THÈSE de DOCTORAT de l'UNIVERSITÉ PARIS VI
Spécialité :
INFORMATIQUE

présentée par :

Mohab SAFEY EL DIN

pour obtenir le grade de DOCTEUR de l'UNIVERSITÉ PARIS VI

Résolution réelle des systèmes polynomiaux en dimension positive.

Soutenue le 5 janvier 2001.

Composition du jury :

M. Michel COSNARD	Examinateur
M. Marc GIUSTI	Rapporteur
M. Daniel LAZARD	Directeur de thèse
M. Laureano GONZALEZ-VEGA	Rapporteur
M. Fabrice ROUILLIER	Co-encadrant
Mme Marie-Françoise ROY	Co-directrice de thèse

Introduction

L'axe de cette thèse est l'étude des solutions réelles des systèmes polynomiaux et plus spécifiquement la «*résolution réelle*» des systèmes polynomiaux de dimension positive, c'est-à-dire contenant une infinité de solutions complexes.

Qu'est-ce que résoudre ?

Dans ce document, nous distinguerons la «*résolution algébrique*» de la «*résolution réelle*» des systèmes polynomiaux. Ces deux termes sont à définir. Le premier concerne l'étude des solutions complexes par la réécriture du système d'équations de départ sous une forme exploitable. Le second concerne l'étude de l'ensemble des solutions réelles qui doit être facilitée par la résolution algébrique.

Considérons à titre d'exemple le cas des systèmes polynomiaux admettant un nombre fini de solutions complexes (systèmes zéro-dimensionnels). Par *résolution algébrique* d'un tel système, on peut entendre, par exemple, la réécriture de l'ensemble des solutions sous la forme suivante :

$$\left\{\begin{array}{l} G_0(T)X_n - G_n(T) = 0 \\ \vdots \\ G_0(T)X_1 - G_1(T) = 0 \\ F(T) = 0 \end{array}\right.$$

où $X_1, \ldots, X_n$ sont les variables du système de départ, T est une nouvelle variable, et les polynômes $F, G_0, G_1, \ldots, G_n$ sont en une seule variable T. Il est alors clair qu'à partir de cette représentation des solutions complexes, l'étude des solutions réelles peut se faire en comptant et en isolant les racines réelles du polynôme F (une bijection entre les racines réelles de F et celles du système de départ étant assurée).

Le calcul de telles représentations a fait l'objet de plusieurs études récentes. Sur la base de ces résultats théoriques et des performances pratiques des logiciels dont ils sont l'aboutissement, il est raisonnable de considérer que la résolution algébrique et la résolution réelle des systèmes polynomiaux zéro-dimensionnels constituent aujourd'hui un domaine bien maitrisé dans le cadre des méthodes formelles.

Nous nous intéresserons donc à la «*résolution réelle*» des systèmes polynomiaux de dimension positive – c'est-à-dire les systèmes polynomiaux admettant un nombre infini de solutions complexes. En dimension positive, plusieurs définitions du terme «*résolution* *réelle»* sont possibles. En effet, il peut s'agir de décider uniquement si une variété algébrique réelle est vide, en donner au moins un point par composante connexe, calculer la dimension réelle de la variété, etc. Les algorithmes récents d'élimination des quantificateurs utilisent intensivement des routines calculant au moins un point par composante connexe d'ensembles semi-algébriques. Ces routines utilisent elles-mêmes d'autres routines calculant au moins un point par composantes connexes sur des variétés algébriques réelles. Nous choisissons donc d'adopter comme définition du mot «résoudre» : *«donner au moins un point par composante connexe sur une variété algébrique réelle»*.

Problématique

L'algorithme le plus connu permettant de calculer au moins un point par composante semi-algébriquement connexe sur une variété algébrique réelle quelconque est l'algorithme de **Décomposition Cylindrique Algébrique** de Collins [28]. Cet algorithme effectue un traitement récursif sur les variables, et ses spécifications sont en fait bien plus générales : il permet notamment de déterminer toutes les conditions de signes vérifiées par une famille de polynômes. Sa complexité théorique est doublement exponentielle en le nombre de variables. En pratique, cet algorithme – qui est le seul permettant de donner au moins un point par composante connexe sur une variété réelle et qui soit implanté – ne permet guère de résoudre des problèmes de plus de trois variables.

Une des alternatives à l'algorithme de Décomposition Cylindrique Algébrique est la famille d'algorithmes que l'on peut regrouper sous le nom de **Méthode des Points Critiques** [53, 55, 74, 56, 12, 13, 87].

Ces algorithmes sont de complexité simplement exponentielle en le nombre de variables en terme de taille de la sortie et de nombre d'opérations. Cette borne de complexité est asymptotiquement optimale. Elle est atteinte sur certains exemples. La méthode des points critiques est basée sur le calcul des points critiques d'une fonction «bien choisie» restreinte à la variété que l'on veut étudier, l'objectif étant de ramener le problème à l'étude d'un système polynomial zéro-dimensionnel, dont l'ensemble des solutions contient les points critiques de cette fonction. Cette méthode permet de se ramener directement à l'étude d'un nombre fini de points complexes et d'éviter tout traitement récursif sur les variables. À notre connaissance, il n'existait aucune implantation d'algorithmes basés sur cette méthode avant le début de nos travaux.

Ainsi, alors qu'en terme d'étude de complexité théorique, le domaine semble bien maitrisé – au point que des algorithmes asymptotiquement optimaux sont proposés –, ce sont des **algorithmes permettant d'obtenir des implantations efficaces en pratique** qui font défaut.

Les causes de cette situation sont multiples. La complexité théorique des algorithmes de la méthode de points critiques – comme la plupart des algorithmes concernant la résolution des systèmes polynomiaux – est évaluée avec une constante en exposant. Ainsi, deux algorithmes de même complexité peuvent avoir des comportements qui diffèrent de façon exponentielle en pratique. Pour peu que cette constante soit élevée, un algorithme de complexité simplement exponentielle peut avoir un mauvais comportement en pratique sur des problèmes que le Calcul Formel peut prétendre résoudre comparativement à un algorithme de complexité doublement exponentielle. Aussi, peut-on remarquer que pour de très nombreux algorithmes de résolution algébrique, des différences de temps de calcul sensibles peuvent être constatées sur des problèmes de taille équivalente en entrée. Ces temps dépendent plus de la *taille de la sortie* de la solution retournée que de la *taille d'entrée* des problèmes que l'on veut résoudre.

Les théorèmes de complexité concernant l'algorithme de Décomposition Cylindrique Algébrique et les algorithmes de la méthode des points critiques indiquent que la sortie de l'algorithme de Décomposition Cylindrique Algébrique est doublement exponentielle alors que celle des algorithmes de la méthode des points critiques est simplement exponentielle.

Ainsi, il est raisonnable de penser qu'en contrôlant mieux la croissance des données intermédiaires apparaissant dans ces algorithmes, nous pourrons obtenir de bons résultats en pratique. L'objectif de cette thèse est d'adapter les algorithmes de la méthode des points critiques de manière à obtenir des algorithmes efficaces en pratique. Le document est composé de deux parties dont nous détaillons le contenu ci-dessous.

Résultats et résumés des différents chapitres

Première Partie : Préliminaires

Nous introduisons les outils mathématiques et algorithmiques que nous utiliserons dans la suite du document. Exceptés certains résultats du chapitre 6, cette partie est essentiellement constituée de rappels.

Chapitre 1 : Rappels de géométrie algébrique réelle

Ce chapitre rappelle certaines notions élémentaires de géométrie algébrique réelle qui seront utilisées dans ce document.

Chapitre 2 : Suite des sous-résultants

La suite des sous-résultants est un outil fondamental en Calcul Formel. En particulier, le calcul de cette suite est l'ingrédient essentiel de la première étape de l'algorithme de Décomposition Cylindrique Algébrique. Dans ce chapitre, nous rappelons la définition de la suite des sous-résultants associée à un couple de polynômes. Puis, nous rappelons comment on peut en déduire le degré du pgcd de ce couple ainsi que les propriétés de spécialisation de cette suite.

Les rappels contenus dans ce chapitre seront utilisés dans le chapitre 3 de la partie I, ainsi que dans les chapitres 2 et 3 de la partie II.

Chapitre 3 : Décomposition Cylindrique Algébrique

Ce chapitre est dédié à l'algorithme de Décomposition Cylindrique Algébrique [28], qui est le plus connu en matière de résolution réelle des systèmes polynomiaux de dimension positive. Soit R un corps réel clos. Nous rappelons dans un premier temps les définitions de décomposition cylindrique dans R^n et de *décomposition cylindrique algébrique adaptée à une famille de polynômes* de $R[X_1,\ldots,X_n]$. Puis, nous réétablissons sous quelles conditions les racines réelles d'un polynôme univarié à coefficients dans $R[X_1,\ldots,X_{n-1}]$ varient continument lorsque $X_1,\ldots,X_{n-1}$ varient continument dans un ensemble semi-algébrique de R^{n-1}. Nous rappelons alors comment en déduire l'opérateur de projection de l'algorithme de Décomposition Cylindrique Algébrique de Collins. Nous décrivons succintement cet algorithme puis nous en rappelons la complexité théorique.

La rédaction de ce chapitre reprend essentiellement celles de [31, 14].

Chapitre 4 : Résolution des systèmes zéro-dimensionnels

Le cas zéro-dimensionnel est important car nous nous y ramènerons dans le cadre des méthodes de points critiques. Nous commençons par rappeler les définitions et propriétés élémentaires des bases de Gröbner. Nous rappelons aussi les résultats de [42, 59] qui énoncent les propriétés de spécialisations des bases de Gröbner. Le calcul d'une base de Gröbner peut être une *étape* du processus de résolution algébrique des systèmes zéro-dimensionnels, dont l'aboutissement est le calcul d'une Représentation Univariée Rationnelle qui est la réécriture de l'ensemble des solutions sous la forme mentionnée ci-dessus. Nous en rappelons la définition que l'on peut trouver dans [80, 81] ainsi que quelques propriétés et les étapes importantes de son calcul à partir d'une base de Gröbner.

Chapitre 5 : Ensembles triangulaires

Après un bref historique, nous rappelons la définition d'ensembles triangulaires de polynômes, d'ensembles triangulaires consistants ainsi que du saturé d'un ensemble triangulaire. Puis nous justifions la notion de régularité pour les ensembles triangulaires et rappelons les définitions d'ensembles triangulaires réguliers et d'ensembles triangulaires réguliers séparables. Nous en donnons ensuite quelques propriétés [6]. Enfin, nous rappelons les spécifications des algorithmes de décomposition en ensembles triangulaires réguliers et séparables au sens de Lazard [65, 72] et au sens de Kalkbrener [60, 5]. Enfin, nous rappelons les spécifications de quelques routines associées aux algorithmes de Lazard et de Kalkbrener.

Ces rappels seront intensivement utilisés dans la partie II de ce document.

Chapitre 6 : Un algorithme de bonne complexité basé sur la méthode des points critiques

La méthode des points critiques consiste à calculer les points critiques d'une fonction «bien choisie» qui atteint ses valeurs critiques en chacune des composantes connexes de la variété étudiée. Il en existe plusieurs variantes [53, 55, 74, 56, 12, 13, 87, 9, 10, 11]. Dans ce chapitre, nous rappelons l'algorithme décrit dans [13, 12, 87]. Après deux déformations infinitésimales, le problème est ramené à l'étude d'une hypersurface lisse dont le lieu réel est compact et telle que le système d'équations polynomiales caractérisant les points critiques de la fonction de projection sur la première coordonnée est directement une base de Gröbner pour l'ordre du degré lexicographique.

Une Représentation Univariée Rationnelle de l'ensemble des solutions peut alors être calculée en utilisant les algorithmes décrits dans [80, 81]. Il faut ensuite calculer les limites des solutions ainsi décrites lorsque les deux infinitésimaux tendent vers zéro pour obtenir un point par composante connexe sur l'hypersurface de départ. Nous donnons dans ce chapitre de nouveaux algorithmes, extraits de [84], permettant d'effectuer ce calcul.

Enfin, nous rappelons la complexité théorique de l'algorithme obtenu. Nous procédons alors à une brève analyse expérimentale de cet algorithme sur un exemple simple. Plusieurs points bloquants apparaissent :

- la première étape qui permet de se ramener au cas d'une hypersurface compacte introduit des singularités;
- la deuxième étape, qui permet de se ramener au cas lisse, engendre automatiquement une base de Gröbner zéro-dimensionnelle pour l'ordre du degré lexicographique;
- le degré de cette base de Gröbner est élevé au point qu'il empêche toute résolution par les techniques connues à ce jour;
- à cette difficulté s'ajoute celle de faire des calculs sur une arithmétique infinitésimale.

Deuxième Partie : Nouveaux Algorithmes

Cette partie contient les contributions de ce document à la résolution réelle des systèmes polynomiaux en dimension positive. Soit K un corps réel, R sa clôture réelle, et C sa cloture algébrique.

Chapitre 1 : Le cas des hypersurfaces

Ce travail, effectué en collaboration avec F. Rouillier et M.-F. Roy [84], reprend une idée classique [89], (voir aussi [56, 80]) et qui consiste à utiliser calculer les points critiques de la fonction distance à un point restreinte à la variété que l'on veut étudier.

Soit P un polynôme irréductible de $K[X_1,\ldots,X_n]$. Pour tout choix de point $A \in K^n$, l'ensemble $\mathcal{C}(V(P),A)$ des points M défini par le système d'équations polynomiales :

$$P(M) = 0, \ \overrightarrow{\mathrm{grad}}_M\ (P) // \overrightarrow{AM}$$

contient les points critiques de la fonction distance au point A restreinte à $V(P) \bigcap R^n$ et donc intersecte chaque composante connexe de $V(P) \bigcap R^n$. De plus, si $\{M \in C^n \mid P(M) = 0, \quad \overrightarrow{\mathrm{grad}}_M\ (P) = \overrightarrow{0}\} = \emptyset$, l'ensemble des points $A \in C^n$ tels que l'idéal défini par le système d'équations ci-dessus n'est pas zéro-dimensionnel et radical est contenu dans une variété algébrique de C^n. On en déduit un premier algorithme prenant en entrée une hypersurface sans singularités complexes définie par un polynôme et retournant au moins un point par composante connexe de $V(P) \bigcap R^n$.

Dans le cas où $\{M \in C^n \mid P(M) = 0, \quad \overrightarrow{\mathrm{grad}}_M\ (P) = \overrightarrow{0}\}$ contient une infinité de points, l'hypersurface définie par $P - \varepsilon$ (où ε est un infinitésimal) est lisse. En appliquant les résultats précédents, on peut trouver un point A tel que $\mathcal{C}(V(P-\varepsilon),A)$ est un ensemble fini de points. Nous montrons alors que l'ensemble des points appartenant qui sont limites de $\mathcal{C}(V(P-\varepsilon),A)$ lorsque ε tend vers 0 intersecte chaque composante semi-algébriquement connexe de $V(P)\bigcap R^n$.

Pour traiter ces cas, il est nécessaire de calculer une base de Gröbner puis calculer une Représentation Univariée Rationnelle à coefficients dans $K(\varepsilon)$, et les limites des racines bornées des solutions qu'elle représente. Nous donnons des outils pour calculer une base de Gröbner dans $K(\varepsilon)$ en utilisant les propriétés de spécialisation décrites dans [42]. Puis nous montrons que si nous spécialisons le paramètre ε en un point tel que le système obtenu engendre un idéal zéro-dimensionnel et radical, tout élément séparant pour les racines de cet idéal est séparant pour les solutions du système d'équations à coefficients dans $K(\varepsilon)$.

En collaboration avec E. Schost, nous complétons ces travaux en montrant comment certifier l'algorithme probabiliste proposé dans [88] pour calculer des Représentations Univariées Rationnelles à coefficients dans $K(\varepsilon)$.

Enfin, nous proposons un nouvel algorithme permettant de calculer les limites des solutions bornées de systèmes d'équations polynomiales à coefficients dans $K(\varepsilon)$. Cet algorithme n'effectue qu'un seul calcul de développements en séries de Puiseux, ce qui permet d'éviter plusieurs calculs difficiles effectués par l'algorithme proposé dans le chapitre 6 de la partie I.

L'algorithme final, nommé **HA3**, effectue une seule déformation infinitésimale uniquement lorsque l'hypersurface étudiée contient une infinité de points singuliers. Les outils algorithmiques mis en œuvre sont divers : bases de Gröbner et Représentations Univariées Rationnelles, remontées de Newton-Hensel , approximants de Padé, et développements en sérires de Pusieux. Dans un cadre certifié, nous n'obtenons pas de résultats de complexité à cause des calculs de bases de Gröbner et du choix du point A. En revanche, dans un cadre probabiliste le premier choix du point A est bon, et les techniques d'élimination basées sur la représentation des polynômes par des programmes d'évaluation [43, 44, 45, 46, 47] permettent d'obtenir une complexité simplement exponentielle car nous avons montré que les systèmes zéro-dimensionnels étudiés par **HA3** vérifient les hypothèses d'applicabilité de ces algorithmes. Il convient néanmoins de signaler qu'à notre connaissance, il n'existe pas d'algorithmes tirant profit de cette structure de données pour compter et isoler les racines réelles d'un polynôme univarié.

Dans une dernière section, nous procédons à l'étude expérimentale de l'algorithme obtenu. Dans les cas où la déformation infinitésimale n'est pas nécessaire, nous pouvons traiter des problèmes qui ne sont pas accessibles à la Décomposition Cylindrique Algébrique de Collins. La sortie de nos algorithmes est de taille bien inférieure à celle de l'algorithme de Collins. Les répercussions dans les temps de calcul sont sensibles. Dans les cas où la déformation infinitésimale est nécessaire, les conclusions sont plus mitigées. En effet, sur les exemples qui ont pu être traités, l'algorithme de Décomposition Cylindrique Algébrique de Collins a une sortie de taille inférieure à celle de l'algorithme **HA3**. Il faut noter que tous les exemples considérés sont en trois variables. En termes de temps de calcul, sur la classe d'exemples considérés, l'algorithme de Décompostion Cylindrique Algébrique résoud en quelques secondes des problèmes que l'algorithme **HA3** ne peut résoudre. Il est alors raisonnable de penser que sur la plupart des problèmes de même degré et contenant plus de variables, l'algorithme **HA3** ne devrait pas pouvoir donner de meilleurs résultats. La principale raison est que le degré de l'idéal zéro-dimensionnel étudié dans $K(\varepsilon)[X_1,\ldots,X_n]$ est trop grand comparativement à la somme des degrés des Représentations Univariées Rationnelles retournées par **HA3** : en effet plusieurs racines bornées du système zéro-dimensionnel à coefficients dans $K(\varepsilon)[X_1,\ldots,X_n]$ convergent vers un seul point lorsque ε tend vers 0. C'est donc la *croissance des données intermédiaires* qui est mal contrôlée.

Chapitre 2 : Systèmes d'équations

L'ensemble des singularités d'une hypersurface est une variété algébrique de dimension strictement inférieure à celle de l'hypersurface. Si on sait étudier les points critiques de la fonction distance restreinte à cet ensemble de singularités, on peut alors donner au moins un point par composante connexe sur une hypersurface réelle, sans déformation infinitésimale, en étudiant des variétés algébriques dont la dimension baisse progressivement.

Ce chapitre présente les travaux, effectués en collaboration avec P. Aubry et F. Rouillier, qui mettent en œuvre cette démarche.

Soit $(P_1,\ldots,P_s) \subset K[X_1,\ldots,X_n]$ une famille de polynômes. Nous montrons dans un premier temps que si $V(P_1,\ldots,P_s)$ est une variété algébrique *équi-dimensionnelle* de dimension d et si $\langle P_1,\ldots,P_s\rangle$ est radical, alors on peut trouver un point $A \in K^n$ tel que l'ensemble de points

$$\mathcal{C}(V(P_1,\ldots,P_s),A) = \{M \in V(P_1,\ldots,P_s) \mid \mathrm{rank}(\overrightarrow{\mathrm{grad}}_M\,(P_1),\ldots,\overrightarrow{\mathrm{grad}}_M\,(P_s),\overrightarrow{AM}) \leq n-d\}$$

intersecte chaque composante semi-algébriquement connexe de $V(P_1,\ldots,P_s)\bigcap R^n$ et est la réunion d'une composante de dimension zéro, contenant les points critiques réguliers de la fonction distance au point A, et de l'ensemble des singularités de $V(P_1,\ldots,P_s)$. On en déduit un algorithme – nommé **SA1** – utilisant une routine de décomposition équi-dimensionnelle radicale, et qui donne au moins un point par composante semi-algébriquement connexe sur une variété algébrique réelle quelconque, sans restriction sur le système d'équations définissant la variété. La terminaison de l'algorithme est assurée car à chaque étape, les variétés considérées sont de dimension inférieure à celles qui étaient étudiées au pas précédents. Cet algorithme n'effectue aucune déformation infinitésimale.

La caractérisation algébrique de $\mathcal{C}(V(P_1,\ldots,P_s),A)$ se fait en effectuant des calculs de déterminants sur des mineurs de la matrice jacobienne à laquelle on a rajouté le vecteur $\overrightarrow{AM}$. Le calcul de ces déterminants peut devenir une étape bloquante de l'algorithme en particulier lorsque la variété est définie par un trop grand nombre de polynômes. En imposant à la routine de décomposition équi-dimensionnelle de renvoyer des bases de Gröbner lexicographiques engendrant le saturé d'un ensemble triangulaire régulier séparable qui en est extrait, on peut calculer ces déterminants en ne considérant que les polynômes de l'ensemble triangulaire. Ainsi, pour une composante de dimension d, seuls d déterminants sont à calculer. Nous obtenons un nouvel algorithme – nommé **SA2** – qui est plus efficace que le précédent. Nous montrons que si la routine de décomposition équi-dimensionnelle radicale est une décomposition en idéaux premiers, non seulement les résultats précédents restent vrais mais la sortie de nos algorithmes est plus petite, et certaines branches de calculs sont évitées.

Ce dernier algorithme utilise des méthodes d'élimination fondées sur les bases de Gröbner. et des propriétés des ensembles triangulaires séparables. Nous avons tenté, en collaboration avec P. Aubry et F. Rouillier, de substituer les calculs de bases de Gröbner par des algorithmes de décompositions en ensembles triangulaires réguliers et séparables.

Nous montrons dans un premier temps comment utiliser les décompositions de Lazard dans nos algorithmes. Puis, afin de garantir une plus grande efficacité à l'algorithme obtenu nous montrons comment calculer une décomposition de Lazard en n'utilisant qu'une routine de décompositions en ensembles triangulaires réguliers au sens de Kalkbrener. Nous montrons alors comment assurer le calcul des composantes de dimension zéro des ensembles $\mathcal{C}(V(P_1,\dots,P_s),A)$ en utilisant une routine prenant en compte des inéquations. Ceci nous permet d'adapter l'algorithme **SA2** de manière à n'utiliser que des décompositions en ensembles triangulaires de Kalkbrener.

Dans la troisième section, nous étudions deux cas particuliers. Le premier – que nous appellerons position générale – est caractérisé par la structure des bases de Gröbner apparaissant en cours de calcul : celles-ci sont des ensemble triangulaires de polynômes qui sont tous de degré 1 et unitaires en leur variable principale. Le deuxième cas particulier étudié est le cas des variétés algébriques réelles compactes. Dans le cas de la position générale, l'algorithme **SA2** est modifié de manière à ramener l'étude de la variété considérée V à celle d'une hypersurface dans C^{d+1} (où d est la dimension de V). Dans le cas des variétés algébriques réelles compactes, la fonction de projection sur un axe peut être utilisé à la place de la fonction distance.

Dans la dernière section, nous validons notre approche : en comparant l'algorithme **SA2** à l'algorithme obtenu dans le chapitre précédent dans le cas d'hypersurfaces contenant une infinité de points singuliers. Nous montrons que l'algorithme **SA2** permet – au moins sur ce type d'exemples – de mieux gérer la croissance des données intermédiaires. L'impact sur les temps de calcul est sensible : nous pouvons traiter en des temps voisins de la minute des problèmes totalement inaccessibles à l'algorithme **HA3**. Puis nous montrons que les optimisations dont bénéficie l'algorithme **SA2** lui permettent d'être plus efficace que l'algorithme **SA1**. Nous montrons ensuite que la sortie de l'algorithme **SA2** est de taille inférieure à celle de l'algorithme de Décomposition Cylindrique Algébrique et que ceci permet de traiter des problèmes beaucoup plus difficiles, que ceux accessibles à l'algorithme de Décomposition Cylindrique Algébrique. Nous procédons ensuite à l'étude des algorithmes obtenus dans le cas particulier de variétés algébriques réelles compactes et montrons que lorsque ce sera possible, il faudra préférer la fonction de projection sur un axe à la fonction distance à un point : les temps de calcul et les sorties sont inférieurs à ceux obtenus lorsqu'on utilise la fonction de projection sur un axe.

Chapitre 3 : Vers les systèmes d'équations et d'inéquations polynomiales

Considérons le système d'équations algébriques :

$$(S)\quad \left\{ \begin{array}{l} xz-1=0 \\ y-x=0 \end{array} \right.$$

L'algorithme **SA2** produit un système zéro-dimensionnel de degré 4 dont l'ensemble des racines réelles intersecte chaque composante semi-algébriquement connexe de $V(S)\bigcap \mathbb{R}^n$. Or, il est clair que l'on peut décider si $V(S)\bigcap \mathbb{R}^n$ est vide ou pas en étudiant uniquement l'hypersurface définie par $y-x=0$. On trouve alors un système zéro-dimensionnel de degré 2. Pour trouver au moins un point par composante connexe, il faut étudier d'une part $V(S)\bigcap V(x)$ et d'autre part l'ensemble semi-algébrique de $\mathbb{R}^2$ défini par :

$$y-x=0,\quad x\neq 0.$$

Cette étude engendre deux systèmes zéro-dimensionnels de degré 1. En collaboration avec P. Aubry et F. Rouillier, nous décrivons dans ce chapitre comment mettre en œuvre ce type de stratégie. Dans une première section, nous montrons comment ramener l'étude d'une variété équi-dimensionnelle de dimension d qui peut être définie par une base de Gröbner lexicographique engendrant le saturé d'un ensemble triangulaire régulier et séparable, à l'étude d'une variété de dimension strictement inférieure à d et l'étude d'un ensemble semi-algébrique de R^d défini par une seule égalité et plusieurs inéquations.

Dans un deuxième temps, nous rappelons les résultats de [13, 12, 87] qui permettent de ramener l'étude d'un ensemble semi-algébrique de R^n à l'étude de plusieurs variétés algébriques réelles de $R\langle\varepsilon\rangle^n$ où ε est un infinitésimal. Nous montrons alors comment adapter l'algorithme **SA2** pour obtenir un algorithme permettant de donner au moins un point par composante semi-algébriquement connexe dans un ensemble semi-algébrique. Nous en déduisons deux algorithmes. Le premier – nommé **SA4** – fonctionne sans infinitésimaux, et décide du vide d'une variété algébrique réelle *dans certains cas fréquents en pratique* que nous avons appelé «position favorable». Le deuxième – nommé **SA5** – introduit un seul infinitésimal et trouve au moins un point par composantes semi-algébriquement connexes sur une variété algébrique réelle quelconque. Notons que cet infinitésimal est de nature différente de celui que nous avons introduit dans le chapitre 1 de cette partie : il n'est pas destiné à tendre vers 0 et simule réellement un rationnel «suffisamment petit». Quelques optimisations sont apportées. Nous montrons en particulier comment calculer une décomposition équi-dimensionnelle d'un système d'équations polynomiales à coefficients dans $K(\varepsilon)$ en ne faisant des calculs que sur les entiers.

Puis nous validons expérimentalement notre démarche. Nous montrons que la sortie des algorithmes **SA4** et **SA5** est de taille inférieure à celle de l'algorithme **SA2**. En terme de temps de calcul, lorsqu'il peut être appliqué, l'algorithme **SA4** s'avère plus efficace que l'algorithme **SA2**. Rappelons néanmoins que sa sortie est plus faible que celle de l'algorithme **SA2**. L'algorithme **SA5**, dont la sortie a les mêmes spécifications que celle de l'algorithme **SA2**, est lui aussi significativement plus efficace que l'algorithme **SA2**. Il est néanmoins apparu sur certains exemples que des progrès restent à faire pour la résolution réelles des systèmes d'équations et d'inégalités : les solutions proposées pour le calcul certifié de Représentations Univariées Rationnelles à coefficients infinitésimaux (voir chapitre 1 de cette partie) se sont montrées assez efficaces pour traiter les problèmes que nous devions résoudre, au point que c'est la dernière phase de l'algorithme – qui consiste à chercher un rationnel suffisamment petit pour remplacer l'infinitésimal – qui devient bloquante.

Chapitre 4 : Problème d'interpolation de Birkhoff

Soit f une fonction de $\mathbb{R}$ dans $\mathbb{R}$, on note $f^{(j)}$ sa j-ième dérivée partielle. Soit $x_1, \ldots, x_n$ (avec $x_1 < \ldots < x_n$) un ensemble ordonné de points réels tels que pour une famille $\mathcal{I}$ de couples $(i,j) \in \{1, \ldots, n\} \times \{0, \ldots, r\}$ les valeurs de $f^{(j)}(x_i) = f_{i,j}$ soient connues. Le problème qui consiste déterminer l'existence et à trouver un unique polynôme $P \in \mathbb{R}[X]$ de degré borné par r tel que, pour $(i,j) \in \mathcal{I}$, le polynôme P vérifie $P^{(j)}(x_i) = f_{i,j}$ s'appelle le problème d'interpolation de Birkhoff.

Pour n et r fixés, on peut se poser le problème de déterminer les familles $\mathcal{I}$ de couples de $\{1, \ldots, n\} \times \{0, \ldots, r\}$ pour lesquelles le problème d'interpolation de Birkhoff est résoluble. Il s'agit alors de fournir une base de données permettant de donner les conditions d'interpolation pour lesquelles le problème d'interpolation de Birkhoff est résoluble. Dans [48], l'auteur montre que ce problème est équivalent à celui qui consiste à décider si une hypersurface contient des points réels n'ayant aucune coordonnée nulle. Dans [80], l'auteur propose une résolution partielle du problème. Nous montrons comment les algorithmes proposés dans ce document permettent d'automatiser la résolution de ce problème.

Table des matières

Liste des tableaux

Index

Première partie

Préliminaires

Chapitre 1

Rappels de géométrie algébrique réelle

Résumé

Dans ce Chapitre, nous rappelons les définitions et propriétés des ensembles et fonctions semi-algébriques. Nous rappelons en particulier le principe de Tarski-Seidenberg ainsi que ses conséquences les plus importantes. Dans un deuxième temps, nous étudions les ensembles algébriques réels, cas particuliers des ensembles semi-algébriques, mais qui ne bénéficient pas des mêmes propriétés de stabilité. Nous rappelons aussi les notions de singularités, de points et de valeurs critiques, ainsi que le théorème de Sard. La rédaction de ce Chapitre est inspirée de [20, 18, 31].

Sommaire

1.1 Ensembles et fonctions semi-algébriques

Définition 1.1 *Soit R un corps ordonné. On dit que R est un corps réel clos si et seulement si $R[X]/\langle X^2+1\rangle$ est algébriquement clos.*

Le corps réel clos le plus connu est $\mathbb{R}$. Dans ce document, nous travaillerons aussi avec le corps réel clos des séries de Puiseux à coefficients réels. Soit ε un *infinitésimal*, c'est-à-dire un élément transcendant strictement positif et strictement inférieur à tout élément strictement positif de R.

Définition 1.2 *Le corps des séries de Puiseux à coefficients réels est l'ensemble des éléments qui s'écrivent*

$$\sum_{i \geq i_0} a_i \varepsilon^{i/q}$$

où $i \in \mathbb{Z}$, $q \in \mathbb{Q}$ et $a_i \in R$.

Soit $a \in R\langle\varepsilon\rangle$. On munit $R\langle\varepsilon\rangle$ de l'ordre suivant :

$$a > 0 \Longleftrightarrow a_{i_0} > 0.$$

Muni de cet ordre, le corps des séries de Puiseux est un corps réel clos.

Définition 1.3 *Un ensemble semi-algébrique de R^n est un sous-ensemble de R^n vérifiant une combinaison booléenne d'équations et d'inégalités polynomiales.*

De façon équivalente, les ensembles semi-algébriques de R^n forment la plus petite classe $\mathcal{C}$ de sous-ensembles de R^n qui vérifie :
- si $P \in R[X_1,\ldots,X_n]$ alors $\{M \in R^n \mid P(M) = 0\} \in \mathcal{C}$ et $\{M \in R^n \mid P(M) > 0\} \in \mathcal{C}$,
- si $A \in \mathcal{C}$ et $B \in \mathcal{C}$ alors $A \bigcup B$, $A \bigcap B$, $R^n \setminus A$ sont dans $\mathcal{C}$.

Proposition 1.1 *Tout ensemble semi-algébrique de R^n est réunion finie de sous-ensembles de la forme*

$$\{M \in R^n \mid P(M) = 0 \text{ et } Q_1(M) > 0 \text{ et } \ldots \text{ et } Q_\ell(M) > 0\}$$

pour $\ell \in \mathbb{N}$ et $P,Q_1,\ldots,Q_\ell \in R[X_1,\ldots,X_n]$.

