^2X_3^2 + 7560X_1^3X_2^2X_3 \\ & -5580X_1^3X_2^2 + 7560X_1^3X_2X_3^2 - 22680X_1^3X_2X_3 + 16740X_1^3X_2 - -5580X_1^3X_3^2 + 16740X_1^3X_3 \\ & -12354X_1^3 + 5580X_1^2X_2^2X_3^2 - 16740X_1^2X_2^2X_3 + 12354X_1^2X_2^2 - 16740X_1^2X_2X_3^2 \\ & +50220X_1^2X_2X_3 - 37062X_1^2X_2 + 12354X_1^2X_3^2 - 37062X_1^2X_3 + 27347X_1^2 - 5400X_1X_2^2X_3^2 \\ & +16200X_1X_2^2X_3 - 11952X_1X_2^2 + 16200X_1X_2X_3^2 - 48600X_1X_2X_3 + 35856X_1X_2 \\ & -11952X_1X_3^2 + +35856X_1X_3 - 26448X_1 + 1926X_2^2X_3^2 - 5778X_2^2X_3 + 4261X_2^2 \\ & -5778X_2X_3^2 + 17334X_2X_3 - 12783X_2 + 4261X_3^2 - 12783X_3 + 9424\end{aligned}$$

An extended GCD computation gives $\partial_{X_2}(T_2)^{-1} \bmod (T_1, T_2) = 2X_2 - 3$, yielding to:

$$\frac{1}{d_3!(d_2-1)d_3!}T_1'(X_1)^{-d_2d_3}\partial_{X_2}(T_2)(X_1,X_2)^{-d_3} \equiv \frac{1}{4}(2X_1-3)^4(2X_2-3)^2 \bmod (T_1,T_2).$$

Hence, following the formula for T_3 of Theorem 2.3, before reduction modulo (T_1, T_2) we get a polynomial in X_3 whose coefficients in X_2 are:

$$\begin{aligned}\boxed{\text{degree } 0\text{: } 1}\ & 59520X_1^8X_2^4 - 357120X_1^8X_2^3 + 801376X_1^8X_2^2 - 797088X_1^8X_2 + 296496X_1^8 \\ & 714240X_1^7X_2^4 + 4285440X_1^7X_2^3 - 9616512X_1^7X_2^2 + 9565056X_1^7X_2 - 3557952X_1^7 \\ & +3736896X_1^6X_2^4 - 22421376X_1^6X_2^3 + 50313200X_1^6X_2^2 - 50043408X_1^6X_2 + 18614520X_1^6 \\ & -11133504X_1^5X_2^4 + 66801024X_1^5X_2^3 - 149898672X_1^5X_2^2 + 149091408X_1^5X_2 - 55455192X_1^5 \\ & +20658568X_1^4X_2^4 - 123951408X_1^4X_2^3 + 278136838X_1^4X_2^2 - 276629178X_1^4X_2 + 102887883X_1^4 \\ & -24444528X_1^3X_2^4 + 146667168X_1^3X_2^3 - 329101332X_1^3X_2^2 + 327301740X_1^3X_2 - 121725882X_1^3 \\ & +18010728X_1^2X_2^4 - 108064368X_1^2X_2^3 + 242474526X_1^2X_2^2 - 241133922X_1^2X_2 + 89671131X_1^2 \\ & -7553952X_1X_2^4 + 45323712X_1X_2^3 - 101693448X_1X_2^2 + 101123640X_1X_2 - 37600848X_1 \\ & +1380564X_2^4 - 8283384X_2^3 + 18584721X_2^2 - 18478935X_2 + 6870096,\end{aligned}$$

$$\begin{aligned}\boxed{\text{degree } 1:\ X_3}\ & -80640X_1^8X_2^4 + 483840X_1^8X_2^3 - 1085760X_1^8X_2^2 + 1080000X_1^8X_2 - 401760X_1^8 \\ & +967680X_1^7X_2^4 - 5806080X_1^7X_2^3 + 13029120X_1^7X_2^2 - 12960000X_1^7X_2 + 4821120X_1^7 \\ & -5063040X_1^6X_2^4 + 30378240X_1^6X_2^3 - 68169888X_1^6X_2^2 + 67807584X_1^6X_2 - 25224048X_1^6 \\ & +15085440X_1^5X_2^4 - 90512640X_1^5X_2^3 + 203111712X_1^5X_2^2 - 202028256X_1^5X_2 + 75151152X_1^5 \\ & -27994032X_1^4X_2^4 + 167964192X_1^4X_2^3 - 376908564X_1^4X_2^2 + 374886828X_1^4X_2 - 139445334X_1^4 \\ & +33128352X_1^3X_2^4 - 198770112X_1^3X_2^3 + 446027544X_1^3X_2^2 - 443617128X_1^3X_2 + 165000564X_1^3 \\ & -24412752X_1^2X_2^4 + 146476512X_1^2X_2^3 - 328675644X_1^2X_2^2 + 326882628X_1^2X_2 - 121572414X_1^2 \\ & +10240992X_1X_2^4 - 61445952X_1X_2^3 + 137873016X_1X_2^2 - 137112264X_1X_2 + 50989176X_1 \\ & -1872072X_2^4 + 11232432X_2^3 - 25202502X_2^2 + 25061562X_2 - 9318807,\end{aligned}$$

$$\boxed{\text{degree } 2:\ X_3^2}\ 26880X_1^8X_2^4 - 161280X_1^8X_2^3 + 361920X_1^8X_2^2 - 360000X_1^8X_2 + 133920X_1^8$$
$$-322560X_1^7X_2^4 + 1935360X_1^7X_2^3 - 4343040X_1^7X_2^2 + 4320000X_1^7X_2 - 1607040X_1^7$$
$$+1687680X_1^6X_2^4 - 10126080X_1^6X_2^3 + 22723296X_1^6X_2^2 - 22602528X_1^6X_2 + 8408016X_1^6$$
$$-5028480X_1^5X_2^4 + 30170880X_1^5X_2^3 - 67703904X_1^5X_2^2 + 67342752X_1^5X_2 - 25050384X_1^5$$
$$+9331344X_1^4X_2^4 - 55988064X_1^4X_2^3 + 125636188X_1^4X_2^2 - 124962276X_1^4X_2 + 46481778X_1^4$$
$$-11042784X_1^3X_2^4 + 66256704X_1^3X_2^3 - 148675848X_1^3X_2^2 + 147872376X_1^3X_2 - 55000188X_1^3$$
$$+8137584X_1^2X_2^4 - 48825504X_1^2X_2^3 + 109558548X_1^2X_2^2 - 108960876X_1^2X_2 + 40524138X_1^2$$
$$-3413664X_1X_2^4 + 20481984X_1X_2^3 - 45957672X_1X_2^2 + 45704088X_1X_2 - 16996392X_1$$
$$+624024X_2^4 - 3744144X_2^3 + 8400834X_2^2 - 8353854X_2 + 3106269.$$

To perform the reduction, first a reduction modulo T_1 of the coefficients in X_1 of monomials X_2^i, for the 3 polynomials in above, and then a reduction modulo $T_2(X_1, X_2)$ in the base ring $K[X_1]$ (which is possible, since T_2 is monic). Yielding to the expected result for T_3:

$$\begin{array}{l|l} \text{degree } 0,\ 1 & 59520X_1^8X_2^4 - \cdots \bmod (T_1, T_2) = 2 \\ \text{degree } 1,\ X_3 & -80640X_1^8X_2^4 + \cdots \bmod (T_1, T_2) = -3 \\ \text{degree } 2,\ X_3^2 & 26880X_1^8X_2^4 - \cdots \bmod (T_1, T_2) = 1 \end{array}$$

so that we get $T_3 = X_3^{@} - 3X_3 + 2$, as foreseen.

Staying in the case of three variables, let us prove the Theorem 2.3: how to get T_1, T_2, T_3 from the Chow form $\mathcal{C}_V$.

Calculation of T_1. As $T_1(X_1) = \mathcal{C}_1(1, X_1)$, and $\mathcal{C}_1(U_1, T)^{d_2 d_3} = \mathcal{C}_V(U_1, 0, 0, T)$ by Lemma 1.3, it is easy to get T_1 from the Chow form V.

Calculation of T_2. Set $F_2(U_1, U_2) = \mathcal{C}_2(U_1, U_2, U_1X_1 + U_2X_2)$. After Lemma 2.3

$$\forall i \in \mathbb{N},\ (\partial_{U_2}^i F_2)(U_1, U_2) = (\partial_2^i \mathcal{C}_2)(U_1, U_2, U_1X_1 + U_2X_2) \equiv 0 \bmod (T_1, T_2) \otimes K[U_1, U_2].$$

It is easy to check that $(\partial_{U_2}^i F_2)(1, 0)$ is a polynomial of degree i in X_2, with coefficients in $K[X_1]$. This polynomial also vanishes on $\pi_2^3(V)$ since it is null modulo (T_1, T_2), thanks to the previous equality. So we now look for which i its leading coefficient is invertible $\bmod(T_1)$. We have the equality:

$$\partial_2^{d_2}(\mathcal{C}_2)(1, 0, X_1) = \sum_{i=0}^{d_2} \partial_{U_2}^{n-i} \partial_T^i (\mathcal{C}_2)(1, 0, X_1), \tag{2.17}$$

so that the leading coefficient in X_2 is $h_2 = \partial_T^{d_2}(\mathcal{C}_2)(1, 0, X_1)$. Since the specialization $U_2 = 0$ commutes with the derivation ∂_T, the equality $\partial_T^{d_2}(\mathcal{C}_2)(U_1, 0, T) = \partial_T^{d_2}(\mathcal{C}_1^{d_2})(U_1, T)$ holds, thanks to Lemma 1.3. Proposition 2.1 then gives:

$$\partial_T^{d_2}(\mathcal{C}_1^{d_2})(X_1) = \sum_{\substack{\mathbf{j}=(j_1,\ldots,j_{d_2}) \\ |\mathbf{j}|=d_2}} \binom{d_2}{\mathbf{j}} \prod_{t=1}^{t=d_2} \partial_T^{j_t}(\mathcal{C}_1)(X_1).$$

But all the d_2-uples $\mathbf{j}$ for which there exists an index $t \in \{1, \dots, d_2\}$ such that $j_t = 0$ we have $\prod_{t=1}^{t=d_2} \partial_T^{j_t}(\mathcal{C}_1)(1, X_1) \equiv 0 \bmod (T_1)$, since $\mathcal{C}_1(1, X_1) \equiv 0 \bmod (T_1)$ the only d_2-uple $\mathbf{j}$ for which this does not happen is $(1, \dots, 1)$. Hence, the equality above is reduced to:

$$\begin{aligned}(\partial_T^{d_2}(\mathcal{C}_1)^{d_2})(1, X_1) &\equiv d_2!\,(\partial_T \mathcal{C}_1(1, X_1))^{d_2} \bmod (T_1)\\ &\equiv d_2!(T_1'(X_1))^{d_2} \bmod (T_1)\end{aligned}$$

Moreover T_1 is square-free so that $T_1'(X_1)$ is coprime with $T_1(X_1)$, i.e. $T_1'(X_1)$ is invertible $\bmod(T_1)$. It follows that $(h_2^{-1} \bmod (T_1)\big(\partial_2^{d_2}(\mathcal{C}_2)(1, 0, X_1)\big) \bmod (T_1)$ is null modulo (T_1, T_2) (it vanishes on $\pi_2^3(V)$), is monic of degree d_2 in X_2 and reduced modulo (T_1). It follows that it is equal to T_2.

Calculation of T_3 Set $F_3(U_1, U_2, U_3) = \mathcal{C}_3(U_1, U_2, U_3, U_1X_1 + U_2X_2 + U_3X_3)$. Lemma 2.3 says that the following family of polynomials:

$$\left\{\partial_2^s \partial_3^\ell(\mathcal{C}_3)(1, 0, 0, X_1)\right\}_{s,\ell \in \mathbb{N}} \Longleftrightarrow \left\{\partial_{U_2}^s \partial_{U_3}^\ell(F_2)(1, 0, 0, X_1)\right\}_{s,\ell \in \mathbb{N}}$$

vanish on $\pi_2^3(V)$. Let us show that when $s = (d_2 - 1)d_3$ and $\ell = d_3$, the leading coefficient in X_3 is invertible $\bmod(T_1, T_2)$.

$$\partial_2^{(d_2-1)d_3} \partial_3^{d_3}(\mathcal{C}_3)(1, 0, 0, X_1) = \sum_{i=0}^{d_3} \partial_2^{(d_2-1)d_3} \partial_{U_3}^{d_3-i} \partial_T^i(\mathcal{C}_3)(1, 0, 0, X_1) \cdot X_3^i \tag{2.18}$$

Thus the leading coefficient is $\partial_2^{(d_2-1)d_3} \partial_T^{d_3}(\mathcal{C}_3)(1, 0, 0, X_1)$. Since ∂_T commutes with the specialization $U_3 = 0$, we have from Lemma 1.3:

$$\partial_T^{d_3}(\mathcal{C}_3)(U_1, U_2, 0, T) = \partial_T^{d_3}(\mathcal{C}_2^{d_3})(U_1, U_2, T).$$

Thanks to the generalized Leibniz formula (Proposition 2.1):

$$\partial_T^{d_3}(\mathcal{C}_2^{d_3})(U_1, U_2, T) = \sum_{\substack{\mathbf{j}=(j_1,\dots,j_{d_3}) \\ |\mathbf{j}|=d_3}} \binom{d_3}{\mathbf{j}} \prod_{t=1}^{t=d_3} \partial_T^{j_t}(\mathcal{C}_2)(U_1, U_2, T).$$

But $\mathcal{C}_2(U_1, U_2, T)$ divides all the products above as soon as at least one index j_t is non zero. The only d_3-uple $\mathbf{j}$ for which this does not happen is $(1, 1, \dots, 1)$. Hence, there exists a polynomial $A \in K[U_1, U_2, T]$, such that:

$$\partial_T^{d_3}(\mathcal{C}_2^{d_3})(U_1, U_2, T) = A \cdot \mathcal{C}_2(U_1, U_2, T) + d_3!(\partial_T(\mathcal{C}_2)(U_1, U_2, T))^{d_3}.$$

From Lemma 2.3 applied with $i = 2$, it follows:

$$\partial_T^{(d_2-1)d_3}(A \cdot \mathcal{C}_2)(U_1, U_2, U_1X_1 + U_2X_2) \equiv 0 \bmod (T_1, T_2) \otimes K[U_1, U_2].$$

Now we focus on the other term $d_3!(\partial_T(\mathcal{C}_2)^{d_3})$.

$$\partial_2^{(d_2-1)d_3} d_3!(\partial_T(\mathcal{C}_2)(U_1, U_2, T))^{d_3} = d_3! \sum_{\substack{\mathbf{j}=(j_1,\dots,j_{d_3}) \\ |\mathbf{j}|=(d_2-1)d_3}} \binom{(d_2-1)d_3}{\mathbf{j}} \prod_{t=1}^{t=d_3} \partial_2^{j_t} \partial_T(\mathcal{C}_2)(U_1, U_2, T).$$

Then Proposition 2.3 shows that for all $\mathbf{j}$ such that there exists $t \in \{1, \ldots, d_3\}$ with $j_t \leq d_2 - 2$, the product above where appears such a $\mathbf{j}$ is null after specialization $U_1, U_2, T \to 1, 0, X_1$ and reduction modulo (T_1, T_2). So, the only d_3-uple leading to a non null product, and respecting $|\mathbf{j}| = (d_2 - 1)d_3$ is $(d_2 - 1, \ldots d_2 - 1)$:

$$d_3!\Big(\partial_2^{(d_2-1)d_3}\big(\partial_T(\mathcal{C}_2)(1,0,X_1)\big)^{d_3}\Big) \equiv d_3!\frac{((d_2-1)d_3)!}{(d_2-1)!^{d_3}}\Big(\partial_2^{d_2-1}\partial_T(\mathcal{C}_2)(1,0,X_1)\Big)^{d_3} \bmod (T_1,T_2),$$

But from Lemma 2.1

$$\partial_2^{d_2-1}\partial_T(\mathcal{C}_2) = \frac{1}{d_2}\partial_{X_2}\partial_2^{d_2}(\mathcal{C}_2)$$

and as $\partial_2^{d_2}(\mathcal{C}_2)(1,0,X_1) \equiv d_2!(T'(X_1))^{d_2}T_2(X_1,X_2) \bmod (T_1,T_2)$ from the previous paragraph "Calculation of T_2", we get:

$$\partial_2^{d_2-1}\partial_T(\mathcal{C}_2)(1,0,X_1) \equiv (d_2-1)!(T_1'(X_1))^{d_2}\partial_{X_2}(T_2)(X_1,X_2) \bmod (T_1,T_2),$$

Finally, the leading coefficient, that we have denoted h_3 in the theorem, is finally equal to $d_3!(d_3(d_2-1))!(T_1'(X_1))^{d_2d_3}(\partial_{X_2}(T_2)(X_1,X_2))^{d_3}$ modulo (T_1,T_2). Let us see this polynomial in $\big(K[X_1]/(T_1)\big)[X_2]$. We have seen in the previous paragraph that $T_1'(X_1)$ is a unit in that ring. Moreover $T_2(X_1,X_2) \bmod (T_1)$ has no common root with $\partial_{X_2}(T_2) \bmod (T_1)$, else T_1, T_2 would not generate a radical ideal. So $\partial_{X_2}(T_2) \bmod (T_1)$ is invertible modulo (T_1,T_2). It follows that h_3 is also invertible modulo (T_1,T_2) as product of invertible elements. Finally, let $P = \big(h_3^{-1} \bmod (T_1,T_2)\big)\partial_2^{(d_2-1)d_3}\partial_3^{d_3}(\mathcal{C}_3)(1,0,0,X_1)$. Then P is of degree d_3 in X_3 (it is Equation (2.18)), and vanishes on V since the second factor of P does. If we add the degree constraints that verify T_3, that is $\deg_{X_1}(T_3) < d_1$ and $\deg_{X_2}(T_2) < d_2$, then we conclude that $P \bmod (T_1,T_2) = T_3$. □

Before proving the general case, let us sketch an algorithm for computing a triangular set for V from the data of its Chow form, through the polynomials $M_1, \ldots, M_n$ (Cf. Definition 2.2 hereafter). It should not be seen with an algorithmic viewpoint: we have the aim at analyzing the size of the coefficients of the output, hence coefficients swell at each step of this algorithm is analyzed. That is only why it is used. Hereunder, `NormalForm`(x, G) outputs the normal form of the polynomial x with respect to the Gröbner basis G; `ModInv`(y, G) outputs the inverse of y, in normal form, modulo the Gröbner basis G, when it exists.

General case

By following the main steps of the calculations of T_2 and T_3 above, we want to compute the analogous formula for the polynomial T_{s+1} (with $s < n$) of the triangular set. It is equivalent to show that Algorithm 2.1 gives the correct output.

Lemma 2.3 shows that the family of polynomials

$$\{\partial_2^{n_2}\cdots\partial_{s+1}^{n_{s+1}}(\mathcal{C}_{s+1})(1,0,\ldots,0,X_1)\}_{(n_2,\ldots,n_{s+1})\in\mathbb{N}^s}$$

vanish on $\pi_{s+1}^n(V)$. Now we aim at showing that for the family of indices

$$n_2 = (d_2-1)d_3\cdots d_{s+1} \quad , \quad n_3 = (d_3-1)d_4\cdots d_{s+1} \quad , \quad \ldots \quad , \quad n_{s+1} = d_{s+1}-1,$$

the leading coefficients in X_{s+1} of $\partial_2^{n_2}\cdots\partial_{s+1}^{n_{s+1}}(\mathcal{C}_{s+1})(1,0,\ldots,0,X_1)$ is invertible $\bmod (T_1,\ldots,T_s)$. To prove this, let us introduce the following notation and definition:

$$\mathbf{d}_{(i,j)} := (d_i-1)d_{i+1}\cdots d_j, \qquad \text{and} \qquad \partial = \partial_2^{\mathbf{d}_{(2,s+1)}}\partial_3^{\mathbf{d}_{(3,s+1)}}\cdots\partial_s^{\mathbf{d}_{(s,s+1)}}.$$

`TurnMintoT`$(M_1, \ldots, M_n)$

#`Inputs`:Family of polynomials $(M_1, \ldots, M_n)$ as defined in Introduction.
`Output`: A triangular set $(T_1, \ldots, T_n)$, $\deg_{X_i}(T_i) = d_i$.

1. if $(n = 1)$ then return $(T_1) = (M_1)$.
2. let $(T_1, \ldots, T_{n-1}) =$ `TurnMintoT`$(M_1, \ldots, M_{n-1})$.
3. write $M_n = \sum_{i=0}^{d_n} m_{n,i} \cdot X_n^i$.
4. for $i = 0$ to d_n do

   $\widetilde{m_{n,i}} :=$ `NormalForm`$(m_{n,i}, (T_1, \ldots, T_{n-1}))$.

   end for
5. write $\widetilde{M_n} := \sum_{i=0}^{d_n} \widetilde{m_{n,i}} X_n^i$.
6. $h_n :=$ init$(\widetilde{M_n})$. Compute $H_n :=$ `ModInv`$(h_n, (T_1, \ldots, T_{n-1}))$.
7. return $\Big(T_1, \ldots, T_{n-1}\,,$ `NormalForm`$\big(H_n \cdot \widetilde{M_n}, (T_1, \ldots, T_{n-1})\big)\Big)$.

Algo **2.1:** How to get recursively the polynomials T_i from the polynomials M_i

Definition 2.2. *Let V be an equiprojectable variety defined by a triangular set $T_1, \ldots, T_n$, with $\mathcal{C}_V$ for Chow form. Denote $\mathcal{C}_i$ instead of $\mathcal{C}_{\pi_i^n(V)}$, the Chow form of the projection $\pi_i^n(V)$. Let $M_1(X_1) = T_1(X_1)$ and for $1 \leq s \leq n-1$, we define the polynomial $M_{s+1} \in K[X_1, \ldots, X_{s+1}]$ in the following way:*

$$M_{s+1}(X_1, \ldots, X_{s+1}) = \partial \partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})(1, 0, \ldots, 0, X_1).$$

Since ∂, $\partial_{U_{s+1}}$ and ∂_T commute, we have:

$$\partial_{s+1}^{d_{s+1}} = (\partial_{U_{s+1}} + X_{s+1}\partial_T)^{d_{s+1}} = \sum_{i=0}^{d_{s+1}} \binom{d_{s+1}}{i} X_{s+1}^i \cdot \partial_T^i \partial_{U_{s+1}}^{d_{s+1}-i}.$$

The coefficient of X_{s+1}^i in $K[X_1, \ldots, X_s]$ of $\partial \partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})$, is:

$$\binom{d_{s+1}}{i} \partial \partial_{U_{s+1}}^{d_{s+1}-i} \partial_T^i (\mathcal{C}_{s+1})(1, 0, \ldots, 0, X_1), \tag{2.19}$$

hence, the leading coefficient we are interested in is:

$$\partial \partial_T^{d_{s+1}}(\mathcal{C}_{s+1})(1, 0, \ldots, 0, X_1).$$

All the derivations $\{\partial_i\}_i$ and ∂_T commute with the specialization $U_{s+1} = 0$; consequently, all the calculations can be conducted modulo this specialization. Moreover, since from Lemma 1.3

$$\mathcal{C}_{s+1}(U_1, \ldots, U_s, 0, T) = \mathcal{C}_s(U_1, \ldots, U_s, T)^{d_{s+1}}$$

we get:

$$\partial_T^{d_{s+1}}(\mathcal{C}_{s+1})(U_1,\dots,U_s,0,T)=\partial_T^{d_{s+1}}(\mathcal{C}_s^{d_{s+1}})(U_1,\dots,U_s,T).$$

The generalized Leibniz formula (Proposition 2.1) applied to the equation above gives:

$$\begin{aligned}\partial_T^{d_{s+1}}(\mathcal{C}_s^{d_{s+1}}) &= \sum_{\substack{\mathbf{j}=(j_1,\dots,j_{d_{s+1}})\\ |\mathbf{j}|=d_{s+1}}} \binom{d_{s+1}}{\mathbf{j}} \partial_T^{j_1}(\mathcal{C}_s)\cdots\partial_T^{j_{d_{s+1}}}(\mathcal{C}_s)\\ &= d_{s+1}!\partial_T(\mathcal{C}_s)^{d_{s+1}} + A\cdot\mathcal{C}_s\,, \quad \text{for a polynomial } A\in K[U_1,\dots,U_s,T].\end{aligned}$$

The first term corresponds to the d_{s+1}-uple $(1,\dots,1)$, and the second term corresponds the the other uples. In fact one of these uples necessary contains a zero, making appear a factor $\partial_T^0(\mathcal{C}_s)=\mathcal{C}_s$. So $\mathcal{C}_s$ divides the product indexed by such a d_{s+1}-ulpe $\mathbf{j}$, that is why there is a polynomial A here.

Again, an application of Lemma 2.3 with $i=s$ and the derivation ∂, and applied to $A\cdot\mathcal{C}_s$ leads to:

$$\partial(A\cdot\mathcal{C}_s)(U_1,\dots,U_s,\sum_{1\le k\le s}U_kX_k)\equiv 0 \bmod (T_1,\dots,T_s)\otimes K[U_1,\dots,U_s]. \tag{2.20}$$

We can therefore only pay attention to the term $d_{s+1}!(\partial_T(\mathcal{C}_s))^{d_{s+1}}$, to which we apply Corollary 2.1:

$$d_{s+1}!\partial(\partial_T(\mathcal{C}_s))^{d_{s+1}} = d_{s+1}!\sum_{j_t^{(\alpha)}}\binom{\mathbf{d}_{(2,s+1)}}{\mathbf{j}^{(2)}}\cdots\binom{\mathbf{d}_{(s,s+1)}}{\mathbf{j}^{(s)}}\prod_{t=1}^{t=d_{s+1}}\partial_2^{j_t^{(2)}}\partial_3^{j_t^{(3)}}\cdots\partial_s^{j_t^{(s)}}\partial_T(\mathcal{C}_s) \tag{2.21}$$

$$\begin{cases}\mathbf{j}^{(\alpha)} &= (j_1^{(\alpha)},\dots,j_{d_{s+1}}^{(\alpha)}) \quad,\ \alpha=2,\dots,s,\\ |\mathbf{j}^{(\alpha)}| &= (d_\alpha-1)d_{\alpha+1}\cdots d_{s+1}\end{cases}$$

To eliminate the d_{s+1}-uples $\mathbf{j}^{(\alpha)}$ which actually will cancel the products above after specialization and reduction, hence useless uples, the following proposition is required.

Proposition 2.4. *Let $2\le i\le s$ be two integers, and denote by $\mathcal{G}_i$ the following set:*

$$\mathcal{G}_i=\{(\mathbf{j}^{(i)},\dots,\mathbf{j}^{(s)})\in(\mathbb{N}^{d_{s+1}})^{s-i+1} \text{ such that there exists } t \text{ verifying } \sum_{\alpha=i}^{s}j_t^{(\alpha)}+1\neq d_i\cdots d_s\}.$$

If $(\mathbf{j}^{(i)},\dots,\mathbf{j}^{(s)})\in\mathcal{G}_i$ then for all $(\mathbf{j}^{(2)},\dots,\mathbf{j}^{(i-1)})\in(\mathbb{N}^{d_{s+1}})^{i-2}$, we have:

$$\prod_{t=1}^{d_{s+1}}\partial_2^{j_t^{(2)}}\partial_3^{j_t^{(3)}}\dots\partial_s^{j_t^{(s)}}\partial_T(\mathcal{C}_s)(U_1,\dots,U_{i-1},0,\dots,0,U_1X_1+\cdots+U_{i-1}X_{i-1})\equiv 0$$

modulo $(T_1,\dots,T_{i-1})\otimes K[U_1,\dots,U_{i-1}]$.

Proof. Suppose first that $(\mathbf{j}^{(i)},\dots,\mathbf{j}^{(s)})\in\mathcal{G}_i$ is such that there exists $t\in\{d_1,\dots,d_{s+1}\}$ with $\sum_{\alpha=i}^{s}j_t^{(\alpha)}+1\le d_i\cdots d_s-1$. Then Proposition 2.3 applied with $\ell=i$ shows that $\mathcal{C}_{i-1}$

divides $\partial_2^{j_t^{(2)}} \cdots \partial_s^{j_t^{(s)}} \partial_T(\mathcal{C}_s)$ specialized at $U_i = \cdots = U_s = 0$. Hence there exists a polynomial $A \in K[U_1, \ldots, U_{i-1}, T]$ such that:

$$\partial_2^{j_t^{(2)}} \cdots \partial_s^{j_t^{(s)}} \partial_T(\mathcal{C}_s)(U_1, \ldots, U_{i-1}, 0, \ldots, 0, T) = A \cdot \mathcal{C}_{i-1}.$$

By Lemma 2.3 used with the derivation $\partial_0 = \partial_2^{j_t^{(2)}} \cdots \partial_{i-1}^{j_t^{(i-1)}}$ and applied to $A \cdot \mathcal{C}_{i-1}$, we have:

$$\partial_0(A \cdot \mathcal{C}_{i-1})(U_1, \ldots, U_{i-1}, \sum_{1 \le k \le i-1} U_k X_k) \equiv 0 \bmod (T_1, \ldots, T_{i-1}) \otimes K[U_1, \ldots, U_{i-1}].$$

It follows that the product on which the statement holds has a null factor, hence is itself null.

We restrict therefore our interest to the $(\mathbf{j}^{(i)}, \ldots, \mathbf{j}^{(s)})$ such that for all t, holds the inequality: $\sum_{\alpha=i}^{s} j_t^{(\alpha)} + 1 \ge d_i d_{i+1} \cdots d_s$. But as $\sum_{t=1}^{d_{s+1}} \sum_{\alpha=i}^{s} j_t^{(\alpha)} + 1 = \sum_{\alpha=i}^{s} |\mathbf{j}^{(\alpha)}| + d_{s+1} = d_i \cdots d_{s+1}$:

$$\forall t\,, \quad \sum_{\alpha=i}^{s} j_t^{(\alpha)} + 1 = d_i \cdots d_s.$$

But then $(\mathbf{j}^{(i)}, \ldots, \mathbf{j}^{(s)}) \notin \mathcal{G}_i$. Hence all the uples in $\mathcal{G}_i$ verify the cancellation identity. □

Corollary 2.2. *Suppose that there exists $t \in \{1, \ldots, d_{s+1}\}$ such that one of the following two conditions holds:*

(i) $j_t^{(s)} \neq d_s - 1$.

(ii) $j_t^{(i)} \neq (d_i - 1) d_{i+1} \cdots d_s$ *for at least one* $i \in \{2, \ldots, s-1\}$,

Then there is the cancellation identity:

$$\prod_{t=1}^{d_{s+1}} \partial_2^{j_t^{(2)}} \partial_3^{j_t^{(3)}} \cdots \partial_s^{j_t^{(s)}} \partial_T(\mathcal{C}_s)(U_1, 0, \ldots, 0, U_1 X_1) \equiv 0 \bmod (T_1, \ldots, T_{s-1}) \otimes K[U_1].$$

Proof. In case *(i)*, the d_{s+1}-uple $\mathbf{j}^{(s)} \in \mathcal{G}_s$. The previous proposition implies that:

$$\prod_{t=1}^{d_{s+1}} \partial_2^{j_t^{(2)}} \partial_3^{j_t^{(3)}} \cdots \partial_s^{j_t^{(s)}} \partial_T(\mathcal{C}_s)(U_1, \ldots, U_{s-1}, 0, , U_1 X_1 + \cdots + U_{s-1} X_{s-1}) \equiv 0$$

modulo $(T_1, \ldots, T_{s-1}) \otimes K[U_1, \ldots, U_{s-1}]$. Specializing $U_2 = \cdots = U_{s-1} = 0$ gives the requested result.

In case *(ii)*, if $(\mathbf{j}^{(i+1)}, \ldots, \mathbf{j}^{(s)}) \in \mathcal{G}_{i+1}$, then we are done by applying the proposition above, and taking the specialization $U_2 = \cdots = U_i = 0$. Else, $(\mathbf{j}^{(i+1)}, \ldots, \mathbf{j}^{(s)}) \notin \mathcal{G}_{i+1}$ and for all t, $\sum_{\alpha=i+1}^{s} j_t^{(\alpha)} + 1 = d_{i+1} \cdots d_s$. If moreover $(\mathbf{j}^{(i)}, \ldots, \mathbf{j}^{(s)}) \notin \mathcal{G}_i$, then $\forall t$, $\sum_{\alpha=i}^{s} j_t^{(\alpha)} + 1 = d_i \cdots d_s$. Whereas $j_t^{(i)} = \left(\sum_{\alpha=i}^{s} j_t^{(\alpha)} + 1 \right) - \left(\sum_{\alpha=i+1}^{s} j_t^{(\alpha)} + 1 \right) = (d_i - 1) d_{i+1} \cdots d_s$, which is false by hypothesis; thus $(\mathbf{j}^{(i)}, \ldots, \mathbf{j}^{(s)}) \in \mathcal{G}_i$ and after the previous proposition, this leads to:

$$\prod_{t=1}^{d_{s+1}} \left(\partial_2^{j_t^{(2)}} \partial_3^{j_t^{(3)}} \cdots \partial_s^{j_t^{(s)}} \partial_T \mathcal{C}_s \right) (U_1, \ldots, U_{i-1}, 0 \ldots, 0, U_1 X_1 + \cdots + U_{i-1} X_{i-1}) \equiv 0$$

modulo $(T_1,\dots,T_{i-1})\otimes K[U_1,\dots,U_{i-1}]$. It is also null modulo the ideal $(T_1,\dots,T_{s-1})\otimes K[U_1,\dots,U_{i-1}]$. Since $i\geq 2$, the corollary is obtained by specializing $U_2=\cdots=U_{i-1}=0$. □

We want to determine h_{s+1} for all s by proving the following theorem:

Theorem 2.4. *The leading coefficient $h_{s+1}\in K[X_1,\dots,X_s]$ of $\partial\partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})(1,0,\dots,0,X_1)$ verifies:*

(i) $h_{s+1}\equiv \dfrac{d_{s+1}!\prod_{i=2}^{s}\mathbf{d}_{(i,s+1)}!}{\left((d_s-1)!\prod_{i=2}^{s-1}\mathbf{d}_{(i,s)}!\right)^{d_{s+1}}}\left(\frac{1}{d_s}\cdot h_s\cdot\partial_{X_s}(T_s)\right)^{d_{s+1}}$ *modulo* $(T_1,\dots,T_s)$,

(ii) h_{s+1} *is invertible modulo* $(T_1,\dots,T_s)$,

(iii) $h_{s+1}\equiv d_{s+1}!\prod_{i=2}^{s}\mathbf{d}_{(i,s+1)}\Big(T_1'(X_1)^{d_2\cdots d_{s+1}}\partial_{X_2}(T_2)(X_1,X_2)^{d_3\cdots d_{s+1}}\cdots$
$\cdots\partial_{X_{s-1}}(T_{s-1})(X_1,\dots,X_{s-1})^{d_sd_{s+1}}\partial_{X_s}(T_s)(X_1,\dots,X_{s-1},X_s)^{d_{s+1}}\Big)$ mod $(T_1,\dots,T_s)$.

As a consequence, the following formula for T_{s+1} holds:

$$T_{s+1}=\big(h_{s+1}^{-1}\bmod(T_1,\dots,T_s)\big)\partial\partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})(1,0,\dots,0,X_1)\bmod(T_1,\dots,T_s).$$

Proof. The proof is the continuation of the reasoning begun at this paragraph "General case". Proposition 2.4 and Corollary 2.2 were required to discard d_{s+1}-uples $\mathbf{j}^{(\alpha)}$ in Equation (2.21). There, an application of Lemma 2.3 had led to compute modulo $(T_1,\dots,T_s)$ in Equation (2.20), and hence it is necessary to carry on those modular calculations.

1st step: Corollary 2.2 applied to Equation (2.21) gives, modulo $(T_1,\dots,T_{s-1})\otimes K[U_1]$:

$$\begin{aligned}d_{s+1}!(\partial(\partial_T(\mathcal{C}_s))^{d_{s+1}})(U_1,0,\dots,0,U_1X_1)&\equiv\ \mathrm{c}\prod_{t=1}^{d_{s+1}}(\partial_2^{(d_2-1)d_3\cdots d_s}\cdots\partial_s^{d_s-1}\partial_T(\mathcal{C}_s))(U_1,0,\dots,0,U_1X_1)\\&\equiv\ \mathrm{c}\left(\partial_2^{(d_2-1)d_3\cdots d_s}\cdots\partial_s^{d_s-1}\partial_T(\mathcal{C}_s)\right)^{d_{s+1}}(U_1,0,\dots,0,U_1X_1)\end{aligned}$$

where,

$$\begin{aligned}\mathrm{c}\ &=d_{s+1}!\binom{\mathbf{d}_{(s,s+1)}}{\mathbf{j}^{(2)}}\binom{\mathbf{d}_{(3,s+1)}}{\mathbf{j}^{(3)}}\cdots\binom{\mathbf{d}_{(s,s+1)}}{\mathbf{j}^{(s)}}\\&=d_{s+1}!\left(\frac{((d_2-1)d_3\cdots d_{s+1})!}{((d_2-1)d_3\cdots d_s)!^{d_{s+1}}}\right)\left(\frac{((d_3-1)d_4\cdots d_{s+1})!}{((d_3-1)d_4\cdots d_s)!^{d_{s+1}}}\right)\cdots\left(\frac{((d_s-1)d_{s+1})!}{(d_s-1)!^{d_{s+1}}}\right)\\&=d_{s+1}!\frac{\mathbf{d}_{(2,s+1)}!}{\mathbf{d}_{(2,s)}!^{d_{s+1}}}\frac{\mathbf{d}_{(3,s+1)}!}{\mathbf{d}_{(3,s)}!^{d_{s+1}}}\cdots\frac{\mathbf{d}_{(s,s+1)}!}{(d_s-1)!^{d_{s+1}}}.\end{aligned}$$

After Lemma 2.1, $\partial_s^{d_s-1}\partial_T(\mathcal{C}_s)=\frac{1}{d_s}\partial_{X_s}\partial_s^{d_s}(\mathcal{C}_s)$. Moreover, ∂_{X_s} commutes with $\partial_2,\partial_3,\dots,\partial_{s-1}$, so that

$$\partial_2^{(d_2-1)d_3\cdots d_s}\cdots\partial_s^{d_s-1}\partial_T(\mathcal{C}_s)=\frac{1}{d_s}\partial_{X_s}\partial_2^{(d_2-1)d_3\cdots d_s}\cdots\partial_s^{d_s}(\mathcal{C}_s).$$

Finally, modulo the ideal $(T_1, \ldots, T_s)$, we have:

$$\partial\partial_T^{d_{s+1}}(\mathcal{C}_{s+1})(1,0,\ldots,0,X_1) \equiv \frac{d_{s+1}! \prod_{i=2}^{s} \mathbf{d}_{(i,s+1)}!}{\left((d_s-1)! \prod_{i=2}^{s-1} \mathbf{d}_{(i,s)}!\right)^{d_{s+1}}} \left(\frac{1}{d_s}\partial_{X_s}\partial_2^{\mathbf{d}_{(2,s)}} \cdots \partial_s^{d_s}(\mathcal{C}_s)(1,0,..,0,X_1)\right)^{d_{s+1}}.$$

2nd step : By induction on s. The previous paragraph gives h_2 and h_3. Suppose that the formula is true for h_s and let us prove it for h_{s+1}. All the equalities hereunder are true modulo $(T_1, \ldots, T_s)$. From the last equation in *Step 1*, we have Equality *(i)*:

$$h_{s+1} \equiv \frac{d_{s+1}! \prod_{i=2}^{s} \mathbf{d}_{(i,s+1)}!}{\left((d_s-1)! \prod_{i=2}^{s-1} \mathbf{d}_{(i,s)}!\right)^{d_{s+1}}} \left(\frac{1}{d_s} \cdot h_s \cdot \partial_{X_s}(T_s)\right)^{d_{s+1}}, \tag{2.22}$$

By induction hypothesis, we get:

$$\begin{aligned} h_s &= d_s! \left(\prod_{i=2}^{s-1}((d_i-1)d_{i+1}\cdots d_s)!\right)\left(\prod_{i=1}^{s-1}\partial_{X_i}(T_i)^{d_{i+1}\cdots d_s}\right), \\ &= d_s! \left(\prod_{i=2}^{s-1}\mathbf{d}_{(i,s)}!\right)\left(\prod_{i=1}^{s-1}\partial_{X_i}(T_i)^{d_{i+1}\cdots d_s}\right). \end{aligned}$$

We compute h_{s+1} by replacing h_s by the formula above.

$$\begin{aligned} \left(\frac{1}{d_s}h_s\partial_{X_s}(T_s)\right)^{d_{s+1}} &= \left((d_s-1)!.\prod_{i=2}^{s-1}\mathbf{d}_{(i,s)}! \cdot \prod_{i=1}^{s-1}\partial_{X_i}(T_i)^{d_{i+1}\cdots d_s} \cdot \partial_{X_s}(T_s)\right)^{d_{s+1}} \\ \text{hence,} \qquad h_{s+1} &= d_{s+1}! \prod_{i=2}^{s}\mathbf{d}_{(i,s+1)}! \left(\prod_{i=1}^{s}\partial_{X_i}(T_i)^{d_{i+1}\cdots d_{s+1}}\right) \end{aligned}$$

This is Formula *(iii)*. Let us now prove *(ii)*.

In the ring $(K[X_1, \ldots, X_{s-1}])/(T_1, \ldots, T_{s-1})[X_s]$ the polynomials $\partial_{X_s}(T_s)$ and T_s have no common root, else the ideal $(T_1, \ldots, T_n)$ would not be radical. Therefore, $\partial_{X_s}(T_s)$ is invertible in that ring. By induction, so it is for h_s, and by Equation (2.22), h_{s+1} also, as a product of invertible elements.

Finally, the polynomial $M_{s+1} := \partial\partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})(1,0,\ldots,0,1)$ vanishes on $\pi_{s+1}^n(V)$, is of degree d_{s+1} in X_{s+1} (from Equation (2.19)). The polynomial $P = \left(h_{s+1}^{-1} \bmod (T_1, \ldots, T_s)\right) M_{s+1}$ verifies the same properties but is moreover monic. The normal form $P \bmod (T_1, \ldots, T_s)$ of P with respect to the Gröbner basis $(T_1, \ldots, T_s)$ verifies the same features, but moreover the degree constraints ensure that it is T_{s+1}. □

This proof also shows the correctness of Algorithm 2.1.

2.1.4 Height of coefficients

In this subsection, bounds are provided for the alternative representation $(M_1, \ldots, M_n)$ of V (Cf. Definition 2.2). These bounds were the first insight to space complexity bounds for triangular representation. In the sequel of this section, the notation hereunder is used:

$$\begin{aligned} f_1 &:= \mathcal{C}_{s+1}(U_1, \ldots, U_{s+1}, T) \\ f_2 &:= \partial_2^{(d_2-1)d_3\cdots d_{s+1}}(\mathcal{C}_{s+1})(U_1, \ldots, U_{s+1}, T) \\ &\vdots \\ f_{s-1} &:= \partial_2^{(d_2-1)d_3\cdots d_{s+1}} \cdots \partial_{s-1}^{(d_{s-1}-1)d_s d_{s+1}}(\mathcal{C}_{s+1})(U_1, \ldots, U_{s+1}, T), \\ &= \partial_2^{\mathbf{d}_{(2,s+1)}} \cdots \partial_{s-1}^{\mathbf{d}_{(s-1,s+1)}}(\mathcal{C}_{s+1})(U_1, \ldots, U_{s+1}, T) \\ f_s &:= \partial_2^{\mathbf{d}_{(2,s+1)}} \cdots \partial_s^{\mathbf{d}_{(s,s+1)}}(\mathcal{C}_{s+1})(U_1, \ldots, U_{s+1}, T). \end{aligned}$$

The following notation denotes the coefficients of f_t :

$$\forall\, 2 \le t \le s, \qquad f_t := \sum_{i=0}^{D} f_{i,t} X_t^i, \quad \text{with } f_{i,t} \in K[U_1, \ldots, U_{s+1}, T][X_2, \ldots, X_{t-1}].$$

Repetitive application of the binomial formula gives:

$$f_{i,t} := \binom{\mathbf{d}_{(t,s+1)}}{i} \partial_2^{\mathbf{d}_{(2,s+1)}} \cdots \partial_{t-1}^{\mathbf{d}_{(t-1,s+1)}} \partial_{U_t}^{\mathbf{d}_{(t,s+1)}-i} \partial_T^i(\mathcal{C}_{s+1}).$$

Lemma 2.5. *With the notation above, the following inequalities hold:*

$$\log|f_t| \le \mathbf{d}_{(t,s+1)} \log D + \log|f_{t-1}| + \log\tbinom{\mathbf{d}_{(t,s+1)}}{\lfloor \mathbf{d}_{(t,s+1)}/2 \rfloor}, \qquad \textit{(number field case)}$$
$$\log|f_t| \le \log|f_{t-1}|, \qquad \textit{(function field case)}$$

Proof. Since ∂_{U_t} and ∂_T commute with $\partial_2, \ldots, \partial_{t-1}$, we have:

$$f_{i,t} = \binom{\mathbf{d}_{(t,s+1)}}{i} \partial_{U_t}^{\mathbf{d}_{(t,s+1)}-i} \partial_T^i \partial_2^{\mathbf{d}_{(2,s+1)}} \cdots \partial_{t-1}^{\mathbf{d}_{(t-1,s+1)}}(\mathcal{C}_{s+1}) = \binom{\mathbf{d}_{(t,s+1)}}{i} \partial_{U_t}^{\mathbf{d}_{(t,s+1)}-i} f_{t-1}.$$

Applying $(NA)_{\partial_{U_t}}$, $(A)_{\partial_T}$, $(A)_{\partial_{U_t}}$ and $(NA)_{\partial_T}$ page 61, at will we get:

$$\log|f_{i,t}|_v \le \begin{cases} \log\tbinom{\mathbf{d}_{(t,s+1)}}{i} + \mathbf{d}_{(t,s+1)} \log D + \log|f_{t-1}|_v, & \text{if } v \text{ is Archimedean} \quad (i), \\ \log|f_{t-1}|_v, & \text{if } v \text{ is non-Archimedean} (ii). \end{cases}$$

It is easy to show that:

$$\max\{|\text{ coeff } f_t|_v\} = \max_i\{\max\{|\text{ coeff } f_{i,t}|_v\}\}.$$

so that $\log|f_t|_v = \max_i \log|f_{i,t}|_v$, and by using (i) and (ii) at will:

$$\begin{aligned} \log|f_t|_v &\le \begin{cases} \mathbf{d}_{(t,s+1)} \log D + \log|f_{t-1}|_v + \max_i \log\tbinom{\mathbf{d}_{(t,s+1)}}{i}, & \text{if } v \text{ is Archimedean} \\ \log|f_{t-1}|_v, & \text{if } v \text{ is non-Archimedean} \end{cases} \\ &\le \begin{cases} \mathbf{d}_{(t,s+1)} \log D + \log|f_{t-1}|_v + \log\tbinom{\mathbf{d}_{(t,s+1)}}{\lfloor \mathbf{d}_{(t,s+1)}/2 \rfloor}, & \text{if } v \text{ is Archimedean} \\ \log|f_{t-1}|_v, & \text{if } v \text{ is non-Archimedean.} \end{cases} \end{aligned}$$

It just remains to use the definition of the height for the function or number field case. □

This lemma provides an inductive relation to get the Chow form $\mathcal{C}_{s+1}$ from $\partial\partial_{s+1}^{d_{s+1}}(\mathcal{C}_{s+1})$ by deleting derivations.

Theorem 2.5. *The height $h(M_{s+1})$ of the polynomial M_{s+1} is upper bounded by the following quantity in the the number field case,*

$$h(\pi_{s+1}^n(V)) + \deg(\pi_{s+1}^n(V))\log(s+3) + (d_2\cdots d_s + d_{s+1} + 1)\log\deg(\pi_{s+1}^n(V))$$

and by the following quantity in the function field case:

$$h(M_{s+1}) \leq h(\pi_{s+1}^n(V)).$$

Proof. Let us start with the Archimedean case. From the previous lemma:

$$\log|\partial\partial_{s+1}\mathcal{C}_{s+1}|_v = \log|f_{s+1}|_v \leq d_{s+1}\log D + \log|f_s|_v + \log\binom{d_{s+1}}{\llcorner d_{s+1}/2\lrcorner},$$

so that

$$\log|f_{s+1}|_v - \log|f_s|_v + \log|f_s|_v - \log|f_{s-1}|_v + \cdots + \log|f_2|_v - \log|f_1|_v$$

is upper-bounded by

$$\sum_{j=2}^{s+1}\mathbf{d}_{(j,s+1)}\log D + \log\binom{\mathbf{d}_{(j,s+1)}}{\llcorner\mathbf{d}_{(j,s+1)}/2\lrcorner}.$$

Since $f_1 = \mathcal{C}_{s+1}(U_1,\ldots,U_{s+1},T)$, it follows that:

$$\log|f_{s+1}|_v \leq d_2\cdots d_{s+1}\log D + \log|\mathcal{C}_{s+1}|_v + \log\left(\prod_{j=2}^{s+1}\binom{\mathbf{d}_{(j,s+1)}}{\llcorner\mathbf{d}_{(j,s+1)}/2\lrcorner}\right).$$

Now we are interested by the specializing; Let us rewrite the polynomial $f_{i,s+1}$:

$$f_{i,s+1} = \partial\partial_{U_{s+1}}^{d_{s+1}-i}\partial_T^i\mathcal{C}_{s+1}(U_1,\ldots,U_{s+1},T) = \sum_{j=0}^{D}a_jT^j\left(\sum_{\alpha\in\mathbb{N}^{s+1}}b_\alpha\mathbf{U}^\alpha\left(\sum_{\beta\in\mathbb{N}^{s-1}}c_\beta\mathbf{X}^\beta\right)\right),$$

where $\mathbf{U}^\alpha = U_1^{\alpha_1}\cdots U_{s+1}^{\alpha_{s+1}}$ for all $s+1$-uples α and $\mathbf{X}^\beta = X_2^{\beta_2}\cdots X_{s+1}^{\beta_{s+1}}$. Using these notations leads to:

$$\log|f_{i,s+1}(U_1,\ldots,U_{s+1},T)|_v = \log\max_{(j,\alpha,\beta)}\{|a_jb_\alpha c_\beta|_v\},$$

and to

$$f_{i,s+1}(1,0,\ldots,0,X_1) = \sum_{j=0}^{D}a_jX_1^j\left(\sum_{\alpha_2=\cdots=\alpha_{s+1}=0}b_\alpha\left(\sum_\beta c_\beta\mathbf{X}^\beta\right)\right).$$

It follows that:

$$\log|f_{i,s+1}(1,0,\ldots,0,X_1)|_v = \log\max_{(j,\beta)}\{|a_jc_\beta(\sum_{\alpha_2=\cdots=\alpha_{s+1}=0}b_\alpha)|_v\}.$$

But,

$$\forall j,\beta\ |a_jc_\beta(\sum_{\alpha_2=\cdots=\alpha_{s+1}=0} b_\alpha)|_v \le D|a_jc_\beta|_v \max_\alpha |b_\alpha|_v$$
$$\le D\max_\alpha\{|a_jc_\beta b_\alpha|_v\}$$
$$\text{so that}\qquad \max_{(j,\beta)}\{|a_jc_\beta(\sum_{\alpha_2=\cdots=\alpha_{s+1}=0} b_\alpha)|_v\} \le D\max_{(j,\alpha,\beta)}\{|a_jc_\beta b_\alpha|_v\},$$

and as a consequence,

$$\log|f_{i,s+1}(1,0,\ldots,0,X_1)|_v \le \log D + \log|f_{i,s+1}(U_1,\ldots,U_{s+1},T)|_v.$$

By using the Lemma 2.5 and the bound (i) page 74, we get:

$$\begin{aligned}\log|f_{i,s+1}(1,0,..,0,X_1)|_v &\le \log D + \log\binom{d_{s+1}}{i} + d_{s+1}\log D + d_2\cdots d_s\log D\\ &\quad + \log|\mathcal{C}_{s+1}|_v + \log\left(\prod_{j=2}^{s}\binom{\mathbf{d}_{(j,s+1)}}{\llcorner\mathbf{d}_{(j,s+1)}/2\lrcorner}\right)\\ \log|f_{i,s+1}(1,0,\ldots,0,X_1)|_v &\le (d_2\cdots d_s + d_{s+1}+1)\log D + \log|\mathcal{C}_{s+1}|_v\\ &\quad + \log\left(\prod_{j=2}^{s+1}\binom{\mathbf{d}_{(j,s+1)}}{\llcorner\mathbf{d}_{(j,s+1)}/2\lrcorner}\right)\end{aligned}\tag{2.23}$$

From inequalities (1.13) and (1.14), we introduce the S_{s+2}-Mahler measure:

$$\begin{aligned}\log|\mathcal{C}_{s+1}| &\le m(\mathcal{C}_{s+1};S_{s+2}) + \deg(\mathcal{C}_{s+1})\left(\sum_{i=1}^{s+1}\frac{1}{2i} + \log(s+3)\right)\\ \log|f_{i,s+1}(1,0,\ldots,0,X_1)|_v &\le (d_2\cdots d_s + d_{s+1}+1)\log D + m(\mathcal{C}_{s+1};S_{s+1})\\ &\quad + \deg(\mathcal{C}_{s+1})\left(\sum_{i=1}^{s+1}\frac{1}{2i} + \log(s+3)\right)\\ &\le (d_2\cdots d_s + d_{s+1}+1)\log D + h_v(\pi^n_{s+1}(V)) + \log(s+3)\deg(\pi^n_{s+1}(V))\end{aligned}$$

This achieves the Archimedean case. The case v non-Archimedean is easier. From Lemma 2.5 $\log|f_{s+1}|_v \le \log|f_s|_v$, and using the ultrametric inequality,

$$\max_{(j,\beta)}\{|a_jc_\beta(\sum_{\alpha_2=\cdots=\alpha_{s+1}=0} b_\alpha)|_v\} \le \max_{(j,\alpha,\beta)}\{|a_jc_\beta b_\alpha|_v\}.$$

So that, with inequality (ii) page 74:

$$\log|f_{i,s+1}(1,\ldots,0,X_1)|_v \le \log|f_{i,s+1}(U_1,\ldots,U_{s+1},T)|_v$$

In fine,

$$\log|f_{i,s+1}(1,\ldots,0,X_1)|_v \le \log|\mathcal{C}_{s+1}|_v.$$

Using the v-adic definition of the height of a variety (1.11), permits to get $h_v(M_{s+1}) \le h_v(\pi^n_{s+1}(V))$. Using the Archimedean definition (1.12) of the height of a variety permits to conclude the proof. □

2.1.5 Attempt of bounds for $(T_i)_i$ from $(M_i)_i$

We investigate in this paragraph what kind of bounds on the polynomials $(T_1, \ldots, T_n)$ can be obtained from the one given on the polynomials $(M_1, \ldots, M_n)$ in Theorem 2.5. We follow Algorithm 2.1 turning the regular chain $(M_1, \ldots, M_n)$ into a triangular set. It relies of course on Theorem 2.4 permitting to inverse the leading coefficient. We quantify the different subroutines of the algorithm, that it to say `NormalForm` and `ModInv`. How do the coefficients of a polynomial grow under these operations ? Langemyr answers in [69] Theorem 3 and 8, in the case of integers. We transcribe his results in term of height (essentially, by taking the logarithm of his bounds, at least in the Archimedean case).

With this method, we obtain quadratic bounds for polynomials $NF_1, \ldots, NF_n$, which is satisfactory. But for $T_1, \ldots, T_n$, this method seems not give polynomial bounds, due to the blowing-up of the coefficients under the `ModInv` operation.

`NormalForm`$(P, (T_1, \ldots, T_n))$

#`Inputs`: $P \in K[X_1, \ldots, X_n]$.
A triangular set $(T_1, \ldots, T_n)$, $\deg_{X_i}(T_i) = d_i$.
#`Output`: The normal form $Q \equiv P \bmod (T_1, \ldots, T_n)$.

1. if $(n = 1)$ then return the remainder Q of the Euclidean division of P by T_1.
2. write $P = \sum_{i=0}^{\deg_{X_n}(P)} P_i X_n^i, \quad P_i \in K[X_1, \ldots, X_{n-1}]$.
3. for $i = 0$ to ℓ_n do

   $$\widetilde{P}_i = \texttt{NormalForm}(P_i, (T_1, \ldots, T_{n-1})).$$

   end for
4. return the remainder Q of the Euclidean division of $\sum_{i=0}^{\deg_{X_n}(P)} \widetilde{P}_i X_n^i$ by T_n, over $K[X_1 \ldots, X_{n-1}]/(T_1, \ldots, T_{n-1})$.