L'ensemble des ensembles semi-algébriques de R^n possède des propriétés de stabilité (par unions et intersections finies, passage au complémentaire, image réciproque par une application polynomiale, produit cartésien). Une autre propriété, importante, de stabilité, est la stabilité par projection.

Théorème 1.1 *(Version géométrique – Tarski-Seidenberg) Soit $\mathcal{S}$ un ensemble semi-algébrique de R^n et Π la projection qui à tout point $(x_1,\ldots,x_n)$ associe $(x_1,\ldots,x_{n-1})$. Alors $\pi(\mathcal{S})$ est un ensemble semi-algébrique de R^{n-1}.*

Les conséquences de ce résultat sont multiples, en voici quelques-unes.

Corollaire 1.1

- *Soit $\mathcal{S}$ un ensemble semi-algébrique de R^{n+k}, son image par la projection qui à tout point $(x_1,\ldots,x_n,x_{n+1},\ldots,x_k)$ associe $(x_1,\ldots,x_n)$ est un ensemble semi-algébrique de R^n.*
- *Soit $\mathcal{S}$ un ensemble semi-algébrique de R^n et F une application polynomiale de R^n dans R^m. Alors, $F(A)$ est un ensemble semi-algébrique de R^m.*

Définition 1.4 *Formule du premier ordre.*

1. *Si $P \in R[X_1,\ldots,X_n]$ alors $P = 0$ et $P > 0$ sont des formules.*
2. *Si Φ et Ψ sont des formules, alors «Φ et Ψ», «Φ ou Ψ», «non Φ» sont des formules.*
3. *Si Φ est une formule, et X une variable réelle, alors $\exists X \Phi$ et $\forall X \Phi$ sont des formules.*

L'ensemble des formules ainsi obtenues s'appelle l'ensemble des formules du premier ordre à paramètres dans R. L'ensemble des formules obtenues au moyen de 1 et 2 s'appelle l'ensemble des formules sans quantificateur.

Un ensemble $\mathcal{S} \subset R^n$ est semi-algébrique si et seulement si il existe une formule sans quantificateur $\Phi(X_1,\ldots,X_n)$ telle que :

$$(x_1,\ldots,x_n) \in \mathcal{S} \Longleftrightarrow \Phi(X_1,\ldots,X_n).$$

Théorème 1.2 *(version logique – Tarski-Seidenberg) Si Φ est une formule du premier ordre dont les variables libres sont $(x_1,\ldots,x_n)$, l'ensemble des $(x_1,\ldots,x_n) \in R^n$ qui satisfont Φ est un sous-ensemble semi-algébrique de R^n.*

Définition 1.5 *Soit $A \subset R^m$ et $B \subset R^n$ deux ensembles semi-algébriques. Une fonction $f : A \to B$ est dite semi-algébrique si et seulement si son graphe*

$$\Gamma_f = \{(M,N) \in A \times B \mid N = f(M)\}$$

est un ensemble semi-algébrique de $R^m \times R^n$.

Des propriétés importantes découlent du théorème de Tarski-Seidenberg :

Corollaire 1.2

- *L'image d'un ensemble semi-algébrique par une fonction semi-algébrique est un ensemble semi-algébrique. De même pour l'image réciproque.*
- *La composée de deux fonctions semi-algébriques est une fonction semi-algébrique.*
- *Les fonctions semi-algébriques de A dans R forment un anneau.*

Proposition 1.2 *(Principe de transfert) Soit Φ une formule sans quantificateurs à paramètres dans R, et soit K une extension réelle close de R. Alors Φ est vraie dans R si et seulement si elle est vraie dans K.*

Soit $\mathcal{S}$ un ensemble semi-algébrique, l'adhérence de $\mathcal{S}$ (que l'on notera $\overline{\mathcal{S}}$) est l'ensemble des points :

$$\overline{\mathcal{S}} = \{x \in R^n \mid \forall t \in R \ \exists y \in \mathcal{S} \ \ (||y - x||^2 < t^2 \ \text{ ou } \ t = 0\}.$$

Lemme 1.1 *(Lemme de sélection des courbes) Soit $\mathcal{S} \in R^n$ un ensemble semi-algébrique. Soit $M \in \overline{\mathcal{S}}$ tel que $M \notin \mathcal{S}$. Alors il existe une fonction semi-algébrique continue γ : $[0,1] \to R^n$ telle que $\gamma(0) = M$ et $\gamma(]0,1]) \subset \mathcal{S}$.*

Proposition 1.3 *Soit γ :$]0,r] \to R$ une fonction semi-algébrique continue sur un intervalle $]0,r] \subset R$, bornée en valeur absolue. Alors γ se prolonge continûment en 0.*

Un corps réel clos n'est pas forcément connexe pour la topologie euclidienne. Par exemple, l'intersection de $]-\infty,\pi]$ avec le corps des nombres réels algébriques est un ouvert fermé du corps des nombres algébriques réels. En fait le seul corps réel clos qui soit connexe (pour la topologie euclidienne) est $\mathbb{R}$

Définition 1.6 *Un ensemble semi-algébrique $\mathcal{S}$ est dit semi-algébriquement connexe (ou semi-algébriquement connexe par arcs) quand pour tout M et tout N de $\mathcal{S}$, il existe une fonction semi-algébrique continue $\phi : [0,1] \to \mathcal{S}$ telle que $\phi(0) = M$ et $\phi(1) = N$.*

Proposition 1.4 *Le nombre de composantes semi-algébriquement connexes d'un ensemble semi-algébrique quelconque est fini.*

1.2 Ensembles algébriques réels

Soit A une partie de $K[X_1,\ldots,X_n]$. On note $V(A)$ l'ensemble

$$V(A) = \{M \in C^n \mid \forall P \in S \ \ P(M) = 0\}.$$

Une partie V de C^n est une K-variété algébrique de C^n si il existe une partie A de $K[X_1,\ldots,X_n]$ telle que $V = V(A)$ (lorsqu'il n'y a pas d'ambiguité sur le corps de base, on dira que V est une variété algébrique).

On définit de manière similaire les ensembles algébriques réels de R^n. Si A est une partie de $K[X_1,\ldots,X_n]$, on note $V_R(A)$ l'ensemble

$$V_R(A) = \{M \in R^n \mid \forall P \in A \ \ P(M) = 0\} = V(A) \bigcap R^n.$$

Si V est une partie de R^n, on note

$$\mathcal{I}(V) = \{P \in K[X_1,\ldots,X_n] \mid \forall M \in V \ \ P(M) = 0\}$$

Les ensembles algébriques de R^n sont les parties V de R^n telles que

$$V_R(\mathcal{I}(V)) = V.$$

On dira que V est une variété algébrique réelle irréductible si $\mathcal{I}(V)$ est irréductible.

Dans ce document, on entendra par «dimension de l'ensemble des solutions d'un système d'équations», la dimension de l'ensemble des solutions du système dans C^n.

Définition 1.7 *Soit V une variété algébrique de C^n irréductible, on pose $I = \mathcal{I}(V)$. La dimension de V est la dimension de $K[X_1,\ldots,X_n]/I$. La dimension d'une variété algébrique quelconque est le maximum des dimensions de ses composantes irréductibles.*

Définition 1.8 *Soit V une variété algébrique irréductible de dimension d définie par $P_1 = \ldots = P_k = 0$ où $(P_1,\ldots,P_k)$ sont des polynômes de $K[X_1,\ldots,X_n]$ tels que $\sqrt{\langle P_1,\ldots,P_k\rangle} = \langle P_1,\ldots,P_k\rangle$. L'ensemble des points singuliers de V est l'ensemble des points $M \in V$ tels que l'espace vectoriel $\mathrm{Vect}(\overrightarrow{grad}_M\ (P_1),\ldots,\ \overrightarrow{grad}_M\ (P_k))$ est de dimension strictement inférieure à $n-d$.*

Définition 1.9 *Soit V une variété algébrique définie par $P_1 = \ldots = P_k = 0$ où les polynômes $(P_1,\ldots,P_k)$ appartiennent à $K[X_1,\ldots,X_n]$ tels que*

$$\sqrt{\langle P_1,\ldots,P_k\rangle} = \langle P_1,\ldots,P_k\rangle$$

L'espace tangent de Zariski à V en $M \in V$ est l'espace vectoriel $T_M^Z(V)$:

$$T_M^Z(V) = \bigcap_{j=1}^{k}\{N = (\nu_1,\ldots,\nu_n) \in C^n \mid \frac{\partial P_j}{\partial X_1}_{(M)}\nu_1 + \ldots + \frac{\partial P_j}{\partial X_n}_{(M)}\nu_n = 0\}.$$

L'espace tangent de Zariski ne dépend pas du choix des générateurs de l'idéal $\langle P_1,\ldots,P_k\rangle$. Notons que l'espace tangent de Zariski est défini même en un point singulier de V.

Soit f une application régulière d'une variété algébrique V dans un espace vectoriel E. On dit qu'un point M de V est un point critique de f si et seulement si la différentielle de f en M est de rang strictement inférieur à la dimension de E. Une valeur critique de f est un point de E qui est la valeur prise par f en un point critique de f.

Théorème 1.3 *(Théorème de Sard) Avec les notations ci-dessus, l'ensemble des valeurs critiques de f est de dimension strictement inférieure à celle de E.*

Enfin, le résultat ci-dessous nous sera utile.

Lemme 1.2 *[20] La fonction distance est une fonction semi-algébriquement continue.*

Chapitre 2

Suite des sous-résultants

Résumé

Dans ce chapitre, nous rappelons la définition de la suite des sous-résultants associée à un couple de polynômes. Puis, nous rappelons comment on peut en déduire le degré du pgcd de ce couple ainsi que les propriétés de spécialisation de cette suite. Dans la dernière section, nous rappelons un algorithme permettant de calculer cette suite. Les rappels contenus dans ce chapitre seront utilisés dans le chapitre 3 de la partie I, ainsi que dans les chapitres 2 et 3 de la partie II. La rédaction de ce chapitre est inspirée de [31, 14].

Sommaire

Introduction

Dans ce chapitre, nous rappelons quelques notions relatives à la suite des sous-résultants associée à un couple de polynômes. La suite des sous-résultants est un outil fondamental en Calcul Formel dont le champ d'application est très vaste : en géométrie réelle, elle est utilisée pour compter et isoler le nombre de racines réelles d'un polynôme univarié [90, 57, 54, 87, 75], en théorie de Galois, elle est utilisée pour le calcul de résolvantes [94], etc.

Nous rappelons ci-dessous les propriétés de cette suite. Elle permet de déterminer le degré du pgcd de deux polynômes univariés en n'effectuant les calculs que dans l'anneau des coefficients des polynômes. Elle bénéficie aussi de bonnes propriétés de spécialisations [49, 50]. La première section de ce chapitre rappelle les définitions et notations nécessaires. Dans la deuxième section, nous rappelons les propriétés de la suite des sous-résultants dont nous allons nous servir dans la suite de ce document.

2.1 Définitions

Soit $P = a_pX^p + \ldots + a_0$ et $Q = b_qX^q + \ldots + b_0$ deux polynômes non nuls de degrés p et q dans $D[X]$ où D est un anneau intègre et euclidien. Considérons trois polynômes U, V et W tels que

$$\begin{aligned} U &= u_{q-\ell-1}X^{q-\ell-1} + \ldots + u_0, \\ V &= v_{p-\ell-1}X^{p-\ell-1} + \ldots + v_0, \\ W &= w_{p+q-\ell-1}X^{p+q-\ell-1} + \ldots + w_0. \end{aligned}$$

Si on considère les polynômes P,Q,U,V,W comme des vecteurs de $D[X]$ dans la base 1,X,$\ldots$,$X^{p+q-\ell}$, dire que $UP + VQ = W$ c'est écrire que les éléments de D

$$u_{q-\ell-1},\ldots,u_0,v_{p-\ell-1},\ldots,v_0,w_{p+q-\ell-1},\ldots,w_0$$

vérifient le système d'équations linéaires dont la matrice associée a pour transposée :

$$SH_\ell(P,Q) = \begin{pmatrix} a_p & \ldots & \ldots & \ldots & \ldots & a_0 & & \\ & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \\ & & a_p & \ldots & \ldots & \ldots & \ldots & a_0 \\ & & & b_q & \ldots & \ldots & \ldots & b_0 \\ & & b_q & \ldots & \ldots & \ldots & b_0 & \\ & \ldots & \ldots & \ldots & \ldots & \ldots & & \\ b_q & \ldots & \ldots & \ldots & b_0 & & & \end{pmatrix}$$

est une matrice extraite de la matrice de Sylvester (voir [68] pour plus de détails) associée à P et Q. Aussi dire qu'il existe U et V deux polynômes non identiquement nuls de degrés respectifs au plus $q-\ell-1$ et $p-\ell-1$ tels que $PU+QV$ soit de degré strictement inférieur à ℓ, c'est dire que le déterminant de la matrice carrée obtenue en prenant les $p+q-2\ell$ premières colonnes de $SH_\ell(P,Q)$ est nul.

Ce déterminant, que l'on note $\mathrm{sr}_\ell(P,Q)$ est le ℓ-ième **coefficient sous-résultant signé** associé à P et Q. On appelle **résultant** de P et Q (et on note $\mathrm{Res}(P,Q)$) le dernier coefficient sous-résultant $\mathrm{sr}_0(P,Q)$. On appelle ℓ-ième polynôme sous-résultant le polynôme $\sum_{i=0}^{\ell} \mathrm{sr}_{\ell,i} X^i$ où $\mathrm{sr}_{\ell,i}$ est le déterminant de la matrice carrée obtenue en prenant les $p+q-2\ell-1$ premières colonnes et la $p+q-3\ell+i$-ième colonne de $SH_\ell(P,Q)$. Le coefficient dominant du ℓ-ième polynôme sous-résultant est donc le ℓ-ième coefficient sous-résultant.

2.2 Propriétés

Les preuves des résultats ci-dessous se trouvent dans [14].

Proposition 2.1 *Si on désigne par* $\gcd(P,Q)$ *le pgcd des polynômes* P *et* Q *de* $D[X]$*, alors* $\deg(\gcd(P,Q)) > \ell$ *si et seulement si*

$$\mathrm{sr}_0(P,Q) = \ldots = \mathrm{sr}_\ell(P,Q) = 0.$$

On en déduit le corollaire suivant :

Corollaire 2.1 $\mathrm{Res}(P,Q) = 0$ *si et seulement si* P *et* Q *ont un facteur commun.*

Enfin, puisque les coefficients sous-résultants sont des déterminants, on a :

Proposition 2.2 *Si* P *est un polynôme unitaire et si* $f : D' \to D$ *est un homomorphisme d'anneau, alors*

$$\mathrm{Res}(f(P),f(Q)) = f(\mathrm{Res}(P,Q)).$$

Une étude plus complète des propriétés de spécialisation de la suite des sous-résultants se trouve dans [49, 50].

Chapitre 3

Décomposition Cylindrique Algébrique

Résumé

Les premiers algorithmes permettant de décider si une formule du premier ordre est vraie se trouvent dans les travaux de Tarski [91] et Seidenberg [89]. La complexité de ceux-ci n'était pas élémentairement récursive. L'algorithme de Décomposition Cylindrique Algébrique de Collins [28] a une bien meilleure complexité et est l'algorithme le plus connu pour décider si une formule du premier ordre est vraie. Dans ce chapitre, nous rappelons la définition de Décomposition Cylindrique Algébrique adaptée à une famille de polynômes. Nous en donnons les propriétés essentielles. Puis, nous montrons comment on peut en déduire un procédé algorithmique permettant de calculer cet objet. La sortie de l'algorithme obtenu est un ensemble de points déterminant toutes les conditions de signe possibles vérifiées par la famille de polynômes donnée en entrée. Enfin, nous évaluons la complexité en terme d'opérations arithmétiques élémentaires. Pour rédiger ce chapitre, nous nous sommes inspirés d'une version préliminaire de [14] et de [31].

Sommaire

Introduction

Contrairement aux premiers algorithmes de Tarski et Seidenberg [91, 89], l'algorithme de décomposition cylindrique algébrique dû à Collins (voir [28]) a une complexité élémentairement récursive puisqu'il est doublement exponentiel en le nombre de variables. Cet algorithme est aussi le premier qui fut implanté et qui permettait de décider si une formule du premier ordre est vraie. De nombreuses optimisations ont été apportées depuis (voir [29, 70] par exemple). Cet algorithme est à notre connaissance le plus répandu et le plus utilisé pour décider si un ensemble semi-algébrique est vide.

Dans ce chapitre, nous rappelons les résultats sur lesquels sont fondés l'algorithme de Collins, et nous le décrivons. Basé sur l'élimination des variables les unes après les autres, cet algorithme construit récursivement des familles de polynômes dont on pourra déduire une *décomposition cylindrique algébrique adaptée à la famille de départ*. Dans la première section de ce chapitre, nous rappelons les définitions et résultats qui permettront de déterminer comment ces familles de polynômes doivent être construites. Dans la deuxième section de ce chapitre, nous présentons l'algorithme de Collins. Il se divise en deux étapes : l'étape de **projection** qui calcule les familles de polynômes en éliminant les variables les unes après les autres, puis l'étape de **remontée** qui va permettre de construire la décomposition cylindrique algébrique. Dans la dernière section, nous analysons la complexité théorique de cet algorithme qui est doublement exponentielle en le nombre de variables.
La rédaction de ce chapitre est inspirée d'une version préliminaire de [14] et de [31] qui contiennent les preuves des résultats exposés. Dans la suite, R est un corps réel clos et C est sa clôture algébrique.

3.1 Premiers résultats

Une *décomposition* d'un ensemble semi-algébrique S est une partition finie de S en sous-ensembles semi-algébriques. Une *décomposition cylindrique algébrique* de R^n est une suite $\mathcal{S}_1, \ldots, \mathcal{S}_n$ telle que pour tout $1 \leq i \leq n$, $\mathcal{S}_i$ est une décomposition de R^i en sous-ensembles semi-algébriques connexes (que nous appellerons *cellules*), ayant les propriétés suivantes :

- a) Toute cellule $S \in \mathcal{S}_1$ est soit un point soit un intervalle ouvert.
- b) Pour tout $1 \leq i \leq n$ et toute cellule $S \in \mathcal{S}_i$, il existe un nombre fini de fonctions semi-algébriques continues

$$\xi_{S,1} < \ldots < \xi_{S,\ell_S} \;:\; S \longrightarrow R$$

telles que le cylindre $S \times R$ est l'union disjointe de cellules de $\mathcal{S}_{i+1}$ qui sont :

- soit le graphe $\Gamma_{S,j}$ d'une des fonctions $\xi_{S,j}$ pour $j \in \{1, \ldots, \ell_S\}$:

$$\Gamma_{S,j} = \{(x', x_{j+1}) \in S \times R \mid x_{j+1} = \xi_{S,j}(x')\}$$

– soit une *bande* $B_{S,j}$ du cylindre borné par les graphes des fonctions $\xi_{S,j}$ et $\xi_{S,j+1}$ pour $j \in \{0, \ldots, \ell_S\}$, où on prend par convention $\xi_{S,0} = -\infty$ et $\xi_{i,\ell_S+1} = +\infty$:

$$B_{S,j} = \{(x', x_{j+1}) \in S \times R \mid \xi_{S,j}(x') < x_{j+1} < \xi_{S,j+1}(x')\}$$

Proposition 3.1 *Toute cellule d'une décomposition cylindrique algébrique est semi-algébriquement homéomorphe à un hypercube ouvert* $]0,1[^i$ *(par convention* $]0,1[^0$ *est un point).*

Etant donnée une famille de polynômes $\mathcal{P}$ dans $K[X_1, \ldots, X_n]$, un sous-ensemble S de R^n est dit $\mathcal{P}$-invariant si tout polynôme $P \in \mathcal{P}$ est de signe constant sur S. Dans la suite de ce chapitre nous allons montrer comment construire une décomposition cylindrique algébrique $\mathcal{S}_n$ de R^n adaptée à $\mathcal{P}$, c'est-à-dire pour laquelle chaque cellule $S \in \mathcal{S}_n$ est $\mathcal{P}$-invariante.

Soit S un ensemble semi-algébrique. Une décomposition cylindrique algébrique adaptée à S est une décomposition cylindrique algébrique de R^n telle que S est une union finie de cellules de cette décomposition. Il est clair que si $\mathcal{P}$ est une famille de polynômes telle que S est la réalisation d'une formule sans quantificateurs avec atomes dans $\mathcal{P}$, une décomposition cylindrique algébrique adaptée à $\mathcal{P}$ est une décomposition cylindrique algébrique adaptée à S.

Puisque nous voulons construire une décomposition cylindrique algébrique adaptée à $\mathcal{P}$, il est utile que pour $S \in \mathcal{S}_{n-1}$ nous puissions choisir chaque fonction $\xi_{S,j}$ de S dans R comme étant une fonction qui à $(x_1, \ldots, x_{n-1}) \in S$ associe une racine de $P \in \mathcal{P}$. Pour cela, il faut que les racines (complexes et réelles) d'un polynôme univarié P varient continûment en fonction des coefficients de P.
Les résultats qui suivent permettent de démontrer le théorème 3.2 (voir [31, 14]). Ce théorème nous dit quelles conditions doit vérifier une cellule $S \in \mathcal{S}_{n-1}$ pour pouvoir choisir chaque fonction $\xi_{S,j}$ de S dans R comme étant une fonction qui à $(x_1, \ldots, x_{n-1}) \in S$ associe une racine de $P \in \mathcal{P}$.

Proposition 3.2 *Soit* $P = a_p X^p + \ldots + a_1 + a_0 \in C[X]$, *avec* $a_p \neq 0$. *Si* $x \in C$ *est une racine de* P *alors*

$$|x| \leq \max_{i=1,\ldots,p}(p \left|\frac{a_{p-i}}{a_p}\right|^{1/i}) = M$$

Dans le lemme suivant, on identifie tout polynôme unitaire $X^p + a_{p-1}X^{p-1} + \ldots + a_0 \in C[X]$ de degré p au point $(a_{p-1}, \ldots, a_0) \in C^p$. On note $D(z,r) = \{\omega \in C \mid |\omega - z| < r\}$ le disque ouvert centré en z de rayon r.

Lemme 3.1 *Etant donné* $r > 0$ *quelconque, il existe un voisinage ouvert* U *de* $(X - z)^\mu$ *dans* C^μ *tel que tout polynôme unitaire dans* U *a ses racines dans* $D(z,r)$.

Lemme 3.2 *Soit* $P^0 \in C^{q+r}$ *un polynôme unitaire tel que* $P^0 = Q^0 R^0$, *où* Q^0 *et* R^0 *sont premiers entre eux, unitaires et de degrés respectifs* q *et* r. *Pour tous voisinages ouverts* V *et* W *de* Q^0 *et* R^0, *il existe un voisinage ouvert* U *de* P^0 *dans* C^{q+r}, *tel que* $\forall P \in U$, P *s'écrit de manière unique comme produit de deux polynômes* $Q \in V$ *et* $R \in W$ *de degrés respectifs* q *et* r.

Théorème 3.1 *Soit $P \in R[X_1,\ldots,X_n]$ et S un ensemble semi-algébrique de R^{n-1} tel que $P(x',X_n)$ est de degré constant $\forall x' \in S$. Soit $a' \in S$. On note $z_1,\ldots,z_j$ les racines distinctes de $P(a',X_n)$ dans C de multiplicités respectives $\mu_1,\ldots,\mu_j$. Pour tout $r > 0$ tel que les disques ouverts $D(z_i,r) \subset C$ sont disjoints, il existe un voisinage ouvert V de a' tel que $\forall x' \in V \cap S$, le polynôme $P(x',X_n)$ a exactement μ_i racines comptées avec multiplicités dans le disque $D(z_i,r)$ pour $i = 1,\ldots,j$.*

Le corollaire suivant est une conséquence du théorème ci-dessus.

Corollaire 3.1 *Soit $P \in R[X_1,\ldots,X_n]$ et S un ensemble semi-algébrique de R^{n-1} tel que $\forall x' \in S$:*

- *$P(x',X_n)$ est de degré constant,*
- *le nombre de racines distinctes dans C de $P(x',X_n)$ est constant.*

Si S est un ensemble semi-algébriquement connexe, alors le nombre de racines réelles distinctes de $P(x',X_n)$ (pour tout $x' \in S$) est constant.

On peut maintenant énoncer :

Théorème 3.2 *Soit $P \in R[X_1,\ldots,X_n]$ et S une composante semi-algébriquement connexe de R^{n-1} tel que*

- *$\forall x' \in S$, le nombre de racines distinctes de $P(x',X_n)$ dans C est constant,*
- *$\forall x' \in S$, le degré de $P(x',X_n)$ est constant.*

Alors, il existe ℓ fonctions semi-algébriques continues $\xi_1,\ldots,\xi_\ell : S \to R$ telles que $\forall x' \in S$, l'ensemble des racines réelles de $P(x',X_n)$ est exactement $\{\xi_1(x'),\ldots,\xi_\ell(x')\}$. De plus, pour $i = 1,\ldots,\ell$, la multiplicité de $\xi_i(x')$ est constante pour $x' \in S$.

'La proposition ci-dessous généralise le théorème ci-dessus au cas d'une famille de deux polynômes. le cas de deux polynômes.

Proposition 3.3 *Soit P et Q deux polynômes dans $R[X_1,\ldots,X_n]$ et S un sous-ensemble semi-algébriquement connexe de R^{n-1} tels que $\forall x' \in S$:*

- *les degrés de $P(x',X_n)$ et de $Q(x',X_n)$ restent constants,*
- *les nombres de racines distinctes dans C de $P(x',X_n)$ et de $Q(x',X_n)$ restent constants,*
- *le degré du* gcd *de $P(x',X_n)$ et de $Q(x',X_n)$ reste constant.*

Alors, il existe ℓ fonctions semi-algébriques continues $\xi_1 < \ldots < \xi_\ell$ telles que pour tout $x' \in S$, l'ensemble des racines réelles de $PQ(x',X_n)$ est exactement $\{\xi_1(x'),\ldots,\xi_\ell(x')\}$. De plus, pour $i = 1,\ldots,\ell$, la multiplicité de $\xi_i(x')$, racine de $P(x',X_n)$ (resp. $Q(x',X_n)$) est constante pour tout $x' \in S$.

3.2 L'algorithme de Collins

L'algorithme de Décomposition cylindrique algébrique se divise en deux étapes : l'étape de projection, et l'étape de remontée.

Etape de projection :

Soit P un polynôme de $R[X_1,\ldots,X_n]$ vu comme un polynôme univarié en X_n à coefficients polynomiaux dans $R[X_1,\ldots,X_{n-1}]$. On note $\mathrm{lcoeff}(P)$ le coefficient dominant de P. On note $\mathrm{Tr}(P)$ le polynôme égal à $P - \mathrm{lcoeff}(P)X_n^{\deg(P,X_n)}$.
On définit un «opérateur» de projection $\mathrm{PROJ}(\mathcal{P})$ comme étant le plus petit ensemble de polynômes de $R[X_1,\ldots,X_{n-1}]$ tel que :

- Si $P \in \mathcal{P}$ et $\deg(P,X_n) = p \geq 2$, $\mathrm{PROJ}(\mathcal{P})$ contient tous les coefficients sous-résultants non constants $\mathrm{sr}_j(P,\partial P/\partial X_n)$, pour $j \in \{0,\ldots,p\}$,
- Si $(P,Q) \in \mathcal{P}^2$, $\mathrm{PROJ}(\mathcal{P})$ contient tous les coefficients sous-résultants non constants $\mathrm{sr}_j(P,Q)$ pour $j \in \{0,\ldots,\min(\deg(P,X_n),\deg(Q,X_n))\}$,
- Si $P \in \mathcal{P}$, $\deg(P,X_n) \geq 1$ et $\mathrm{lcoeff}(P)$ est non constant alors $\mathrm{PROJ}(\mathcal{P})$ contient $\mathrm{lcoeff}(P)$ et $\mathrm{PROJ}(\mathcal{P} \setminus \{P\} \cup \{\mathrm{Tr}(P)\})$,
- Si $P \in \mathcal{P}$, $\deg(P,X_n) = 0$ et P est non constant, alors $\mathrm{PROJ}(\mathcal{P})$ contient P.

Le théorème suivant est une conséquence des résultats précédemment démontrés.

Théorème 3.3 *Soit $\mathcal{P}$ une famille finie de polynômes dans $R[X_1,\ldots,X_n]$ et soit S une composante semi-algébriquement connexe d'un sous-ensemble semi-algébrique de R^{n-1}, qui est* $\mathrm{PROJ}(\mathcal{P})$*-invariant. Alors, il existe ℓ fonctions continues $\xi_1 < \ldots < \xi_\ell : S \to R$ telles que $\forall x' \in S$, l'ensemble de points $\{\xi_1(x'),\ldots,\xi_\ell(x')\}$ est exactement l'ensemble des racines de réelles de tous les polynômes non nuls $P(x',X_n)$ avec $P \in \mathcal{P}$. Le graphe de chaque fonction ξ_i ainsi que chaque bande du cylindre $S \times R$ borné par ces graphes, sont des ensembles semi-algébriques semi-algébriquements connexes, et semi-algébriquements homéomorphes soit à S soit à $S\times]0,1[$, et $\mathcal{P}$-invariants.*

Ainsi, étant donnée une décomposition cylindrique algébrique de R^{n-1} adaptée à $\mathrm{PROJ}(\mathcal{P})$ et S vérifiant les hypothèses du théorème précédent, les cellules de cette décomposition cylindrique algébrique, on voit alors qu'il existe une décomposition cylindrique algébrique de R^n adaptée à $\mathcal{P}$.
On définit récursivement des sous-ensembles finis de polynômes $\mathcal{P}_i$ tels que :

- $\mathcal{P}_n = \mathcal{P}$,
- Pour tout $i \in \{1,\ldots,n-1\}$, $\mathcal{P}_i = \mathrm{PROJ}(\mathcal{P}_{i+1})$.

Etape de remontée :

Il est alors clair que $\mathcal{P}_1$ est une famille de polynômes univariés. La construction d'une décomposition cylindrique algébrique $\mathcal{S}_1$ adaptée à $\mathcal{P}_1$ se fait en donnant un point représentatif dans chaque composante semi-algébriquement connexe de $\mathcal{S}_1$. Ceci revient à isoler, puis trier les racines réelles des polynômes de $\mathcal{P}_1$: $\{\alpha_{1,1},\ldots,\alpha_{1,s_1}\}$ et donner un point dans chaque intervalle $]\alpha_{1,i},\alpha_{1,i+1}[$ pour $i \in \{0,\ldots,s_1+1\}$ (avec $\alpha_{1,0} = -\infty$ et $\alpha_{1,s_1+1} = +\infty$). Il nous faut maintenant montrer comment construire une décomposition cylindrique algébrique adaptée à $\mathcal{P}_2$ à partir de $\mathcal{S}_1$.

Tous les polynômes de $\mathcal{P}_2$ sont des polynômes en les variables X_1,X_2. Lorsque l'on spécialise la variable X_1 aux nombres algébriques réels de $\mathcal{S}_1$ on obtient une famille de polynômes univariés en X_2 à coefficients dans R. On peut donc réitérer le processus d'isolation et de tri de cette famille de polynômes. Ainsi, on construit *récursivement* une décomposition cylindrique algébrique adaptée à $\mathcal{P}_{i+1}$ à partir d'une décomposition cylindrique algébrique adaptée à $\mathcal{P}_i$. Ceci constitue la phase de *remontée* de l'algorithme de décomposition cylindrique algébrique.

L'algorithme de décomposition cylindrique de Collins est alors la succession des deux routines décrites ci-dessus.

Remarque 3.1 *La phase de remontée, telle que nous la décrivons, nécessite la manipulation symbolique de nombres algébriques réels. Ceci ne se fait pas de manière simple : de telles manipulations nécessitent de coder les nombres algébriques réels par des conditions d'annulation sur des polynômes ainsi qu'un intervalle d'isolation des racines. Il est important de noter qu'au cours de l'étape de remontée, il est nécessaire de travailler au-dessus de tours d'extensions algébriques lorsque les racines d'un polynôme ne sont pas rationnelles. Ceci induit des calculs et des méthodes délicats à mettre en oeuvre et que nous ne détaillons pas (voir [75, 76, 67]).*

On obtient le théorème suivant :

Théorème 3.4 *Pour toute famille finie $\mathcal{P}$ de polynômes dans $R[X_1,\ldots,X_n]$, il existe une décomposition cylindrique algébrique de R^n adaptée à $\mathcal{P}$.*

Une décomposition cylindrique algébrique $\mathcal{S}$ adaptée à une famille de polynômes $\mathcal{P}$ est donnée par une liste de points représentant chaque cellule de $\mathcal{S}$. On a la propriété suivante :

Théorème 3.5 *Soit $\mathcal{P}$ une famille de polynômes dans $R[X_1,\ldots,X_n]$ et $\mathcal{S}$ une décomposition cylindrique algébrique adaptée à $\mathcal{P}$. Pour toute condition de signe σ vérifiée par $\mathcal{P}$, on note D_σ une composante semi-algébriquement connexe du lieu des points vérifiant σ. Il existe au moins une cellule de $\mathcal{S}$ telle que tout point de cette cellule est contenue dans D_σ.*

Dans [70], l'auteur optimise l'opérateur de projection lorsque le nombre de variables est inférieur à 3. Etant donnée une famille de polynômes $\mathcal{P}$ (dont chacun des éléments est vu comme un polynôme univarié en une variable X) il démontre qu'il suffit de calculer :

- les coefficients des polynômes de $\mathcal{P}$ vus comme des polynômes univariés en X,
- les discriminants de chacun des polynômes de $\mathcal{P}$ par rapport à X,
- les résultants de chaque couple $(P,Q) \in \mathcal{P}^2$ par rapport à X (avec $P \neq Q$)

La validité de cette optimisation est démontrée uniquement lorsque le nombre de variables est inférieur ou égal à trois.

3.3 Complexité théorique

Soit d le maximum des degrés totaux des polynômes de $\mathcal{P}$. On note m le nombre de polynômes dans $\mathcal{P}$. Par ailleurs, on suppose que la multiplication de polynômes univariés dont le degré est borné par d se fait en d^2 opérations. Dans PROJ($\mathcal{P}$), on a :

- $m(d-1)$ polynômes de degré maximal $O(d^2)$ correspondant aux coefficients sous-résultants des polynômes de $\mathcal{P}$ et de leur dérivée par rapport à une variable.
- $\binom{m}{2}(d-1)$ polynômes de degré maximal $O(d^2)$ correspondant aux coefficients sous-résultants de chaque couple de polynômes dans $\mathcal{P}$.
- $m(d+1)$ polynômes de degré maximal $O(d)$ correspondant aux coefficients de chaque polynôme de $\mathcal{P}$.

Donc PROJ($\mathcal{P}$) contient $O(m^2d)$ polynômes de degré maximal $O(d^2)$. Par ailleurs, comme le calcul des coefficients sous-résultants se fait en $O(d^2)$ opérations dans $K[X_1,\ldots,X_{n-1}]$, ces calculs se font en $O(d^{2n})$ opérations arithmétiques dans K. Donc le calcul de PROJ($\mathcal{P}$) se fait en $O(m^2d^{2n})$ opérations arithmétiques dans K.

En itérant ce processus, on s'aperçoit que lorsqu'on a éliminé $i-1$ variables, la i-ième itération de PROJ($\mathcal{P}$) se fait en $O(m^{2^i}d^{n2^i})$ opérations arithmétiques dans K. Il est alors aisé de constater que l'opérateur de projection de l'algorithme de Décomposition Cylindrique Algébrique s'effectue en au moins $O(m^{2^n}d^{n2^n})$ opérations arithmétiques dans K. Nous obtenons donc une complexité théorique doublement exponentielle en le nombre de variables similaire à celle obtenue dans [28].

D'un point de vue pratique, l'algorithme de Collins subit cette complexité à deux niveaux :

- lors de la phase de projection, le nombre de polynômes ainsi que leur degré devient une étape bloquante de l'algorithme lorsque le nombre de variables est supérieur à trois.
- lors de la phase de remontée, la gestion des nombres algébriques réels est cruciale. Les résultats de [75, 76] sont une avancée notable dans ce domaine, mais le problème qui reste à résoudre réside dans le nombre de points retournés qui est trop grand.

Ainsi, il arrive très souvent que pour des problèmes de plus de trois variables la phase de projection ne passe pas. Même lorsque celle-ci passe, la phase de remontée est aussi bloquante du fait du nombre de points réels qui doivent être manipulés. Dans la partie II, nous analyserons plus en détail le comportement pratique de l'Algorithme de Décomposition Cylindrique Algébrique.

Chapitre 4

Résolution des systèmes zéro-dimensionnels

Résumé

Dans le chapitre 6 de la partie I et les algorithmes proposés dans la partie II, nous ramènerons l'étude de systèmes algébriques de dimension positive à l'étude de systèmes zéro-dimensionnels. Dans ce chapitre, nous décrivons les outils utilisés pour l'étude de ces systèmes. Dans la première section, nous rappelons les définitions et propriétés élémentaires des bases de Gröbner. Dans le cas zéro-dimensionnel, une base de Gröbner permet de calculer une table de multiplication dans l'anneau des polynômes quotienté par l'idéal engendré par le système de départ. A partir de cette table, une Représentation Univariée Rationnelle – qui est la réécriture de l'ensemble des solutions sous une forme agréable et plus exploitable – peut être calculée. Nous en rappelons la définition ainsi que quelques propriétés dans la deuxième section.

Sommaire

Introduction

Dans le cas des systèmes d'équations polynomiales de dimension positive, une des alternatives à la Décomposition Cylindrique Algébrique est l'ensemble d'algorithmes que l'on peut regrouper sous le nom de *méthode des points critiques* (voir chapitre 6). Ces méthodes ramènent l'étude des ensembles semi-algébriques à l'étude de systèmes d'équations polynomiales zéro-dimensionnels à coefficients dans des corps de séries de Puiseux. De nombreuses méthodes spécifiques existent pour résoudre de tels systèmes : (bases de Gröbner [32], Représentations Univariées Rationnelles [4, 80, 81], Résolutions Géométriques [62, 43, 47], ensembles triangulaires [65, 60], entre autres méthodes). Nous verrons que, dans notre cadre de résolution, le calcul de Représentations Univariées Rationnelles [81] – qui revient à réécrire l'ensemble des solutions sous une forme «agréable» et plus exploitable – s'avère particulièrement bien adapté.

Un première indication de ce que devrait être la définition des Représentations Univariées Rationnelles se trouve dans [62]. Plusieurs travaux ont alors succédé (voir [74, 4, 26, 87, 13] entre autres) dans un cadre de réécriture ainsi que [43, 46, 45, 44] entre autres dans un cadre d'évaluation. Les travaux présentés dans [80, 81] et [47] décrivent les algorithmes les plus efficaces du moment dans leurs cadres respectifs. Dans ce chapitre, comme dans le reste du document, nous choisissons de rester dans le cadre de réécriture car c'est le seul qui a permis, à ce jour, d'aboutir à des algorithmes efficaces en pratique non probabilistes, et calculant des Représentations Univariées Rationnelles sans restriction sur le système d'équations polynomiales de départ.

Le point de départ d'un tel calcul est la table de multiplication de l'anneau des polynômes quotienté par l'idéal engendré par les équations du système polynomial. Cette table de multiplication peut être obtenue à partir d'une base de Gröbner de l'idéal. Dans la première section de ce chapitre, nous rappelons les définitions ainsi que quelques propriétés relatives aux bases de Gröbner. Puis, dans la deuxième section, nous rappelons les définitions et propriétés de la Représentation Univariée Rationnelle et mettons en évidence les problèmes que posent le calcul d'un tel objet.

4.1 Bases de Gröbner

Définition 4.1 *Un ordre admissible sur les monômes (unitaires) de* $K[X_1,\ldots,X_n]$ *est une relation «*$>$*» binaire sur* $\mathbb{N}^n$ *telle que :*

1. $>$ *est une relation d'ordre total sur* $\mathbb{N}^n$*,*
2. *si* $\alpha > \beta$ *et* $\gamma \in \mathbb{N}^n$*, alors* $\alpha + \gamma > \beta + \gamma$*,*
3. *pour l'ordre* $>$*, tout ensemble non vide admet un plus petit élément sur* $\mathbb{N}^n$*.*

En pratique, nous utiliserons plusieurs ordres sur les monômes, et en particulier l'ordre du Degré Lexicographique Inverse (que nous noterons dans la suite du document DRL, pour Degree Reverse Lexicographic) et l'ordre lexicographique. Ces deux ordres peuvent être combinés sur des blocs de variables : on obtient un ordre d'élimination (voir [17]).

Définition 4.2 *Soit* $p = \sum_\alpha c_\alpha x^\alpha \in K[X_1,\ldots,X_n]$ *et* $>$ *un ordre sur les monômes de*

$K[X_1, \ldots, X_n]$.

1. *le multi-degré de p est :*

$$\text{multideg}(p) = \max\{\alpha \in \mathbb{N}^n \mid c_\alpha \neq 0\}$$

2. *le coefficient de plus haut degré de p est :*

$$\text{lc}(p) = c_{\text{multideg}(p)} \in K$$

3. *le monôme de plus haut degré de p est :*

$$\text{lm}(p) = x^{\text{multideg}(p)}$$

4. *le terme initial de p est :*

$$\text{in}(p) = \text{lc}(p)\text{lm}(p)$$

Dans la suite de cette section, on suppose fixé un ordre admissible sur les monômes de $K[X_1, \ldots, X_n]$.

Théorème 4.1 *Si $F = \{f_1, \ldots, f_s\}$ est une famille de polynômes de $K[X_1, \ldots, X_n]$, alors tout $f \in K[X_1, \ldots, X_n]$ peut s'écrire :*

$$f = a_1 f_1 + \ldots + a_s f_s + r$$

où $\forall i \in \{1, \ldots, s\}$, a_i et r sont des polynômes de $K[X_1, \ldots, X_n]$ tels qu'aucun des monômes de r ne soit divisible par l'un des $\text{in}(f_i)$.

On appellera r *une forme normale* de f modulo F. En fait, la preuve de ce théorème peut être établie en exhibant un algorithme calculant le reste d'un polynôme modulo une liste ordonnée. C'est l'algorithme de forme normale. Cet algorithme a fait l'objet de nombreuses optimisations. La sortie de cet algorithme dépend de l'ordre dans lequel les polynômes interviennent dans l'algorithme de réduction. Elle n'est donc pas canonique.

Définition 4.3 *Une famille génératrice finie $G = (g_1, \ldots, g_s)$ d'éléments d'un idéal $\mathcal{I}$ de $K[X_1, \ldots, X_n]$ est une base de Gröbner si :*

$$\langle \text{in}(g_1), \ldots, \text{in}(g_s) \rangle = \langle \text{in}(\mathcal{I}) \rangle$$

La principale propriété des bases de Gröbner peut être résumée par :

Proposition 4.1 *On pose $G = (g_1, \ldots, g_s)$ une base de Gröbner d'un idéal $\mathcal{I}$ dans $K[X_1, \ldots, X_n]$. Pour tout polynôme f de $K[X_1, \ldots, X_n]$, le reste de f modulo G (ou la forme normale de f modulo G) est déterminé de manière unique. En particulier, f est un élément de $\mathcal{I}$ si et seulement si son reste modulo G est nul.*

Le réduit d'un polynôme f par rapport à une base de Gröbner G est appelé forme normale de f modulo G.

Définition 4.4 *Une base de Gröbner totalement réduite pour un idéal $\mathcal{I}$ dans l'anneau $K[X_1, \ldots, X_n]$ est une base de Gröbner pour $\mathcal{I}$ dont les polynômes sont constitués de monômes irréductibles modulo $\mathcal{I}$.*

Les bases de Gröbner ont aussi de bonnes propriétés de spécialisation. On considère un ordre $<$ d'élimination de $X_{i+1},\ldots,X_n$ défini sur les monômes de $K[X_1,\ldots,X_n]$.

Définition 4.5 *On dit que Φ est une spécialisation adaptée à un ordre d'élimination $<$ si Φ est un homéomorphisme de $K[X_1,\ldots,X_i]$ dans K défini par l'image $(x_1,\ldots,x_i)$ de $(X_1,\ldots,X_i)$ par Φ.*

Le résultat suivant est issu de [42].

Proposition 4.2 *[42] Soit G une base de Gröbner d'un idéal $\mathcal{I}$ de $K[X_1,\ldots,X_n]$ pour un ordre d'élimination de $X_{i+1},\ldots,X_n$. Si l'image des coefficients de plus haut degré de G par Φ n'est jamais nulle, alors $\Phi(G)$ est une base de Gröbner de $\mathcal{I}$.*

Preuve : D'après les hypothèses, $\mathrm{in}(\Phi(G)) = \Phi(\mathrm{in}(G))$. Puisque G est une base de Gröbner on a $\langle \mathrm{in}(G)\rangle = \mathrm{in}(\langle G\rangle)$. De plus, il est clair que $\langle \Phi(G)\rangle = \Phi(\langle G\rangle)$. Donc le résultat est prouvé. ∎

Notons que $\Phi(G)$ n'est pas forcément une base de Gröbner réduite.

D'après [69], les bases de Gröbner sont de taille doublement exponentielles en le nombre de variables dans le pire des cas. En pratique, cette complexité n'est vérifiée que dans des cas pathologiques, et l'algorithme de Buchberger [23] a été amélioré et a fait l'objet de nombreuses implantations dont les plus efficaces permettent de résoudre des applications.

Récemment, J.-C. Faugère a mis au point de nouveaux algorithmes calculant des bases de Gröbner et permettant de gagner plusieurs ordres de grandeur par rapport à l'algorithme de Buchberger (voir [37, 1]).

4.2 Représentation Univariée Rationnelle

Si S est zéro-dimensionnel et $I = \langle S\rangle$, l'algèbre-quotient $K[X_1,\ldots,X_n]/I$ est un K-espace vectoriel dont la dimension est égale au nombre de solutions de S comptées avec multiplicités dans C^n. Il est possible de calculer ce nombre à partir d'une base de Gröbner : il est égal au nombre de monômes «sous l'escalier» (c'est-à-dire le nombre de monômes ne pouvant être réduits à zéro modulo G). La dimension du K-espace vectoriel $K[X_1,\ldots,X_n]/\sqrt{I}$ (qui est égal au nombre de solutions de S dans C^n) peut être calculée à partir de G en construisant et en réduisant la forme quadratique de Hermite [16] (voir aussi [80, 51, 16]).
Soit S un système d'équations polynomiales dans $K[X_1,\ldots,X_n]$, pour toute solution $x = (x_1,\ldots,x_n) \in V(S)$, on note $\mu(x)$ la multiplicité de x.

Proposition 4.3 *[81] Etant donné $u \in K[X_1,\ldots,X_n]$ on définit :*
- $f_u(T) = \prod_{x\in V(S)} (T - u(x))^{\mu(x)}$
- $g_0(T) = \sum_{x\in V(S)} \mu(x) \prod_{y\in V(S), u(y)\neq u(x)} (T - u(y))$

– $g_i(T) = \sum_{x \in V(S)} \mu(x) x_i \prod_{y \in V(S), u(y) \neq u(x)} (T - u(y))$

pour $i = 1,\ldots,n$. *Si* u *sépare* S *(c'est-à-dire si* $\forall (x,y) \in V(S)^2, x \neq y \Rightarrow u(x) \neq u(y)$*), alors les polynômes univariés*

$$\{f_u(T), g_0(T), g_1(T), \ldots, g_n(T)\}$$

definissent une Représentation Univariée Rationnelle (RUR) de S associée à u. *La RUR de* S *a les propriétés suivantes :*

– $f_u(T), g_0(T), g_1(T), \ldots, g_n(T)$ *sont des éléments de* $K[X_1,\ldots,X_n]$

– *l'application :*

$$\begin{array}{rccc} \Pi_u : & C^n & \longrightarrow & C \\ & x & \longmapsto & u(x) \end{array}$$

définit une bijection entre $V(S)$ *et* $V(f_u)$, *dont la réciproque est donnée par :*

$$\begin{array}{rccc} \Pi_u^{-1} : & V(f_u) & \longrightarrow & V(S) \\ & a & \longmapsto & \left(\dfrac{g_1(a)}{g_0(a)}, \ldots, \dfrac{g_n(a)}{g_0(a)}\right) \end{array}$$

– Π_u *préserve les multiplicités* ($\mu(u(x)) = \mu(x)$)*.*

De plus, un élément séparant u *pour* S *peut être choisi parmi les éléments d'une famille*

$$\mathcal{U} = \{X_1 + jX_2 + \ldots + j^{n-1}X_n, j = 0 \ldots nD(D-1)/2\},$$

où D *est la dimension de* $K[X_1,\ldots,X_n]/\langle S \rangle$ *vu comme étant un* K*-espace vectoriel.*

Le point $\left(\dfrac{g_1(a)}{g_0(a)}, \ldots, \dfrac{g_n(a)}{g_0(a)}\right)$ de $V(S)$ est *associé* à la racine a de f_u.

Le calcul d'une Représentation Univariée Rationnelle peut être décomposé en deux étapes :

– prédire un élément séparant,

– calculer une RUR associée à cet élément séparant.

D'après la proposition ci-dessus, on peut procéder comme décrit ci-dessous pour vérifier si un élément $u \in K[X_1,\ldots,X_n]$ est un élément séparant du système zéro-dimensionnel S :

Algorithme CSE

– **Entrée :** Un système d'équations polynomiales S zéro-dimensionnel dans $K[X_1,\ldots,X_n]$, une base de Gröbner G de $I = \langle S \rangle$, le nombre $\sharp V(S)$ de solutions distinctes du système dans C et un élément $u \in K[X_1,\ldots,X_n]$

– **Sortie :** *true* si u est un élément séparant de S, *false* si ce n'est pas le cas.

– Comparer le degré de la partie sans carrés du polynôme minimal de la multiplication par u dans $K[X_1,\ldots,X_n]/I$ et $\sharp V(S)$. Si ces deux nombres sont égaux, retourner *true*, sinon retourner *false*.

Algorithme SE

- **Entrée :** Un système d'équations polynomiales S zéro-dimensionnel dans $K[X_1,\ldots,X_n]$ et une base de Gröbner G de $I = \langle S \rangle$.
- **Sortie :** un élément séparant pour S.

1. Calculer $\sharp V(S)$ en construisant puis en réduisant la forme quadratique de Hermite.
2. Choisir $u \in \mathcal{U}$. Enlever u de $\mathcal{U}$.
3. Utiliser CSE sur u. Si la sortie est *false*, revenir au pas 2.
4. Retourner u.

Dans le cas particulier où l'idéal $I = \langle S \rangle$ est radical, le pas 1 n'est pas nécessaire puisque $\sharp V(S)$ est exactement la dimension du K-espace vectoriel $K[X_1,\ldots,X_n]/I$. Plusieurs stratégies peuvent être utilisées pour calculer la RUR. Par exemple, lorsqu'on sait à l'avance que l'idéal est radical, un élément séparant u est un élément primitif de l'anneau quotient $K[X_1,\ldots,X_n]/I$, et le calcul d'une RUR consiste à exprimer les coordonnées X_i, $i = 1,\ldots,n$ dans l'extension algébrique $K[u]$ (ce qui se fait en inversant la matrice dont les vecteurs colonnes sont les coordonnées de $1,u,\ldots,u^{D-1}$ dans $K[X_1,\ldots,X_n]/I$).

Dans le cas général (l'idéal n'est pas supposé radical), un algorithme pour calculer une RUR est décrit dans [81]. Tous les coefficients de tous les polynômes de la RUR peuvent être déduits des scalaires Trace$(u^i \cdot X_j)$,$i = 0\ldots,D$, $j = 1\ldots n$ où u est l'élément séparant choisi et Trace(P), pour tout $P \in K[X_1,\ldots,X_n]/I$, est la trace l'endomoprhisme K-linéaire de $K[X_1,\ldots,X_n]/I : f \longmapsto f\dot{P}$. Cette méthode permet en particulier de choisir un élément arbitraire u et de vérifier, après calcul, si u est séparant ou pas. Dans le cas particulier de systèmes à coefficients rationnels, un algorithme optimisé, basé sur l'usage de calculs modulaires pour prédire si un élément choisi est séparant, est aussi proposé dans [81]. On note **RUR** l'algorithme suivant :

Algorithme RUR

- **Entrée :** Un système d'équations polynomiales S zéro-dimensionnel dans $K[X_1,\ldots,X_n]$ et une base de Gröbner G de $\langle S \rangle$.
- **Sortie :**Une Représentation Univariée Rationnelle de S.

1. Construire la table de multiplication de $K[X_1,\ldots,X_n]/I$.
2. Utiliser SE pour trouver un élément séparant.
3. Calculer une RUR $(f_u(T),g_0(T),g_1(T),\ldots,g_n(T))$ pour cet élément séparant.

Finalement, compter et décrire les solutions réelles du système polynomial est équivalent à compter et décrire les racines réelles du premier polynôme de la RUR $f_u(T)$ (voir [80, 81]) qui est univarié à coefficients dans K. Ceci peut être fait avec les algorithmes de Sturm-Habicht (voir [87]) ou l'algorithme d'Uspensky (voir [86]) si K est archimédien.

Dans le chapitre 6, nous utiliserons aussi la variante suivante de la RUR, que nous noterons ARUR (voir [4]).

Définition 4.6 *Etant donné un élément séparant* $u \in K[X_1,\ldots,X_n]$ *on définit :*

- $f_u(T) = \prod_{x\in V(S)} (T-u(x))^{\mu(x)}$
- $\tilde{g}_0(T) = \sum_{x\in V(S)} \mu(x)(T-u(x))^{\mu(x)-1} \prod_{y\in V(S)\setminus\{x\}} (T-u(y))^{\mu(y)}$
- $\tilde{g}_i(T) = \sum_{x\in V(S)} \mu(x)x_i(T-u(x))^{\mu(x)-1} \prod_{y\in V(S)\setminus\{x\}} (T-u(y))^{\mu(y)}$

pour $i = 1,\ldots,n$*. Notons que* $\tilde{g}_0$ *est la dérivée de* f_u*. Les polynômes univariés de*

$$\{f_u(T),\tilde{g}_0(T),\tilde{g}_1(T),\ldots,\tilde{g}_n(T)\}$$

définissent ce que nous appellerons l'Ancienne Représentation Univariée Rationnelle (que l'on notera ARUR) de S associée à u.

Cette ARUR de S *a les propriétés suivantes :*

- $f_u(T),\tilde{g}_0(T),\tilde{g}_1(T),\ldots,\tilde{g}_n(T)$ *sont des éléments de* $K[X_1,\ldots,X_n]$*;*
- *l'application :*

$$\begin{array}{rccc} \Pi_u : & C^n & \longrightarrow & C \\ & x & \longmapsto & u(x) \end{array}$$

définit une bijection entre $V(S)$ *et* $V(f_u)$*, dont la réciproque est donnée par :*

$$\begin{array}{rccc} \Pi_u^{-1} : & V(f_u) & \longrightarrow & V(S) \\ & a & \longmapsto & \left(\frac{\tilde{g}_1^{(\mu-1)}(a)}{\tilde{g}_0^{(\mu-1)}(a)},\ldots,\frac{\tilde{g}_n^{(\mu-1)}(a)}{\tilde{g}_0^{(\mu-1)}(a)}\right) \end{array}$$

où μ *est la multiplicité de la racine* a *de* f_u*.*

Tous les coefficients de tous les polynômes de la ARUR peuvent se déduire des scalaires $\mathrm{Trace}(u^i \cdot X_j), i = 0,\ldots,D, j = 1\ldots n$.

Complexité théorique : Soit S un système zéro-dimensionnel dans $K[X_1,\ldots,X_n]$, tel que l'idéal $\mathcal{I}$ est de degré D. Dans [80, 81], F. Rouillier montre que le calcul d'une Représentation Univariée Rationnelle à partir de la table de multiplication de l'anneau-quotient $K[X_1,\ldots,X_n]/\mathcal{I}$ peut se faire en $O(nD^5)$ opérations arithmétiques dans K. De plus, la table de multiplication peut se construire à partir d'une base de Gröbner en $O(D^4)$ opérations arithmétiques dans K. En termes du nombre d'opérations arithmétiques, cet algorithme est polynomial en le nombre de racines complexes d'un système zéro-dimensionnel, une fois la base de Gröbner calculée.

Chapitre 5

Ensembles triangulaires

Résumé

Dans ce chapitre, nous rappelons quelques définitions et propriétés relatives aux ensembles triangulaires de polynômes qui constituent une alternative aux méthodes de résolution fondées sur les bases de Gröbner. Nous rappelons en particulier les définitions d'ensembles triangulaires réguliers et/ou séparables et montrons l'intérêt de ces notions qui seront très souvent utilisées dans la suite du document. Pour terminer, nous rappelons les spécifications des algorithmes de décomposition en ensembles triangulaires réguliers et/ou séparables de Kalkbrener et de Lazard.

Sommaire

Introduction

Les ensembles triangulaires de polynômes apparaissent dans de nombreux travaux concernant la résolution algébrique des systèmes polynomiaux. Plusieurs notions et théories ont été introduites depuis 1932 (ensembles caractéristiques de Ritt [78, 79], ensembles caractéristiques de Wu [103, 104, 39, 40, 97, 98, 99], chaines régulières de Kalkbrener [61, 60], la notion d'ensembles triangulaires normalisés séparables est introduite dans [65]).

Toutes les théories ci-dessus sont assez proches. Les travaux de P. Aubry et M. Moreno Maza, ont permis de clarifier l'ensemble de ces notions théoriques (voir [6, 72, 5]) en montrant que sous certaines hypothèses elles sont équivalentes. Ces algorithmes sont une alternative aux méthodes d'élimination fondées sur les bases de Gröbner mais n'atteignent pas encore leur degré d'efficacité. Les décompositions en ensembles triangulaires présentent l'avantage d'être des sorties agréables dès lors qu'un certain nombre de propriétés leur sont imposées. Signalons toutefois [52] et [71, 64] qui décrivent des algorithmes de décompositions en ensembles triangulaires en faisant intervenir des calculs de bases de Gröbner (au cours de l'algorithme pour le premier, et en entrée pour les deux seconds).

Dans ce document, nous utiliserons intensivement les notions et propriétés relatives aux ensembles triangulaires en particulier dans les chapitres 2 et 3 de la partie II.

La première section de ce chapitre est consacrée au rappel des définitions d'ensembles triangulaires, de quasi-composantes et d'idéal saturé associés à un ensemble triangulaire. Puis, nous rappelons en quoi ces définitions sont insuffisantes pour constituer une sortie «acceptable». Nous rappelons alors les notions de régularité et de séparabilité dans la deuxième section de ce chapitre. La notion de régularité assure de bonnes propriétés par projection aux ensembles triangulaires, et la notion de séparabilité permet de donner au saturé d'un ensemble triangulaire la propriété d'être radical. Enfin dans la dernière section, nous donnons les spécifications des deux algorithmes de décomposition en ensembles triangulaires réguliers et séparables que nous utiliserons dans la suite du document.

5.1 Premières définitions

On considère l'anneau de polynômes $K[X_1,\ldots,X_n]$ et on se donne l'ordre $X_1 < \ldots < X_n$ sur les variables. Si p est un polynôme de $K[X_1,\ldots,X_n]$, on note $\mathrm{mvar}(p)$ la plus grande variable apparaissant dans p pour l'ordre sur les variables qu'on s'est fixé.

Définition 5.1 *Soit $\mathcal{T} = (t_{d+1},\ldots,t_n)$ un ensemble fini de polynômes de $K[X_1,\ldots,X_n]$. On dit que $\mathcal{T}$ est un ensemble triangulaire si et seulement si :*

$$\forall (i,j) \mid i \neq j, \quad \mathrm{mvar}(t_i) \neq \mathrm{mvar}(t_j).$$

Exemple 5.1 *L'ensemble de polynômes*

$$\mathcal{T} = \left\{ \begin{array}{c} xz - 1 \\ y - x \end{array} \right.$$

est un ensemble triangulaire pour l'ordre $x < y < z$ mais n'en est pas un pour l'ordre $z < y < x$.

Dans [6], les auteurs montrent que cette définition n'est pas suffisante pour espérer pouvoir *résoudre* de manière satisfaisante les systèmes polynomiaux. En effet, l'ensemble

$$\mathcal{T} = \begin{cases} xy - 1 \\ x \end{cases}$$

est un ensemble triangulaire pour l'ordre $x < y$, mais il n'admet pas de solutions (complexes ou réelles). En revanche, l'ensemble

$$\mathcal{T} = \begin{cases} xz - y \\ y - x \end{cases}$$

admet des solutions. Il faut donc donner un peu plus de propriétés aux ensembles triangulaires pour qu'ils puissent constituer une sortie satisfaisante d'un éventuel algorithme.

Notation 5.1 *Soit $\mathcal{T} = (t_{d+1}, \ldots, t_n)$ un ensemble triangulaire dans $K[X_1, \ldots, X_n]$. On note h_i le coefficient dominant du polynôme t_i considéré comme un polynôme univarié en* $\mathrm{mvar}(t_i)$. *On appellera ce coefficient initial de t_i et on note h le produit des h_i (pour $i \in \{d+1, \ldots, n\}$.*

Définition 5.2 *Soit $\mathcal{T} \subset K[X_1, \ldots, X_n]$ un ensemble triangulaire. On appelle zéro régulier de $\mathcal{T}$ toute solution de $\mathcal{T}$ dans C^n qui n'annule pas h. L'ensemble des zéro réguliers de $\mathcal{T}$ est noté $W(\mathcal{T})$:*

$$W(\mathcal{T}) = V(\mathcal{T}) \setminus V(h).$$

Un ensemble triangulaire $\mathcal{T}$ est dit consistant si et seulement si $W(\mathcal{T}) \neq \emptyset$.

Reprenons les exemples précédents. Si on a

$$\mathcal{T} = \begin{cases} xy - 1 \\ x \end{cases}$$

alors on a $W(\mathcal{T}) = \emptyset$. Si on a

$$\mathcal{T} = \begin{cases} xz - y \\ y - x \end{cases}$$

alors on a $W(\mathcal{T}) \neq \emptyset$. Notons que $W(\mathcal{T})$ contient les points de C^3 tels que $z = 1$ et $y = x$ qui sont inclus dans la variété algébrique définie par $\mathcal{T}$. En revanche, cette variété contient une composante irréductible définie par

$$\begin{cases} y = 0 \\ x = 0 \end{cases}$$

qui n'est pas incluse dans $W(\mathcal{T})$. Ceci met donc en évidence que l'idéal $\langle\mathcal{T}\rangle$ n'est pas la bonne structure algébrique associée à $W(\mathcal{T})$.

Définition 5.3 *Soit $\mathcal{T} = (t_{d+1},\ldots,t_n)$ un ensemble triangulaire dans $K[X_1,\ldots,X_n]$. On appelle idéal saturé de $\mathcal{T}$ (et on note* $\mathrm{sat}(\mathcal{T})$*) l'idéal*

$$\mathrm{sat}(\mathcal{T}) = \{p \in K[X_1,\ldots,X_n] \mid (\exists n \in \mathbb{N})\ \ h^n p \in \langle\mathcal{T}\rangle\}.$$

Ainsi, lorsque

$$\mathcal{T} = \begin{cases} xy - 1 \\ x \end{cases}$$

l'idéal $\mathrm{sat}(\mathcal{T})$ est $K[x,y,z]$. Aussi, lorsque

$$\mathcal{T} = \begin{cases} xz - y \\ y - x \end{cases}$$

l'idéal $\mathrm{sat}(\mathcal{T})$ est $\langle y - x, z - 1\rangle$.

L'idéal saturé de $\mathcal{T}$ a de meilleures propriétés que celles de l'idéal $\langle\mathcal{T}\rangle$. En particulier, l'idéal saturé d'un ensemble triangulaire est équi-dimensionnel lorsque c'est un idéal propre et si c est le cardinal de $\mathcal{T}$, la dimension de $\mathrm{sat}(\mathcal{T})$ est $n - c$ (voir [5]).

Si F est un sous-ensemble de C^n, on note $\overline{F}$ la clôture de F pour la topologie de Zariski. Dans [6], les auteurs montrent que

$$\overline{W(\mathcal{T})} = V(\mathrm{sat}(\mathcal{T})).$$

Ainsi, un ensemble triangulaire est *consistant* si et seulement si $h \notin \sqrt{\langle\mathcal{T}\rangle}$ (ou encore $\mathrm{sat}(\mathcal{T}) \neq K[X_1,\ldots,X_n]$).

5.2 Notions de régularité et de séparabilité pour les ensembles triangulaires

Soit $\mathcal{T}$ un ensemble triangulaire. On note :

- $\mathrm{algVar}(\mathcal{T})$, l'ensemble des variables dites *algébriques* de $\mathcal{T}$, c'est-à-dire celles qui sont variables principales d'un polynôme de $\mathcal{T}$.
- Si $X_i \in \mathrm{algVar}(\mathcal{T})$, on note $\mathcal{T}_{<i}$ l'ensemble triangulaire constitué des polynômes $t \in \mathcal{T}$ tels que $\mathrm{mvar}(t) < X_i$. Notons que $\mathcal{T}_{<i} = \mathcal{T} \bigcap K[X_1,\ldots,X_{i-1}]$. Par convention, on note $\mathcal{T}_{<1}$ l'ensemble $\mathcal{T} \bigcap K$.
- $\mathrm{sat}_{i-1}(\mathcal{T})$ est l'idéal $\mathrm{sat}(\mathcal{T}) \bigcap K[X_1,\ldots,X_{i-1}]$. Par convention, on note $\mathrm{sat}_0(\mathcal{T})$, l'idéal $\mathrm{sat}(\mathcal{T}) \bigcap K$.

Considérons l'ensemble triangulaire pour l'ordre $x < y$:

$$\mathcal{T} = \begin{cases} xy - 1 \\ x^2 - x \end{cases}$$

Cet ensemble triangulaire est consistant. On a $\mathrm{sat}(\mathcal{T}) \bigcap K[x] = \langle x - 1 \rangle$ tandis que $\mathrm{sat}(\mathcal{T} \cap K[x]) = \langle x^2 - x \rangle$. Ainsi, la projection sur l'axe des x de $W(\mathcal{T})$ est différente de $\overline{W(\mathcal{T} \bigcap K[x])}$. La notion de régularité est introduite pour remédier à ce problème.