Algo **2.2:** Recursive algorithm for the normal form of a polynomial

Define $\widetilde{P} := \sum_{i=0}^{\ell_n} \widetilde{P}_i X_n^i$ the polynomial obtained after the for loop (Step 3 of Algo. 2.2), where polynomials $\widetilde{P}_i$ are in normal form modulo $(T_1, \ldots, T_{n-1})$. We can see it as a polynomial over $K[X_1, \ldots, X_{n-1}]/(T_1, \ldots, T_{n-1})$. Let v be an value over K. It is well known that for univariate polynomials A and B over K, with B monic and with $\deg(B) < \deg(A)$, the remainder R of the Euclidean division of A by B verifies:

$$h_v(R) \leq \begin{cases} h_v(A) + (\deg(A) - \deg(B) + 1)(h_v(B) + \log(2)), & \text{if } v \text{ is Archimedean,} \\ h_v(A) + (\deg(A) - \deg(B) + 1)h_v(B), & \text{if } v \text{ is non-Archimedean..} \end{cases}$$

We need to extend this inequality to the base ring $K[X_1, \ldots, X_i]/(T_1, \ldots, T_i)$. An element a in this ring is represented by a reduced polynomial $a(X_1, \ldots, X_i)$ with $\deg_{X_j}(a) < d_j$, for $j = 1, \ldots, i$. Let Q_i^v be an upper bound on the growth of the coefficients under the multiplication operation in $K[X_1, \ldots, X_i]/(T_1, \ldots, T_i)$, that is to say: if a and b are elements in

this ring, then the product $c \equiv a \cdot b \bmod (T_1, \ldots, T_i)$ verifies:

$$h_v(c) \leq \begin{cases} h_v(a) + h_v(b) + Q_i^{\infty}, & \text{if } v \text{ is Archimedean,} \\ h_v(a) + h_v(b) + Q_i^{0}, & \text{if } v \text{ is non-Archimedean.} \end{cases}$$

In Langemyr [69, Theorem 3] to any Archimedean absolute value v, and get:

$$Q_i^{\infty} = \sum_{j=1}^{i} d_{j+1} \cdots d_i \log(d_j) + d_j \cdots d_i (\log(2) + h_v(T_j)), \text{ for } i \geq 1.$$

It not difficult to prove by induction that for the non-Archimedean case, we have:

$$Q_i^{0} = \sum_{j=1}^{i} d_j \cdots d_i h_v(T_j), \quad \text{for} \quad i \geq 1.$$

Hence when quantifying the coefficients swell during the Euclidean division in Step 4, we get:

$$h_v(Q) \leq \begin{cases} h_v(\widetilde{P}) + (\deg_{X_n}(P) - d_n + 1)(h_v(T_n) + Q_{n-1}^{\infty} + \log(2)), & \text{if } v \text{ is Archimedean,} \\ h_v(\widetilde{P}) + (\deg_{X_n}(P) - d_n + 1)(h_v(T_n) + Q_{n-1}^{0}), & \text{if } v \text{ is non-Archimedean.} \end{cases}$$

Denote $\ell_i = \deg_{X_i}(P) - d_i + 1$. Recursively, we get the height of $Q = \texttt{NormalForm}(P, (T_1, \ldots, T_n))$:

$$h_v(Q) \leq \begin{cases} h_v(P) + \sum_{j=1}^{n} \ell_j (h_v(T_j) + Q_{j-1}^{\infty} + \log(2)), & \text{if } v \text{ is Archimedean,} \\ h_v(P) + \sum_{j=1}^{n} \ell_j (h_v(T_j) + Q_{j-1}^{0}), & \text{if } v \text{ is non-Archimedean.} \end{cases}$$

We make simplifications such as $\log(d_i) \leq d_i$ and $\log(2) \leq 1$, which do not devalue the quality of the bounds; yielding, for $i \geq 1$:

$$Q_i^{\infty} \leq d_1 \cdots d_i (2 + h_v(T_1)) + d_2 \cdots d_i (2 + h_v(T_2)) + \cdots + d_{i-1} d_i (2 + h_v(T_{i-1})) + d_i (2 + h_v(T_i)). \tag{2.24}$$

The aim of this paragraph is an attempt to obtain bounds on polynomials NF_i, and polynomials T_i through Algorithm 2.1. It is require to compute first bounds for polynomials NF_i, since they appear to be a "step" for getting bounds on T_i, regarding to Step 4 of Algorithm 2.1. In our problem, these are the coefficients of M_n that we need to reduce. In fact, from Formula (2.19):

$$M_n = \sum_{i=0}^{d_n} X_n^i \cdot \binom{d_n}{i} \partial \partial_{U_n}^{d_n - i} \partial_T^i (\mathcal{C}_V)(1, \ldots, 0, X_1).$$

Hence, for $i < n$, $\deg_{X_i}(M_n) = \mathbf{d}_{(i,n)} = (d_i - 1) d_{i+1} \cdots d_n$. Therefore, we go on with polynomials in $n - 1$ variables, to plug the results for the polynomials M_n.

Let us go back to Equation (2.24). By taking ℓ_i equal to $\deg_{X_i}(P) - d_i + 1$, it comes for v non-Archimedean:

$$\begin{aligned}
h_v(Q) \leq\ & h_v(P) + \ell_1\big(h_v(T_1) + 2\big) \\
& + \ell_2\Big(h_v(T_2) + d_2(2 + h_v(T_2)) + d_1d_2(2 + h_v(T_1))\Big) \\
& \vdots \\
& + \ell_{n-2}\Big(h_v(T_{n-2}) + d_{n-2}(2 + h_v(T_{n-2})) + d_{n-2}d_{n-3}(2 + h_v(T_{n-3})) + \cdots \\
& \qquad\qquad \cdots + d_1 \cdots d_{n-2}(2 + h_v(T_1))\Big) \\
& + \ell_{n-1}\Big(h_v(T_{n-1}) + d_{n-1}(2 + h_v(T_{n-1})) + d_{n-2}d_{n-1}(2 + h_v(T_{n-2})) + \cdots \\
& \qquad\qquad \cdots + d_1 \cdots d_{n-1}(2 + h_v(T_1))\Big).
\end{aligned}$$

And if v is non-Archimedean, we have:

$$h_v(Q) \leq h_v(P) + \ell_1 h_v(T_1) + \sum_{j=2}^{n-1} \ell_j \left(h_v(T_j) + \sum_{k=1}^{j-1} d_k \cdots d_j h_v(T_k) \right)$$

In these sums, the term $h_v(T_i)$ appears for $i \geq 2$ with the factor:

$$\ell_i(1 + d_i) + \ell_{i+1}d_id_{i+1} + \cdots + \ell_{n-2}d_i \cdots d_{n-2} + \ell_{n-1}d_i \cdots d_{n-1}. \tag{2.25}$$

For $h_v(T_1)$, it appears with the same factor above except for the first term which is equal to ℓ_1 and not $\ell_1(1 + d_1)$. Now we do not replace ℓ_i by $\mathbf{d}_{(i,n)} - d_i + 1$, but only by $\mathbf{d}_{(i,n)}$, which is an acceptable simplification. The sum (2.25) is now bounded for $i \geq 2$ by

$$C_i := (1 + d_i)\mathbf{d}_{(i,n)} + d_id_{i+1}\mathbf{d}_{(i+1,n)} + \cdots + d_i \cdots d_{n-1}\mathbf{d}_{(n-1,n)}.$$

If $D = d_1 \cdots d_n$, then :

$$C_i \leq (d_i + \cdots + d_{n-1} - n + i)\frac{D}{d_1 \cdots d_{i-1}}.$$

The inequality for $h_v(Q)$ is then rewritten:

$$h_v(Q) \leq h_v(P) + C_1h_v(T_1) + C_2h_v(T_2) + \cdots + C_{n-1}h_v(T_{n-1})\Big[+ \mathsf{R}\Big]_{v \text{ Archimedean}},$$

where $\mathsf{R} = \ell_1 + \sum_{j=2}^{n-1} \ell_j \,(2d_j + 2d_{j-1}d_j + \cdots + 2d_1 \cdots d_j)$. It simplifies into $D\left(\sum_{j=2}^{n} j(d_{j-1} - 1)\right)$. The bound for v non-Archimedean is the same *without* the term R.

Let us consider the initial h_n of M_n. We want to evaluate the height of its normal form H_n with respect to $(T_1, \ldots, T_{n-1})$. Then Q is replaced by H_n and P by h_n in the inequality above:

$$h_v(H_n) \leq h_v(h_n) + C_1h_v(T_1) + C_2h_v(T_2) + \cdots + C_{n-1}h_v(T_{n-1})\Big[+ \mathsf{R}\Big]_{v \text{ Archimedean}}.$$

In introduction, we have stated some polynomial bounds for the polynomials $T_1, \ldots, T_n$. These bounds will be proved in the next section. Let us try to get similar bounds by following Algorithm 2.1.

In this aim, we need to compute a modular inverse. Thanks to Theorem 2.4, H_n admits an inverse modulo $(T_1, \ldots, T_{n-1})$. Theorem 8 of Langemyr's paper [69] says that $h_v(a^{-1}) \in O(l)$,where:

$$l = h_v(a)\Big(\sum_{i=1}^{n-1} d_i \cdots d_{n-1} + h_v(T_i)\Big(\sum_{j=i}^{n-1} d_1 \cdots d_j + d_2 \cdots d_j + \cdots + d_{i-1} \cdots d_j + d_i \cdots d_j\Big)(d_i \cdots d_j)\Big).$$

The formula becoming sophisticated, it is preferable before simplifying expressions to get bound for $n = 2, 3$ or 4. Even for such small values of n, I have preferred to use a computer to calculate these recursive formulas in order not to make any simplification. Thus, if the bounds obtained in that way are bad, then it is not of use to investigate further this attempt. We compute a $2i$ -variate polynomial $\mathrm{Bound}_i(x_1, \ldots, x_i, y_1, \ldots, y_i) \in \mathbb{Z}[x_1, \ldots, x_i, y_1, \ldots, y_i]$ verifying:

$$h_v(T_i) \leq \mathrm{Bound_i}(h_v(\pi_1^n(V)), \ldots, h_v(\pi_i^n(V)), d_1, \ldots, d_i),$$

and obtained recursively by using the recursive bounds given by Langemyr for the `NormalForm` and `ModInv` algorithm. Here is a report of the computations: For $n = 4$ and $i = 3$, the monomials $h_v(\pi_3^4(V))\big(1 + d_2 + d_2 d_3\big)$ and $d_1^4 d_2^3 d_3$ appears in Bound_3. In the polynomial Bound_4, the monomial $h_v(V)(d_1 d_2 d_3 + d_2 d_3 + d_3)$ appear and the highest degree monomial is $d_1^5 d_2^4 d_3^3 d_4$. It is already not a polynomial bound, getting worst as n grows, and it seems impossible to get a polynomial bound with respect to $d_1 \cdots d_n$ with this method. We give up the attempt to get bounds for the heights of triangular sets through Algorithm 2.1.

Consequently, in order to analyze the coefficient swell in the `NormalForm` algorithm, we make use of the bounds proved in the next section here. Theorem 2.7 together with the notations of Equations (2.1) yield for v Archimedean:

$$\begin{aligned} h_v(H_n) \;\leq\; & h_v(h_n) + C_1(\mathsf{G}_1 h_v(\pi_1^n(V)) + \mathsf{I}_1) + C_2(\mathsf{G}_2 h_v(\pi_2^n(V)) + \mathsf{I}_2) + \cdots \\ & \cdots + C_{n-1}(\mathsf{G}_{n-1} h_v(\pi_{n-1}^n(V)) + \mathsf{I}_{n-1}) + \mathsf{R}, \end{aligned}$$

and for v non-Archimedean:

$$h_v(H_n) \leq h_v(h_n) + C_1 \mathsf{G}_1 h_v(\pi_1^n(V)) + C_2 \mathsf{G}_2 h_v(\pi_2^n(V)) + \cdots + C_{n-1} \mathsf{G}_{n-1} h(\pi_{n-1}^n(V)).$$

This bound is actually valid for any coefficient $b_i := \binom{d_n}{i} \partial \partial_{U_n}^{d_n - i} \partial_T^i(\mathcal{C}_V)(1, \ldots, 0, X_1)$ of X_n^i in M_n. Since the height of M_n is equal to the maximal height of its coefficients b_i, it follows that, for v Archimedean, we have:

$$\begin{aligned} h_v(NF_n) \;\leq\; & h_v(M_n) + C_1 \cdot h_v(\mathsf{G}_1 h_v(\pi_1^n(V)) + \mathsf{I}_1) + C_2 \cdot h_v(\mathsf{G}_2 h_v(\pi_2^n(V)) + \mathsf{I}_2) + \cdots \\ & \cdots + C_{n-1} h_v(\mathsf{G}_{n-1} h_v(\pi_{n-1}^n(V)) + \mathsf{I}_{n-1}) + \mathsf{R} \\ \leq\; & h_v(M_n) + \sum_{i=1}^{n-1} d_i \cdots d_{n-1}\,(d_i + d_{i+1} + \cdots + d_{n-1} - n + i)\,(\mathsf{G}_i h_v(\pi_i^n(V)) + \mathsf{I}_i) + \mathsf{R}. \end{aligned}$$

And when v is not Archimedean, we have:

$$h_v(NF_n) \leq h_v(M_n) + \sum_{i=1}^{n-1} d_i \cdots d_{n-1} \left(d_i + d_{i+1} + \cdots + d_{n-1} - n + i\right) \mathsf{G}_i h_v(\pi_i^n(V)).$$

It is easy to check from Definitions 2.1 that:

- $\mathsf{G}_i d_i \cdots d_{n-1} < D$
- $\mathsf{H}_i d_i \cdots d_{n-1} \leq 5 \log(i+3) D$, implying $\sum_{i=1}^{n-1} \mathsf{H}_i d_i \cdots d_{n-1} \leq 5n^2 D$
- $\mathsf{I}_i d_i \cdots d_{n-1} \leq 5 \log(i+3) D + 3D d_1 \cdots d_{i-1} \log(i+2)$

It follows that $\sum_{i=1}^{n-1} \mathsf{I}_i d_i \cdots d_{n-1} \leq 5n^2 D + 3n^3 D^2$. Moreover $\mathsf{R} \leq D \left(\sum_{j=2}^{n} j(d_{j-1} - 1) \right)$ is upper bounded by $n^2 D^2 \leq n^3 D^2$. If we replace $h_v(M_n)$ by the bounds of Theorem 2.5, we finally have if v is Archimedean:

$$\begin{aligned} h_v(NF_n) &\leq h_v(V) + D \log((n+2)D) + D \left(\sum_{i=1}^{n-1} (d_i + \cdots + d_{n-1} - n + i) h_v(\pi_i^n(V)) \right) \\ &\quad + 5n^2 D + 4n^3 D^2 \\ &\leq h_v(V) + D \log((n+2)D) + (n-1) D^2 h_v(V) + 5n^2 D + 4n^3 D^2 \end{aligned}$$

Let us turn to the non-Archimedean case:

$$\begin{aligned} h_v(NF_n) &\leq h_v(V) + D \left(\sum_{i=1}^{n-1} (d_i + \cdots + d_{n-1} - n + i) h_v(\pi_i^n(V)) \right) \\ &\leq h_v(V) + (n-1) D^2 h_v(V) \end{aligned}$$

By using the definition of the height of a variety, we finally get the following theorem:

Theorem 2.6. *Let V be an equiprojectable variety defined by a triangular set over K of degree $d_1, \ldots, d_n$. Then the height of the polynomials NF_i introduced in Definition 2.3 verifies:*

$$\begin{aligned} h(NF_i) &\leq h(\pi_i^n(V)) + \deg(\pi_i^n(V)) \Big(\log\left((n+2)\deg(\pi_i^n(V))\right) \\ &\quad + (n-1)\deg(\pi_i^n(V)) h(\pi_i^n(V)) + 5n^2 + 4n^3 \Big), \qquad \textit{(K is a number field)} \end{aligned}$$

$$h(NF_i) \leq h(\pi_i^n(V)) \left(1 + (n-1)\deg(\pi_i^n(V))^2\right), \qquad \textit{(K is a function field)}$$

The bound is indeed polynomial, and actually cubic with respect to the degree and the height of V, because of the term $D^2 h_v(V)$. The bounds for the polynomials T_i that we have used are quadratic. However, experiments show that the coefficients of the NF_i are often smaller than those of T_i (see Table 2.2 page 51). I can not explain this fact, except that bound for $h_v(NF_n)$ is weakly cubic, regarding to the simplifications made.

2.2 Bounds from interpolation formulas

We use here some generalized Lagrange interpolation formulas to make appear a geometric link between the points of the underlying variety of our triangular set. This allows to use the classical height bounds of § 1.2.2 taken from [66] Lemma 2.1. The results of this paragraph have been published in [32] with É. Schost.

Definition 2.3. *Let $D_1 = 1$ and $N_1 = T_1$. For ℓ in $2, \ldots, n$, define*

$$\begin{aligned} D_\ell &= \prod_{1 \le i \le \ell-1} \frac{\partial T_i}{\partial X_i} \mod (T_1, \ldots, T_{\ell-1}), \\ N_\ell &= D_\ell T_\ell \mod (T_1, \ldots, T_{\ell-1}). \end{aligned}$$

Note that $D_\ell \in K[X_1, \ldots, X_{\ell-1}]$, $N_\ell \in K[X_1, \ldots, X_{\ell-1}, X_\ell]$, and D_ℓ is the leading coefficient of N_ℓ viewed as a univariate polynomial in X_ℓ and with coefficients in $K[X_1, \ldots, X_{\ell-1}]$.

The difference between these polynomials N_ℓ and the polynomials T_ℓ of the corresponding triangular set is to be compared with the difference between the Kronecker representation and the Shape Lemma representation (see Ch. I, § 1.1.2). We are goring to prove the same diminution of the size of the coefficients. It should be noted that our estimates are a faithful extension of these results to triangular representations.

2.2.1 Interpolation formulas

The classical Lagrange formula permits to interpolate some values f_i at given points e_i in $\bar{K}$:

$$\mathrm{Lag}(X_1) = \sum_i f_i \prod_{j \neq i} \frac{X_1 - e_j}{e_i - e_j}.$$

We extend in this paragraph this formula to the polynomials of a triangular set $\mathbf{T}$. A natural generalization is to consider for $T_{\ell+1}$:

$$\sum_{\alpha \in \pi_\ell^n(V)} T_{\ell+1}(\alpha_1, \ldots, \alpha_\ell, X_{\ell+1}) F_\alpha(X_1, \ldots, X_\ell), \quad \text{with } F_\alpha(\beta) = \begin{cases} 1 & \text{if } \beta = \alpha \text{ and,} \\ 0 & \text{if } \beta \in \pi_\ell^n(V) \setminus \{\alpha\}. \end{cases} \tag{2.26}$$

The role played by F_α is the same as the one played by $\prod_{j \neq i} \frac{X_1 - e_j}{e_i - e_j}$, they are called *idempotents*. An idempotent fits if it verifies the following formula:

Lemma 2.6. *If $\deg_{X_i} F_\alpha < d_i$, $\forall\, i = 1, \ldots, n$, then $T_{\ell+1}$ is equal to the polynomial (2.26) above. Such an idempotent is then unique.*

Proof. By definition, the polynomial (2.26) is equal to $T_{\ell+1}$ modulo $(T_1, \ldots, T_\ell)$. The degree constraint insures that they are both reduced with respect to the Gröbner basis $(T_1, \ldots, T_\ell)$, so they are equal. This argument also proves uniqueness. □

It remains to construct such polynomials F_α, for each $\alpha \in \pi_\ell^n(V)$. For such a fixed α, consider the following subfamilies of points of $\pi_{\ell+1}^n(V)$

- $V_\alpha^1 := \{(\beta_1, \ldots, \beta_{\ell+1}) \in \pi_{\ell+1}^n(V) \text{ such that } \beta_1 \neq \alpha_1\}$.
- $V_\alpha^2 := \{(\alpha_1, \beta_2, \ldots, \beta_{\ell+1}) \in \pi_{\ell+1}^n(V) \text{ such that } \beta_2 \neq \alpha_2\}$.

- generally, for $i = 1, \dots, \ell$, $V_\alpha^i = \{(\alpha_1, \dots, \alpha_{i-1}, \beta_i, \beta_{i+1}, \dots, \beta_{\ell+1}) \in \pi^n_{\ell+1}(V)$ such that $\beta_i \neq \alpha_i\}$
- finally, $V_\alpha^{\ell+1} := \{(\alpha_1, \dots, \alpha_\ell, \beta_{\ell+1}) \in \pi^n_{\ell+1}(V)\}$.

We also consider the projection of V_α^i on on the X_i-axis: this set of elements of $\bar{K}$ is denoted by $v_{\alpha,i}$. Its cardinal is $d_i - 1$ for $i \leq \ell$ and $\# v_{\alpha,\ell+1} = d_{\ell+1}$. Define also $e_{\alpha,i} := \prod_{x \in v_{\alpha,i}} (X_i - x)$, and $T_{\alpha,i} = T_i(\alpha_1, \dots, \alpha_{i-1}, X_i)$.

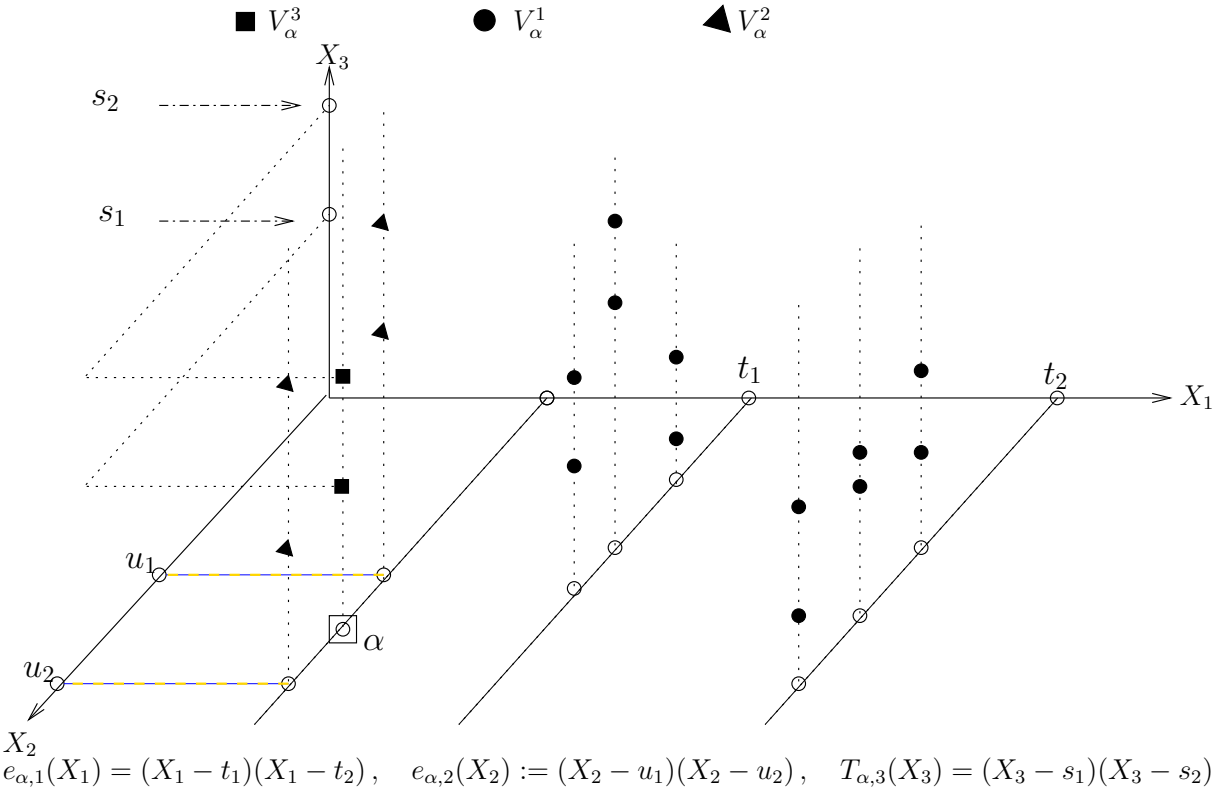

Figure 2.3: Example of partition in a 3-dimensional space

Proposition 2.5. *With the notation of Lemma 2.6:*

$$F_\alpha = E_\alpha(X_1, \dots, X_\ell)/E_\alpha(\alpha), \qquad \textit{where} \qquad E_\alpha := \prod_{i=1}^{\ell} e_{\alpha,i}.$$

Hence, $T_{\ell+1} = \sum_{\alpha \in \pi^n_\ell(V)} T_{\alpha,\ell+1} \frac{E_\alpha}{E_\alpha(\alpha)}$.

Proof. $E_\alpha(\alpha) \neq 0$ since none of the $e_{\alpha,i}$ vanishes on α_i, by construction. However, if $\beta \in \pi^n_\ell(V) \setminus \{\alpha\}$, then at least one coordinate β_i is distinct from α_i. Thus $\beta_i \in v_{\alpha,i}$, and $e_{\alpha,i}(\beta_i) = 0$. We have $E_\alpha(\beta) = 0$, implying that $E_\alpha(\,.\,)/E_\alpha(\alpha)$ is an idempotent. Moreover the degree constraints of Lemma 2.6 are verified, and we conclude by the unicity assertion of that lemma. □

Corollary 2.3. *The polynomial $N_{\ell+1}$ defined in Definition 2.3 is rewritten:*

$$N_{\ell+1} = \sum_{\alpha \in \pi_\ell^n(V)} T_{\ell+1}(\alpha_1, \ldots, \alpha_\ell, X_{\ell+1}) E_\alpha(X_1 \ldots, X_\ell).$$

Proof. Let $\alpha \in \pi_\ell^n(V)$. Since $E_\beta(\alpha) = 0$ for $\beta \neq \alpha$, the right-hand side reduces to $E_\alpha(\alpha) T_{\ell+1}(\alpha, X_{\ell+1})$, so we are left to estimate the value $E_\alpha(\alpha)$. Since the roots of $T_{\alpha,i}(X_i)$ are the values of $v_\alpha^i \cup \{\alpha_i\}$ for $i \leq \ell$, we have

$$T_{\alpha,i}(X_i) = T_i(\alpha_1, \ldots, \alpha_{i-1}, X_i) = (X_i - \alpha_i).e_{\alpha,i}(X_i),$$

from which we deduce that

$$e_{\alpha,i}(\alpha_i) = \frac{\partial T_i}{\partial X_i}(\alpha_1, \ldots, \alpha_i).$$

And we get $E_\alpha(\alpha) = \left(\prod_{i=1}^{\ell} \frac{\partial T_i}{\partial X_i}(\alpha)\right)$. On the other hand, from Definition 2.3 ,

$$N_{\ell+1}(\alpha, X_{\ell+1}) = \left(\prod_{1 \leq i \leq \ell} \frac{\partial T_i}{\partial X_i}(\alpha)\right) T_{\ell+1}(\alpha, X_{\ell+1}) \in \bar{K}[X_{\ell+1}]. \tag{2.27}$$

Hence $N_{\ell+1}(\alpha, X_{\ell+1}) = E_\alpha(\alpha) T_{\ell+1}(\alpha, X_{\ell+1})$. Both sides of the equation agree on $\pi_\ell^n(V)$, hence agree modulo $(T_1, \ldots, T_\ell)$. To conclude at the equality, we prove that the right-hand term is reduced with respect that Gröbner basis, since it is the case by definition for $N_{\ell+1}$. For $1 \leq i \leq \ell$, $\deg_{X_i}(E_\alpha) = \deg(e_{\alpha,i})$ which is equal by definition to $d_i - 1$. It follows that E_α is in normal form with respect to the Gröbner basis $T_1, \ldots, T_\ell$. The degree in X_i of the right-hand term in the corollary is at most the degree in X_i of E_α, hence this term is also reduced with respect to $T_1, \ldots, T_\ell$. □

Some previous works exist and give satisfactory results for interpolating Gröbner basis from a family of points. The earlier work of Buchberger-Möller [23], gives a strikingly simple recursive algorithm to construct the minimal reduced lexicographic Gröbner basis of a given finite family of points. But they do not provide interpolations formula workable for our purpose. The Lazard structural theorem [70] gives explicit formulas that verify the polynomials of a bivariate Gröbner basis. There is no doubt that height bounds can be deduced for these polynomials, using the same technique as here. The specificity of our work is a simplicity, circumvent the technical aspect of the Lagrange Gröbner basis interpolation of Möller. Moreover the partition of the variety by the V_α^i is a key point of our work. Let us mention some recent works concerning interpolation of Gröbner bases, notably [83, 84].

Let us conclude this paragraph by defining the constants useful for the sequel.

$$e_i = \prod_{\alpha \in \pi_i^n(V)} e_{\alpha,i}(\alpha_i) \quad \text{for } i \leq \ell, \qquad \text{and} \qquad E_\ell = \prod_{1 \leq i \leq \ell} e_i.$$

The equation of Proposition 2.5 is equivalent to write $T_{\ell+1}$ as the quotient of

$$\mathfrak{T}_{\ell+1} = \sum_{\alpha \in \pi_\ell^n(V)} \frac{E_\alpha T_{\alpha,\ell+1} E_\ell}{E_\alpha(\alpha)} = E_\ell T_{\ell+1}$$

by E_ℓ. We now show that both quantities are defined over K; in Section 2.2.3, we will actually prove bounds on $\mathfrak{T}_{\ell+1}$, and deduce bounds for $T_{\ell+1}$.

Lemma 2.7. *The polynomial $\mathfrak{T}_{\ell+1}$ is in $K[X_1, \ldots, X_{\ell+1}]$.*

Proof. Since $T_{\ell+1}$ is defined over K, it suffices to prove that for $i \leq \ell$, e_i is in K. Given α in $\pi_i^n(V)$, we saw in the proof of Corollary 2.3 that $e_{\alpha,i}(\alpha_i) = \partial T_i/\partial X_i(\alpha)$. Thus, e_i is the determinant of the endomorphism of multiplication by $\partial T_i/\partial X_i$ modulo $(T_1, \ldots, T_i)$, so it is in K. □

2.2.2 Links with Chow forms

In this paragraph we rewrite the polynomials T_i and N_i in terms of relevant Chow forms; this step is fundamental to link the geometry of the underlying variety and the involved polynomials. It relies on the interpolation formulas of the previous paragraph.

- Denote by $\mathcal{C}_{\ell+1} \in K[X_1, \ldots, X_{\ell+1}, T]$ the Chow form of $\pi_{\ell+1}^n(V)$ for $\ell = 1, \ldots, n$,
- by $\mathcal{C}_{\alpha,i} \in \bar{K}[X_1, \ldots, X_{\ell+1}, T]$ the Chow form of V_α^i for $\alpha \in \pi_\ell^n(V)$, and for $i = 1, \ldots, \ell+1$.
- The multiplicative property of the Chow form (1.5) induces the following factorization:

$$\mathcal{C}_{\ell+1} = \prod_{i=1}^{\ell+1} \mathcal{C}_{\alpha,i}. \tag{2.28}$$

Let us start with some preliminary lemmas.

Lemma 2.8. *For α in $\pi_\ell^n(V)$ and $i \leq \ell$, we have*

$$\mathcal{C}_{\alpha,i}(0, \ldots, 0, 1, 0, \ldots, 0, X_i) = e_{\alpha,i}^{d_{i+1}\cdots d_{\ell+1}}(X_i) \tag{2.29}$$

$$\mathcal{C}_{\alpha,\ell+1}(1, 0, \ldots, 0, X_{\ell+1}) = T_{\alpha,\ell+1}(X_{\ell+1}). \tag{2.30}$$

Proof. By definition,

$$\mathcal{C}_{\alpha,i} = \prod_{\beta \in V_\alpha^i} (T - \beta_1 X_1 - \cdots - \beta_{\ell+1} X_{\ell+1}). \tag{2.31}$$

Thus the polynomial $\mathcal{C}_{\alpha,i}(0, \ldots, 1, \ldots, 0, X_i)$ is equal to $\prod_{\beta \in V_\alpha^i}(X_i - \beta_i)$. As for each value x of v_α^i there are $d_{i+1} \ldots d_{\ell+1}$ points of V_α^i which projects on x, we deduce that this product is actually equal to $\prod_{x \in v_\alpha^i}(X_i - x)^{d_{i+1}\cdots d_{\ell+1}}$. This proves equality (2.29). Equation (2.30) is easy to obtain. □

We insert two others formulas, not involving Chow forms but related to the ones above.

Lemma 2.9. *For $\alpha \in \pi_\ell^n(V)$, the following equality holds:*

$$\frac{E_\ell}{E_\alpha(\alpha)} = \prod_{1 \leq i \leq \ell} \prod_{\substack{\beta \in \pi_i^n(V) \\ \beta \neq \pi_i^\ell(\alpha)}} e_{\beta,i}(\beta_i). \tag{2.32}$$

For $i \leq \ell$, the following equality holds:

$$\prod_{\beta \in \pi_\ell^n(V)} e_{\beta,i}(X_i - \beta_i)^{d_i - 1} = \prod_{\beta \in \pi_i^n(V)} (X_i - \beta_i)^{2d_i - 2}. \tag{2.33}$$

Proof. The two equalities follow directly from the definitions of E_ℓ, E_α and $e_{\beta,i}$. □

2.2.3 From interpolation to height bounds

Using formulas in Proposition 2.5 and in Corollary 2.3, and the height inequalities of Section 1.2.2, we deduce in this subsection the height bounds announced.

Theorem 2.7. *For $0 \leq \ell \leq n-1$, the following inequalities hold:*

$$h(N_{\ell+1}) \leq \begin{cases} h(\pi^n_{\ell+1}(V)) \ , & \textit{(function fields)}, \\ h(\pi^n_{\ell+1}(V)) + \mathsf{H}_{\ell+1}, & \textit{(number fields)} \end{cases}$$

$$h(T_{\ell+1}) \leq \begin{cases} \mathbf{G}_{\ell+1}\, h(\pi^n_{\ell+1}(V)), & \textit{(function fields)} \\ \mathbf{G}_{\ell+1}\, h(\pi^n_{\ell+1}(V)) + \mathsf{I}_{\ell+1}, & \textit{(number fields)}. \end{cases}$$

The core of the proof is the following lemma, which involves the polynomials $\mathfrak{T}_{\ell+1}$ defined at the beginning of the paragraph.

Lemma 2.10. *Let $0 \leq \ell \leq n-1$. For $v \in M_K^0$ we have*

$$h_v(N_{\ell+1}) \leq h_v(\mathcal{C}_{\ell+1})\,, \qquad \textit{and} \qquad h_v(\mathfrak{T}_{\ell+1}) \leq \mathbf{G}_{\ell+1}\, h_v(\mathcal{C}_{\ell+1}).$$

For $v \in M_K^\infty$ and σ_v an isometric embedding into $\mathbb{C}$, we have

$$h_v(N_{\ell+1}) \leq m(\sigma_v(\mathcal{C}_{\ell+1})) + \mathsf{H}_{\ell+1}\,, \qquad \textit{and} \qquad h_v(\mathfrak{T}_{\ell+1}) \leq \mathbf{G}_{\ell+1}\, m(\sigma_v(\mathcal{C}_{\ell+1})) + \mathsf{I}_{\ell+1}.$$

Let us show how to derive Theorem 2.7. Plugging the estimates for $h_v(N_{\ell+1})$ in the definition of height, gives:

$$h(N_{\ell+1}) \ \leq \ \frac{1}{[K:K_0]} \sum_{v \in M_K^0} h_v(\mathcal{C}_{\ell+1}) + \frac{1}{[K:K_0]} \sum_{v \in M_K^\infty} (m(\sigma_v(\mathcal{C}_{\ell+1})) + \mathsf{H}_{\ell+1})\,,$$

and the first part of Theorem 2.7 follows from inequality $\mathbf{A_6}$ page 28. Similar arguments apply to $\mathfrak{T}_{\ell+1}$ and yield the bound

$$h(\mathfrak{T}_{\ell+1}) \leq \begin{cases} \mathbf{G}_{\ell+1}\, h(\pi^n_{\ell+1}(V)) + \mathsf{I}_{\ell+1} & (number field case), \\ \mathbf{G}_{\ell+1}\, h(\pi^n_{\ell+1}(V)) & \text{(function field case)} \end{cases}$$

Now, $T_{\ell+1}$ is obtained by dividing out $\mathfrak{T}_{\ell+1}$ by its leading coefficient in $X_{\ell+1}$. By the product formula, this operation lowers the global height, whence Theorem 2.7 follows. Thus, we can now focus on proving the lemma, using freely the notation of § 2.2.1 and § 2.2.2.

In what follows, we consider ℓ in $1, \ldots, n-1$. The case $\ell = 0$ follows along the same lines, by noting that $T_1 = N_1 = M_1$ is obtained by a suitable specialization of the Chow form $\mathcal{C}_1$ of $\pi^n_1(V)$ (see Section 2.1).

Let then L be a finite extension of K that contains all coordinates of all points in V. Let $w \in M_L$ extending an absolute value $v \in M_K$. Consider α in $\pi^n_\ell(V)$. Specializing indeterminates at zero decreases height, so

$$h_w(\mathcal{C}_{\alpha,i}(0, \ldots, 0, X_i, 0, \ldots, 0, T)) \leq h_w(\mathcal{C}_{\alpha,i}) \ \text{ for } i \leq \ell.$$

Since $\mathcal{C}_{\alpha,i}(0,\dots,0,X_i,0,\dots,0,T)$ is homogeneous, its local height coincides with that of $\mathcal{C}_{\alpha,i}(0,\dots,0,1,0,\dots,0,X_i)$. Then, Equations (2.29) and (2.30) finally give

$$\begin{aligned} h_w(e_{\alpha,i}^{d_{i+1}\cdots d_{\ell+1}}) &\leq h_w(\mathcal{C}_{\alpha,i}) \quad \text{for } i \leq \ell \qquad (2.34)\\ h_w(T_{\alpha,\ell+1}) &\leq h_w(\mathcal{C}_{\alpha,\ell+1}). \qquad (2.35)\end{aligned}$$

Case 1: w is non-Archimedean. We use equality $\mathbf{N_1}$ (Cf. § 1.2.2, p. 28) and Equations (2.34) and (2.35) to give

$$\begin{aligned} h_w(E_\alpha T_{\alpha,\ell+1}) &= \sum_{i\leq\ell} h_w(e_{\alpha,i}) + h_w(T_{\alpha,\ell+1})\\ &\leq \sum_{i\leq\ell} h_w(\mathcal{C}_{\alpha,i}) + h_w(\mathcal{C}_{\alpha,\ell+1}) = h_w(\mathcal{C}_{\ell+1}).\end{aligned}$$

Summing on all α, we deduce $h_w(N_{\ell+1}) \leq h_w(\mathcal{C}_{\ell+1})$ by inequality $\mathbf{N_2}$. Since both polynomials have coefficients in K, and w extends v, this proves the first part of Lemma 2.10.

Next, we consider $\mathfrak{T}_{\ell+1}$. Inequality $\mathbf{E}$ yields

$$h_w\left(\frac{E_\alpha T_{\alpha,\ell+1} E_\ell}{E_\alpha(\alpha)}\right) \leq h_w(E_\alpha T_{\alpha,\ell+1}) + h_w\left(\frac{E_\ell}{E_\alpha(\alpha)}\right). \qquad (2.36)$$

The term $h_w(E_\alpha T_{\alpha,\ell+1})$ was dealt with above. As to the other term, inequality $\mathbf{E}$ and Equation (2.32) shows that

$$h_w\left(\frac{E_\ell}{E_\alpha(\alpha)}\right) \leq \sum_{1\leq i\leq\ell} \sum_{\beta\in\pi_i^n(V)} h_w(e_{\beta,i}(\beta)), \qquad (2.37)$$

since the positivity of height enables us to complete the product in Equation (2.32). Then inequality $\mathbf{N_3}$ gives the upper bound

$$\sum_{1\leq i\leq\ell} \sum_{\beta\in\pi_i^n(V)} \left(h_w(e_{\beta,i}) + (d_i-1)h_w(\beta)\right).$$

Note that $h_w(\beta_i) = h_w(X_i - \beta_i)$, so by equality $\mathbf{N_1}$, the innermost term is $h_w(e_{\beta,i}(X_i - \beta_i)^{d_i-1})$. Using Equation (2.33), the inner sum is then bounded from above by

$$\sum_{\beta\in\pi_i^n(V)} h_w(e_{\beta,i}(X_i-\beta_i)^{d_i-1}) = h_w\Big(\prod_{\beta\in\pi_i^n(V)} (X_i-\beta_i)^{2d_i-2}\Big).$$

This quantity can be bounded from above by $2(d_i-1)h_w(\mathcal{C}_i)$. Note that $h_w(\mathcal{C}_i) \leq h_w(\mathcal{C}_{\ell+1})$; summing on $i \leq \ell$ and introducing the constant $\mathbf{G}_{\ell+1}$ gives the second point in Lemma 2.10.

Case 2: w is Archimedean. Let σ_v and σ_w be the isometric injections from K or L into $\mathbb{C}$. They coincide on polynomials with coefficients in K.

For $i \leq \ell$, since $\mathcal{C}_{\alpha,i}$ has degree $(d_i-1)d_{i+1}\cdots d_{\ell+1}$, inequality $\mathbf{A_2}$ gives

$$h_w(\mathcal{C}_{\alpha,i}) \leq m(\sigma_w(\mathcal{C}_{\alpha,i})) + (d_i-1)d_{i+1}\cdots d_{\ell+1}\log(\ell+2).$$

Thus, we deduce from inequality $\mathbf{A_4}$ and Equation (2.34)

$$\begin{aligned} h_w(e_{\alpha,i}) &\leq \frac{h_w(\mathcal{C}_{\alpha,i})}{d_{i+1}\cdots d_{\ell+1}} + 2(d_i-1)\log(\ell+2) \\ &\leq \frac{m(\sigma_w(\mathcal{C}_{\alpha,i}))}{d_{i+1}\cdots d_{\ell+1}} + 3(d_i-1)\log(\ell+2) \end{aligned}$$

Using $(d_{i+1}\cdots d_{\ell+1}) \geq 1$ and inequality $\mathbf{A_3}$, we obtain

$$h_w(E_\alpha) \leq \sum_{i\leq\ell} m(\sigma_w(\mathcal{C}_{\alpha,i})) + 4\log(\ell+2)\sum_{i\leq\ell}(d_i-1).$$

We next deduce from Equation (2.35) and inequality $\mathbf{A_2}$

$$h_w(T_{\alpha,\ell+1}) \leq m(\sigma_w(\mathcal{C}_{\alpha,\ell+1})) + \log(\ell+3)d_{\ell+1}.$$

Now, from (2.28), it follows that $m(\sigma_w(\mathcal{C}_{\ell+1})) = m(\sigma_w(\mathcal{C}_{\alpha,\ell+1})) + \sum_{i\leq\ell} m(\sigma_w(\mathcal{C}_{\alpha,i}))$, so applying inequality $\mathbf{A_3}$ yields

$$h_w(E_\alpha T_{\alpha,\ell+1}) \leq m(\sigma_w(\mathcal{C}_{\ell+1})) + 4\log(\ell+3)\sum_{i\leq\ell+1} d_i.$$

Summing over α and using inequality $\mathbf{A_5}$, we finally get

$$h_w(N_{\ell+1}) \leq m(\sigma_w(\mathcal{C}_{\ell+1})) + 4\log(\ell+3)\sum_{i\leq\ell+1} d_i + \log(d_1\cdots d_{\ell+1}).$$

Next, we use the inequality $\log(d_i) \leq d_i$ for all i. With the introduction of the constant $\mathsf{H}_{\ell+1}$, this finishes the proof of the third point in Lemma 2.10, since $N_{\ell+1}$ and $\mathcal{C}_{\ell+1}$ both have coefficients in K.

As for the last point of that lemma, note first that inequalities (2.36) and (2.37) hold in the Archimedean case as well; we now only have to bound the rightmost term of Equation (2.37).

Using inequalities $\mathbf{A_7}$ and $\mathbf{A_3}$ and Equation (2.33), an easy check proves that for $i \leq \ell$, the sum $\sum_{\beta\in\pi_i^n(V)} h_w(e_{\beta,i}(\beta))$ is bounded from above by

$$2(d_i-1)m\Big(\prod_{\alpha\in\pi_i^n(V)} \sigma_w(X_i-\alpha_i)\Big) + 3d_i(d_i-1)\log(2).$$

Now, we remark that $m(\sigma_w(X_i-\alpha_i)) = m(\sigma_w(T-\alpha_i X_i))$. Using the additivity of the Mahler measure, we deduce that the above quantity equals

$$2(d_i-1)m(\sigma_w(\mathcal{C}_i(0,\ldots,0,X_i,0,\ldots,0,T))) + 3d_i(d_i-1)\log(2).$$

Inequality $\mathbf{A_8}$ now shows that this can be bounded from above by $2(d_i-1)m(\sigma_w(\mathcal{C}_i)) + 3d_i(d_i-1)\log(2)$. Noticing that $m(\sigma_w(\mathcal{C}_i)) \leq m(\sigma_w(\mathcal{C}_{\ell+1}))$ and using the above estimates yields

$$\begin{aligned} h_w\left(\frac{E_\alpha T_{\alpha,\ell+1}E_\ell}{E_\alpha(\alpha)}\right) &\leq m(\sigma_w(\mathcal{C}_{\ell+1}))\Big(1+2\sum_{i\leq\ell}(d_i-1)\Big) \\ &+ 4\log(\ell+3)\sum_{i\leq\ell+1} d_i \\ &+ 3\log(2)\sum_{i\leq\ell} d_i(d_i-1). \end{aligned}$$

Summing on all α and using inequality $\mathbf{A_5}$ as above, we conclude the proof of Lemma 2.10.

As a corollary, it would be interesting to compare the bounds obtained for $N_1, \ldots, N_n$ above and for the Kronecker representation, in the specific situation where X_1 is separating. This problem has already been mentioned in introduction of chapter, in Equations (2.2) and (2.3). In fact, there is a the strong analogy between, in one hand, Formula (1.2) page 16 and the formula for $T_{\ell+1}$ in Proposition 2.5 and in the other hand between formulas of Corollary 2.3, and Corollary 1.1, page 17.

Corollary 2.4. *Let $V \subset \mathbb{A}^n_{\bar{K}}$ a zero-dimensional variety with vanishing ideal is defined over K and verified the Separability Assumption. Let u be a separating linear form for V and $(\chi_u, w_1, \ldots, w_n)$ the associated Kronecker representation. Define the polynomials $N_1, \ldots, N_{n+1}$ as follows:*

$$\left|\begin{array}{l} N_1(X_1) = \chi_u(X_1) \\ N_2(X_1, X_2) = \chi'_u(X_1)X_2 - w_2(X_1) \\ \quad \vdots \\ N_n(X_1, \ldots, X_n) = \chi'_u(X_1)X_n - w_{n-1}(X_1) \\ N_{n+1}(X_1, \ldots, X_{n+1}) = \chi'_u(X_1)X_{n+1} - w_n(X_1) \end{array}\right. ,$$

The family $\{N_i\}_{1\leq i\leq n+1}$ verifies the equality in Corollary 2.3. For $2 \leq i \leq n+1$, the equality $\max\{h(\chi'_u), h(w_{i-1})\} = h(N_i)$ holds. Moreover, the bounds obtained for polynomials $N_2, \ldots, N_{n+1}$ (Theorem 2.5) applied to polynomials $w_1, \ldots, w_n$ are equal, modulo negligible logarithmic terms, to the bounds obtained for the Kronecker representation (Theorem 2.2). The reciprocal is also true.

Proof. The first point is a consequence of Equations (2.2) page 49. We apply to them the formula of Defintion 2.3 page 82, and we obtain the polynomials χ'_u and w_i of the Kronecker representation. The equality $h(N_i) = \max\{h(\chi'_u), h(w_{i-1})\}$ then follows from the definition of height of a polynomial. Let us prove the estimate of the last point of the corollary. Let $V' \subset \mathbb{A}^{n+1}_{\bar{K}}$ the zero-set of $N_1, \ldots, N_{n+1}$. As $N_1(X_1) = \chi_u(X_1)$, it follows that $\deg_{X_1}(N_1) = \deg(V)$. As the others polynomials $N_1, \ldots, N_{n+1}$ have degree one, it follows that $\deg_{X_1}(N_1) = \deg(V')$ also, and $\deg(\pi_i^{n+1}(V') = \deg(V') = \deg(V)$. Let us denote by D this common degree. The Chow form of V' verifies:

$$\mathcal{C}_{V'} = \prod_{\alpha\in V} (T - u(\alpha)U_1 - \alpha_1 U_1 \cdots - \alpha_n U_{n+1}).$$

$$h_v(T - u(\alpha)U_1 - \alpha_1 U_2 - \cdots - \alpha_n U_{n+1}) = \log\max\{1, |u(\alpha)|_v, |\alpha_1|_v, \ldots, |\alpha_n|_v\}.$$

Let us write $u = u_1X_1 + \cdots + u_nX_n$.

$$|u(\alpha)|_v \leq \begin{cases} \sum_{i=1}^n |u_i|_v \cdot |\alpha_i|_v \leq n \left(\max\limits_{1\leq i\leq n} |u_i|_v\right)\left(\max\limits_{1\leq i\leq n} |\alpha_i|_v\right) & \text{if } v \text{ is Archimedean} \\ \max\limits_{1\leq i\leq n} |u_i|_v \cdot |\alpha_i|_v \leq \left(\max\limits_{1\leq i\leq n} |u_i|_v\right)\left(\max\limits_{1\leq i\leq n} |\alpha_i|_v\right), & \text{if } v \text{ is non-Archimedean.} \end{cases}$$

It follows that

$$\log\max\{1,|u(\alpha)|_v,|\alpha_1|_v,\ldots,|\alpha_n|_v\}\leq\begin{cases}\log\max\{1,\max\limits_{1\leq i\leq n}\{|u_i|_v\}\}+\log\max\{1,\max\limits_{1\leq i\leq n}\{|\alpha_i|_v\}\}\\ \qquad\qquad+\log(n),\quad\text{if } v \text{ is Archimedean}\\ \log\max\{1,\max\limits_{1\leq i\leq n}\{|u_i|_v\}\}+\log\max\{1,\max\limits_{1\leq i\leq n}\{|\alpha_i|_v\}\},\\ \qquad\qquad\text{if } v \text{ is non-Archimedean.}\end{cases}$$

By definition of the height of a polynomial, we deduce that:

$$h_v(T-u(\alpha)U_1-\alpha_1U_2-\cdots-\alpha_nU_{n+1})\leq\begin{cases}h_v(u)+h_v(T-\alpha_1U_1-\cdots-\alpha_nU_n)+\log(n),\\ \qquad\qquad\text{if } v \text{ is Archimedean}\\ h_v(u)+h_v(T-\alpha_1U_1-\cdots-\alpha_nU_n),\\ \qquad\qquad\text{if } v \text{ is non-Archimedean.}\end{cases}$$

Suppose now that v is a non-Archimedean absolute value in M_K^0. Then inequality $\mathbf{A}_3$ gives:

$$\begin{aligned}h_v(\mathcal{C}_{V'}) &\leq \sum_{\alpha\in V}h_v(T-u(\alpha)U_1-\alpha_1U_2-\cdots-\alpha_nX_{n+1})+D\log(n+3)\\ &\leq \sum_{\alpha\in V}h_v(u)+\log(n)+h_v(T-\alpha_1U_1-\cdots-\alpha_nU_n\\ \text{From } \mathbf{A}_4: &\leq D(h_v(u)+\log(n))+h_v(\mathcal{C}_V)+2D\log(n+2)\\ &\leq D\left(h_v(u)+3\log(n+2)\right)+h_v(\mathcal{C}_V).\end{aligned}$$

Suppose now that $v\in M_K^\infty$ is non-Archimedean. equality $\mathbf{N}_1$ yields:

$$\begin{aligned}h_v(\mathcal{C}_{V'}) &= \sum_{\alpha\in V}h_v(T-u(\alpha)U_1-\alpha_1U_2-\cdots-\alpha_nX_{n+1})\\ &\leq \sum_{\alpha\in V}h_v(u)+h_v(T-\alpha_1U_1-\cdots-\alpha_nU_n)\\ \text{From Equality } \mathbf{N}_1: &\leq Dh_v(u)+h_v(\mathcal{C}_V)\end{aligned}$$

Using the definition of the height of a variety, and Corollary 1.2 for the number field case, leads to:

$$h(V')\leq\begin{cases}h(V)+Dh(u), & \text{if } K \text{ is a function field}\\ h(V)+Dh(u)+4\log(n+2), & \text{if } K \text{ is a number field.}\end{cases}$$

Let us apply to w_n the bounds proved for N_{n+1} in Theorem 2.5. We get:

$$\begin{aligned}h(w_n) &\leq \begin{cases}h(V')+5\log(n+3)\deg(V), & \text{(numbers case)}\\ h(V'), & \text{(functional case),}\end{cases}\\ &\leq \begin{cases}h(V)+Dh(u)+4\log(n+2)+5D\log(n+3), & \text{(numbers case)}\\ h(V)+Dh(u), & \text{(functional case).}\end{cases}\end{aligned}$$

As foreseen, minor changes in the logarithmic terms and we recognize the bounds obtained in Theorem 2.2 for the Kronecker representation.

Conversely, Llet us estimate the Chow form of V in function of the Chow form of V'. It suffices to notice that

$$\mathcal{C}_{V'}(0, U_1, \ldots, U_n, T) = \mathcal{C}_V(U_1, \ldots, U_n, T).$$

Specializing a variable to zero lowers the height, and also the Mahler measure from inequality $\mathbf{A}_8$. Thus, for any absolute value $v \in M_K$, $h_v(\mathcal{C}_V) \leq h_v(\mathcal{C}_{V'})$ and that $m(\sigma_v(\mathcal{C}_V)) \leq m(\sigma_v(\mathcal{C}_{V'})$. With Corollary 1.2, it follows that:

$$h(V) \leq \begin{cases} h(V') & \text{(functional case)} \\ h(V') + D\log(n+2) & \text{(number field case).} \end{cases}$$

we notice that $N_1, N_2, \ldots, N_{n+1}$ is a primitive element representation *à la* Kronecker for V where X_1 is a separating linear form, whose minimal polynomial is N_1. Then applying the estimates obtained for the polynomials of the Kronecker representation in Theorem 2.2 page 2.2 leads to:

$$\begin{aligned} h(N_{n+1}) &\leq \begin{cases} h(V) + Dh(X_1) + D\log(n+2) + (n+1)\log(D), & \text{(number field case)} \\ h(V) + Dh(X_1), & \text{(functional case).} \end{cases} \\ &\leq \begin{cases} h(V) + D(2 + \log(n+2)) + (n+1)\log(D), & \text{(number field case)} \\ h(V) + D, & \text{(functional case).} \end{cases} \end{aligned}$$

We notice that hese bounds are similar to the ones obtained in Theoerem 2.5 page 75 for polynomial $N_1, \ldots, N_{n+1}$. □

Chapter 3

Change of order for regular chains in positive dimension

We discuss changing the variable order for a regular chain in positive dimension. This quite general question has applications going from implicitization problems to the symbolic resolution of some systems of differential algebraic equations.

We propose a modular method, reducing the problem to computations in dimension zero and one. The problems raised by the choice of the specialization points and the lack of the (crucial) information of what are the free and algebraic variables for the new order are discussed. Strong (but not unusual) hypotheses for the initial regular chain are required; the main required subroutines are change of order in dimension zero and a formal Newton iteration.

This is a joint work with Xin Jin, Marc Moreno Maza, and Éric Schost. An implementation of this algorithm is available in Maple 11, inside the `RegularChains` library; I have not contributed to this program, only due to the other three authors.

3.1 Introduction

Many operations with multivariate polynomials, such as implicitization, rely on manipulations involving one or several lexicographic orders. These lexicographic orders are also a key component to define *regular chains* (see definition below) [63, 88, 80], so that these regular chains appear as a natural tool to handle situations where orders on the variables matter.

Explicitly, suppose that we are given a regular chain for some input order, as well as a *target* order on the variables; we are interested in converting the input into a new regular chain with respect to the target order, while describing the same (generic) solutions. This is required by many applications (the implicitization problem falls into this category), as in the following example.