Définition 5.4 *Soit $\mathcal{T}$ un ensemble triangulaire dans $K[X_1, \ldots, X_n]$. On dit que $\mathcal{T}$ est régulier si pour toute variable $X_i \in \mathrm{algVar}(\mathcal{T})$, l'initial de $t \in \mathcal{T}$ tel que $\mathrm{mvar}(t) = X_i$ n'est pas diviseur de zéro dans l'anneau quotient $K[X_1, \ldots, X_{i-1}]/\mathrm{sat}_{i-1}(\mathcal{T}_{<i})$.*

De manière équivalente, on peut dire que l'initial de t n'appartient à aucun des idéaux premiers associés au saturé de $\mathcal{T}_{<i}$. Ainsi, dans l'exemple ci-dessus, $\mathcal{T}$ n'est pas un ensemble triangulaire régulier.
On a la propriété suivante :

Théorème 5.1 *[5] Si $\mathcal{T}$ est un ensemble triangulaire régulier dans $K[X_1, \ldots, X_n]$ alors pour tout $i \in \{0, \ldots, n\}$, on a :*

$$\mathrm{sat}(\mathcal{T}) \bigcap K[X_1, \ldots, X_i] = \mathrm{sat}(\mathcal{T} \bigcap K[X_1, \ldots, X_i]).$$

Une preuve de ce résultat se trouve dans [5], page 50.

Corollaire 5.1 *Tout ensemble triangulaire régulier est consistant.*

Preuve : Puisque $\mathrm{sat}_0(\mathcal{T}) \bigcap K = \mathrm{sat}_0(\emptyset) = \{0\}$, 1 n'appartient pas à $\mathrm{sat}(\mathcal{T})$, et donc le corollaire est immédiat. ■

Nous définissons maintenant la notion de séparabilité pour les ensembles triangulaires réguliers.

Définition 5.5 *Soit $\mathcal{T}$ un ensemble triangulaire régulier dans $K[X_1, \ldots, X_n]$. On dit que $\mathcal{T}$ est séparable si pour tout $t \in \mathcal{T}$ (on note $X_i = \mathrm{mvar}(t)$), le polynôme $\frac{\partial t}{\partial X_i}$ (que l'on appellera polynôme séparant de t) n'est pas diviseur de zéro dans l'anneau quotient $K[X_1, \ldots, X_i]/\mathrm{sat}_i(\mathcal{T}_{<i+1})$.*

De manière équivalente, on peut dire que le polynôme séparant de t n'appartient à aucun des idéaux premiers associés au saturé de $\mathcal{T}_{<i+1}$. On a la propriété suivante :

Théorème 5.2 *Le saturé d'un ensemble triangulaire régulier séparable est un idéal radical.*

Proposition 5.1 *Soit $\mathcal{T}$ un ensemble triangulaire régulier séparable de $K[X_1,\ldots,X_n]$. Pour tout $t \in \mathcal{T}$ on a :*

$$\dim(\overline{W(\mathcal{T})} \bigcap V(\frac{\partial t}{\partial \mathrm{mvar}(t)})) < \dim(\overline{W(\mathcal{T})}).$$

Preuve : La preuve est évidente dès lors que le polynôme séparant de tout polynôme de $\mathcal{T}$ n'appartient à aucun des idéaux premiers associés au saturé de $\mathcal{T}$ qui est radical. ∎

5.3 Quelques propriétés supplémentaires

Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite engendrant un idéal premier de dimension d dans $K[X_1,\ldots,X_n]$ pour l'ordre $X_1 < \ldots < X_n$. On note mvar(p) (variable principale de p) la plus grande variable apparaissant dans p. Si F est un sous-ensemble constructible de C^n, on note $\overline{F}$ la clôture de Zariski F dans C^n.

Théorème 5.3 *([6, 5]) Soit $\mathcal{T} = (t_{d+1},\ldots,t_n) \subset \mathcal{G}$ un ensemble de polynômes tels que*

$$\forall (t_i,t_j) \in \mathcal{T} \times \mathcal{T} \ \mathrm{mvar}(t_i) \neq \mathrm{mvar}(t_j),$$

et $\forall g \in \mathcal{G}$, $\forall i \in \{d+1,\ldots,n\}$ tels que $\mathrm{mvar}(t_i) = \mathrm{mvar}(g)$ ([5]) :

$$\deg(t_i,\mathrm{mvar}(t_i)) \leq \deg(g,\mathrm{mvar}(t_i)).$$

On note
- *h_i le coefficient dominant de t_i (lorsqu'il est considéré comme un polynôme univarié en sa variable principale) et $\mathcal{H}(\mathcal{T}) = \{h_{d+1},\ldots,h_n\}$.*
- *$W(\mathcal{T}) = \{M \in V(\mathcal{T}) \setminus V(\Pi_{i=d+1}^n h_i)\}$,*
- *$\mathrm{sat}(\mathcal{T}) = \{p \in K[X_1,\ldots,X_n] \mid \exists m \in \mathbb{N}, \exists h \in \langle \mathcal{H}(\mathcal{T}) \rangle, h^m p \in \langle \mathcal{T} \rangle\}$.*

On a alors :
1. *$\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$,*
2. *$\overline{W(\mathcal{T})} = V(\mathcal{G})$,*

Ainsi, $\mathcal{T} = (t_{d+1},\ldots,t_n)$ est un ensemble triangulaire régulier séparable et $\mathcal{G}$ engendre son saturé.

Notons prem(p,q,X) le pseudo-reste classique de deux polynômes p et q par rapport à la variable X. Si $p \in K[X_1,\ldots,X_n]$, sa forme réduite prem$(p,\mathcal{T})$ peut être calculée par la procédure récursive suivante :
- si $\mathcal{T} = \emptyset$, alors $\mathrm{prem}(p,\mathcal{T}) = p$.
- sinon, si X_i est la plus grande variable apparaissant dans un polynôme $t \in \mathcal{T}$,

$$\mathrm{prem}(p,\mathcal{T}) = \mathrm{prem}(\mathrm{prem}(p,t,X_i),\mathcal{T} \setminus \{t\}).$$

En particulier, ceci implique qu'il existe des polynômes $q_{d+1},\ldots,q_n$ et des entiers positifs $i_{d+1},\ldots,i_n$ tels que :

$$\mathrm{prem}(p,\mathcal{T}) = q_{d+1}t_{d+1} + \ldots + q_n t_n + h_{d+1}^{i_{d+1}} \ldots h_n^{i_n} p.$$

Ainsi, $V(\mathcal{G}) \bigcap V(\mathrm{prem}(p,\mathcal{T})) = V(\mathcal{G}) \bigcap (V(p) \cup V(h_{d+1} \ldots h_n))$. Par conséquent, on a :

$$\dim(V(\mathcal{G}) \bigcap V(p)) < \dim(V(\mathcal{G})) \Longrightarrow \dim(V(\mathcal{G}) \bigcap V(\mathrm{prem}(p,\mathcal{T})) < \dim(V(\mathcal{G})).$$

5.4 Algorithmes

Comme nous l'avons précisé dans l'introduction, il existe plusieurs algorithmes de décomposition des systèmes d'équations polynomiales en ensembles triangulaires mais ils n'ont pas tous les mêmes spécifications. Nous nous intéressons plus particulièrement aux décompositions au sens de Lazard [65, 72] et au sens de Kalkbrener [60, 5].

Définition 5.6 *Soit S un système d'équations polynomiales dans $K[X_1,\ldots,X_n]$.*

- *Soit $\mathcal{T}_1,\ldots,\mathcal{T}_\ell$ une famille finie d'ensembles triangulaires réguliers. On dit que la famille $\mathcal{T}_1,\ldots,\mathcal{T}_\ell$ est une décomposition de S au sens de Kalkbrener si et seulement si :*

$$V(S) = \bigcup_{i=1}^{\ell} \overline{W(\mathcal{T}_i)}.$$

- *Soit $\mathcal{T}_1,\ldots,\mathcal{T}_\ell$ une famille finie d'ensembles triangulaires réguliers. On dit que la famille $\mathcal{T}_1,\ldots,\mathcal{T}_\ell$ est une décomposition de S au sens de Lazard si et seulement si :*

$$V(S) = \bigcup_{i=1}^{\ell} W(\mathcal{T}_i).$$

Notons que de manière générale *les ensembles triangulaires, même réguliers et séparables, ne sont pas des systèmes de générateurs de leur saturé.* C'est pourquoi chacune de ces représentations des solutions peut être qualifiée de «paresseuse». Les décompositions en ensembles triangulaires de Kalkbrener sont plus paresseuses que celles de Lazard car l'ensemble des solutions est renvoyé sous la forme de clôtures de quasi-composantes d'ensembles triangulaires, c'est-à-dire que *toutes les solutions ne sont pas décrites.*

Exemple 5.2 *Considérons le système d'équations polynomiales suivant :*

$$\left\{ \begin{array}{l} xz^2 + yz - 1 = 0 \\ y^2 + x^2 - 1 = 0 \end{array} \right.$$

- *une décomposition de ce système pour l'ordre $x < y < z$ au sens de Lazard renvoie deux ensembles triangulaires :*

$$\mathcal{T}_1 = \left\{ \begin{array}{l} xz^2 + yz - 1 \\ y^2 + x^2 - 1 \end{array} \right. \quad \mathcal{T}_2 = \left\{ \begin{array}{l} yz - 1 \\ y^2 - 1 \\ x \end{array} \right.$$

- *une décomposition de ce système pour l'ordre $x < y < z$ au sens de Kalkbrener renvoie un seul ensemble triangulaire :*

$$\mathcal{T} = \begin{cases} xz^2 + yz - 1 \\ y^2 + x^2 - 1 \end{cases}$$

La décomposition au sens de Lazard a étudié le lieu des points de la variété où le seul initial de $\mathcal{T}_1$ s'annule. Ce lieu est représenté par $\mathcal{T}_2$. La décomposition au sens de Kalkbrener ne fait pas cette étude.

L'algorithme de décomposition en ensembles triangulaires au sens de Lazard est incrémental. Il existe une routine **decompose** (voir [72]) qui étant donné un ensemble triangulaire régulier séparable $\mathcal{T}$ et un polynôme p calcule une famille d'ensembles triangulaires réguliers séparables $\mathcal{T}_1, \ldots, \mathcal{T}_\ell$ telle que :

$$W(\mathcal{T}) \bigcap V(p) \subset W(\mathcal{T}_1) \cup \ldots \cup W(\mathcal{T}_\ell) \subset \overline{W(\mathcal{T}) \bigcap V(p)}.$$

Dans [100, 101], l'auteur donne une généralisation de ces méthodes au cas des systèmes dits *quasi-algébriques*, c'est-à-dire des systèmes d'équations et d'inéquations polynomiales.

L'algorithme de décomposition en ensembles triangulaires au sens de Kalkbrener n'est pas incrémental et ne dispose pas d'une opération similaire à **decompose**. En revanche, P. Aubry a récemment implanté une routine **QuasiKalkbrener** basée sur les algorithmes décrits dans [5] et qui :

- prend un entrée deux familles de polynômes F_1 et F_2,
- retourne une famille d'ensembles triangulaires réguliers et séparables $\mathcal{T}_1, \ldots, \mathcal{T}_\ell$ telle que :

$$\overline{V(F_1) \setminus V(F_2)} = \overline{W(\mathcal{T}_1)} \bigcup \ldots \bigcup \overline{W(\mathcal{T}_\ell)}.$$

Notons que ceci implique que pour tout $i \in \{1, \ldots, \ell\}$ et pour tout $p \in F_2$, on a :

$$\dim(\overline{W(\mathcal{T}_i)} \bigcap V(p)) < \dim(\overline{W(\mathcal{T}_i)}).$$

Cette routine sera utilisée dans le chapitre 2 de la partie II.

Chapitre 6

Un algorithme de bonne complexité basé sur la méthode des points critiques

Résumé

Dans ce chapitre, nous décrivons une alternative à la Décomposition Cylindrique Algébrique qui permet de donner au moins un point par composante semi-algébriquement connexe sur une variété algébrique réelle : la méthode des points critiques. Elle consiste à calculer les points critiques d'une fonction bien choisie restreinte à une variété pour ramener l'étude à celle d'un système algébrique zéro-dimensionnel. L'algorithme que nous étudions dans ce chapitre est extrait de [12, 13, 87]. Après avoir déformé la variété considérée, il calcule les points critiques de la fonction de projection sur un axe. Le système zéro-dimensionnel obtenu est à coefficients infinitésimaux. Il faut ensuite calculer les limites des racines bornées de ce système lorsque ces infinitésimaux tendent vers 0. Nous donnons un nouvel algorithme extrait de [84] permettant de faire ce calcul. A la fin de ce chapitre, nous procédons à une brève analyse expérimentale de cet algorithme.

Sommaire

Introduction

Les travaux de Grigoriev et Vorobjov (voir [53]) sont le point de départ de nouveaux algorithmes permettant de donner au moins un point par composante semi-algébriquement connexe d'un ensemble semi-algébrique, qui sont polynomiaux en le nombre et le degré des polynômes et simplement exponentiels en le nombre de variables. Leur méthode est basée sur le calcul d'un nombre fini de points qui sont les points critiques d'une fonction bien choisie. Nous appellerons cette méthode «la méthode des points critiques».

Plusieurs autres algorithmes fondés sur des variantes de cette méthode ont été proposés plus récemment (voir [55, 56, 74, 12]). Quelques idées permettant d'optimiser les calculs en pratique sont exposées dans [56, 80]. La stratégie proposée (voir [13, 87]) pour calculer au moins un point par composante semi-algébriquement connexe dans un ensemble semi-algébrique est basée sur la construction de routines réduisant le problème de départ à un problème plus facile :

- a) Trouver au moins un point par composante semi-algébriquement connexe dans un ensemble semi-algébrique.
- b) Trouver au moins un point par composante semi-algébriquement connexe dans un ensemble algébrique réel défini par un système d'équations.
- c) Trouver au moins un point par composante semi-algébriquement connexe dans un ensemble algébrique réel défini par une seule équation.
- d) Trouver au moins un point par composante semi-algébriquement connexe dans un ensemble algébrique algébrique réel défini par un système d'équations ayant un nombre fini de solutions complexes.
- e) Compter et isoler les racines d'un polynôme univarié.

Par exemple, le problème b) peut être réduit au problème c) en étudiant la somme des carrés des polynômes intervenant dans le système que l'on veut étudier. De même, le problème d) est réduit au problème e) en calculant une Représentation Univariée Rationnelle et en étudiant le premier polynôme de la sortie.

Dans ce chapitre, nous allons montrer comment ces méthodes proposent d'étudier les hypersurfaces réelles en nous référant principalement à [87]. Dans la première section, nous rappelons les résultats de [87] qui montrent comment opérer diverses déformations infinitésimales sur une hypersurface pour :

- en obtenir une qui soit lisse et compacte;
- obtenir sans calcul une base de Gröbner du système d'équations caractérisant les points critiques de la fonction de projection sur l'axe de première coordonnée.

On peut alors calculer une Représentation Univariée Rationnelle du système zéro-dimensionnel obtenu.

A partir d'une Représentation Univariée Rationnelle de ces points, on doit alors faire tendre certains infinitésimaux vers 0 pour retrouver des points sur l'hypersurface de départ. Cette routine, extraite de [84], est décrite en détail dans la deuxième section de ce chapitre, les solutions précédemment proposées pour résoudre ce problème n'étant pas exactes.

Dans la troisième section de ce chapitre, l'algorithme obtenu est décrit et sa complexité théorique rappelée. Puis, nous montrons en quoi cet algorithme doit être profondément remanié et illustrons nos arguments par le traitement d'un petit exemple. En particulier, nous mettons en évidence que les diverses élévations au carré ainsi que les déformations infinitésimales le rendent impraticable pour des entrées de taille raisonnable.

6.1 Du cas des systèmes à l'étude des hypersurfaces

Pour passer du cas des systèmes au cas des hypersurfaces, la méthode classique proposée (voir [53, 55, 56, 74, 87]) consiste à étudier l'hypersurface définie par la somme des carrés des polynômes du système de départ. En effet, ce type de manipulation ne perturbe pas la complexité théorique de l'algorithme.
Soit $P \in K[X_1, \ldots, X_n]$, on note $V(P) \subset C^n$ l'hypersurface définie par $P = 0$. Soit X_{n+1} une nouvelle variable, on pose

$$P_1 = P^2 + (X_1^2 + \ldots + X_{n+1}^2 - \Omega^2)^2$$

où Ω est une variable positive infiniment grande. Il est démontré dans [13, 12, 87] que l'hypersurface $V(P_1) \bigcap R\langle 1/\Omega\rangle^{n+1}$ est contenue dans la boule ouverte de centre l'origine et de rayon $\Omega + 1$ et que l'extension de toute composante semi-algébriquement connexe de $V(P) \bigcap R^n$ à $R\langle 1/\Omega\rangle$ contient la projection d'une composante de $V(P_1) \bigcap R\langle 1/\Omega\rangle^{n+1}$ sur $R\langle 1/\Omega\rangle^n$.

On note d le degré total de P et d_i (pour $i \in \{1, \ldots, n\}$) les degrés maximaux des monômes de P contenant la variable X_i et on suppose (sans nuire à la généralité) que $d_1 \geq \ldots \geq d_n$. On pose

$$P_1 = (1-\zeta)P + \zeta(X_1^{2(d_1+1)} + \ldots + X_n^{2(d_n+1)} + X_{n+1}^6 - (n+1)(\Omega^{2(d+1)}))$$

où ζ est un infinitésimal positif. Alors, il est démontré dans [13, 12, 87] que :

1. L'ensemble $V(P_1) \bigcap R\langle\zeta\rangle^n$ est borné et lisse.
2. Les polynômes
$$P_1, \frac{\partial P_1}{\partial X_2}, \ldots, \frac{\partial P_1}{\partial X_{n+1}}$$
forment une base de Gröbner pour l'ordre du degré lexicographique $X_1 > \ldots > X_n$. On note $\mathcal{C}$ l'ensemble des solutions de ce système d'équations.
3. L'ensemble $\mathcal{C}$ est un nombre fini de points dans C^n.
4. Soit $\mathcal{C}'$ l'ensemble des points critiques réels. Pour toute composante semi-algébriquement connexe D de $V(P) \bigcap R^n$, il existe un point $M \in \mathcal{C}'$ tel que $\lim_0(M) \in D$.

L'ensemble des points critiques ainsi obtenus après les déformations infinitésimales de l'hypersurface de départ définie par $P = 0$ est fini, et on doit résoudre le système

$$P_1, \frac{\partial P_1}{X_2}, \ldots, \frac{\partial P_1}{X_n}, \frac{\partial P_1}{X_{n+1}}$$

pour pouvoir calculer les coordonnées de ces points critiques. Notons que grâce aux déformations précédemment effectuées, aucun calcul de bases de Gröbner n'est nécessaire, on peut donc directement utiliser l'algorithme décrit dans [80, 81] pour calculer une Représentation Univariée Rationnelle de l'ensemble des solutions. Puisque cet algorithme est polynomial en le nombre de solutions (comptées avec multiplicités), et puisqu'en appliquant le théorème de Bezout, on trouve que ce nombre de solutions est en $\mathcal{O}(d)^n$, il est évident qu'on a un algorithme en $d^{\mathcal{O}(n)}$ opérations arithmétiques. Nous verrons ci-dessous que cette complexité n'est pas perturbée par les routines nécessaires au calcul des limites des racines bornées lorsque ζ tend vers 0 et lorsque Ω tend vers l'infini. En effet, celles-ci sont basées sur des calculs de polynômes caractéristiques dans l'anneau quotienté par l'idéal étudié, ce qui est polynomial en le degré de l'idéal.

Considérons maintenant l'hypersurface définie par le polynôme dans $K[X_1,\ldots,X_n]$ ci-dessous :

$$\sum_{i=1}^{n}(\prod_{j=1}^{d}(X_i-j))^2=0.$$

Ce polynôme est de degré $2d$ et et le lieu réel de l'hypersurface qu'il définit est un ensemble de d^n points isolés. Ainsi, sur cet exemple, la taille de la sortie est en $\mathcal{O}(d)^n$, on peut donc considérer qu'un algorithme simplement exponentiel pour donner au moins un point par composante semi-algébriquement connexe est asymptotiquement optimal.

Dans la section suivante, nous montrons comment on peut calculer les limites de racines bornées d'un système d'équations polynomiales à coefficients dans $K(\varepsilon)$ (où ε est un infinitésimal) lorsque ε tend vers 0. La section suivante rappelle l'algorithme proposé dans [84].

6.2 Calculer les limites de solutions bornées

Rappelons quelques définitions :

- On note $R\langle\varepsilon\rangle$ (resp. $C\langle\varepsilon\rangle$) le corps réel clos (resp. corps algébriquement clos) des *séries de Puiseux algébriques à coefficients dans R (resp. C)* (voir [20, 96]).
- Soit $\alpha=\sum_{i\geq i_0}a_i\varepsilon^{i/q}$ un élément de $R\langle\varepsilon\rangle$ (resp. $C\langle\varepsilon\rangle$) où $i_0\in\mathbb{Z}$, $q\in\mathbb{N}$ et $a_i\in R$ (resp. C), $a_{i_0}\neq 0$ (par convention, on pose $a_i=0$ if $i<i_0$).
- Le nombre rationnel $o(\alpha)=i_0/q$ est l'*ordre* de α, le *coefficient initial* $\mathrm{in}(\alpha)$ de α est le coefficient de $\varepsilon^{o(\alpha)}$ dans α.
- On dit que l'élément α est *borné sur R (resp. C)* si $o(\alpha)$ est positif ou nul. Les éléments de $R\langle\varepsilon\rangle$ (resp. $C\langle\varepsilon\rangle$) qui sont bornés sur R (resp. sur C) forment un anneau de valuation V_ε (resp. $\mathcal{V}_\varepsilon$), la fonction $\lim_0$ de V_ε sur R (resp. $\mathcal{V}_\varepsilon$ sur C) définie par $\lim_0(\alpha)=a_0$ est un homomorphisme d'anneaux.
- On dit que l'élément α est *infinitésimal* sur R (resp. C) si $o(\alpha)$ est strictement positif. On dit que des points $x=(x_1,\ldots,x_n)$ et $y=(y_1,\ldots,y_n)$ dans $R\langle\varepsilon\rangle^n$ (resp. $C\langle\varepsilon\rangle^n$) sont *infiniment proches* si pour tout $i=1,\ldots,n$, x_i-y_i est infinitésimal.

Si $S_\varepsilon\subset K(\varepsilon)[X_1,\ldots,X_n]$ est un système d'équations polynomiales zéro-dimensionnel, on note $V_b(S_\varepsilon)\subset C\langle\varepsilon\rangle^n$ (resp. $V_{R,b}(S_\varepsilon)\subset R\langle\varepsilon\rangle^n$) l'ensemble des solutions bornées de S_ε, à

coordonnées dans $\mathcal{V}_\varepsilon^n$ (resp. V_ε^n).
Considérons $S_\varepsilon \subset K(\varepsilon)[X_1,\ldots,X_n]$ un système d'équations polynomiales zéro-dimensionnel et $A_\varepsilon = K(\varepsilon)[X_1,\ldots,X_n]/\langle S_\varepsilon\rangle$. L'objet de cette section est de calculer une liste de Représentations Univariées Rationnelles à coefficients dans K telles que

- l'ensemble des points de C^n qu'elles représentent est exactement égal à $\lim_0(V_b(S_\varepsilon))$,
- l'ensemble des points de R^n qu'elles représentent est exactement égal à $\lim_0(V_b(S_\varepsilon)) \cap R^n$.

Soit $u \in K[X_1,\ldots,X_n]$. Remarquons que puisque u est à coefficients dans K, l'image par u des éléments bornés de $V(S_\varepsilon)$ dans $C\langle\varepsilon\rangle^n$ est bornée. On note $Z = \lim_0(V_b(S_\varepsilon))$.

Définition 6.1 *Soit S_ε un système zéro-dimensionnel à coefficients dans $K(\varepsilon)$ et $u \in K[X_1,\ldots,X_n]$. On note $f_u(\varepsilon,T)$ le polynôme caractéristique de la multiplication par u dans A_ε. On note $\zeta_1,\ldots,\zeta_p$ les racines de mutiplicité $\mu_1,\ldots,\mu_p$ de $V(S_\varepsilon)$. L'élément u est dit* bien séparant *pour S_ε si et seulement si :*

- *u est un élément séparant pour S_ε;*
- *pour toute racine non bornée ζ_i de $V(S_\varepsilon)$, on a :*

$$o(u(\zeta_i)) = \min(o(X_1(\zeta_i)),\ldots,o(X_n(\zeta_i)));$$

- *pour tout couple de racines bornées (ζ_1,ζ_2) de $V(S_\varepsilon)$, on a :*

$$\lim_0(u(\zeta_1) - u(\zeta_2)) = 0 \Longleftrightarrow \forall i \in \{1,\ldots,n\}\ \lim_0(X_i(\zeta_1) - X_i(\zeta_2)) = 0.$$

Notons que cette définition implique que u est une bijection de Z sur les limites des racines bornées de $f_u(\varepsilon,T)$ lorsque ε tend vers zéro.

Afin d'illustrer les phénomènes pouvant survenir lorsque l'on fait tendre ε vers 0, considérons les exemples suivants, qui justifient la définition donnée ci-dessus :

- **Exemple 1 :** On considère le système d'équations polynomiales

$$XY = 1, X = \varepsilon$$

  La seule solution de ce système est

$$(\varepsilon, \frac{1}{\varepsilon})$$

  qui est non bornée. Or, la forme $u = X$, qui est bien évidemment séparante envoie cette solution sur ε qui est bornée.

- **Exemple 2 :** On considère le système d'équations polynomiales

$$X^2 + Y^2 - 1 = 0, \varepsilon Y = X$$

  Les seules solutions de ce système sont

$$(\frac{\varepsilon}{(1+\varepsilon^2)^{1/2}}, \frac{1}{(1+\varepsilon^2)^{1/2}}), (\frac{-\varepsilon}{(1+\varepsilon^2)^{1/2}}, \frac{-1}{(1+\varepsilon^2)^{1/2}})$$

qui sont bornées et non infiniment proches. La forme séparante $u = X$ envoie ces solutions sur

$$\frac{\varepsilon}{(1+\varepsilon^2)^{1/2}}, \frac{-\varepsilon}{(1+\varepsilon^2)^{1/2}},$$

qui sont infiniment proches.

Ainsi, de manière générale il peut arriver que :

- des solutions non bornées de S_ε soient envoyées sur des éléments bornés de $C\langle\varepsilon\rangle^n$;
- des solutions non infiniment proches sont envoyées sur des éléments infiniment proches dans $C\langle\varepsilon\rangle^n$.

On va montrer que si on choisit une forme linéaire qui est un *élément bien séparant* de S_ε, on n'aura pas de difficulté à faire tendre ε vers 0 en gardant toutes les informations dont nous avons besoin.
Soit $u = u_1X_1 + \ldots + u_nX_n$, (avec $u_i \in K$) un élément séparant de S_ε. Puisque le système d'équations polynomiales S_ε est contenu dans $K[\varepsilon][X_1, \ldots, X_n]$, les polynômes

$$(f_u(\varepsilon,T), \tilde{g}_0(\varepsilon,T), \tilde{g}_1(\varepsilon,T), \ldots, \tilde{g}_n(\varepsilon,T))$$

de la ARUR associée à u sont des éléments de $K(\varepsilon)[T]$. Remarquons que $f_u(\varepsilon,T)$ est unitaire. Soit ν le plus petit entier tel que $\varepsilon^\nu c(\varepsilon) f_u(\varepsilon,T)$ est un élément de $K[\varepsilon][T]$, avec $c(\varepsilon) \in K[\varepsilon], c(0) \neq 0$. On appelle alors

$$(F_u(\varepsilon,T), G_0(\varepsilon,T), G_1(\varepsilon,T), \ldots, G_n(\varepsilon,T)),$$

Représentation Univariée Rationnelle (NRUR) l'ensemble de polynômes

$$(\varepsilon^\nu c(\varepsilon) f_u(\varepsilon,T), \varepsilon^\nu \tilde{g}_0(\varepsilon,T), \varepsilon^\nu \tilde{g}_1(\varepsilon,T), \ldots, \varepsilon^\nu \tilde{g}_n(\varepsilon,T)).$$

Cette NRUR définit les mêmes points que la RUR initiale. S'il existe des polynômes $c_1(\varepsilon), \ldots, c_n(\varepsilon)$ dans $K[\varepsilon]$ tels que pour tout $i \in \{1, \ldots, n\}$, on a $c(0) \neq 0$ et $c_i(\varepsilon) G_i(\varepsilon,T) \in K[\varepsilon,T]$, on dira que la NRUR est *bien normalisée.*
Dans l'exemple 1, on a avec $u = X$

$$G_0(\varepsilon,T) = 1, G_1(\varepsilon,T) = \varepsilon, G_2(\varepsilon,T) = \frac{1}{\varepsilon.}$$

Lemme 6.1 *Soit $u = \sum_{i=1}^n u_i X_i$ avec $u_i \in K$.*

- *Le polynôme $f_u(\varepsilon,T)$ a des racines non bornées dans $C\langle\varepsilon\rangle$ si et seulement si*

$$\deg_T(F_u(0,T)) < \deg_T(f_u(\varepsilon,T))$$

- *ν égale $\sum_{j=\ell+1,\ldots,p} -o(\alpha_j)\mu_j$ où $\alpha_{\ell+1}, \ldots, \alpha_p$ sont les racines non bornées de $f_u(\varepsilon,T)$ d'ordres respectifs $o(\alpha_{\ell+1}), \ldots, o(\alpha_p)$ et de multiplicités respectives $\mu_{\ell+1}, \ldots, \mu_p$,*
- *si α est une racine de $f(\varepsilon,T)$ dans $C\langle\varepsilon\rangle$ bornée sur C, alors $a = \lim_0(\alpha)$ est une racine de $F_u(0,T)$.*

Preuve : Soit $\alpha_1,\ldots,\alpha_\ell$ les racines bornées de $f_u(\varepsilon,T)$, de multiplicités $\mu_1,\ldots,\mu_\ell$. Alors,

$$f_u(\varepsilon,T) = \prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j} \prod_{j=\ell+1}^{p}(T-\alpha_j)^{\mu_j} \in K(\varepsilon)[T],$$

et il est clair que l'ordre du coefficient de $T^{\sum_{j=1}^{\ell}\mu_j}$ dans $f_u(\varepsilon,T)$ est $\sum_{j=\ell+1}^{p}\mu_j o(\alpha_j)$, et que l'ordre de n'importe quel autre coefficient de $f_u(\varepsilon,T)$ est inférieur à $\sum_{j=\ell+1}^{p}\mu_j o(\alpha_j)$. On note $c(\varepsilon)$ un dénominateur commun des coefficients de

$$\prod_{j=\ell+1}^{p}(\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j} \prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j}.$$

Il est alors clair que $c(0) \neq 0$ et

$$F_u(\varepsilon,T) = c(\varepsilon) \prod_{j=\ell+1}^{p}(\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j} \prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j} \in K[\varepsilon][T].$$

On note $a_j = \lim_0(\alpha_j)$ pour $j = 1,\ldots,\ell$,

$$F_u(0,T) = c(0) \prod_{j=\ell+1}^{p}(-\mathrm{in}(\alpha_j))^{\mu_j} \prod_{j=1}^{\ell}(T-a_j)^{\mu_j}.$$

■

On peut maintenant montrer :

Lemme 6.2 *Soit* $u = \sum_{i=1}^{n} u_i X_i$, $u_i \in K$ *est un élément séparant pour* S_ε. *Alors* u *est un* élément bien séparant *si et seulement si :*

- *Il existe des polynômes* $c_1(\varepsilon),\ldots,c_n(\varepsilon)$ *dans* $K[\varepsilon]$ *tels que pour tout* $i \in \{1,\ldots,n\}$, $c_i(0)$ *est non nul et les polynômes*

$$c_1(\varepsilon)G_1(\varepsilon,T),\ldots,c_n(\varepsilon)G_n(\varepsilon,T)$$

  appartiennent à $K[\varepsilon,T]$,
- *les images par* u *de deux éléments bornés* x *et* y *de* $V(S_\varepsilon)$ *sont infiniment proches, si et seulement si* x *et* y *sont infiniment proches.*

Preuve : Soit $\alpha_1,\ldots,\alpha_p$ les racines de $f_u(\varepsilon,T)$ de multiplicités respectives $\mu_1,\ldots,\mu_p$ numérotées de manière à ce que $\alpha_1,\ldots,\alpha_\ell$ soient les racines bornées de $f_u(\varepsilon,T)$. On note $\nu = \sum_{j=\ell+1}^{p} -o(\alpha_j)\mu_j$.

- Supposons que u est un élément bien séparant. A fortiori, u est donc séparant, et l'application Π_u définie dans la proposition 4.3 est inversible. On note Π_u^{-1} son inverse. On a pour tout $i \in \{1,\ldots,n\}$:

$$\tilde{g}_i(\varepsilon,T) = \sum_{k=1}^{p}\mu_k X_i(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1}\prod_{j\neq k}(T-\alpha_j)^{\mu_j}.$$

$$\tilde{g}_i(\varepsilon,T) = \sum_{k=1}^{\ell} \mu_k X_i(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1} \prod_{j\neq k, j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j=\ell+1}^{p} (T-\alpha_j)^{\mu_j}$$
$$+ \sum_{k=\ell+1}^{p} \mu_k X_i(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1} \prod_{j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j\neq k, j=\ell+1}^{p} (T-\alpha_j)^{\mu_j}.$$

Ainsi, on a :

$$\varepsilon^{\nu}\tilde{g}_i(\varepsilon,T) = \sum_{k=1}^{\ell} \mu_k X_i(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1} \prod_{j\neq k, j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j=\ell+1}^{p} (\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j}$$
$$+ \sum_{k=\ell+1}^{p} \mu_k \varepsilon^{-o(\alpha_k)} X_i(\Pi_u^{-1}(\alpha_k))(\varepsilon^{-o(\alpha_k)}T - \varepsilon^{-o(\alpha_k)}\alpha_k)^{\mu_k-1} \prod_{j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j\neq k, j=\ell+1}^{p} (\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j}.$$

Comme le système de départ est à coefficients dans $K[\varepsilon]$, il est clair que $\varepsilon^{\nu}\tilde{g}_i(\varepsilon,T)$ est un polynôme à coefficients dans $K(\varepsilon)$. Comme u est bien séparant, les deux termes de la somme ci-dessus ne tendent pas vers l'infini lorsque ε tend vers 0, ce qui implique que pour $i \in \{1,\ldots,n\}$, il existe $c_i(\varepsilon) \in K[\varepsilon]$ tel que $c_i(0) \neq 0$ et $\varepsilon^{\nu}c_i(\varepsilon)\tilde{g}_i(\varepsilon,T) \in K[\varepsilon,T]$.

- Pour démontrer la réciproque nous allons démontrer sa contraposée. On suppose que u est un élément séparant tel que :
  - les images par u de deux éléments bornés x et y de $V(S_\varepsilon)$ sont infiniment proches si et seulement si x et y sont infiniment proches,
  - et il existe une racine non bornée ζ de S_ε telle que
  $$o(u(\zeta)) > \min(o(\zeta_1),\ldots,o(\zeta_n))$$
  où ζ_i est la i-ième coordonnée de ζ.

  Soit $i_0 > \ell$ tel que $o(\zeta_{i_0}) = \min(o(\zeta_1),\ldots,o(\zeta_n))$. Nous allons montrer que ceci implique que $G_{i_0} \notin K[\varepsilon,T]$. On a :
  $$\tilde{g}_{i_0}(\varepsilon,T) = \sum_{k=1}^{p} \mu_k X_{i_0}(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1} \prod_{j\neq k} (T-\alpha_j)^{\mu_j}.$$
  Supposons (sans nuire à la généralité) que α_p soit la racine de f telle que $\Pi_u^{-1}(\alpha_p) = \zeta$. On a alors :
  $$\varepsilon^{\nu}\tilde{g}_{i_0}(\varepsilon,T) = \sum_{k=1}^{\ell} \mu_k X_{i_0}(\Pi_u^{-1}(\alpha_k))(T-\alpha_k)^{\mu_k-1} \prod_{j\neq k, j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j=\ell+1}^{p} (\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j} +$$
  $$\sum_{k=\ell+1}^{p} \mu_k \varepsilon^{-o(\alpha_k)} X_{i_0}(\Pi_u^{-1}(\alpha_k))(\varepsilon^{-o(\alpha_k)}T - \varepsilon^{-o(\alpha_k)}\alpha_k)^{\mu_k-1} \prod_{j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j\neq k, j=\ell+1}^{p} (\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j} +$$
  $$\mu_p \varepsilon^{-o(\alpha_p)} \zeta_{i_0} (\varepsilon^{-o(\alpha_p)}T - \varepsilon^{-o(\alpha_p)}\alpha_p)^{\mu_p-1} \prod_{j=1}^{\ell} (T-\alpha_j)^{\mu_j} \prod_{j=\ell+1}^{p-1} (\varepsilon^{-o(\alpha_j)}T - \varepsilon^{-o(\alpha_j)}\alpha_j)^{\mu_j}.$$
  Il est alors clair que l'ordre du coefficient de $T^{\sum_{k=1}^{\ell}\mu_k}$ est
  $$\min(o(\zeta_{i_0}) - o(\alpha_p),(o(X_{i_0}(\Pi_u^{-1}(\alpha_k))) - o(\alpha_k),k=\ell,\ldots,p-1))$$
  Donc, on a bien $c_{i_0}(0) = 0$.

■

Afin de pouvoir trouver un élément bien séparant pour S_ε nous allons montrer qu'il en existe un dans une liste de formes linéaires pré-déterminée que nous noterons $\mathcal{U}$.

Lemme 6.3 *Soit D la dimension de l'espace vectoriel $K[X_1,\ldots,X_n]/\langle S_\varepsilon\rangle$. Un élément bien séparant u pour S_ε peut être choisi parmi les éléments de la famille*

$$\mathcal{U} = \{X_1 + jX_2 + \ldots + j^{n-1}X_n, j = 0\ldots(n-1)D^2\}.$$

Preuve : Soit x une solution de S_ε, on note x_i la i-ième coordonnée de x. Il est clair que si u est un élément bien séparant alors :

- $\forall(x,y) \in V(S_\varepsilon)^2\ u(x-y) \neq 0$,
- si x et y sont deux solutions bornées non infiniment proches de S_ε alors $u(\lim_0(x) - \lim_0(y)) \neq 0$,
- si on note c_i le coefficient de $\varepsilon^{\min_{i=1,\ldots,n}(o(x_i))}$ dans x_i, alors $u(c_1,\ldots,c_n) \neq 0$ (ce qui implique que la NRUR est $G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T)$ *bien normalisée*).

On définit alors :

1. $\mathcal{W}_1$, de cardinalité $\leq D(D-1)/2$, comme étant l'ensemble des vecteurs $x-y$ où x et y sont des solutions distinctes de S_ε dans $C\langle\varepsilon\rangle^n$,
2. $\mathcal{W}_2$, de cardinalité $\leq D$, comme étant l'ensemble de vecteurs $c = (c_1,\ldots,c_n)$ où c_i est le coefficient de $\varepsilon^{\min_{i=1,\ldots,k}(o(x_i))}$ dans x_i,
3. $\mathcal{W}_3$, de cardinalité $\leq D(D-1)/2$, comme étant l'ensemble de vecteurs $\lim_0(x) - \lim_0(y)$ où x et y sont des solutions distinctes non infiniment proches de S_ε dans $C\langle\varepsilon\rangle^n$,
4. $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2 \cup \mathcal{W}_3$. Notons que $\mathcal{W}$ est de cardinalité $\leq D^2$.

Ainsi, pour que u soit bien séparant, il suffit que $\forall w \in \mathcal{W}\ u(w) \neq 0$. Soit $w \in \mathcal{W}$ fixé. Il y a au plus $n-1$ éléments dans $\mathcal{U}$ tels que $u(w) = 0$. En effet, ces éléments sont caractérisés par le fait qu'on peut leur associer un entier j qui annule le polynôme $P(T) = w_1 + w_2T + \ldots + w_nT^{n-1}$. Or, chacun de ces polynômes ne peut avoir au plus que $n-1$ racines. Comme $\mathcal{W}$ est de cardinalité inférieure ou égale à D^2, il existe au moins un élément bien séparant dans $\mathcal{U}$. ■

On en déduit l'algorithme **NWSE** (**N**aive **W**ell **S**eparating **E**lement) qui trouve un élément bien séparant.

Algorithme NWSE

- **Entrée :** Un système zéro-dimensionnel S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$ et une base de Gröbner G_ε de l'idéal I_ε engendré par S_ε.
- **Sortie :** un *élément bien séparant* u de S_ε, une décomposition square-free de $F_u(0,T)$ et

$$(G_0(0,T),G_1(0,T),\ldots,G_n(0,T))$$

où

$$(G_0(\varepsilon,T),G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T))$$

est une NRUR *bien normalisée* pour u.

1. Pour tout $u \in \mathcal{U}$
   - Vérifier que u est séparant.
   - Calculer une NRUR associée à u

   $$(F_u(\varepsilon,T),G_0(\varepsilon,T),G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T)),$$

   garder u si et seulement si la NRUR est *bien normalisée.*
2. Parmi les éléments de $\mathcal{U}'$ gardés au pas 1 Caluler la décomposition square-free de $F_u(0,T)$

   $$F_u(0,T) = f_1 f_2^2 \ldots f_m^m.$$

3. choisir u tel que le degré de la décomposition square-free de $F_u(0,T)$ est maximal.
4. Retourner $(u,f_1,\ldots,f_m)$ et

   $$(G_0(0,T),G_1(0,T),\ldots,G_n(0,T)).$$

Notons qu'en général une forme linéaire choisie au hasard sera un élément bien séparant. Nous suivons donc la démarche suivante : on choisit une forme linéaire puis on vérifie que cette forme est un élément bien séparant. Ainsi, il faut pouvoir vérifier si une forme linéaire fixée est un élément bien séparant.

Lemme 6.4 *Si $u = u_1X_1 + \ldots + u_nX_n$, (avec $u_i \in K$) est un élément bien séparant pour S_ε et si a est une racine de $F_u(0,T)$ de multiplicité m alors il existe une racine α de $f_u(\varepsilon,T)$ dans $C\langle\varepsilon\rangle$ avec $\lim_0(\alpha) = a$. De plus, pour tout α racine de $f_u(\varepsilon,T)$ dans $C\langle\varepsilon\rangle$ de multiplicité μ avec $\lim_0(\alpha) = a$, on a*

$$\lim_0\left(\frac{g_i^{(\mu-1)}(\varepsilon,\alpha)}{g_0^{(\mu-1)}(\varepsilon,\alpha)}\right) = \frac{G_i^{(m-1)}(0,a)}{G_0^{(m-1)}(0,a)}.$$

Preuve : Soit $\alpha_1,\ldots,\alpha_p$ les racines de $f_u(\varepsilon,T)$ et μ_j la multiplicité de α_j. Soit $\alpha_1,\ldots,\alpha_\ell$ les racines bornées de $f_u(\varepsilon,T)$. On note $\varepsilon_j = \varepsilon^{-o(\alpha_j)}$ (pour $j = \ell+1,\ldots,p$). On a alors :

$$f_u(\varepsilon,T) = \prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j}\prod_{j=\ell+1}^{p}(T-\alpha_j)^{\mu_j},$$

$$\varepsilon^{\nu} f_u(\varepsilon,T) = \prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j}\prod_{j=\ell+1}^{p}(\varepsilon_j T-\varepsilon_j\alpha_j)^{\mu_j},$$

$$F_u(\varepsilon,T) = c(\varepsilon)\prod_{j=1}^{\ell}(T-\alpha_j)^{\mu_j}\prod_{j=\ell+1}^{p}(\varepsilon_j T-\varepsilon_j\alpha_j)^{\mu_j},$$

$$F_u(0,T) = c\prod_{j=1}^{\ell}(T-a_j)^{\mu_j}$$

$$G_0(0,T) = \sum_{j=1}^{\ell}\mu_j(T-a_j)^{\mu_j-1}\prod_{i\in\{1,\dots,\ell\}\setminus\{j\}}(T-a_i)^{\mu_i}.$$

Supposons que $a = \lim_0(\alpha_1) = \ldots = \lim_0(\alpha_s)$, $a \neq \lim_0(\alpha_j), j > s$, alors

$$G_0^{(m-1)}(0,a) = c.\ell\ldots(\ell-m+1)\prod_{j=m+1}^{\ell}(a-a_j),$$

où $m = \mu_1 + \ldots + \mu_s$.
On note ξ_j le point de $V(S_\varepsilon)$ tel que $u(\xi_j) = \alpha_j$, et ξ_{ij} la i-ème coordonnée de ξ_j. On a aussi

$$\tilde{g}_i(\varepsilon,T) = \sum_{j=1}^{p}\xi_{ij}\mu_j(T-\alpha_j)^{\mu_j-1}\prod_{i\in\{1,\dots p\}\setminus\{j\}}(T-\alpha_i)^{\mu_i}$$

$$\varepsilon^{\nu}\tilde{g}_i(\varepsilon,T) = \sum_{j=1}^{\ell}\xi_{ij}\mu_j(T-\alpha_j)^{\mu_j-1}\prod_{i\in\{1,\dots\ell\}\setminus\{j\}}(T-\alpha_i)^{\mu_i}\prod_{i=\ell+1}^{p}(\varepsilon_i T-\varepsilon_i\alpha_i)^{\mu_i}$$
$$+\prod_{i=1}^{\ell}(T-\alpha_i)^{\mu_i}\Big(\sum_{j=\ell+1}^{p}\varepsilon_j\xi_{ij}\mu_j(\varepsilon_j T-\varepsilon_j\alpha_j)^{\mu_j-1}\prod_{i\in\{\ell+1,\dots p\}\setminus\{j\}}(\varepsilon_i T-\varepsilon_i\alpha_i)^{\mu_i}\Big).$$

Il est clair que

$$\sum_{j=1}^{\ell}\xi_{ij}\mu_j(T-\alpha_j)^{\mu_j-1}\prod_{i\in\{1,\dots\ell\}\setminus\{j\}}(T-\alpha_i)^{\mu_i}\prod_{i=\ell+1}^{p}(\varepsilon_i T-\varepsilon_i\alpha_i)^{\mu_i}\in\mathcal{V}_\varepsilon[T].$$

Puisque $G_i(\varepsilon,T)\in K[\varepsilon,T]$, $\varepsilon^{\nu}\tilde{g}_i(\varepsilon,T)\in\mathcal{V}_\varepsilon[T]$, $\prod_{i=1}^{\ell}(T-\alpha_i)^{\mu_i}\in\mathcal{V}_\varepsilon[T]$ est unitaire

$$G_i(\varepsilon,T) = A + \prod_{i=1}^{\ell}(T-\alpha_i)^{\mu_i}B$$

avec $A,B\in\mathcal{V}_\varepsilon[T]$. Ainsi

$$G_i(0,T) = \bar{A} + \prod_{i=1}^{\ell}(T-a_i)^{\mu_i}\bar{B},$$

avec $\lim_0(\alpha_j) = a_j$, $\bar{A}$ et $\bar{B}$ les polynômes de $C[T]$, obtenus en remplaçant chaque coefficient c de A et B par $\lim_0(c)$. Donc, puisque $a = \lim_0(\alpha_1) = \ldots = \lim_0(\alpha_s)$, $a \neq \lim_0(\alpha_j), j > s$, en notant $x = \lim_0(\xi_1) = \ldots = \lim_0(\xi_s)$, avec $u(\xi_i) = \alpha_i$,

$$G_i^{(m-1)}(0,a) = c.\ell\ldots(\ell-m-1)x_i\prod_{j=m+1}^{\ell}(a-a_j),$$

où $m = \mu_1 + \ldots + \mu_s$ et finalement

$$x_i = \frac{G_i^{(m-1)}(0,a)}{G_0^{(m-1)}(0,a)}.$$

■

Remarquons que la multiplicité d'un point x de Z est la somme des multiplicités des points $\xi \in V(S_\varepsilon)$ tels que $\lim_0(\xi) = x$.

Lemme 6.5 *Soit $u = u_1X_1 + \ldots + u_nX_n$, $u_i \in K$, tel que la NRUR soit dans $K[\varepsilon,T]$. Soit $h_i(\varepsilon,\Lambda_i)$ le polynôme caractéristique de l'endomorphisme de multiplication par X_i dans A_ε, et $H_i \in K[\varepsilon][\Lambda_i],H_i \notin \varepsilon K[\varepsilon][\Lambda_i]$ un multiple de h_i, que l'on appellera polynôme caractéristique normalisé de X_i. Soit $F_u(0,T) = f_1f_2^2 \ldots f_s^s$, la décomposition sans carrés de $F_u(0,T)$. L'élément u est bien séparant si et seulement si en notant*

$$K_i(\Lambda_i) = \prod_{k=1}^{s} \mathrm{Res}(G_0(0,T)^{(k-1)}\Lambda_i - G_i(0,T)^{(k-1)},f_k)^k,$$

K_i divise $H_i(0,\Lambda_i)$.

Preuve : Si u est une forme bien séparante, les racines de K_i sont les racines qui sont de la forme

$$x_i = \frac{G_i^{(m-1)}(0,a)}{G_0^{(m-1)}(0,a)}$$

où a est une racine de f_m (i.e. une racine de multiplicité m de $F_u(0,T)$) et $u(x) = a$,

$$x_i = \frac{G_i^{(m-1)}(0,a)}{G_0^{(m-1)}(0,a)}.$$

Aussi les racines de K_i sont racines de $H_i(0,\Lambda_i)$ avec les mêmes multiplicités.

Réciproquement, si u n'est pas une forme bien séparante, il existe une racine a de $G(0,T)$ avec $x_1,\ldots,x_s$ des éléments de Z de multiplicités $n_1,\ldots,n_s$ et $u(x_1) = \ldots = u(x_s) = a$. Soit $m = m_1 + \ldots + m_s$, alors il est clair, d'après les définitions, que :

$$\frac{G_i^{(m-1)}(0,a)}{G_0^{(m-1)}(0,a)} = \frac{m_1x_1 + \ldots + m_sx_s}{m}.$$

Aussi le s-uplet $x_1,\ldots,x_s$ est remplacé par le barycentre b des points x_i de poids n_i. En appliquant le lemme ci-dessous, on conclut qu'il existe une racine x_i d'un des polynômes $H_i(0,\Lambda_i)$ de multiplicité m dont la multiplicité dans K_i est supérieure à m. ■

Lemme 6.6 *Soit Z un multi-ensemble fini de points $x \in C^n$ de multiplicité $\mu(x)$. On note $\Pi_i(Z)$ le multi-ensemble obtenu en projetant les points de Z sur leur i-ème coordonnée (en ajoutant les multiplicités si les points ont la même i-ème coordonnée). Soit Z' un multi-ensemble obtenu en remplaçant les sous-ensembles disjoints de Z par leur barycentre (en prenant en compte les multiplicités). Alors $Z \neq Z'$ si et seulement si il existe un i tel que $\Pi_i(Z) \neq \Pi_i(Z')$.*

Preuve : Supposons que $Z \neq Z'$ et notons W l'ensemble des points de Z qui ne sont pas dans Z'. Soit H l'enveloppe convexe de W, x un point extrême de H, et i tel que x_i est distinct de la i-ième coordonnée du barycentre qui le remplace. Puisqu'un barycentre de points est est dans l'intérieur de l'enveloppe convexe de ces points, la multiplicité de x_i dans $\Pi_i(Z')$ est strictement inférieure à celle de de x_i dans $\Pi_i(Z)$. On en déduit qu'il existe un point y tel que la multiplicité de y_i dans $\Pi_i(Z')$ est strictement supérieure à celle de y_i dans $\Pi_i(Z)$. ■

D'après les lemmes précédents, l'algorithme **WSE** (**W**ell **S**eparating **E**lement) décrit ci-dessous retourne un élément bien séparant.

Algorithme WSE

- **Entrée :** Un système zéro-dimensionnel S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$ et une base de Gröbner G_ε de l'idéal I_ε engendré par S_ε.
- **Sortie :** un *élément bien séparant* u de S_ε, une décomposition square-free de $F_u(0,T)$ et

$$(G_0(0,T),G_1(0,T),\ldots,G_n(0,T))$$

où

$$(G_0(\varepsilon,T),G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T))$$

est une NRUR *bien normalisée* pour u.

1. Calculer pour tout i H_i le polynôme caractéristique normalisé de X_i dans A_ε.
2. Choisir $u \in \mathcal{U}$. Enlever u de $\mathcal{U}$.
3. Vérifier que u est séparant.
4. Calculer une NRUR

$$(F_u(\varepsilon,T),G_0(\varepsilon,T),G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T)).$$

Si cette NRUR n'est pas *bien normalisé*, revenir à 2.

5. Calculer la décomposition square-free

$$F_u(0,T) = f_1 f_2^2 \ldots f_m^m.$$

Calculer

$$K_i(\Lambda_i) = \prod_{j=1}^{m} \mathrm{Res}(G_0(0,T)^{(j-1)}\Lambda_i - G_i(0,T)^{(j-1)},f_j)^j,$$

Si il existe i tel que K_i ne divise pas $H_i(0,\Lambda_i)$, revenir à 1.

6. Retourner $u,,f_1,\ldots,f_m$ et

$$(G_0(0,T),G_1(0,T),\ldots,G_n(0,T)).$$

Dans l'exemple 2, quand $u = X$, $F_u(\varepsilon,T) = (1+\varepsilon^2)T^2 - \varepsilon^2$ est sans carrés et $F_u(0,T) = T^2$. Le polynôme caractéristique normalisé $H_2(\varepsilon,\Lambda_2)$ de la multiplication par Y est $(1+\varepsilon^2)\Lambda_2^2 - 1$, aussi $H_2(0,\Lambda_2) = \Lambda_2^2 - 1$, alors que $G_0(\varepsilon,T) = 2(1+\varepsilon^2)T$, $G_0(0,T) = 2T$, $G_1(\varepsilon,T) = 2\varepsilon^2$ $G_2(0,T) = 0$, $K_2(0,\Lambda_2) = \Lambda_2^2$. Donc, $u = X$ n'est pas un élément bien séparant.

Maintenant que nous savons détecter un élément bien séparant pour S_ε, nous décrivons ci-dessous l'algorithme **LRB** (**L**imites **R**acines **B**ornées) qui calcule $Z = \lim_0(V_b(S_\varepsilon))$.

Algorithme LRB

- **Entrée :** Un système zéro-dimensionnel S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$ et une base de Gröbner G_ε de l'idéal I_ε engendré par S_ε.
- **Sortie :**Une liste de solutions réelles de Représentations Univariées Rationnelles contenant les limites des solutions bornées dans $R\langle\varepsilon\rangle$ de S_ε.

- Utiliser l'algorithme WSE pour trouver un élément séparant de S_ε.
- Retourner $listrurs = \{(f_i,G_0(0,T)^{(i-1)},G_1(0,T)^{(i-1)}\ldots,G_n(0,T)^{(i-1)})_{i\in\{1,\ldots,m\}}\}$,
- Pour tout élément de *listrurs* compter et isoler les racines réelles de son premier polynôme.

Remarque 6.1 *Notons que les Représentations Univariées Rationnelles retournées par cet algorithme forment un système de générateurs d'un idéal radical (en effet, puisque les polynômes éliminants de ces Représentations Univariées Rationnelles sont square-free, toutes les solutions sont simples).*

Exemple 3 : Considérons le système d'équations polynomiales à coefficients dans $K[\varepsilon]$ ci-dessous :

$$P(X,Y) = \varepsilon,\ \ \overrightarrow{\mathrm{grad}}_{X,Y}\ (P)//(X,Y).$$

où $P(X,Y) = Y^2 + (XY-1)^2$. Il est facile de vérifier que la variable X est un élément séparant Nous allons dérouler l'algorithme pour montrer que X est un élément bien séparant pour ce système. Le polynôme caractéristique de la multiplication par X dans l'anneau $K(\varepsilon)[X,Y]$ quotienté par l'idéal engendré par le système est

$$\begin{gathered}F(\varepsilon,X) = \varepsilon X^{10} + (4\varepsilon-1)X^8 + (-3-2\varepsilon^2+4\varepsilon)X^6 + (10\varepsilon-1-4\varepsilon^2)X^4 \\ +(10\varepsilon-7+\varepsilon^3-4\varepsilon^2)X^2 - \varepsilon^2 + 2\varepsilon - 1\end{gathered}$$

On a :

$$\begin{gathered}G_0(\varepsilon,X) = 10\varepsilon X^9 + 8(4\varepsilon-1)X^7 + 6(-3-2\varepsilon^2+4\varepsilon)X^5 \\ +4(10\varepsilon-1-4\varepsilon^2)X^3 + 2(10\varepsilon-7+\varepsilon^3-4\varepsilon^2)X, \\ G_1(\varepsilon,X) = 10\varepsilon X^8 + (16\varepsilon-8)X^6 + (4\varepsilon^2-10\varepsilon-4)X^4 \\ +(-8\varepsilon^2-4\varepsilon+12)X^2 + 2(\varepsilon-1)(\varepsilon^2-2\varepsilon+1), \\ G_2(\varepsilon,X) = (-8\varepsilon+2)X^8 + (12+8\varepsilon^2-16\varepsilon)X^6 + (-60\varepsilon+6+24\varepsilon^2)X^4 \\ +(-80\varepsilon+56-8\varepsilon^3+32\varepsilon^2)X^2 + 10\varepsilon^2 - 20\varepsilon + 10.\end{gathered}$$

On a aussi :

$$\begin{gathered}F(0,X) = -X^8 - 3X^6 - X^4 - 7X^2 - 1 \\ G_0(0,X) = -8X^7 - 18X^5 - 4X^3 - 14X \\ G_1(0,X) = -8X^6 - 4X^4 + 12X^2 - 2 \\ G_2(0,X) = 2X^8 + 12X^6 + 6X^4 + 56X^2 + 10\end{gathered}$$

Le polynôme caractéristique normalisé $H_1(\varepsilon,\Lambda_1)$ est égal à :

$$H_1(\varepsilon,\Lambda_1) = \Lambda_1^{10} - \varepsilon\Lambda_1^8 + (-2\varepsilon-2)\Lambda_1^6 + (2\varepsilon+1+2\varepsilon^2)\Lambda_1^4 + (\varepsilon^2-2\varepsilon+1)\Lambda_1^2 - \varepsilon - \varepsilon^3 + 2\varepsilon^2$$

et

$$H_1(0,\Lambda_1) = \Lambda_1^{10} - 2\Lambda_1^6 + \Lambda_1^4 + \Lambda_1^2.$$

Nous pouvons maintenant calculer $K_1(\Lambda_1)$ et $K_2(\Lambda_2)$ pour vérifier que la variable X est un élément bien séparant.

$$K_1(\Lambda_1) \text{ est proportionnel à } -\Lambda_1^8 + 2\Lambda_1^4 - \Lambda_1^2 - 1$$
$$K_2(\Lambda_2) = F(0,\Lambda_2).$$

Puisque K_1 divise $H_1(0,\Lambda_1)$ et $K_2 = H_1(0,\Lambda_1)$, X est un élément bien séparant.

6.3 Algorithme théorique

Dans cette section, nous décrivons l'algorithme obtenu à partir des résultats des deux sections précédentes et qui calcule au moins un point par composante connexe sur une variété algébrique réelle définie par un système $S = (P_1,\ldots,P_k)$ d'équations polynomiales dans $K[X_1,\ldots,X_n]$. On se ramène au cas d'une seule équation en étudiant l'hypersurface définie par $P = P_1^2 + \ldots + P_k^2 = 0$. Introduisons la variable X_{n+1} et les infinitésimaux $\zeta,\varepsilon = 1/\Omega$, posons

$$Q = P^2 + (X_1^2 + \ldots + X_n^2 + X_{n+1}^2 - 1/\varepsilon^2)^2$$

et considérons l'hypersurface définie par :

$$Q_1 = (1-\zeta)Q + \zeta\left(X_1^{2(d_1+1)} + X_2^{2(d_2+1)} + \ldots + X_n^{2d_n+1} + X_{n+1}^6 - (n+1)1/\varepsilon^{2(d_1+1)}\right) = 0,$$

où $d_1,\ldots,d_{n+1}$ sont les degrés totaux de Q en $X_1,\ldots,X_{n+1}$. Sans nuire à la généralité, on suppose que $d_1 \geq \ldots \geq d_n \geq d_{n+1}$.
La famille

$$Q_1,\frac{\partial Q_1}{\partial X_2},\ldots,\frac{\partial Q_1}{\partial X_{n+1}}$$

est une base de Gröbner pour l'ordre du degré lexicographique [13, 12, 87] avec $X_1 > \ldots > X_n > X_{n+1}$, on peut dans un premier temps appliquer l'algorithme **LRB** à cette famille. On obtient une liste de Représentations Univariées Rationnelles à coefficients dans $K(\varepsilon)$ représentant les limites des points critiques de la fonction de projection sur l'axe X_1 restreinte à $V(Q_1)$ lorsque ζ tend vers 0. On obtient alors au moins un point par composante connexe sur $V(Q)$. D'après [13, 12, 87], la projection de ces points sur $R\langle\varepsilon\rangle^n$ donne au moins un point par composante connexe sur $V(P)\bigcap R\langle\varepsilon\rangle^n$. D'après la remarque 6.1, ces points sont représentés par des Représentations Univariées Rationnelles

$$\begin{cases} f(\varepsilon,T) = 0 \\ g_0(\varepsilon,T)X_1 - g_n(\varepsilon,T) = 0 \\ \vdots \\ g_0(\varepsilon,T)X_n - g_n(\varepsilon,T) = 0 \end{cases}$$

dont le polynôme f est square-free, donc on peut inverser g_0 modulo f et obtenir une base de Gröbner lexicographique :

$$\begin{cases} f(\varepsilon,T) = 0 \\ X_1 - g_n(\varepsilon,T)g_0(\varepsilon,T)\mathrm{modulo} f = 0 \\ \vdots \\ X_n - g_n(\varepsilon,T)g_0(\varepsilon,T)\mathrm{modulo} f = 0 \end{cases}$$

On peut alors réappliquer l'algorithme **LRB** aux bases de Gröbner ainsi obtenues.

Algorithme CPM

- **Entrée :** Un système $S = (P_1,\ldots,P_k)$ d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** Une liste de Représentations Univariées Rationnelles représentant au moins un point par composante semi-algébriquement connexe de $V(S)$.

1. Poser $P := P_1^2 + \ldots + P_k^2$.
2. Introduire une nouvelle variable X_{n+1} et poser $Q := P + (X_1^2 + \ldots + X_n^2 + X_{n+1}^2 - 1/\varepsilon^2)^2$.
3. Poser $Q_1 := (1-\zeta)Q + \zeta(X_1^{2(d_1+1)} + \ldots + X_n^{2(d_n+1)} + X_{n+1}^6 - (n+1)1/\varepsilon^{2(d_1+1)})$, où $d_1,\ldots,d_n,d_{n+1}$ sont les degrés totaux de P_1 en $X_1,\ldots,X_n,X_{n+1}$ tels que $d_1 \geq \ldots \geq d_{n+1}$.
4. Calculer les dérivées partielles $\frac{\partial Q_1}{\partial X_1},\ldots,\frac{\partial Q_1}{\partial X_n}$.
5. Calculer une Représentation Univariée Rationnelle à coefficients dans $K(\varepsilon)\langle\zeta\rangle$ de la base de Gröbner $[Q_1,\frac{\partial Q_1}{\partial X_1},\ldots,\frac{\partial Q_1}{\partial X_n}]$, associée à un élément bien séparant et calculer les limites des racines bornées de cette base lorsque ζ tend vers 0 en utilisant l'algorithme LRB.
6. Pour chaque Représentation Univariée Rationnelle $(f(\varepsilon,T),g_0(\varepsilon,T),g_1(\varepsilon,T),\ldots,g_{n+1}(\varepsilon,T))$ à coefficients dans $K(\varepsilon)$ retournée par l'étape précédente, calculer l'inverse Q de g_0 modulo f, et utiliser LRB sur la base de Gröbner lexicographique $(f(\varepsilon,T),X_1 - (Qg_1\mathrm{modulo} f),\ldots,X_n - (Qg_n\mathrm{modulo} f))$.
7. Retourner les RUR obtenues pour lesquelles le premier polynôme a des racines réelles.

Analyse expérimentale :

De manière à tester la taille des données intermédiaires apparaissant au cours d'un tel algorithme, nous avons simulé celui-ci en Maple sur le système d'équations polynomiales suivant :

$$\begin{cases} x^2 + y^2 + z^2 - 1 = 0 \\ xyz - 1 = 0 \end{cases}$$

Notons que ce système d'équations est très simple et que l'algorithme de Décomposition Cylindrique Algébrique parvient à le résoudre. Nous obtenons en premier lieu le polynôme nommé P dans l'algorithme ci-dessus :

```
P:=x^2*y^2*z^2+x^4+2*x^2*y^2+2*x^2*z^2+y^4+2*y^2*z^2+z^4-2*x*y*z-2*x^2-2*y^2-2*z^2+2
```

Notons que le degré total de ce polynôme est 6 alors que le degré total des polynômes de départ est borné par 3. Remarquons aussi que l'hypersurface ainsi considérée contient systématiquement une infinité de points singuliers.

Voici le polynôme Q que nous calculons en rajoutant la variable t :

```
Q:=x^2*y^2*z^2+2*x^4+4*x^2*y^2+4*x^2*z^2+2*x^2*t^2+2*y^4+4*y^2*z^2+2*y^2*t^2+2*z^4
```

```
+2*z^2*t^2+t^4-2*x*y*z-2/eps^2*x^2-2*x^2-2*y^2-2/eps^2*y^2-2/eps^2*z^2-2*z^2-
2/eps^2*t^2+2+1/eps^4
```

Remarquons que l'hypersurface définie par Q contient elle aussi une infinité de singularités. Ainsi, la déformation infinitésimale par ζ est inévitable.