Example. Consider the polynomials P in $\mathbb{Q}[X_1, X_2]$ such that $P(X_1, X_2) = P(-X_1, -X_2)$. Invariant theory tells us that any such polynomial can be written as a polynomial in X_1^2, X_2^2 (the *primary* invariants P_1 and P_2) and X_1X_2 (the *secondary* invariant S); natural questions to ask are whether such a representation is unique, and how to perform the rewriting.

This can be done by getting an expression of X_1 and X_2 in function of P_1 and P_2, hence by changing the order of the following system from $X_2 < X_1 < S < P_2 < P_1$ to

$P_2 < P_1 < S < X_1 < X_2$. Given

$$\left|\begin{array}{lcl} P_1 & = & X_1^2 \\ P_2 & = & X_2^2 \\ S & = & X_1X_2 \end{array}\right. \quad \text{or equivalently} \quad \left|\begin{array}{lcl} P_1 - X_1^2 & = & 0 \\ P_2 - X_2^2 & = & 0 \\ S - X_1X_2 & = & 0, \end{array}\right.$$

we wish to obtain

$$\left|\begin{array}{lcl} SX_2 - P_1X_1 & = & 0 \\ X_1^2 - P_1 & = & 0 \\ S^2 - P_1P_2 & = & 0 \end{array}\right. \quad \text{or equivalently} \quad \left|\begin{array}{lcl} X_2 & = & \frac{S}{P_1}X_1 \\ X_1^2 & = & P_1 \\ S^2 & = & P_1P_2. \end{array}\right.$$

In this form, we observe the relation $S^2 = P_1P_2$ between our basic invariants, which establishes that the representation cannot be unique. Furthermore, the new form of the system can be used as a set of rewriting rules, so as to obtain a canonical form for any invariant polynomial.

In this article, we present an algorithm for performing such conversions, concentrating on the case of varieties of positive dimension. Representing such a variety by a regular chain then involves decomposing the set of coordinates into free / algebraic variables; for instance, in the input of the previous algorithm, (X_1, X_2) are free and (P_1, P_2, S) algebraic. We will then use modular techniques (consisting in "specializing" and "lifting" the free variables) to keep the size of intermediate expressions involving the free variables under control.

To get a hint of the way such techniques work, one can consider the over-simplified case where the free (resp. algebraic) variables are the same for both the input and the target order (this is not the case in the previous example), so that only the order of the algebraic variables actually matters. In this case, a direct approach consists in specializing the free variables at a random value (thus reducing to dimension zero), use change of order in dimension zero to operate on the algebraic variables, and recover the dependence in the free variables using a formal version of Newton iteration (Figure 3.1).

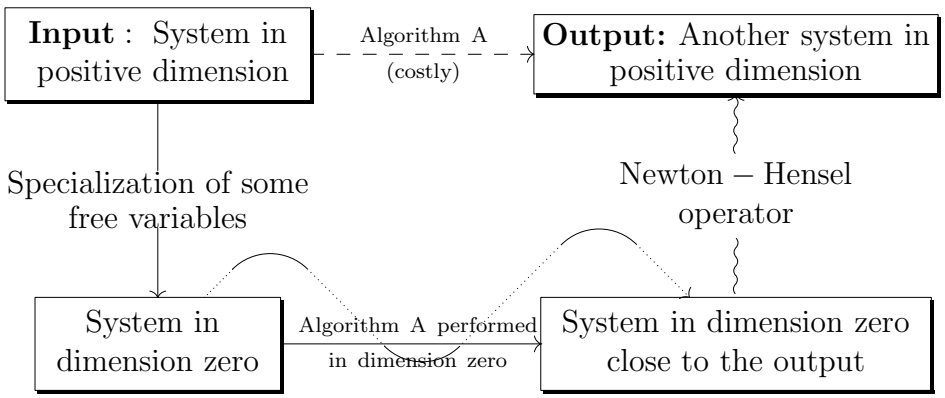

Figure 3.1: Prototype of a modular method using Newton-Hensel technique

We will extend this approach to the general case, where the sets of free (resp. algebraic) variables differ in the input and output. Of course, we do not know *a priori* what the free (resp. algebraic) variables are in the output, so they will have to be determined; using this information will enable us to design a fully modular algorithm.

Representing varieties by regular chains. After this general introduction, we can define more formally the objects we will compute with. To start with, let us consider a family $\mathbf{X} = (X_1, \ldots, X_n)$ of indeterminates over a *perfect* field $\mathbb{K}$, and suppose that these variables are ordered. In this paragraph, our order will simply be $X_1 < \cdots < X_n$, a situation to which one can always reduce at the cost of renaming the variables. We refer to Section 1.1.3 in the preliminary chapter for the definitions and basic properties of regular chains and triangular sets. We recall that they are all *Lazard* triangular sets, as along this thesis.

Given a variety W, what are the regular chains $\mathbf{R}$ such that $W = V(\mathsf{Sat}(\mathbf{R}))$? In what follows, we will let $W \subset \overline{\mathbb{K}}^n$ be an *irreducible* variety of dimension r, defined over $\mathbb{K}$, and we let I be its defining ideal in $\mathbb{K}[\mathbf{X}]$. Since we make a heavy use of projections, we use a special notation: if $\mathbf{Z}$ is a subset of $\mathbf{X}$ of cardinality ℓ, we denote by $\pi_{\mathbf{Z}} : \overline{\mathbb{K}}^n \to \overline{\mathbb{K}}^\ell$ the projection on the $\mathbf{Z}$-space, that forgets all coordinates not in $\mathbf{Z}$. For $\mathbf{z}$ in $\overline{\mathbb{K}}^\ell$, we then denote by $W_{\mathbf{z}}$ the fiber $W \cap \pi_{\mathbf{Z}}^{-1}(\mathbf{z})$, that is, the subset of points of W that project onto $\mathbf{z}$.

A subset $\mathbf{Z}$ of $\mathbf{X}$ is then a set of *free* variables for W if $I \cap \mathbb{K}[\mathbf{Z}] = \{0\}$, *i.e.* if the image $\pi_{\mathbf{Z}}(W)$ is dense. If $\mathbf{Z}$ is such a set of free variables, it is then called *maximal* if it is additionally maximal (for inclusion) among the sets of free variables; in this case, for a generic choice of $\mathbf{z}$, the fiber $W_{\mathbf{z}}$ has dimension zero. Theorem 1.2 p. 18, shows that given a maximal set $\mathbf{Z}$ of free variables of the irreducible variety W, there exists a regular chain having $\mathbf{Z}$ for free variables, and $\mathbf{X} - \mathbf{Z}$ for algebraic variables. There is no unicity, the first reason being that we have not specified the variable order. But even then, there is *a priori* no canonical choice, due to the possible choices of initials. The following proposition restores canonicity, by introducing a normal form for these initials.

Proposition 3.1. *Let $<$ be an order on $\mathbf{X}$. Then all regular chains $\mathbf{R}$ for the order $<$ for which $I = \mathsf{Sat}(\mathbf{R})$ have the same set of algebraic variables $\mathbf{Y}$ (resp. free variables $\mathbf{Z}$). Furthermore, there exists a unique triangular set $\mathbf{T}$ in $\mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ for the order induced by $<$ on $\mathbf{Y}$ such that $(\mathbf{T}) = I \cdot \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$.*

In the situation of the previous proposition, $\mathbf{T}$ represents the *generic points* of W. If we clean all denominators from $\mathbf{T}$, we obtain a regular chain $\mathbf{R}$ in $\mathbb{K}[\mathbf{Z}][\mathbf{Y}] = \mathbb{K}[\mathbf{X}]$, having all its initials in $\mathbb{K}[\mathbf{Z}]$ and such that $\mathsf{Sat}(\mathbf{R}) = I$ (this regular chain is called *strongly normalized* in [79]). We will call $\mathbf{T}$ and $\mathbf{R}$ the *canonical representations* associated to the order $<$.

Lifting fibers. As usual in this kind of situation, one has to be careful to avoid a combinatorial explosion due to the sheer number of monomials that may appear in representations such as $\mathbf{T}$ or $\mathbf{R}$ mentioned above.

A natural measure of the complexity of the problem is the *degree* of the variety W (see [55], from where we take all our results on this notion). Now, if W has arbitrary positive dimension, the number of monomials that can appear in $\mathbf{T}$ or $\mathbf{R}$ is *not* polynomial in the degree of W. To overcome this difficulty, we use *lifting fibers* [52, 77]: an irreducible variety W of dimension r will be represented by a specialization of the associated canonical representation $\mathbf{T}$ at some point $\mathbf{z} \in \mathbb{K}^r$, thus describing a fiber $W_{\mathbf{z}}$ of some projection $\pi_{\mathbf{Z}}(W)$.

Precisely, let $<$ be an order on the set $\mathbf{X}$. Associated with this order, let the set of free variables $\mathbf{Z}$, its complement $\mathbf{Y} = \mathbf{X} - \mathbf{Z}$, and the canonical representation $\mathbf{T} \in \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ be as

in Proposition 3.1. We will then put natural non-degeneracy conditions on our specialization point $\mathbf{z} \in \mathbb{K}^r$.

H_1. The point $\mathbf{z} \in \mathbb{K}^r$ cancels no denominator in $\mathbf{T}$.

In this case, we denote by $\mathbf{T}_\mathbf{z}$ the triangular set in $\mathbb{K}[\mathbf{Y}]$ obtained by specializing $\mathbf{Z}$ at $\mathbf{z}$ in $\mathbf{T}$. Even under condition H_1, $\mathbf{T}_\mathbf{z}$ does not necessarily represent the fiber $W_\mathbf{z}$; we thus take it as an assumption.

H_2. The fiber $W_\mathbf{z} = W \cap \pi_\mathbf{Z}^{-1}(\mathbf{z})$ equals $\{\mathbf{z}\} \times V(\mathbf{T}_\mathbf{z})$; in other words, the roots of $\mathbf{T}_\mathbf{z}$ are the points of W above $\mathbf{z}$.

H_3. The triangular system $\mathbf{T}_\mathbf{z}$ defines a radical ideal (hence is a triangular set).

Finally, we need a system of equations to recover W from the fiber $W_\mathbf{z}$. In our case, we will be given a system of equations $\mathbf{F} = F_1, \ldots, F_s$ and an inequation h in $\mathbb{K}[\mathbf{X}]$ such that W is the Zariski-closure of $V(\mathbf{F}) - V(h)$ (later, $\mathbf{F}$ will be our input regular chain, and h the product of its initials). We also require that the conditions of the implicit function theorem are satisfied:

H_4. The Jacobian determinant of $\mathbf{F}$ with respect to $\mathbf{Y}$ does not vanish on $W_\mathbf{z}$.

Then, a *lifting fiber* for $(\mathbf{F}, h, <)$ is the data of $\mathbf{z}$ and $\mathbf{T}_\mathbf{z}$ satisfying assumptions $\mathsf{H}_1, \ldots, \mathsf{H}_4$. Using Newton iteration, if needed, one can then recover the canonical representation $\mathbf{T} \in \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ from such a lifting fiber, see Proposition 3.7 below. The main interest of this notion is thus that it enables us to handle objects of dimension zero instead of positive dimension, avoiding the cost of representing all monomials in positive dimension, without losing any information.

Let us illustrate this notion on the invariant problem met before. Consider again the system of equations $\mathbf{F}$ over the field $\mathbb{K}$:

$$S - X_1X_2, \quad P_2 - X_2^2, \quad P_1 - X_1^2,$$

and let W be its zero-set in $\overline{\mathbb{K}}^5$, so that the inequation h is here 1. In this order, this family of polynomials is already a regular chain for the order $X_2 < X_1 < S < P_2 < P_1$, admitting $\mathbf{Z} = (X_1, X_2)$ as free variables. Then one checks that the point $\mathbf{z} = (1, 1)$ satisfies assumptions $\mathsf{H}_1, \ldots, \mathsf{H}_4$; the corresponding lifting fiber is given by $\mathbf{z}$, together with

$$\mathbf{T}_{(1,1)} \left| \begin{array}{l} P_1 - 1 \\ P_2 - 1 \\ S - 1 \end{array} \right. \quad \text{which is a specialization of} \quad \mathbf{T} \left| \begin{array}{l} P_1 - X_1^2 \\ P_2 - X_2^2 \\ S - X_1X_2. \end{array} \right.$$

Observe next that $\mathbf{Z}' = (P_1, P_2)$ is also a maximal set of free variables. Using the order $P_2 < P_1 < S < X_1 < X_2$, one checks that the point $\mathbf{z}' = (1, 1)$ satisfies assumptions $\mathsf{H}_1, \ldots, \mathsf{H}_4$ as well; the corresponding lifting fiber is given by $\mathbf{z}'$, together with

$$\mathbf{T}'_{(1,1)} \left| \begin{array}{l} X_2 - SX_1 \\ X_1^2 - 1 \\ S^2 - 1 \end{array} \right. \quad \text{which is a specialization of} \quad \mathbf{T}' \left| \begin{array}{l} X_2 - \frac{S}{P_1}X_1 \\ X_1^2 - P_1 \\ S^2 - P_1P_2. \end{array} \right.$$

Lifting fibers are defined using variable orders. However, to have more notational flexibility in what follows, we also associate a notion of lifting fiber to a given set of free variables $\mathbf{Z}$ (resp. a set of algebraic variables $\mathbf{Y}$): this is a lifting fiber for $(\mathbf{F}, h, <)$, where $<$ is any order inducing $\mathbf{Z}$ as free variables for W (resp. $\mathbf{Y}$ as algebraic variables).

Main results. In what follows, we denote by MT a function that assigns to an irreducible variety W an upper bound on the cost of all operations $(+, -, \times)$, invertibility testing and inversion modulo triangular sets arising as lifting fibers for W. The precise definition is given in Subsections 1.3.2 and 3.2.2, together with various estimates; in the meantime, we point out that $\mathsf{MT}(W)$ is *polynomial* in the degree $(\deg W)$ of W. We also denote by M a *multiplication time* function for univariate polynomials, see again Subsection 1.3.2.

Given an input regular chain and a target order, our main result is then a polynomial-time bound on the complexity of computing a lifting fiber for the output regular chain. Since our algorithms use Newton iteration, a natural encoding for the input system is through a *straight-line program*, as this representation is especially well adapted to such evaluation-intensive routines. The counterpart of this representation is that it does not immediately give information such as total or partial degrees, which are needed below; while it would be possible to determine these quantities at some extra cost, we adopt the simpler solution of taking them as input.

Theorem 3.1. *Let $\mathbf{F} = (F_1, \ldots, F_s)$ be a regular chain in $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_n]$ for an input order $<$, and assume that the following assumptions hold:*

- *The characteristic of $\mathbb{K}$ is larger than d^n, where d is an upper bound on the degrees of the polynomials in $\mathbf{F}$.*
- *The saturated ideal of $\mathbf{F}$ is prime.*

Let $W = V(\mathsf{Sat}(\mathbf{F}))$ and let h be the product of the initials of $\mathbf{F}$. Suppose also that the regular chain $\mathbf{F}$ is given by a straight-line program of size L, that the main variables of $\mathbf{F}$ are known, as well as the degree of these polynomials in their main variables.

Given a target order $<'$ on $\mathbf{X}$, one can compute by a probabilistic algorithm a lifting fiber for $(\mathbf{F}, h, <')$. In case of success, the algorithm uses

$$O\left(s(n^4 + nL)\,\mathsf{MT}(W)\,\mathsf{M}\big((\deg W)^2\big)\,\log(\deg W)\right) \subset (nL \deg W)^{O(1)}$$

operations in $\mathbb{K}$. The algorithm chooses $n + s$ parameters in $\mathbb{K}$. If these parameters are chosen uniformly at random in a finite subset S of $\mathbb{K}$, writing $m = \max(n, d)$, the probability of failure is at most

$$\frac{2d^n(3d^{2n} + n2^n + (6 + 13m)md^n + m^2)}{|S|}.$$

Let us illustrate the probabilistic aspect by the example of a system with $n = 10$ unknowns, with input equations of maximal degree $d = 4$, solved over a finite field $\mathbb{K}$ with approximately 10^{19} elements (so that the field elements fit into a 64-bit word). Then if one chooses all random values in $\mathbb{K}$, by the previous theorem, the probability of failure is at most $\simeq 6 \cdot 10^{-7}$.

As was mentioned before, from our output lifting fiber, recovering the full expansion of the target regular chain is a well-known question, that is solved using again Newton iteration: for the sake of reference, the cost of this operation is reviewed in Proposition 3.7. However, one should bear in mind that in general, using dense monomial representation, the cost of this last step may be prohibitive due to the sheer number of monomials that may appear, which is not polynomial in the degree of W.

To conclude, we mention some workarounds to this issue. First, in several situations, knowing a single lifting fiber is actually enough: for instance, it enables one to recover any *other* lifting fiber efficiently (that is, in a time that remains polynomial in the degree of W). If the multivariate representation of the target regular chain is really required, then it can be computed in polynomial time using *straight-line program* encoding, following the ideas of [50, 49, 47, 57, 61]; however, as of now, there is no software package enabling easily such manipulations. Finally, when using dense representation, a direction of future research will consist in using *sparse lifting techniques*, taking into account the possible sparse nature of the output.

Outlook of the algorithm. The algorithm is an iterative process: the input regular chain provides us with a first lifting fiber, for the initial order. We will then compute a finite sequence of lifting fibers, the last one being a lifting fiber for the target order.

The algorithm works in two steps. As was said before, we do not know *a priori* what are the algebraic variables in the output; the first step of the algorithm will determine them. Since this will be required in the second stage of the algorithm, we will actually compute a more precise information: a whole *sequence* of sets of algebraic variables $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$, where $\mathbf{Y}_0$ is the set of algebraic variables in the input regular chain, and $\mathbf{Y}_s$ is that for the target regular chain. Writing $\mathbf{Y}_i$ for the set of algebraic variables at step i, we will then arrange that $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ differ by a single element. This will be done by linear algebra (with algebraic number coefficients), using a characterization of $\mathbf{Y}_s$ as the maximal element of a suitable *matroid*.

The second step consists in computing an associated sequence of lifting fibers. This is an inductive process: given a lifting fiber for $\mathbf{Y}_i$, we will deduce a lifting fiber for $\mathbf{Y}_{i+1}$. Our requirements on the sequence $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$ make this task easy, using change of order in dimension zero and Newton iteration in one variable. Hence, all the objects that we see will be either zero- or one-dimensional; this will allow us to keep a good control on the complexity.

Let us illustrate the behavior of this algorithm with our previous example (see Figure 3.2 for a visual explanation). The set of algebraic variables for the input regular chain is $\mathbf{Y}_0 = \{S, P_1, P_2\}$. In the first part of the algorithm, we will obtain the following sets of algebraic variables:

$$\mathbf{Y}_1 = \mathbf{Y}_0 - \{P_2\} \cup \{X_2\} = \{S, P_1, X_2\}$$
$$\mathbf{Y}_2 = \mathbf{Y}_1 - \{P_1\} \cup \{X_1\} = \{S, X_1, X_2\}.$$

In the second phase, we obtain the associated lifting fibers:

$$\left|\begin{array}{l} P_1 - 1 \\ P_2 - 1 \\ S - 1 \end{array}\right. \text{ with } (X_1 = 1, X_2 = 1) \qquad \left|\begin{array}{l} X_2 - S \\ S^2 - 1 \\ P_1^2 - 1 \end{array}\right. \text{ with } (X_1 = 1, P_2 = 1) \qquad \left|\begin{array}{l} X_2 - SX_1 \\ X_1^2 - 1 \\ S^2 - 1 \end{array}\right. \text{ with } (P_1 = 1, P_2 = 1),$$

the last one being the output of our algorithm.

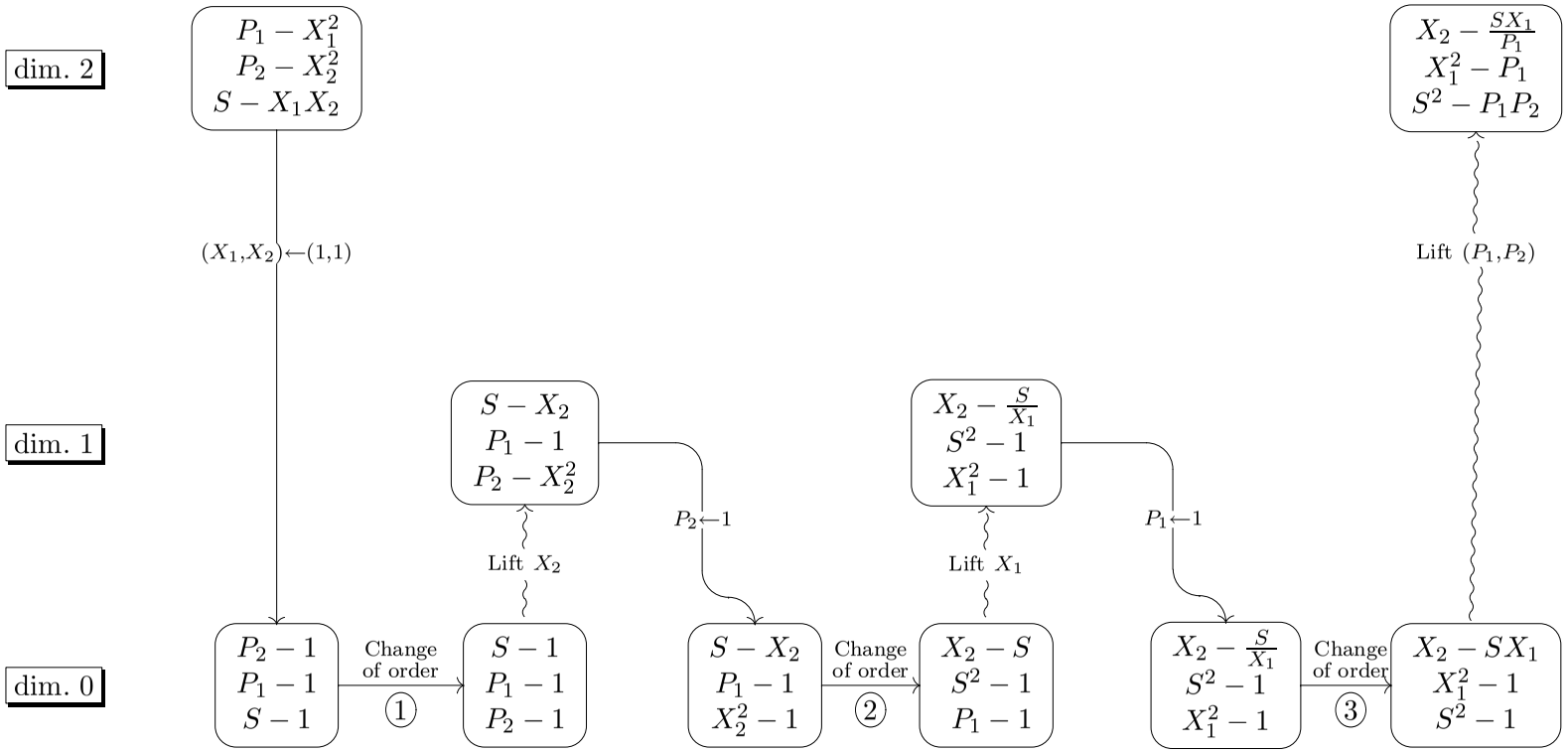

Figure 3.2: The main steps of the algorithm applied an example

The sequence of variables to specialize and lift is here equal to $\{(X_2, P_2), (X_1, P_1)\}$.

$$\mathbf{Y}_0 = (S, P_2, P_1) \ , \quad \mathbf{Y}_1 = (X_2, P_1, S) \ , \quad \mathbf{Y}_2 = (X_1, S, X_2).$$

① This change of order puts the variable P_2 at the smallest position for $\prec$. In fact, P_2 is the next variable to be specialized.

② For the same reason, since P_1 is the next variable to be specialized, P_1 becomes by this change of ordering the smallest variable.

③ This is the last change of ordering, hence there is no "next" variable to be specialized like above. The change of order is conducted by the order on the algebraic variables of the output system $S \prec X_1 \prec X_2$.

Applications. Change of order is an ubiquitous problem. A first vast family of applications is coming from *implicitization* problems, which essentially consist in finding the polynomial relations between several multivariate rational functions. This problem fits naturally in our setting: to a system of rational functions of the form

$$\varphi_i = \frac{f_i(Z_1, \ldots, Z_r)}{g_i(Z_1, \ldots, Z_r)} \quad i = 1, \ldots, s$$

one associates the regular chain

$$F_i : \quad g_i(Z_1, \ldots, Z_r)Y_i - f_i(Z_1, \ldots, Z_r) \quad i = 1, \ldots, s$$

having $\mathbf{Z} = Z_1, \ldots, Z_r$ as free variables and $\mathbf{Y} = Y_1, \ldots, Y_s$ as algebraic variables. Changing to an order where the $\mathbf{Z}$ variables are larger than the $\mathbf{Y}$ variables enables us to find the relations between the rational function φ_i, but also to recover the parameters $\mathbf{Z}$ as algebraic functions of the image points $\mathbf{Y}$ (when it is possible).

As was illustrated in the introductory example, several other families of problems fit into a similar setting, such as many questions coming from invariant theory, using the above "tag variables" techniques [107]. In all these cases, our primality assumption is indeed satisfied.

Several other application examples are coming from *differential algebra*: as illustrated in [20], characteristic sets conversion in a differential ring can partly be reduced to perform change of orders for positive-dimensional regular chains in a polynomial ring (see the example Euler's equations for a perfect fluid in [20]). Again, in this context, our primality assumption is satisfied.

Previous work. As was said above, the concept of regular chain was introduced in [63], following previous work initiated by Ritt [100] and Wu [120]. Other contributors were [71, 72], Aubry et al. [7] and Moreno Maza [88]; a recent overview is also given in [60, 59].

In this paper, we focus on the case of *positive* dimension. There already exist many algorithms to perform the change of order in this context, either under the point of view of Gröbner bases [28, 64, 113] or regular chains [20]. As was said above, an important application of change of order is the implicitization problem, for which many specialized algorithms have been developed, relying on resultant formalisms and homological algebra techniques, see for instance [25, 33, 29] and the numerous references therein.

However, as far as we know, the complexity of these algorithms is not well known, and in most cases, cannot be expected to be polynomial in the degree of W. Our specificity is to provide a fine algorithmic study, relying on a few well-identified subroutines, such as change of order in dimension zero, and Newton iteration. This enables us to offer a clear view of the complexity of the problem: the central operation presented in this article, computing a lifting fiber for the target regular chain, can be done in a time that is polynomial in the natural complexity measures of the problem. Recovering the full monomial expansion of the target regular chain can then be done using standard techniques.

This notion of lifting fiber (though not exactly with the same requirements as ours) explicitly appeared in [52, 77], following extensive previous work of Giusti, Heintz, Pardo and collaborators [50, 49, 47], with the purpose of computing *geometric resolutions*. A similar idea appeared again in the context of *numerical algebraic geometry*, with the name of *witness sets* [110].

Linked with the notion of lifting fiber, other aspects of this work are following the ideas of the references [50, 49, 47, 52, 77] cited above. Besides the use of straight-line programs and of Newton iteration, the approach used in the second part of our algorithm bears some similarity with the above works in its iterative lifting / intersection process. However, in our case, we obtain finer complexity estimates and a sharp control on the probabilistic aspects: our algorithm is polynomial in the degree of the variety defined by the input system $\mathbf{F}$, whereas none of the above methods is known to reach this bound.

Organization of the chapter. Section 3.2 adds to Subsections 1.1.3 and 1.3.2 some basic geometric and algorithmic results on regular chains that are used throughout this article. Section 3.3 then introduces the language of *matroids* as a convenient tool to describe independence properties: this will give a general framework for us to design the latter algorithms. Using this language, in Section 3.4, we use linear algebra to determine the set of algebraic variables that appear in the target regular chain. Section 3.5 shows how to use that information to compute a sequence of lifting fibers, and Section 3.6 gives the proof of the main theorem. We finish this article with a conclusion section, and an appendix devoted to the computation of inverses modulo a triangular set.

3.2 Preliminaries

This section goes into detail the properties of regular chains (addressed in Subsection 1.1.3) and related algorithmic questions (addressed in Subsection 1.3.2) Many of those are already known; a few new facts are introduced here. In all that follows, $\mathbb{K}$ is a perfect field.

3.2.1 Additional results on regular chains

We start by discussing some properties of regular chains in dimension zero. Following Theorems 1.2 and 1.3, if W is an irreducible zero-dimensional variety defined over $\mathbb{K}$, then for any order $<$ on the variables, there exists a unique triangular set $\mathbf{T}$ for the order $<$ such that $(\mathbf{T})$ equals the generating ideal $I(W)$ of W; this triangular set is the Gröbner basis of $I(W)$ for the lexicographic order induced by $<$.

When W is not irreducible, this does not have to be the case anymore: $I(W)$ is generated by a triangular set for the order $<$ if and only if W is *equiprojectable* for a suitable family of projections [9]. In what follows, our zero-dimensional objects will be obtained as sections of irreducible varieties of positive dimension. Using generic sections will ensure that equiprojectability holds.

We next discuss Proposition 3.1, whose statement is the following: *Let $<$ be an order on $\mathbf{X}$. Then all regular chains $\mathbf{R}$ for the order $<$ for which $I = \mathsf{Sat}(\mathbf{R})$ have the same set of algebraic variables $\mathbf{Y}$ (resp. free variables $\mathbf{Z}$). Furthermore, there exists a unique triangular set $\mathbf{T}$ in $\mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ for the order induced by $<$ on $\mathbf{Y}$ such that $(\mathbf{T}) = I \cdot \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$.*

The first point will be proved in Proposition 3.10, where we actually give a more precise statement. To obtain the second part of the proposition, we establish some more precise results, needed later on.

Lemma 3.1. *Let* $\mathbf{Z}$ *be a maximal set of free variables for* W *and let* $\mathbf{Y} = \mathbf{X} - \mathbf{Z}$. *Then,* $\mathbb{K}(W) \simeq \mathbb{K}(\mathbf{Z})[\mathbf{Y}]/I$, *and the extension* $\mathbb{K}(\mathbf{Z}) \to \mathbb{K}(W)$ *is finite. If the characteristic of* $\mathbb{K}$ *is larger than* $\deg(W)$, *then this extension is separable.*

PROOF: Since I contains no polynomial in $\mathbb{K}[\mathbf{Z}]$, one checks that $I \cdot \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ is still prime, and the isomorphism $\mathbb{K}(W) \simeq \mathbb{K}(\mathbf{Z})[\mathbf{Y}]/I$ follows easily. We next show that $\mathbb{K}(\mathbf{Z}) \to \mathbb{K}(W)$ is finite and separable. Let Y thus be in $\mathbf{Y}$. Since $\mathbf{Z} + \{Y\}$ is not free, there exists a non-zero polynomial P_Y in $I \cap \mathbb{K}[\mathbf{Z}, Y]$, of degree at most $(\deg W)$. Note that P_Y does not reduce modulo I to a polynomial in $\mathbb{K}[\mathbf{Z}]$, since $\mathbf{Z}$ are free variables. Hence $Y \in \mathbb{K}(W)$ is algebraic over $\mathbb{K}(\mathbf{Z})$. Furthermore, if $\mathrm{char}(\mathbb{K}) > (\deg W) \geq \deg_Y P_Y$, Y is separable over $\mathbb{K}(\mathbf{Z})$, so our claim follows. □

Observe now that the second point in Proposition 3.1 is an immediate consequence of this lemma, in view of the previous discussion on triangular sets for zero-dimensional varieties.

Quantifying degeneracies. We will need two different statements regarding the degeneracies of specializations. The first result will be used to control the degeneracies in the input regular chain $\mathbf{F}$ of our main algorithm. The second statement will be used to control degeneracies attached to the intermediate and output regular chains, which feature stronger properties (e.g., they are strongly normalized), but with a looser control on the degrees.

Proposition 3.2. *Let* $\mathbf{F} = (F_1, \ldots, F_s)$ *be a regular chain in* $\mathbb{K}[\mathbf{X}]$, *let* W *be the zero-set of* $\mathsf{Sat}(\mathbf{F})$ *and let* $r = n - s$. *Let* $\mathbf{Z}$ *be the free of variables of* $\mathbf{F}$, *and let* $\mathbf{Y} = \mathbf{X} - \mathbf{Z}$ *be its algebraic variables, so that* Y_i *is the main variable of* F_i. *Suppose that* W *is irreducible and that the Jacobian determinant* σ *of* $\mathbf{F}$ *with respect to* $\mathbf{Y}$, *given by*

$$\sigma = \prod_{1 \leq i \leq s} \frac{\partial F_i}{\partial Y_i},$$

does not vanish identically on W. *Let finally* d *be a bound on the degrees of the polynomials in* $\mathbf{F}$.

There exists a non-zero polynomial $\Delta_{\mathrm{reg}} \in \mathbb{K}[\mathbf{Z}]$ *of degree at most* $2sd^{n+1}$ *with the following property. For* $\mathbf{z} \in \mathbb{K}^r$, *if* $\Delta_{\mathrm{reg}}(\mathbf{z})$ *is not zero, then* $\mathbf{F}_{\mathbf{z}} = \mathbf{F}(\mathbf{z}, \mathbf{Y})$ *is a regular chain in* $\mathbb{K}[\mathbf{Y}]$ *and defines a radical ideal.*

PROOF: Let V be the zero-set of $\mathbf{F}$; for $i \leq s$, let us denote by h_i the initial of F_i and let $h \in \mathbb{K}[\mathbf{X}]$ be the product $h_1 \cdots h_s$. We start by a lemma.

Lemma 3.2. *The projection* $\pi_{\mathbf{Z}}(V \cap V(h))$ *has dimension less than* r.

PROOF: The intersection $V \cap V(h)$ can be rewritten as

$$\big(V_0 \cap V(h_1)\big) \ \cup \ \big(V_1 \cap V(h_2)\big) \ \cup \ \cdots \ \cup \ \big(V_{s-1} \cap V(h_s)\big)$$

where V_i is the Zariski closure of $V - V(h_1 \cdots h_i)$. Let us denote by W_i the Zariski-closure of $V(F_1, \ldots, F_i) - V(h_1 \cdots h_i)$ in $\overline{\mathbb{K}}^{r+i}$. Since $\mathbf{F}$ is a regular chain, $W_i \cap V(h_{i+1})$ has dimension less than r, so that its projection on the $\mathbf{Z}$-space has dimension less than r as well. This implies that $V_i \cap V(h_{i+1})$ satisfies the same property. □

Let us return to the proof of the proposition. By Bézout's inequality [55], $V \cap V(h)$ has degree at most $(\deg V)(\deg h) \leq d^n \times sd = sd^{n+1}$; by the previous lemma, its image through

$\pi_{\mathbf{Z}}$ has dimension less than r. Hence, there exists a non-zero polynomial Δ_1 of degree at most sd^{n+1} such that if $\mathbf{z} \in \mathbb{K}^r$ does not cancel Δ_1, $h(\mathbf{z}, \mathbf{Y})$ vanishes nowhere on $V(\mathbb{F}_{\mathbf{z}})$. Hence, each $h_i(\mathbf{z}, \mathbf{Y})$ is a non zero-divizor modulo $(F_1(\mathbf{z}, \mathbf{Y}), \ldots, F_{i-1}(\mathbf{z}, \mathbf{Y}))$. For such a value of $\mathbf{z}$, $\mathbf{F}_{\mathbf{z}}$ is a regular chain and the fiber $W_{\mathbf{z}}$ equals $\{\mathbf{z}\} \times V(\mathbf{F}_{\mathbf{z}})$.

We then deal with the zeros of the polynomial σ. By assumption, $W \cap V(\sigma)$ has dimension less than r; by Bézout's inequality, its degree is at most sd^{n+1}. Hence, there exists a non-zero polynomial Δ_2 of degree at most sd^{n+1} such that if $\mathbf{z} \in \mathbb{K}^r$ does not cancel Δ_2, $\sigma(\mathbf{z}, \mathbf{Y})$ vanishes nowhere on $V(\mathbf{F}_{\mathbf{z}})$; in this case, $\mathbf{F}_{\mathbf{z}}$ defines a radical ideal, by the Jacobian criterion. To conclude, it suffices to take $\Delta_{\text{reg}} = \Delta_1 \Delta_2$. □

We next address the degeneracies that may occur in the latter stages of the algorithm. We thus still consider the input regular chain $\mathbf{F}$ in $\mathbb{K}[\mathbf{X}]$, the product h of its initials, and the variety $W = V(\mathsf{Sat}(\mathbf{F}))$ of dimension r; we assume that $\mathsf{Sat}(\mathbf{F})$ is prime. Let next $<$ be an order on the set $\mathbf{X}$ (not necessarily the order associated with $\mathbf{F}$), and let the sets of variables $(\mathbf{Z}, \mathbf{Y})$ and the canonical representation $\mathbf{T} \in \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ be associated to the order $<$ by Proposition 3.1. The following proposition quantifies the specializations $\mathbf{z} \in \mathbb{K}^r$ of $\mathbf{Z}$ that do not yield lifting fibers for $(\mathbf{F}, h, <)$.

Proposition 3.3. *Suppose that all polynomials in* $\mathbf{F}$ *have degree bounded by* d*, and that the Jacobian determinant of* $\mathbf{F}$ *with respect to* $\mathbf{Y}$ *does not vanish identically on* W*. Then there exists a non-zero polynomial* $\Delta_{\text{lift}} \in \mathbb{K}[\mathbf{Z}]$ *of degree at most* $nd^n(3d^n + n + d)$ *such that for* $\mathbf{z} \in \mathbb{K}^r$*, if* $\Delta_{\text{lift}}(\mathbf{z})$ *is not zero, then* $\mathbf{T}_{\mathbf{z}}$ *is well-defined and* $(\mathbf{z}, \mathbf{T}_{\mathbf{z}})$ *is a lifting fiber for* $(\mathbf{F}, h, <)$*.*

Proof: By Theorem 2 in [105], there exists a non-zero polynomial $\Delta_1 \in \mathbb{K}[\mathbf{Z}]$ of degree at most $n \deg W(3 \deg W + n)$ such that if $\Delta_1(\mathbf{z})$ is not zero, then $\mathbf{z}$ satisfies assumptions $\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3$.

Let next V be the intersection $W \cap V(\sigma)$, where σ is the Jacobian determinant of $\mathbf{F}$ with respect to $\mathbf{Y}$. By assumption, V has dimension at most $r - 1$ and degree at most sd^{n+1}, so there exists a non-zero polynomial $\Delta_2 \in \mathbb{K}[\mathbf{Z}]$ of degree at most sd^{n+1} such that $\pi_{\mathbf{Z}}(V)$ is contained in $V(\Delta_2)$. To conclude, we define $\Delta_{\text{lift}} = \Delta_1 \Delta_2$; the requested degree bound follows from the inequality $\deg W \leq d^n$. □

3.2.2 Algorithmic prerequisites

We will make use of basic operations for univariate polynomials as presented in the preliminary chapter, in Subsection 1.3.2. In particular M is a multiplicatioin time. We precise also the cost function MT for operations modulo a triangular set.

First, we require that MT enables us to describe the cost of ring operations modulo an arbitrary zero-dimensional triangular set. In other words, MT is such that for any n and any triangular set $\mathbf{T} = (T_1, \ldots, T_n)$ in $\mathbb{K}[X_1, \ldots, X_n]$ for the order $X_1 < \cdots < X_n$, all operations $(+, -, \times)$ modulo $\mathbf{T}$ can be computed in $\mathsf{MT}(d_1, \ldots, d_n)$ base field operations, with $d_i = \deg_{X_i}(T_i)$.

Second, we ask that MT enables us to describe the cost of inversion, assuming that we work modulo a triangular set that generates a zero-dimensional *radical* ideal (the radicality assumption is used to derive the bounds given below). In other words, MT is such that for any n and any triangular set $\mathbf{T} = (T_1, \ldots, T_n)$ generating a radical ideal in $\mathbb{K}[X_1, \ldots, X_n]$, given $A \in \mathbb{K}[X_1, \ldots, X_n]$ reduced with respect to $\mathbf{T}$, one can test if A is a unit modulo $\mathbf{T}$ and

if so, compute its inverse, using $\mathsf{MT}(d_1, \ldots, d_n)$ base field operations (with $d_i = \deg_{X_i} T_i$, and assuming that the variables are ordered as above).

Finally, we request that there exists a constant c such that the inequalities

$$\begin{aligned} \mathsf{MT}(d_1, \ldots, d_n) &\leq \mathsf{c}\,\mathsf{MT}(d_1, \ldots, d_n, d_{n+1}, \ldots, d_m) \\ \mathsf{MT}(d_1, \ldots, d_n + 1) &\leq \mathsf{c}\,\mathsf{MT}(d_1, \ldots, d_n) \\ \mathsf{MT}(d_1, \ldots, d_n)d_{n+1} &\leq \mathsf{c}\,\mathsf{MT}(d_1, \ldots, d_n, d_{n+1}) \end{aligned}$$

hold for all values of the arguments. The following proposition then gives an upper bound the complexity of all the previous operations.

Proposition 3.4. *Let $\mathsf{M} : \mathbb{N} \to \mathbb{R}$ be a multiplication time. There exists a constant C such that one can take*

$$\mathsf{MT}(d_1, \ldots, d_n) = \mathsf{C}^{n'} \prod_{i \leq n, d_i \neq 1} \mathsf{M}(d_i)\mathrm{logp}^3(d_i),$$

where n' is the number of elements of $\{d_1, \ldots, d_n\}$ different from 1.

This proposition is proved in Corollary **??**. the chapter "On the complexity of the D5 principle". For the addition and multiplication, the result in Subsection 1.3.2 gives better estimates: there is no logarithmic terms. Observe that for *fixed* n, this bound is *linear* in $d_1 \cdots d_n$, up to logarithmic factors. As a corollary, we also obtain the following result, that shows that the first factor $\mathsf{C}^{n'}$ is controlled by the second one, proving that all these operations can be done in *polynomial* time.

Corollary 3.1. *One can take $\mathsf{MT}(d_1, \ldots, d_n) \leq (d_1 \cdots d_n)^\kappa$, for some constant κ.*

Proof: Let us fix a multiplication time M; hence, there exists a constant λ such that $\mathsf{M}(d)\log^3(d)$ is upper-bounded by d^λ for all d. Let next C be the constant appearing in the previous proposition and let $\mu = \log_2(\mathsf{C})$, so that that $\mathsf{C} = 2^\mu$. Then, for any integer $d > 1$, $\mathsf{C} \leq d^\mu$ holds. To conclude, it suffices to take $\kappa = \lambda\mu$. □

To conclude on this question, we associate a similar notion of cost to operations with an irreducible variety. Let thus $W \subset \overline{\mathbb{K}}^n$ be an irreducible variety defined over $\mathbb{K}$, let r be its dimension, and let I be the defining ideal of W in $\mathbb{K}[\mathbf{X}]$.

Let next $<$ be a variable order, and let $\mathbf{Z}$, $\mathbf{Y}$ and $\mathbf{T} = (T_1, \ldots, T_s) \subset \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ be the canonical representation defined in Proposition 3.1. Writing d_i for the degree of T_i in its main variable, we define $\mathsf{MT}(W, <) = \mathsf{MT}(d_1, \ldots, d_s)$; this will be used to represent the cost of operations modulo a generic specialization of $\mathbf{T}$. To give upper-bounds independent of the choice of $\mathbf{Z}$, we write $\mathsf{MT}(W) = \max \mathsf{MT}(W, <)$, for all orders $<$. Remarking that for any choice of $\mathbf{Z}$, the product $d_1 \cdots d_s$ is upper-bounded by $(\deg W)$, we derive using Corollary 3.1 the polynomial upper bound $\mathsf{MT}(W) \leq (\deg W)^\kappa$. To simplify some estimates, we will also suppose that $(degW) \leq \mathsf{MT}(W)$ holds for all W.

Further operations in dimension zero. Among the needed operations modulo a zero-dimensional triangular set $\mathbf{T}$, we will be led to perform matrix inversion, assuming that $\mathbf{T}$ generates a radical ideal. We expect that for a matrix of size ℓ, this can be done with an order of ℓ^ω operations modulo $\mathbf{T}$, where ω is the exponent of linear algebra over the base field [24]. However, managing the difficulties raised by the fact that $\mathbb{K}[\mathbf{X}]/(\mathbf{T})$ is not a field

but a product of fields is beyond the scope of this article. Hence, we will content ourselves with the following result.

Lemma 3.3. *Let $\mathbf{T} \subset \mathbb{K}[\mathbf{X}]$ be a zero-dimensional triangular set, that generates a radical ideal, and let $\mathbf{m}$ be an $\ell \times \ell$ matrix over $\mathbb{K}[\mathbf{X}]/(\mathbf{T})$. Then one can test if $\mathbf{m}$ is invertible and, if so, compute its inverse, using $O(\ell^4)$ arithmetic operations modulo $\mathbf{T}$.*

Proof: Berkowitz's algorithm [14] computes the characteristic polynomial of $\mathbf{m}$ in the requested complexity, using only ring operations. From this, a single invertibility tests tells whether $\mathbf{m}$ is a unit, and if so, one can deduce the inverse of A for $O(\ell)$ additional $\ell \times \ell$ matrix additions and multiplications. □

Our final subroutine is change of order in dimension zero. Given a zero-dimensional triangular set $\mathbf{T}$ for an input order $<$ and a target order $<'$, we want to compute a triangular set $\mathbf{T}'$ for the order $<'$, such that $(\mathbf{T}) = (\mathbf{T}')$ holds. As was mentioned in the previous subsection, there is no guarantee that the requested output exists (unless $\mathbf{T}$ generates a prime ideal). However, supposing that this output $\mathbf{T}'$ exists, several solutions are available to compute it. Recalling that zero-dimensional triangular sets are actually lexicographic Gröbner bases, we will use the FGLM algorithm [42] to do this operation, obtaining the following complexity estimate.

Proposition 3.5. *Let $\mathbf{T} = (T_1, \ldots, T_n)$ be a zero-dimensional triangular set in $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_n]$ for an input order $<$ and let $<'$ be a target order on $\mathbf{X}$. Suppose that there exists a triangular set $\mathbf{T}'$ in $\mathbb{K}[\mathbf{X}]$ for the target order, such that the equality $(\mathbf{T}) = (\mathbf{T}')$ holds. Then one can compute $\mathbf{T}'$ using $O(n(d_1 \cdots d_n)^3)$ operations in $\mathbb{K}$, where d_i is the degree of T_i in its main variable.*

Newton iteration for triangular sets. Newton iteration enables us to obtain positive-dimensional information starting from a zero-dimensional input. In the case at hand, we start from a lifting fiber $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ for a system $(\mathbf{F}, h, <)$. Then, Newton iteration, combined by rational function reconstruction, enables us to recover the canonical representation $\mathbf{T} \subset \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ associated to $<$, where $\mathbf{Z}, \mathbf{Y}$ and $\mathbf{T}$ are as in Proposition 3.1.

We give a simplified result of the Newton lifting, when only *one* free variable is lifted, since this is what is needed later on. The algorithm is probabilistic (we use a probabilistic criterion to stop the lifting); the following proposition gives the complexity of the process and quantifies the probability of error.

Proposition 3.6. *Let $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ be a lifting fiber for the system $(\mathbf{F}, h, <)$, with $\mathbf{z} = (z_1, \ldots, z_r)$. Suppose that the polynomials in $\mathbf{F}$ can be computed by a straight-line program of size L. Then one can compute $\mathbf{T}(z_1, \ldots, z_{r-1}, Z_r, \mathbf{Y}) \subset \mathbb{K}(Z_r)[\mathbf{Y}]$ using*

$$O\big((n^4 + nL)\,\mathsf{MT}(W)\,\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big)$$

operations in $\mathbb{K}$. The algorithm chooses a value z_r' in $\mathbb{K}$; all possible choices except at most $nd^{2n}(n + 16\log d + 11)$ lead to success.

Proof: Proposition 1.8 gives the cost of one iteration:

$$O\big((nL + n^3)\mathsf{MT}(d_1, \ldots, d_n)\mathsf{MS}(2^\kappa, 1),$$

where $\mathsf{MS}(2^\kappa, 1)$ is the cost multplication of univariate series. In the preliminary chapter, we have seen that $\mathsf{MS}(2^\kappa, 1) \in O(\kappa \mathsf{M}(2^\kappa))$. Moreover $\mathsf{MT}(d_1, \ldots, d_n) \leq \mathsf{MT}(\deg(W))$, and if we compute a matrix inversion as in Lemma 3.3, not taken into account in this estimate, this adds a $O(n^4)$. The bounds in Theorem 2.7 and Equation (1.18) p. 36 shows that the lifting can stop as soon as $\kappa > \log_2(2\deg(W)^2+1)$. After as performing a few simplifications yields our complexity statement (observe that in [105], a matrix inversion in size n over the ring $\mathbb{K}[\mathbf{Y}]/(\mathbf{T}_\mathbf{z})$ was not taken into account; computing this inverse by Lemma 3.3 yields an additional n^4 term in the complexity).

As pointed out in Subsection 1.4.2, the univariate rational reconstruction is not probabilistic. Hence, the only random choice comes from the stop criterion (Figure 1.3, p. 41). To test if a candidate triangular set $\mathsf{U} \subset \mathbb{K}(Z_r)[\mathbf{Y}]$ is indeed the requested output, we specialize it at the random value $z'_r \in \mathbb{K}$, and check if the resulting triangular set $\mathsf{U}_{\mathbf{z}'}$ coincides with $\mathbf{T}_{\mathbf{z}'}$, where $\mathbf{z}'$ denotes the point $(z_1, \ldots, z_{r-1}, z'_r)$. Since of course $\mathbf{T}_{\mathbf{z}'}$ is unknown, to do this check, we use a slight modification of the stop criterion given in Subsection 1.4.3 and sketched in Figure 1.3. testing if:

- the triangular set $\mathsf{U}_{\mathbf{z}'}$ defines a radical ideal;
- the lifting system $\mathbf{F}(\mathbf{z}', \mathbf{Y})$ reduces to zero modulo $\mathsf{U}_{\mathbf{z}'}$;
- the polynomial $h(\mathbf{z}', \mathbf{Y})$ is a unit modulo $\mathsf{U}_{\mathbf{z}'}$.

Assuming that $\mathbf{z}'$ is a lifting fiber for $(\mathbf{F}, h, <)$ and that $\mathbf{z}'$ is not in the projection $\pi_\mathbf{Z}(W \cap V(h))$, the previous conditions imply that $\mathsf{U}_{\mathbf{z}'} = \mathbf{T}_{\mathbf{z}'}$, which is the property we want to test.

Taking this modification into account, in the analysis of [105, Section 7.2.2, page 38], only the second and third items of that reference have to be taken care of. Taking into account the upper bound $2\deg(W)^2 \leq 2d^{2n}$ on the degrees of the polynomials in $\mathbf{T}$ yields the result reported here, after a few simplifications. □

While this is not the main purpose of this article, we also mention (without proof) the complexity and probability analysis for lifting *all* free variables starting from the output lifting fiber of our algorithm. The result is essentially that of [105, Section 7.2], up to the minor modifications already reported in the proof of the previous proposition.

In the complexity estimate, we denote by $\mathsf{MS} : \mathbb{N}^2 \to \mathbb{R}$ a function that bounds the cost of *multivariate power series* arithmetic, that is, such that all operations $(+, -, \times)$ in $\mathbb{K}[Z_1, \ldots, Z_r]/(Z_1, \ldots, Z_r)^d$ can be computed in $\mathsf{MS}(r, d)$ base field operations. We refer to [78, 115] for estimates on this question.

Proposition 3.7. *Let assumptions and notation be as in Proposition 3.6. Then one can compute* $\mathbf{T} \subset \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ *using*

$$O^\sim\big((n^4 + nL)\,\mathsf{MT}(W)\,\mathsf{M}\big((\deg W)^2\big)\,\mathsf{MS}\big(m-1, 8(\deg W)^2\big)\big)$$

operations in $\mathbb{K}$*, where* $O^\sim$ *denotes the omission of logarithmic factors. The algorithm chooses* $2r-1$ *values in* $\mathbb{K}$*. If these values are chosen uniformly at random in a finite subset* S *of* $\mathbb{K}$*, then the algorithm fails for at most* $130\,d^{6n}|S|^{2r-2}$ *choices.*

3.3 Matroids

A substantial part of what follows relies on discussion of *independence properties*. All the required notions are conveniently described through the concept of *matroid* [119, 96]. We give here the basic definitions and introduce a few fundamental examples. We also discuss a greedy algorithm for finding a maximal element among the bases of a matroid, which will be used in the next section.

3.3.1 Definition and examples

A *matroid* $\mathcal{M}$ is given by a finite set $\mathsf{V}(\mathcal{M})$ and a non-empty family $\mathsf{Ind}(\mathcal{M})$ of subsets of $\mathsf{V}(\mathcal{M})$ satisfying the properties below:

Heredity: for all $\mathbf{Z}$ in $\mathsf{Ind}(\mathcal{M})$, every subset of $\mathbf{Z}$ belongs to $\mathsf{Ind}(\mathcal{M})$.

Augmentation: for all $\mathbf{Z}, \mathbf{Z}'$ in $\mathsf{Ind}(\mathcal{M})$ with $|\mathbf{Z}| < |\mathbf{Z}'|$, there exists Z in $\mathbf{Z}' - \mathbf{Z}$ such that $\mathbf{Z} \cup \{Z\}$ is in $\mathsf{Ind}(\mathcal{M})$.

The members of $\mathsf{V}(\mathcal{M})$ and $\mathsf{Ind}(\mathcal{M})$ are the *elements* and the *independents* of the matroid $\mathcal{M}$ (in most of our applications, $\mathsf{V}(\mathcal{M})$ will be the set of variables $\mathbf{X}$ on the ambient space $\overline{\mathbb{K}}^n$). The independents of $\mathcal{M}$ that are maximal for inclusion form a non-empty family $\mathsf{B}(\mathcal{M})$, called the set of *bases* of $\mathcal{M}$. They satisfy the following properties:

Equicardinality: for all $\mathbf{Z}, \mathbf{Z}'$ in $\mathsf{B}(\mathcal{M})$ we have $|\mathbf{Z}| = |\mathbf{Z}'|$,

Exchange: for all $\mathbf{Z}, \mathbf{Z}'$ in $\mathsf{B}(\mathcal{M})$, for every Z in $\mathbf{Z} - \mathbf{Z}'$ there exists Z' in $\mathbf{Z}' - \mathbf{Z}$ such that $\mathbf{Z} - \{Z\} \cup \{Z'\}$ is in $\mathsf{B}(\mathcal{M})$.

The common cardinality of the bases of $\mathcal{M}$ is called the *rank* of $\mathcal{M}$.

Example 1: Vectorial matroids. A first example of a matroid is given by sets of independent vectors. Precisely, let $\mathbf{X}$ be a finite set of cardinality n, let $\mathbb{K}$ be a field, and let $\mathbf{m}$ be an $s \times n$ matrix over $\mathbb{K}$; we suppose that the columns of $\mathbf{m}$ are indexed by the elements of $\mathbf{X}$. Then, we say that a subset $\mathbf{Y} \subset \mathbf{X}$ is *independent* if the corresponding $s \times |\mathbf{Y}|$ submatrix of $\mathbf{m}$ has full rank. One then easily checks that this collection of sets are indeed the independents of a matroid $\mathcal{M}$ over $\mathbf{X}$, which we call the *vectorial matroid generated* by the columns of $\mathbf{m}$. The bases of $\mathcal{M}$ are the subsets $\mathbf{Y}$ corresponding to invertible $s \times s$ submatrices of $\mathbf{m}$.

Example 2: Coordinate matroids. Let $\mathbb{K}$ be a field and let us now consider an irreducible variety $W \subset \overline{\mathbb{K}}^n$ of dimension r, defined over $\mathbb{K}$. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be our usual set of n variables and let I be the prime ideal of $\mathbb{K}[\mathbf{X}]$ defining W; we also write $s = n - r$. Let finally Ind be the family of subsets $\mathbf{Z} \subset \mathbf{X}$ such that $I \cap \mathbb{K}[\mathbf{Z}]$ is the trivial ideal $\{0\}$.

Proposition 3.8. *The family* Ind *is the collection of independent sets of a matroid on* $\mathbf{X}$ *of rank* r*.*

PROOF: Let ℓ be the natural homomorphism $\mathbb{K}[\mathbf{X}] \to \mathbb{K}(W)$ and let $\mathbf{Z}$ be a non-empty subset of $\mathbf{X}$. By definition, we have $\mathbf{Z} \notin \mathsf{Ind}$ if and only there exists a non-constant polynomial $P \in \mathbb{K}[\mathbf{Z}]$ such that $\ell(P) = 0$, that is, the elements $\ell(Z)$, for all $Z \in \mathbf{Z}$, are algebraically dependent over $\mathbb{K}$. We conclude with Theorem 1 p. 183 in [119]. □

In what follows, we denote this matroid by $\mathcal{M}_{\text{coord}}(W)$ and we call it the *coordinate matroid of the variety* W. We can then restate Theorem 1.2 in this language: let $\mathbf{Z}$ be a subset of $\mathbf{X}$ with cardinal r. Then, $\mathbf{Z}$ is a basis of $\mathcal{M}_{\text{coord}}(W)$ if and only if there exists a regular chain $\mathbf{R}$ in $\mathbb{K}[\mathbf{X}]$ having I as saturated ideal and $\mathbf{Z}$ as free variables.

Dual matroids. We continue by introducing the notion of a *dual matroid*. Assume $\mathcal{M}$ is a matroid over $\mathbf{X}$, of rank $r < n$. Denote by $\mathsf{B}^*(\mathcal{M})$ the set of all sets $\mathbf{X} - \mathbf{Z}$ for $\mathbf{Z} \in \mathsf{B}(\mathcal{M})$. Then, the set $\mathsf{B}^*(\mathcal{M})$ is the set of bases of a matroid $\mathcal{M}^*$ of rank $s = n - r$, called the *dual matroid* of $\mathcal{M}$. A subset $\mathbf{Y}$ of $\mathbf{X}$ is an independent of $\mathcal{M}^*$ if and only if there exists a basis $\mathbf{Z} \in \mathsf{B}(\mathcal{M})$ such that $\mathbf{Z} \cap \mathbf{Y}$ is empty.

In particular, we will use this notion with $\mathcal{M} = \mathcal{M}_{\text{coord}}(W)$, the coordinate matroid of an irreducible variety W as above. Let then $\mathcal{M}^* = \mathcal{M}^*_{\text{coord}}(W)$ be its dual. By Theorem 1.2, a subset of $\mathbf{Y}$ of $\mathbf{X}$ is a basis of $\mathcal{M}^*$ if and only if there exists a regular chain $\mathbf{R}$ in $\mathbb{K}[\mathbf{X}]$ having $I = I(W)$ as saturated ideal and $\mathbf{Y}$ as algebraic variables.

Restriction of a matroid. The final needed concept is that of *restriction* of matroids. Let $\mathcal{M}$ be a matroid over $\mathbf{X}$ and let $\mathbf{X}'$ be a subset of $\mathbf{X}$. Then, the collection of the independent sets of $\mathcal{M}$ that are contained in $\mathbf{X}'$ is the family of the independent sets of a matroid on $\mathbf{X}'$, called the *restriction* of $\mathcal{M}$ to $\mathbf{X}'$.

3.3.2 A greedy optimization algorithm

Let $\mathcal{M}$ be a matroid of rank s over $\mathbf{X} = (X_1, \ldots, X_n)$; later on, $\mathcal{M}$ will be the dual of the coordinate matroid of an irreducible variety W, so we denote its independent sets by $\mathbf{Y}$. Suppose that $\mathbf{X}$ is endowed with the order $X_1 < \cdots < X_n$ (one can always suppose that this is the case, up to renaming the variables). In this paragraph, we show how to extend the order $<$ given on $\mathbf{X}$ to the bases of $\mathcal{M}$, and give a greedy algorithm to find the maximal basis.

First, observe that any basis $\mathbf{Y}$ of $\mathcal{M}$ can be ordered as $\mathbf{Y} = (X_{i_1} < \cdots < X_{i_s})$. Let $\mathbf{Y}' \neq \mathbf{Y}$ be another basis of $\mathcal{M}$, which we similarly write $\mathbf{Y}' = (X_{j_1} < \cdots < X_{j_s})$. Let $\kappa \leq s$ be the largest index such that

$$X_{i_s} = X_{j_s}, \quad X_{i_{s-1}} = X_{j_{s-1}}, \quad \ldots, \quad X_{i_\kappa} \neq X_{j_\kappa}.$$

Then if $X_{i_\kappa} > X_{j_\kappa}$, we say that $\mathbf{Y} > \mathbf{Y}'$, and if $X_{i_\kappa} < Xj_\kappa$, we say that $\mathbf{Y} < \mathbf{Y}'$.

In the next section, we will need to compute the maximal basis $\mathbf{Y}_{\max}$ of $\mathcal{M}$ for this order, in the particular case where $\mathcal{M}$ is the dual of the coordinate matroid of an irreducible variety. We now give a general algorithm for finding this maximum basis.

To do so, we will assume that a basis $\mathbf{Y}_0$ of $\mathcal{M}$ is known. Using only independence tests, we will construct a sequence $\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_s$ of bases of $\mathcal{M}$, such that $\mathbf{Y}_s = \mathbf{Y}_{\max}$ and for $i < s$, $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ differ by at most one element. In other words, for all i, either $\mathbf{Y}_{i+1} = \mathbf{Y}_i$,

or there exists B_i and A_i in $\mathbf{X}$ such that the following holds:

$$B_i \in \mathbf{Y}_i, \quad A_i \notin \mathbf{Y}_i, \quad \mathbf{Y}_{i+1} = \mathbf{Y}_i - \{B_i\} \cup \{A_i\} \in \mathcal{M} \tag{3.1}$$

Our algorithm starts by finding the last entry of $\mathbf{Y}_{\max}$, then the last two ones, and so on. The basis of this algorithm is thus the following lemma.

Lemma 3.4. *Let $\mathbf{Y}_{\max}$ be written as $(X_{\ell_1} < \cdots < X_{\ell_s})$ and let $\mathbf{Y} = (X_{\ell'_1} < \cdots < X_{\ell'_s})$ be another basis of $\mathcal{M}$, such that*

$$\ell'_s = \ell_s, \quad \ldots, \quad \ell'_{j+1} = \ell_{j+1}$$

holds. Then ℓ_j equals $\max\{\ell \in \{\ell'_j, \ldots, \ell_{j+1} - 1\} \mid (X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s}) \in \mathsf{Ind}(\mathcal{M})\}$.

Proof: Let S be the set

$$\{\ell \in \{\ell'_j, \ldots, \ell_{j+1} - 1\} \mid (X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s}) \in \mathsf{Ind}(\mathcal{M})\}.$$

We start by showing that ℓ_j is in S. Observe first that $\ell_j \leq \ell_{j+1} - 1$. Next, by definition, we have the inequality $\mathbf{Y}_{\max} > \mathbf{Y}$. Since the entries of indices $j+1, \ldots, s$ of $\mathbf{Y}_{\max}$ and $\mathbf{Y}$ coincide, we deduce that $\ell_j \geq \ell'_j$. Furthermore, since $\mathbf{Y}_{\max} = (X_{\ell_1}, \ldots, X_{\ell_s})$ is in $\mathsf{Ind}(\mathcal{M})$, $(X_{\ell_j}, \ldots, X_{\ell_s})$ is in $\mathsf{Ind}(\mathcal{M})$ as well, by the heredity property. This shows that ℓ_j is in S.

We next prove that ℓ_j is the maximal element of S. Suppose thus that there exist $\ell \in S$ with $\ell > \ell_j$. Since ℓ is in S, $\mathbf{Y}' = (X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s})$ is in $\mathsf{Ind}(\mathcal{M})$. Applying the augmentation property as many times as necessary to $\mathbf{Y}'$ and $\mathbf{Y}_{\max}$, we can complete $\mathbf{Y}'$ into a $\mathbf{Y}''$ basis of $\mathcal{M}$. Since all elements added to $\mathbf{Y}'$ are taken from $\mathbf{Y}_{\max}$, they are all less than X_ℓ. This implies the inequality $\mathbf{Y}'' > \mathbf{Y}_{\max}$, a contradiction. □

The previous lemma yields the following algorithm to compute $\mathbf{Y}_{\max}$. Given a basis $\mathbf{Y}_0$ of $\mathcal{M}$, letting $\ell_{s+1} = n + 1$, we do the following for $j = s, \ldots, 1$.

1. Let $k = s - j$ and write $\mathbf{Y}_k$ as $(X_{\ell_{k,1}} < \cdots < X_{\ell_{k,s}})$.

2. Let ℓ_j be the maximum element of the set

$$\{\ell \in \{\ell_{k,j}, \ldots, \ell_{k,j+1} - 1\} \mid (X_\ell, X_{\ell_{k,j+1}}, \ldots, X_{\ell_{k,s}}) \in \mathsf{Ind}(\mathcal{M})\}.$$

3. If $\ell_j = \ell_{k,j}$, let $\mathbf{Y}_{k+1} = \mathbf{Y}_k$.

4. If $\ell_j > \ell_{k,j}$, let $A_k = X_{\ell_j}$, and find $B_k < A_k$ in $\mathbf{Y}_k$ such that $\mathbf{Y}_k - \{B_k\} \cup \{A_k\}$ is a basis of $\mathcal{M}$. Define $\mathbf{Y}_{k+1} = \mathbf{Y}_k - \{B_k\} \cup \{A_k\}$.

Lemma 3.5. *The previous algorithm correctly computes $\mathbf{Y}_s = \mathbf{Y}_{\max}$.*

Proof: We prove by induction that the last k entries of $\mathbf{Y}_k$ and $\mathbf{Y}_{\max}$ coincide. This is indeed the case for $j = s$ (and hence $k = 0$), so we do the induction step. If we go through Line (3), our claim holds; suppose then that we go through Line (4).

The previous lemma shows that the index ℓ_j is indeed the jth index of $\mathbf{Y}_{\max}$. Observe now that it is indeed possible to find $B_k < A_k$ such that $\mathbf{Y}_k - \{B_k\} \cup \{A_k\}$ is a basis of $\mathcal{M}$. This is done by augmenting the independent set $(X_{\ell_j}, X_{k,\ell_{j+1}}, \ldots, X_{k,\ell_s})$ by elements of $\mathbf{Y}_k$ into a basis of $\mathcal{M}$. An element B_k will be left out, and by construction, $B_k < A_k$. This concludes the proof. □

3.4 Computing the exchange data

Getting back to the context of regular chains, this section describes the first part of our main algorithm: given the input regular chain $\mathbf{F}$ in $\mathbb{K}[\mathbf{X}]$, with $\mathsf{Sat}(\mathbf{F})$ prime, and given the target order $<'$, we compute a sequence of subsets $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$ of $\mathbf{X}$ with the following properties, where we write $W = V(\mathsf{Sat}(\mathbf{F}))$:

- $\mathbf{Y}_0$ is the set of algebraic variables in $\mathbf{F}$;
- $\mathbf{Y}_s$ is the set of algebraic variables in the target regular chain;
- each intermediate $\mathbf{Y}_i$ is a basis of $\mathcal{M}^*_{\text{coord}}(W)$;
- for $i = 0, \ldots, s-1$, either $\mathbf{Y}_{i+1} = \mathbf{Y}_i$, or there exists $A_i \in \mathbf{X} - \mathbf{Y}_i$ and B_i in $\mathbf{Y}_i$ such that the following equation holds:

$$\mathbf{Y}_{i+1} = \mathbf{Y}_i - \{B_i\} \cup \{A_i\}$$

The sequence $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$ will be called the *exchange data*. The main result in this section is an estimate on the cost of computing this sequence.

Proposition 3.9. *Suppose that the input regular chain $\mathbf{F} = (F_1, \ldots, F_s)$ is given by a straight-line program of size L. Let d be an upper bound on the total degree of the polynomials $(F_1, \ldots, F_s)$.*

Suppose that for $i \leq s$, the main variable of F_i is known, as well as its degree d_i in this main variable. Suppose also that $\mathrm{char}\ \mathbb{K}$ *is larger than d^n. Then one can compute the exchange data by a probabilistic algorithm, that uses*

$$O((n^4 + nL)\,\mathsf{MT}(W))$$

operations in $\mathbb{K}$ in case of success. The algorithm uses a random point $\mathbf{z} \in \mathbb{K}^r$; there exists a non-zero polynomial Δ_{lin} in $\mathbb{K}[\mathbf{Z}]$ of degree at most $n(2d)^{n+1}$ such that if $\Delta_{\text{lin}}(\mathbf{z})$ is not zero, the algorithm succeeds.

We start this section by characterizing the algebraic variables for the target order as maximal bases in a suitable matroid (the dual of the coordinate matroid of W). Since testing independence in such a matroid is a difficult problem in general, we will then present a workaround relying on a linearization of the problem, that reduces to linear algebra operations in a product of fields.

3.4.1 Characterization of the target set of algebraic variables

Let $\mathbf{R} = (R_1, \ldots, R_s)$ be a regular chain for the target order $<'$, such that $W = V(\mathsf{Sat}(\mathbf{R}))$. Recall from Subsection 3.3.2 that the order $<'$ induces an order $<'$ on the bases of $\mathcal{M}^*_{\text{coord}}(W)$. Using this order leads us to a characterization of the algebraic variables in the regular chain $\mathbf{R}$.

Proposition 3.10. *The set of the algebraic variables of $\mathbf{R}$ is the maximum basis of $\mathcal{M}^*_{\text{coord}}(W)$ for the order $<'$.*

PROOF: We start by a lemma, using the notion of restriction of a matroid.

Lemma 3.6. *Let m be an index less than n, and let $\mathbf{Z}$ be the set of the first m variables of $\mathbf{X}$ for the target order $<'$. Let also W' be the Zariski closure of $\pi_{\mathbf{Z}}(W)$.*

Then, the matroid $\mathcal{M}_{\text{coord}}(W')$ is the restriction of $\mathcal{M}_{\text{coord}}(W)$ to $\mathbf{Z}$. Moreover, it has rank $r-t$, where t is the number of variables in $\mathbf{X}-\mathbf{Z}$ that are not algebraic variables of $\mathbf{R}$.

PROOF: First, since W' is irreducible [30, Theorem 3 p. 122], $\mathcal{M}_{\text{coord}}(W')$ is well-defined. In addition, that results shows that a subset of $\mathbf{Z}$ is a an independent set of $\mathcal{M}_{\text{coord}}(W')$ if and only if it is an independent set of $\mathcal{M}_{\text{coord}}(W)$ contained in $\mathbf{Z}$. This proves the first claim.

Define $\mathbf{R}_m = \mathbf{R} \cap \mathbb{K}[\mathbf{Z}]$. It follows from the definition of a regular chain that $\mathbf{R}_m$ is a regular chain. Moreover, it follows from Proposition 5.1 and Theorem 6.1 in [7] that the saturated ideal of $\mathbf{R}_m$ in $\mathbb{K}[\mathbf{Z}]$ is $I \cap \mathbb{K}[\mathbf{Z}]$. Then, Theorem 1.2 implies that the rank of $\mathcal{M}_{\text{coord}}(W')$ is $m-|\mathbf{R}_m|$. Observe now that the number of elements in $\mathbf{R}_m$ is $|\mathbf{R}|-(n-m)+t$. Hence, the rank of $\mathcal{M}_{\text{coord}}(W')$ is $n-|\mathbf{R}|-t$, that is, $r-t$. □

We can now prove the proposition. Let $\mathbf{Y}$ be the set of the algebraic variables of $\mathbf{R}$ and recall first that $\mathbf{Y}$ is indeed in $\mathcal{M}^*_{\text{coord}}(W)$. Assuming that there exists a basis $\mathbf{Y}'$ of $\mathcal{M}^*_{\text{coord}}(W)$ such that $\mathbf{Y} < \mathbf{Y}'$ holds, we will derive a contradiction. To this effect, let $X_{\max}$ be the largest element (for the order $<'$) that belongs to $\mathbf{Y}'$ and not to $\mathbf{Y}$; let m be such that $X_{\max}$ is the $m+1$th element of $\mathbf{X}$, and let $\mathbf{Z}$ and W' be as in Lemma 3.6. By Lemma 3.6, $\mathcal{M}_{\text{coord}}(W')$ is the restriction of $\mathcal{M}_{\text{coord}}(W)$ to $\mathbf{Z}$. As in the lemma, we let t be the number of variables in $\mathbf{X}-\mathbf{Z}$ that are not algebraic variables of $\mathbf{R}$.

Let us prove that the intersection of $\mathbf{X}-\mathbf{Y}'$ with $\mathbf{Z}$ is an independent set of $\mathcal{M}_{\text{coord}}(W')$ of cardinality is $r-t+1$. We have $|\mathbf{Y}'| = s = n-r$, since $\mathbf{Y}'$ is a basis of $\mathcal{M}^*_{\text{coord}}(W)$. Now, the definitions of m and t imply the equality $|\mathbf{Y}' \cap (\mathbf{X}-\mathbf{Z})| = n-m-t+1$, which leads to $|\mathbf{Y}' \cap \mathbf{Z}| = m+t-1-r$, proving our claim. We have reached a contradiction, since Lemma 3.6 states that the rank of $\mathcal{M}_{\text{coord}}(W')$ is $r-t$. □

3.4.2 Linearization

In what follows, we use all the notation of Proposition 3.9. The previous subsection showed that the set of algebraic variables in the target regular chain is the maximum basis of $\mathcal{M}^*_{\text{coord}}(W)$. In order to apply the algorithm of Subsection 3.3.2 to find this maximum, we need to perform the required independence tests. To do so, we will use that fact that for a random point $\mathbf{x}$ on W, the coordinate matroids $\mathcal{M}_{\text{coord}}(W)$ and $\mathcal{M}_{\text{coord}}(T_{\mathbf{x}}W)$ coincide, where $T_{\mathbf{x}}W$ is the tangent space of W at $\mathbf{x}$. This will enable us to perform the required independence tests by linear algebra.

We will assume that the characteristic of $\mathbb{K}$ is larger than d^n, where d is an upper bound on the degrees of the polynomials is $\mathbf{F}$; hence, by Bézout's inequality, char $\mathbb{K}$ is larger than $(\deg W)$, so in particular Lemma 3.1 applies.

Let $\mathbf{Z}$ (resp. $\mathbf{Y}$) be the free (resp. algebraic) variables in $\mathbf{F}$, and let **jac** be the Jacobian matrix of $\mathbf{F}$. In what follows, if $\mathbf{Y}'$ is a subset of $\mathbf{X}$ of cardinality s and $\mathbf{m}$ a matrix with s rows and with columns indexed by $\mathbf{X}$, we denote by $\mathbf{m}(\mathbf{Y}')$ the determinant of the submatrix of $\mathbf{m}$ corresponding to the columns indexed by $\mathbf{Y}'$.

Given $\mathbf{z}$ in $\mathbb{K}^r$, we denote by $\mathbf{F}_\mathbf{z}$ the family of polynomials $\mathbf{F}(\mathbf{z}, \mathbf{Y})$ in $\mathbb{K}[\mathbf{Y}]$, by $Q_\mathbf{z}$ the residue class ring $\mathbb{K}[\mathbf{Y}]/(\mathbf{F}_\mathbf{z})$ and by $\mathbf{jac}_\mathbf{z}$ the Jacobian matrix of $\mathbf{F}$, seen as a matrix with entries in $Q_\mathbf{z}$. We then denote by $\mathsf{B}_\mathbf{z}(\mathbf{F})$ the set

$$\{\mathbf{Y}' \subset \mathbf{X} \text{ such that } |\mathbf{Y}'| = s \text{ and } \mathbf{jac}_\mathbf{z}(\mathbf{Y}') \text{ is invertible}\}.$$

In general, $Q_\mathbf{z}$ is not a field, so that $\mathsf{B}_\mathbf{z}(\mathbf{F})$ is not evidently the set of bases of a vectorial matroid over $\mathbf{X}$. The following proposition shows that for most choices of $\mathbf{z}$, however, there is such a matroid structure.

Proposition 3.11. *There exists a non-zero polynomial $\Delta_{\text{lin}} \in \mathbb{K}[\mathbf{Z}]$ of degree at most $n(2d)^{n+1}$ such that if $\Delta_{\text{lin}}(\mathbf{z})$ is not zero, $\mathbf{F}_\mathbf{z}$ is a regular chain in $\mathbb{K}[\mathbf{Y}]$ that defines a radical ideal, and $\mathsf{B}_\mathbf{z}(\mathbf{F})$ is the set of bases of $\mathcal{M}_{\text{coord}}(W)^*$.*

Hence, this proposition says that for most choices of $\mathbf{z}$, $Q_\mathbf{z}$ is a product of finite field extensions of $\mathbb{K}$, and the maximal minors of the Jacobian matrix $\mathbf{jac}_\mathbf{z}$ over $Q_\mathbf{z}$ correspond to the sets of algebraic variables for W. The rest of this subsection is devoted to prove this proposition.

To start with, let $\mathcal{TM}(\mathbf{F}) \subset \mathbf{X}$ be the vectorial matroid generated by the columns of $\mathbf{jac}$ over $\mathbb{K}(W)$. Then we have the following linearization property, which is a rewording of the implicit function theorem adapted to our context.

Lemma 3.7. *The matroid $\mathcal{TM}(\mathbf{F})$ equals $\mathcal{M}_{\text{coord}}(W)^*$.*

Proof: Let $\mathbf{Y}'$ be a subset of $\mathbf{X}$ and let $\mathbf{Z}' = \mathbf{X} - \mathbf{Y}'$. We have to prove that $\mathbf{Z}'$ is a maximal set of free variables for W if and only $\mathbf{jac}(\mathbf{Y}')$ is a unit in $\mathbb{K}(W)$, that is, if it does not vanish identically on W.

Suppose that $\mathbf{jac}(\mathbf{Y}')$ does not vanish identically on W, and let M be the sequence $(\mathbf{jac}(\mathbf{Y}')^i)_{i\geq 0}$. Our assumption implies that the multiplicative set M does not intersect $(\mathbf{F})$. Then, Proposition 3.2.a in [90] shows that each prime component J of $(\mathbf{F}) : M^\infty$ admits $\mathbf{Z}'$ as a maximal set of free variables, and $\mathbf{Y}'$ as algebraic variables. Writing h for the product of the initials in $\mathbf{F}$, the ideal $I = (\mathbf{F}) : h^\infty$ appears as one of these components, proving the first direction of our equivalence.

Suppose next that $\mathbf{Z}'$ is a maximal set of free variables. Using Lemma 3.1, Lemma 16.15 in [38] implies that the module of differentials $\Omega_{\mathbb{K}(W)/\mathbb{K}(\mathbf{Z})} = 0$. Letting $\mathbf{G}$ be a set of generators of $\mathsf{Sat}(\mathbf{F})$, this means that the Jacobian matrix of $\mathbf{G}$ with respect to $\mathbf{Y}'$ has maximal rank over $\mathbb{K}(W)$. Then, the definition of $\mathbf{G}$ implies that $\mathbf{jac}(\mathbf{Y}')$ is has full rank over $\mathbb{K}(W)$ as well. □

We continue the proof by discussing specialization properties. For any $\mathbf{x} \in W$, let us denote by $\mathcal{TM}_\mathbf{x}(\mathbf{F})$ the vectorial matroid generated over $\overline{\mathbb{K}}$ by the columns of the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{x}$.

Lemma 3.8. *There exists a non-zero polynomial $\Delta_1 \in \mathbb{K}[\mathbf{Z}]$ of degree at most $sd^{n+1}\binom{s}{n}$ with the following property. Let $\mathbf{z}$ be in $\mathbb{K}^r$ such that $\Delta_1(\mathbf{z}) \neq 0$; then, for any $\mathbf{x}$ in the fiber $W_\mathbf{z}$, the equality $\mathcal{TM}_\mathbf{x}(\mathbf{F}) = \mathcal{TM}(\mathbf{F})$ holds.*

Proof: Let $\mathbf{Y}'$ be a subset of $\mathbf{X}$ of cardinality s. If $\mathbf{Y}'$ is not a basis of $\mathcal{TM}(W)$, then $\mathbf{jac}(\mathbf{Y}')$ vanishes identically on W, so for any $\mathbf{x}$ in W, $\mathbf{Y}'$ is not in $\mathcal{TM}_\mathbf{x}(\mathbf{F})$.

Conversely, suppose that $\mathbf{Y}'$ is a basis of $\mathcal{TM}_\mathbf{x}(\mathbf{F})$, so that $\mathbf{jac}(\mathbf{Y}')$ does not vanish in $\mathbb{K}(W)$, and let $V_{\mathbf{Y}'}$ be the projection of $V(\mathbf{jac}(\mathbf{Y}')) \cap W$ on the $\mathbf{Z}$-space. Since W

is irreducible, $V_{\mathbf{Y}'}$ has dimension at most $m-1$ and degree at most $(d \deg W) \leq sd^{n+1}$. Thus, there exists a non-zero polynomial $\Delta_{\mathbf{Y}'} \in \mathbb{K}[\mathbf{Z}]$ of degree at most sd^{n+1}, such that if $\Delta_{\mathbf{Y}'}(\mathbf{z}) \neq 0$, then $\mathbf{jac}(\mathbf{Y}')$ vanishes on none of the points $\mathbf{x} \in W$ above $\mathbf{z}$.

It suffices to take for Δ_1 the product of all $\Delta_{\mathbf{Y}'}$, for $\mathbf{Y}'$ in $\mathcal{TM}(W)$. Since the rank of $\mathcal{TM}(\mathbf{F})$ is at most $\binom{s}{n}$, the conclusion follows. □

We can now conclude the proof of Proposition 3.11. Observe first that the assumption of Proposition 3.2 is satisfied: by Theorem 1.2, the set of algebraic variables $\mathbf{Y}$ of $\mathbf{F}$ is in $\mathcal{M}^*_{\text{coord}}(W)$; Lemma 3.7 then implies that the Jacobian determinant σ of $\mathbf{F}$ with respect to $\mathbf{Y}$ does not vanish identically on W, as requested. We then let Δ_{reg} be the polynomial defined in Proposition 3.2. Observe that if $\Delta_{\text{reg}}(\mathbf{z})$ is not zero, the fiber $W_{\mathbf{z}}$ equals $\{\mathbf{z}\} \times V(\mathbf{F}_{\mathbf{z}})$, and $\mathbf{F}_{\mathbf{z}}$ is a regular chain that generates a radical ideal. Then, for a polynomial $G \in \mathbb{K}[\mathbf{X}]$, $G(\mathbf{z}, \mathbf{Y})$ is a unit in $Q_{\mathbf{z}}$ if and only if G is non-zero at every point in the fiber $W_{\mathbf{z}}$.

If we suppose additionally that $\Delta_1(\mathbf{z})$ is not zero, then by Lemma 3.8, for any $\mathbf{x}$ in $W_{\mathbf{z}}$, $\mathcal{TM}_{\mathbf{x}}(\mathbf{F}) = \mathcal{TM}(\mathbf{F})$. In particular, for any $\mathbf{Y}' \subset \mathbf{X}$ of cardinality s, $\mathbf{Y}'$ is a basis of $\mathcal{TM}(\mathbf{F})$ if and only if $\mathbf{Y}'$ is a basis of $\mathcal{TM}_{\mathbf{x}}(\mathbf{F})$ for all $\mathbf{x}$ above $\mathbf{z}$, that is, if and only if the corresponding determinant $\mathbf{jac}(\mathbf{z}, \mathbf{Y}')$ vanishes on none of these points $\mathbf{x}$. By the preceding remarks, this is the case exactly when this determinant is a unit in $Q_{\mathbf{z}}$. Hence, it suffices to take $\Delta_{\text{lin}} = \Delta_1 \Delta_{\text{reg}}$; the degree estimates comes from a straightforward simplification.

3.4.3 Computing the initial specialization

The previous subsection gives the theoretical foundation of our algorithm for computing the exchange data; this paragraph is devoted to study a preliminary subroutine for this algorithm. As before, given the input regular chain $\mathbf{F}$, having $\mathbf{Z}$ as free variables (resp. $\mathbf{Y}$ as algebraic variables), and a point $\mathbf{z} \in \mathbb{K}^r$, we denote by $\mathbf{F}_{\mathbf{z}}$ the set of polynomials of $\mathbb{K}[\mathbf{Y}]$ obtained by specializing $\mathbf{Z}$ at $\mathbf{z}$ in $\mathbf{F}$.

We will assume here that $\mathbf{z}$ satisfies the assumption of Proposition 3.11; hence $\mathbf{F}_{\mathbf{z}}$ is a regular chain and defines a radical ideal. Let $\mathbf{T}_{\mathbf{z}} \subset \mathbb{K}[\mathbf{Y}]$ be the monic form of $\mathbf{F}_{\mathbf{z}}$, that is, the triangular set obtained by inverting all initials of $\mathbf{F}_{\mathbf{z}}$. We estimate here the cost of computing $\mathbf{T}_{\mathbf{z}}$ from the input regular chain $\mathbf{F}$, showing that this can be done in time polynomial in the degree of the variety $W = V(\mathsf{Sat}(\mathbf{F}))$, and the complexity of evaluation of $\mathbf{F}$.

Proposition 3.12. *Suppose that the input regular chain $\mathbf{F} = (F_1, \ldots, F_s)$ is given by a straight-line program of size L, and assume that the main variable of F_i and the degree d_i of F_i in this main variable are known for all i. Let $\mathbf{z}$ be in $\mathbb{K}^r$ that does not cancel the polynomial Δ_{lin} of Proposition 3.11. Then the monic form $\mathbf{T}_{\mathbf{z}}$ of $\mathbf{F}_{\mathbf{z}}$ can be computed in $O(s\,L\,\mathsf{MT}(W))$ operations in $\mathbb{K}$.*

Proof: We compute inductively the polynomials $T_1, \ldots, T_s$ of $\mathbf{T}_{\mathbf{z}}$. Supposing that $T_1, \ldots, T_{i+1}$ are known, we deduce the cost of computing T_{i+1}. We write the entries of $\mathbf{Y}$ as $(Y_1, \ldots, Y_s)$, where Y_i is the main variable of F_i. We also let Γ be the straight-line program computing $\mathbf{F}$; in particular, Γ compute F_{i+1}. By replacing all indeterminates $Y_{i+2}, \ldots, Y_s$ by 0, we may assume without loss of generality that Γ involves only the variables $\mathbf{Z}, Y_1, \ldots, Y_{i+1}$.

The main idea is then to evaluate Γ modulo $(T_1, \ldots, T_i)$, after specializing $\mathbf{Z}$ at $\mathbf{z}$.

However, we need to control the degree in Y_{i+1} as well; hence the evaluation will be done in

$$Q = \mathbb{K}[Y_1, \ldots, Y_{i+1}]/(T_1, \ldots, T_i, Y_{i+1}^{d_{i+1}+1}),$$

as this is enough to recover $F_{i+1}(\mathbf{z}, Y_1, \ldots, Y_i)$ modulo $(T_1, \ldots, T_i)$. In view of the discussion in Subsection 3.2.2, and in particular of Equations (3.1), the cost of a single operation in Q is $\mathsf{MT}(d_1, \ldots, d_i, d_{i+1}+1) \in O(\mathsf{MT}(W))$. Hence, the whole cost of this step is in $O(L\,\mathsf{MT}(W))$.

By assumption on $\mathbf{z}$, the initial h_{i+1} is a unit modulo $(T_1, \ldots, T_i)$; computing its inverse g_i can then be done in time $\mathsf{MT}(d_1, \ldots, d_i)$. Once this inverse is known, we multiply all coefficients of F_{i+1} by g_i modulo $T_1, \ldots, T_i$ to conclude. The cost is $\mathsf{MT}(d_1, \ldots, d_i)d_{i+1}$ which is in $O(\mathsf{MT}(W))$, again by Equations (3.1). Putting all estimates together and summing over i finishes the proof. □

3.4.4 Computing the exchange data

We conclude this section by proving Proposition 3.9. The exchange data will be computed by applying the algorithm of Subsection 3.3.2 in our particular case, using the previous linearization results to perform independence tests. We will write $\mathbf{Z}_0 = \mathbf{Z}$ and $\mathbf{Y}_0 = \mathbf{Y}$. Recall that given the initial basis $\mathbf{Y}_0$ of $\mathcal{M}^*_{\text{coord}}(W)$, the algorithm of Subsection 3.3.2 computes a sequence of bases $\mathbf{Y}_1, \ldots, \mathbf{Y}_s$, where $\mathbf{Y}_s = \mathbf{Y}_{\max}$ is the set of algebraic variables in the output regular chain.

Let $\mathbf{z}$ be in $\mathbb{K}^r$, such that $\mathbf{z}$ does not cancel the polynomial Δ_{lin} of Proposition 3.11, let $\mathbf{T}_\mathbf{z} \subset \mathbb{K}[\mathbf{Y}]$ be the triangular set obtained by inverting all initials of $\mathbf{F}_\mathbf{z}$, and let $Q_\mathbf{z}$ be the residue class ring $\mathbb{K}[\mathbf{Y}]/(\mathbf{T}_\mathbf{z})$. Then, $Q_\mathbf{z}$ is is a product of finite field extensions of $\mathbb{K}$. Let $\mathbf{jac}_\mathbf{z}$ be the Jacobian matrix of $\mathbf{F}$, seen as a matrix with entries in $Q_\mathbf{z}$. Then, in addition, a subset $\mathbf{Y}'$ of size s of $\mathbf{X}$ is a basis of $\mathcal{M}^*_{\text{coord}}(W)$ if and only if the submatrix $\mathbf{jac}_\mathbf{z}(\mathbf{Y}')$ is invertible.

To prove Proposition 3.9, it will be enough to give the cost of deducing $\mathbf{Y}_{k+1}$ from $\mathbf{Y}_k$. We will actually assume that at step k, in addition to $\mathbf{Y}_k$, the inverse of the matrix $\mathbf{jac}_\mathbf{z}(\mathbf{Y}_k)$ is known, and we will deduce simultaneously the new basis $\mathbf{Y}_{k+1}$ and the inverse of the matrix $\mathbf{jac}_\mathbf{z}(\mathbf{Y}_{k+1})$. Below, we write $\mathbf{Y}_{\max} = (X_{\ell_1} < \cdots < X_{\ell_s})$.

Proposition 3.13. *Given the matrix* $\mathbf{jac}_\mathbf{z}$*, the basis* $\mathbf{Y}_k$ *and the inverse of the matrix* $\mathbf{jac}_\mathbf{z}(\mathbf{Y}_k)$*, one can compute the basis* $\mathbf{Y}_{k+1}$ *and the inverse of* $\mathbf{jac}_\mathbf{z}(\mathbf{Y}_{k+1})$ *using* $O(n^2(\ell_{k+1} - \ell_k))$ *arithmetic operations in* $Q_\mathbf{z}$*.*

Proof: Following the description in Subsection 3.2.2, we let $j = s - k$ and we write

$$\mathbf{Y}_k = (X_{\ell_{k,1}} < \cdots < X_{\ell_{k,s}}),$$

so that $\ell_{k,j+1} = \ell_{j+1}, \ldots, \ell_{k,s} = \ell_s$ holds. Recall then that from Lemma 3.4, ℓ_j is the maximal element of

$$S = \{\ell \in \{\ell_{k,j}, \ldots, \ell_{j+1} - 1\} \mid (X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s}) \in \mathsf{Ind}(\mathcal{M}^*_{\text{coord}}(W))\}.$$

It is then easy to describe the set S. Let $\mathbf{m}$ be the matrix $\left(\mathbf{jac}_\mathbf{z}(\mathbf{Y}_k)\right)^{-1}\mathbf{jac}_\mathbf{z}$. Our basic

remark is that the matrix $\mathbf{m}$ has the following shape:

$$\begin{bmatrix} \star & \cdots & \star & 1 & \star & \cdots & \star & 0 & \star & \cdots & \star & 0 & \star & \cdots & \star \\ \star & \cdots & \star & 0 & \star & \cdots & \star & 1 & \star & \cdots & \star & 0 & \star & \cdots & \star \\ \star & \cdots & \star & \vdots & \star & \cdots & \star & \vdots & \star & \cdots & \star & \vdots & \star & \cdots & \star \\ \star & \cdots & \star & 0 & \star & \cdots & \star & 0 & \star & \cdots & \star & 1 & \star & \cdots & \star \end{bmatrix},$$

having an identity submatrix at the columns indexed by $\mathbf{Y}_k$.

Lemma 3.9. *Let ℓ be in $\{\ell_{k,j}, \ldots, \ell_{j+1} - 1\}$. Then ℓ is in S if and only if the (j, ℓ)-entry $\mathbf{m}_{j,\ell}$ of $\mathbf{m}$ is a unit.*

Proof: Let us write $\mathbf{Y}' = (X_{\ell_{k,1}}, \ldots, X_{\ell_{k,j-1}}, X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s})$, and observe that the submatrix $\mathbf{m}(\mathbf{Y}')$ is diagonal with 1's on the diagonal, except for its ℓ-column. If the entry $\mathbf{m}_{j,\ell}$ is a unit, $\mathbf{m}(\mathbf{Y}')$ is invertible, which implies that $\mathbf{jac}_{\mathbf{z}}(\mathbf{Y}')$ is invertible too, and thus that ℓ is in S.

Conversely, suppose that ℓ is in S, so that $(X_\ell, X_{\ell_{j+1}}, \ldots, X_{\ell_s})$ is an independent set in $\mathcal{M}^*_{\text{coord}}(W)$. This independent set can be augmented into a basis $\mathbf{Y}'$ of $\mathcal{M}^*_{\text{coord}}(W)$. The submatrix $\mathbf{m}(\mathbf{Y}')$ is then a unit; in view of the shape of the matrix $\mathbf{m}$, this implies that the entry $\mathbf{m}_{j,\ell}$ is a unit. □

We can then conclude the proof of Proposition 3.13. Assuming that ℓ_j is known, let us define $\mathbf{Y}_{k+1} = \mathbf{Y}_k - \{X_{\ell_{k,j}}\} \cup \{X_{\ell_j}\}$. Since by construction the submatrix $\mathbf{m}(\mathbf{Y}_{k+1})$ is a unit, $\mathbf{Y}_{k+1}$ is indeed a basis of $\mathcal{M}^*_{\text{coord}}(W)$.

It remains to estimate the complexity of this process. First, observe that we do not need the full matrix $\mathbf{m}$, but only its submatrix $\mathbf{m}(X_{\ell_{k,j}}, \ldots, X_{\ell_{j+1}-1})$, since this is where the search takes place. Furthermore, its columns can be computed one at a time, starting from the ones of highest indices, until an invertible entry is found: the cost for computing the requested part of $\mathbf{m}$ is thus $O(n^2(\ell_{j+1} - \ell_j))$ operations $(+, -, \times)$ in $Q_{\mathbf{z}}$.

Finding ℓ_j requires at most $\ell_{j+1} - \ell_{k,j}$ invertibility tests in $Q_{\mathbf{z}}$, starting from index $\ell_{j+1} - 1$. To conclude, we need to compute the inverse of $\mathbf{jac}_{\mathbf{z}}(\mathbf{Y}_{k+1})$. Since $\mathbf{Y}_k$ and $\mathbf{Y}_{k+1}$ differ by a single entry, the inverse of $\mathbf{jac}_{\mathbf{z}}(\mathbf{Y}_{k+1})$ can be obtained in $O(n^2)$ operations $(+, -, \times)$ in $Q_{\mathbf{z}}$, together with the inversion of the (j, ℓ_j)-entry of $\mathbf{m}$. Putting all costs together gives the bound of Proposition 3.13. □

We can then finish the proof of Proposition 3.9. Correctness of the previous algorithm follows from Lemma 3.5, so it remains to deal with the complexity analysis. As a preliminary, we need to compute the triangular set $\mathbf{T}_{\mathbf{z}}$: the cost is estimated in Proposition 3.12.

Using backward derivation [13], the Jacobian matrix of $\mathbf{F}$ can be evaluated in $O(nL)$ operations, so that its modular image $\mathbf{jac}_{\mathbf{z}}$ can be evaluated in $O(nL)$ operations in $Q_{\mathbf{z}}$. Using Lemma 3.3, one can compute the inverse of the submatrix $\mathbf{jac}_{\mathbf{z}}(\mathbf{Y}_0)$ in $O(n^4)$ operations in $Q_{\mathbf{z}}$, involving only the inversion of its determinant. Finally, summing the complexity estimate of the previous proposition for all values $k = 0, \ldots, s-1$, the total cost of the final part of the algorithm is $O(n^3)$ operations in $Q_{\mathbf{z}}$, so that the total number of operations in $Q_{\mathbf{z}}$ for finding the maximal basis is $O(n^4 + nL)$. Using the definition of the function MT, this concludes the proof of Proposition 3.9.

3.5 Changing the lifting fiber

In this section, we describe the operations in the second phase of our algorithm. Given the input regular chain $\mathbf{F}$, we suppose at this stage that the *exchange data* has been computed previously. This means that we know a sequence $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$ in $\mathcal{M}^*_{\text{coord}}(W)$, for $W = V(\mathsf{Sat}(\mathbf{F}))$, where $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ differ by at most one element for all i.

Starting from a lifting fiber associated to the choice of algebraic variables $\mathbf{Y}_0$, we will now compute a sequence of lifting fibers associated to the algebraic variables $\mathbf{Y}_1, \ldots$ and finally output a lifting fiber associated to the set of algebraic variables $\mathbf{Y}_s$.

The ith step goes as follows. Suppose that $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ are such that $\mathbf{Y}_{i+1} = \mathbf{Y}_i - \{B_i\} \cup \{A_i\}$, with $\mathbf{Y}_{i+1} \neq \mathbf{Y}_i$ (if they coincide, there is nothing to do). Hence, A_i is a free variable at step i that becomes algebraic, and B_i is algebraic at step i and becomes free. Suppose also that we know a lifting fiber for $\mathbf{Y}_i$. First, we change the order in this lifting fiber, so that B_i becomes the smallest algebraic variable: this is done using a routine for change of order in dimension zero. Then, we lift the free variable A_i using Newton iteration, clean all denominators (if needed), and specialize B_i at a random value. Making all polynomials monic in the resulting regular chain yields the next lifting fiber.

As an illustration, consider the variety W given in the introduction, defined over the field $\mathbb{K}$ by the equations

$$P_1 - X_1^2 = 0, \quad P_2 - X_2^2 = 0, \quad S - X_1 X_2 = 0.$$

The initial set of free variables is (X_1, X_2), with algebraic variables (S, P_1, P_2); the first lifting fiber is $(X_1 = 1, X_2 = 1)$, together with the zero-dimensional triangular set

$$\left| \begin{array}{l} P_1 - 1 \\ P_2 - 1 \\ S - 1. \end{array} \right.$$

The second set of free variables is (X_1, P_2), with algebraic variables (S, P_1, X_2). To obtain the corresponding lifting fiber, the first operation consists in putting P_2 as last free variable in the previous lifting fiber. Here, this is a trivial computation, yielding

$$\left| \begin{array}{l} P_1 - 1 \\ S - 1 \\ P_2 - 1. \end{array} \right.$$

We then lift X_2, using Newton's iteration. Here again, the computation is trivial; we obtain

$$\left| \begin{array}{l} P_1 - 1 \\ S - X_2 \\ P_2 - X_2^2. \end{array} \right.$$

Finally, we specialize P_2 at a "random" value, here 1, and rearrange the equations (making every equation monic again), to obtain a lifting fiber corresponding to the set of algebraic variables (S, P_1, X_2).

$$\left| \begin{array}{l} P_1 - 1 \\ S - X_2 \\ 1 - X_2^2 \end{array} \right. \quad \rightsquigarrow \quad \left| \begin{array}{l} P_1 - 1 \\ S - X_2 \\ X^2 - 1. \end{array} \right.$$

This section describes this process, gives a complexity analysis and quantifies the bad specialization choices. Since the whole second step of our main algorithm essentially amounts to perform at most s times the variable exchange process just described, we concentrate on proving the following proposition.

Proposition 3.14. *Let $\mathbf{Y}$ and $\mathbf{Y}'$ be two sets of algebraic variables for W, such that $\mathbf{Y}' = \mathbf{Y} - \{B\} \cup \{A\}$ holds. Suppose that a lifting fiber $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ for the set of algebraic variables $\mathbf{Y}$ is known, and write $\mathbf{z} = (z_1, \ldots, z_{r-1}, a)$.*

Then one can compute a lifting fiber $(\mathbf{z}', \mathsf{U}_{\mathbf{z}'})$ for the set of algebraic variables $\mathbf{Y}'$ by a probabilistic algorithm, using

$$O\big((n^4 + nL)\,\mathsf{MT}(W)\,\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big)$$

operations in $\mathbb{K}$ in case of success. The algorithm chooses two values values (a', b) in $\mathbb{K}$, letting in particular $\mathbf{z}' = (z_1, \ldots, z_{r-1}, b)$.

There exists a non-zero polynomial $\Delta_{\text{exchange}} \in \mathbb{K}[Z_1, \ldots, Z_{r-1}, A', B]$ of degree at most $2d^n(3d^{2n}+(6m+13m^2)d^n+m^2)$, with $m = \max(n, d)$, such that if $\Delta_{\text{exchange}}(z_1, \ldots, z_{r-1}, a', b)$ is not zero, the algorithm succeeds.

Given the exchange data $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$, applying successively this proposition to

$$(\mathbf{Y}_0, \mathbf{Y}_1), \ldots, (\mathbf{Y}_{s-1}, \mathbf{Y}_s)$$

will easily yield the proof of our main theorem. Hence, the rest of this section is devoted to prove this proposition.

3.5.1 Setup and preliminaries

We first detail some preparatory steps for our algorithm, using the notation of Proposition 3.14. Let thus $\mathbf{Y}$ and $\mathbf{Y}'$ be two bases of $\mathcal{M}^*_{\text{coord}}(W)$, and let $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$ and $\mathbf{Z}' = \mathbf{X} - \mathbf{Y}'$. We suppose that $\mathbf{Y}$ and $\mathbf{Y}'$ differ by a single variable, so that we will write

$$\mathbf{Y} = (B, Y_2, \ldots, Y_s) \quad \text{and} \quad \mathbf{Y}' = (A, Y_2, \ldots, Y_s),$$

with $A \neq B$, or equivalently

$$\mathbf{Z} = (Z_1, \ldots, Z_{r-1}, A) \quad \text{and} \quad \mathbf{Z}' = (Z_1, \ldots, Z_{r-1}, B).$$

Suppose finally that we know a lifting fiber in $\mathbb{K}[\mathbf{Y}]$ for the input set of algebraic variables $\mathbf{Y}$. First, we perform a change of order in dimension zero on this lifting fiber, to make it comply to the order given by

$$Z_1 < \cdots < Z_{r-1} < A < B < Y_2 < \cdots < Y_s,$$

which we will call the *input order*. The cost of this operation is given in Subsection 3.2.2: using the FGLM algorithm, it is at most $n(\deg W)^3$ operations in $\mathbb{K}$. Without loss of generality, we suppose from now on that the input lifting fiber $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ supports this order. Accordingly, we let $\mathbf{T} = (T_1, \ldots, T_s) \subset \mathbb{K}(\mathbf{Z})[\mathbf{Y}]$ and $\mathbf{R} = (R_1, \ldots, R_s) \in \mathbb{K}[\mathbf{Z}][\mathbf{Y}] = \mathbb{K}[\mathbf{X}]$ be the canonical representations associated to this order, coming from Proposition 3.1.

Let us write $\mathbf{z}$ as $(z_1, \ldots, z_r) \in \mathbb{K}^r$ and let us define $\mathbf{Z}_- = (Z_1, \ldots, Z_{r-1})$. In the computation to follow, all variables in $\mathbf{Z}_-$ will be specialized at the value $\mathbf{z}_- = (z_1, \ldots, z_{r-1}) \in \mathbb{K}^{r-1}$. Hence, we write $\mathbf{T}_-$ for the triangular set in $\mathbb{K}(A)[\mathbf{Y}]$ obtained by specializing $\mathbf{Z}_-$ at $\mathbf{z}_-$ in all coefficients of $\mathbf{T}$; we also define $\mathbf{R}_-$ as the family of polynomials in $\mathbb{K}[A, \mathbf{Y}] = \mathbb{K}[A, B, Y_2, \ldots, Y_s]$ obtained by cleaning all denominators in $\mathbf{T}_-$. Observe that due to possible simplifications, $\mathbf{R}_-$ does not have to coincide with the specialization of $\mathbf{R}$ at $(z_1, \ldots, z_{r-1})$, see Lemma 3.11 below.

Since $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ is a lifting fiber for the input order, Newton iteration enables us to use it to recover $\mathbf{T}_-$. Proposition 3.6 shows that the complexity of this operation is

$$O\big((n^4 + nL)\,\mathsf{MT}(W)\,\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big);$$

the algorithm chooses one random value a' in the base field, and all choices except at most $nd^{2n}(n + 16\log d + 11)$ lead to success.

Knowing $\mathbf{T}_-$, we deduce $\mathbf{R}_-$ by a least common multiple computation and some polynomial multiplications. To be precise, we write

$$\mathbf{T}_- = (T_{-,1}, \ldots, T_{-,s}) \quad \text{and} \quad \mathbf{R}_- = (R_{-,1}, \ldots, R_{-,s}),$$

with $T_{-,i}$ in $\mathbb{K}(A)[B, Y_2, \ldots, Y_i]$ and $R_{-,i}$ in $\mathbb{K}[A, B, Y_2, \ldots, Y_i]$. For $i \leq s$, we then let $\ell_i \in \mathbb{K}[A]$ be the least common multiple of the denominators of the coefficients of $T_{-,i}$; hence, $R_{-,i} = \ell_i T_{-,i}$ and ℓ_i is the initial of $R_{-,i}$ for the input order. The following lemma gives degree bounds for the polynomials in $\mathbf{T}_-$ and $\mathbf{R}_-$; the cost of deducing $\mathbf{R}_-$ from $\mathbf{T}_-$ is given next.

Lemma 3.10. *The polynomial ℓ_i and all coefficients of $R_{-,i}$ have degree bounded by $(\deg W)$ for $i = 1$, and $2(\deg W)^2$ for $i = 2, \ldots, s$.*

Proof: This is Theorem 2 in [32]. □

Corollary 3.2. *Suppose that $\mathbf{T}_-$ is known. Then one can recover $\mathbf{R}_-$ using*

$$O(n(\deg W)\mathsf{M}\big((\deg W)^2\big)\log(\deg W))$$

operations in $\mathbb{K}$.

Proof: Let us fix $i \leq s$. Since the least common multiple of two polynomials of degree d can be computed in $O(\mathsf{M}(d)\log(d))$ base field operations, in view of the previous lemma, the cost for computing ℓ_i is in

$$O\big(d_i\,\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big).$$

Then, deducing $R_{-,i}$ requires $d_1 \cdots d_{i-1}$ multiplications in $\mathbb{K}[A]$ in degree at most $2(\deg W)^2$. Using the upper bounds $d_1 \cdots d_{i-1} \leq \deg W$ and $d_i \leq \deg W$, this shows that $R_{-,i}$ can be obtained in

$$O\big((\deg W)\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big)$$

base field operations. Summing over all i gives the result. □

To conclude this paragraph, the next lemma makes the relation between the families $\mathbf{R} = (R_1, \ldots, R_s) \subset \mathbb{K}[\mathbf{Z}][\mathbf{Y}]$ and $\mathbf{R}_- = (R_{-,1}, \ldots, R_{-,s}) \subset \mathbb{K}[A][\mathbf{Y}]$ more precise.

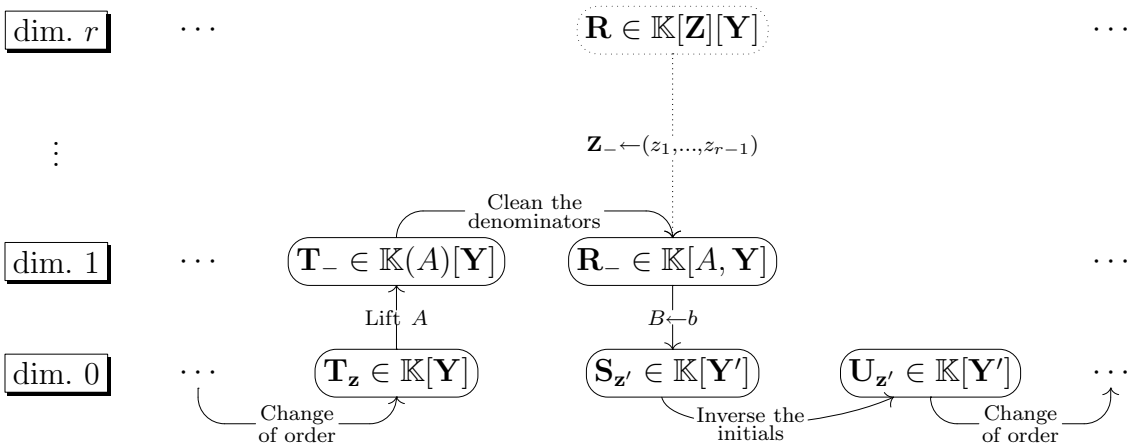

Figure 3.3: Changing the lifting fiber from $(\mathbf{z}, \mathbf{T}_\mathbf{z})$ to $(\mathbf{z}', \mathbf{U}_{\mathbf{z}'})$

Lemma 3.11. *For $i = 1, \ldots, s$, there exists $m_i \in \mathbb{K}[A]$ such that the equality*

$$R_i(z_1, \ldots, z_{r-1}, A, B, Y_2, \ldots, Y_s) = m_i R_{-,i}$$

holds.

Proof: Let $L_i \in \mathbb{K}[Z_1, \ldots, Z_{r-1}, A]$ be the least common multiple of the coefficients of T_i. Then ℓ_i divides $L_i(z_1, \ldots, z_{r-1}, A)$, and the requested equality comes by letting m_i be their quotient. □

Corollary 3.3. *Let $\mathbf{x} = (z_1, \ldots, z_{r-1}, a, b, y_2, \ldots, y_s)$ be in $\overline{\mathbb{K}}^n$. Then if the point $(a, b, y_2, \ldots, y_s)$ is a root of $\mathbf{R}_-$, the point $\mathbf{x}$ is a root of $\mathbf{R}$.*

Proof: This is a direct consequence of Lemma 3.11. □

Corollary 3.4. *Let a be in $\overline{\mathbb{K}}$, such that no denominator of $\mathbf{T}$ vanishes at $(z_1, \ldots, z_{r-1}, a)$. Then the triangular set $\mathbf{T}_-$ is well-defined, and $\mathbf{x}$ is a root of $\mathbf{R}$ if and only if $(a, b, y_2, \ldots, y_s)$ is a root of $\mathbf{R}_-$.*

Proof: The first point is immediate. The second follows by using Lemma 3.11, and observing that for $i = 1, \ldots, s$, m_i does not vanish at a, since it would imply that the denominator L_i of T_i (using the notation of the proof of Lemma 3.11) vanishes at $(z_1, \ldots, z_{r-1}, a)$. □

3.5.2 Finding the new lifting fiber

We now detail the main operations needed to obtain the lifting fiber for the new set of algebraic variables $\mathbf{Y}'$. As input, we take $\mathbf{z}_- = (z_1, \ldots, z_{r-1}) \in \mathbb{K}^{r-1}$ as well as the polynomials $\mathbf{R}_- \in \mathbb{K}[A, B, Y_2, \ldots, Y_s]$ obtained in the previous subsection.

Recall that we write $\mathbf{Z}' = (Z_1, \ldots, Z_{r-1}, B)$. Given a value $b \in \mathbb{K}$, writing $\mathbf{z}' = (z_1, \ldots, z_{r-1}, b)$, we let $\mathbf{S}_{\mathbf{z}'}$ be the polynomials in $\mathbb{K}[A, Y_2, \ldots, Y_s] = \mathbb{K}[\mathbf{Y}']$ obtained by specializing B at b in $\mathbf{R}_-$. Defining the *target* order $<'$ by

$$Z_1 < \cdots < Z_{r-1} < B < A < Y_2 < \cdots < Y_s,$$

we will now show that for most values b of B, $\mathbf{S}_{\mathbf{z}'}$ defines a lifting fiber for $(\mathbf{F}, h, <')$, where $\mathbf{F}$ denotes our initial regular chain, and h is the product of its initials.

Proposition 3.15. *There exists a non-zero polynomial $\Gamma_1 \in \mathbb{K}[\mathbf{Z}']$ of degree at most $d^n(6d^{2n} + (9d^n + 2)m^2)$, with $m = \max(n, d)$, such that, if $\Gamma_1(\mathbf{z}') \neq 0$, the following holds:*

- *$\mathbf{S}_{\mathbf{z}'}$ is a regular chain for the target order $<'$, and defines a radical ideal.*
- *Let $\mathsf{U}_{\mathbf{z}'}$ be the triangular set obtained by inverting all leading coefficients in $\mathbf{S}_{\mathbf{z}'}$. Then $(\mathbf{z}', \mathsf{U}_{\mathbf{z}'})$ is a lifting fiber for $(\mathbf{F}, h, <')$.*

Furthermore, if the previous properties hold, $\mathsf{U}_{\mathbf{z}'}$ can be deduced from $\mathbf{R}_-$ using

$$O\big(n\mathsf{M}\big((\deg W)^2\big)\log(\deg W)\big)$$

operations in $\mathbb{K}$.