Nous obtenons alors le polynôme Q_1 :

```
Q1:=zeta*x^14+zeta*y^13+zeta*z^13+zeta*t^9+x^2*y^2*z^2-zeta*x^2*y^2*z^2-2*zeta*x^4+
2*x^4+4*x^2*y^2-4*zeta*x^2*y^2+4*x^2*z^2-4*zeta*x^2*z^2-2*zeta*x^2*t^2+2*x^2*t^2+
2*y^4-2*zeta*y^4-4*zeta*y^2*z^2+4*y^2*z^2+2*y^2*t^2-2*zeta*y^2*t^2-2*zeta*z^4+
2*z^4+2*z^2*t^2-2*zeta*z^2*t^2-zeta*t^4+t^4-2*x*y*z+2*zeta*x*y*z-2*x^2+
2*zeta/eps^2*x^2+2*zeta*x^2-2/eps^2*x^2-2/eps^2*y^2-2*y^2+2*zeta/eps^2*y^2+
2*zeta*y^2+2*zeta/eps^2*z^2-2/eps^2*z^2+2*zeta*z^2-2*z^2+2*zeta/eps^2*t^2-
2/eps^2*t^2-56*zeta/eps-4005*zeta/eps^4-12012*zeta/eps^6-13728*zeta/eps^7-
12012*zeta/eps^8-8008*zeta/eps^9-4004*zeta/eps^10-1456*zeta/eps^11-364*zeta/eps^12-
56*zeta/eps^13-4*zeta/eps^14+2-364*zeta/eps^2-1456*zeta/eps^3-8008*zeta/eps^5+
1/eps^4-6*zeta
```

Après ces manipulations, nous obtenons directement une base de Gröbner pour l'ordre du degré lexicographique avec $x > y > z > t$. On la note $\mathcal{G}$ et nous l'affichons ci-dessous :

```
[zeta*x^14+zeta*y^13+zeta*z^13+zeta*t^9+x^2*y^2*z^2-zeta*x^2*y^2*z^2-2*zeta*x^4+
2*x^4+4*x^2*y^2-4*zeta*x^2*y^2+4*x^2*z^2-4*zeta*x^2*z^2-2*zeta*x^2*t^2+2*x^2*t^2+
2*y^4-2*zeta*y^4-4*zeta*y^2*z^2+4*y^2*z^2+2*y^2*t^2-2*zeta*y^2*t^2-2*zeta*z^4+
2*z^4+2*z^2*t^2-2*zeta*z^2*t^2-zeta*t^4+t^4-2*x*y*z+2*zeta*x*y*z-2*x^2+
2*zeta/eps^2*x^2+2*zeta*x^2-2/eps^2*x^2-2/eps^2*y^2-2*y^2+2*zeta/eps^2*y^2+
2*zeta*y^2+2*zeta/eps^2*z^2-2/eps^2*z^2+2*zeta*z^2-2*z^2+2*zeta/eps^2*t^2-
2/eps^2*t^2-56*zeta/eps-4005*zeta/eps^4-12012*zeta/eps^6-13728*zeta/eps^7-
12012*zeta/eps^8-8008*zeta/eps^9-4004*zeta/eps^10-1456*zeta/eps^11-
364*zeta/eps^12-56*zeta/eps^13-4*zeta/eps^14+2-364*zeta/eps^2-1456*zeta/eps^3-
8008*zeta/eps^5+1/eps^4-6*zeta,

13*zeta*y^12+2*x^2*y*z^2-2*zeta*x^2*y*z^2+8*x^2*y-8*zeta*x^2*y+8*y^3-8*zeta*y^3+
8*y*z^2-8*zeta*y*z^2+4*y*t^2-4*zeta*y*t^2-2*x*z+2*zeta*x*z-4/eps^2*y+4*zeta*y+
4*zeta/eps^2*y-4*y,

13*zeta*z^12+2*x^2*y^2*z-2*zeta*x^2*y^2*z+8*x^2*z-8*zeta*x^2*z+8*y^2*z-
8*zeta*y^2*z+8*z^3-8*zeta*z^3+4*z*t^2-4*zeta*z*t^2-2*x*y+2*zeta*x*y-4/eps^2*z+
4*zeta*z+4*zeta/eps^2*z-4*z,

9*zeta*t^8+4*x^2*t-4*zeta*x^2*t+4*y^2*t-4*zeta*y^2*t-4*zeta*z^2*t+4*z^2*t-4*zeta*t^3+
4*t^3-4/eps^2*t+4*zeta/eps^2*t]
```

En observant le degré des monômes dominants, on trouve que le degré de l'idéal défini par cette base de Gröbner est 16128. Nous devons alors en calculer une Représentation Univariée Rationnelle. Il est évident que même sur les entiers, un tel calcul est trop coûteux. Dans le cas présent, nous devons effectuer ces calculs dans $\mathbb{Q}\langle\varepsilon,\zeta\rangle$, ce qui complique le problème. Il n'est donc pas étonnant de constater que ce calcul ne passe pas. En posant l'hypothèse que pour une entrée de taille plus importante, le temps de calcul est plus important, il apparait clairement que cet algorithme ne pourra pas donner de bons résultats en pratique. Analysons les étapes bloquantes :

- sur l'exemple ci-dessus, le degré de l'idéal zéro-dimensionnel est un facteur bloquant. Soit d un entier qui borne le degré des polynômes du système d'entrée. Les degrés des polynômes P et Q calculés par l'algorithme sont alors bornés par $2d$. En bornant

d_1 par d, on trouve que le degré de Q_1 est borné par $2d(2d+1)$. Comme on a rajouté une variable, on trouve que le degré du système zéro-dimensionnel produit est alors borné par $6(4d)^n$, ce qui donne sur notre exemple simple 20736. On constate que l'élévation au carré du pas 1 de l'algorithme ainsi que la déformation du pas 3 sont responsables de la taille de ces systèmes zéro-dimensionnels. Il est clair que la déformation du pas 3 de l'algorithme engendre une croissance de degré pour se ramener sans calcul à une base de Gröbner.

- Remarquons que même si on ne considère en entrée que des hypersurfaces, cette croissance de degrés intervient : le pas 2 de l'algorithme en est responsable. Or, cette déformation est rendue nécessaire par l'usage de la fonction de projection sur un axe : on doit se ramener au cas d'une hypersurface compacte pour qu'elle atteigne ses valeurs critiques sur chacune des composantes semi-algébriquement connexes.
- Supposons que les méthodes de résolution des systèmes zéro-dimensionnels permettent de résoudre des problèmes dont la taille est de l'ordre de ce que nous avons obtenu. Notons qu'au pas 2 nous n'avons introduit qu'un seul infinitésimal. En revanche, il est clair que l'hypersurface obtenue contient une infinité de singularités. Ceci implique – en partie – l'introduction de l'infinitésimal dans le pas 3 de l'algorithme. Nous devrions alors travailler sur une arithmétique à deux infinitésimaux, dont les opérations élémentaires sont bien plus coûteuses que sur les entiers.

On peut retenir de ces trois points que la mise en œuvre algorithmique des méthodes de points critiques n'est pas encore suffisamment aboutie pour en espérer des implantations efficaces. Le problème réside essentiellement dans les manipulations effectuées pour passer du cas général de variétés algébriques réelles définies par un système d'équations quelconque au cas très particulier d'une hypersurface réelle lisse et compacte. Cette démarche se retrouve d'ailleurs dans la plupart des algorithmes proposés dans le cadre de la méthode des points critiques.

6.4 Conclusions

Pour obtenir une bonne complexité théorique, il a fallu effectuer une déformation infinitésimale qui permette d'éviter un calcul de base de Gröbner, mais qui provoque une croissance des données intermédiaires apparaissant en cours de calcul (notamment le degré de l'idéal zéro-dimensionnel étudié) qui empêche l'obtention d'une implantation efficace. À cette difficulté vient s'ajouter celle d'effectuer des calculs avec infinitésimaux.

Il est bien connu (nous rappelons ce résultat dans le chapitre suivant) que si P définit une hypersurface contenant une infinité de points singuliers, l'hypersurface définie par $P - \zeta = 0$ où ζ est un infinitésimal ne contient pas de points singuliers. Si on s'autorise un calcul de base de Gröbner, on peut remplacer la deuxième déformation infinitésimale de l'algorithme **CPM** par celle mentionnée ci-dessus. On obtient aussi une hypersurface lisse dont le lieu réel est compact. Il nous faut alors calculer les points critiques de la fonction de projection sur un axe de coordonnée. On obtient un idéal de degré 392 sur l'exemple que nous avons considéré dans le chapitre précédent. Comparativement à l'idéal que nous devions étudier et qui était de degré 16128, le progrès est remarquable. Notons que la méthode est prometteuse : nous nous trouvons dans les hypothèses d'application des résultats de [9, 10]. Dans un cadre d'évaluation, nous obtenons donc une complexité simplement exponentielle pour la phase de résolution du système zéro-dimensionnel. Il reste néanmoins la phase de calcul des limites des racines bornées du système lorsque les deux infinitésimaux tendent vers 0 et une phase de comptage et d'isolation des racines réelles d'un polynôme univarié. A notre connaissance, ces problèmes ne sont pas traités dans ce cadre.

Finalement, on peut réduire le degré du système zéro-dimensionnel en considérant une déformation infinitésimale bien connue mais qui implique un calcul de bases de Gröbner. En revanche, nous devons *systématiquement* introduire deux infinitésimaux. Le premier permet de se ramener au cas compact mais a introduit des singularités. On ne peut donc pas éviter la seconde déformation infinitésimale que nous préconisons ci-dessus.
Une première approche du problème consiste à revenir aux sources de la méthode des points critiques et ne déformer l'hypersurface qu'une seule fois et uniquement lorsque c'est nécessaire tout en gardant un bon contrôle sur la taille des données intermédiaires.

Deuxième partie

Nouveaux Algorithmes

Chapitre 1

Le cas des hypersurfaces

Résumé

En collaboration avec F. Rouillier et M.-F. Roy, nous utilisons la méthode des points critiques avec la fonction distance. Un nouvel algorithme qui donne au moins un point par composante connexe sur une hypersurface réelle quelconque est obtenu. Il introduit un seul infinitésimal uniquement dans les cas où l'hypersurface contient une infinité de singularités. On ramène alors l'étude à des systèmes zéro-dimensionnels à coefficients dans $K(\varepsilon)$ dont on doit calculer les limites des racines bornées lorsque ε tend vers 0. En tirant profit des propriétés de ces systèmes, nous donnons des outils permettant de calculer efficacement des bases de Gröbner et un élément séparant pour ces systèmes ainsi qu'une Représentation Univariée Rationnelle (en collaboration avec E. Schost). Puis, nous donnons un nouvel algorithme pour calculer les limites des racines bornées des systèmes obtenus.

Sommaire

Introduction

Dans le but de n'introduire qu'un seul infinitésimal et uniquement lorsque c'est nécessaire, nous reprenons une idée classique [89] (voir aussi [56, 80]) et qui consiste à appliquer la méthode des points critiques avec la fonction distance (au lieu de la fonction de projection sur un axe [55, 12, 13, 9, 10, 11]) qui atteint ses minima sur toute composante connexe d'une variété algébrique réelle. Ainsi, dans les cas où l'hypersurface étudiée est lisse – ce qui est facilement vérifiable – on peut espérer pouvoir utiliser la fonction distance pour en donner au moins un point par composante connexe sans faire la moindre déformation infinitésimale. Reprenons l'exemple du chapitre précédent illustrant notre étude :

$$\begin{cases} x^2 + y^2 + z^2 - 1 = 0 \\ xyz - 1 = 0 \end{cases}$$

En faisant la somme des carrés des polynômes du système, nous obtenons l'hypersurface $\mathcal{H}$ contenant une infinité de singularités et définie par le polynôme :

```
P:=x^2*y^2*z^2+x^4+2*x^2*y^2+2*x^2*z^2+y^4+2*y^2*z^2+z^4-2*x*y*z-2*x^2-2*y^2-2*z^2+2
```

L'hypersurface de C^n définie par $P - \varepsilon = 0$ est lisse. Choisissons le point $A = (1{,}0{,}0)$ et étudions le système d'équations polynomiales :

$$P - \varepsilon = 0, \overrightarrow{\mathrm{grad}}_M\,(P)//\,\overrightarrow{AM}$$

qui définit un ensemble algébrique contenant les points critiques de la fonction distance au point A restreinte à $\mathcal{H}$. Cet ensemble algébrique est un ensemble fini de points. Le degré de l'idéal engendré par ce système d'équations est 216, ce qui est encore inférieur à celui étudié en utilisant la fonction de projection un axe (voir les conclusions du chapitre précédent), alors que nous n'avons introduit qu'un seul infinitésimal. Bien évidemment, le degré de cet idéal dépend du choix du point A, mais il est raisonnable d'estimer que les variations de degré en fonction du choix du point A ne devraient pas être sensibles. Pour valider notre démarche, nous devons répondre aux questions suivantes :

- Etant donnée une hypersurface lisse définie par $P = 0$, peut-on toujours trouver un point A tel que le système d'équations

  $$P = 0, \quad \overrightarrow{\mathrm{grad}}_M\,(P)//\,\overrightarrow{AM}$$

  soit zéro-dimensionnel ?
- Dans le cas d'une hypersurface $\mathcal{H}$ définie par $P = 0$ contenant une infinité de singularités, comment obtenir au moins un point par composante connexe sur $\mathcal{H}$ à partir des points critiques de la fonction distance restreinte à l'hypersurface définie par $P - \varepsilon = 0$?

Dans ce chapitre, nous décrivons un travail effectué en collaboration avec F. Rouillier et M.-F. Roy [84] et qui répond à ces questions. Dans la première section, on montre que dans le cas d'une hypersurface ne contenant au plus qu'un nombre fini de points singuliers, on peut choisir un point $A \in K^n$ tel que l'ensemble algébrique défini par :

$$P(M) = 0, \quad \overrightarrow{grad}_M\,(P)//\,\overrightarrow{AM}$$

(où P est un polynôme) rencontre chaque composante semi-algébriquement connexe de l'hypersurface réelle et est zéro-dimensionnel. Dans un deuxième temps, nous montrons que lorsque l'hypersurface contient une infinité de points singuliers, on peut se ramener au cas précédent en étudiant l'hypersurface définie par $P-\varepsilon = 0$ où ε est un infinitésimal.

Ainsi, dans les cas où l'hypersurface contient une infinité de points singuliers, son étude est ramenée à celle d'un système zéro-dimensionnel à coefficients dans $K(\varepsilon)$. La deuxième section de ce chapitre montre comment calculer une base de Gröbner à coefficients infinitésimaux et trouver un élément séparant pour de tels systèmes en n'effectuant des calculs que sur le corps des rationnels.

Nous avons complété ce travail dans deux directions. En effet, le calcul de Représentations Univariées Rationnelles à coefficients infinitésimaux, tout comme les algorithmes calculant les limites des solutions bornées décrites par ces Représentations Univariées Rationnelles (voir chapitre 6 de la partie I) se sont révélées trop peu efficaces en pratique pour pouvoir résoudre des exemples significatifs.

La troisième section de ce chapitre décrit un travail plus récent, effectué en collaboration avec E. Schost, dont le but était le calcul efficace de Représentations Univariées Rationnelles à coefficients infinitésimaux. Pour cela, nous remarquons que les systèmes polynomiaux zéro-dimensionnels à coefficients infinitésimaux peuvent être vus comme des systèmes à un seul paramètre. Les méthodes basées sur des techniques d'évaluation et de remontée de Hensel [47] peuvent alors être avantageuses. Nous avons donc adapté un algorithme probabiliste de E. Schost (voir [88]) pour calculer ces Représentations Univariées Rationnelles à coefficients infinitésimaux.

Dans la quatrième section de ce chapitre, nous décrivons un nouvel algorithme calculant les racines bornées des solutions décrites par la Représentation Univariée Rationnelle précédemment calculée. Pour cela, nous calculons des développements en séries de Puiseux afin de vérifier que l'élément séparant choisi est un élément bien séparant. Comparativement à l'algorithme exposé dans le chapitre 6 de la partie I, cet algorithme permet d'éviter le calcul systématique des polynômes caractéristiques de la multiplication par chacune des variables.

La dernière section de ce chapitre est consacrée à la validation expérimentale de nos algorithmes. Nous constaterons que cet algorithme apporte des progrès significatifs dans les cas où l'hypersurface contient au plus un nombre fini de points singuliers. Dans les cas où l'ensemble des singularités contient une infinité de points complexes, le bilan n'est pas positif. L'algorithme proposé ne permet pas de réaliser des progrès significatifs comparativement à l'algorithme de Décomposition Cylindrique.

1.1 L'algorithme

On note K un corps réel clos, R sa clôture réelle et C sa clôture algébrique. Soit P un polynôme irréductible dans $K[X_1,\ldots,X_n]$, et $A = (a_1,\ldots,a_n)$ un point de K^n tel que $P(A) \neq 0$. On donne dans cette section une description d'une variété algébrique contenant les points de $V(P)$ qui sont à distance minimale du point A. Considérons l'ensemble algébrique $\mathcal{C}(V(P),A)$ défini par le système d'équations polynomiales :

$$P(M) = 0,\ \ \overrightarrow{\mathrm{grad}}_M\ (P) // \overrightarrow{AM}\ ,$$

où $M = (X_1,\ldots,X_n)$. Le fait que le vecteur $\overrightarrow{\mathrm{grad}}_M\ (P)$ et le vecteur $\overrightarrow{AM}$ sont parallèles est exprimé en imposant aux mineurs d'ordre (2,2) de la matrice dont les colonnes sont les vecteurs $\overrightarrow{\mathrm{grad}}_M\ (P)$ et $\overrightarrow{AM}$ de s'annuler. On note $\Delta_A(P)$ la liste de ces mineurs, pour un point A et un polynôme P donné.
Un point M est *singulier* dans $V(P)$ si

$$P(M) = 0,\ \ \overrightarrow{\mathrm{grad}}_M\ (P) = \overrightarrow{0}\ .$$

Ainsi, l'ensemble des singularités de $V(P)$ est inclus dans $\mathcal{C}(V(P),A)$.

Il est clair que toute composante semi-algébriquement connexe de $\mathcal{C}(V(P),A) \bigcap R^n$ est contenue dans une composante semi-algébriquement connexe de $V(P) \bigcap R^n$. Le lemme suivant nous servira à prouver la correction des algorithmes que nous proposons.

Lemme 1.1 *Toute composante semi-algébriquement connexe de $V(P) \bigcap R^n$ a une intersection non vide avec $\mathcal{C}(V(P),A) \bigcap R^n$.*

Preuve: Soit A un point de K^n, D une composante semi-algébriquement connexe de $V(P) \bigcap R^n$ et M un point de D à distance minimale du point A. On distingue deux cas :

- Si M est un point singulier de $V(P)$, alors le vecteur $\overrightarrow{grad}_M\ (P)$ est nul et M appartient à $\mathcal{C}(V(P),A)$.
- Supposons que M ne soit pas un point singulier de $V(P)$ (le vecteur $\overrightarrow{grad}_M\ (P)$ est donc non nul, et il existe un hyperplan tangent à D en M normal au vecteur $\overrightarrow{grad}_M\ (P)$) et notons d la distance de A à M. Considérons maintenant la boule $\mathcal{B}$ de centre A et de rayon d. Puisque M est un point de D à distance minimale de A, la boule $\mathcal{B}$ et la composante D sont tangentes au point M, elles ont donc le même hyperplan tangent en M et le vecteur $\overrightarrow{grad}_M\ (P)$ est bien parallèle au vecteur $\overrightarrow{AM}$. Le point M appartient doncà $\mathcal{C}(V(P),A)$.

∎

On distingue alors trois cas :

1. **Premier Cas** : l'ensemble algébrique $\mathcal{C}(V(P),A)$ est constitué d'un nombre fini de points (remarquons que ceci implique que $V(P)$ ne contient pas une infinité de points singuliers);
2. **Deuxième Cas** : l'ensemble algébrique $\mathcal{C}(V(P),A)$ contient une infinité de points mais $V(P)$ n'a au plus qu'un nombre fini de points singuliers. Nous allons alors montrer que l'on se ramène à l'étude d'un système d'équations polynomiales ne contenant qu'un nombre fini de solutions en changeant de point A.

3. **Troisième Cas** : la variété algébrique $V(P)$ contient une infinité de points singuliers. Nous allons alors montrer qu'en opérant une seule déformation infinitésimale, nous pouvons nous ramener à l'étude d'un système d'équations polynomiales à coefficients dans $K(\varepsilon)$ ne contenant qu'un nombre fini de solutions.

Exemple 1.1 *Considérons l'hypersurface définie par l'équation*

$$x^2 + y^2 - 1 = 0$$

- *si on choisit le point* $A = (1,1)$, $\mathcal{C}(V(P),A)$ *est défini par :*

$$\begin{cases} x^2 + y^2 - 1 = 0 \\ x - y = 0 \end{cases}$$

  Ce système a deux solutions $(\sqrt{2}/2,\sqrt{2}/2)$ *et* $(-\sqrt{2}/2,-\sqrt{2}/2)$. *Nous sommes dans le premier cas.*
- *si on choisit le point* $A = (0,0)$, $\mathcal{C}(V(P),A)$ *est l'hypersurface toute entière, mais il facile de voir que l'ensemble des points singuliers de cette hypersurface est vide. Nous sommes dans le second cas.*

Considérons maintenant l'hypersurface définie par

$$x^2 - y^2z^2 + z^3 = 0$$

En choisissant le point $A = (1,2,3)$, $\mathcal{C}(V(P),A)$ *est défini par le système suivant :*

$$\begin{cases} x^2 - y^2z^2 + z^3 = 0 \\ 2xy - 4x + 2yz^2x - 2yz^2 = 0 \\ -2yz^3 + 3yz^2 + 2y^3z - 4y^2z + 6z^2 = 0 \\ 2xz - 6x + 2y^2zx - 2y^2z - 3z^2x + 3z^2 = 0 \end{cases}$$

Si on calcule une base lexicographique de ce système pour l'ordre $x > y > z$, *on obtient :*

```
[1619623935890321149608*x+5398746452967773716536*y^2*z-2655259823035388651O4*z^17+_
10013693624911756548960*z^16-16401086283785572629200*z^15+27022636119689087494080*z^14+_
```

Ici, $\mathcal{C}(V(P),A)$ est une courbe algébrique de dimension 1 et de degré 1*. Il est facile de voir que l'ensemble des points singuliers est la courbe définie par le système d'équations :*

$$\begin{cases} z = 0 \\ x = 0 \end{cases}$$

Nous sommes dans le troisième cas.

1.1.1 Premier Cas

Le lemme 1.1 donne immédiatement un algorithme permettant de décider si $V(P) \bigcap R^n$ est vide et en donne au moins un point par composante semi-algébriquement connexe lorsque la variété $\mathcal{C}(V(P),A)$ est un ensemble fini de points. On définit les routines suivantes, dont nous nous servirons dans l'algorithme ci-dessous :

- **Gröbner** : prend en entrée un système d'équations polynomiales S et calcule une base de Gröbner de $\langle S \rangle$.
- **Dim** : prend en entrée une base de Gröbner et renvoie la dimension de l'idéal engendré par la base de Gröbner.
- **RUR** : Calcule une Représentation Univariée Rationnelle d'un système zéro-dimensionnel à partir d'une base de Gröbner.
- **RRCI** : compte et isole les racines réelles d'un polynôme univarié.

Nous obtenons l'algorithme **HA1** (**H**ypersurfaces **A**lgorithme **1**) ci-dessous.

Algorithme HA1

- **Entrée:** un polynôme $P \in K[X_1,\ldots,X_n]$ et un point $A \in K^n$.
- **Sortie:** *no answer* si $\mathcal{C}(V(P),A)$ contient un nombre infini de points, dans les cas contraires, on renvoie *false* si $V(P) \bigcap R^n$ est vide, ou *true* et au moins un point par composante semi-algébriquement connexe si $V(P) \bigcap R^n$ n'est pas vide.

1. $G :=$ Grobner$([P,\partial P/\partial X_1,\ldots,\partial P/\partial X_n])$,
2. Si $\mathrm{Dim}(G) > 0$, alors retourner *no answer*.
3. Sinon calculer $(f_u(T),g_0(T),g_1(T),\ldots,g_n(T)) :=$ RUR(G).
4. Utiliser RRCI sur $f_u(T)$. Si il a au moins une racine réelle renvoyer *true* et RUR(G), sinon renvoyer *false*.

1.1.2 Deuxième Cas

On note $\mathcal{G}$ le système d'équations polynomiales

$$P(M) = 0,\ \overrightarrow{\mathrm{grad}}_M\,(P) = \overrightarrow{0}\ .$$

La variété algébrique $V(\mathcal{G})$ est donc l'ensemble des points singuliers de $V(P)$. Dans cette section, on suppose que $V(\mathcal{G})$ est un ensemble fini de points et que la variété $\mathcal{C}(V(P),A)$ contient une infinité de points.

Définition 1.1 *On dit que $A = (a_1,\ldots,a_n) \in C^n$ est un* bon centre *pour P si $\mathcal{C}(V(P),A)$ est fini.*

Dans cette section, nous allons prouver que l'ensemble des points $A \in C^n$ qui ne sont pas de bons centres pour P est contenu dans un sous-ensemble algébrique strict de C^n. En conséquence, l'ensemble des points de K^n qui ne sont pas de bons centres pour P est un sous-ensemble algébrique strict de K^n. Soit

$$\begin{gathered} Q_1 = \lambda \frac{\partial P}{\partial X_1} - X_1, \\ \vdots \\ Q_n = \lambda \frac{\partial P}{\partial X_n} - X_n. \end{gathered}$$

et considérons l'ensemble de points $\mathcal{C}'(A)$ défini par le système d'équations polynomiales :

$$\begin{gathered} P = 0, \\ Q_1 + a_1 = 0, \\ \vdots \\ Q_n + a_n = 0 \end{gathered}$$

Il est évident que $\mathcal{C}(V(P),A) = V(\mathcal{G}) \cup \pi(\mathcal{C}'(A))$ où π est la projection de $(x_1,\ldots,x_n,\lambda)$ sur $(x_1,\ldots,x_n)$.

Lemme 1.2 *Soit P un polynôme de $K[X_1,\ldots,X_n]$ et*

$$\mathcal{H} = \{(M,\lambda) \in C^{n+1} \mid P(M) = 0, \overrightarrow{\mathrm{grad}}_M\ (P) \neq \overrightarrow{0}\}.$$

$$\begin{gathered} \mathcal{A} = \{A = (a_1,\ldots,a_n) \in C^n \mid \\ \mathcal{H} \cap V(Q_1 + a_1,\ldots,Q_n + a_n, Jac(P,Q_1 + a_1,\ldots,Q_n + a_n)) \neq \emptyset\} \end{gathered}$$

est contenu dans un sous-ensemble algébrique strict de C^n.

Preuve: Soit F l'application de $\mathcal{H}$ sur C^n qui à (M,λ) associe $Q_1(M,\lambda),\ldots,Q_n(M,\lambda)$. Les valeurs critiques de F sont les points $A = (a_1,\ldots,a_n)$ de C^n tels que $V(Q_1 + a_1,\ldots,Q_n + a_n, Jac(P,Q_1 + a_1,\ldots,Q_n + a_n)) \neq \emptyset$. D'après le théorème de Sard algébrique sur C [73] on en déduit que $\mathcal{A}$ est un ensemble constructible de dimension strictement inférieure à n dans C^n. ■

On en déduit alors le corollaire suivant :

Corollaire 1.1 *Un point $A \notin \mathcal{A}$ est un bon centre pour P. De plus, $\mathcal{C}'(A)$ est un ensemble fini de points simples.*

Preuve: Soit $A = (a_1,\ldots,a_n) \notin \mathcal{A}$. Puisque le rang de la matrice jacobienne est maximal en les solutions du système d'équations polynomiales $\mathcal{C}'(A)$, ces solutions sont isolées et non singulières. Ainsi, $\mathcal{C}'(A)$ est un ensemble fini de points simples. ■

Ainsi, si le point A est choisi en dehors d'un ensemble algébrique, il est un bon centre pour P. Il faut maintenant montrer que par un procédé algorithmique on peut trouver un tel point A sans calculer l'ensemble $\mathcal{A}$.

Lemme 1.3 *Soit g un polynôme non identiquement nul dans $R[X_1,\ldots,X_n]$ de degré d. On peut alors choisir un point A dans $\{0,\ldots,d\}^n$ tel que $g(A) \neq 0$.*

Preuve : On raisonne par récurrence sur n. Le résultat est évident pour $n = 1$. Supposons qu'il soit vrai pour $n-1$ et soit g un polynôme dans $R[X_1,\ldots,X_n]$. On peut choisir $(a_1,\ldots,a_{n-1}) \in K^{n-1}$ tel que le coefficient dominant de g (où g est vu comme un polynôme univarié à coefficients dans $R[X_1,\ldots,X_{n-1}]$) ne s'annule pas. Quand on spécialise $X_1,\ldots,X_{n-1}$ en $a_1,\ldots,a_{n-1}$ dans g, on obtient un polynôme univarié de degré d. ∎

Ainsi, on peut choisir successivement des valeurs de $A = (a_1,\ldots,a_n)$ dans une boîte $\{0,\ldots,d\}^n$ et garantir que si d est suffisamment grand, pour l'un de ces choix l'ensemble $\mathcal{C}'(A)$ est un ensemble fini de points simples. Puisque nous n'avons pas de borne précise sur le degré de l'ensemble algébrique définissant $\mathcal{A}$, il nous faut choisir successivement un point dans $\{0,\ldots,d\}^n$ jusqu'à ce que nous trouvions un bon centre pour P. Si tous les points de cette boite ont été testés sans que l'on puisse trouver un bon centre, on agrandit la boite.

La procédure **CC** (**C**hangement de **C**entre) ci-dessous trouve un bon centre pour P.

Algorithme CC

- **Entrée:** Un polynôme $P \in K[X_1,\ldots,X_n]$ tel que $V(\mathcal{G})$ est un ensemble fini de points.
- **Sortie:** Un point A tel que $\mathcal{C}(V(P),A)$ est un ensemble fini de points, et une base de Gröbner G de $\langle [P,\Delta_A(P)] \rangle$

1. Choisir un point A dans $\{0,\ldots,d\}^n \setminus \{0,\ldots,d-1\}^n$.
2. $G :=$ Grobner($[P,\Delta_A(P)]$).
3. Si $\mathrm{Dim}(G) = 0$, alors retourner A et G.
4. Si tous les points de $\{0,\ldots,d\}^n$ ont été testés, faire $d := d+1$.
5. retourner à 1.

Nous sommes maintenant en mesure de décrire l'algorithme **HA2** (**H**ypersurfaces **A**lgorithme **2**) qui décide du vide de $V(P) \bigcap R^n$ (et retourne au moins un point par composante semi-algébriquement connexe si $V(P) \bigcap R^n$ est non vide) dans les cas où l'ensemble algébrique $V(\mathcal{G})$ est un nombre fini de points.

Algorithme HA2

- **Entrée:** un polynôme $P \in K[X_1,\ldots,X_n]$.
- **Sortie:** *no answer* si $V(\mathcal{G})$ contient une infinité de points, *false* si $V(P)\bigcap R^n$ est vide, *true* et au point un point par composante semi-algébriquement connexe si $V(P)\bigcap R^n$ n'est pas vide.

1. $\widetilde{G}$:= Grobner($[P,\partial P/\partial X_1,\ldots,\partial P/\partial X_n]$),
2. Si $\mathrm{Dim}(\widetilde{G}) > 0$, alors retourner *no answer*.
3. $[A,G] := \mathrm{CC}(P)$,
4. Utiliser RRCI sur le premier polynôme de $RUR(G)$. Si il a au moins une racine réelle renvoyer *true* et RUR(G), sinon renvoyer *false*.

Notons qu'un point A choisi au hasard est en général un bon centre pour P.

1.1.3 Troisième Cas

Il nous faut maintenant traiter le cas où $V(P)$ contient une infinité de singularités. Une idée classique pour traiter ce cas est de procéder à une déformation infinitésimale sur $V(P)$ afin de se ramener au cas d'une variété lisse.

Si $S_\varepsilon \subset K[\varepsilon][X_1,\ldots,X_n]$ est un système d'équations polynomiales zéro-dimensionnel, on note $V_b(S_\varepsilon) \subset C\langle\varepsilon\rangle^n$ (resp. $V_{R,b}(S_\varepsilon) \subset R\langle\varepsilon\rangle^n$) l'ensemble des solutions bornées de S_ε, à coordonnées dans $\mathcal{V}_\varepsilon^n$ (resp. V_ε^n).

Remarque 1.1 *On peut faire les deux remarques suivantes :*

1. $\lim_0(V_b(S_\varepsilon)) = \lim_0(V_b(S_{-\varepsilon}))$, *où* $S_{-\varepsilon}$ *un système d'équations polynomiales obtenu en substituant* $-\varepsilon$ *à* ε *dans* S_ε*;*
2. $\lim_0(V_{R,b}(S_\varepsilon) \cup V_{R,b}(S_{-\varepsilon})) \subset \lim_0(V(S_\varepsilon)) \cap R^n$.

Le résultat suivant montre comment on se ramène au cas d'une hypersurface lisse :

Lemme 1.4 *Les ensembles algébriques définis par* $P-\varepsilon=0$ *(resp.* $P+\varepsilon=0$*) dans* C^n *sont des hypersurfaces lisses.*

Preuve : On note $\mathcal{G}'$ le système d'équations polynomiales $\overrightarrow{grad}_M\,(P) = \overrightarrow{0}$. La variété $V(\mathcal{G}')$ a un nombre fini de composantes connexes sur lesquelles le polynôme P est constant. Soit F la fonction de C^n dans C qui à M associe $P(M)$. L'ensemble des valeurs critiques de F est donc l'ensemble des valeurs que prend P sur les composantes connexes de $\mathcal{G}'$ et est donc fini. Donc l'ensemble des valeurs de t tel que le système d'équations polynomiales

$$P-t=0, \quad \overrightarrow{grad}_M\,(P) = \overrightarrow{0}$$

définit l'ensemble vide contient un ouvert de R, est constructible pour la topologie de Zariski et est non vide, il contient donc un infinitésimal ε. ∎

Il nous faut maintenant montrer quelles sont les relations entre les solutions de $P = 0$ et les solutions de $P - \varepsilon = 0$ et $P + \varepsilon = 0$.