Proof: By Proposition 3.3, there exists a non-zero polynomial $\Delta_{\text{lift}} \in \mathbb{K}[\mathbf{Z}]$ of degree at most $nd^n(3d^n + n + d)$, such that, for $\mathbf{z} = (z_1, \ldots, z_{r-1}, a) \in \overline{\mathbb{K}}^r$, if $\Delta_{\text{lift}}(\mathbf{z})$ is not zero, then $\mathbf{z}$ is a lifting fiber for $(\mathbf{F}, h, <)$; in particular, $\mathbf{z}$ then satisfies conditions H_1, H_2, H_3.

Lemma 3.12. *If $\mathbf{z}'$ does not belong to $\pi_{\mathbf{Z}'}(V(\mathbf{R}) \cap V(\Delta_{\text{lift}}))$, then we have the equivalence $(a, y_2, \ldots, y_s) \in V(\mathbf{S}_{\mathbf{z}'}) \iff (z_1, \ldots, z_{r-1}, a, b, y_2, \ldots, y_s) \in W$.*

Proof: Let $\mathbf{x} = (z_1, \ldots, z_{r-1}, a, b, y_2, \ldots, y_s)$ be in W. Since W is contained in $V(\mathbf{R})$, by Corollary 3.4, $(a, b, y_2, \ldots, y_s)$ is a root of $\mathbf{R}_-$. In other words, $(a, y_2, \ldots, y_s)$ is a root of $\mathbf{S}_{\mathbf{z}'}$

Conversely, let $(a, y_2, \ldots, y_s) \in \overline{\mathbb{K}}^s$ be a root of $\mathbf{S}_{\mathbf{z}'}$ and let us define the point $\mathbf{x} = (z_1, \ldots, z_{r-1}, a, b, y_2, \ldots, y_s)$. By Corollary 3.4, $\mathbf{x}$ is a root of $\mathbf{R}$, so by assumption, $\mathbf{z} = (z_1, \ldots, z_{r-1}, a)$ does not cancel Δ_{lift}. Hence, $\mathbf{z}$ satisfies conditions H_1, H_2 and H_3 for the input order $<$. We deduce by Corollary 3.4 that $\mathbf{x}$ is a root of $\mathbf{R}$. Condition H_2 then implies that $\mathbf{x}$ is in W. □

Lemma 3.13. *If $\mathbf{z}'$ does not belong to $\pi_{\mathbf{Z}'}(V(\mathbf{R}) \cap V(\Delta_{\text{lift}}))$, then $\mathbf{S}_{\mathbf{z}'}$ is a regular chain in $\mathbb{K}[\mathbf{Y}']$.*

Proof: Recall that we write $\mathbf{R}_- = (R_{-,1}, \ldots, R_{-,s})$, where R_1 is in $\mathbb{K}[A, B]$ and R_i is in $\mathbb{K}[A, B, Y_2, \ldots, Y_i]$ for $i > 1$. Recall also that by construction, the initial ℓ_i of $R_{-,i}$ is the least common multiple of the denominators of the coefficients of T_i; in particular, it is in $\mathbb{K}[A]$. By construction, the ith polynomial in $\mathbf{S}_{\mathbf{z}'}$ is $R_{-.i}(A, b, Y_2, \ldots, Y_s)$, so for $i > 1$, its initial is ℓ_i as well.

By assumption, none of the points in $V(\mathbf{R}_-) \cap V(B - b)$ cancels Δ_{lift}. Hence, by definition of Δ_{lift}, none of the denominators of $\mathbf{T}$ vanishes on $V(\mathbf{R}_-) \cap V(B - b)$. This implies that no polynomial ℓ_i vanishes on $V(\mathbf{R}_-) \cap V(B - b)$, that is, on $V(\mathbf{S}_{\mathbf{z}'})$. Hence, ℓ_i is a zero-divizor modulo the $i - 1$ first polynomials in $\mathbf{S}_{\mathbf{z}'}$; by defintion, it is a regular chain. □

Lemma 3.14. *Let $D \in \mathbb{K}[\mathbf{Z}']$ be the resultant of R_1 and $\partial R_1/\partial A$ with respect to A. If $\mathbf{z}'$ does not belong to $\pi_{\mathbf{Z}'}(V(\mathbf{R}) \cap V(D\Delta_{\text{lift}}))$, then $\mathbf{S}_{\mathbf{z}'}$ defines a radical ideal in $\mathbb{K}[\mathbf{Y}']$.*

Proof: Let $(a, y_2, \ldots, y_s) \in \overline{\mathbb{K}}^s$ be a root of $\mathbf{S}_{\mathbf{z}'}$, and let us write the polynomials of $\mathbf{S}_{\mathbf{z}'}$ as $(S_{\mathbf{z}',1}, \ldots, S_{\mathbf{z}',s}) \subset \mathbb{K}[A, Y_2, \ldots, Y_s]$. We will prove that none of the partial derivatives $\partial S_{\mathbf{z}',1}/\partial A$ and $\partial S_{\mathbf{z}',i}/\partial Y_i$, for $i > 2$, vanishes at $(a, y_2, \ldots, y_s)$, which is enough to conclude by the Jacobian criterion.

Let us define $\mathbf{z} = (z_1, \ldots, z_{r-1}, a)$ and consider the triangular set $\mathbf{T}_{\mathbf{z}} \subset \overline{\mathbb{K}}[B, Y_2, \ldots, Y_s]$. By assumption on $\mathbf{z}'$, $\mathbf{T}_{\mathbf{z}}$ is well-defined and generates a radical ideal in $\overline{\mathbb{K}}[\mathbf{Y}]$. In other

words, none of the partial derivatives $\partial T_{\mathbf{z},i}/\partial Y_i$ vanishes on the zero-set of $\mathbf{T}_\mathbf{z}$. Now, the point $\mathbf{x} = (z_1, \ldots, z_{r-1}, a, b, y_2, \ldots, y_s) \in \overline{\mathbb{K}}^n$ is in the zero-set of $\mathbf{T}_\mathbf{z}$, and at this point, the values of the partial derivatives $\partial S_{\mathbf{z}',i}/\partial Y_i$ and $\partial T_{\mathbf{z},i}/\partial Y_i$ coincide, up to the non-zero factor $\ell_i(a)$. Hence, none of the partial derivatives $\partial S_{\mathbf{z}',i}/\partial Y_i$ is zero at $(a, y_2, \ldots, y_s)$ for $i > 2$.

It remains to deal with the partial derivative $\partial S_{\mathbf{z}',1}/\partial A$ of the first polynomial $S_{\mathbf{z}',1}$. Since $\mathbf{z}_- = (z_1, \ldots, z_{r-1})$ does not cancel the leading coefficient of R_1, if $D(\mathbf{z}')$ is not zero, then Lemma 3.11 shows that $R_{-,1}(z_1, \ldots, z_{r-1}, A, b) = S_{\mathbf{z}',1}(A)$ has no multiple root, which is what we wanted to prove. □

We can now prove Proposition 3.15. Remark that the first polynomial R_1 in $\mathbf{R}$ belongs to $\mathbb{K}[\mathbf{Z}, B]$. By the definition of $\mathbf{R}$, it admits no factor in $\mathbb{K}[\mathbf{Z}]$, and has total degree at most $(\deg W)$. In particular, its resultant with Δ_{lift} with respect to A is a non-zero polynomial C in $\mathbb{K}[Z_1, \ldots, Z_{r-1}, B] = \mathbb{K}[\mathbf{Z}']$. All points $\mathbf{z}' = (z_1, \ldots, z_{r-1}, b)$ which belong to $\pi_{\mathbf{Z}'}(V(\mathbf{R}) \cap V(\Delta_{\text{lift}}))$ cancel this resultant C, whose degree is at most $(2 \deg W \deg \Delta_{\text{lift}})$.

We continue by considering the resultant D appearing in the last lemma. Recall that the polynomial $R_1 \in \mathbb{K}[Z_1, \ldots, Z_{r-1}, A, B]$ defines the closure of $\pi_{Z_1,\ldots,Z_{r-1},A,B}(W)$. Then, R_1 has non-zero degree in A, since otherwise $\mathbf{Z}' = Z_1, \ldots, Z_{r-1}, B$ would not be a set of free variables for W. Furthermore, R_1 is irreducible in $\mathbb{K}[Z_1, \ldots, Z_{r-1}, A, B]$; hence, its discriminant D is non-zero, of degree at most $2(\deg R_1)^2$. Using again Theorem 2 in [32], we get that the degree of R_1 is upper-bounded by $(\deg W)$, so that the degree of D is at most $2(\deg W)^2$.

To conclude the probability analysis, let $\Delta'_{\text{lift}} \in \mathbb{K}[\mathbf{Z}']$ be the polynomial associated by Proposition 3.3 to the projection $\pi_{\mathbf{Z}'}$, so that if $\Delta'_{\text{lift}}(\mathbf{z}')$ is not zero, then $\mathbf{z}'$ satisfies the lifting conditions $\mathsf{H}_1, \ldots, \mathsf{H}_4$ for the system $(\mathbf{F}, h, <')$. We then take $\Gamma_1 = CD\Delta'_{\text{lift}}$, which is non-zero and of the requested degree. Then, if $\mathbf{z}'$ does not cancel Γ_1, $\mathbf{z}'$ satisfies the lifting conditions. Besides, by the previous lemmas, the monic form $\mathsf{U}_{\mathbf{z}'}$ of $\mathbf{S}_{\mathbf{z}'}$ is a triangular set, defining a radical ideal, and having for zero-set $\{\mathbf{z}'\} \times W_{\mathbf{z}'}$; this implies that $(\mathbf{z}', \mathsf{U}_{\mathbf{z}'})$ is a lifting fiber for $(\mathbf{F}, h, <')$.

The final part of the proof is the complexity analysis. As input, recall that we receive the polynomials $\mathbf{R}_-$ in $\mathbb{K}[A, B, Y_2, \ldots, Y_s]$ obtained in the previous subsection. The first step consists in specializing B at b in these polynomials: this can be done in time $O(\deg W)$. Next, we invert all initials $\ell_i \in \mathbb{K}[A]$ modulo the univariate polynomial $S_{\mathbf{z}',1} \in \mathbb{K}[A]$. All initials ℓ_i have degree at most $2(\deg W)^2$ and can be inverted modulo $S_{\mathbf{z}',1}$, so this operation takes $O(n\mathsf{M}((\deg W)^2)\log(\deg W))$ operations in the base field. This finishes the proof of Proposition 3.15. □

3.5.3 Proof of Proposition 3.14

We conclude this section with the proof of Proposition 3.14 announced in the introduction of this section. The complexity estimate follows from taking the sum of all contributions seen previously in this section: using the fact that $\mathsf{MT}(W)$ is at least linear in $\deg W$, the dominant term comes from the lifting step of Subsection 3.5.1.

The probability analysis comes easily too: a first source of error is in the choice of a value a' used to stop Newton's iteration; the second one comes from the possibility that $(z_1, \ldots, z_{r-1}, b)$ cancels the polynomial $\Gamma_1 \in \mathbb{K}[Z_1, \ldots, Z_{r-1}, B]$ of the previous proposition. Since the values a' that provoke error are in finite number, there is a non-zero polynomial $\Gamma_2 \in \mathbb{K}[A']$ having these values as roots. It then suffices to let $\Delta_{\text{exchange}} = \Gamma_1\Gamma_2 \in$

$\mathbb{K}[Z_1, \ldots, Z_{r-1}, A', B]$; the degree bound comes easily after a few simplifications.

3.6 Proof of Theorem 3.1

We finally turn to the proof of Theorem 3.1. Our analysis will use the so-called Zippel-Schwartz lemma [106, 123]: if P is a non-zero polynomial in $\mathbb{K}[V_1, \ldots, V_t]$ and if S is a finite subset of $\mathbb{K}$, then P has at most $(\deg P)|S|^{t-1}$ roots in S^t.

The algorithm first chooses a specialization value $\mathbf{z} = (z_1, \ldots, z_r)$ for the free variables $\mathbf{Z}$ of the input regular chain $\mathbf{F}$; using those, we determine the exchange data $\mathbf{Y}_0, \ldots, \mathbf{Y}_s$. The cost and probability analysis of this first step are given in Proposition 3.9.

In the second step of the algorithm, we use the exchange data to compute a sequence of lifting fibers, calling at most s times the subroutine described in Proposition 3.14; we then use a last change of order in dimension zero to order the algebraic variables $\mathbf{Y}_s$ in the final lifting fiber according to the target order $<'$. The complexity analysis of Proposition 3.14 dominates all other ones and establishes the cost reported in Theorem 3.1. We conclude with the probability analysis.

Without loss of generality, we can suppose that for all i, $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ do actually differ, so that we need to perform exactly s times the operations described in the last section (if $\mathbf{Y}_i$ and $\mathbf{Y}_{i+1}$ coincide, there is nothing to do). Hence, the algorithm will chose $2s$ values in the base field: s of them, written $b_1, \ldots, b_s$ to match the notation of Proposition 3.14, will be used as the specialization values in the sequence of lifting fibers, and the s remaining ones, written $a'_1, \ldots, a'_s$, are used in the stop criterion used in the successive Newton lifting processes.

Suppose thus that $z_1, \ldots, z_r$, $b_1, \ldots, b_s$ and $a'_1, \ldots, a'_s$ are chosen uniformly at random in a finite subset S of $\mathbb{K}$; observe that the size of the sample set is then $|S|^{n+s}$. To ensure success, we first require that $z_1, \ldots, z_r$ do not cancel the polynomial Δ_{lin} of Proposition 3.9: by Zippel-Schwartz's lemma, this discriminates at most $n(2d)^{n+1}|S|^{n+s-1}$ elements in S^{n+s}; for all remaining points, we obtain the correct exchange data.

In the second step, we do s calls to the algorithm presented in Proposition 3.14. For $i \leq s$, let $(Z_{i,1}, \ldots, Z_{i,r-1}, Z_{i,r}) \subset (Z_1, \ldots, Z_r, B_1, \ldots, B_{i-1})$ be the indeterminates that give the coordinates of the specialization value $(z_{i,1}, \ldots, z_{i,r})$ used in the ith lifting fiber. The ith call to Proposition 3.14 involves replacing one of these indeterminates, say $Z_{i,r}$ for definiteness, by B_i, and do the analogous replacement in the specialization value; we use the value a'_i along the way to stop Newton's iteration.

Hence, by Proposition 3.14, there exists a non-zero polynomial $\Delta_{\text{exchange},i}$ such that if $(z_{i,1}, \ldots, z_{i,r-1}, b_i, a'_i)$ is non zero, the ith step succeeds. Using Zippel-Schwartz's lemma, the degree bound given in that proposition shows that this discriminates at most $2d^n(3d^{2n} + m((6 + 13m)d^n + m))|S|^{n+s-1}$ points in S^{n+s}, writing $m = \max(n, d)$.

Summing all previous estimates concludes the proof of Theorem 3.1.

3.7 Conclusions and future work

We have presented an algorithm to perform change of order on regular chains in positive dimension, that reduces mostly to a well-identified set of basic operations: lifting techniques and change of order in dimension zero. As output, we compute a lifting fiber for the target

regular chain, which enables us to maintain a polynomial complexity, while allowing for the recovery of the full "expanded" representation of the target if needed. The algorithm is probabilistic, and we provide a fine control on the probability of failure.

There is an implementation in Maple for which I have not participated, due to X. Jin, É Schost and M. Moreno Maza; it is now part of the `RegularChains` library [80]. As of now, not all of the techniques presented here are implemented: for instance, we still use classical arithmetic to perform operations modulo a triangular set. It is expected to improve on this situation in the near future. More work is also planned to obtain an efficient lower-level implementation in the Aldor language, following the first experiments reported in [43]; in such an environment, it is expected to make full use of the algorithms described here.

At the conceptual level, our next objective is to lift the primality assumption. Moving to the more general situation of *equidimensional* vaieties already raises several difficulty, since we will then have to split our object into its *equiprojectable components* [31]. Then, the study of the possible degeneracies promises to become much more involved, but should still follow the mains ideas presented here.

As was mentioned in the introduction, another of our projects consists in improving the multivariate Newton iteration that takes place if one wants to recover the full multivariate representation of the target regular chain. At the moment, multivariate power series multiplication remains a difficult problem, with no quasi-linear solution known in general. As a workaround, sparse lifting and interpolation techniques are expected to improve on the current generalist approach, inherited from [103].

Chapter 4

Lifting techniques for triangular decompositions

This chapter presents lifting techniques for triangular decompositions of zero-dimensional varieties, that extend the range of the previous methods. This work has been published in [31] by Y. Xie, W. Wu, M. Moreno Maza, E. Schost and myself; what is presented here is a slighlty ameliorated version, while not up-to-date (the "Split-and-Merge" algorithm Section 4.2 does not use new results of dynamic evaluation of Chapter 5). We discuss complexity aspects, and report on an implementation in Maple 10, realised essentially by the four other co-authors. The theoretical results are comforted by these experiments.

4.1 Introduction

Modular methods for computing polynomial GCDs and solving linear algebra problems have been well-developed for several decades, see [117] and the references therein. Without these methods, the range of problems accessible to symbolic computations would be dramatically limited. Such methods, in particular Hensel lifting, also apply to solving polynomial systems. Standard applications are the resolution of systems over $\mathbb{Q}$ after specialization at a prime, and over the rational function field $k(Y_1, \ldots, Y_m)$ after specialization at a point $(y_1, \ldots, y_m)$. These methods have already been put to use for Gröbner bases [114, 5] and primitive element representations, starting from [50, 52, 92]. *Triangular decompositions* of algebraic varieties are well-suited to many practical problems: see some examples in [40, 44, 105]. It permits to split the problem into smaller systems, with less coefficients swell than lexicaographic Gröbner bases, whereas they also have the elimination properties. In addition, these techniques are commonly used in differential algebra [19, 59]. Triangular decompositions of polynomial systems can be obtained by various algorithms [63, 72, 88] but none of them uses modular computations, restricting their practical efficiency. Our goal in this chapter is to discuss such techniques, extending the preliminary results of [105].

Framework We consider 0-dimensional varieties defined over $\mathbb{Q}$. Let thus $\mathbf{F} = F_1, \ldots, F_n$ be a polynomial system in $\mathbb{Z}[X_1, \ldots, X_n]$. Since we have in mind to apply Hensel lifting techniques, we will only consider the *simple roots* of $\mathbf{F}$, that is, those where the Jacobian determinant J of $\mathbf{F}$ does not vanish. We write $\mathcal{Z}(\mathbf{F})$ for this set of points; by the Jacobian

criterion [38, Ch. 16], $\mathcal{Z}(\mathbf{F})$ is finite, even though the whole zero-set of $\mathbf{F}$, written $V(\mathbf{F})$, may have higher dimension.

We want to triangulate the system $\mathbf{F}$ following the hereunder scheme:

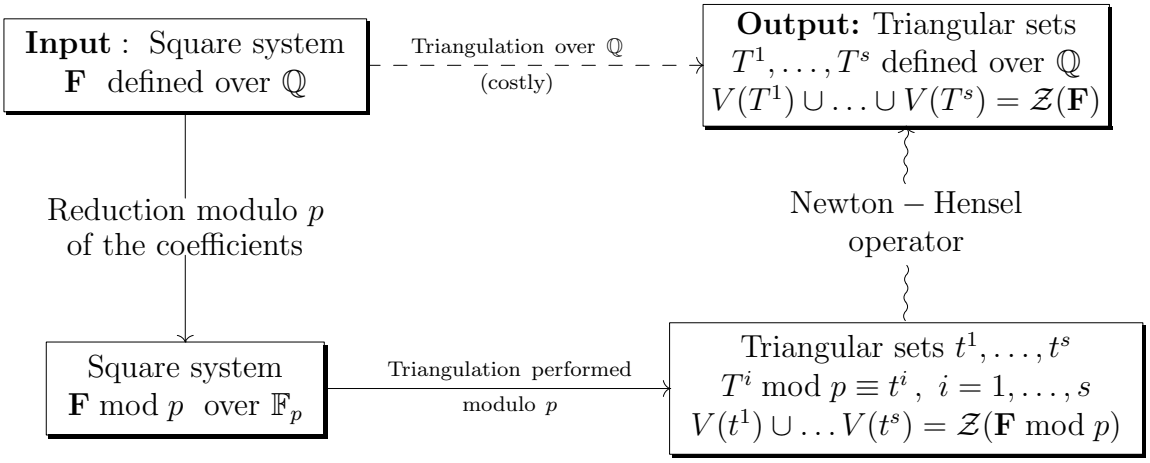

Figure 4.1: Prototype of a modular method modulo p using Newton-Hensel technique

We note the analogy with the positive dimension situation (Cf. Fig 4.1), where instead of reducing the coefficients modulo p, we specialize the free variables at a chosen value (we refer to the discussion for lifting techniques in Ch. 1, § 1.4, for both approaches).

The triangulation algorithm won't be discussed here, and we will fix one that works over $\mathbb{Q}$ and $\mathbb{F}_p$. The reader can refer to the preliminaries chapter, Ch. 1, § 1.1.3 for mention of several algorithms. What will be studied is the process of *reduction* and *lifting*. The new difficulty arisen here comparing to previous modular algorithms, is the compatibility of the triangular sets obtained modulo p and over $\mathbb{Q}$. Indeed, extra factorizations or recombinations can occur modulo p. Thus, we have no guarantee that there exist triangular sets $T^1, \ldots, T^s$ defined over $\mathbb{Q}$, that describe $\mathcal{Z}(\mathbf{F})$, and with $t^1, \ldots, t^s$ as modular images. Furthermore, if we assume no control over the modular resolution process, there is little hope of obtaining a quantification of primes p of "bad" reduction.

Example Consider for instance the variety $V \subset \mathbb{C}^2$ defined by the square polynomial system

$$\mathbf{F} \left| \begin{array}{l} F_2(X_1, X_2) = 326X_1 - 10X_2^6 + 51X_2^5 + 17X_2^4 + 306X_2^2 + 102X_2 + 34, \\ F_1(X_1, X_2) = X_2^7 + 6X_2^4 + 2X_2^3 + 12 \end{array} \right.$$

For the order $X_2 > X_1$, the only possible description of V by triangular sets with rational coefficients corresponds to its irreducible decomposition, that has two components A and B:

$$A := V(T^1) \left| \begin{array}{l} T_2^1(X_1, X_2) = X_2^3 + 6 \\ T_1^1(X_1) = X_1 - 1 \end{array} \right. \qquad B := V(T^2) \left| \begin{array}{l} T_2^2(X_1, X_2) = X_2^2 + X_1 \\ T_1^2(X_1) = X_1^2 + 2 \end{array} \right.$$

As illustrate in Figure 4.2 the prime $p = 7$ allows several decompositions, one of which (consisting of C and D on the picture) leads to an incompatibility between the decomposition

obtained over $\mathbb{Q}$ (named A and B). In fact, the following triangular sets describe the zeros of $\mathbf{F}$ mod 7:

$$C := V(t^1) \left| \begin{array}{l} t_2^1(X_1, X_2) = X_2^2 + 6X_2X_1^2 + 2X_2 + X_1 \\ t_1^1(X_1) = X_1^3 + 6X_1^2 + 5X_1 + 2 \end{array} \right. \qquad D := V(t^2) \left| \begin{array}{l} t_2^2(X_1, X_2) = X_2 + 6 \\ t_1^2(X_1) = X_1 + 6 \end{array} \right. ,$$

which are not the reduction modulo 7 of T^1 and T^2:

Figure 4.2: Incompatible triangular decompositions over $\mathbb{Q}$ and modulo 7

A lifting algorithm should discard t^1 and t^2, and replace them by the better choice:

$$t'^1 \left| \begin{array}{l} t'^1_2(X_1, X_2) = X_2^3 + 6 \\ t'^1_1(X_1) = X_1 + 6 \end{array} \right. \quad \text{and} \quad t'^2 \left| \begin{array}{l} t'^2_2(X_1, X_2) = X_2^2 + X_1 \\ t'^2_1(X_1) = X_1^2 + 2 \end{array} \right.$$

which are the reduction of T^1 and T^2 modulo 7. In [105], this difficulty was bypassed by restricting to *equiprojectable* varieties, *i.e.* varieties defined by a single triangular set, where no such ambiguity occurs. However, as this example shows, this assumption discards simple cases. Our main concern is to lift this limitation, thus extending these techniques to handle *triangular decompositions*.

Main result 1 Our answer consists in using a canonical decomposition of a 0-dimensional variety V, its *equiprojectable decomposition*, described as follows. Consider the map

$$\begin{array}{rcl} \pi : V \subset \mathbb{A}_k^n & \longrightarrow & \mathbb{A}_k^{n-1} \\ (x_1, \ldots, x_n) & \longmapsto & (x_1, \ldots, x_{n-1}) \end{array}$$

that forgets the last coordinate. The definition relies on the cardinality of successive projections fibers. We introduce the following cardinality function, attached to the projection π:

$$\begin{array}{rcl} N : V & \longrightarrow & \mathbb{N} \\ x & \longmapsto & \#\pi^{-1}(\{\pi(x)\}) \end{array}$$

that is, $N(x)$ the number of points lying in the same π-fiber as x. Then, we split V into the disjoint union $V_1 \cup \cdots \cup V_d$, where for all $i = 1, \ldots, d$, V_i equals $N^{-1}(\{i\})$, *i.e.*, the set of points $x \in V$ where $N(x) = i$. This splitting process is applied recursively to all $V_1, \ldots, V_d$, taking into account the fibers of the successive projections $\mathbb{A}_k^i \to \mathbb{A}_k^i$, for $i = n-1, \ldots, 1$. In the end, we obtain a family of pairwise disjoint, equiprojectable varieties, whose reunion equals V, which form the equiprojectable decomposition of V. As requested, each of them is representable by a triangular set with coefficients in the definition field of V.

The above algorithm sketch is thus improved by applying lifting only after computing the equiprojectable decomposition of the modular output. Theorem 4.1 shows how to control the primes of bad reductions for the equiprojectable decomposition, thus overcoming the limitation that we pointed out previously. In what follows, the height of $x \in \mathbb{Z}$ is defined as $h(x) = \log|x|$; the height of $f \in \mathbb{Z}[X_1, \ldots, X_n]$ is the maximum of the heights of its coefficients; that of $p/q \in \mathbb{Q}$, with $\gcd(p, q) = 1$, is $\max\{h(p), h(q)\}$.

Theorem 4.1. *Let $F_1, \ldots, F_n$ have total degree bounded by d and height bounded by h. Let $T^1, \ldots, T^s$ be the triangular description of the equiprojectable decomposition of $\mathcal{Z}(\mathbf{F})$. There exists $A \in \mathbb{N} - \{0\}$, with $h(A) \leq \mathfrak{a}(n, d, h)$, and, for $n \geq 2$,*

$$\mathfrak{a}(n, d, h) = n^2 d^{2n}\Big(2h(d+1) + \log(n+2)(3d+6) + n\log(d)(4+2d) + 6 + 5d\Big),$$

with the following property:

If a prime p does not divide A, then p cancels none of the denominators of the coefficients of $T^1, \ldots, T^s$, and these triangular sets reduced mod p define the equiprojectable decomposition of $\mathcal{Z}(\mathbf{F} \bmod p)$.

Thus, the set of bad primes is finite and we have an explicit control on its size. Since we have to avoid some "discriminant locus", it is natural, and probably unavoidable, that the bound should involve the square of the Bézout number; It largely dominates the growth of the function $\mathfrak{a}(n, d, h)$.

Main result 2 A second question is the coefficient size of the output. In what follows, we write $\deg V$ and $h(V)$ for the *degree* and *height* of a 0-dimensional variety V defined over $\mathbb{Q}$: the former denotes its number of points, and the later estimates its arithmetic complexity (Cf. §. 1.2.2) Let then $T^1, \ldots, T^s$ be the triangular sets that describe the equiprojectable decomposition of $\mathcal{Z} = \mathcal{Z}(\mathbf{F})$. Theorem 2.7 p. 86 shows that all coefficients in $T^1, \ldots, T^s$ have height at most in $O\big(\log(n)\deg(\mathcal{Z})^2 + \deg(\mathcal{Z})h(\mathcal{Z})\big)$. Using the alternative representation denoted by $N^1, \ldots, N^s$ in Chapter 2, Definition 2.3 where for $i \leq s$, $N^i = N^i_1, \ldots, N^i_n$, $N^i_\ell \in k[X_1, \ldots, X_\ell]$, and defined as follows:

$$D^i_1 = 1\,, \quad N^i_1 = T^i_1\,, \quad \text{for } 2 \leq \ell \leq n \quad \text{and for } 1 \leq i \leq s,$$

$$D^i_\ell = \prod_{1 \leq j \leq \ell-1} \frac{\partial T^i_j}{\partial X_j} \quad \text{and} \quad N^i_\ell = D^i_\ell T^i_\ell \bmod (T^i_1, \ldots, T^i_{\ell-1}),$$

permits to reduce the height to $O\big(\log(n)\deg\mathcal{Z} + h(\mathcal{Z})\big)$. Since $T^1, \ldots, T^s$ are easily recovered from $N^1, \ldots, N^s$, our algorithm will compute the latter, their height bounds being the better.

Theorem 4.2 below states our main result regarding lifting techniques for triangular decompositions; in what follows, we say that an algorithm has a *quasi-linear* complexity in terms of some parameters if its complexity is linear in all of these parameters, up to polylogarithmic factors. We need the following assumptions:

- For any $C \in \mathbb{N}$, let $\Gamma(C)$ be the sets of primes in $[C+1, \ldots, 2C]$. We assume the existence of an oracle O_1 which, for any $C \in \mathbb{N}$, outputs a random prime in $\Gamma(C)$, with the uniform distribution.

- We assume the existence of an oracle O_2, which, given a system $\mathbf{F}$ and a prime p, outputs the representation of the equiprojectable decomposition of $\mathcal{Z}(\mathbf{F} \bmod p)$ by means of triangular sets. We give in Section 4.2 an algorithm to convert any triangular decomposition of $\mathcal{Z}(\mathbf{F} \bmod p)$ to the equiprojectable one; its complexity analysis is subject of current research (note after revision of the manuscript: this has lead to the study of the complexity of the D5 principle, tackled in Chapter 5, but yet not applied to solve this complexity study).

- For $\mathbf{F}$ as in Theorem 4.1, we write:

  $\mathfrak{a}_{\mathbf{F}} = \mathfrak{a}(n,d,h)$, bounds the height of the number A of Theorem 4.1
  $\mathfrak{h}_{\mathbf{F}} = nd^n(h + 2\log(n+1) + 7)$, bounds the height of the coefficients of the output polynomials $N^1, \ldots, N^s$
  $\mathfrak{b}_{\mathbf{F}} = 5(\mathfrak{h}_{\mathbf{F}} + 1)\log(2\mathfrak{h}_{\mathbf{F}} + 1)$, bounds occuring in the probability analysis of § 4.4

  The input system is given by a straight-line program of size L, with constants of height at most h_L.

- $\mathsf{C} \in \mathbb{N}$ is such that for any ring R, any $d \geq 1$ and monic $t \in R[X]$ of degree d, all operations $(+,-,\times)$ in $R[X]/t$ can be computed in $\mathsf{C}d\log(d)\log\log(d)$ operations in R [117, Ch. 8,9]. Then all operations $(+,-,\times)$ modulo a triangular set T in n variables can be done in quasi-linear complexity in C^n and $\deg V(T)$ (this result is precisely discussed in Ch. 5, Prop. 5.2).

Theorem 4.2. *Let $\varepsilon > 0$. There exists an algorithm which, given $\mathbf{F}$, satisfying*

$$\frac{4\mathfrak{a}_{\mathbf{F}} + 2\mathfrak{b}_{\mathbf{F}}}{\varepsilon} + 1 < \frac{1}{2}\exp(2\mathfrak{h}_{\mathbf{F}} + 1),$$

computes $N^1, \ldots, N^s$ defined above. The algorithm uses two calls to O_1 with

$$C = 4\mathfrak{a}_{\mathbf{F}} + 2\mathfrak{b}_{\mathbf{F}}/\varepsilon,$$

two calls to O_2 with p in $[C+1, \ldots, 2C]$, and its bit complexity is quasi-linear in

$$L, h_L, d, \log h, \mathsf{C}^n, \deg \mathcal{Z}, (\deg \mathcal{Z} + h(\mathcal{Z})), |\log \varepsilon|.$$

The algorithm is probabilistic, with success probability greater than $1 - \varepsilon$.

To illustrate these estimates, suppose e.g. that we have $n = 10, d = 4, h = 100$, hence potentially 1048576 solutions; to ensure a success probability of 99%, the primes should have only about 20 decimal digits, hence can be generated without difficulty. Thus, even for such "large" systems, our results are quite manageable. Besides, computing the polynomials N^i instead of T^i enables us to benefit from their improved height bounds.

In the sequel, we use the following notation. For $n \in \mathbb{N}$, for $1 \leq j \leq i \leq n$ and any field k, we define:

$$\begin{array}{rcl} \pi_j^i \ : \ \mathbb{A}_k^i & \longrightarrow & \mathbb{A}_k^j \\ (x_1, \ldots, x_i) & \longmapsto & (x_1, \ldots, x_j). \end{array}$$

The cardinality of a finite set G is written $\#G$.

4.2 Split-and-Merge algorithm

We start by reviewing the notion of equiprojectable decomposition of a 0-dimensional variety V. Then, in preparation for the modular algorithm of Section 4.4, we present an algorithm for computing this decomposition, given an arbitrary triangular decomposition of V. We call it *Split-and-Merge*, after its two phases: the *splitting* of what we call *critical pairs* (which is achieved by GCD computations) and the *merging* of what we call *solvable families* (which is performed by Chinese remaindering). The complexity analysis of the Split-and-Merge algorithm is work in progress, deducible from results of Chapter 5. We believe that suitable improvements of the Split-and-Merge algorithm can run in quasi-linear time in the degree of V.

Let k be a perfect field and $\overline{k}$ one of its algebraic closures. Let $V \subset \mathbb{A}^n_{\overline{k}}$ variety V can be decomposed as the disjoint union of equiprojectable ones in possibly several ways. Any such decomposition amounts to represent V as the disjoint union of the zeros of some triangular sets. The equiprojectable decomposition is a canonical way of doing so, defined by combinatorial means (see Figure 4.3).

Equiprojectable decomposition. Let first W be a 0-dimensional variety in $\mathbb{A}^i_{\overline{k}}$, for some $1 \leq i \leq n$. For x in $\mathbb{A}^{i-1}_{\overline{k}}$, we define the preimage

$$\mu(x, W) = (\pi^i_{i-1})^{-1}(x) \cap W;$$

for any $d \geq 1$, we can then define

$$A(d, W) = \left\{ x \in W \mid \#\mu(\pi^i_{i-1}(x), W) = d \right\}.$$

Thus, x is in $A(d, W)$ if W contains exactly d points x' such that $\pi^i_{i-1}(x) = \pi^i_{i-1}(x')$ holds. Only finitely many of the $A(d, W)$ are not empty and the non-empty ones form a partition of W. Let $1 \leq i \leq n$. Writing $W = \pi^n_i(V)$, we define

$$B(i, d, V) = \left\{ x \in V \mid \pi^n_i(x) \in A(d, W) \right\}.$$

Thus, $B(i, d, V)$ is the preimage of $A(d, W)$ in V, so these sets form a partition of V. If V is i-equiprojectable, then all $B(i, d, V)$ are $(i-1)$-equiprojectable. We then define inductively $B(V) = V$, and, for:

$$1 < i \leq n, \quad B(d_i, \ldots, d_n, V) = B(i, d_i, B(d_{i+1}, \ldots, d_n, V)).$$

All $B(d_i, \ldots, d_n, V)$ are $(i-1)$-equiprojectable, only finitely many of them are not empty, and the non-empty ones form a partition of V.

The *equiprojectable decomposition* of V is its partition into the family of all non-empty $B(d_2, \ldots, d_n, V)$ (see illustration of this definition on Figure 4.3). All these sets being equiprojectable, they are defined by triangular sets. Note that we have not proved yet that the $B(d_2, \ldots, d_n, V)$ are defined over the same field as V. This will come as a by-product of the algorithms of this section. To do so, we introduce now the notions of *critical pair* and *solvable pair*.

Critical and solvable pairs. Let $T \neq T'$ be two triangular sets. The least integer ℓ such that $T_\ell \neq T'_\ell$ is called the *level* of the pair T, T'.

$$\text{If } \ell = 1 \text{ let } \quad K_\ell := k, \quad \text{otherwise} \quad K_\ell := k[X_1, \ldots, X_{\ell-1}]/(T_1, \ldots, T_{\ell-1}).$$

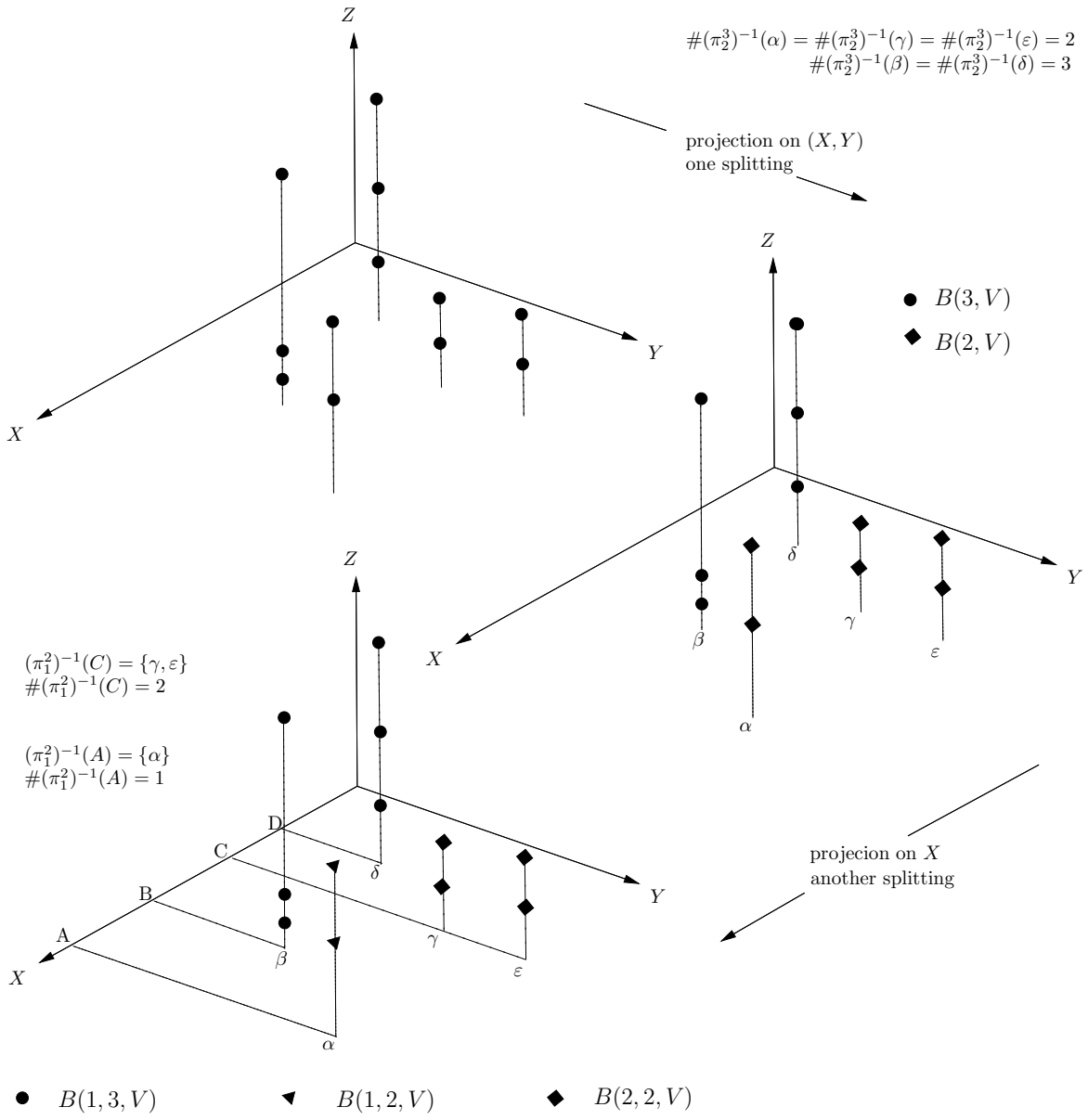

Figure 4.3: Recursive definition of the equiprojectable decomposition

Since a triangular set generates a radical ideal, the residue class ring K_ℓ is a direct product of fields. Therefore, every pair of univariate polynomials with coefficients in K_ℓ has a GCD in the sense of [89]. The pair T, T' is *critical* if T_ℓ and T'_ℓ are not relatively prime in $K_\ell[X_\ell]$. If T, T' is not critical, it is *certified* if $U, U' \in K_\ell[X_\ell]$ such that $UT_\ell + U'T'_\ell = 1$ are known. The pair T, T' is *solvable* if it is not critical and if for all $\ell < j \leq n$ we have $\deg_{X_j} T_j = \deg_{X_j} T'_j$.

Introducing the notion of a certified solvable pair is motivated by efficiency considerations. Indeed, during the splitting step, solvable pairs are discovered. Then, during the merging step, the Bézout coefficients U, U' of these solvable pairs will be needed for Chinese Remaindering.

Solvable families. We extend the notion of *solvability* from a pair to a family of triangular sets. A family $\mathfrak{T}$ of triangular sets is *solvable* (resp. *certified solvable*) at level ℓ if every pair $\{T, T'\}$ of elements of $\mathfrak{T}$ is solvable (resp. certified solvable) of level ℓ.

The following proposition shows how to recombine such families. When this is the case, we say that all T in $\mathfrak{T}$ *divide* S. In what follows, we write $V(\mathfrak{T})$ for $\cup_{T \in \mathfrak{T}} V(T)$.

Proposition 4.1. *If $\mathfrak{T}$ is certified solvable at level ℓ, one can compute a triangular set S such that $V(S) = V(\mathfrak{T})$, using only multiplications in $K_\ell[X_\ell]$.*

Proof: First, we assume that $\mathfrak{T}$ consists of the pair $\{T, T'\}$. We construct S as follows. We set $S_i = T_i$ for $1 \leq i < \ell$ and $S_\ell = T_\ell T'_\ell$. Let $\ell < i \leq n$. For computing S_i, we see T_i and T'_i in $K_\ell[X_\ell][X_{\ell+1}, \ldots, X_i]$. We apply Chinese remaindering to the coefficients in T_i and T'_i of each monomial in $X_{\ell+1}, \ldots, X_i$ occurring in T_i or T'_i: since the Bézout coefficients U, U' for T_ℓ, T'_ℓ are known, this can be done using multiplications in $K_\ell[X_\ell]$ only.

$$\begin{array}{l} (K_\ell[X_\ell]/(T_\ell))\,[X_{\ell+1}, \ldots, X_i] \quad \times \quad (K_\ell[X_\ell]/(T'_\ell))\,[X_{\ell+1}, \ldots, X_i] \quad \simeq (K_\ell[X_\ell]/(S_\ell))\,[X_{\ell+1}, \ldots, X_i] \\ \left(\displaystyle\sum_{\alpha \in \mathbb{N}^{\mathbf{i}-\ell}} c_\alpha X_{\ell+1}^{\alpha_1} \cdots X_i^{\alpha_{i-\ell}} \quad , \quad \sum_{\alpha \in \mathbb{N}^{\mathbf{i}-\ell}} c'_\alpha X_{\ell+1}^{\alpha_1} \cdots X_i^{\alpha_{i-\ell}} \right) \quad \longrightarrow \\ \qquad\qquad \displaystyle\sum_{\alpha \in \mathbb{N}^{\mathbf{i}-\ell}} (c_\alpha U' T'_\ell + c'_\alpha U T_\ell \bmod S_\ell) X_{\ell+1}^{\alpha_1} \cdots X_i^{\alpha_{i-\ell}} \end{array}$$

It follows from the Chinese Remaindering Theorem the following equalities of ideals of $K_\ell[X_\ell, X_{\ell+1}]$:

$$(S_{\ell+1}, T_\ell) = (T_{\ell+1}, T_\ell) \qquad \text{and} \qquad (S_{\ell+1}, T'_\ell) = (T'_{\ell+1}, T'_\ell),$$

proving that $(S_{\ell+1}, S_\ell) = (T_{\ell+1}, T_\ell) \cap (T'_{\ell+1}, T'_\ell)$. By induction, we get $(S_n, \ldots, S_\ell) = (T_n, \ldots, T_\ell) \cap (T'_n, \ldots, T'_\ell)$ over K_ℓ. Since T and T' are of level ℓ, this yields $(S) = (T) \cap (T')$. Since they are also assumed solvable, for $i > \ell$ the equality $\deg_{X_i} T_i = \deg_{X_i} T'_i$ holds, showing that S is monic in X_i, as requested.

Assume that $\mathfrak{T}$ consists of $s > 2$ triangular sets $T^1, \ldots, T^s$. First, we apply the case $s = 2$ to T^1, T^2, obtaining a triangular set $T^{1,2}$. Observe that every pair $T^{1,2}, T^j$, for $3 \leq j \leq s$, is solvable but not certified solvable: we obtain the requested Bézout coefficient by updating the known ones. Let us fix $3 \leq j \leq s$. Given $A_1, A_2, B_1, B_j, C_2, C_j \in K_\ell[X_\ell]$ such that $A_1 T^1_\ell + A_2 T^2_\ell = B_1 T^1_\ell + B_j T^j_\ell = C_2 T^2_\ell + C_j T^j_\ell = 1$ hold in $K_\ell[X_\ell]$, we let $\alpha = B_1 C_2 \mod T^j_\ell$ and $\beta = A_1 C_j T^1_\ell + A_2 B_j T^2_\ell \mod T^1_\ell T^2_\ell$. Then, $\alpha T^{1,2}_\ell + \beta T^j_\ell = 1$ in $K_\ell[X_\ell]$, as requested. Proceeding by induction ends the proof. □

Splitting critical pairs. Let now V be a 0-dimensional variety over k. Proposition 4.3 below encapsulates the first part of the *Split-and-Merge* algorithm: given any triangular

decomposition $\mathfrak{T}$ of V, it outputs another one, without critical pairs. We first describe the basic splitting step.

Proposition 4.2. *Let $\mathfrak{T}$ be a triangular decomposition of V which contains critical pairs. Then one can compute a triangular decomposition $\mathsf{Split}(\mathfrak{T})$ of V which has cardinality larger than that of $\mathfrak{T}$.*

Proof: Let T, T' be a critical pair of $\mathfrak{T}$ of level ℓ and let G be a GCD of T_ℓ, T'_ℓ in $K_\ell[X_\ell]$. First, assume that G is monic, in the sense of [89]; let Q and Q' be the quotients of T_ℓ and T'_ℓ by G in $K_\ell[X_\ell]$. We define the sets

$$\begin{aligned} A &= T_1, \ldots, T_{\ell-1}, G, T_{\ell+1}, \ldots, T_n, \\ B &= T_1, \ldots, T_{\ell-1}, Q, T_{\ell+1}, \ldots, T_n, \\ A' &= T_1, \ldots, T_{\ell-1}, G, T'_{\ell+1}, \ldots, T'_n, \\ B' &= T_1, \ldots, T_{\ell-1}, Q', T'_{\ell+1}, \ldots, T'_n. \end{aligned}$$

We let $\mathsf{Split}(\mathfrak{T}) = \{A, B, A', B'\}$, excluding the triangular sets defining the empty set. Since the pair T, T' is critical, $V(A)$ and $V(A')$ are non-empty. Since T_ℓ and T'_ℓ are not associate in $K_\ell[X_\ell]$, at least Q or Q' is not constant. Thus, $\mathsf{Split}(\mathfrak{T})$ has cardinality at least 3. Since $\langle T \rangle$ and $\langle T' \rangle$ are radical, if $Q \notin K_\ell$, G and Q are coprime in $K_\ell[X_\ell]$, so $V(T)$ is the disjoint union of $V(A)$ and $V(B)$. The same property holds for A' and B'. Thus, the proposition is proved.

Assume now that T_ℓ, T'_ℓ have no monic GCD in $K_\ell[X_\ell]$. Then, there exist triangular sets $C^1, \ldots, C^s, D^1 \ldots D^s$ such that $V(T)$ is the disjoint union of $V(C^1)$, …, $V(C^s)$, $V(T')$ is the disjoint union of $V(D^1)$, …, $V(D^s)$, at least one pair C^i, D^j is critical and C^i_ℓ, D^j_ℓ admits a monic GCD in $K_\ell[X_\ell]$. These triangular sets are obtained by the algorithms of [89] when computing a GCD of T_ℓ, T'_ℓ in $K_\ell[X_\ell]$. Then the results of the monic case prove the existence of $\mathsf{Split}(\mathfrak{T})$. □

Proposition 4.3. *Let $\mathfrak{T}$ be a triangular decomposition of V. One can compute a triangular decomposition $\mathfrak{T}'$ of V with no critical pairs, and where each pair of triangular sets is certified.*

Proof: Write $\mathfrak{T}_0 = \mathfrak{T}$, and define a sequence $\mathfrak{T}_i$ by $\mathfrak{T}_{i+1} = \mathsf{Split}(\mathfrak{T}_i)$, if $\mathfrak{T}_i$ contains critical pairs, and $\mathfrak{T}_{i+1} = \mathfrak{T}_i$ otherwise. Testing the presence of critical pairs is done by GCD computations, which yields the Bézout coefficients in case of coprimality. Let D be the number of irreducible components of V. Any family $\mathfrak{T}_i$ has cardinality at most D, so the sequence $\mathfrak{T}_i$ becomes stationary after at most D steps. □

Thus, we can now suppose that we have a triangular decomposition $\mathfrak{T}$ of V without critical pairs, and where every pair is certified, such as the one computed in Proposition 4.3. We describe the second part of the *Split-and-Merge* algorithm: merging solvable families in a suitable order, to obtain the equiprojectable decomposition of V.

For $0 \leq \kappa \leq n$, we say that $\mathfrak{T}$ satisfies property P_κ if:

$$\forall T, T' \in \mathfrak{T}\,, \quad \{T, T'\} \text{ is certified}\,, \quad \mathrm{level}(T, T') \leq \kappa\,, \quad \forall \kappa < i \leq n\,, \deg_{X_i} T_i = \deg_{X_i} T'_i.$$

Observe that if $\mathsf{P}_0(\mathfrak{T})$ holds, then $\mathfrak{T}$ contains only one triangular set, and that the input family $\mathfrak{T}$ satisfies P_n.

The basic merging algorithm. Let $1 \leq \kappa \leq n$. We now define the procedure Merge_κ, which takes as input a family $\mathfrak{T}_\kappa$ of triangular sets which satisfies P_κ, and outputs several families of triangular sets, whose reunion defines the same set of points, and all of which satisfy $\mathsf{P}_{\kappa-1}$. First, we partition $\mathfrak{T}_\kappa$ using the equivalence relation $T \equiv T'$ if and only if $T_1, \ldots, T_{\kappa-1} = T'_1, \ldots, T'_{\kappa-1}$. Assumption P_κ shows that each equivalence class is certified and solvable of level κ. We then let $\mathfrak{S}^{(\kappa)}$ be the family of triangular sets obtained by applying Proposition 4.1 to each equivalence class.

Lemma 4.1. *Let $S \neq S'$ in $\mathfrak{S}^{(\kappa)}$. The pair $\{S, S'\}$ is non-critical, certified, of level $\ell < \kappa$.*

Proof: Let $T, T' \in \mathfrak{T}$, which respectively divide S and S'. Due to assumption P_κ, there exists $0 \leq \ell \leq \kappa$ such that $T_1, \ldots, T_{\ell-1} = T'_1, \ldots, T'_{\ell-1}$ and $(T_1, \ldots, T_\ell)$ and $(T'_1, \ldots, T'_\ell)$ have no common zero. Then, $\ell < \kappa$, since $T \not\equiv T'$. Thus, $T_1, \ldots, T_\ell = S_1, \ldots, S_\ell$ and $T'_1, \ldots, T'_\ell = S'_1, \ldots, S'_\ell$. Since $\{T, T'\}$ is certified of level $\ell < \kappa$, $\{S, S'\}$ is also. □

We partition $\mathfrak{S}^{(\kappa)}$ some more, into the classes of the equivalence relation $S \equiv' S'$ if and only if $\deg_{X_\kappa} S_\kappa = \deg_{X_\kappa} S'_\kappa$. Let $\mathfrak{S}_1^{(\kappa)}, \ldots, \mathfrak{S}_\delta^{(\kappa)}$ be the equivalence classes, indexed by the common degree in X_κ; we define $\mathsf{Merge}_\kappa(\mathfrak{T}_\kappa)$ as the data of all these equivalence classes.

Lemma 4.2. *Each family $\mathfrak{S}_d^{(\kappa)}$ satisfies $\mathsf{P}_{\kappa-1}$.*

Proof: Let $S \neq S'$ in $\mathfrak{S}_d^{(\kappa)}$, and let T, T' be as in the proof of Lemma 4.1; we now prove the degree estimate. For $\kappa < i \leq n$, we have $\deg_{X_i} T_i = \deg_{X_i} S_i$ and $\deg_{X_i} T'_i = \deg_{X_i} S'_i$; assumption P_κ shows that $\deg_{X_i} S_i = \deg_{X_i} S'_i$ for $\kappa < i \leq n$. Since $\deg_{X_\kappa} S_\kappa = \deg_{X_\kappa} S'_\kappa = d$, the lemma is proved. □

Proposition 4.4. *$V(\mathfrak{S}_d^{(\kappa)}) = B(\kappa, d, V(\mathfrak{T}_\kappa))$ for all d.*

Proof: We know that $V(\mathfrak{T}_\kappa)$ is the union of the $V(\mathfrak{S}_d^{(\kappa)})$. Besides, both families $\{V(\mathfrak{S}_d^{(\kappa)})\}$ and $\{B(\kappa, d, V(\mathfrak{T}))\}$ form a partition of $V(\mathfrak{T}_\kappa)$. Thus, it suffices to prove that for x in $V(\mathfrak{T}_\kappa)$, $x \in V(\mathfrak{S}_d^{(\kappa)})$ implies that $\pi_\kappa^n(x) \in A(d, W)$, with $W = \pi_\kappa^n(V(\mathfrak{T}_\kappa))$. First, for S in $\mathfrak{S}^{(\kappa)}$, write $W_S = \pi_\kappa^n(S)$. Then Lemma 4.1 shows that the W_S form a partition of W, and that their images $\pi_{\kappa-1}^\kappa(W_S)$ are pairwise disjoint.

Let now $x \in V(\mathfrak{S}_d^{(\kappa)})$ and $y = \pi_\kappa^n(x)$. There exists a unique $S \in \mathfrak{S}^{(\kappa)}$ such that $x \in V(S)$. The definition of $\mathfrak{S}_d^{(\kappa)}$ shows that there are exactly d points y' in W_S such that $\pi_{\kappa-1}^\kappa(y) = \pi_{\kappa-1}^\kappa(y')$. On the other hand, for any $y \in W_{S'}$, with $S' \neq S$, the above remark shows that $\pi_{\kappa-1}^\kappa(y) \neq \pi_{\kappa-1}^\kappa(y')$. Thus, there are exactly d points y' in W such that $\pi_{\kappa-1}^\kappa(y) = \pi_{\kappa-1}^\kappa(y')$; this concludes the proof. □

The main merging algorithm. We can now give the main algorithm. We start from a triangular decomposition $\mathfrak{T}$ of V without critical pairs, and where every pair is certified, so it satisfies P_n. Let us initially define $\mathfrak{T}_n = \{\mathfrak{T}\}$; note that $\mathfrak{T}_n$ is a *set of families of triangular sets*. Then, for $1 \leq \kappa \leq n$, assuming $\mathfrak{T}_\kappa$ is defined, we write $\mathfrak{T}_{\kappa-1} = \cup_{\mathfrak{U}^{(\kappa)} \in \mathfrak{T}_\kappa} \mathsf{Merge}_\kappa(\mathfrak{U}^{(\kappa)})$. Lemma 4.2 shows that this process is well-defined; note that each $\mathfrak{T}_\kappa$ is a set of families of triangular sets as well.

Let $\mathfrak{U}$ be a family of triangular sets in $\mathfrak{T}_0$. Then $\mathfrak{U}$ satisfies P_0, so by the remarks make previously, $\mathfrak{U}$ consists in a single triangular set. Proposition 4.4 then shows that the triangular sets in $\mathfrak{T}_0$ form the equiprojectable components of V.

$$C\left|\begin{array}{l} C_2 = X_2^2 + 6X_2X_1^2 + 2X_2 + X_1 \\ C_1 = X_1^3 + 6X_1^2 + 5X_1 + 2 \end{array}\right. , \quad D\left|\begin{array}{l} D_2 = X_2 + 6 \\ D_1 = X_1 + 6 \end{array}\right.$$

$$\downarrow \text{ Split C : GCD } \downarrow$$

$$E\left|\begin{array}{l} E_2 = X_2^2 + X_1 \\ E_1 = X_1^2 + 5 \end{array}\right. , \quad F\left|\begin{array}{l} F_2 = X_2^2 + X_2 + 1 \\ F_1 = X_1 + 6 \end{array}\right. , \quad D\left|\begin{array}{l} D_2 = X_2 + 6 \\ D_1 = X_1 + 6 \end{array}\right.$$

$$\downarrow \text{ Merge F and D : CRT } \downarrow$$

$$E\left|\begin{array}{l} E_2 = X_2^2 + X_1 \\ E_1 = X_1^2 + 5 \end{array}\right. , \quad G\left|\begin{array}{l} G_2 = X_2^3 + 6 \\ G_1 = X_1 + 6 \end{array}\right.$$

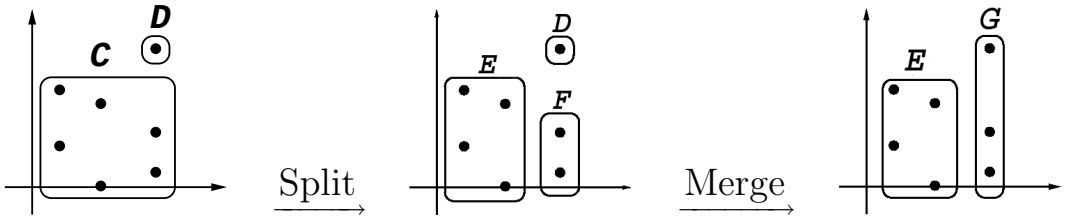

Figure 4.4: The Split and Merge algorithm on the example

4.3 proof of Theorem 4.1

In this section, we consider the simple solutions $\mathcal{Z}(\mathbf{F})$ of a system $\mathbf{F} = F_1, \ldots, F_n$ in $\mathbb{Z}[X_1, \ldots, X_n]$, that is, those where the Jacobian determinant J of $\mathbf{F}$ does not vanish. We prove that for all primes p but a finite number, the equiprojectable decomposition of $\mathcal{Z}(\mathbf{F})$ reduces modulo p to that of $\mathcal{Z}(\mathbf{F} \bmod p)$. These results require to control the cardinality of the "specialization" of a variety at p. Such questions are easy to formulate using *primitive elements* (Ch. 1 § 1.1.2, p. 14).

Geometric considerations. Let now $\mathcal{Z} = \mathcal{Z}(\mathbf{F})$. For $1 \leq i \leq n$, let Δ_i be a linear form in $\mathbb{Z}[X_1, \ldots, X_i]$ which is a primitive element for $\pi_i^n(\mathcal{Z})$, let $\mu_i \in \mathbb{Q}[T]$ be its minimal polynomial, and let $W_1, \ldots, W_n \in \mathbb{Q}[T]$ be the parametrization of $\mathcal{Z}$ associated to Δ_n. Let finally p a prime. We first introduce assumptions on p (denoted by $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$), that yield the conclusion of Theorem 4.1 in a series of lemmas; we then give quantitative estimates for these assumptions.

$\mathbf{H_1}$. The prime p divides no coefficients in $\mu_n, W_1, \ldots, W_n$ and μ_n remains squarefree modulo p.

Let $\mathbb{F}_q$ be a finite extension of $\mathbb{F}_p$ such that $(\mu_n \bmod p)$ splits in $\mathbb{F}_q$, let $\mathbb{Q}_q$ be the corresponding unramified extension of $\mathbb{Q}_p$ [86] and $\mathbb{Z}_q$ its ring of integers; then, μ_n splits in $\mathbb{Q}_q$, and has all its roots in $\mathbb{Z}_q$; thus, $\mathcal{Z}$ lies in $\mathbb{Z}_q^n$. Note that p divides no coefficient in $\mu_1, \ldots, \mu_n$: the roots of μ_i are the values of Δ_i on $\pi_i^n(\mathcal{Z})$, so they are in $\mathbb{Z}_q$, hence the coefficients of μ_i are in $\mathbb{Z}_q \cap \mathbb{Q} = \mathbb{Z}_p$. The map $\mathbb{Z}_q \to \mathbb{F}_q$ of reduction modulo p extends to maps $a \in \mathbb{Z}_q^i \mapsto \overline{a} \in \mathbb{F}_q^i$ for all i. Given $A \subset \mathbb{Z}_q^i$, $\overline{A}$ is the set $\{\overline{a} \mid a \in A\}$. The same notation is used for the reduction of polynomials modulo p.

$\mathbf{H_2}$. All polynomials $\overline{\mu_i}$ are squarefree.

Lemma 4.3. *For $i \leq n$, $\#\pi_i^n(\mathcal{Z})$ equals $\#\pi_i^n(\overline{\mathcal{Z}})$.*

PROOF: The inequality $\#\pi_i^n(\overline{\mathcal{Z}}) \leq \#\pi_i^n(\mathcal{Z})$ is obvious. By assumption $\mathbf{H_2}$, all values taken by Δ_i on $\pi_i^n(\overline{\mathcal{Z}})$ are distinct, so $\#\pi_i^n(\overline{\mathcal{Z}}) \geq \deg \mu_i = \#\pi_i^n(\mathcal{Z})$. □

Lemma 4.4. *For all $d_2, \ldots, d_n$, $B(d_2, \ldots, d_n, \overline{\mathcal{Z}})$ equals $\overline{B(d_2, \ldots, d_n, Z)}$.*

PROOF: We prove on $\ell = n+1, \ldots, 2$ that for all $d_\ell, \ldots, d_n$, $B(d_\ell, \ldots, d_n, \overline{\mathcal{Z}})$ equals $\overline{B(d_\ell, \ldots, d_n, \mathcal{Z})}$; taking $\ell = 2$ gives the lemma. Since $B(X) = X$ for any variety X, this property holds for $\ell = n+1$. Assuming it for $B(d_{\ell+1}, \ldots, d_n, \mathcal{Z})$, we prove it for $B(d_\ell, \ldots, d_n, \mathcal{Z})$. Let $B = B(d_{\ell+1}, \ldots, d_n, \mathcal{Z})$, $B_\ell = \pi_\ell^n(B)$ and $B_{\ell-1} = \pi_{\ell-1}^n(B)$; Lemma 4.3 implies that reduction modulo p is one-to-one on both B_ℓ and $B_{\ell-1}$. For y in $B_{\ell-1}$ and z in $\overline{B_{\ell-1}}$, we define

$$\mu(y) = (\pi_{\ell-1}^\ell)^{-1}(y) \cap B_\ell \quad \text{and} \quad \mu(z) = (\pi_{\ell-1}^\ell)^{-1}(z) \cap \overline{B_\ell}.$$

We first prove that $\mu(y)$ and $\mu(\overline{y})$ have the same cardinality for all y in $B_{\ell-1}$. To this effect, observe the equalities

$$\sum_{y \in B_{\ell-1}} \#\mu(y) = \#B_\ell, \qquad \sum_{z \in \overline{B_{\ell-1}}} \#\mu(z) = \#\overline{B_\ell}.$$

Let now y in $B_{\ell-1}$. Since $\overline{\mu(y)} \subset \mu(\overline{y})$, injectivity of the reduction mod p on B_ℓ implies that $\#\mu(y) \leq \#\mu(\overline{y})$. Thus,

$$\#B_\ell = \sum_{y \in B_{\ell-1}} \#\mu(y) \leq \sum_{y \in B_{\ell-1}} \#\mu(\overline{y}).$$

Injectivity of the reduction mod p on $B_{\ell-1}$ implies that

$$\sum_{y \in B_{\ell-1}} \#\mu(\overline{y}) = \sum_{z \in \overline{B_{\ell-1}}} \#\mu(z) = \#\overline{B_\ell}.$$

This sum equals $\#B_\ell$. Thus, all inequalities are equalities, giving our claim.

For x in B_ℓ, write $\nu(x) = \mu(\pi_{\ell-1}^\ell(x))$; define similarly $\nu(z)$ for z in $\overline{B_\ell}$. By the previous point, $\nu(x)$ and $\nu(\overline{x})$ have the same cardinality. Recalling from Section 4.2 that for $d \in \mathbb{N}$, we have defined $A(d, B_\ell)$ as the set $\{x \in B_\ell \mid \#\nu(x) = d\}$, and $A(d, \overline{B_\ell})$ as the set $\{z \in \overline{B_\ell} \mid \#\nu(z) = d\}$, one can see $\overline{A(d, B_\ell)} = A(d, \overline{B_\ell})$. To conclude, recall that by definition $\{x \in \overline{\mathcal{Z}} \mid \pi_\ell^n(x) \in A(d, \pi_\ell^n(B(d_{\ell+1}, \ldots, d_n, \overline{\mathcal{Z}})))\} = B(d, d_{\ell+1}, \ldots, d_n, \overline{\mathcal{Z}})$. By the induction assumption, this equals $\{x \in \overline{\mathcal{Z}} \mid \pi_\ell^n(x) \in A(d, \overline{B_\ell})\}$, and we have proved that this equals $\overline{\{x \in \mathcal{Z} \mid \pi_\ell^n(x) \in A(d, B_\ell)\}}$. By definition, this is $\overline{B(d, d_\ell, \ldots, d_n, \mathcal{Z})}$, which is what we wanted. □

Lemma 4.5. *Let $T^1, \ldots, T^s$ be the triangular sets that describe the equiprojectable decomposition of $\mathcal{Z}$. Then p cancels no denominator in the coefficients of $T^1, \ldots, T^s$, and the reduction of these triangular sets modulo p defines the equiprojectable decomposition of $\overline{\mathcal{Z}}$.*

PROOF: For $i \leq s$, let $\mathcal{Z}_i = \mathcal{Z}(T^i)$. By Lemma 4.4, $\overline{\mathcal{Z}}_1, \ldots, \overline{\mathcal{Z}}_s$ are the equiprojectable components of $\overline{\mathcal{Z}}$. For $i \leq s$, $\overline{\mathcal{Z}}_i$ is described by a triangular set t^i with coefficients in $\mathbb{F}_p$. The coefficients of T^i are rational functions of the points in $\mathcal{Z}_i$, given by interpolation formulas [32, §3]. With these formulas, Lemma 4.3 shows that all denominators are non-zero modulo p. The coefficients of t^i are obtained using the same formulas, using the coordinates of the points in $\overline{\mathcal{Z}}_i$. Thus, $t^i = T^i \bmod p$. □

$\mathbf{H_3}$. The Jacobian determinant of $\mathbf{F}$ vanishes nowhere on $\overline{\mathcal{Z}}$.

Lemma 4.6. *The set $\overline{\mathcal{Z}}$ equals $\mathcal{Z}(\overline{\mathbf{F}})$.*

PROOF: First, we prove that $\overline{\mathbf{F}}$ vanishes on $\overline{\mathcal{Z}}$. Indeed, all F_i belong to the ideal generated by $I = (\mu_n, X_1 - W_1, \ldots, X_n - W_n)$ in $\mathbb{Q}[T, X_1, \ldots, X_n]$. Now, I is a Gröbner basis, so any F_i can be written in terms of I. Since p divides no denominator and no leading term in I, the division equality specializes modulo p, and $\overline{\mathbf{F}}$ vanishes on $\overline{\mathcal{Z}}$, as requested. Let then $\mathcal{Z}' = \mathcal{Z}(\overline{\mathbf{F}})$. By Assumption $\mathbf{H_3}$, $\overline{\mathcal{Z}} \subset \mathcal{Z}'$, so it suffices to prove that $\#\mathcal{Z}' \leq \#\overline{\mathcal{Z}}$. Let $\mathbb{F}_r$ be a finite extension of $\mathbb{F}_p$ that contains the coordinates of all these points and let $\mathbb{Q}_r$ be the corresponding unramified extension of $\mathbb{Q}_p$. By Hensel's lemma, all points in $\mathcal{Z}'$ lift to pairwise distinct simple roots of $\mathbf{F}$ in $\mathbb{Q}_r^n$. Thus, $\#\mathcal{Z}' \leq \#\mathcal{Z} = \#\overline{\mathcal{Z}}$. □

Quantitative estimates. By Lemmas 4.5 and 4.6, assumptions $\mathbf{H_1}$, $\mathbf{H_2}$ and $\mathbf{H_3}$ imply Theorem 4.1. Thus, it suffices to give quantitative estimates for these assumptions. Let us introduice the quantities D, H, h_Δ and H_Δ verifying:

$$\begin{aligned} D &\geq \deg \mathcal{Z} \geq \deg \pi_i^n(\mathcal{Z}),\ i = 1, \ldots, n \\ H &\geq h(\mathcal{Z}) \geq h(\pi_1^n(\mathcal{Z})),\ i = 1, \ldots, n \\ h_\Delta &\geq \max\{h(\Delta_1), \ldots, h(\Delta_n)\} \\ H_\Delta &= H + Dh_\Delta + D\log(n+2) + (n+1)\log D \end{aligned}$$

From height bounds of Chapter 2 Th. 2.2, H_Δ bounds the height of any polynomials of the Kronecker representations of $\pi_1^n(\mathcal{Z}), \ldots, \pi_{n-1}^n(\mathcal{Z}), \mathcal{Z}$ attached to $\Delta_1, \ldots, \Delta_n$.

Lemma 4.7. *There exists a in $\mathbb{N} - \{0\}$ such that if p does not divide a, $\mathbf{H_1}$ and $\mathbf{H_2}$ hold. Moreover a verifies:*

$$h(a) \leq n\left((2D-1)H_\Delta + D\log(2D-1)\right).$$

PROOF: Fix i in $1, \ldots, n$, and let $\chi, \chi', v_1, \ldots, v_i$ the polynomials of the Kronecker representation of $\pi_i^n(\mathcal{Z})$ associated to the separating linear form Δ_i; By Theorem 2.2 all of them have integer coefficients of height at most H_Δ. Let now a_i be the resultant of χ and χ'; by Hadamard's bound, $h(a_i) \leq (2D-1)H_\Delta + D\log(2D-1)$.

Suppose that p does not divide a_i. Then, χ keeps the same degree and remains square-free modulo p. Furthermore, p divides no coefficient in any W_j, since all denominators in $1/\chi' \bmod \chi$ divide a_i. Thus, assumption $\mathbf{H_1}$ holds. Repeating this argument for all projections $\pi_i^n(\mathcal{Z})$, and taking $a = a_1 \cdots a_n$ gives assumption $\mathbf{H_2}$. The height bound $h(a)$ follows easily. □

Lemma 4.8. *There exists a' in $\mathbb{N} - \{0\}$ such that if p does not divide aa', $\mathbf{H_1}$, $\mathbf{H_2}$ and $\mathbf{H_3}$ hold, and with $h(a') \leq \theta(n, d, h, D, H_\Delta)$, with $\theta \in O\big(nD(dH_\Delta + dD + h)\big)$.*

PROOF: Let $\chi, v_1, \ldots, v_n$ be the Kronecker representation of $\mathcal{Z}$ associated to the separating linear form Δ_n, let J^h be the homogenization of J with respect to a new variable. Define $a' \in \mathbb{Z}$ by:

$$a' := \mathrm{Res}(\chi, J^h(\chi', v_1, \ldots, v_n)).$$

Since $\mathcal{Z}$ is the set of simple roots of $V(\mathbf{F})$, it follows that $a' \neq 0$. From Corollary 1.3, the Jacobian determinant J has coefficients of height at most $n(h + \log(d) + d\log(n+1))$ (in fact, the polynomial entries have height at most $\log(d) + h$ and degree at most $d-1$). Its

specialization at $\chi', v_1, \ldots, v_n$ has degree at most $n(d-1)D$, and from Inequality $\mathbf{A_9}$ has height at most:

$$\begin{aligned} h(J^h(\chi', v_1, \ldots, v_n)) &\leq h(J^h) + \deg(J^h)(H_\Delta + \log(n+2) + D\log(2)) \\ &\leq n\,(h + \log(d) + dH_\Delta + 2d\log(n+2) + (d-1)D) \end{aligned}$$

The Hadamard's bound adapted to the case of Sylvester matrix gives:

$$h(a') \leq n(d-1)DH_\Delta + Dh(J^h(\chi', v_1, \ldots, v_n)) + \frac{1}{2}\left(D\log(n(d-1)D) + n(d-1)D\log(D)\right)$$

By replacing by the above bound for the $h(J^h(\chi', v_1, \ldots, v_n))$:

$$\begin{aligned} h(a') &\leq H_\Delta\left(n(d-1)D + ndD\right) + nhD + n(d-1)D^2 + 2ndD\log(n+2) \\ &+ D\log(d) + \frac{1}{2}\log(D)\left(n(d-1) + D\right) + \frac{1}{2}D\log(n) + \frac{1}{2}D\log(d-1) \end{aligned} \quad (4.1)$$

This proves the height estimates of $\theta(n, d, h, D, H_\Delta)$.

Suppose now that p does not divide aa'. Then the degree of χ does not drop modulo p, and thus no root of $\overline{\chi}$ cancels $\overline{J^h(\chi', v_1, \ldots, v_n)}$. In other words, all points described by $\overline{\chi}(T) = 0$ and $\overline{\chi'}(T)X_i = \overline{v_i}(T)$, $1 \leq i \leq n$, are simple for $\overline{\mathbf{F}}$. This set of points equals Z, giving $\mathbf{H_3}$. □

In view of Lemma 4.8, we prove Theorem 4.1 with $A = aa'$. We turn now to extrinsic quantitative estimates for a', and begin with H_Δ, using:

$$\begin{aligned} H &\leq nd^n(h + 2\log(n+1)), \quad \text{arithmetic Bézout theorem (Thm. 1.5)} \\ h_\Delta &\leq n(\log(n) + 2n\log(d)) \quad \text{by [101, Lemma 2.1]} \end{aligned}$$

All linear forms Δ_i can be bounded by h_Δ.

$$\begin{aligned} H_\Delta &\leq H + Dh_\Delta + D\log(n+2) + (n+1)\log(D) \\ &\leq d^n\left(nh + 2n\log(n+1)\right) + nd^n\left(\log(n) + 2n\log(d)\right) + d^n\log(n+2) \\ &+ (n+1)h?\log(d) \\ &\leq nd^n\left(h + \log(n+2)(3 + \frac{1}{n}) + \log(d)(2n + \frac{n+1}{d^n})\right) \end{aligned}$$

In Equality (4.1), this gives:

$$\begin{aligned} h(a') &\leq nd^n\left(h + \log(n+2)(3 + \frac{1}{n}) + \log(d)(2n + \frac{n+1}{d^n})\right) + nhd^n + nd^{2n+1} \\ &+ 2nd^{n+1}\log(n+2) + d^n\log(d) + \frac{1}{2}\Big(n\log(d)(nd + d^n) + d^n\log(n) + d^n\log(d)\Big) \\ &\leq n^2d^{2n+1}\left(h\Big(1 + \frac{1}{nd^{n+1}}\Big) + \log(n+2)\Big(3 + \frac{1}{n} + \frac{2}{nd^n} + \frac{1}{2n^2d^{n+1}}\Big)\right. \\ &+ \left.\log(d)\Big(2n + \frac{n+1}{d^n} + \frac{1}{2d^{2n}} + \frac{1}{2nd^{n+1}} + \frac{1}{2n^2d^{n+1}}\Big) + \frac{1}{n}\right) \\ &\leq n^2d^{2n+1}\left(2h + 3\log(n+2) + 2n\log(d) + g(n,d)\right), \end{aligned}$$

with

$$g(n,d) = \log(n+2)\,\frac{2nd^{n+1}+4nd+1}{2n^2d^{n+1}} + \log(d)\,\frac{2n^2(n+1)d^n+n^2+nd^{n-1}+d^{n-1}}{2n^2d^{2n}} + \frac{1}{n}.$$

It is bounded by 5, for $(n,d) \in \mathbb{N}^* \times \mathbb{N}^*$. As for $h(a)$, we get:

$$\begin{aligned}
h(a) &\leq n\left((2D-1)H_\Delta + D\log(2D-1)\right)\\
&\leq n\left((2d^n-1)nd^n\Big(h+\log(n+2)(3+\frac{1}{n})+\log(d)(2n+\frac{n+1}{d^n})\Big)\right.\\
&+ d^n(\log(2)+n\log(d))\\
&\leq 2n^2d^{2n}\left(h+\log(n+2)(3+\frac{1}{n})+\log(d)(2n+\frac{n+1}{d^n})+\frac{1}{2n^2d^n}(\log(2)+n\log(d)\right)\\
&\leq 2n^2d^{2n}\left(h+3\log(n+2)+2n\log(d)+f(n,d)\right),
\end{aligned}$$

where

$$f(n,d) = \frac{1}{n}\log(n+2) + \log(d)\Big(\frac{n+1}{d^n}+\frac{1}{2nd^n}\Big) + \frac{1}{2n^2d^n}\log(2).$$

This function is bounded by 3 over $\mathbb{N}^* \times \mathbb{N}^*$. Finally:

$$\begin{aligned}
h(a) &\leq 2n^2d^{2n}(h+3\log(n+2)+2n\log(d)+3)\\
h(a') &\leq n^2d^{2n+1}(2h+3\log(n+2)+2n\log(d)+5)
\end{aligned}$$

4.4 Proof of Theorem 4.2

We now give the details of our lifting algorithm: given a polynomial system $\mathbf{F}$, it outputs a triangular representation of its set of simple solutions $\mathcal{Z} = \mathcal{Z}(\mathbf{F})$, by means of the polynomials $N^1, \ldots, N^s$ defined in the introduction. First of all, we describe the required subroutines, freely using the notation of Theorem 4.2, and that preceding it. We do not give details of the complexity estimates for lack of space; they are similar to those of [105].

- `EquiprojDecomposition` takes as input a polynomial system $\mathbf{F}$ and outputs the equiprojectable decomposition of $\mathcal{Z}(\mathbf{F})$, encoded by triangular sets. This routine is called here for systems defined over finite fields. For the experiments in the next section, we applied the triangularization algorithm of [88], followed by the Split-and-Merge algorithm of Section 4.2, modulo a prime. Studying the complexity of this task is certainly easily deducible form results of Chapter 5, but is still a work in progress; hence, we consider this subroutine as an oracle here, which is called O_2 in Theorem 4.2.

- `Lift` applies the Hensel lifting algorithm of [105], but this time to a *family of triangular sets*, defined first modulo a prime p_1, to triangular sets defined modulo the successive powers $p_1^{2^\kappa}$. From [105], one easily sees that the κth lifting step has a bit complexity quasi-linear in $(L, h_L, \mathsf{C}^n, \sum_{i\leq s}\deg V(T^i), 2^\kappa, \log p_1)$, *i.e.* in $(L, h_L, \mathsf{C}^n, \deg\mathcal{Z}, 2^\kappa, \log p_1)$.

- `Convert` computes the polynomials N^i starting from the polynomials T^i. Only multiplications modulo triangular sets are needed to perform this operation, so its complexity is negligible before that of `Lift`.

- `RationalReconstruction` does the following. Let $a = p/q \in \mathbb{Q}$, and $m \in \mathbb{N}$ with $\gcd(q, m) = 1$. If $h(m) \geq 2h(a) + 1$, given $a \bmod m$, this routine outputs a. If $h(m) < 2h(a) + 1$, the output may be undefined, or differ from a. We extend this notation to the reconstruction of all coefficients of a family of triangular sets. Using the fast Euclidean algorithm [117, Ch 5,11], its complexity is negligible before that of `Lift`.

- We do not consider the cost of prime number generation. We see them as input here; formally, in Theorem 4.2, this is handled by calls to oracle O_1.

`modularTriangularize`

\# `Inputs:` The system $\mathbf{F}$, primes p_1, p_2
\# `Output:` The polynomials $N^1, \ldots, N^s$.

1. $T^{1,0}, \ldots, T^{s,0} \leftarrow$ `EquiprojDecomposition`$(\mathcal{Z}(\mathbf{F} \bmod p_1))$
2. $u^1, \ldots, u^{s'} \leftarrow$ `EquiprojDecomposition`$(\mathcal{Z}(\mathbf{F} \bmod p_2))$
3. $m^1, \ldots, m^{s'} \leftarrow$ `Convert`$(u^1, \ldots, u^{s'})$
4. $\kappa \leftarrow 1$
5. `while not(Stop) do`

   $T^{1,\kappa}, \ldots, T^{s,\kappa} \leftarrow$ `Lift`$(T^{1,\kappa-1}, \ldots, T^{s,\kappa-1}) \bmod p_1^{2^\kappa}$
   $N^{1,\kappa}, \ldots, N^{s,\kappa} \leftarrow$ `Convert`$(T^{1,\kappa}, \ldots, T^{s,\kappa})$
   $N_{\mathbb{Q}}^{1,\kappa}, \ldots, N_{\mathbb{Q}}^{s,\kappa} \leftarrow$ `RationalReconstruction`$(N^{1,\kappa}, \ldots, N^{s,\kappa})$
   `Stop` $\leftarrow \{m^1, \ldots, m^{s'}\}$ `Equals` $\{N_{\mathbb{Q}}^{1,\kappa}, \ldots, N_{\mathbb{Q}}^{s,\kappa}\} \bmod p_2$
   $\kappa \leftarrow \kappa + 1$

6. `end while`
7. `r eturn` $N_{\mathbb{Q}}^{1,\kappa-1}, \ldots, N_{\mathbb{Q}}^{s,\kappa-1}$

Algo **4.1:** Computing a triangular decomposition by lifting technique

We still use the notation and assumption of Theorem 4.2. From 2.7, all coefficients of $N^1, \ldots, N^s$ have height in $5 \deg \mathcal{Z} \log(n+3) + h(\mathcal{Z}))$, which can explicitly be bounded by $\mathfrak{h}_{\mathbf{F}}$. For $p_1 \leq \exp(2\mathfrak{h}_{\mathbf{F}} + 1)$, define

$$\mathfrak{d} = \mathfrak{d}(p_1) = \left\lceil \log_2 \left(\frac{2\mathfrak{h}_{\mathbf{F}} + 1}{\log p_1} \right) \right\rceil.$$

Then, $p_1^{2^{\mathfrak{d}(p_1)}}$ has height at least $2\mathfrak{h}_{\mathbf{F}} + 1$. In view of the prerequisites for rational reconstruction, $\mathfrak{d}(p_1)$ bounds the number of lifting steps. From an intrinsic viewpoint, at the last lifting step, 2^κ is in $O(\log(n) \deg \mathcal{Z} + h(\mathcal{Z}))$.

Suppose that p_1 does not divide the integer A of Theorem 4.1. Then, Hensel lifting computes approximations $T^{1,\kappa}, \ldots, T^{s,\kappa} = T^1, \ldots, T^s$ modulo $p_1^{2^\kappa}$. At the κth lifting step, let $N^{1,\kappa}, \ldots, N^{s,\kappa}$ be the output of `Convert` applied to $T^{1,\kappa}, \ldots, T^{s,\kappa}$, computed modulo $p_1^{2^\kappa}$; let $N_{\mathbb{Q}}^{1,\kappa}, \ldots, N_{\mathbb{Q}}^{s,\kappa}$ be the same polynomials after rational number reconstruction, if possible. By construction, they have rational coefficients of height at most $2^{\kappa-1} \log p_1$. Supposing that p_2 does not divide the integer A of Theorem 4.1, failure occurs only if for some κ in $0, \ldots, \mathfrak{d} - 1$, and some j in $1, \ldots, s$, $N_{\mathbb{Q}}^{j,\kappa}$ and N^j differ, but coincide modulo p_2. It happens

when the two coefficients $c_{\mathbb{Q}}^{j,\kappa}$ and c^j of at least one monomial of one of the polynomials in $N_{\mathbb{Q}}^{j,\kappa}$ and N^j respectively verifies:

$$c_{\mathbb{Q}}^{j,\kappa} \neq c^j \quad \text{but} \quad p_2 | c_{\mathbb{Q}}^{j,\kappa} - c^j.$$

Writing $c_{\mathbb{Q}}^{j,\kappa} = a/b$ and $c^j = a'/b'$, p_2 divides then $a'b - ab'$ which is of height $\mathfrak{h}_{\mathbf{F}} + 2^{\kappa-1} \log p_1 + 1$.

Now, p_1 is supposed not dividing A of Theorem 4.1, hence the lifting process succeeds. Let $i_0 \leq \mathfrak{d}$ be the number of iterations required. This implies that the firsts i_0 iterations fail. For each of those, p_2 divides a number of height $\mathfrak{h}_{\mathbf{F}} + 2^{\kappa-1} \log p_1 + 1$, with $1 \leq \kappa < i_0$. Multiplying them, p_2 divides a number B_{p_1} of height at most $(\mathfrak{h}_{\mathbf{F}} + 1)\mathfrak{d} + 2^{\mathfrak{d}} \log p_1$. After simplifications, $h(B_{p_1}) \leq \mathfrak{b}_{\mathbf{F}}$.

Let $C \in \mathbb{N}$ be such that

$$C = \left\lceil \frac{4\mathfrak{a}_{\mathbf{F}} + 2\mathfrak{b}_{\mathbf{F}}}{\varepsilon} \right\rceil, \quad \text{so that } C \leq \frac{1}{2} \exp\left(2\mathfrak{h}_{\mathbf{F}} + 1\right);$$

let Γ be the set of pairs of primes in $[C+1, \ldots, 2C]^2$ and γ be the number of primes in $C+1, \ldots, 2C$; note that $\gamma \geq C/(2 \log C)$ and that $\#\Gamma = \gamma^2$. The upper bound on C shows that all primes p less than $2C$ satisfy the requested inequality $\log p \leq 2\mathfrak{h}_{\mathbf{F}} + 1$. We can then estimate how many choices of (p_1, p_2) in Γ lead to failure. There are at most $\mathfrak{a}_{\mathbf{F}}/\log C$ primes p_1 in $C+1, \ldots, 2C$ which divide the integer A of Theorem 4.1, discriminating at most $\gamma \mathfrak{a}_{\mathbf{F}}/\log C$ pairs (p_1, p_2). For any other value of p_1, there are at most $(\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}})/\log C$ choices of p_2 which divide A and B_{p_1}. This discriminates at most $\gamma(\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}})/\log C$ pairs (p_1, p_2). Thus the number of choices in Γ leading to failure is at most $\gamma(2\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}})/\log C$. The lower bound on γ shows that if (p_1, p_2) is chosen randomly with uniform probability in Γ, the probability that it leads to failure is at most

$$\frac{\gamma(2\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}})}{\#\Gamma \log C} = \frac{\gamma(2\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}})}{\gamma^2 \log C} = \frac{2\mathfrak{a}_{\mathbf{F}} + \mathfrak{b}_{\mathbf{F}}}{\gamma \log C} \leq \frac{4\mathfrak{a}_{\mathbf{F}} + 2\mathfrak{b}_{\mathbf{F}}}{C},$$

which is at most ε, as requested.

To estimate the complexity of this algorithm, note that since we double the precision at each lifting step, the cost of the last lifting step dominates. From the previous discussion, the number of bit operations cost at the last step is quasi-linear in $(L, h_L, \mathsf{C}^n, \deg \mathcal{Z}, 2^\kappa, \log p_1)$. The previous estimates show that at this step, 2^κ is in $O\big(\log(n) \deg \mathcal{Z} + h(\mathcal{Z})\big)$, whereas $\log p_1$ is quasi-linear in $|\log \varepsilon|, \log h, d, \log n$. Putting all these estimates ends the proof of Theorem 4.2.

4.5 Experimentation

We realized a first MAPLE 9.5 implementation of our modular algorithm on top of the `RegularChains` library [41]. Tests on benchmark systems [1] reveal its strong features, compared with two other MAPLE solvers, `Triangularize`, from the `RegularChains` library, and `gsolve`, from the `Groebner` library. Remark that the triangular decompositions modulo a prime, that are needed in our algorithm, are performed by `Triangularize`. This function

Sys	Name	n	d	h	$\mathfrak{h}$
1	Cyclohexane	3	4	3	4395
2	Fee_1	4	4	2	24464
3	fabfaux	3	3	13	2647
4	geneig	6	3	2	116587
5	eco6	6	3	0	105718
6	Weispfenning-94	3	5	0	7392
7	Issac97	4	2	2	1511
8	dessin-2	10	2	7	358048
9	eco7	7	3	0	387754
10	Methan61	10	2	16	450313
11	Reimer-4	4	5	1	55246
12	Uteshev-Bikker	4	3	3	7813
13	gametwo5	5	4	8	159192
14	chemkin	13	3	11	850088102

Table 4.1: Features of the polynomial systems

Sys	p_1	$\mathfrak{d}$	a	C_a
1	4423	7	2	15
2	24499	8	4	70
3	2671	7	5	110
4	116663	10	5	162
5	105761	10	3	40
6	7433	7	3	31
7	1549	6	5	102
8	358079	11	7	711
9	387799	11	4	89
10	450367	11	6	362
11	55313	9	2	19
12	7841	7	5	125
13	159223	10	-	-
14	850088191	18	-	-

Table 4.2: Data for the modular algorithm

Sys	Δ_p	E_p	Lift	Total	Mem.	Output size
1	1	0.3	2	7	5	243
2	3	1	9	20	6	4157
3	8	0.4	6	22	7	5855
4	5	1	5	18	6	4757
5	12	1.5	6	35	6	2555
6	16	1.5	11	43	7	3282
7	66	0.4	4	133	8	4653
8	47	9	232	427	13	122902
9	1515	9	35	2873	11	9916
10	2292	6	82	4686	25	50476
11	3507	1	9	5569	38	2621
12	4879	2	22	8796	63	12870
13	∞	-	-	-	-	-
14	-	-	-	-	fail	-

Table 4.3: Results from our modular algorithm

Sys	`Triang.`	Mem.	Size	`gsolve`	Mem.	Size
1	0.4	4	169	0.2	3	239
2	2	6	1680	504	18	34375
3	512	275	6250	1041	34	27624
4	2.5	4	743	-	fail	-
5	5	5	3134	9	5	2236
6	3000	250	2695	4950	66	34932
7	-	fail	-	1050	31	31115
8	-	fail	-	-	error	-
9	1593	18	55592	-	fail	-
10	∞	-	-	-	fail	-
11	-	fail	-	-	fail	-
12	-	fail	-	-	fail	-
13	-	fail	-	∞	-	-
14	-	fail	-	-	fail	-

Table 4.4: Results from `Triangularize` and `gsolve`

is a generic code: essentially the same code is used over $\mathbb{Z}$ and modulo a prime. Thus, `Triangularize` is not optimized for modular computations.

Our computations are done on a 2799 MHz Pentium 4. For the time being our implementation handles square systems that generate radical ideals. We compare our algorithm called `TriangularizeModular` with `gsolve` and `Triangularize`;

For each benchmark system, Table 4.1 lists the numbers n, d, h $\mathfrak{h}$ and Table 4.2 lists the prime p_1, the *a priori* and actual number of lifting steps ($\mathfrak{d}$ and a) and the maximal height of the output coefficients (C_a). Table 4.3 gives the time of one call to `Triangularize` modulo p_1 (Δ_p), the equiprojectable decomposition (E_p), and the lifting (Lift.) in seconds — the first two steps correspond to the "oracle calls" O_2 mentioned in Theorem 4.2, for which a study is a work in progress. Table 4.3 gives also the total time, the total memory usage and output size for `TriangularizeModular`, whereas Table 4.4 gives that data for `Triangularize` and `gsolve`.

The maximum time is set up to 10800 seconds; we set the probability of success to be at least 90%.

`TriangularizeModular` solves 12 of the 14 test systems before the timeout, while `Triangularize` succeeds with 7 and `gsolve` with 6. Among most of the problems which `gsolve` can solve, `TriangularizeModular` shows less time consumed, less memory usage, and smaller output size. Noticeably, quite a few of the large systems can be solved by `TriangularizeModular` with time extension: system 13 is solved in 18745 seconds. Another interesting system is Pinchon-1 (from the FRISCO project), for which $n = 29, d = 16, h = 20, \mathfrak{h} = 1409536095e + 29$, which we solve in 64109 seconds. Both `Triangularize` and `gsolve` fail these problems due to memory allocation failure. Our modular method demonstrates its efficiency in reducing the size of the intermediate computations, whence its ability to solve difficult problems.

We observed that for every test system, for which E_p can be computed, the Hensel lifting always succeeds, *i.e.* the equiprojectable decomposition over $\mathbb{Q}$ can be reconstructed from E_p. In addition, `TriangularizeModular` failed `chemkin` at the Δ_p phase rather than at the lifting stage. Furthermore, the time consumed in the equiprojectable decomposition and the Hensel lifting is rather insignificant comparing with that in triangular decomposition modulo a prime. For every tested example the Hensel lifting achieves its final goal in less steps than the theoretical bound. In addition, the primes derived from our theoretical bounds are of quite moderate size, even on large examples.

4.6 Conclusions

We have presented a modular algorithm for triangular decompositions of 0-dimensional varieties over $\mathbb{Q}$ and have demonstrated the feasibility of Hensel lifting in computing triangular decompositions of non-equiprojectable varieties. Experiments show the capacity of this approach to improve the practical efficiency of triangular decomposition.

By far, the bottleneck is the modular triangularization phase. This is quite encouraging, since it is the part for which we relied on generic, non-optimized code. The next step is to extend these techniques to specialize variables as well during the modular phase, following the approach initiated in [50] for primitive element representations, and treat systems of positive dimension.

Chapter 5

On the complexity of the D5 principle

The D5 Principle was introduced in 1985 by Jean Della Dora, Claire Dicrescenzo and Dominique Duval in their celebrated note "About a new method for computing in algebraic number fields". This innovative approach automatizes reasoning based on case discussion and is also known as "Dynamic Evaluation". Applications of the D5 Principle have been made in Algebra, Computer Algebra, Geometry and Logic.

Many algorithms for solving polynomial systems symbolically need to perform standard operations, such as GCD computations, over coefficient rings that are direct products of fields rather than fields. This chapter shows how asymptotically fast algorithms for polynomials over fields can be adapted to this more general context, thanks to the D5 Principle. This chapter provides a big part of the proofs, but is still a preliminary study. An extended abstract relating to this question has been published in [121] co-authored with Y. Xie, M. Moreno Maza and É. Schost.

Note after revision of the manuscript: This chapter only puts highlights on the feasibility of our strategy. In particular brutal simplifications and hiding constants in big-O are done without more justification. Also, the emphasize is put on the complexity analysis rather on the proofs of correctness of the algorithms presented. This is particularly true for the Half-GCD, which needs a careful proof extending the arguments of Yap [122] to the product of fields situation. Finally, it misses a carefully analysis of the splitting algorithm 5.2 which is essential and is revealing not easy. The finalization of this work is still in progress.

5.1 Introduction

The standard approach for computing with an algebraic number is through the data of its irreducible minimal polynomial over some base field k. However, in typical tasks such as polynomial system solving, involving many algebraic numbers of high degree, following this approach will require using probably costly factorization algorithms. Jean Della Dora, Claire Dicrescenzo and Dominique Duval introduced "Dynamic Evaluation" techniques (also termed "D5 Principle") as a means to compute with algebraic numbers, while avoiding factorization. Roughly speaking, this approach leads one to compute over *direct products of field extensions of* k, instead of only field extensions.

Applications of Dynamic Evaluation have been made by many authors: [54], [53], [37], [81] and others. Many algorithms for polynomial system solving rely on this philosophy; see,

for instance, the work of [72], [63], [34], [88], [87], [21]. Recently, Noro proposed to handle the splitting for the inverse operation by decomposing the quotient algebra with modular Gröbner basis computations [91]. An implementation is proposed, but complexity estimates are not deducible with this method.

This work is aiming at filling the lack of complexity results for dynamic evaluation. The addition and multiplication over a direct product of fields are easily proved to be *quasi-linear* (in a natural complexity measure). As for the inversion, it has to be replaced by *quasi-inversion*: following the D5 philosophy, meeting zero-divisors in the computation will lead to *splitting* the direct product of fields into a family thereof. It is much more tricky to prove quasi-linear complexity estimate for quasi-inversion, because the algorithm relies on other algorithms, for which such an estimate has to be proved: the GCD and the splitting algorithms.

Direct product of fields will be described using radical *triangular sets* (as usual in this thesis, they are *Lazard* triangular sets, as pointed out in Definition 1.4). In what follows, we assume that the base field k is *perfect*. If T is a radical triangular set, the residue class ring $\mathbb{K}(T) := k[X_1, \ldots, X_n]/\langle T\rangle$ is a direct product of fields. Hence, our questions can be basically rephrased as studying the complexity of operations (addition, multiplication, quasi-inversion) modulo triangular sets. The following notation helps us quantify the complexity of these algorithms.

Definition 5.1. *We denote by* $\deg_i(T)$ *the degree of* T_i *in* X_i*, for all* $1 \leq i \leq n$*, and by* $\deg(T)$ *the product* $\deg_1(T) \cdots \deg_n(T)$*. We call it the* degree *of* T*.*

We recall and generalize the notion of triangular decomposition already introduced for the Split-and-Merge algorithm of Chapter 4, in § 4.2.

Definition 5.2. *A* triangular decomposition *of a zero-dimensional radical ideal* $I \subset k[X_1, \ldots, X_n]$ *is a family* $\mathbf{T} = T^1, \ldots, T^e$ *of triangular sets, such that* $I = \langle T^1\rangle \cap \cdots \cap \langle T^e\rangle$ *and* $\langle T^i\rangle + \langle T^j\rangle = \langle 1\rangle$ *for all* $i \neq j$*.*

A triangular decomposition $\mathbf{T}'$ *of* I refines *another decomposition* $\mathbf{T}$ *if for every* $T \in \mathbf{T}$ *there exists a (necessarily unique) subset* $\mathrm{decomp}(T, \mathbf{T}') \subseteq \mathbf{T}'$ *which is a triangular decomposition of* $\langle T\rangle$*.*

Let T be a triangular set, let $\mathbf{T} = T^1, \ldots, T^e$ be a triangular decomposition of $\langle T\rangle$, and define $\mathbb{K}(\mathbf{T}) := \mathbb{K}(T^1) \times \cdots \times \mathbb{K}(T^e)$. Then by the Chinese remainder theorem, $\mathbb{K}(T) \simeq \mathbb{K}(\mathbf{T})$. Now let $\mathbf{T}'$ be a refinement of $\mathbf{T}$. For each triangular set T^i in $\mathbf{T}$, denote by $U^{i,1}, \ldots, U^{i,e_i}$ the triangular sets in $\mathrm{decomp}(T^i, \mathbf{T}')$. We have the following e isomorphisms:

$$\phi_i : \ \mathbb{K}(T^i) \simeq \mathbb{K}(U^{i,1}) \times \cdots \times \mathbb{K}(U^{i,e_i}), \tag{5.1}$$

which extend to the following e isomorphisms, where y is a new variable.

$$\Phi_i : \ \mathbb{K}(T^i)[y] \simeq \mathbb{K}(U^{i,1})[y] \times \cdots \times \mathbb{K}(U^{i,e_i})[y]. \tag{5.2}$$

Definition 5.3. *For* $\mathbf{h} = (h_1, \ldots, h_e) \in \mathbb{K}(T^1)[y] \times \cdots \times \mathbb{K}(T^e)[y]$*, we call* split *of* $\mathbf{h}$ *with respect to* $\mathbf{T}$ *and* $\mathbf{T}'$*, and write* $\mathrm{split}(\mathbf{h}, \mathbf{T}, \mathbf{T}')$ *the vector* $(\Phi_1(h_1), \ldots, \Phi_e(h_e)) \in \prod_{i=1}^{e} \prod_{j=1}^{e_i} \mathbb{K}(U^{i,j})$.

Note that if $g \in \mathbb{K}(T)[y]$, then we have $\text{split}(g, T, \mathbf{T}') = \text{split}(\text{split}(g, T, \mathbf{T}), \mathbf{T}, \mathbf{T}')$.

We recall the notion of *non-critical* decompositions (used in Chapter 4). It is motivated by the following remark. Let $\mathbf{T} = T^1, \ldots, T^e$ be a family of triangular sets, with $T^j = (T_1^j(X_1), T_2^j(X_1, X_2), \ldots, T_n^j(X_1, \ldots, X_n))$. For $1 \leq i \leq n$, we write $T_{\leq i}^j = (T_1^j(X_1)\,,\, T_2^j(X_1, X_2)\,,\, \ldots\,,\, T_i^j(X_1, \ldots, X_i))$ and define the family $\mathbf{T}_{\leq i}$ by:

$$\mathbf{T}_{\leq i} = \{T_{\leq i}^j \;\mid\; j \leq e\,\} \quad \text{(with no repetition allowed)}.$$

Even if $\mathbf{T}$ is a triangular decomposition of a 0-dimensional radical ideal $I \subset k[X_1, \ldots, X_n]$, $\mathbf{T}_{\leq i}$ is not necessarily a triangular decomposition of $I \cap k[X_1, \ldots, X_i]$. Indeed, with $n = 2$ and $e = 2$, consider $T^1 = ((X_1 - 1)(X_1 - 2), X_2)$ and $T^2 = ((X_1 - 1)(X_1 - 3), X_2 - 1)$. The family $\mathbf{T} = T^1, T^2$ is a triangular decomposition of the ideal $I = \langle T^1 \rangle \cap \langle T^2 \rangle$. However, the family of triangular sets

$$\mathbf{T}_{\leq 1} = \{T_1^1 = (X_1 - 1)(X_1 - 2),\; T_2^1 = (X_1 - 1)(X_1 - 3)\}$$

is not a triangular decomposition of $I \cap k[X_1]$ since $\langle T_1^1 \rangle + \langle T_1^2 \rangle = \langle X_1 - 1 \rangle$.

Definition 5.4. *Let T be a triangular set in $k[X_1, \ldots, X_n]$. Two polynomials $a, b \in \mathbb{K}(T)[y]$ are* coprime *if the ideal $\langle a, b \rangle \subset \mathbb{K}(T)[y]$ equals $\langle 1 \rangle$.*

Definition 5.5. *Let $T \neq T'$ be two triangular sets, with $T = (T_1, \ldots, T_n)$ and $T' = (T_1', \ldots, T_n')$. The least integer ℓ such that $T_\ell \neq T_\ell'$ is called the* level *of the pair $\{T, T'\}$. The pair $\{T, T'\}$ is* critical *if T_ℓ and T_ℓ' are not coprime in $k[X_1, \ldots, X_{\ell-1}]/\langle T_1, \ldots, T_{\ell-1} \rangle[X_\ell]$. A family of triangular sets $\mathbf{T}$ is* non-critical *if it has no critical pairs, otherwise it is said to be* critical.

The pair $\{T^1, T^2\}$ in the above example has level 1 and is critical. Consider $U^{1,1} = (X_1 - 1, X_2)$, $U^{1,2} = (X_1 - 2, X_2)$, $U^{2,1} = (X_1 - 1, X_2 - 1)$ and $U^{2,2} = (X_1 - 3, X_2 - 1)$. Observe that $\mathbf{U} = \{U^{1,1}, U^{1,2}, U^{2,1}, U^{2,2}\}$ is a non-critical triangular decomposition of I refining $\{T^1, T^2\}$ and that $\mathbf{U}_{\leq 1}$ is a triangular decomposition $I \cap k[X_1, X_2]$.

This notion of critical pair is fundamental. In fact, fast algorithms for the innocuous splitting operations Φ_i of Equation (5.2) are not guaranteed for critical decompositions, as shown in the following extension of the previous example. Consider a third triangular set $T^3 = ((X_1 - 2)(X_1 - 3), X_2 + X_1 - 3)$. One checks that $\mathbf{V} = [T^1, T^2, T^3]$ is a triangular decomposition of $T = ((X_1 - 1)(X_1 - 2)(X_1 - 3), X_2(X_2 - 1))$. However, splitting an element p from $[T]$ to $\mathbf{V}$ requires to compute

$$p \bmod (X_1 - 1)(X_1 - 2),\; p \bmod (X_1 - 1)(X_1 - 3),\; p \bmod (X_1 - 2)(X_1 - 3),$$

whence some redundancies. In general, these redundancies prevent the splitting computation from being quasi-linear with respect to $\deg(T)$: since the complexity involves the sum of the degrees of the divisor polynomials, so that redundancies make this degree bigger than $\deg(T)$. But if the triangular decomposition is non-critical, then there is no more redundancy, and the complexity of splitting p can be hoped to be quasi-linear.

Removing critical pairs of a critical triangular decomposition in order to be able to split fast requires to delete the common factors between the polynomials involved in the decomposition. To do it fast (in quasi-linear time), the *coprime factorization* or *gcd-free basis*

computation algorithm is used. Of course to implement this algorithm over a direct product of fields, one first need to be able to compute GCD's over such a product in quasi-linear time.

Since $\mathbb{K}(T)$ is a direct product of fields, any pair of univariate polynomials $f, g \in \mathbb{K}(T)[y]$ admits a GCD h in $\mathbb{K}(T)[y]$, in the sense that the ideals $\langle f, g\rangle$ and $\langle h\rangle$ coincide, see [89]. However, even if f, g are both monic, there may not exist a *monic* polynomial h in $\mathbb{K}(T)[y]$ such that $\langle f, g\rangle = \langle h\rangle$ holds:

Example 5.1: Consider for instance $f = y + \frac{a+1}{2}$ (assuming that 2 is invertible in k) and $g = y + 1$ where $a \in \mathbb{K}(T)$ satisfies $a^2 = a$, $a \neq 0$ and $a \neq 1$ (a possibility is to take $T = (T_1) = (X_1^2 - X_1)$, then $\mathbb{K}(T) \simeq k[X_1]/\langle T\rangle \simeq k[X_1]/\langle X_1\rangle \times k[X_1]/\langle X_1 - 1\rangle$, and $a = (1, 0)$ or $(0, 1)$ in this product). Since $f - g = \frac{a-1}{2}$ then the degree of any generator h of the ideal $\langle f, g\rangle$ is zero. Hence, such a generator h is monic if and only if it is 1, which can be easily proved to be impossible.

GCD's with non-invertible leading coefficients are of limited practical interest; this leads us to the following definition.

Definition 5.6. *Let f, g be two polynomials in $\mathbb{K}(T)[y]$. An* extended greatest common divisor (XGCD) *of f and g is the data of a non-critical decomposition $\mathbf{T} = T^1, \ldots, T^e$ of T and $\mathbf{h}, \mathbf{u}, \mathbf{v}$ sequences of polynomials indexed by the triangular sets in $\mathbf{T}$, such that:*

Let $[f_1, \ldots, f_e] = \mathrm{split}(f, T, \mathbf{T})$ and $[g_1, \ldots, g_e] = \mathrm{split}(g, T, \mathbf{T})$; then:

- *h_i is monic or null,*
- *the inequalities $\deg u_i < \deg g_i$ and $\deg v_i < \deg f_i$ hold,*
- *the equalities $\langle f_i, g_i\rangle = \langle h_i\rangle$ and $h_i = u_i f_i + v_i g_i$ hold.*

For convenience, and especially in Section 5.5, we will simply denote $\gcd(f, g, T)$ the data of $\mathbf{T}, \mathbf{g}$.

One easily checks that such XGCD's exists, and can be computed, for instance by applying the D5 Principle to the Euclidean algorithm. To compute GCD's in quasi-linear time over a direct product of fields, we will actually adapt the *Half-GCD* techniques [122] in Section 5.3.

Our last basic ingredient is to take into account non-critical decomposition and monic leading coefficients for the inverse operation, as a suitable generalization of the notion of inverse to direct products of fields.

Definition 5.7. *A* quasi-inverse *of an element $f \in \mathbb{K}(T)$ is the data of a non-critical decomposition $\mathbf{T} = T^1, \ldots, T^e$ of T and a sequence $\mathbf{u}$ indexed by the triangular sets in $\mathbf{T}$, such that:*

Let $[f_1, \ldots, f_e] = \mathrm{split}(f, T, \mathbf{T})$; then for $1 \leq i \leq e$ we have either $f_i = u_i = 0$, or $f_i u_i = 1$.

Obtaining fast algorithms for GCD's, quasi-inverses and removal of critical pairs requires a careful inductive process (Figure 5.1).

- We first need complexity estimates for multiplication modulo a triangular set and splitting with respect to triangular decompositions. This is done in Section 5.2.

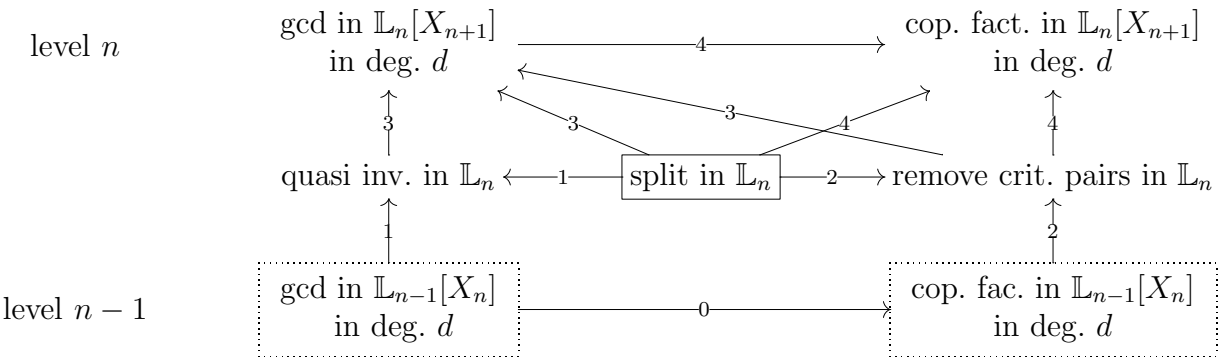

Figure 5.1: The inductive process of the proof: from level $n-1$ to level n

- Assuming that multiplications and quasi-inverse computations can be computed fast in $\mathbb{K}(T)$, and assuming that we can remove critical pairs from critical triangular decompositions of $\langle T \rangle$, we obtain in Section 5.3 a fast algorithm for computing GCD's in $\mathbb{K}(T)[y]$. In the article [69], it is stated that GCD's over products of fields can be computed in quasi-linear time, but with not a clue for a proof, underlying that it might have been obvious; the author (and the referees) probably missed the problem arisen by the splitting of an element after a decomposition.
- Assuming that GCD's can be computed fast in $\mathbb{K}(T_1, \ldots, T_{n-1})[X_n]$, we present fast algorithms for quasi-inverses in $\mathbb{K}(T)$ (Section 5.4), coprime factorization for polynomials in $\mathbb{K}(T_1, \ldots, T_{n-1})[X_n]$ (Section 5.5) and refining a triangular decomposition $\mathbf{T}$ of T into a non-critical one (Section 5.6).

More precisely, the way how the proof is built is done through the multiple crossed induction (Fig. 5.1). The cost of the splitting operation is proved in all generality and is not involved in the induction hypothesis. What is supposed is the cost of the gcd operation and coprime factorization modulo triangular set in $n-1$ variables (both put in a box in the figure). Then we deduce quasi-inversion modulo triangular set in n variables (step 1), as well as the critical pairs removal (step 2). Then the estimate for gcds computation modulo triangular sets in n variables is obtained (step 3), and finally, so it is for the critical pairs removal in n variables (step 4). This achieves the proof by induction.

These are the basic blocks for our inductive process, which yields our main results:

Theorem 5.1. *There exists a constant* C *independent of* T *and of the degree of the polynomials of* T *such that addition, multiplication and quasi-inversion in* $\mathbb{K}(T)$ *can be computed in* $\mathsf{C}^n \prod_{1 \leq i \leq n} \mathsf{M}(d_i)\mathrm{logp}(d_i)^3$ *operations over* k.

Theorem 5.2. *One can compute an extended greatest common divisor of polynomials in* $\mathbb{K}(T)[y]$*, with degree at most* d*, using at most* $\mathsf{C}^n \prod_{1 \leq i \leq n} \mathsf{M}(d_i)\mathrm{logp}(d_i)^3\mathsf{M}(d)\mathrm{logp}(d)$.

We now define our key complexity notion, arithmetic time for triangular sets.

Definition 5.8. *An* arithmetic time *is a function* $T \mapsto \mathsf{A}_n(T)$ *with real positive values and defined over all triangular sets in* $k[X_1, \ldots, X_n]$ *such that the following conditions hold.*

(E_0) *For every triangular decomposition* $\mathbf{T} = [T^1, \ldots, T^e]$ *of* T*, we have* $\mathsf{A}_n(T^1) + \cdots + \mathsf{A}_n(T^e) \leq \mathsf{A}_n(T)$.

(E_1) *Every addition or multiplication in* $\mathbb{K}(T)$ *can be done in at most* $\mathsf{A}_n(T)$ *operations in* k.