Lemme 1.5

$$\lim_0(V_{R,b}(P - \varepsilon) \cup V_{R,b}(P + \varepsilon)) = \lim_0(V_b(P - \varepsilon)) \cap R^n = \\ = \{M \in R^n \mid P(M) = 0\}.$$

Preuve :

- Soit $M \in R^n$ un point de $V(P) \bigcap R^n$. Dans toutes les boules de centre M il existe un point N n'annulant pas le polynôme P. D'après le lemme de sélection des courbes, il est possible de trouver un chemin semi-algébriquement continu ϕ de $[0,1]$ dans R^n tel que $\phi(0) = M$. On a bien sur $P^2(\phi(x)) > 0$ pour $x \in]0,1]$. On note ϕ_ε l'extension de ϕ dans $R\langle\varepsilon\rangle$. D'après le théorème des valeurs intermédiaires, il existe y tel que $P(\phi_\varepsilon(y))^2 = \varepsilon^2$. Puisque $\lim_0(P(\phi_\varepsilon(y))) = P(\phi(\lim_0(y))) = 0$, $\lim_0(y) = 0$ et $\lim_0(\phi_\varepsilon(y)) = M$. Il est alors évident que $\phi_\varepsilon(y)$ est borné sur R.
- Soit $N \in C\langle\varepsilon\rangle^n$ un point borné tel que $P(N) - \varepsilon = 0$. On note $M = \lim_0(N)$. Alors, par continuité, on a $P(M) = 0$.

■

Maintenant que nous avons la correspondance entre les solutions de $P = 0$ et la réunion des solutions bornées de $P - \varepsilon = 0$ et celles de $P + \varepsilon = 0$, il nous faut montrer qu'en calculant les points critiques de la fonction distance à un point A de $V(P - \varepsilon)$ et de $V(P+\varepsilon)$ on va bien retrouver au moins un point par composante connexe sur $V(P) \bigcap R^n$. Soit $\mathcal{C}_\varepsilon(V(P-\varepsilon),A)$ l'ensemble algébrique défini par le système d'équations polynomiales

$$P = \varepsilon,\ \ \overrightarrow{\mathrm{grad}}_M\ (P) // \overrightarrow{AM}\ .$$

et $\mathcal{I}_{\varepsilon,A} = \langle[P - \varepsilon,\Delta_A(P - \varepsilon)]\rangle$.

Lemme 1.6 $\lim_0(V_b([P-\varepsilon,\Delta_A(P-\varepsilon)])) \cap R^n$ *intersecte chaque composante semi-algébriquement connexe de* $V(P) \bigcap R^n$.

Preuve : Soit A un point de K^n, D une composante semi-algébriquement connexe de $V(P) \bigcap R^n$ et $\mathcal{M}$ l'ensemble des points de D à distance minimale de A. Soit $r > 0$ un rationnel suffisament petit tel que l'ensemble semi-algébrique fermé borné

$$T = \{x \in R^n \mid \exists y \in \mathcal{M}\ \mathrm{dist}(x,\mathcal{M}) \leq r\}$$

n'intersecte pas $(V(P) \setminus D) \bigcap R^n$. D'après le lemme 1.5, il existe $N \in V_{R,b}(P^2 - \varepsilon^2)$ tel que $\lim_0(N) \in \mathcal{M}$. On note

$$T' = \{x \in R^n \mid \exists y \in \mathcal{M}\ \mathrm{dist}(x,\mathcal{M}) = r\}.$$

Remarquons que les points de $V_{R,b}(P^2 - \varepsilon^2) \cap T'$ sont infiniment proches des points de $V(P) \bigcap(T' \cap R^n)$ qui ne sont pas à distance minimale du point A. Aussi, la distance minimale au point A de $V(P^2 - \varepsilon^2) \bigcap(T \cap R^n)$ n'est pas obtenue sur T'. Ainsi,la distance minimale est obtenue en un point borné N qui est un point critique de la fonction distance à A restreinte à $V(P^2 - \varepsilon^2) \bigcap R^n$. Il est alors clair que $\lim_0(N) \in \mathcal{M}$. ■

D'après les résultats précédents, si on peut :

- trouver un point A tel que $\mathcal{C}_\varepsilon(V(P-\varepsilon),A)$ soit un ensemble fini de points;
- calculer les limites des solutions bornées d'un système d'équations polynomiales à coefficients dans $K(\varepsilon)$;

on sait décider du vide de $V(P)\bigcap R^n$ et donner au moins un point par composante semi-algébriquement connexe sur cette variété réelle lorsqu'elle n'est pas vide. Or, en appliquant le principe de transfert [20] aux résultats de la section précédente, on montre qu'on peut trouver $A \in K^n$ tel que $\mathcal{C}_\varepsilon(V(P-\varepsilon),A)$ est un ensemble fini de points.
On note **LRB** une routine qui :

- prend en entrée un système d'équations polynomiales S_ε de $K[\varepsilon][X_1,\ldots,X_n]$ de dimension nulle et une base de Gröbner G_ε de $I_\varepsilon = \langle S_\varepsilon\rangle$ pour un ordre quelconque sur les monômes.
- renvoie une liste de Représentations Univariées Rationnelles à coefficients dans K codant les limites des solutions bornées de S_ε.

Dans le chapitre 6 de la partie I, nous décrivons un algorithme dont les spécifications sont identiques à cette routine. Nous obtenons l'algorithme **HA3** (**H**ypersurfaces **A**lgorithme 3) ci-dessous :

Algorithme HA3

- **Entrée:** un polynôme $P \in K[X_1,\ldots,X_n]$.
- **Sortie:** *false* si $V(P)\bigcap R^n$ est vide, *true* et au moins un point par composante semi-algébriquement connexe de $V(P)\bigcap R^n$ est non vide.

1. $\widetilde{G}$:= Grobner($\mathcal{G}$), si Dim$\widetilde{G} > 0$, aller à 3.
2. $[A,G]$:= CC(P) et *rur* := RUR(G). Si RRCI trouve au moins une racine réelle au premier premier polynôme de *rur*, alors renvoyer *true* et la RUR calculée, sinon renvoyer *false*.
3. $[A,G_\varepsilon]$:= CC($P-\varepsilon$)
4. list := LRB(G_ε). Pour chaque Représentation Univariée Rationnelle de list utiliser RRCI pour compter et isoler les racines réelles du premier polynôme. Si il en existe, mettre la RUR correspondante dans result. Si result $\neq$ [] renvoyer *true* et result sinon renvoyer *false*.

1.2 Optimisations

Nous avons décrit dans les sections précédentes un algorithme permettant de donner au moins un point par composante semi-algébriquement connexe d'une hypersurface réelle. Cet algorithme n'opère qu'une seule déformation infinitésimale dans les cas où l'hypersurface contient une infinité de points singuliers.
Rappelons que :

- le calcul efficace des bases de Gröbner repose en partie sur un pré-calcul effectué sur une arithmétique modulaire (voir [36]);
- le calcul efficace de la Représentation Univariée Rationnelle repose en partie sur une prédiction modulaire de l'élément séparant.

Nous donnons ci-dessous des méthodes qui permettent de :
- calculer une base de Gröbner dans $K(\varepsilon)$;
- prédire un élément séparant d'un système d'équations polynomiales à coefficients dans $K(\varepsilon)$ en utilisant une arithmétique modulaire [81];

en n'effectuant les calculs que sur K.

1.2.1 Calculer une base de Gröbner dans $K(\varepsilon)[X_1,\ldots,X_n]$

Définition 1.2 *Une E-spécialisation Φ est un homomorphisme*

$$\Phi\ :\ R[E] \longrightarrow R,$$

défini par l'image e de E par Φ.

Considérons un ordre par blocs éliminant les variables $X_1,\ldots,X_n$ (i.e. E est plus petite que toutes les variables X_i), et la restriction de cet ordre à $X_1,\ldots,X_n$, on note $\mathrm{lm}_{X_1,\ldots,X_n}(P)$ le monôme dominant de P pour cet ordre et $\mathrm{lt}_{X_1,\ldots,X_n}(P)$ le terme dominant.

Définition 1.3 *Pour tout P dans $R[E,X_1,\ldots,X_n]$, on définit*

$$P = \mathrm{lc}(P)X^A + Q$$

avec $\mathrm{lc}(P) \in R[E]$ et $\mathrm{lm}_{X_1,\ldots,X_n}(Q) < \mathrm{lm}_{X_1,\ldots,X_n}(P) = X^A$. Alors, si $e \in R$ n'est pas une racine de $\mathrm{lc}(P)$, en notant $c = \mathrm{lc}(P(e))$,

$$\mathrm{lm}_{X_1,\ldots,X_n}(P_e) = X^A$$
$$\mathrm{lt}(P_e) = cX^A.$$

Etant donné un système d'équations polynomiales S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$, on note $\mathcal{L}$ les polynômes en la variable E qui sont les coefficients des monômes dominants de la base de Gröbner de $I_E = \langle S_E \rangle$ pour un ordre éliminant $X_1,\ldots,X_n$.

Lemme 1.7 *[42] Soit Φ la E-spécialisation qui associe e à E, et G une base de Gröbner (pour un ordre d'élimination de $X_1,\ldots,X_n$) de I_E. Si e n'est pas une racine d'un polynôme dans $\mathcal{L}$, alors $\Phi(G)$ est une base de Gröbner de $\Phi(I_E)$.*

Algorithme ε-Gröbner

- **Entrée:** un système d'équations polynomiales dans $K[\varepsilon][X_1,\ldots,X_n]$.
- **Sortie:** une base de Gröbner non réduite dans $K(\varepsilon)[X_1,\ldots,X_n]$ de l'idéal engendré par le système d'entrée.

1. Remplacer le système d'entrée par le système obtenu en substituant une nouvelle variable E à ε.
2. Calculer une base de Gröbner du système obtenu pour un ordre éliminant $X_1,\ldots,X_n$.
3. Spécialiser la variable E à ε dans la base de Gröbner obtenue.

On en déduit aisément une procédure εTest-Dim qui teste si un système d'équations polynomiales à coefficients dans $K[\varepsilon]$ est de dimension zéro.

1.2.2 Trouver un élément séparant et un bon centre

Dans cette section, nous allons montrer comment prédire un élément séparant pour un système d'équations polynomiales zéro-dimensionnel et radical à coefficients dans $K[\varepsilon]$ en utilisant une arithmétique modulaire. Ceci est un point crucial pour le calcul efficace d'une Représentation Univariée Rationnelle.

Supposons que le système d'équations polynomiales $S_\varepsilon \subset K[\varepsilon][X_1,\ldots,X_n]$ soit zéro-dimensionnel et $I_\varepsilon = \langle S_\varepsilon \rangle$. On note S_e $(e \in K)$ le système d'équations polynomiales obtenu en subsituant e à ε dans S_ε et notons $I_e = \langle S_e \rangle$.

En utilisant les notations du dernier paragraphe, un élément e de K qui n'est pas racine d'un polynôme dans $\mathcal{L}$ est tel que $\dim(K[X_1,\ldots,X_n]/I_e) = \dim(K\langle\varepsilon\rangle[X_1,\ldots,X_n]/I_\varepsilon)$. Notons $\mathcal{E}$ l'ensemble des éléments de C qui ne sont pas racines d'un polynôme dans $\mathcal{L}$. Il est alors évident que le complémentaire de $\mathcal{E}$ contient au plus un nombre fini d'éléments de K.

Lemme 1.8 *Les assertions suivantes sont équivalentes :*
- *a) I_ε est un idéal radical,*
- *b) Il existe $e_0 \in \mathcal{E} \cap K$ tel que I_{e_0} est radical,*
- *c) L'ensemble complémentaire de $\mathcal{E}' = \{e \in \mathcal{E} \mid I_e \text{ est radical}\}$ est fini.*

Preuve:
- b) implique c):
  Il est évident que $\mathcal{E}'$ est non vide puisque $e_0 \in \mathcal{E}'$. Soit e un élément de $\mathcal{E}'$. Puisque I_e est radical, toutes les solutions de I_e sont des solutions simples. De plus, dans un voisinage U de e, la dimension du quotient est la même pour tout $e' \in U$. Donc les solutions varient continuement et il existe un voisinage de e dans lequel toutes les solutions sont simples. On sait donc que l'ensemble $\mathcal{E}'$ est un ensemble non vide, ouvert et constructible pour la topologie de Zariski. On sait donc que le complémentaire de $\mathcal{E}'$ est un fermé constructible dans C, donc c'est un ensemble fini de points.
- c) implique a):
  Puisque le complémentaire de $\mathcal{E}'$ est fini, il existe un intervalle ouvert du type $(0,\alpha)$ (où $\alpha \in R$) tel que $\forall e \in (0,\alpha)$, e est un élément de $\mathcal{E}'$. Donc, si on note $\mathcal{E}'$ l'extension de $\mathcal{E}'$ dans $R\langle\varepsilon\rangle$, ε appartient à $\mathcal{E}'$.
- a) implique b):
  L'extension de l'ensemble ouvert constructible $\mathcal{E}' = \{e \in \mathcal{E} \mid I_e \text{ est radical}\}$ à $C\langle\varepsilon\rangle$ contient ε et est non vide. Ainsi le complémentaire de $\mathcal{E}'$ est fini et $\mathcal{E}' \cap K$ est non vide.

■

Soit $e_0 \in \mathcal{E} \cap K$, tel que S_{e_0} est zéro-dimensionnel, et tel que I_{e_0} est radical . Soit u un élément séparant pour S_{e_0}.

Lemme 1.9 *L'idéal I_ε est radical et u est un élément pour S_ε.*

Preuve: Puisque les zéros de I_{e_0} sont simples, et les zéros de I_e restent simples et varient continument pour $e \in \mathcal{E}$ dans un voisinage de e_0, si u est un élément séparant pour S_{e_0}, alors u reste un élément séparant pour S_e dans un voisinage de e.

Considérons

$$\mathcal{E}'' = \{e \in \mathcal{E}' \mid I_e \text{ est radical, } u \text{ est séparant pour } S_e\}.$$

L'ensemble $\mathcal{E}''$ est non vide et ouvert pour la topologie euclidienne. Il est constructible pour la topologie de Zariski car il peut être défini par une formule du premier ordre. Ainsi, le complémentaire de $\mathcal{E}''$ (qui est fermé et constructible) est un ensemble fini de points. Donc, $\varepsilon \in \mathcal{E}''$, et I_ε est un idéal radical et u est un élément séparant pour S_ε. ■

Finalement, nous obtenons l'algorithme ε-**CSE** (**C**heck **S**eparant **E**lement) qui trouve un élément séparant pour un système d'équations polynomiales à coefficients dans $K(\varepsilon)$ dans le cas zéro-dimensionnel et radical.

Algorithme ε-CSE

- **Entrée :** Un système zéro-dimensionel S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$, une base de Gröbner G_ε de $I_\varepsilon = \langle S_\varepsilon \rangle$, un élément $e \in K$ tel que I_e est radical et $\dim(K[X_1,\ldots,X_n]/I_e) = \dim(K\langle\varepsilon\rangle[X_1,\ldots,X_n]/I_\varepsilon)$, G_e est une base de Gröebner de I_e, et enfin un élément $u \in K[X_1,\ldots,X_n]$.
- **Sortie :** *true* si u est un élément séparant pour S et *false* si ce n'est pas le cas.
- renvoyer CSE(S_e).

Un *bon couple* $(A,e) \in R^n \times R$ pour P est tel que :

- le système $\mathcal{C}_\varepsilon(V(P-\varepsilon),A)$ défini par $P-\varepsilon$, $\overrightarrow{\text{grad}}_M\,(P) // \overrightarrow{AM}$ est un ensemble fini de points.
- le système d'équations polynomiales $\mathcal{C}_e(A)$ défini par $P-e$, $\overrightarrow{\text{grad}}_M\,(P) // \overrightarrow{AM}$ est un ensemble fini de points et l'idéal $\mathcal{I}_{A,e} = \langle [P-e,\Delta_A(P-e)] \rangle$ est radical,
- $\dim(K[X_1,\ldots,X_n]/\mathcal{I}_{A,e}) = \dim(K\langle\varepsilon\rangle[X_1,\ldots,X_n]/\mathcal{I}_{A,\varepsilon})$ où $\mathcal{I}_{A,\varepsilon} = \langle [P-\varepsilon,\Delta_A(P-\varepsilon)] \rangle$.

D'après les résultats précédents et ceux de la section 2, l'algorithme suivant trouve un bon couple pour l'hypersurface définie par $P-\varepsilon = 0$.

Algorithme ε-CC

- **Entrée:** $P \in K[X_1,\ldots,X_n]$ tel que $V(\mathcal{G})$ contient une infinité de points.
- **Sortie:** Un bon centre A pour $P-\varepsilon$, une base de Gröbner G_ε de $\mathcal{I}_{A,\varepsilon}$, un élément e tel que $\mathcal{I}_{A,e}$ est radical, $\dim(K[X_1,\ldots,X_n]/\mathcal{I}_{A,e}) = \dim(K\langle\varepsilon\rangle[X_1,\ldots,X_n]/\mathcal{I}_{A,\varepsilon})$ et G_e est une base de Gröbner de $\mathcal{I}_{A,e}$.

1. Faire $d := 1$.
2. Pour $A \in \{0,\ldots,d\}^n, e = d$ et $A \in \{0,\ldots,d\}^n \setminus \{0,\ldots,d-1\}^n, e \in \{1,\ldots,d-1\}$, Tester si $\mathcal{C}_e(A)$ est zéro-dimensionel et si $\mathcal{I}_{A,e}$ est radical.
3. Vérifier que $\mathcal{C}_\varepsilon(V(P-\varepsilon),A)$ est un ensemble fini de points.
4. Dès qu'un bon couple (A,e) est détecté, retourner ce couple et la base de Gröbner de $\mathcal{I}_{A,e}$, G_e.
5. Si aucun bon couple n'a été trouvé, incrémenter d de 1, et retourner au pas 2.
6. Retourner A, une base de Gröbner G_ε de $\mathcal{I}_{A,\varepsilon}$, et e.

1.3 Représentations Univariées Rationnelles à coefficients dans $K(\varepsilon)$

Le calcul de Représentations Univariées Rationnelles à coefficients infinitésimaux pose au moins deux problèmes :

- la prédiction de l'élément séparant : nous apportons une solution à ce problème dans le paragraphe précédent,
- et le calcul de la RUR elle-même : les coefficients sont des polynômes en ε. Ainsi, l'arithmétique utilisée est couteuse (comparativement à une arithmétique sur les entiers). Dans [84], nous proposons de faire ce calcul en «précision fixée» sur ε, c'est-à-dire de choisir un entier p et de borner les puissances de ε apparaissant en cours de calcul par p. Si cet entier n'est pas assez grand, une division par zéro apparait et on augmente p.

Cette stratégie s'est avérée difficile à mettre en œuvre.
Notons que :

- dans le cadre que nous nous sommes fixé, nous ne travaillons qu'avec des systèmes polynomiaux qui engendrent des idéaux radicaux : en effet, l'algorithme ε-**CC** renvoie un point A et un rationnel e tels que :
  - l'idéal engendré par les équations $P-\varepsilon=0$, $\overrightarrow{grad}_M(P) // \overrightarrow{AM}$ est radical et zéro-dimensionnel,
  - l'idéal engendré par les équations $P-e=0$, $\overrightarrow{grad}_M(P) // \overrightarrow{AM}$ est radical et zéro-dimensionnel,
- de manière à garder un bon contrôle des calculs intermédiaires, on peut espérer pouvoir utiliser des techniques de remontée «à la Hensel» puisqu'on n'a qu'une seule variable à remonter.

Dans [47], les auteurs s'inspirent des travaux exposés dans [43, 44, 45, 46] pour proposer un algorithme probabliliste calculant des Représentations Univariées Rationnelles (appelées *Résolution Géométriques* dans ces travaux) de systèmes zéro-dimensionnels dans le cas d'idéaux radicaux. Dans [88], E. Schost propose un algorithme probabiliste pour calculer des Représentations Univariées Rationnelles paramétrées. Ces algorithmes utilisent un opérateur dit «de Newton» (plus connu sous le nom de remontée de Hensel) dont on peut trouver une description dans [63] et qui est déjà utilisé dans [92, 102] pour calculer des bases de Gröbner lexicographiques. Comme nous désirons rester dans un cadre certifié, nous avons adapté, en collaboration avec E. Schost, ces techniques de calcul à notre problème pour calculer des Représentations Univariées Rationnelles à coefficients infinitésimaux tout en certifiant le résultat obtenu, dans le cas d'idéaux zéro-dimensionnels radicaux. Tout d'abord, rappelons le principe de base de l'opérateur de Newton utilisé. Soit $S_\varepsilon = (p_1,\ldots,p_n)$ un système d'équations polynomiales dans $K(\varepsilon)[X_1,\ldots,X_n]$, zéro-dimensionnel et tel que $\langle S_\varepsilon \rangle$ soit radical. Soit e_0 un rationnel tel que :

- le système S_{e_0} obtenu en spécialisant ε à e_0 dans S_ε définisse un ensemble fini de points,
- les idéaux $\langle S_{e_0} \rangle$ et $\langle S_\varepsilon \rangle$ ont même degré,
- l'idéal $\langle S_{e_0} \rangle$ est radical.

On se donne une Représentation Univariée Rationnelle de

$$S_{e_0} = (p_1(e_0,X_1,\ldots,X_n),\ldots,p_n(e_0,X_1,\ldots,X_n)) \ :$$

$$\begin{cases} f(T) = 0 \\ g_0(T)X_1 = g_1(T) \\ \vdots \\ g_0(T)X_n = g_n(T) \end{cases}$$

Une fois ce calcul effectué, l'algorithme exposé se déroule en trois étapes :

- on calcule un développement de Taylor de la RUR à coefficients infinitésimaux au voisinage de $\varepsilon - e_0$, jusqu'à un certain ordre,
- une fois ce développement calculé, on reconstruit les coefficients de la RUR à coefficients infinitésimaux à l'aide d'approximants de Padé,
- on vérifie le résultat retourné.

Nous exposons ci-dessous, l'opérateur de Newton (voir [47]) qui va permettre de calculer un développement de la RUR à coefficients infinitésimaux autour de $\varepsilon - e_0$. Pour simplifier la suite de l'exposé, on suppose que la variable X_1 est séparante. Puisque l'idéal $\langle S_{e_0} \rangle$ est radical, les solutions s'écrivent (en appliquant l'algorithme d'Eculide étendu à f et g_0) :

$$\begin{cases} f(T) = 0 \\ X_1 = T \\ X_2 = q_2(T) \\ \vdots \\ X_n = q_n(T) \end{cases}$$

Cette Représentation Univariée Rationnelle peut être vue comme une approximation des solutions en $O(\varepsilon - e_0)$. Pour obtenir une approximation en $O((\varepsilon - e_0)^2)$, on applique l'opérateur de Newton

$$\begin{pmatrix} Q_1(\varepsilon,T) \\ \vdots \\ Q_n(\varepsilon,T) \end{pmatrix} = \overrightarrow{v}^t - J^{-1}_{(\varepsilon-e_0,\,\overrightarrow{v})} \begin{pmatrix} p_1(\varepsilon,\,\overrightarrow{v}) \\ \vdots \\ p_n(\varepsilon,\,\overrightarrow{v}) \end{pmatrix}$$

- où J est la matrice jacobienne associée à la famille de polynômes $(p_1,\ldots,p_n)$,
- où $\overrightarrow{v}$ est le vecteur $(T,q_1(T),\ldots,q_n(T))$.

La paramétrisation

$$f(T)=0, \quad \begin{cases} X_1 = Q_1(\varepsilon,T) \\ \vdots \\ X_n = Q_n(\varepsilon,T) \end{cases}$$

est une solution en $O((\varepsilon-e_0)^2)$. L'expression $Q_{1(\varepsilon,T)}$ peut aussi être réécrite sous la forme

$$Q_1(\varepsilon,T) = T + (\varepsilon-e_0)\Delta(T) + O((\varepsilon-e_0)^2)$$

et ainsi

$$T = X_1 - (\varepsilon-e_0)\Delta(X_1) + O((\varepsilon-e_0)^2)$$

En remplaçant T par le membre de droite dans l'égalité ci-dessus dans $f(T)$ et les Q_j, on obtient

$$f(X_1) - (\varepsilon-e_0)\left(f'(X_1)\Delta(X_1)\mathrm{mod} f(X_1)\right) + O((\varepsilon-e_0)^2) = 0$$

et

$$X_j = Q_j(\varepsilon,X_1) - (\varepsilon-e_0)\left(\frac{\partial Q_j}{\partial T}\Delta(X_1)\mathrm{mod} f(X_1)\right) + O((\varepsilon-e_0)^2) = 0$$

pour $j\in\{1,\ldots,n\}$ qui est une solution approchée de S_{e_0} en précision $O((\varepsilon-e_0)^2)$. On itère ce processus jusqu'à une certaine précision «suffisante». Pour plus de détail sur cette méthode, nous invitons le lecteur à consulter [47]. Remarquons que cet opérateur de Newton permet de calculer une Représentation Univariée Rationnelle modulo des puissances de l'idéal $\langle\varepsilon-e_0\rangle$, c'est-à-dire à $(\varepsilon-e_0)^p$ près dans $K(\varepsilon-e_0)$ (où p est la précision jusque laquelle on est allé).

Considérons maintenant q un élément de $K(\varepsilon)$ dont le numérateur et le dénominateur sont de degrés bornés par η. On doit maintenant reconstruire q à partir de son développement en série en e_0. Ceci peut se faire en utilisant les approximants de Padé (algorithme d'Euclide étendu) si le développement va jusqu'à la précision 2η (voir [88, 19, 41]).

Le caractère probabiliste intervient à trois niveaux :

- lorsque les variables sont spécisalisées en de «mauvaises valeurs», il arrive que l'idéal spécialisé ne soit pas radical, et/ou que des solutions soient oubliées (le degré de l'idéal spécialisé est inférieur à celui de l'idéal de départ);
- dans [88], l'auteur donne une précision suffisante p pour que le développement calculé par l'opérateur de Newton modulo $\langle\varepsilon-e_0\rangle^p$ définisse de manière unique une Représentation Univariée Rationnelle du système de départ à coefficients dans $K(\varepsilon)$, cette borne est bonne dans l'absolu mais pas assez précise en pratique, un test d'arrêt probabiliste est donc nécesssaire;

– les calculs de remontée sont faits modulo des puissances d'un entier p le test d'arrêt est probabiliste. Il nous faut donc vérifier que la Représentation Univariée Rationnelle ainsi calculée définisse bien les points qui sont solutions du système de départ.

Nous apportons des solutions à ces deux problèmes :

– Par un pré-calcul qui consiste à calculer une base de Gröbner du système de départ pour un ordre par bloc où $\varepsilon < X_1, \ldots, X_n$, on est en mesure de déterminer si une spécialisation de ε en un rationnel e_0 est bonne. Pour cela,

  – on vérifie que e_0 n'annule aucun coefficient de tête de la base de Gröbner calculée, on est ainsi sûr de n'oublier aucune solution,

  – on utilise la prédicition d'un élément séparant et on vérifie que l'idéal spécialisé est radical.

  On peut alors calculer une Représentation Univariée Rationnelle avec l'algorithme décrit dans [81] qui sera le point de départ de l'étape de remontée.

– Une fois l'étape de remontée effectuée, pour vérifier que la RUR calculée

$$\begin{cases} f(\varepsilon,T) = 0 \\ g_0(\varepsilon,T)X_1 = g_1(\varepsilon,T) \\ \vdots \\ g_0(\varepsilon,T)X_n = g_n(\varepsilon,T) \end{cases}$$

  définit bien les solutions du système de départ, on homogénéise les polynômes de départ, on remplace la variable d'homogénéisation par le polynôme $g_0(\varepsilon,T)$, puis les variables X_i par les polynômes $g_i(\varepsilon,T)$, le tout modulo $f(\varepsilon,T)$. Si on obtient 0, le résultat retourné est exact (puisqu'on a le bon degré en T), si ce n'est pas le cas, on reprend l'étape de remontée à l'étape où nous l'avions laissée.

Considérons les routines suivantes :

– **NewtonLifting :** qui calcule un développement de la Représentation Univariée Rationnelle à coefficients infinitésimaux modulo $\langle \varepsilon - e_0 \rangle^p$, en prenant en entrée un système d'équations spécialisé en un rationnel e_0 (tel que le système engendre un idéal zéro-dimensionnel et radical), un élément séparant, une RUR de ce système associé à cet élément séparant et un entier d.

– **Check :** qui prend en entrée une Représentation Univariée Rationnelle et un système d'équations et retourne *false* si les points définis par la RUR ne sont pas solutions du système sinon elle retourne *true*.

Nous obtenons alors l'algorithme décrit ci-dessous :

Algorithme ε-RUR

- **Entrée :** Un système S d'équations polynomiales dans $K[\varepsilon][X_1,\ldots,X_n]$ ayant un nombre fini de solutions et tel que $\langle S\rangle$ soit radical.
- **Sortie :** Une Représentation Univariée Rationnelle à coefficients dans $K(\varepsilon)$ décrivant les solutions de S.

1. Calculer une base de Gröbner G_ε DRLDRL pour l'ordre $\varepsilon < X_1,\ldots,X_n$.
2. Choisir e_0 un rationnel n'annulant aucun des termes de tête de G_ε (dont les polynômes sont vus à coefficients dans $K[\varepsilon]$).
3. Vérifier que $\langle S_{e_0}\rangle$ est zéro-dimensionnel et radical.
4. si ce n'est pas le cas, faire $e_0 := e_0 + 1$ et revenir au pas précédent.
5. trouver un élément séparant u pour S_{e_0}.
6. Choisir e_1 un rationnel différent de e_0.
7. Vérifier que $\langle S_{e_1}\rangle$ est zéro-dimensionnel et radical et que u est un élément séparant pour $\langle S_{e_1}\rangle$.
8. Si ce n'est pas le cas, faire $e_1 := e_1 + 1$ et revenir au pas précédent.
9. Calculer une RUR de $\langle S_{e_0}\rangle$ et de $\langle S_{e_1}\rangle$ pour l'élément séparant u. Soit $rur0$ et $rur1$ les résultats.
10. Faire
    - a) Utiliser NewtonLifting prenant en entrée S_{e_0}, $rur0$, et u et $d = 1$,
    - b) Reconstruire le résultat dans $K(\varepsilon)$.
    - c) Si la reconstruction a échoué ou si le résultat spécialisé en e_1 est différent de $rur1$, retourner au pas a) avec $d := d + 1$.
    - d) Poser rur le résultat obtenu. Si Check(rur, S_ε)=false, alors retourner au pas a).
11. retourner rur.

Remarque 1.2

- *Le test d'arrêt de la phase de remontée est particulièrement important pour l'efficacité des calculs. Notons que l'inversion de matrices est ce qui coûte le plus cher en terme de temps de calcul dans la phase de remontée. Nous avons retenu deux tests d'arrêt potentiellement efficaces :*
  - *l'opérateur de Newton est exécuté jusqu'à ce que pour deux puissances successives de p le résultat retourné est identique,*
  - *on calcule deux Représentations Univariées Rationnelles associées au même élément séparant de l'idéal spécialisé en deux bonnes valeurs e_0 et e_1 différentes et à chaque exécution de l'opérateur de Newton, on vérifie que le résultat retourné et spécialisé en e_0 et e_1 est égal aux Représentations Univariées Rationnelles précédemment calculées.*

  Nous avons constaté qu'en pratique, le second test d'arrêt est meilleur en terme de temps de calcul.
- *Dans [84], nous proposons de faire le calcul de Représentation Univariée Rationnelle en précision tronquée. Ce type de stratégie revient à calculer un développement de la RUR à coefficients infinitésimaux autour de ε. Ceci n'est malheureusement pas conciliable de manière simple avec l'algorithme que nous proposons. En effet, nous sommes contraints de calculer un développement de cette RUR en un rationnel tel que lorsqu'on spécialise ε en ce rationnel, l'idéal spécialisé reste zéro-dimensionnel, ce qui n'est pas le cas lorsqu'on spécialise en 0.*

Dans la section 1.6 de ce chapitre, nous procédons à l'analyse expérimentale détaillée de cet algorithme.

1.4 Limites des racines bornées des systèmes zéro-dimensionnels à coefficients dans $K(\varepsilon)$

Dans la section 6.2 du chapitre 6 de la partie I, un algorithme prenant en entrée un système d'équations polynomiales à coefficients infinitésimaux et retournant une liste de Représentations Univariées Rationnelles décrivant les limites des racines bornées du système d'entrée est décrit (voir aussi [84]). Quelque soient les choix de l'élément séparant qui peuvent être effectués, cet algorithme calcule les polynômes caractéristiques des multiplications par chacune des variables dans l'anneau quotient. L'objectif de ces calculs est de détecter si l'élément séparant choisi est un *élément bien séparant* (voir chapitre précédent).

Nous avions implanté cet algorithme en Maple et constaté que ces calculs de polynômes caractéristiques étaient particulièrement coûteux en pratique. Par ailleurs, le cout de ces calculs est difficilement justifiable puisqu'un élément séparant «choisi au hasard» est un élément bien séparant. Nous avons donc cherché un autre algorithme plus efficace en pratique. Nous exposons cet algorithme ci-dessous.