(E_2) *Every quasi-inverse in* $\mathbb{K}(T)$ *can be computed in at most* $\mathsf{A}_n(T)$ *operations in* k.

(E_3) *Given a triangular decomposition* $\mathbf{T}$ *of* T, *one can compute a non-critical triangular decomposition* $\mathbf{T}'$ *which refines* $\mathbf{T}$, *in at most* $\mathsf{A}_n(T)$ *operations in* k.

(E_4) *For every* $\alpha \in \mathbb{K}(T)$ *and every non-critical triangular decomposition* $\mathbf{T}$ *of* T, *one can compute* $\mathrm{split}(\alpha, T, \mathbf{T})$ *in at most* $\mathsf{A}_n(T)$ *operations in* k.

Our main goal in this paper is then to give estimates for arithmetic times. This is done through an inductive proof; the following proposition gives such a result for the base case, triangular sets in one variable.

Proposition 5.1. *If* $n = 1$, *then* $T \in k[X_1] \mapsto \mathsf{C}\,\mathsf{M}(\deg T)\mathrm{logp}(\deg T)$ *is an arithmetic time.*

Proof: A triangular set in one variable is simply a squarefree monic polynomial in $k[X_1]$. Hence, (E_1), (E_2) and (E_4) respectively follow from points 2, 6 and 4 in Proposition 1.7. Property (E_0) is clear. Since $n = 1$, all triangular decompositions are non-critical, and (E_3) follows. □

5.2 Basic complexity results: multiplication and splitting

In this section, upper bounds on the cost of multiplication modulo a triangular set, and the splitting of an element defined over a triangular set, onto a decomposition of it. In general, we do not know how to perform this last operation in quasi-linear time; however, when the decomposition is non-critical, quasi-linearity can be reached.

Proposition 5.2. *Let* M *be a multiplication function, and let* C_M *be the constant from Proposition 1.7. Let* T *be a triangular set in* $k[X_1, \ldots, X_n]$. *Then:*

- *Additions and multiplications modulo* T *can be done in at most* $\mathsf{C}_M^n \prod_{i \leq n} \mathsf{M}(\deg_i T)$ *operations in* k.
- *If* $\mathbf{T}$ *is a non-critical decomposition of* T, *then for any* h *in* $\mathbb{K}(T)$, $\mathrm{split}(h, T, \mathbf{T})$ *can be computed in at most* $n\,\mathsf{C}_M^n \prod_{i \leq n} \mathsf{M}(\deg_i T)\mathrm{logp}(\deg_i T)$ *operations in* k.

Proof: The first part of the proposition is easy to deal with: the case of additions is obvious, using the inequality $\mathsf{M}(d) \geq d$; as to multiplication, an easy induction using point 1. in Proposition 1.7 gives the result. The end of the proof uses point 4. in Proposition 1.7; the non-critical assumption is then used through the following lemma. □

Lemma 5.1. *Consider a non-critical decomposition* $\mathbf{T}$ *of the triangular set* $T = (T_1, \ldots, T_n)$. *Write* $\mathbf{T}_{\leq n-1} = [U^1, \ldots, U^s]$, *and, for all* $i \leq s$, *denote by* $T^{i,1}, \ldots, T^{i,e_i}$ *the triangular sets in* $\mathbf{T}$ *such that* $T^{i,j} \cap k[X_1, \ldots, X_{n-1}] = U^i$ *(thus* $\mathbf{T}$ *is the set of all* $T^{i,j}$,

split$(a, T, \mathbf{S})$ ==

Input An element $a \in \mathbb{K}(T)$, given as an n-variate polynomial in normal form for T, and a non-critical decomposition $\mathbf{S}$ of T

Output The sequence **res** of a mod S for all $S \in \mathbf{S}$.

1: Write $a(X_1, \ldots, X_n) = \sum_{i=0}^{d_n-1} a_i(X_1, \ldots, X_{n-1}) X_n^i$
2: **for** $i = 0, \ldots, d_n - 1$ **do**
3: $\quad \mathbf{a}_i \leftarrow \text{split}(a_i, T_{\leq n-1}, \mathbf{S}_{\leq n-1})$
4: **end for**
5: **res** $\leftarrow [\,]$
6: **for** $R \in \mathbf{S}_{<n-1}$ **do**
7: $\quad P \leftarrow \sum_{i=0}^{d_n-1} a_i^R X_n^i$
8: $\quad \mathbf{s}_n^R \leftarrow \{S_n\,,\ S \in \mathbf{S} \,|\, (S_1, \ldots, S_{n-1}) = R\}$
9: $\quad$ **res** $\leftarrow$ **res** cat multiRem$(P, \mathbf{s}_n^R, R)$
10: **end for**
11: **return res**

split$(\mathbf{a}, \mathbf{S}, \mathbf{T})$

Input A triangular decomposition $\mathbf{S}$ of a triangular set T, a sequence $\mathbf{a} = [a^S, S \in \mathbf{S}]$ with $a^S \in \mathbb{K}(S)$, and a non-critical refinement $\mathbf{T}$ of $\mathbf{S}$

Output The sequence **res** of Definition 5.3

1: **res** $\leftarrow [\,]$
2: **for** $S \in \mathbf{S}$ **do**
3: $\quad \mathbf{U} \leftarrow \text{decomp}(S, \mathbf{T})$
4: $\quad \mathbf{a} \leftarrow \text{split}(a, S, \mathbf{U})$
5: $\quad$ **res** $\leftarrow$ **res** cat **a**
6: **end for**
7: **return res**

Algo **5.2:** Splitting onto a non-critical decomposition

with $i \leq s$ and $j \leq e_i$). Then $\mathbf{T}_{\leq n-1}$ is a non-critical decomposition of the triangular set $(T_1, \ldots, T_{n-1})$. Moreover, for all $i \leq s$, we have:

$$\sum_{j \leq e_i} \deg_n T^{i,j} = \deg_n T.$$

As an illustration, consider again, for $n = 2$, the triangular sets

$$\begin{aligned} T^1 &= ((X_1 - 1)(X_1 - 2),\ X_2) \\ T^2 &= ((X_1 - 1)(X_1 - 3),\ X_2 - 1) \\ T^3 &= ((X_1 - 2)(X_1 - 3),\ X_2 + X_1 - 3). \end{aligned}$$

These triangular sets form a critical decomposition $\mathbf{T}$ of the ideal $\langle T^1 \rangle \cap \langle T^2 \rangle \cap \langle T^3 \rangle$, which is also generated by $T = ((X_1 - 1)(X_1 - 2)(X_1 - 3), X_2(X_2 - 1))$.

Here, $\mathbf{T}_{\leq 1}$ is given by $[U^1, U^2, U^3] = [(X_1-1)(X_1-2), (X_1-1)(X_1-3), (X_1-1)(X_1-3)]$, so that $s = 3$. Take for instance $U^1 = (X_1-1)(X_1-2)$; then we have $e_1 = 1$ and $T^{1,e_1} = T^1$.

Note then that $\deg_2(T^{1,e_1}) = 1$ differs from $\deg_2(T) = 2$, so the conclusion of the previous lemma is indeed violated.

Proof of Lemma 5.1 This lemma shows the real interest of using non-critical pairs for fast splitting. The main task will be for the rest of the chapter to remove critical pairs from a given triangular decomposition. Let us start the proof of the lemma with two intermediate results of commutative algebra.

Lemma 5.2. *Let $J_1, \ldots, J_s$ be ideals in a ring R, such that $J_i + J_{i'} = \langle 1 \rangle$ holds for all $i \neq i'$. Then for any ideal I in R, we have the relation*

$$I + (J_1 \cap \cdots \cap J_s) = (I + J_1) \cap \cdots \cap (I + J_s).$$

Proof: All ideals $I + J_i$ are pairwise coprime, so their intersection equals their product, and their product is easily seen to be contained in $I + (J_1 \cap \cdots \cap J_s)$. The other inclusion is clear. □

Lemma 5.3. *Let $I_1, \ldots, I_s$ and $J_1, \ldots, J_s$ be ideals in a ring R, such that*

$$I_1 \cap \cdots \cap I_s = J_1 \cap \cdots \cap J_s \tag{5.3}$$

holds. Suppose also that for all $i \neq i'$, the equalities $I_i + I_{i'} = J_i + J_{i'} = I_i + J_{i'} = \langle 1 \rangle$ hold. Then $I_i = J_i$ for all i.

Proof: Take the sum of both sides of Equation (5.3) with I_i; applying the previous lemma to both sides gives the equality $I_i = I_i + J_i$. Proceeding similarly with J_i yields $I_i + J_i = J_i$, concluding the proof. □

Proof of Lemma **??**: Let U^i and U^j and T^i and T^j be in $\mathbf{T}$ such that $U^i = (T^i_1, \ldots, T^i_{n-1})$ and $U^j = (T^j_1, \ldots, T^j_{n-1})$. Since U^i and U^j differ, the level ℓ of T^i and T^j is at most $n-1$. Then, coprimality at level ℓ for T^i and T^j implies coprimality at level ℓ for U^i and U^j.

The pairwise coprimality of $T^{i,1}, \ldots, T^{i,e_i}$ modulo $\langle U^i \rangle$ implies that

$$\bigcap_{j \leq e_i} \langle T^{i,j} \rangle = \langle U^i \rangle + \langle T^{i,1} \cdots T^{i,e_i} \rangle.$$

We write $A_i = \langle U^i \rangle + \langle T^{i,1} \cdots T^{i,e_i} \rangle$. Note that we have the equality $A_i + A_{i'} = \langle 1 \rangle$.

Next, from the definition of a triangular decomposition, we have the equality between ideals in $k[X_1, \ldots, X_n]$:

$$\langle T \rangle = \bigcap_{i \leq s} \bigcap_{j \leq e_i} \langle T^{i,j} \rangle = \bigcap_{i \leq s} A_i. \tag{5.4}$$

On the other hand, note the equality in $k[X_1, \ldots, X_{n-1}]$

$$\langle T_1, \ldots, T_{n-1} \rangle = \bigcap_{i \leq s} \langle U^i \rangle,$$

which extends to an equality in $k[X_1, \ldots, X_n]$. Since the ideals $\langle U^i \rangle$ are pairwise coprime, Lemma 5.2 gives the equality

$$\langle T \rangle = \bigcap_{i \leq s} \langle U^i \rangle + \langle T_n \rangle. \tag{5.5}$$

Applying Lemma 5.3 to Equations (5.4) and (5.5), we deduce that $A_i = \langle U^i \rangle + \langle T_n \rangle$. □

5.3 Fast GCD computations modulo a triangular set

GCD's of univariate polynomials over a field can be computed in quasi-linear time by means of the *Half-GCD* algorithm [22, 122]. We show how to adapt this technique over the direct product of fields $\mathbb{K}(T)$ and how to preserve its complexity class. Throughout this section, we consider an arithmetic time $T \mapsto \mathsf{A}_n(T)$ for triangular sets in $k[X_1, \ldots, X_n]$.

Proposition 5.3. *For all $a, b \in \mathbb{K}(T)[y]$ with $\deg a$, $\deg b \leq d$, one can compute an extended greatest common divisor of a and b in $O(\mathsf{M}(d)\log(d))\mathsf{A}_n(T)$ operations in k.*

We prove this result by describing our GCD algorithm over the direct product of fields $\mathbb{K}(T)$ and its complexity estimate. We start with two auxiliary algorithms.

Monic forms. Any polynomial over a field can be made monic by division through its leading coefficient. Over a product of fields, this division may induce splittings. We now study this issue.

Definition 5.9. *A* monic form *of a polynomial $f \in \mathbb{K}(T)[y]$ is the data of $\mathbf{S}, \mathbf{u}, \mathbf{v}, \mathbf{m}$ where $\mathbf{S}$ is a non-critical triangular decomposition of T and $\mathbf{u}, \mathbf{v}, \mathbf{m}$ are sequences indexed by $\mathbf{S}$, verifying:*

Let $[f^S,\ S \in \mathbf{S}] = \mathrm{split}(f, T, \mathbf{S})$, so that $f^S \in \mathbb{K}(S)[y]$ and let $\mathrm{lc}(f^S)$ the leading coefficient of f^S. Then for all $S \in \mathbf{S}$, we have $u^S = \mathrm{lc}(f^S)$, $m^S = v^S f^S$ and either $u^S = v^S = 0$ or $u^S v^S = 1$.

Observe that for all $S \in \mathbf{S}$, the polynomial m^S is monic or null.

Algorithm 5.3 shows how to compute a monic form. This function uses a procedure quasiInverse($\mathbf{f}, \mathbf{T}$), that will be defined in § 5.4 Definition 5.7.

The number at the end of a line, multiplied by $\mathsf{A}_n(T)$, gives an upper bound for the total time spent at this line. Therefore, the following algorithm computes a monic form of f in at most $(8d + 6)\mathsf{A}_n(T)$ operations in k.

Division with monic remainder. The previous notion can then be used to compute Euclidean divisions, producing *monic* remainders: they will be required in our fast Euclidean algorithm for XGCD's.

Definition 5.10. *Let $f, g \in \mathbb{K}(T)[y]$ with g monic. A* division with monic remainder *of f by g is the data of $\mathbf{S}, \mathbf{g}, \mathbf{q}, \mathbf{v}, \mathbf{u}, \mathbf{r}$. where $\mathbf{S}$ is a non-critical decomposition of T and $\mathbf{g}, \mathbf{q}, \mathbf{v}, \mathbf{u}, \mathbf{r}$ are polynomials in y indexed by the triangular sets in $\mathbf{S}$, such that:*

Let $\mathbf{f} := [f^S,\ S \in \mathbf{S}] = \mathrm{split}(f, T, \mathbf{S})$ and $\mathbf{g} := [g^S,\ S \in \mathbf{S}] = \mathrm{split}(g, T, \mathbf{S})$. Then, for all $S \in \mathbf{S}$, $f^S = g^S q^S + u^S r^S$, and $\mathbf{S}, \mathbf{u}, \mathbf{v}, \mathbf{r}$ is a monic form of the remainder of the Eucidean division of f by g.

Algorithm5.4 computes a division with monic remainder of f by g and requires at most $(5\mathsf{M}(d) + O(d))\mathsf{A}_n(T)$ operations in k. We write $(q, r) = \mathrm{div}(f, g)$ for the quotient and the remainder in the (standard) division with remainder in $\mathbb{K}(T)[y]$.

monic(f, T) ==

Input Two polynomials f and g defined over $\mathbb{K}(T)$

Output A non-critical triangular decomposition $\mathbf{T}$ of T, and some decomposed elements $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{m}$ over $\mathbf{T}$ as specified in Definition 5.9

1: $\mathbf{T} \leftarrow [T]$; $\mathbf{v} \leftarrow [0]$; $g \leftarrow f$
2: **while** $g \neq 0$ **do**
3: $\quad \mathbf{u} \leftarrow \text{split}(\text{leadingCoefficient}(g), T, \mathbf{T})$ $[d+1]$
4: $\quad \mathbf{T}', \mathbf{w} \leftarrow \text{quasiInverse}(\mathbf{u}, \mathbf{T})$ $[3d+3]$
5: $\quad \mathbf{v} \leftarrow \text{split}(\mathbf{v}, \mathbf{T}, \mathbf{T}')$; Write $\mathbf{v} = [v^R, R \in \mathbf{T}']$ $[d+1]$
6: $\quad$ **for** $R \in \mathbf{T}'$ **do**
7: $\quad\quad$ **if** $v^R = 0$ **then** $v^R \leftarrow w^R$; **end if** $[d+1]$
8: $\quad$ **end for**
9: $\quad \mathbf{T} \leftarrow \mathbf{T}'$
10: $\quad g \leftarrow g - \text{leadingTerm}(g)$
11: **end while**
12: $\mathbf{f} \leftarrow \text{split}(f, T, \mathbf{T})$ $[d]$
13: $\mathbf{u} \leftarrow \text{leadingCoefficient}(\mathbf{f})$
14: $\mathbf{m} \leftarrow \mathbf{v} \cdot \mathbf{f}$ $[d]$
15: **return** $\mathbf{T}, \mathbf{u}, \mathbf{v}, \mathbf{m}$

Algo **5.3:** Monic form of a polynomial over a triangular set

XGCD's. We are now ready to generalize the *Half-Gcd* method as exposed in [122]. We introduce the following operations. For $a, b \in \mathbb{K}(T)[y]$ with $0 < \deg b < \deg a = d$, each of the following algorithms $\mathrm{M}_{\mathrm{gcd}}(a, b, T)$ and $\mathrm{M}_{\mathrm{hgcd}}(a, b, T)$ returns a sequence $\mathbf{S}, \mathbf{M}$

(s_1) $\mathbf{S}$ is a non-critical triangular decomposition of T,

(s_2) $\mathbf{M} = [M^S,\ S \in \mathbf{S}]$ is a sequence of square matrix of order 2 indexed by $\mathbf{S}$. i.e. M^S has coefficients in $\mathbb{K}(S)[y]$,

such that, if we define $[a^S,\ S \in \mathbf{S}] = \text{split}(a, T, \mathbf{S})$ and $[b^S,\ S \in \mathbf{S}] = \text{split}(b, T, \mathbf{S})$, then, for all $S \in \mathbf{S}$ defining $(t^S, s^S) = (a^S, b^S)\ {}^tM^S$, we have

(s_3) in the case of $\mathrm{M}_{\mathrm{gcd}}$, the polynomial t^S is a GCD of a^S, b^S and $s^S = 0$ holds,

(s_3') in the case of $\mathrm{M}_{\mathrm{hgcd}}$, the ideals $\langle t^S, s^S \rangle$ and $\langle a^S, b^S \rangle$ of $\mathbb{K}(S)[y]$ are identical, and $\deg s^S < \lceil d/2 \rceil \leq \deg t^S$ holds.

Algorithm 5.5 below implements $\mathrm{M}_{\mathrm{gcd}}(a, b, T)$, and is an extension of the analogue algorithm known over fields. Observe that if the input triangular set T is not decomposed during the algorithm, in particular if $\mathbb{K}(T)$ is a field, then the algorithm yields generators of the ideal $\langle a, b \rangle$.

Now, we give running time estimates for $\mathrm{M}_{\mathrm{hgcd}}(a, b, T)$ and $\mathrm{M}_{\mathrm{gcd}}(a, b, T)$. For $0 < \deg b < \deg a = d$, we denote by $G(d)$ and $H(d)$ respective upper bounds for the running time of $\mathrm{M}_{\mathrm{gcd}}(a, b)$ and $\mathrm{M}_{\mathrm{hgcd}}(a, b)$, in the sense that both operations can be done in respective times $G(d)\mathsf{A}_n(T)$ and $H(d)\mathsf{A}_n(T)$.

mdiv(f, g, T) ==

Input Two polynomials f and g defined over $\mathbb{K}(T)$

Output A non-critical triangular decomposition $\mathbf{S}$ of T, and the output as specified in Definition 5.10

1: $(q, r) \leftarrow \text{div}(f, g)$ $[5\mathsf{M}(d) + O(d)]$
2: $\mathbf{T}, \mathbf{u}, \mathbf{v}, \mathbf{r} \leftarrow \text{monic}(r, T)$ $[O(d)]$
3: $\mathbf{q} \leftarrow \text{split}(q, T, \mathbf{T})$ $[d + 1]$
4: $\mathbf{g} \leftarrow \text{split}(g, T, \mathbf{T})$ $[d]$
5: **return** $\mathbf{S}, \mathbf{g}, \mathbf{q}, \mathbf{r}, \mathbf{u}, \mathbf{v}$

Algo **5.4:** Division with monic remainder

The number at the end of an above line, multiplied by $\mathsf{A}_n(T)$, gives an upper bound of the running time of this line. These estimates follow from the super-linearity of the arithmetic time for triangular sets, the running time estimates of the operation mdiv(f, g, T) and classical degree bounds for the intermediate polynomials in the Extended Euclidean Algorithms; see for instance Chapter 3 in [117]. Therefore, counting precisely the degrees appearing, we have: $G(d) \leq G(d/2) + H(d) + (33/2)\mathsf{M}(d) + O(d)$. The operation $\mathrm{M}_{\text{hgcd}}(a, b, T)$ makes two recursive calls with input polynomials of degree at most $d/2$, leading to $H(d) \leq 2H(d/2) + (33/2)\mathsf{M}(d) + O(d)$. The super-linearity of M implies

$$H(d) \leq \frac{33}{2}\mathsf{M}(d)\,\log d + O(d\,\log\,d) \quad \text{and} \quad G(d) \leq 2H(d) + 2\mathsf{M}(d) + O(d).$$

This leads to the result reported in Proposition 5.3.

Specification of gcd. We conclude with a specification of the gcd function used in the remaining sections. For a triangular decomposition $\mathbf{T} = [T^1, \ldots, T^e]$ of T, two sequences $\mathbf{f} = [f_1, \ldots, f_e]$ and $\mathbf{g} = [g_1, \ldots, g_e]$ of polynomials in $\mathbb{K}(T^1)[y], \ldots, \mathbb{K}(T^e)[y]$, the operation xgcd$(\mathbf{f}, \mathbf{g}, \mathbf{T})$ returns a sequences $\mathbf{S}, \mathbf{h}, \mathbf{u}, \mathbf{v}$ where:

- $\mathbf{S}$ is non-critical refinement of $\mathbf{T}$
- for all $i = 1, \ldots, e$, let xgcd$(f_i, g_i, T^i) := \mathbf{S}_i, \mathbf{u}_i, \mathbf{v}_i$, and let $\mathbf{S}$:=removeCriticalPairs(cat$_{i=1}^{e}\mathbf{S}_i$)
- then $\mathbf{h} := \text{split}(\mathbf{h}_i, \mathbf{S}_i, \mathbf{S})$, $\mathbf{u} := \text{split}(\mathbf{u}_i, \mathbf{S}_i, \mathbf{S})$, $\mathbf{v} := \text{split}(\mathbf{v}_i, \mathbf{S}_i, \mathbf{S})$.

This specification also extends to the gcd$(\mathbf{f}, \mathbf{g}, \mathbf{T})$.

Proposition 5.3 implies that if $f_1, \ldots, f_e, g_1, \ldots, g_e$ have degree at most d then xgcd$(\mathbf{f}, \mathbf{g}, \mathbf{T})$ runs in at most $O(\mathsf{M}(d)\log(d))\mathsf{A}_n(T)$ operations in k.

5.4 Fast computation of quasi-inverses

Throughout this section, we consider an arithmetic time A_{n-1} for triangular sets in $n-1$ variables. We explain how a quasi-inverse can be computed fast with the algorithms *split*, *xgcd*, and *removeCriticalPairs*.

$\mathrm{M}_{\mathrm{gcd}}$((a,b,T)) ==

Input $a, b \in \mathbb{K}(T)[y],\ d \geq \deg(a) > \deg(b) \geq 0$

Output A non-critical decomposition $\mathbf{T}'$ of T and a sequence of 2×2 matrices indexed by $\mathbf{T}'$, verifying properties (s_1) and (s_2) above.

1: $\mathbf{G} \leftarrow [\,]$; $\mathbf{T} \leftarrow [\,]$
2: $[M^U,\ U \text{ in } \mathbf{U}] \leftarrow \mathrm{M}_{\mathrm{hgcd}}(a, b, T)$ $[H(d)]$
3: $[a^U,\ U \text{ in } \mathbf{U}] \leftarrow \mathrm{split}(a, T, \mathbf{U})$ $[O(d)]$
4: $[b^U,\ U \text{ in } \mathbf{U}] \leftarrow \mathrm{split}(b, T, \mathbf{U})$ $[O(d)]$
5: **for** U in $\mathbf{U}$ **do**
6: $\quad (t^U, s^U) \leftarrow (a^U, b^U)\ {}^tM^U$ $[4\,\mathsf{M}(d) + O(d)]$
7: $\quad$ **if** $s^U = 0$ **then**
8: $\quad\quad \mathbf{G} \leftarrow \mathbf{G}$ cat $[M^U]$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $[U]$
9: $\quad$ **end if**
10: $\quad \mathbf{W}, \mathbf{s}, \mathbf{q}, \mathbf{r}, \mathbf{u}, \mathbf{v} \leftarrow \mathrm{mdiv}(t^U, s^U, U)$ $[\frac{5}{2}\mathsf{M}(d) + O(d)]$
11: $\quad \mathbf{W}, [M^W,\ W \in \mathbf{W}] \leftarrow \mathrm{split}(M^U, U, \mathbf{W})$ $[O(d)]$
12: $\quad$ **for** W in $\mathbf{W}$ **do**
13: $\quad\quad M^W \leftarrow \begin{pmatrix} 0 & 1 \\ v^W & -q^W v^W \end{pmatrix} M^W$ $[2\,\mathsf{M}(d) + O(d)]$
14: $\quad\quad$ **if** $r^W = 0$ **then** $\mathbf{G} \leftarrow \mathbf{G}$ cat $\left[M^W\right]$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $[W]$
15: $\quad\quad [N^S,\ S \text{ in } \mathbf{S}] \leftarrow \mathrm{M}_{\mathrm{gcd}}(s^W, r^W, W)$ $[G(d/2)]$
16: $\quad\quad [M^S,\ S \in \mathbf{S}] \leftarrow \mathrm{split}(M^W, W, \mathbf{S})$ $[O(d)]$
17: $\quad\quad$ **for** S in $\mathbf{S}$ **do**
18: $\quad\quad\quad M^S \leftarrow N^S \cdot M^S$ $[8\,\mathsf{M}(d) + O(d)]$
19: $\quad\quad\quad \mathbf{G} \leftarrow \mathbf{G}$ cat $[M^S]$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $[S]$
20: $\quad\quad$ **end for**
21: $\quad$ **end for**
22: **end for**
23: $\mathbf{T}' \leftarrow \mathrm{removeCriticalPairs}(\mathbf{T})$; $\mathbf{M} \leftarrow \mathrm{split}(\mathbf{G}, \mathbf{T}, \mathbf{T}')$ $[O(d)]$
24: **return** $\mathbf{T}', \mathbf{M}$

Algo **5.5:** Half-GCD Modulo a Triangular Set

Proposition 5.4. *Let $T = (T_1, \ldots, T_n)$ be a triangular set with $\deg_i(T) = d_i$ for all $1 \leq i \leq n$. Let f be in $\mathbb{K}(T)$. Then one can compute a quasi-inverse of f modulo T in $O\big(\mathsf{M}(d_n)\log(d_n)\big)\mathsf{A}_{n-1}(T_{<n})$ operations in k.*

We consider first the case where f is a non-constant polynomial and its degree w.r.t. X_n is positive and less than d_n; we give the algorithm, followed by the necessary explanations. Here, the quantity at the end of a line, once multiplied by $\mathsf{A}_{n-1}(T_{<n})$, gives the total amount of time spent at this line. At the end of this section, we briefly discuss the other cases to be considered for f.

We first calculate an extended greatest common divisor of f and T_n modulo the triangular set $T_{<n} = [T_1, \ldots, T_{n-1}]$. This induces a non-critical decomposition $\mathbf{S}$ of $T_{<n}$. For further operations, we compute the images of T_n and f over this decomposition in Step 2. and 3.

Let $S \subset k[X_1, \ldots, X_{n-1}]$ a triangular set in $\mathbf{S}$. If the value of g^S is 1, then u^S is the inverse of f modulo (S, t_n^S). Otherwise, $\deg g^S > 0$, and the computation needs to be split into two branches.

quasiInverse(f, T) ==

Input

Output

1: $\mathbf{S}, [\mathbf{g}, \mathbf{u}, \mathbf{v}] \leftarrow \text{xgcd}(f, T_n, T_{<n})$ $\quad [O\big(\mathsf{M}(d_n)\log(d_n)\big)]$
2: $\mathbf{t}_n \leftarrow \text{split}(T_n, T_{<n}, \mathbf{S})$ $\quad [O(d_n)]$
3: $\mathbf{f} \leftarrow \text{split}(f, T_{<n}, \mathbf{S})$ $\quad [O(d_n)]$
4: $\mathbf{T} \leftarrow [\,]$; $\mathbf{u} \leftarrow [\,]$
5: **for** $S \in \mathbf{S}$ **do**
6: **if** $\deg(g^S) = 0$ **then**
7: $\mathbf{u} \leftarrow \mathbf{u}$ cat $[u^S]$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $\big[S$ cat $[t_n^S]\big]$
8: **else**
9: $\mathbf{u} \leftarrow \mathbf{u}$ cat $[0]$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $\big[S$ cat $[g^S]\big]$
10: $q^S \leftarrow t_n^S$ quo g^S $\quad [5\mathsf{M}(d_n) + O(d_n)]$
11: $\mathbf{W}, [\mathbf{1}, \mathbf{u}', \mathbf{v}'] \leftarrow \text{xgcd}(f^S, q^S, S)$ $\quad [O\big(\mathsf{M}(d_n)\log(d_n)\big)]$
12: $\mathbf{p} \leftarrow \text{split}(q^S, S, \mathbf{W})$ $\quad [O(d_n)]$
13: $\mathbf{u} \leftarrow \mathbf{u}$ cat $\mathbf{u}'$; $\mathbf{T} \leftarrow \mathbf{T}$ cat $\big[[W$ cat $[p^W]],\ W \in \mathbf{W}\big]$
14: **end if**
15: **end for**
16: $\mathbf{R}_{<n} \leftarrow \text{removeCriticalPairs}(\mathbf{T}_{<n})$ $\quad [O(1)]$
$\mathbf{T}_{<n}$ is by construction a triangular decomposition of $T_{<n}$
17: $\mathbf{t}_n \leftarrow [S_n,$ the n-th polynomial of the triangular set S in $\mathbf{T}]$
18: $\mathbf{r}_n \leftarrow \text{split}(\mathbf{t}_n, \mathbf{T}_{<n}, \mathbf{R}_{<n})$
19: $\mathbf{w} \leftarrow \text{split}(\mathbf{u}, \mathbf{T}_{<n}, \mathbf{R}_{<n})$
20: $\mathbf{W} \leftarrow$ cat $_{R \in \mathbf{R}_{<n}} \big[[R$ cat $[r_n]],\ r_n \in \mathbf{r}_n^R\big]$
21: **return** $\mathbf{W}, \mathbf{w}$

Algo **5.6:** Quasi-inverse

In one branch, at line 9., we build the triangular set (S, g^S), modulo which f reduces to zero. In the other branch, starting from line 10., we build the triangular set as (S, q^S), modulo which f is invertible. Indeed since the triangular set (S, q^S) generates a radical ideal, t_n^S is squarefree modulo S, and $\gcd(f^s, q^S, S)$ must be 1 modulo (S, q^S). Therefore we can simply use the *xgcd* (Step 11) once to compute the quasi-inverse of f modulo (S, q^S).

After collecting all the quasi-inverses, we remove the critical pairs in the new family of triangular sets. Since no critical pairs are created at level n in the previous computation, the removal of critical pairs needs only to be performed below level n. Regarding the induction hypothesis, this step costs $O(1)\mathsf{A}_{n-1}(T_{<n})$. At the end, we split the inverses and the top polynomials w.r.t the last non-critical decomposition.

We also need quasi-inverse computations in two other different situations. One is when f may not have the same main variable as the triangular set T. Second is to handle the quasi-inverses in the sense of quasiInverse$(\mathbf{f}, \mathbf{T})$ introduced in Section 5.3 Algorithm 5.4 where $\mathbf{T}$ is a triangular decomposition of T, and $\mathbf{f}$ is a sequence indexed by the triangular sets in $\mathbf{T}$, of polynomials in $k[X_1, \ldots, X_n]$. They are simply built on top of the quasiInverse(f, T), with additional splits and removal of critical pairs (Algorithm 5.7). The dominant cost is the two xgcd calls. Therefore, in each situation, the total cost is bounded by $O\big(\mathsf{M}(d_n)\log(d_n)\big)\mathsf{A}_{n-1}(T_{<n})$.

```
quasiInverse(f, T)
 1: T' ← [] ; w' ← []
 2: for S ∈ T do
 3:    W, w ← quasiInverse(f^S, S)
 4:    T' ← T' cat W ; w' ← w' cat w
 5: end for
 6: T ← removeCriticalPairs(T')
 7: w ← split(w', T', T)
 8: return T, w
```

Algo **5.7:** Quasi-inverse for a polynomial decomposed over a triangular decomposition

5.5 Coprime factorization

Other fast algorithms for this problem are given by [46], with a concern for parallel efficiency, and in [17], in a wider setting, but with a slightly worse computation time. Remark that the research announcement [16] has a time complexity that essentially matches ours.

Definition 5.11. *Let $A = a_1, \ldots, a_e$ be squarefree polynomials in $k[x]$. Some polynomials $b_1, \ldots, b_\ell$ in $k[x]$ are a* gcd-free basis *of the set A if $\gcd(b_i, b_j) = 1$ for $i \neq j$, each a_i can be written (necessarily uniquely) as a product of some of the b_j, and each b_j divides one of the a_i. The associated* coprime factorization *of A consists in the factorization of all polynomials a_i in terms of the polynomials $b_1, \ldots, b_t$.*

Proposition 5.5. *Let d be the sum of the degrees of $A = a_1, \ldots, a_e$. Then a coprime factorization of A can be computed in $O(\mathsf{M}(d)\mathrm{logp}(d)^3)$ operations in k.*

For brevity's sake, we will only prove how to compute a gcd-free basis of A, assuming without loss of generality that all a_i have positive degree. Deducing the coprime factorization of A involves some additional bookkeeping operations, keeping track of divisibility relations; it induces no new arithmetic operations, and thus has no consequence on complexity.

The algorithm relies on three subroutines multiGcd (Algo. 5.8), pairsOfGcd (Algo. 5.9) and MergeGCDFreeBases (Algo. 5.10), presented in the next paragraphs. Following the inductive scheme shown in Figure 5.1, we assume in all this section that we are given an arithmetic function $\mathsf{A}_n(T)$, as in Definition 5.8.

5.5.1 Computing multiple gcd's

The first algorithm takes as input p and $[a_1, \ldots, a_e]$ in $\mathbb{K}(T)[y]$, and outputs the sequence of all $\gcd(p, a_i, T)$, split over the same non-critical decomposition $\mathbf{U}_{e+1}$. The idea of this algorithm is to first reduce p modulo all a_i using fast simultaneous reduction, and then take the gcd's of all remainders with the polynomials a_i (see also Exercise 11.4 in [117]). We make the assumption that all a_i are non-constant in the pseudo-code below, so as to apply the results of Proposition 1.7. To cover the general case, it suffices to introduce a wrapper function, that strips the input sequence $[a_1, \ldots, a_e]$ from its constant entries, and produces 1 as corresponding gcd's; this function induces no additional arithmetic cost. Finally, we write $d = \sum_{i=1}^{e} \deg a_i$.

multiGcd$(p, [a_1, \ldots, a_e], T)$ ==

Input Polynomials $p, a_1, \ldots, a_e$, $\deg p < d$, in $\mathbb{K}(T)[y]$ for a triangular set T

Output A non-critical triangular decomposition $\mathbf{U}_{e+1}$ of T, and the sequence $\mathbf{r}_1, \ldots, \mathbf{r}_e$ where $\mathbf{r}_i$ is the projection of $\gcd(p, a_i, T)$ over the refinement $\mathbf{U}_{e+1}$ of T

1: $[p_1, \ldots, p_e] \leftarrow \text{multiRem}(p, [a_1, \ldots, a_e])$
2: $\mathbf{U}_1 \leftarrow [T]$
3: **for** $i = 1, \ldots, e$ **do**
4: $\quad \mathbf{a}_i \leftarrow \text{split}(a_i, T, \mathbf{U}_i)$ $[\deg a_i]$
5: $\quad \mathbf{p}_i \leftarrow \text{split}(p_i, T, \mathbf{U}_i)$ $[\deg p_i]$
6: $\quad \mathbf{U}_{i+1}, \mathbf{g}_i \leftarrow \gcd(\mathbf{p}_i, \mathbf{a}_i, \mathbf{U}_i)$ $[O\big(\mathsf{M}(\deg a_i)\text{logp}(\deg a_i)\big)]$
7: **end for**
8: **for** $i = 1, \ldots, e-1$ **do**
9: $\quad \mathbf{r}_i \leftarrow \text{split}(\mathbf{g}_i, \mathbf{U}_{i+1}, \mathbf{U}_{e+1})$ $[\deg \mathbf{g}_i]$
10: **end for**
11: $\mathbf{r}_e \leftarrow \mathbf{g}_e$; **return** $\mathbf{U}_{e+1}, [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_e]$

multiGcd$(\mathbf{p}, [\mathbf{b}_1, \ldots, \mathbf{b}_\ell], \mathbf{S})$

Input A triangular decomposition of $\mathbf{S}$ (of an unspecified triangular set), the family $\mathbf{p} = [p^S, S \in \mathbf{S}]$, the sequence $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$, with $\mathbf{b}_i = [b_i^S, S \in \mathbf{S}]$

Output A non-critical refinement $\mathbf{T}$ of $\mathbf{S}$, and the sequence $\mathbf{h}_1, \ldots, \mathbf{h}_\ell$ where $\mathbf{h}_i$ is the union over each $S \in \mathbf{S}$ of the projection of $\gcd(p^S, b_i^S, S)$ over the refinement of S in $\mathbf{T}$

1: $\mathbf{T}' \leftarrow [\,]$; $\mathbf{h}'_i \leftarrow [\,]$
2: **for** $S \in \mathbf{S}$ **do**
3: $\quad \mathbf{U}, [\mathbf{g}_1, \ldots, \mathbf{g}_\ell] \leftarrow \text{multiGcd}(p^S, [b_1^S, \ldots, b_\ell^S], S)$
4: $\quad \mathbf{T}' \leftarrow \mathbf{T}'$ cat $\mathbf{U}$; $\mathbf{h}'_i \leftarrow \mathbf{h}'_i$ cat $\mathbf{g}_i$, $1 \le i \le \ell$
5: **end for**
6: $\mathbf{T} \leftarrow \text{removeCriticalPairs}(\mathbf{T}')$
7: $\mathbf{h}_i \leftarrow \text{split}(\mathbf{h}'_i, \mathbf{T}', \mathbf{T})$, $1 \le i \le \ell$
8: **return** $\mathbf{T}, [\mathbf{h}_1, \ldots, \mathbf{h}_\ell]$

Algo **5.8:** Multiple GCDs Modulo a Triangular Set

Proposition 5.6. *Let f be in $\mathbb{K}(T)[y]$. If T is endowed of an arithmetic time $\mathsf{A}_n(T)$, one can compute within*

$$O\big(\,(\mathsf{M}(\deg f) + \mathsf{M}(d)\log(d))\,\big)\mathsf{A}_n(T)$$

operations in k a non-critical decomposition $\mathbf{U}$ of T, as well as $\gcd(f, a_i, T)$ split over $\mathbf{U}$, for $i = 1, \ldots, e$, where $d = \sum_{i \le r} a_i$.

Proof: The first step is to compute the subproduct tree associated with the polynomials $a_1, \ldots, a_e$, and in particular the product $a_1 \cdots a_e$. Then, we reduce f modulo $a_1 \cdots a_e$, before reducing it modulo all polynomials a_i; this yields the polynomials $f_i = (f \bmod a_i)$, $i = 1, \ldots, e$. From Proposition 1.7, points 3-4 and Lemma 1.6, the cost of these operations admits an upper bound of the form

$$O\big(\,(\mathsf{M}(\deg f) + \mathsf{M}(d)\log d)\,\mathsf{A}_n(T)\,\big).$$

If $\mathbb{K}(T)$ is a field, it then suffices to successively compute all $\gcd(f_i, a_i)$ as done in the above. In case $\mathbb{K}(T)$ is not a field, these gcd computations may induce some splittings of T, which must be taken into account. Setting initially $\mathbf{U}_1 = [T]$, we compute successive refinements $\mathbf{U}_2, \ldots, \mathbf{U}_{e+1}$ of $\mathbf{U}_1$. To this effect, at the j-th step, given $\mathbf{f}_i = \mathrm{split}(f_i, T, \mathbf{U}_i)$ and $\mathbf{a}_i = \mathrm{split}(a_i, T, \mathbf{U}_i)$ it suffices to compute

$$\mathbf{U}_{i+1}, \mathbf{h}_{i+1} = \gcd(\mathbf{f}_i, \mathbf{a}_i, \mathbf{U}_i),$$

and to compute $\mathbf{f}_{i+1} = \mathrm{split}(\mathbf{f}_i, \mathbf{U}_i, \mathbf{U}_{i+1})$ and $\mathbf{g}_{i+1} = \mathrm{split}(\mathbf{g}_i, \mathbf{U}_i, \mathbf{U}_{i+1})$ As viewed in "Specification of xgcd" page 155 the cost is

$$O\big(\,\mathsf{M}(\deg a_i)\log(\deg a_i)\,\mathsf{A}_n(T)\,\big)$$

base field operations. Let then $\mathbf{T}$ be a non-critical refinement of $\mathbf{U}_{e+1}$. After performing all these computations, the final part of the algorithm consists in splitting all $\mathbf{h}_i$ over $\mathbf{T}$; the cost for any $i \leq e$ is at most $(\deg a_i)\mathsf{A}_n(T)$. Summing over all i, and using the super-additivity of the function $d \mapsto \mathsf{M}(d)\log(d)$ finishes the proof. □

5.5.2 Computing all pairs of gcd's

The next step is to compute several pairs of gcd's. On input, we take two families of polynomials $(a_1, \ldots, a_e)$ and $(b_1, \ldots, b_s)$, where all a_i (resp. all b_i) are squarefree and pairwise coprime. The following algorithm computes all $\gcd(a_i, b_j)$. As above, we suppose that all a_i are non-constant; to cover the general case, it suffices to introduce a wrapper function, with arithmetic cost 0, that removes each constant a_i from the input, and adds the appropriate sequence $(1, \ldots, 1)$ in the output. Here, we write $d = \max(\sum_i \deg a_i, \sum_j \deg b_j)$.

The algorithm uses a divide and conquer strategy, on the subproduct tree built over the sequence $a_1, \ldots, a_e$. For convenience, we can assume that $e = 2^h$, by eventually completing the sequence $a_1, a_2, \ldots$ by polynomials equal to 1. Hence, the trees Tree, Left and Right of line 6 are complete. To extend it to polynomials defined over a triangular sets T, the multiGcd Algorithm 5.8 appears at line 2. The remaining of Algorithm 5.9 presents no difficulty. As usual, the number in brackets denotes, when multiplied by $\mathsf{A}_n(T)$, the cost at the corresponding line.

Proposition 5.7. *Let us consider Algorithm 5.9 and the notations therein. It computes within*

$$O\big(\mathsf{M}(d)\mathrm{logp}(d)^2\big)\mathsf{A}_n(T), \qquad \textit{with} \quad d = \max(\sum_i \deg a_i, \sum_j \deg b_j)$$

operations over k, a non-critical triangular decomposition $\mathbf{W}$ of $\mathbf{T}$ as well as the sequence of polynomials $\mathbf{h}_{i,j}$ indexed by $\mathbf{W}$, such that if $\mathbf{S}_{i,j}, \mathbf{g}_{i,j} = \gcd(a_i, b_j, T)$, then $\mathbf{h}_{i,j} = \mathrm{split}(\mathbf{g}_{i,j}, \mathbf{S}_{i,j}, \mathbf{T})$.

Proof: First, we compute the subproduct tree Tree associated with $a_1, \ldots, a_e$, which cost fits the required complexity. We focus next to the pairsOfGcd algorithm which takes Tree in its input. The cost at line 2 comes from Proposition 5.6.

To get an inductive relation to analyze the cost of the algorithm, we introduce the degrees $d_{\alpha,\beta} = \sum_{i=\alpha}^{\beta} \deg(a_i)$, and the complexity cost $P(d_1, \ldots, d_e)$, whence multiplied by

pairsOfGcd($[a_1, \ldots, a_e], [b_1, \ldots, b_\ell], T$)

Input A triangular set T and polynomials $a_1, \ldots, a_e$ and $b_1, \ldots, b_\ell$ over $\mathbb{K}(T)$, such that the a_i's (respectively the b_i's) are pairwise coprime

Output A non-critical decomposition $\mathbf{T}'$ of T. All the gcds $\gcd(a_i, b_j, T)$ decomposed over $\mathbb{K}(U)$, $U \in \mathbf{T}'$

1: $\mathsf{Tree} \leftarrow$ subProductTree$(a_1, \ldots, a_e)$ $[O(\mathsf{M}(d)\text{logp}(d)]$
2: **return** pairsOfGcd$(T, [T], \mathsf{Tree}, [b_1, \ldots, b_\ell])$

pairsOfGcd$(T, \mathbf{S}, \mathsf{Tree}, [\mathbf{b}_1, \ldots, \mathbf{b}_\ell])$

Input A triangular decomposition § of T, a sequence $[\mathbf{b}_1, \ldots, \mathbf{b}_\ell]$ of polynomials decomposed over $\mathbf{S}$, i.e. $\mathbf{b}_j = (b_j^S, S \in \mathbf{S})$
A complete binary tree Tree with polynomials in $\mathbb{K}(T)[y]$ at the nodes

Output A non-critical refinement $\mathbf{W}$ of $\mathbf{S}$, as well as all the union over $S \in \mathbf{S}$ of the gcds $\gcd(p_j^S, b_i^S, S)$ decomposed over the refinement of S in $\mathbf{W}$, where p_i is a polynomial at a leaf of Tree, and p_i^S its projection over $\mathbb{K}(S)$

1: $\mathbf{f} \leftarrow$ split(rootOf(Tree), $T, \mathbf{S}$) $[d]$
2: $\mathbf{U}, [\mathbf{g}_1, \ldots, \mathbf{g}_\ell] \leftarrow$ multiGcd$(\mathbf{f}, [\mathbf{b}_1, \ldots, \mathbf{b}_\ell], \mathbf{S})$ $[O(\mathsf{M}(d)\text{logp}(d))]$
3: **if** Tree has no child **then**
4: **return** $\mathbf{U}, [\mathbf{g}_1, \ldots, \mathbf{g}_\ell]$
5: **end if**
6: $\mathsf{Left} \leftarrow$ leftTree(Tree) ; $\mathsf{Right} \leftarrow$ rightTree(Tree) ; $e \leftarrow$ #leaves(Tree)
7: $\mathbf{R}, [\mathbf{h}_{ij},\ 1 \le i \le \ell,\ 1 \le j \le \lfloor e/2 \rfloor] \leftarrow$ pairsOfGcd$(T, \mathbf{U}, \mathsf{Left}, [\mathbf{g}_1, \ldots, \mathbf{g}_\ell])$ $[P(d_1, \ldots, d_{\frac{e}{2}})]$
8: $\mathbf{g}_i \leftarrow$ split$(\mathbf{g}_i, \mathbf{U}, \mathbf{R})$, $1 \le i \le \ell$ $[d]$
9: $\mathbf{W}, [\mathbf{b}_{ij},\ 1 \le i \le \ell,\ 1 \le j \le \lceil e/2 \rceil] \leftarrow$ pairsOfGcd$(T, \mathbf{U}, \mathsf{Right}, [\mathbf{g}_1, \ldots, \mathbf{g}_\ell])$ $[P(d_{\frac{e}{2}+1}, \ldots, d_e)]$
10: $\mathbf{h}_{ij} \leftarrow$ split$(\mathbf{h}_{ij}, \mathbf{U}, \mathbf{W})$, $1 \le i \le \ell$, $1 \le j \le \lfloor e/2 \rfloor$ $[d]$
11: **return** $\mathbf{W}, [\mathbf{h}_{ij},\ 1 \le i \le \ell,\ 1 \le j \le \lfloor e/2 \rfloor]$ cat $[\mathbf{b}_{ij},\ 1 \le i \le \ell,\ 1 \le j \le \lceil e/2 \rceil]$

Algo **5.9:** All Pairs of Gcds

$\mathsf{A}_n(T)$, bounds the number of arithmetic operations of the pairsOfGcd algorithm, whith input $[a_1, \ldots, a_e]$ verifying $d_i := \deg(a_i)$.

The cost of line 8 is at most $\sum_{i=1}^{\ell} \deg(\mathbf{g}_i)\big(\sum_{U \in \mathbf{U}} \mathsf{A}_n(U)\big)$. Since $\mathbf{U}$ is non-critical triangular decomposition of T, by the property (E_0) of an arithmetic time, this is lower than $\sum_{i=1}^{\ell} \deg(\mathbf{g}_i)\mathsf{A}_n(T)$. For each triangular set $U \in \mathbf{U}$, $g_i^U = \gcd(f^U, b^U)$, where f^U is in split$(\mathbf{f}, \mathbf{S}, \mathbf{U})$, d^U is in $\mathbf{g}_i$ and b^U in split$(\mathbf{b}, \mathbf{S}, \mathbf{U})$. and are indexed by the triangular set $U \in \mathbf{U}$. By hypothesis, the families of polynomials $[b_1^S, \ldots, b_\ell^S]$ are pairwise coprime, hence so are their decomposition onto the refinement $\mathbf{U}$ of $\mathbf{S}$. It follows that the gcd's computed at line 2 are also pairwise coprime: for all triangular set $U \in \mathbf{U}$, $g_i^U \in \mathbf{g}_i$ and $g_j^U \in \mathbf{g}_j$ are coprime. Hence, $\sum_{i=1}^{\ell} \deg(g_i^U) \le \deg(f^U)$, for each triangular set $U \in \mathbf{U}$, so that $\sum_{i=1}^{\ell} \deg(\mathbf{g}_i) \le \deg(\mathbf{f})$. The cost at line 8 is at most $\deg(\mathbf{f})\mathsf{A}_n(T)$.

A similar analysis shows that line 10 costs at most $d_{1,\frac{e}{2}}\mathsf{A}_n(T)$. This leads to the recursive relation:

$$P(d_1, \ldots, d_e) \le d + O\big(\mathsf{M}(d)\log(d)\big) + P(d_1, \ldots, d_{\frac{e}{2}}) + d + P(d_{\frac{e}{2}+1}, \ldots, d_e) + d_1 + \ldots + d_{\frac{e}{2}}.$$

From $P(d_1,\dots,d_e) - P(d_1,\dots,d_{\frac{e}{2}}) - P(d_{\frac{e}{2}+1},\dots,d_e) \le O\big(\mathsf{M}(d)\mathrm{logp}(d)\big)$, we deduce that:

$$\sum_{j=0}^{h-1}\sum_{i=0}^{2^j-1} P(d_{i\frac{e}{2^j}+1},\dots,d_{\frac{e}{2^j}(i+1)}) - P(d_{i\frac{e}{2^j}+1},\dots,d_{\frac{e}{2^j}(i+\frac{1}{2})}) - P(d_{\frac{e}{2^j}(i+\frac{1}{2})+1},\dots,d_{\frac{e}{2^j}(i+1)})$$
$$\le \sum_{j=0}^{h-1}\sum_{i=0}^{2^j-1} O\big(\mathsf{M}(d_{i\frac{e}{2^j}+1,\frac{e}{2^j}(i+1)})\mathrm{logp}(d_{i\frac{e}{2^j}+1,\frac{e}{2^j}(i+1)})\big)$$

By super linearity, it gives:

$$P(d_1,\dots,d_e) - \sum_{i=1}^{e} P(d_i) \le \sum_{j=1}^{h-1} O\big(\mathsf{M}(d)\mathrm{logp}(d)\big) \le O\big(\mathsf{M}(d)\mathrm{logp}(d)^2\big).$$

The complexity cost $P(d_i)$ corresponds to the case where the input sequence is reduced to $[a_i]$. The algorithm in that case stops after line 2 and the multiGcd call. Its complexity is in $O(\mathsf{M}(d_i)\mathrm{logp}(d_i))$. By super-linearity, $\sum_{i=1}^{e} P(d_i) \le O\big(\mathsf{M}(d)\mathrm{logp}(d)\big)$, which permits to conclude the proof of the complexity analysis.

5.5.3 A special case of coprime factorization

In the field case, the input of this subroutine are sequences of polynomials $[a_1,\dots,a_e]$ and $[b_1,\dots,b_\ell]$, where all a_i (resp. all b_i) are squarefree and pairwise coprime. We compute a gcd-free basis of $[a_1,\dots,a_e,b_1,\dots,b_\ell]$; this is done by computing all $\gcd(a_i,b_j)$, as well as the quotients $\delta_i = a_i/\prod_j \gcd(a_i,b_j)$ and $\gamma_j = b_j/\prod_i \gcd(a_i,b_j)$.

We denote by removeConstants(L) a subroutine that removes all constant polynomials from a sequence L (such a function requires no arithmetic operation, so its cost is zero in our model). In the complexity analysis, we still write $d = \max(\sum_i \deg a_i, \sum_j \deg b_j)$.
The validity of this algorithm is easily checked. The estimates for the cost of lines 2.2, 2.3, 3.2 and 3.3 come for the cost necessary to build a subproduct tree and perform Euclidean division, together with the fact that β_j (resp. α_i) divides b_j (resp. a_i). The total cost is thus in $O(\mathsf{M}(d)\mathrm{logp}(d)^2)$.

Proposition 5.8. *One can compute a decomposition* $\mathbf{U}$ *of* T *and a coprime factorization of* $a_1,\dots,a_e,b_1,\dots,b_\ell$ *over* $\mathbf{U}$ *in*

$$O\big(\,\mathsf{M}(d)\log^2(d)\,\big)\,\mathsf{A}(T)$$

operations in k*, where* $d = \max(\sum_{i\le e}\deg a_i, \sum_{j\le \ell}\deg b_j)$.

Proof: We first deal with a special case, when all pairs of polynomials a_i, b_j admit a monic gcd in $\mathbb{K}(T)[y]$, where the situation is similar to the field case.

Lemma 5.4. *Suppose that for all* $i \le e, j \le \ell$, a_i *and* b_j *admit a monic gcd in* $\mathbb{K}(T)[y]$. *Then given all* $g_{i,j} = \gcd(a_i,b_j,T)$ *(note that no splittings occur in this situation), one can compute a coprime factorization of* $a_1,\dots,a_e,b_1,\dots,b_e$ *over* $\mathbb{K}(T)$ *in*

$$O\big(\mathsf{M}(d)\log(d)\big)\,\mathsf{A}_n(T)$$

operations in k.

mergeGCDFreeBases($[a_1, \ldots, a_e], [b_1, \ldots, b_\ell], T$)

1: $\mathbf{W}, [\mathbf{h}_{ij} \;\; 1 \le i \le \ell \,,\; 1 \le j \le e] \leftarrow$ pairsOfGcd($[a_1, \ldots, a_e], [b_1, \ldots, b_\ell], T$)
2: $\mathbf{a}_i \leftarrow$ split$(a_i, T, \mathbf{W}),\; 1 \le i \le e$
3: $\mathbf{b}_j \leftarrow$ split$(b_j, T, \mathbf{W}),\; 1 \le j \le \ell$
4: **for** $W \in \mathbf{W}$ **do**
5: **for** $j = 1, \ldots, \ell$ **do**
6: $L_j^W \leftarrow$ removeConstants$(h_{1j}^W, \ldots, h_{ej}^W)$
7: $\beta_j^W \leftarrow \prod_{\lambda \in L_j^W} \lambda$; $\gamma_j^W \leftarrow b_j^W$ quo β_j^W
8: **end for**
9: **for** $i = 1, \ldots, e$ **do**
10: $L_i^W \leftarrow$ removeConstants$(h_{i1}^W, \ldots, h_{i\ell}^W)$
11: $\alpha_i^W \leftarrow \prod_{\lambda \in L_i^W} \lambda$; $\delta_i^W \leftarrow a_i^W$ quo α_i^W
12: **end for**
13: $A^W \leftarrow$ removeConstants$(h_{11}^W, \ldots, h_{es}^W, \gamma_1^W, \ldots, \gamma_\ell^W, \ldots, \delta_1^W, \ldots, \delta_e^W)$
14: **end for**
15: **return** $\mathbf{W}, [A^W \,,\; W \in \mathbf{W}]$

mergeGCDFreeBases($[\mathbf{a}_1, \ldots, \mathbf{a}_e], [\mathbf{b}_1, \ldots, \mathbf{b}_\ell], \mathbf{T}$)

1: $\mathbf{T}' \leftarrow [\,]$; $\mathbf{h} \leftarrow [\,]$
2: **for** $S \in \mathbf{T}$ **do**
3: $\mathbf{W}, \mathbf{u} \leftarrow$ mergeGCDFreeBases$([a_1^S, \ldots, a_e^S], [b_1^S, \ldots, b_\ell^S], S)$
4: $\mathbf{T}' \leftarrow \mathbf{T}'$ cat $\mathbf{W}$; $\mathbf{h} \leftarrow \mathbf{h}$ cat $\mathbf{u}$
5: **end for**
6: $\mathbf{T} \leftarrow$ removeCriticalPairs$(\mathbf{T}')$
7: $\mathbf{h} \leftarrow$ split$(\mathbf{h}, \mathbf{T}', \mathbf{T})$
8: **return** $\mathbf{T}, \mathbf{h}$

Algo **5.10:** Merge the GCDs of two gcd-free bases

Proof: For all $i \le e, j \le \ell$, we compute the products

$$\alpha_j = \prod_{i \le e} g_{i,j} \quad \text{and} \quad \beta_i = \prod_{j \le \ell} g_{i,j}.$$

Recall that $g_{i,j} = \gcd(a_i, b_j, T)$. Since the polynomials b_j are pairwise coprime, for $i \le \ell$, β_i divides a_i; let then $\delta_i \in \mathbb{K}(T)[y]$ be the quotient. Similarly, since the polynomials a_i are pairwise coprime, we define for $j \le \ell$ γ_j as the quotient of b_j by α_j. Let $a_1, \ldots, a_t$ be the all non-constant polynomials in the family of $\{g_{i,j}, \delta_i, \gamma_j\}$. It is clear by construction that all polynomials a_i and b_j can be written as products of the polynomials $a_1, \ldots, a_t$, and that any polynomial in $a_1, \ldots, a_t$ divides either one of the polynomials a_i or b_j. We finally prove coprimality by case inspection.

- *$g_{i,j}$ and $g_{i',j'}$ are coprime*: either $i \ne i'$, in which case this follows from a_i and $a_{i'}$ being coprime, or $j \ne j'$, and a symmetric argument applies.
- *$g_{i,j}$ and δ_i are coprime*: if not, β_i and δ_i would have a common factor, so a_i wouldn't be squarefree. The same holds for $g_{i,j}$ and γ_j.
- *$g_{i,j}$ and $\delta_{i'}$ are coprime, for $i \ne i'$*: else, a_i and $a_{i'}$ would have a common factor. The same holds for $g_{i,j}$ and $\gamma_{j'}$, with $j \ne j'$.

- *δ_i and γ_j are coprime*: this is because δ_i divides m_i and γ_j divides b_j.

Let us give cost estimates. For all i, j, the polynomials β_i and α_j can be computed in respectively

$$O\big(\,\mathsf{M}(\deg a_i)\log(\deg a_i)\,\big)\,\mathsf{A}_n(T) \quad \text{and} \quad O\big(\,\mathsf{M}(\deg b_j)\log(\deg b_j)\,\big)\,\mathsf{A}_n(T)$$

operations in k, using the subproduct trees associated to $[a_1, \ldots, a_e]$ and $[b_1, \ldots, b_\ell]$, and complexity estimate of Proposition 1.7, 3. Using fast Euclidean division, the polynomials δ_i and γ_j can respectively be deduced in

$$O\big(\,\mathsf{M}(\deg a_i)\,\big)\,\mathsf{A}_n(T) \quad \text{and} \quad O\big(\,\mathsf{M}(\deg b_j)\,\big)\,\mathsf{A}_n(T)$$

operations in k (Prop. 1.7, 1.). Summing over all i, j gives the required upper bound. □

To treat the general case, we follow the lines of Algorithm 5.10. The call to pairsOfGcd algorithm at line 1 produces a non-critical decomposition $\mathbf{W}$ of T. Hence the for loop at line 4, on the triangular sets W of $\mathbf{W}$ can be handled by the super-linearity of the arithmetic times $\mathsf{A}_n(W)$ and of the costs functions involved by the instructions. This permits to reduce the complexity analysis to the field case, discussed in Lemma 5.4, where no such loop occurred. Hence, the only additional cost caused by the decomposition $\mathbf{W}$ is the splitting instructions at lines 2 and 3. Both requires at most $2d\mathsf{A}_n(T)$ operations over k, fitting the bound stated in Proposition 5.8.

Specification of merging the GCD's The main algorithm of coprime factorization requires a generalization of the algorithm merging two coprime factorizations of two families of polynomials. It is described in the utter half part of Algorithm 5.10. The input is now a non-critical decomposition $\mathbf{T}$ of a triangular set T and two families of lists of polynomials of cardinality e and s respectively, and indexed by $\mathbf{T}$: $[a_1^S, \ldots, a_e^S]\,,\ S \in \mathbf{T}$ and $[b_1^S, \ldots, b_\ell^S]\ S \in \mathbf{T}$. The complexity costs written at the end of each line are easy to prove regarding the super-linearity of all the costs functions involved.

5.5.4 Conclusion: Proof of the main result

We finally give an algorithm for gcd-free basis. As input, we take squarefree, non-constant polynomials $a_1, \ldots, a_e$, with $d = \sum_{i \le e} \deg a_i$. We need a construction close to the subproduct tree: we form a binary tree Sub' whose nodes will be labeled by sequences of polynomials. Initially the leaves contain the sequences of length 1 $(a_1), \ldots, (a_e)$, and all other nodes are empty. Then, we go up the tree; at a node N, we use the subroutine above to compute a gcd-free basis of the sequences labeling the children of N.

Notations Let $MFB(d_a, d_b)\mathsf{A}_n(T)$ be an upper bound on the number of arithmetic operations necessary to compute mergeGCDFreeBasis$([a_1, \ldots, a_e], [b_1, \ldots, b_\ell], T)$, where $d_a := \sum_{i \le e} \deg(a_i)$ and $d_b := \sum_{j \le \ell} \deg(b_j)$. Let $F(d_1, \ldots, d_e)\mathsf{A}_n(T)$ the same for the gcdFeeBasis algorithm, with inputs $[c_1, \ldots, c_e]$, list of polynomials in $\mathbb{K}T[y]$ with $d_i := \deg(c_i)$. First let us deduce an upper bound for the cost of the specification algorithm written in the utter half part of Algorithm 5.11. There, we denote by each $d_i := \max_{U \in \mathbf{T}} \deg(a_i^U)$, where

gcdFreeBasis($[a_1, \ldots, a_e], T$)
1: **if** $e = 1$ **then return** $[T], [a_e]$; **end if**
2: $\ell \leftarrow e/2$
3: $\mathbf{U}, \mathbf{g} \leftarrow$ gcdFreeBasis($[a_1, \ldots, a_\ell], T$)
4: $\mathbf{a}_i \leftarrow \text{split}(a_i, T, \mathbf{U})$, $\ell < i \leq e$
5: $\mathbf{W}, \mathbf{h} \leftarrow$ gcdFreeBasis($[\mathbf{a}_{\ell+1}, \ldots, \mathbf{a}_e], \mathbf{U}$)
6: $\mathbf{u} \leftarrow \text{split}(\mathbf{g}, \mathbf{U}, \mathbf{W})$
7: $\mathbf{u} \leftarrow$ removeConstants($\mathbf{u}$)
8: **return** mergeGCDFreeBases($\mathbf{u}, \mathbf{h}, \mathbf{W}$)

gcdFreeBasis($[\mathbf{a}_1, \ldots, \mathbf{a}_e], \mathbf{T}$)
1: $\mathbf{T}' \leftarrow [\,]$; **res** $\leftarrow [\,]$
2: **for** $U \in \mathbf{T}$ **do**
3: $\quad \mathbf{W}, \mathbf{h} \leftarrow$ gcdFreeBasis($[a_1^U, \ldots, a_e^U], U$)
4: $\quad \mathbf{T}' \leftarrow \mathbf{T}'$ cat $\mathbf{W}$; **res** $\leftarrow$ **res** cat $[\mathbf{h}]$
5: **end for**
6: $\mathbf{T} \leftarrow$ removeCriticalPairs($\mathbf{T}'$)
7: **res** $\leftarrow \text{split}(\mathbf{res}, \mathbf{T}', \mathbf{T})$
8: **res** $\leftarrow$ removeConstants(**res**)
9: **return** $\mathbf{T}$, **res**

Algo **5.11:** Gcd-free Basis Modulo a Triangular Set

$\mathbf{a}_i := [a_i^U,\ U \in \mathbf{T}]$ by definition (so that, for all $U \in \mathbf{T}$, $\sum_{i \leq e} \deg(a_i^U) \leq \sum_{i \leq e} \deg(a_i)$). Then, the for loop of Step 2 requires at most:

$$\sum_{U \in \mathbf{T}} F(d_1, \ldots, d_e) \mathsf{A}_n(U)$$

arithmetic operations over k. The splitting operation at line 7 holds on a family of lists of polynomials **res** indexed by the triangular sets in $\mathbf{T}$. For each of these lists, the sum of the degrees of the polynomials inside is bounded by $\sum_{i \leq e} \deg(a_i^U)$. Hence Line 7, requires at most $d\mathsf{A}_n(T)$ Finally, the utter half algorithm in Algo 5.11 has a cost fitting in:

$$\big(F(d_1, \ldots, d_e) + O(1) + d\big)\mathsf{A}_n(T),$$

operations over k, if T is such that $\mathbf{T}$ is a refinement of T.

This recursive identity follows:

$$F(d_1, \ldots, d_e) \leq F(d_1, \ldots, d_{\frac{e}{2}}) + d_{\frac{e}{2}+1,e} + F(d_{\frac{e}{2}+1}, \ldots, d_e) + d_{\frac{e}{2}+1,e} + O(1) + d_{1,\frac{e}{2}} + MFB(d_{1,\frac{e}{2}}, d_{\frac{e}{2}+1,e})$$

$$\sum_{j=0}^{h-1} \sum_{i=0}^{2^j-1} F(d_{i\frac{e}{2^j}+1}, \ldots, d_{\frac{e}{2^j}(i+1)}) - F(d_{i\frac{e}{2^j}+1}, \ldots, d_{\frac{e}{2^j}(i+\frac{1}{2})}) - F(d_{\frac{e}{2^j}(i+\frac{1}{2})+1}, \ldots, d_{\frac{e}{2^j}(i+1)}) \leq \sum_{j=0}^{h-1} \sum_{i=0}^{2^j-1} d_{i\frac{e}{2^j}+1,\frac{e}{2^j}(i+1)} + d_{\frac{e}{2^j}(1+\frac{1}{2})+1,\frac{e}{2^j}(i+1)} + O(1) + MFB(d_{i\frac{e}{2^j}+1,\frac{e}{2^j}(i+\frac{1}{2})}, d_{\frac{e}{2^j}(i+\frac{1}{2})+1,\frac{e}{2^j}(i+1)})$$

By super-linearity:

$$F(d_1,\ldots,d_e)-\sum_{i=1}^{e}G(d_i)\leq\sum_{j=0}^{h-1}d_{1,e}+d_{\frac{e}{2}+1,e}+MFB(d_{1,\frac{e}{2}},d_{\frac{e}{2}+1,e})$$

But $F(d_i)=0$, $h\leq\operatorname{logp}(d)$, $d_{\frac{e}{2}+1,e}\leq d$ and $MFB(x,y)\in O\big(M(\max(x,y)\operatorname{logp}(\max(x,y))^2\big)$:

$$F(d_1,\ldots,d_e)\leq 2d\operatorname{logp}(d)+O\big(M(d)\operatorname{logp}(d)^3\big).$$

5.6 Removing critical pairs

We next show how to remove critical pairs. This is an inductive process, whose complexity is estimated in the following proposition and its corollary. We need to extend the notion of "refining" introduced previously. Extending Definition 5.2, we say that a family of triangular sets $\mathbf{T}'$ refines another family $\mathbf{T}$ if for every $T\in\mathbf{T}$, there exists a subset of $\mathbf{T}'$ that forms a triangular decomposition of $\langle T\rangle$.

Note the difference with the initial definition: we do not impose that the family $\mathbf{T}$ forms a triangular decomposition of some ideal I. In particular, the triangular sets in $\mathbf{T}$ do not have to generate coprime ideals.

Proposition 5.9. *There exists a constant K such that the following holds. Let $\mathsf{A}_1(),\ldots,\mathsf{A}_{n-1}()$ be arithmetic times for triangular sets in $1,\ldots,n-1$ variables.*

Let T be a triangular set in n variables, and let $\mathbf{U}$ be a triangular decomposition of $\langle T\rangle$. Then for all $j=1,\ldots,n$, the following holds: given $\mathbf{U}_{\leq j}$, one can compute a non-critical triangular decomposition $\mathbf{W}$ of $T_{\leq j}$ that refines $\mathbf{U}_{\leq j}$ using a_j operations in k, where a_j satisfies the recurrence inequalities $a_0=0$ and for $j=0,\ldots,n-1$,

$$a_{j+1}\leq 2a_j+K\mathsf{M}(d_{j+1}\cdots d_n)\operatorname{logp}(d_{j+1}\cdots d_n)^3\mathsf{A}_j(T_{\leq j}),$$

and where $d_j=\deg_j T$ for $j=1,\ldots,n$.

Before discussing the proof of this assertion, let us give an immediate corollary, which follows by a direct induction.

Corollary 5.1. *Given a triangular decomposition $\mathbf{U}$ of $\langle T\rangle$, one can compute a non-critical triangular decomposition $\mathbf{W}$ of $\langle T\rangle$ that refines $\mathbf{U}$ in time*

$$K\left(2^{n-1}\mathsf{M}(d_1\cdots d_n)\operatorname{logp}(d_1\cdots d_n)^3+\cdots+\mathsf{M}(d_n)\operatorname{logp}(d_n)^3\mathsf{A}_{n-1}(T_{\leq n-1})\right).$$

Proof: We only sketch the proof of the proposition. Let thus j be in $0,\ldots,n-1$ and let $\mathbf{U}=U^1,\ldots,U^e$ be a triangular decomposition of $\langle T\rangle$; we aim at removing the critical pairs in $\mathbf{U}_{\leq j+1}$. Let $\mathbf{V}$ be obtained by removing the critical pairs in $\mathbf{U}_{\leq j}$. Thus, $\mathbf{V}$ consists in triangular sets in $k[X_1,\ldots,X_j]$, and has no critical pair.

Let us fix $i\leq e$, and write $U^i=(U^i_1,\ldots,U^i_n)$. By definition, there exists a subset $\mathbf{V}_i=V^{i,1},\ldots,V^{i,e_i}$ of $\mathbf{V}$ which forms a non-critical decomposition of $(U^i_1,\ldots,U^i_j)$. Our next step is to compute

$$U^{i,1}_{j+1},\ldots,U^{i,e_i}_{j+1}=\operatorname{split}(U^i_{j+1},(U^i_1,\ldots,U^i_j),\mathbf{V}_i).$$

Consider now a triangular set V in $\mathbf{V}$. There may be several subsets $\mathbf{V}_i$ such that $V \in \mathbf{V}_i$. Let $S_V \subset \{1, \ldots, e\}$ be the set of corresponding indices; thus, for any $i \in S_V$, there exists $\ell(i)$ in $1, \ldots, e_i$ such that $V = V^{i,e_{\ell(i)}}$. We will then compute a coprime factorization of all polynomials $U_{j+1}^{i,e_{\ell(i)}}$ in $\mathbb{K}(V)[X_{j+1}]$, for $i \in S_V$, and for all V.

This process will refine the family $\mathbf{V}$, creating possibly new critical pairs: we get rid of these critical pairs, obtaining a decomposition $\mathbf{W}$. It finally suffices to split all polynomials in the coprime factorization obtained before from $\mathbf{V}$ to $\mathbf{W}$ to conclude. The cost estimates then takes into account the cost for the two calls to the same process in j variables, hence the term $2a_j$, and the cost for coprime factorization and splitting. Studying the degrees of the polynomials involved, this cost can be bounded by

$$K\mathsf{M}(d_{j+1} \cdots d_n)\mathrm{logp}(d_{j+1} \cdots d_n)^3 \mathsf{A}_j(T_{\leq j})$$

for some constant K, according to the results in the last section. □

5.7 Concluding the proof

All ingredients are now present to give the proof of the following result, which readily implies the main theorems stated in the introduction.

Theorem 5.3. *There exists a constant* C_1 *such that, writing*

$$\mathsf{A}_n(d_1, \ldots, d_n) = \mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}(d_i)^3,$$

the function $T \mapsto \mathsf{A}_n(\deg_1 T, \ldots, \deg_n T)$ *is an arithmetic time for triangular sets in* n *variables, for all* n.

Proof: The proof requires to check that taking C_1 big enough, all conditions defining arithmetic times are satisfied. We do it by induction on n; the case $n = 1$ is settled by Proposition 5.1, taking C_1 larger than the constant C in that proposition, and using the fact that $\mathrm{logp}(x) \geq 1$ for all x.

Let us now consider index n; we can thus assume that the function A_j is an arithmetic time for triangular sets in j variables, for $j = 1, \ldots, n-1$. Then, at index n, condition (E_0) makes no difficulty, using the super-additivity of the function M. Addition and multiplication (condition (E_1)) and splitting (condition (E_4)) follow from Proposition 5.2, again as soon as the condition $\mathsf{C}_1 \geq \mathsf{C}$ holds. The computation of quasi-inverses (condition (E_2)) is taken care of by Proposition 5.4, using our induction assumption on arithmetic times A, as soon as C_1 is large enough to compensate the constant factor hidden in the $O(\ )$ estimate of that proposition.

The cost for removing critical pairs is given in the previous section. In view of Corollary 5.1, and using the condition $\mathsf{M}(dd') \leq \mathsf{M}(d)\mathsf{M}(d')$, after a few simplifications, to satisfy condition (E_3), C_1 must satisfy the inequality

$$K(2^{n-1} + 2^{n-2}\mathsf{C}_1 + \cdots + \mathsf{C}_1^{n-1}) \leq \mathsf{C}_1^n,$$

where K is the constant introduced in Corollary 5.1. This is the case for $\mathsf{C}_1 \geq K + 2$. □

Appendix: merging triangular sets for inversion

This subsection is devoted to prove a complexity result for the Chinese Remaindering theorem over a triangular set.

In all that follows, referring to a triangular set $\mathbf{T} = (T_1, \ldots, T_n)$, d_i denotes the degree of the polynomial T_i in its main variable X_i. Then, from the previous sections of this chapter, there exists a constant C_1 such that the following holds for any triangular set $\mathbf{T}$:

$\mathsf{D5_1}$ One can do all operations $(+, \times)$ modulo $\mathbf{T}$ in time $\mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)$.

$\mathsf{D5_2}$ If $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ is a non-critical decomposition of $\mathbf{T}$, then the reduction map

$$\mathbb{K}[\mathbf{X}]/\mathbf{T} \to \prod_{\mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}} \mathbb{K}[\mathbf{X}]/\mathsf{U}$$

can be computed in time $\mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}(d_i)$.

$\mathsf{D5_3}$ Let $A \in \mathbb{K}[\mathbf{X}]$ be reduced modulo $\mathbf{T}$. Then one can test if A is a unit modulo $\mathbf{T}$ in time

$$\mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}^3(d_i).$$

If so, one can compute a non-critical decomposition $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ of $\mathbf{T}$, as well as a set of polynomials

$$\{B_{\mathsf{U}} \in \mathbb{K}[\mathbf{X}] \mid \mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\},$$

with B_{U} reduced modulo U and such that $B_{\mathsf{U}} = A^{-1} \bmod \mathsf{U}$, in the same time.

$\mathsf{D5_4}$ Let Q be the quotient $\mathbb{K}[\mathbf{X}]/\langle \mathbf{T} \rangle$. If A, B are polynomials of degrees at most d in $Q[Y]$, with B monic, such that $\langle A, B \rangle = 1$, then one can compute a non-critical decomposition $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ of $\mathbf{T}$, as well as as a set of polynomials

$$\{C_{\mathsf{U}} \in \mathbb{K}[\mathbf{X}][Y] \mid \mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\},$$

with C_{U} reduced modulo U and such that $AC_{\mathsf{U}} = 1 \bmod (\mathsf{U}, B)$, in time

$$\mathsf{C}_1^{n+1} \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}^3(d_i)\ \mathsf{M}(d)\mathrm{logp}(d).$$

All that is missing to prove our main assertion is inversion: even if A is a unit modulo $\langle \mathbf{T} \rangle$, computing its inverse will induce a decomposition of $\mathbf{T}$.

To fill this gap, we will give an algorithm for recombination, based on Chinese remaindering. Recall thus (see for instance [15, Section 23]) that there exists a constant C_2 with the following property.

$\mathsf{CRT_1}$ Let $\mathbb{A}$ be a ring, let $A_1, \ldots, A_L$ be monic squarefree polynomials in $\mathbb{A}[Y]$, such that $\langle A_i, A_j \rangle = 1$ for all $i < j \leq L$. Let $A = A_1 \cdots A_L$, and suppose that $(A')^{-1}$ modulo A is known. Let finally $d = \sum_{\ell \leq L} \deg(A_\ell)$.

Given $B_1, \ldots, B_L$ in $\mathbb{A}[Y]$, with $\deg B_\ell < \deg A_\ell$ for all ℓ, one can compute the unique $B \in \mathbb{A}[Y]$ of degree less than d such that $B = B_\ell \bmod A_\ell$ holds for all ℓ, in time $\mathsf{C}_2\mathsf{M}(d)\mathrm{logp}(d)$.

We now present an algorithm for inversion modulo a Lazard triangular set $\mathbf{T}$, assuming that $\mathbf{T}$ generates a radical ideal: To invert A modulo $\langle \mathbf{T} \rangle$, we will first apply point $\mathsf{D5_3}$ above, inducing a splitting of $\mathbf{T}$. We will then use recursively the previous result $\mathsf{CRT_1}$ to recombine the results. Without loss of generality, in what follows, we assume that $\mathsf{C}_1 = \mathsf{C}_2$.

Step 1: One level of Chinese remaindering modulo a triangular set. We start by a simple version of Chinese remaindering, where the triangular set $\mathbf{T}$ has been split only once. Let thus $\mathbf{T} = (T_1, \ldots, T_n)$ be a Lazard triangular set in $\mathbb{K}[X_1, \ldots, X_n]$ that generates a radical ideal. Let then i be an index $\leq n$, and let $T_i^{(1)}, \ldots, T_i^{(L)}$ in $\mathbb{K}[X_1, \ldots, X_i]$ be such that $T_i = T_i^{(1)} \cdots T_i^{(L)}$ holds modulo $\langle T_1, \ldots, T_{i-1} \rangle$. Then, since $\mathbf{T}$ generates a radical ideal, the family of Lazard triangular sets

$$\begin{aligned} \mathbf{U}^{(1)} &= \big(T_1, \ldots, T_{i-1}, T_i^{(1)}, T_{i+1} \bmod \langle T_1, \ldots, T_i^{(1)} \rangle, \ldots, T_n \bmod \langle T_1, \ldots, T_i^{(1)} \rangle\big) \\ &\vdots \\ \mathbf{U}^{(L)} &= \big(T_1, \ldots, T_{i-1}, T_i^{(L)}, T_{i+1} \bmod \langle T_1, \ldots, T_i^{(L)} \rangle, \ldots, T_n \bmod \langle T_1, \ldots, T_i^{(L)} \rangle\big) \end{aligned}$$

is a non-critical decomposition of $\mathbf{T}$.

Lemma 5.5. *Suppose that $(T_i')^{-1} \bmod \langle T_1, \ldots, T_i \rangle$ is known. Given $B_1, \ldots, B_L$ in $\mathbb{K}[X_1, \ldots, X_n]$ with B_ℓ reduced modulo $\mathbf{U}^{(\ell)}$ for all ℓ, one can compute the unique $B \in \mathbb{K}[X_1, \ldots, X_n]$ reduced modulo $\mathbf{T}$ and such that $B = B_\ell \bmod \mathsf{U}^{(\ell)}$ holds for all ℓ in*

$$\mathsf{C}_1^\ell \mathsf{M}(d_1) \cdots \mathsf{M}(d_{i-1}) \mathsf{M}(d_i) \mathrm{logp}(d_i) d_{i+1} \cdots d_n$$

operations in $\mathbb{K}$.

Proof: We apply point $\mathsf{CRT_1}$ to all coefficients of the polynomials B_ℓ, seen in $Q[X_i][X_{i+1}, \ldots, X_n]$, with $Q = \mathbb{K}[X_1, \ldots, X_{i-1}]/\langle T_1, \ldots, T_{i-1} \rangle$. □

Step 2: More complex Chinese remaindering. We continue with a slightly more complex version of the question, where we perform several instances of Chinese remaindering at the various branches of a triangular decomposition, but always at the same level.

Let thus $\mathbf{T} = (T_1, \ldots, T_n)$ be a Lazard triangular set in $\mathbb{K}[X_1, \ldots, X_n]$ that generates a radical ideal. Let i be an index $\leq n$ and let $\mathsf{U}^{(1)}, \ldots, \mathsf{U}^{(L)}$ be a non-critical triangular decomposition of $(T_1, \ldots, T_i)$ in $\mathbb{K}[X_1, \ldots, X_i]$, with $\mathsf{U}^{(\ell)} = (U_1^{(\ell)}, \ldots, U_i^{(\ell)})$. Associated with this decomposition of $(T_1, \ldots, T_i)$, we have the corresponding non-critical decomposition of $\mathbf{T}$ itself as

$$\begin{aligned} \mathbf{A}^{(1)} &= \big(U_1^{(1)}, \ldots, U_i^{(1)}, T_{i+1} \bmod \mathsf{U}^{(1)}, \ldots, T_n \bmod \mathsf{U}^{(1)}\big) \\ &\vdots \\ \mathbf{A}^{(L)} &= \big(U_1^{(L)}, \ldots, U_i^{(L)}, T_{i+1} \bmod \mathsf{U}^{(L)}, \ldots, T_n \bmod \mathsf{U}^{(L)}\big). \end{aligned} \tag{5.6}$$

We will also be interested in another non-critical decomposition of $\mathbf{T}$, defined by regrouping some of the $\mathbf{A}^{(L)}$ together at level i. For $\ell \leq L$, let thus $\mathbf{V}^{(\ell)}$ be defined by $\mathbf{V}^{(\ell)} = (U_1^{(\ell)}, \ldots, U_{i-1}^{(\ell)})$, so that $\mathbf{V}^{(\ell)}$ is a triangular set in $\mathbb{K}[X_1, \ldots, X_{i-1}]$. Up to renumbering, we may assume that there exists integers

$$M_1 = 1 < \cdots < M_S < M_{S+1} = L + 1$$

such that the equalities

$$\begin{array}{ccccc} \mathbf{V}^{(M_1)} & = & \cdots & = & \mathbf{V}^{(M_2-1)} \\ & & \vdots & & \\ \mathbf{V}^{(M_s)} & = & \cdots & = & \mathbf{V}^{(M_{S+1}-1)} \end{array}$$

hold, with furthermore $\mathbf{V}^{(M_i)}$ and $\mathbf{V}^{(M_j)}$ pairwise distinct for $i \neq j$. Then, $\mathbf{V}^{(M_1)}, \ldots, \mathbf{V}^{(M_S)}$ form a non-critical triangular decomposition of $(T_1, \ldots, T_{i-1})$, so that

$$\begin{array}{c} \mathbf{B}^{(1)} = \left(\mathbf{V}^{(M_1)}, T_i \bmod \mathbf{V}^{(M_1)}, \ldots, T_n \bmod \mathbf{V}^{(M_1)}\right) \\ \vdots \\ \mathbf{B}^{(S)} = \left(\mathbf{V}^{(M_S)}, T_i \bmod \mathbf{V}^{(M_S)}, \ldots, T_n \bmod \mathbf{V}^{(M_S)}\right) \end{array} \tag{5.7}$$

is a non-critical decomposition of $\mathbf{T}$ that refines the decomposition (5.6). Indeed, for $s \leq S$, $\mathbf{A}^{(M_s)}, \ldots, \mathbf{A}^{(M_{s+1}-1)}$ is a non-critical decomposition of $\mathbf{B}^{(s)}$.

Let $B_1, \ldots, B_L$ be in $\mathbb{K}[X_1, \ldots, X_n]$, with B_ℓ reduced modulo $\mathbf{A}^{(\ell)}$ for all ℓ. In view of the previous point, there exist unique $C_1, \ldots, C_S$ in $\mathbb{K}[X_1, \ldots, X_n]$, with C_s reduced modulo $\mathbf{B}^{(s)}$, such that $B_\ell = C_s \bmod \mathbf{A}^{(\ell)}$, for $M_s \leq \ell < M_{s+1}$.

Lemma 5.6. *Assume that the inverse K_i of T_i' modulo $\langle T_1, \ldots, T_i \rangle$ is known. The polynomials $C_1, \ldots, C_S$ can be computed in time*

$$2\mathsf{C}_1^i \mathsf{M}(d_1)\mathrm{logp}(d_1) \cdots \mathsf{M}(d_i)\mathrm{logp}(d_i) d_{i+1} \cdots d_n.$$

Proof: We first reduce K_i modulo $\mathbf{V}^{(M_1)}, \ldots, \mathbf{V}^{(M_S)}$. This is done coefficient by coefficient; using point $\mathsf{D5_2}$, this can be done in time

$$\mathsf{C}_1^{i-1} \mathsf{M}(d_1)\mathrm{logp}(d_1) \cdots \mathsf{M}(d_{i-1})\mathrm{logp}(d_{i-1}) d_i.$$

Then, Lemma 5.5 shows that the cost of computing C_s is

$$\mathsf{C}_1^i \mathsf{M}(d_{1,s}) \cdots \mathsf{M}(d_{i-1,s}) \mathsf{M}(d_i)\mathrm{logp}(d_i) d_{i+1} \cdots d_n,$$

where $d_{j,s}$ is the X_j-degree of $U_j^{(M_s)}$. Summing over all s gives the requested upper bound, since the super-additivity of M implies that

$$\sum_{s \leq S} \mathsf{M}(d_{1,s}) \cdots \mathsf{M}(d_{i-1,s}) \leq \mathsf{M}(d_1) \cdots \mathsf{M}(d_{i-1})$$

holds. □

Conclusion. We prove our main result; we start by giving the cost for Chinese remaindering, assuming that some inverses are known.

Proposition 5.10. *Let $\mathbf{T} = (T_1, \ldots, T_n)$ be a Lazard triangular set in $\mathbb{K}[\mathbf{X}]$ that generates a radical ideal, and suppose that for $j = 1, \ldots, n$, the inverse K_j of T_j' modulo $\langle T_1, \ldots, T_j \rangle$ is known. Let $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ be a non-critical triangular decomposition of $\mathbf{T}$, and consider a family of polynomials $\{B_\mathsf{U} \mid \mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\}$, where B_U is reduced modulo U.*

Then one can compute the unique polynomial B reduced modulo $\mathbf{T}$ such that $B = B_\mathsf{U} \bmod \mathsf{U}$ holds for all U in time

$$2n\mathsf{C}_1^n \mathsf{M}(d_1)\mathrm{logp}(d_1) \cdots \mathsf{M}(d_n)\mathrm{logp}(d_n).$$

PROOF: It suffices to apply Lemma 5.6 for $i = n, \ldots, 1$. □

We continue by working out the complexity of computing the required inverses.

Proposition 5.11. *Let assumptions be as in the previous proposition, and let K_i be the inverse of T_i' modulo $\langle T_1, \ldots, T_i \rangle$. Then $K_1, \ldots, K_n$ can be computed in time*

$$(3n^2 + n)\mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}^3(d_i).$$

PROOF: Supposing that $K_1, \ldots, K_{i-1}$ are known, we work out the complexity of computing K_i. Applying point $\mathsf{D5_4}$ to T_i and T_i', we can compute a non-critical decomposition $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ of $(T_1, \ldots, T_{i-1})$ as well as $\{K_i \bmod \mathsf{U} \mid \mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\}$, in time

$$\mathsf{C}_1^i \prod_{j \leq i-1} \mathsf{M}(d_j)\mathrm{logp}^3(d_j)\mathsf{M}(d_i)\mathrm{logp}(d_i).$$

Then, it suffices to apply Proposition 5.10 to recover K_i, in time

$$2i\mathsf{C}_1^i \mathsf{M}(d_1)\mathrm{logp}(d_1) \cdots \mathsf{M}(d_i)\mathrm{logp}(d_i).$$

Summing over all i gives the result. □

We can then conclude the proof of our main assertion. All notation being as above, let A be a unit modulo $\mathbf{T}$, and let $B = A^{-1}$. We first precompute the needed inverses $K_1, \ldots, K_n$ using the previous proposition. Applying point $\mathsf{D5_3}$, we next compute a non-critical decomposition $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}$ of $\mathbf{T}$ as well as $\{B \bmod \mathsf{U} \mid \mathsf{U} \in \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\}$, in time

$$\mathsf{C}_1^n \prod_{j \leq n} \mathsf{M}(d_j)\mathrm{logp}^3(d_j).$$

Since the required inverses are known, applying Proposition 5.10, we can recover B. Putting all costs together yields a complexity for computing A^{-1} of

$$(3n^2 + 3n + 1)\mathsf{C}_1^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}^3(d_i),$$

which is bounded by

$$\mathsf{C}^n \prod_{i \leq n} \mathsf{M}(d_i)\mathrm{logp}^3(d_i)$$

for C large enough.

Conclusion

Les contributions de cette thèse permettent de placer la résolution des systèmes polynomiaux par triangulation à un niveau plus compétitif et novateur : de part la possibilité d'étendre le champs d'application des méthodes modulaires, et par l'ébauche d'une étude de complexité sérieuse des calculs menés selon le principe D5. Les implantations des différents algorithmes, sauf ceux relevant du principe D5, dans le logiciel MAPLE ont montré l'efficacité de ces résultats, surtout quand les données initiales requièrent beaucoup de mémoire. Par ailleurs, ces contributions ont aussi donné lieu à de nouveaux problèmes.

Bornes sur les coefficients

Nous avons prouvé des bornes sur les degrés en les Y d'ensembles triangulaires de Lazard $T \subset k(Y_1, \ldots, Y_m)[X_1, \ldots, X_n]$. Qu'en est-il de la taille des coefficients rationnels dans le cas où $k = \mathbb{Q}$? Un résultat dans cette direction permettrait d'améliorer la borne de probabilité du théorème 3.1 ; seul le cardinal de l'ensemble témoin S est pris en compte, aussi grandes ses valeurs soient elles, ceci ne joue pas sur cette probabilité. Pour cela, il faut disposer de ces bornes sur les ensembles triangulaires.

Maintenant que des résultats plus fins existent pour ces bornes, vient la question de leur optimalité. Il s'agit d'établir un exemple où les quantités grandissent dans le même ordre que le disent les bornes.

Décomposition équiprojetable

Si l'on enlève l'hypothèse d'engendrer un idéal radical, mais de rester des bases de Gröbner lexicographiques réduites, zéro-dimensionnelles, avec coefficients dominants égaux à 1, les multiplicités que peuvent représenter les ensembles triangulaires de Lazard sont simples : l'idéal monomial de l'escalier en un point multiple est de la forme $\langle x_1^{r_1}, \ldots, x_n^{r_n} \rangle$.

Pour pouvoir représenter un zéro avec une multiplicité plus complexe, par exemple $\langle x^2, xy, y^2 \rangle$ il faut deux ensembles triangulaires U et V :

$$\left| \begin{array}{l} U_2(x,y) = y^2 \\ U_1(x) = x \end{array} \right. \qquad \left| \begin{array}{l} V_2(x,y) = y \\ V_1(x) = x^2 \end{array} \right.$$

Il serait intéressant de savoir s'il est possible de généraliser la décomposition équiprojetable du chapitre 4 aux cas de systèmes zéro-dimensionnels non radicaux.

Autres perspectives

En ce qui concerne le changement d'ordre du chapitre 3, nous avons du supposer que la chaîne régulière en entrée engendrait un idéal premier. L'algèbre linéaire en début d'algorithme

qui détermine les couples de variables appelées *exchange data* et qui vont conduire les différentes spécialisations et remontées, doit aussi détecter, dans le cas de plusieurs composantes irréductibles, celles pour qui un jeu de variables sélectionnées est libre, et celles où ça n'est pas le cas.

Enfin, pour l'algorithmique du principe D5, nous souhaiterions étendre les résultats obtenus pour l'algorithme du Half-Gcd à tous les algorithmes ne manipulant que des opérations de base et des tests à zéro. Gageons que ça n'est pas une mince affaire, mais très prometteuse vue qu'une étape déterminante, l'estimation du coût de l'inversion, a été accomplie.

List of Figures

List of Tables

Algorithms

Bibliography

[1] The symbolicdata project, 2000–2002.

[2] I. Abdeljaouad-Tej, S. Orange, G. Renault, and A. Valibouze. Computation of the decomposition group of a triangular ideal. *Appl. Algebra Eng., Commun. Comput.*, 15(3):279–294, 2004.

[3] M.-E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Zeros, multiplicities, and idempotents for zero-dimensional systems. In *Progress in Math.*, volume 143 of *Algorithms in Algebraic Geometry and Applications*, pages 1–15. Proceedings MEGA'94, Birkhäuser, 1996.

[4] H. Anai, M. Noro, and K. Yokoyama. Computation of the splitting fields and the Galois groups of polynomials. In *Algorithms in algebraic geometry and applications (Santander, 1994)*, volume 143 of *Progr. Math.*, pages 29–50. Birkhäuser, Basel, 1996.

[5] E. A. Arnold. Modular algorithms for computing Gröbner bases. *J. Symb. Comp.*, 35(4):403–419, 2003.

[6] P. Aubry. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques*. PhD thesis, Université Paris VI, 1999.

[7] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *J. of Symbolic Computation*, 28(1,2):45–124, 1999.

[8] P. Aubry and M. Moreno Maza. Triangular sets for solving polynomial systems: a comparative implementation of four methods. *J. Symb. Comput.*, 28(1-2):125–154, 1999.

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

[10] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *International Conference on Numerical Mathematics*, volume 86 of *Inter.Series of Num. Math*, pages 11–30. Birkhäuser, 1988.

[11] Magali Bardet, Jean-Charles Faug?re, and Bruno Salvy. On the complexity of grbner basis computation of semi-regular overdetermined algebraic equations. In *International Conference on Polynomial System Solving*, page 71???74, November 2004. Proceedings of a conference held in Paris, France in honor of Daniel Lazard.

[12] S. Basu, R. Pollack, and M.-F. Roy. *Algorothms in real algebraic geometry*, volume 10 of *Algorithms and computation in mathematics*. Springer-Verlag, 2003.

[13] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoret. comp. Sci.*, 22:317–330, 1983.

[14] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147–150, 1984.

[15] D. Bernstein. Fast multiplication and its applications.

[16] D. J. Bernstein. Faster factorization into coprime.

[17] D. J. Bernstein. Factoring into coprimes in essentially linear time. *J. Algorithms*, 54(1):1–30, 2005.

[18] J.-B. Bost, H.Gillet, and C. Soulé. Heights of projective varieties and positive Green forms. *J. Amer. Math. Soc.*, 7(4):903–1027, 1994.

[19] F. Boulier and F. Lemaire. Computing canonical representatives of regular differential ideals. In *Proc. of ISSAC*, pages 37–46, Lilles, France, 2000. ACM Press.

[20] F. Boulier, F. Lemaire, and M. Moreno-Maza. Well known theorems on triangular systems. Technical report, LIFL, November 2001.

[21] F. Boulier, F. Lemaire, and M. Moreno Maza. Well known theorems on triangular systems and the D5 principle. In *Transgressive Computing 2006*, 2006.

[22] R.P. Brent, F.G. Gustavson, and D.Y.Y. Yun. Fast solution of Toeplitz systems of equations and computations of Padé approximants. *Journal of Algorithms*, 1:259–295, 1980.

[23] B. Buchberger and H. Möller. The construction of multivariate polynomials with preassigned zeros. In *Lecture Notes in Computer Science (EUROCAM'82)*, volume 144, pages 24–31, London, UK, 1982.

[24] P. Bürgisser, M. Clausen, and A. Shokrollahi. *Algebraic complexity theory*. Springer, 1997.

[25] L. Busé and M. Chardin. Implicitizing rational hypersurfaces using approximation complexes. *J. Symb. Comp.*, 40, 2005.

[26] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[27] F. Chen and T.W. Sederberg. Implicitization using moving curves and surfaces. In *Computer Graphics Annual Conference Series*, pages 301–308, 1995.

[28] S. Collart, M. Kalkbrenner, and D. Mall. Converting bases with the gröbner walk. *J. Symb. Comp.*, 24(3-4):465–470, 1997.

[29] D. Cox. Curves, surfaces and syzygies. *Contemp. math.*, 334:131–150, 2003.

[30] D. A. Cox, J. B. Little, and D. O'Shea. *Using algebraic geometry*, volume 185. Springer, New-York, 1998.

[31] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115. ACM, 2005.

[32] X. Dahan and É. Schost. Sharp estimates for triangular sets. In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 103–110. ACM Press, 2004.

[33] C. D'Andreas and A. Khetan. Implicitization of rational surfaces with toric varieties. Preprint.

[34] S. Dellière. *Triangularisation des systèmes constructibles - Applications à l'évaluation dynamique*. PhD thesis, Université de Limoges, 1999.

[35] Jean Della Dora, Claire Dicrescenzo, and Dominique Duval. About a new method for computing in algebraic number fields. In *EUROCAL '85: Research Contributions from the European Conference on Computer Algebra-Volume 2*, pages 289–290, London, UK, 1985. Springer-Verlag.

[36] D. Duval. *Diverses questions relatives au calcul formel avec des nombres algébriques*. PhD thesis, Université scientifique, technologique et médicale de Grenoble, 1987.

[37] D. Duval. Algebraic numbers: an example of dynamic evaluation. *J. Symb. Comput.*, 18(5):429–445, 1994.

[38] D. Eisenbud. *Commutative algebra. With a view toward algebraic geometry*, volume 150. Springer-Verlag, Berlin, 1995.

[39] M. Elkadi and B. Mourrain. Géométrie algébrique effective. à paraitre. Ch. 3 : Algèbres de dimension 0.

[40] T. Henin F. Boulier, L. Denis-Vidal and F. Lemaire. Lépisme. In *ICPSS*, pages 23–27. University of Paris VI, France, 2004.

[41] M. Moreno Maza F. Lemaire and Y. Xie. The `RegularChains` library.

[42] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993.

[43] A. Filatei, X. Li, M.Moreno Maza, and É Schost. Implementation techniques for fast polynomial arithmetic in a high-level programming environment. In *ISSAC'06*, pages 93–100. ACM, 2006.

[44] M.V. Foursov and M. Moreno Maza. On computer-assisted classification of coupled integrable equations. *J. Symb. Comp.*, 33:647–660, 2002.

[45] G. Gallo and B. Mishra. Efficient algorithms and bounds for Wu-Ritt characteristic sets. In T. Mora and C. Traverso, editors, *Progress in Math.*, volume 94, pages 235–248. Proceedings MEGA'90, Birkhäuser, 1990.

[46] T. Gautier and J.-L. Roch. NC2 computation of gcd-free basis and application to parallel +algebraic numbers computation. In *PASCO '97: Proceedings of the second international symposium on +Parallel symbolic computation*, pages 31–37. ACM Press, 1997.

[47] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for diophantine approximation. *J. of Pure and Applied Algebra*, 117/118:277–317, 1997.

[48] M. Giusti and J. Heintz. La détermination des points isolés et de la dimension d'une variété peut se faire en temps polynomial. In L. Robbiano and D. Eisenbud, editors, *Computational algebraic geometry and commutative algebra*, volume XXXIV of *Symposia Matematica*, pages 216–256. Cambridge U. Press, 1993.

[49] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *J. of Pure and Applied Algebra*, 124:101–146, 1998.

[50] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast ? In G. Cohen, M. Giusti, and T. Mora, editors, *Applied algebra, algebraic algorithms and Error Coding Codes*, volume 948 of *Lectures Notes in Computer Science*, pages 205–231. Proceedings AAECC-11, Springer, 1995.

[51] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. Le rôle des structures de données dans les problèmes d'élimination. *C. R. Acad. Paris*, 325:1223–1228, 1997.

[52] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *J. of Complexity*, 17(2):154–211, 2001.

[53] T. Gomez-Diàs. *Quelques applications de l'évaluatin dynamique*. PhD thesis, Université de Limoges, 1994.

[54] M.J. González-López and T. Recio. The ROMIN inverse geometric model and the dynamic evaluation method. In Arjeh M. Cohen, editor, *Proc. of the 1991 SCAFI Seminar, Computer Algebra in Industry*. Wiley, 1993.

[55] J. Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239–277, 1983.

[56] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *J. Complex.*, 16(1):70–109, 2000.

[57] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16(1):70–109, 2000.

[58] M. Hindry and J. Silverman. *Diophantine Geometry*. Number 201 in GTM. Springer-Verlag, 2001.

[59] É. Hubert. Notes on triangular sets and triangulation-decomposition algorithms. I. Polynomial systems. In *Symbolic and numerical scientific computation (Hagenberg, 2001)*, volume 2630 of *Lecture Notes in Comput. Sci.*, pages 1–39. Springer, Berlin, 2003.

[60] É. Hubert. Notes on triangular sets and triangulation-decomposition algorithms. II. Differential systems. In *Symbolic and numerical scientific computation (Hagenberg, 2001)*, volume 2630 of *Lecture Notes in Comput. Sci.*, pages 40–87. Springer, Berlin, 2003.

[61] G. Jeronimo, T. Krick, S. Sabia, and M. Sombra. The computational complexity of the Chow form. *Found. Computational Mathematics*, 4(1):41–117, 2004.

[62] M. Kalkbrener. *Three contributions to elimination theory.* PhD thesis, Kepler University, Linz, 1991.

[63] M. Kalkbrener. A generalized Euclidean algorithm for computing triangular representations of algebraic varieties. *J. Symbolic Comput.*, 15(2):143–167, 1993.

[64] M. Kalkbrener. On the complexity of Gröbner bases conversion. *J. Symbolic Computation*, 28(1–2):265–273, 1999.

[65] E. Kaltofen. Greatest common divisors of polynomials given by straight-line programs. *J. ACM*, 35(1):231–264, 1988.

[66] T. Krick, L. M. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke math. Journal*, 109:521–598, 2001.

[67] S. Lang. *Fundamentals of Diophantine geometry.* Springer-Verlag, New York, 1983.

[68] S. Lang. *Number Theory III: Diophantine Geometry*, volume 60 of *Encyclopædia of Mathematical Sciences.* Springer-Verlag, 1991.

[69] L. Langemyr. Algorithms for computing in algebraic extension. In T. Mora and C. Traverso, editors, *Effective Methods in Algebraic Geometry, Proceedings of MEGA'90*, volume 141, pages 235–248. Birkhäuser, 1990.

[70] D. Lazard. Ideal bases and primary decomposition: case of two variables. *J. Symbolic Comput.*, 1(3):261–270, 1985.

[71] D. Lazard. A new mthod for solving algebraic systems of positive dimension. *Disc. Appl. Math.*, 33, 1991.

[72] D. Lazard. Solving zero-dimensional algebraic systems. *J. of symbolic computation*, 13:147–160, 1992.

[73] D. Lazard. Resolution of polynomial systems. In X.-S. Gao and D. Wang, editors, *Proceedings of the fourth Asian Symposium (ASCM)*, pages 1–8, 2000.

[74] D. Lazard. 25 years of polynomial systems solving... and now ? In *Porceedings of ICPSS*, pages 10–12, Paris, France, November 2004.

[75] G. Lecerf. *Une alternative aux méthodes de réecriture pour la résolution des systèmes algébriques*. PhD thesis, École Polytechnique, 2001.

[76] G. Lecerf. Kronecker 0.166, a magma package for polynomial system solving. `http://www.math.uvsq.fr/~lecerf/software/kronecker`, 2002.

[77] G. Lecerf. Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. *Journal of Complexity*, 19(4):564–596, 2003.

[78] G. Lecerf and É. Schost. Fast multivariate power series multiplication in characteristic zero. *SADIO Electronic Journal on Informatics and Operations Research*, 5(1):1–10, September 2003.

[79] F. Lemaire. *Contribution à l'algorithmique en algèbre différentielle*. PhD thesis, Université Lille I, LIFL, 2002.

[80] F. Lemaire, M. Moreno Maza, and Y. Xie. The `regularchains` library. In *Maple conference 2005*, pages 355–368. I. Kotsireas Ed., 2005.

[81] H. Lombardi. Structures algébriques dynamiques, espaces topologiques sans points et programme de Hilbert. To appear in the Journal of Pure and Applied Algebra, 2003.

[82] F.-S. Macaulay. *The algebraic theory of modular systems*. Cambridge U. Press, 1916.

[83] M. G. Marinari, H. M. Möller, and T. Mora. On multiplicities in polynomial system solving. *Trans. Amer. Math. Soc.*, 348(8):3283–3321, 1996.

[84] M. G. Marinari and T. Mora. A remark on a remark by Macaulay or enhancing Lazard structural theorem. *Bull. Iranian Math. Soc.*, 29(1):1–45, 85, 2003.

[85] M. Moreno Maza. *Calculs de pgcd au dessus de tours d'extensions simples et résolution des sysèmes d'équations algébriques*. PhD thesis, Université Paris VI, 1997.

[86] P. J. McCarthy. *Algebraic extensions of fields*. Dover, New York, 1991.

[87] T. Mora. *Solving Polynomial Equation Systems I. The Kronecker-Duval Philosophy*. Number 88 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2003.

[88] M. Moreno Maza. On triangular decompositions of algebraic varieties. Technical Report TR 4/99, NAG Ltd, Oxford, UK, 1999. Presented at the MEGA-2000 Conference, Bath, England. `http://www.csd.uwo.ca/~moreno`.

[89] M. Moreno Maza and R. Rioboo. Polynomial gcd computations over towers of algebraic extensions. In *Proc. AAECC-11*, pages 365–382. Springer, 1995.

[90] S. Morrison. The differential ideal $[P] : M^\infty$. *J. Symb. Comput.*, 28(4-5):631–656, 1999.

[91] M. Noro. Modular dynamic evaluation. In *Proc. of ISSAC MMVI*, Genova, Italy, 2006. ACM Press.

[92] M. Noro and K. Yokoyama. A modular method to compute the rational univariate representation of zero-dimensional ideals. *J. Symb. Comput.*, 28(1-2):243–263, 1999.

[93] V. Y. Pan. Simple multivariate polynomial multiplication. *J. Symb. Comput.*, 18(3):183–186, 1994.

[94] P. Philippon. Critères pour l'indépendance algébrique. *IHES Publ. math*, 64:5–52, 1986.

[95] P. Philippon. Sur des hauteurs alternatives III. *J. Math. Pures Appl.*, 74(4):345–365, 1995.

[96] A. Recski. *Matroid theory and its applications in electric network theory and in statics*. Springer-Verlag, New-York, 1989.

[97] G. Renault. *Cacul efficace de corps de décomposition*. PhD thesis, Université de Paris VI, 2005.

[98] Renaud Rioboo. Real algebraic closure of an ordered field: implementation in axiom. In *ISSAC '92: Papers from the international symposium on Symbolic and algebraic computation*, pages 206–215, New York, NY, USA, 1992. ACM Press.

[99] J. F. Ritt. Differential equations from an algebraic standpoint. *Colloquiium publications of the AMS*, 14, 1932.

[100] J F. Ritt. *Differential algebra*. Dover publication, 1966.

[101] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Appl. Algebra Eng. Commun. Comput.*, 9(5):433–461, 1999.

[102] É. Schost. *Sur la résolution des systèmes polynomiaux à paramètres*. PhD thesis, École Polytechnique, 2000.

[103] É. Schost. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing*, 13(5):349–393, 2003.

[104] É. Schost. Multivariate power series multiplication. In *ISSAC '05: Proceedings of the 2005 international symposium on Symbolic and algebraic computation*, pages 293–300. ACM Press, 2005.

[105] Éric Schost. Complexity results for triangular sets. *J. Symb. Comput.*, 36(3-4):555–594, 2003.

[106] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, October 1980.

[107] D. Shannon and M. Sweedler. Using Gröbner bases to determine algebra membership, split surjective algebra homomorphisms determine birational equivalence. *J. Symbolic Computation*, 6(2–3):267–273, 1988.

[108] J. Silverman. An introduction to height functions. `http://www.math.brown.edu/~jhs/HtSurveyMSRIJan06.pdf`, January 2006. Slides from lectures presented at MSRI.

[109] M. Sombra. *Estimaciones para el teorema de ceros de Hilbert*. PhD thesis, Universidad de Buenos Aires, 1998.

[110] A. J. Sommese, J. Verschelde, and C. W. Wampler. Numerical irreducible decomposition using projections from points on the components. In *Symbolic computation: solving equations in algebra, geometry, and engineering*, volume 286 of *Contemp. Math.*, pages 37–51. Amer. Math. Soc., 2001.

[111] C. Soulé. Géometrie d'Arakelov et nombres transcendants. In *Journées arithmétiques de Luminy (1989)*, pages 355–371. Astérique, volume 198-200, 1992.

[112] A. Szanto. *Computation with polynomial systems*. PhD thesis, Cornell university, 1999.

[113] Q.-N. Tran. Efficient gröbner walk conversion for implicitization of geometric objects. *Comput. auided Geom. Des.*, 21(9):837–857, 2004.

[114] W. Trinks. On improving approximate results of buchberger's algorithm by newton's method. *SIGSAM Bull.*, 18(3):7–11, 1984.

[115] J. van der Hoeven. Notes on the Truncated Fourier Transform. Technical Report 2005-5, Universit? Paris-Sud, Orsay, France, 2005.

[116] Joris van der Hoeven. The truncated fourier transform and applications. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 290–296. ACM Press, 2004.

[117] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, NY, USA, 1999.

[118] D. Wang. An elimination method for polynomial systems. *J. Symb. Comput.*, 16(2):83–114, 1993.

[119] D. Welsh. *Matroid theory*. Academic Press [Harcourt Brace Jovanovich Publishers], London, 1976.

[120] W. T. Wu. On zeros of algebraic equations - an application of Ritt principle. *Kexue Tongbao*, 5:1–5, 1986.

[121] É. Schost X. Dahan, M. Moreno Maza and Y. Xie. On the complexity of the d5 principle. In *Proc. of Transgressive Computing 2006*, Granada, Spain, 2006.

[122] C.K. Yap. *Fundamental Problems in Algorithmic Algebra*. Princeton University Press, 1993.

[123] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings EUROSAM' 79*, volume 72 of *Lecture Notes in Computer Science*. Springer, 1979.

Résumé

Les systèmes polynomiaux sous forme triangulaire, notamment les *chaînes régulières* et en particulier les *ensembles triangulaires de Lazard*, sont des structures de données simples, permettant d'envisager des calculs modulaires (par spécialisation des coefficients, puis remontée via un opérateur de Newton-Hensel), de "résoudre" les systèmes de polynômes (méthodes de "triangulations") et de représenter des tours d'extensions de corps pour calculer avec les nombres algébriques.

Dans ces trois domaines, les méthodes et résultats nouveaux apportés, notamment sur le plan de la complexité, étendent le champs d'application des ensembles triangulaires, et leur impact face à d'autres méthodes de manipulation des équations polynomiales, surtout les bases de Gröbner.

Tout d'abord la complexité en espace des coefficients n'est qu'en croissance quadratique en fonction de données géometriques naturelles. Conséquence directe en est un opérateur de Newton (triangulaire) requérant moins d'étapes de remontée, et donc des méthodes modulaires plus encourageantes. Il en est ainsi pour la *décomposition équiprojetable*, premier algorithme de triangulation des systèmes basé sur une méthode modulaire, et pour le problème du changement d'ordres monomiaux en *dimension positive*, dans des cas assez particuliers toutefois pour une première approche.

Par ailleurs, calculer modulo un ensemble triangulaire en suivant le modèle de l'*évaluation dynamique*, se voit doté, 20 ans après sa création, d'un premier résultat de complexité satisfaisant.

Mots-clés. Résolution des systèmes polynomiaux, décompositions triangulaires, évaluation dynamique, méthodes modulaires.

Abstract

The polynomial systems in their triangular shape, notably the *regular chains* and especially the *Lazard triangular sets*, are simple data structures, permitting to consider modular computations (by specialization of the coefficients, then lifting through the Newton-Hensel operator), to "solve" the polynomial systems ("decomposition-triangulations" methods) and to represent tours of fields extensions to compute with algebraic numbers.

For those three topics, the methods and results provided here, notably on the complexity front, extend the fields of applications of triangular sets, and their impact compared to other methods of manipulation of algebraic equations, especially the Gröbner bases.

First of all the space complexity of the coefficients is only on quadratic growth in function of natural geometric data. Straightforward corollary is a (triangular) Newton operator requiring less lifting steps, hence more promising modular methods. So it is for the *equiprojectable decomposition*, first algorithm of triangulation of polynomial systems based on a modular method, and for the problem of the change of monomials orderings in *positive dimension*, yet in some quite specific cases for a first approach.

In addition, computing modulo a triangular set by following the *dynamic evaluation* model, is now endowed, 20 years after its apparition, of a first statisfaying complexity study.

Keywords. Polynomial system solving, triangular decompositions, dynamic evaluation, modular methods.