Soit une Représentation Univariée Rationnelle à coefficients infinitésimaux associée à un élément séparant u d'un système zéro-dimensionnel à coefficients dans $K[\varepsilon]$ et engendrant un idéal radical :

$$\left\{\begin{array}{l} f(\varepsilon,T)=0 \\ g_0(\varepsilon,T)X_1-g_1(\varepsilon,T)=0 \\ \vdots \\ g_0(\varepsilon,T)X_n-g_n(\varepsilon,T)=0 \end{array}\right.$$

Puisque l'idéal engendré par le système d'entrée est radical, les notions de ARUR et de RUR coïncident (il n'y a pas de racines multiples). Soit ν le plus petit entier tel que :

- il existe un polynôme $c(\varepsilon)\in K[\varepsilon]$ vérifiant $c(0)\neq 0$, et
- le polynôme $F(\varepsilon,T)=\varepsilon^\nu c(\varepsilon)f(\varepsilon)$ appartient à $K[\varepsilon,T]$.

Pour $i\in\{0,\ldots,n\}$, on note $G_i(\varepsilon,T)=\varepsilon^\nu g_i(\varepsilon,T)$. Rappelons le lemme 6.2 (chapitre 6 de la partie I).

Lemme 1.10 *Soit $u=\sum_{i=1}^n u_iX_i$, (avec $u_i\in K$) est un élément séparant pour S_ε. Alors u est un* élément bien séparant *si et seulement si :*

- *Il existe des polynômes $c_1(\varepsilon),\ldots,c_n(\varepsilon)$ dans $K[\varepsilon]$ tels que pour tout $i\in\{1,\ldots,n\}$, $c_i(0)\neq 0$ et les polynômes*

$$c_1(\varepsilon)G_1(\varepsilon,T),\ldots,c_n(\varepsilon)G_n(\varepsilon,T)$$

  appartiennent à $K[\varepsilon,T]$,
- *les images par u de deux éléments bornés x et y de $V(S_\varepsilon)$ sont infiniment proches, si et seulement si x et y sont infiniment proches.*

La preuve du lemme 6.2 (chapitre 6 de la partie I) montre que s'il existe des polynômes $c_1(\varepsilon),\ldots,c_n(\varepsilon)$ dans $K[\varepsilon]$ tels que pour tout $i \in \{1,\ldots,n\}$, $c_i(0) \neq 0$ et les polynômes

$$c_1(\varepsilon)G_1(\varepsilon,T),\ldots,c_n(\varepsilon)G_n(\varepsilon,T)$$

appartiennent à $K[\varepsilon,T]$, alors les racines du système S_ε ont des images par u bornées si et seulement si ces racines sont bornées.

Une fois la Représentation Univariée Rationnelle à coefficients infinitésimaux calculée, on peut dans un premier temps tester si le premier point du lemme ci-dessus est vérifié. Si ce n'est pas le cas, on doit recalculer la RUR avec un autre élément séparant. Si c'est le cas, on doit vérifier le second point du lemme ci-dessus.

Dans la suite, on suppose que $(f(\varepsilon,T),g_0(\varepsilon,T),g_1(\varepsilon,T),\ldots,g_n(\varepsilon,T))$ est une Représentation Univariée Rationnelle à coefficients infinitésimaux telle qu'il existe des polynômes $c_1(\varepsilon),\ldots,c_n(\varepsilon)$ dans $K[\varepsilon]$ vérifiant $c_i(0) \neq 0$ (pour $i \in \{1,\ldots,n\}$) tels que les polynômes

$$c_1(\varepsilon)G_1(\varepsilon,T),\ldots,c_n(\varepsilon)G_n(\varepsilon,T)$$

appartiennent à $K[\varepsilon,T]$. D'après le lemme précédent, il suffit de vérifier que pour tout couple de racines bornées (ζ_1,ζ_2), $\lim_0(u(\zeta_1)-u(\zeta_2))=0$ si et seulement si

$$\forall i \in \{1,\ldots,n\},\ \lim_0(X_i(\zeta_1)-X_i(\zeta_2))=0.$$

Notons que d'après les hypothèses ci-dessus, pour toute racine bornée α de $f(\varepsilon,T)$, et pour tout $i \in \{1,\ldots,n\}$, $\lim_0\left(\frac{g_i(\varepsilon,\alpha)}{g_0(\varepsilon,\alpha)}\right)$ existe.
Soit α_1 et α_2 deux racines **bornées** de $f(\varepsilon,T)$. On note $\tilde{\alpha}_{1,\delta}$ et $\tilde{\alpha}_{2,\delta}$ les développements en séries de Puiseux de α_1 et α_2 tronqués à l'ordre δ (c'est-à-dire tel que toute puissance de ε apparaissant dans le développement est inférieur à δ). Le lemme suivant est évident :

Lemme 1.11

$$\forall \delta \in \mathbb{N} \quad \lim_0(\alpha_1-\alpha_2)=0 \Longleftrightarrow \lim_0(\tilde{\alpha}_{1,\delta}-\tilde{\alpha}_{2,\delta})=0.$$

Considérons maintenant un polynôme $G(\varepsilon,T) \in K[\varepsilon,T]$ de degré d en ε tel qu'aucune puissance de ε ne divise G, $\alpha \in C$ tel que $G(\varepsilon,\alpha) \neq 0$ et un développement en séries de Puiseux de α tronqué à un ordre δ. Comme $G(\varepsilon,T)$ s'écrit :

$$G(\varepsilon,T)=\gamma_0(T)+\gamma_1(T)\varepsilon+\ldots+\gamma_d(T)\varepsilon^d$$

avec $\gamma_i(T) \in K[T]$, il est clair que si $G(\varepsilon,\tilde{\alpha}_\delta) \neq 0$, alors $o(G(\varepsilon,\tilde{\alpha}_\delta))=o(G(\varepsilon,\alpha))$. De plus, on a alors :

$$\lim_0(G(\varepsilon,\tilde{\alpha}_\delta))=\lim_0(G(\varepsilon,\alpha)).$$

Comme la Représentation Univariée Rationnelle calculée décrit les solutions d'un système engendrant un idéal radical, pour toute racine α de $f(\varepsilon,T)$, on a : $c(\varepsilon)G_0(\varepsilon,\alpha) \neq 0$. Donc, si $\forall i \in \{1,\ldots,n\}$ $c_i(\varepsilon)G_i(\varepsilon,\alpha) \neq 0$, et si δ est un entier tel que $G_i(\varepsilon,\tilde{\alpha}_\delta) \neq 0$ alors :

$$\lim_0\left(\frac{g_i(\varepsilon,\alpha)}{g_0(\varepsilon,\alpha)}\right)=\lim_0\left(\frac{c(\varepsilon)c_i(\varepsilon)G_i(\varepsilon,\tilde{\alpha}_\delta)}{c(\varepsilon)c_i(\varepsilon)G_0(\varepsilon,\tilde{\alpha}_\delta)}\right).$$

On en déduit le lemme suivant :

Lemme 1.12 *Soit* $(F(\varepsilon,T),G_0(\varepsilon,T),G_1(\varepsilon,T),\ldots,G_n(\varepsilon,T))$ *une NRUR telle que* F *est square-free, et* $\alpha \in C$ *une racine de* $F(\varepsilon,T)$ *telle que* $\exists i_0 \in \{1,\ldots,n\}$, $G_{i_0}(\varepsilon,\alpha) \neq 0$. *Soit* $\delta \in \mathbb{N}$ *tel que* $G_{i_0}(\varepsilon,\tilde{\alpha}_\delta) \neq 0$. *Alors, on a :*

$$\lim_0 \left(\frac{c(\varepsilon)c_{i_0}(\varepsilon)G_{i_0}(\varepsilon,\alpha)}{c(\varepsilon)c_{i_0}(\varepsilon)G_0(\varepsilon,\alpha)} \right) = \lim_0 \left(\frac{c(\varepsilon)c_{i_0}(\varepsilon)G_{i_0}(\varepsilon,\tilde{\alpha}_\delta)}{c(\varepsilon)c_{i_0}(\varepsilon)G_0(\varepsilon,\tilde{\alpha}_\delta)} \right).$$

Dans les cas où il existe une racine α de $f(\varepsilon,T)$ telle que il existe $i \in \{1,\ldots,n\}$ vérifiant $G_i(\varepsilon,\alpha) = 0$, on a bien évidemment $\lim_0 \left(\frac{c(\varepsilon)c_i(\varepsilon)G_i(\varepsilon,\alpha)}{c(\varepsilon)c_i(\varepsilon)G_0(\varepsilon,\alpha)} \right) = 0$. Ces cas sont détectables par le fait que α est une racine de $\gcd(f(\varepsilon,T),g_i(\varepsilon,T))$.

Ainsi, pour toute racine bornée de $f(\varepsilon,T)$, nous pouvons calculer $\forall i \in \{1,\ldots,n\}$ les limites $\lim_0 \left(\frac{g_i(\varepsilon,\alpha)}{g_0(\varepsilon,\alpha)} \right)$, soit (dans les cas où $g_i(\varepsilon,\alpha) \neq 0$) en calculant un développement en séries de Puiseux tronqué à un certain ordre suffisamment grand, soit en garantissant que cette limite est nulle car $g_i(\varepsilon,\alpha) = 0$.

On déduit des deux lemmes précédents et de la remarque ci-dessus un algorithme vérifiant que l'élément séparant choisi pour calculer la Représentation Univariée Rationnelle à coefficients infinitésimaux est un *élément bien séparant* et qui calcule les limites des solutions d'un système zéro-dimensionnel et engendrant un idéal radical.

On note **Puiseux** (voir [35, 96]) la routine qui prend en entrée un polynôme univarié à coefficients dans $K[\varepsilon]$, sans facteurs carrés et un entier positif δ et qui renvoie les développements en séries de Puiseux des racines du polynôme d'entrée tronqués à l'ordre δ. On note ε**-RUR** la routine prenant en entrée un système d'équations polynomiales engendrant un idéal radical et une forme linéaire séparante et qui calcule une Représentation Univariée Rationnelle à coefficients infinitésimaux associée à cette forme séparante décrivant les solutions du système.

Dans l'algorithme **LRBP** (**L**imites des **R**acines **B**ornées via **P**uiseux), on suppose qu'aucune des racines du système d'entrée n'a des coordonnées nulles de manière à pouvoir appliquer le lemme 1.12. Dans les cas où de telles racines existent, l'algorithme est facilement adaptable en incluant des calculs de pgcd entre le premier polynôme de la RUR calculée et les paramétrisations de manière à détecter les racines qui ont une coordonnée nulle.

Algorithme LRBP

- **Entrée :** Un système d'équations polynomiales $S \subset K(\varepsilon)[X_1,\ldots,X_n]$ ayant un nombre fini de solutions.
- **Sortie :** Une liste de Représentations Univariées Rationnelles représentant les limites des solutions bornées de S.

1. Choisir $u \in \mathcal{U}$ et enlever u de $\mathcal{U}$,
2. Vérifier que u est séparant,
3. Calculer une NRUR normalisée de S associée à u,

$$F_u(\varepsilon,T),G_0(\varepsilon,T),\ldots,G_n(\varepsilon,T).$$

 Si cette NRUR n'est pas bien normailsée, revenir au pas 1,
4. Calculer les développements en séries de Puiseux de toutes les racines de F_u à un ordre δ suffisamment grand pour que :

$$\forall \alpha \neq 0 \mid F_u(\varepsilon,\alpha) = 0, \quad \widetilde{\alpha}_\delta \neq 0,$$

$$\forall i \in \{1,\ldots,n\} \quad G_{i_0}(\varepsilon,\widetilde{\alpha}_\delta) \neq 0.$$

5. Pour tout couple de développements $(\alpha_{1,\delta},\alpha_{2,\delta})$ tel que $\lim_0(\alpha_{1,\delta} - \alpha_{2,\delta}) = 0$ vérifier que :

$$\forall i \in \{1,\ldots,n\} \quad \lim_0\left(\frac{c(\varepsilon)c_{i_0}(\varepsilon)G_{i_0}(\varepsilon,\widetilde{\alpha}_{1,\delta})}{c(\varepsilon)c_{i_0}(\varepsilon)G_0(\varepsilon,\widetilde{\alpha}_{1,\delta})}\right) - \lim_0\left(\frac{c(\varepsilon)c_{i_0}(\varepsilon)G_{i_0}(\varepsilon,\widetilde{\alpha}_{2,\delta})}{c(\varepsilon)c_{i_0}(\varepsilon)G_0(\varepsilon,\widetilde{\alpha}_{2,\delta})}\right).$$

6. Si c'est le cas, renvoyer ..
7. sinon retourner au pas 1.

Remarque 1.3 *L'utilisation des développements en séries de Puiseux implique que les calculs soient faits avec des nombres algébriques : ainsi, on ne travaille plus uniquement dans K.*

Comme nous le montrerons dans la section 1.6 de ce chapitre, cet algorithme s'est avéré plus efficace que celui proposé dans [84] sur les exemples considérés.

1.5 Discussion sur la complexité de HA3

Le nombre de composantes semi-algébriquement connexes d'une variété algébrique réelle définie par des polynômes de degré d en n variables est en $\mathcal{O}(d)^n$, et on sait que cette borne est atteinte pour certains exemples. D'après le théorème de Bezout, il est clair que le degré des polynômes de la Représentation Univariée Rationnelle que nous calculons est en $\mathcal{O}(d)^n$. Donc, la sortie de notre algorithme est, en un sens, satisfaisante.

En terme de nombre d'opérations effectuées, la complexité théorique de notre algorithme est pire que $d^{\mathcal{O}(n)}$ qui est atteinte dans les algorithmes proposés dans [25, 74, 12]. En effet, on ne peut donner qu'une borne doublement exponentielle à l'algorithme **HA3** car des calculs de bases de Gröbner sont effectués. Par ailleurs, le choix du point A nécessite en théorie jusqu'à $\mathcal{O}(d^{n^2})$ essais. Néanmoins, notons que les bases de Gröbner sont doublement exponentielles sur des exemples pathologiques, et que génériquement le premier choix du point A est le bon.

Dans [85], les auteurs s'autorisent un calcul probabilste de Représentation Univariée Rationnelle pour améliorer la complexité théorique de l'algorithme **HA3**. Ils utilisent un cadre d'évaluation où les polynômes sont représentés sous la forme de programmes d'évaluation et les théorèmes de complexité de [43, 44, 45, 46, 47]. Si d est le degré du polynôme P et que ce polynôme est représenté par un programme d'évaluation de longueur L, on obtient le théorème ci-dessous en notant $\mathcal{M}(d)$ le cout de la multiplication de deux polynômes univariés de degré borné par d.

Proposition 1.1 *Une Représentation Univariée Rationnelle du système de dimension* 0

$$P(M) = \varepsilon, \quad \overrightarrow{grad}_M\,(P) // \overrightarrow{AM}$$

peut être calculée par un algorithme probabiliste en

$$\mathcal{O}(n^2(L + n^3)\mathcal{M}(d^n)^2)$$

opérations dans K.

Comme un bon point A est trouvé dès le premier choix, on obtient bien une complexité simplement exponentielle pour l'algorithme que nous proposons.

1.6 Validation expérimentale

1.6.1 Méthodologie

Dans la suite de cette section, nous comparerons :

– la taille de la sortie de l'algorithme **HA3** avec la taille de la sortie de l'algorithme de Décomposition Cylindrique Algébrique de Collins. En plus de cette simple comparaison, nous insisterons sur la taille des données intermédiaires qui **caractérisent** le comportement des algorithmes. Dans le cas des méthodes de points critiques il s'agit du degré du système zéro-dimensionnel étudié. Dans le cas de la Décomposition Cylindrique Algébrique, il s'agit de la somme des degrés des systèmes zéro-dimensionnels étudiés pendant la phase remontée. Nous n'avons pas pu obtenir cette somme, c'est pourquoi nous donnons le nombre de cellules retournées par l'algorithme de Décomposition Cylindrique Algébrique.
– les temps de calcul de l'algorithme **HA3** avec ceux de l'algorithme de Décomposition Cylindrique Algébrique de Collins sur un large panoplie d'exemples. En particulier, nous avons vu que les cas des hypersurfaces contenant une infinité de points singuliers posent problèmes. Nous ferons une étude précise du comportement de l'algorithme **HA3**.

Nous comparons l'implantation de notre algorithme avec le logiciel **QEPCAD** implanté par H. Hong et al. (voir [29]) et calculant une décomposition cylindrique algébrique avec notre algorithme, du point de vue de la taille de la sortie, mais aussi du point de vue des temps de calcul. Pour l'implantation de notre algorithme, nous utilisons :

– Gb, implanté en C++ par J.-C. Faugère, dédié au calcul de base de Gröbner,

– AGb, implanté en C++ par J.-C. Faugère, qui est une version «accélérée» de Gb pour le calcul de bases de Gröbner DRL,
– RS, implanté en C par F. Rouillier, dédié au calcul de Représentation Univariée Rationnelle à partir d'une base de Gröbner et au comptage et à l'isolation des racines réelles d'un polynôme univarié.
– Kronecker, implanté dans le système de Calcul Formel Magma [24] par G. Lecerf, dédié au calcul de Représentations Univariées Rationnelles, nous en utilisons l'opérateur de Newton (voir section 1.3).

Nous distinguons deux cas :
– le cas des hypersurfaces ne contenant au plus qu'un nombre fini de singularités,
– le cas des hypersurfaces contenant une infinité de singularités : ce cas nous est beaucoup moins favorable, puisqu'il nécessite l'introduction d'un infinitésimal.

Sauf mention contraire, tous les calculs décrits ci-dessous ont été effectués sur les machines PC 400 MHz avec 512 Mo de RAM de l'UMS MEDICIS [3]. Les temps de calcul sont donnés en secondes.

1.6.2 Cas sans singularités

Les 5 hypersurfaces ci-dessous contiennent au plus un nombre fini de points singuliers :

– Hyp1 :
```
2*u6^2*u5*u4*u3*u2+4*u6^2*u5*u4*u3+4*u6^2*u5*u4*u2+4*u6^2*u5*u3*u2-1
```
– Hyp2 :
```
36*u5^2*u4^2*u3^2*u2^2+88*u5^2*u4^2*u3^2*u2+32*u5^2*u4^2*u3^2+_
32*u5^2*u4^2*u3*u2^3+152*u5^2*u4^2*u3*u2^2-1
```
– Hyp3 :
```
36*u5^2*u4^2*u3^2*u2^2+88*u5^2*u4^2*u3^2*u2+32*u5^2*u4^2*u3^2+_
32*u5^2*u4^2*u3*u2^3+152*u5^2*u4^2*u3*u2^2+64*u5^2*u4^2*u3*u2_
+64*u5^2*u4^2*u2^3+32*u5^2*u4^2*u2^2+32*u5^2*u4*u3^3*u2^2-1
```
– Hyp4 :
```
552*u2*u3^2*u4+62208*u2+1492992*u3+2799360*u4-3*u2^2*u4^2_
-7842*u2*u3*u4+420*u2*u3*u4^2-314*u2^2*u3*u4+3*u2^2*u3^2*u4_
-62208*u2^2+429*u4^3+20736*u3^2-4*u2^3*u3^2-1157*u2^2*u3^2_
-18801*u2^2*u3-83520*u2*u3^2+39744*u2*u3+3*u2*u4^2+864*u2*u4_
+17280*u3^2*u4+60912*u4^2-864*u2^2*u4-207*u2^3*u3_
+1152*u3^2*u4^2+156*u4^3*u3+18540*u3*u4^2-554688*u3*u4_
+8*u2*u3^2*u4^2+2*u2^3*u3*u4-2*u2*u3*u4^3+u4^4-8957952
```
– Hyp5 :
```
110*u5^2*u4*u3+190*u5*u4^2*u3+80*u4^3*u3+80*u5^2*u3^2+_
270*u5*u4*u3^2+160*u4^2*u3^2+80*u5*u3^3+80*u4*u3^3-_
32*u4*u3^2*u2-32*u3^3*u2-80*u5^2*u2^2-128*u5*u4*u2^2-_
160*u5*u3*u2^2-112*u4*u3*u2^2-64*u3^2*u2^2-80*u5*u2^3-_
```

Hypersurface	HA3		CAD	
Hyp1	150	20	277	74
Hyp2	144	20	203	60
Hyp3	236	24	1399	394
Hyp4	84	10	*RE*	*RE*
Hyp5	151	26	*RE*	*RE*

TAB. 1.1 – *Algorithmes HA3 et CAD : comparaison de la taille de la sortie*

```
32*u3*u2^3+60*u5^2*u4+220*u5*u4^2+160*u4^3+67*u5*u4*u3+_
136*u4^2*u3-24*u5*u3^2-88*u4*u3^2-64*u3^3-100*u5^2*u2+_
32*u5*u4*u2+96*u4^2*u2-228*u5*u3*u2-108*u4*u3*u2-_
120*u3^2*u2+20*u5*u2^2+96*u4*u2^2-56*u3*u2^2+110*u5*u4+_
80*u4^2+48*u4*u3-32*u3^2+30*u5*u2+48*u4*u2-20*u3*u2
```

Chacun des polynômes définissant ces hypersurfaces est de *degré impair en au moins une variable* : l'usage de la fonction distance est donc pleinement justifié, car le lieu réel (qui est non vide) de chacune de ces hypersurfaces n'est pas borné.

Nous avons testé l'algorithme **HA3** en utilisant AGb pour le calculs de bases de Gröbner DRL et RS pour le calcul de Représentations Univariées Rationnelles et le comptage de racines réelles de polynômes univariés.

Dans le tableau 1.1, nous donnons le degré des systèmes zéro-dimensionnels produits par notre algorithme ainsi que le nombre de racines réelles trouvées (colonne **HA3**) avec le nombre de cellules produites et le nombre de points trouvés sur l'hypersurface par l'algorithme de Décomposition Cylindrique Algébrique (colonne **CAD**). Lorsque nous mettons le signe *RE*, ceci signifie qu'aucun résultat n'a été retourné au bout de 12 heures de calculs.

Le degré des systèmes zéro-dimensionnels engendrés par l'algorithme **HA3** est bien inférieur au nombre de cellules produites par l'algorithme de Décomposition Cylindrique Algébrique de Collins. Cette différence est moins sensible sur le nombre de points trouvés sur l'hypersurface.

Dans le tableau 1.2, nous comparons les temps de calcul de l'algorithme **HA3** avec ceux de l'algorithme de Décomposition Cylindrique Algébrique. Dans la première colonne **HA3** nous donnons le temps de calcul d'une base de Gröbner DRL avec AGb puis le temps de calcul d'une Représentation Univariée Rationnelle et du comptage du nombre de racines réelles du premier polynôme de la RUR calculée. Dans la colonne **CAD**, nous donnons le temps de calcul de la Décomposition Cylindrique Algébrique avec le logiciel **qepcad**. Le symbole ∞ signifie qu'aucun résultat n'a été retourné après plus de 12 heures de calcul.

Le gain obtenu en terme de taille de la sortie a des répercussions sensibles sur les temps de calcul : des exemples inaccessibles à l'algorithme de Décomposition Cylindrique Algébrique peuvent être traités.

Hypersurface	HA3		CAD
Hyp1	0,05	4,0	0,05
Hyp2	0,05	8,2	30
Hyp3	5,96	101,36	∞
Hyp4	0,66	23,76	∞
Hyp5	88,55	86	∞

TAB. 1.2 – *Algorithmes HA3 et CAD: comparaison des temps de calcul*

Signalons par ailleurs, que cet algorithme a été testé avec succès sur à peu près un millier d'hypersurfaces issues du problème d'Interpolation de Birkhoff (voir chapitre 4 de la partie II). Lors de ces tests, il n'a jamais été nécessaire de changer de point A.

1.6.3 Cas avec singularités

Le cas des hypersurfaces contenant une infinité de points singuliers complexes est un cas défavorable pour nos algorithmes. En effet, une déformation infinitésimale est nécessaire à la résolution de ces problèmes. En collaboration avec E. Schost, nous avons implanté dans le système de Calcul Formel Magma l'algorithme décrit dans la section 1.3.

Dans ce paragraphe, nous allons analyser les différentes étapes de cet algorithme. Puis, afin de valider les choix et algorithmes proposés dans la section 1.4, nous comparerons l'algorithme proposé dans cette section avec celui proposé dans le chapitre 6 de la partie I (voir aussi [84]). Pour des raisons techniques, tous ces calculs ont été effectués sur les machines ALPHA EV6 500MHz avec 640 Mo de RAM de l'UMS Medicis.
Pour faire ces tests, nous avons utilisé cette implantation de notre algorithme sur 15 hypersurfaces obtenues à partir du problème d'interpolation de Birkhoff en 3 variables (voir chapitre 4 de la partie II). Ces 15 exemples sont donnés en Annexe B.

Dans le tableau 1.3, nous évaluons le coût de l'introduction d'un infinitésimal dans le calcul d'une base de Gröbner pour l'ordre du degré (DRL). Dans la colonne «Avec ε» nous donnons le temps de calcul d'une base de Gröbner DRLDRL avec $\varepsilon < X_1, \ldots, X_n$ de l'idéal zéro-dimensionnel $P - \varepsilon = 0, \quad \overrightarrow{grad}_M\ (P) // \overrightarrow{AM}$ à résoudre. Dans la colonne «Sans ε», nous donnons le temps de calcul obtenu pour le même système où on a remplacé ε par 0. On constate que l'introduction de cet infinitésimal a un coût non négligeable.

La difficulté essentielle se situe dans le calcul de la Représentation Univariée Rationnelle. Dans le tableau 1.4, nous donnons le détail des temps de calcul de l'algorithme proposé dans la section 1.3. Cet algorithme peut être divisé en 3 étapes:

- la colonne «Système spécialisé» contient le temps de calcul de la base de Gröbner DRLDRL et de deux Représentations Univariées Rationnelles pour des valeurs de ε telles que l'idéal spécialisé reste radical et zéro-dimensionnel,
- la colonne «Remontée Hensel» contient le temps de calcul de la phase de remontée par l'opérateur de Newton et la reconstruction rationnelle de la RUR à coefficients infinitésimaux,

Hypersurface	**Avec** ε	**Sans** ε
Birk.3-1	1,23	0,02
Birk.3-2	0,01	0,01
Birk.3-3	1,23	0,02
Birk.3-4	0,02	0,01
Birk.3-5	0,99	0,05
Birk.3-6	1,97	0,06
Birk.3-7	0,38	0,06
Birk.3-8	0,16	0,12
Birk.3-9	0,16	0,12
Birk.3-10	1,85	0,45
Birk.3-11	1,53	0,46
Birk.3-12	2,07	0,59
Birk.3-13	2,75	0,48
Birk.3-14	5	1,1
Birk.3-15	10,20	1,95

TAB. 1.3 – *Influence de l'introduction d'infinitésimaux sur le temps de calcul d'une base de Gröbner DRL*

– la colonne «Vérification» contient le temps de calcul de la phase de vérification du résultat.

La colonne «Total» contient le temps de calcul total. Le symbole ∞ signifie que le calcul n'a pas abouti au bout de 12 heures.

Pour des systèmes zéro-dimensionnels de «petit degré» (deux ou trois dizaines) la majeure partie du temps de calcul est passée dans la phase dite «de remontée», la phase de vérification du résultat étant négligeable. En revanche, lorsque la taille des problèmes augmente, la phase de vérification du résultat n'est plus négligeable devant la phase de remontée. Cette tendance s'accentue au point que cette étape devient bloquante. Ceci s'explique par la croissance des coefficients et des degrés en ε que nous avons constatée.

Dans le tableau 1.5, nous donnons dans la première colonne le degré en ε du polynôme éliminant puis nous donnons dans la deuxième colonne le «degré utile» en ε, c'est-à-dire la puissance de ε minimale à partir de laquelle on peut tronquer les coefficients sans annuler un monôme du polynôme éliminant. Ainsi, sur ces exemples, le degré en ε et le «degré utile» en ε sont identiques.

Analysons maintenant le comportement de l'algorithme proposé dans la section 1.4. Nous avons implanté en Magma l'algorithme décrit dans le chapitre 6 de la partie I et utilisé les fonctionnalités de Magma calculant des développements en séries de Puiseux pour implanter l'algorithme décrit dans la section 1.4 de ce chapitre. Ce qui motivait ce travail était le constat que le premier algorithme calculait des polynômes caractéristiques qu'il était souvent inutile de calculer et que plus le calcul de la RUR à coefficients infi-

Hypersurface	**Degré**	**Système spécialisé**	**Remontée Hensel**	**Vérification**	**Total**
Birk.3-1	16	0,1	2,9	0,2	3,2
Birk.3-2	16	0,1	2,9	0,1	3,1
Birk.3-3	34	0,1	17,6	2,9	20,6
Birk.3-4	36	0,2	23	1,7	24,9
Birk.3-5	40	0,2	27,2	28,8	56,2
Birk.3-6	52	0,4	33	16,7	50,2
Birk.3-7	52	0,4	47,4	11,7	59,5
Birk.3-8	130	6	474,2	482,6	962,8
Birk.3-9	132	6,4	656,2	410,1	1072,7
Birk.3-10	136	6,6	935	5533	6474,6
Birk.3-11	138	7	609,6	8808	9424.6
Birk.3-12	138	7	909,5	5117	6033,5
Birk.3-13	252	55	3140,83	∞	∞
Birk.3-14	264	55	7075,6	∞	∞
Birk.3-15	272	60	7408,25	∞	∞

TAB. 1.4 – *Calcul de RUR à coefficients infinitésimaux: détail des temps de calcul*

Hypersurface	**Degré en ε**	**Degré utile en ε**
Birk.3-1	3	3
Birk.3-2	3	3
Birk.3-3	5	5
Birk.3-4	5	5
Birk.3-5	5	5
Birk.3-6	7	7
Birk.3-7	7	7
Birk.3-8	19	19
Birk.3-9	19	19
Birk.3-10	31	31
Birk.3-11	31	31
Birk.3-12	31	31

TAB. 1.5 – *RUR à coefficients infinitésimaux: comparaison des degrés utiles et du degré en ε*

Hypersurface	**LRB**	**LRBP**
Birk.3-1	19	< 1
Birk.3-2	5,4	< 1
Birk.3-3	1917	< 1
Birk.3-4	623	< 1
Birk.3-5	∞	< 1
Birk.3-6	∞	< 1
Birk.3-7	∞	< 1
Birk.3-8	∞	< 1
Birk.3-9	∞	< 1
Birk.3-10	∞	< 1
Birk.3-11	∞	< 1
Birk.3-12	∞	< 1

TAB. 1.6 – *Temps de calcul : comparaison des algorithmes calculant les limites des racines bornées*

nitésimaux serait difficile, plus ces caluls serait difficile. Ces calculs ont été faits via des calculs de résultants en Magma. Dans le tableau 1.6. nous comparons les temps de calcul de l'algorithme proposé dans le chapitre 6 de la partie I (colonne «LRB») avec celui de l'algorithme proposé dans la section 1.4 de ce chapitre (colonne «LRBP»).

Même si la technique utilisée pour calculer les polynômes caractéristiques nécessaires à l'algorithme décrit dans [84] n'est pas optimale (calculs de résultants), il apparait que l'algorithme que nous proposons s'avère plus efficace devant celui proposé dans [84], au moins sur ces exemples. Notons que l'usage des développements en séries de Puiseux rend cette phase de l'algorithme décidant du vide des hypersurfaces négligeable devant le reste des calculs. Enfin, ces calculs ont permis de montrer que les premiers éléments séparants choisis ne sont pas des éléments bien séparants.

Dans le tableau 1.7, nous comparons :

- le degré de l'idéal zéro-dimensionnel étudié par l'algorithme **HA3** (colonne **Ideal**),
- la somme des degrés des Représentations Univariées Rationnelles représentant les limites des racines bornées de l'idéal précédemment étudié (colonne **RB**),
- le nombre de cellules retournées par l'algorithme de Décomposition Cylindrique Algébrique.

Contrairement à ce qui avait été constaté pour les cas sans singularités, le nombre de cellules renvoyées par la Décomposition Cylindrique Algébrique est bien inférieur au degré des idéaux que nous considérons. Il convient de signaler que pour la plupart de ces hypersurfaces le lieu réel est soit vide soit strictement inclus dans l'ensemble des singularités ce qui en plus de simplifier le calcul de la Décomposition Cylindrique Algébrique réduit considérablement la taille de sa sortie. De plus, nous pouvons remarquer que nous n'avons pas fait varier le nombre de variables. Ceci atténue les conclusions que l'on pourrait tirer trop précipitamment du tableau 1.7. Il est clair que le degré de la Représentation Univariée Rationnelle à coefficients infinitésimaux est borné par $\mathcal{O}(d)^n$. Le problème réside

Hypersurface	Ideal	RB	CAD
Birk.3-1	16	12	13
Birk.3-2	16	7	12
Birk.3-3	34	25	12
Birk.3-4	36	16	9
Birk.3-5	40	31	12
Birk.3-6	52	37	12
Birk.3-7	52	38	12
Birk.3-8	130	45	10
Birk.3-9	132	47	10
Birk.3-10	136	48	21
Birk.3-11	138	50	21
Birk.3-12	138	50	21
Birk.3-13	252	32	10
Birk.3-14	264	60	14
Birk.3-15	272	60	41

TAB. 1.7 – *Algorithmes HA3 et CAD : comparaison de la taille de la sortie dans les cas singuliers*

dans le fait que ces objets sont trop difficiles à calculer avec les techniques actuellement connues.

Pour terminer, rappelons que la dernière étape de l'algorithme est le calcul des limites des racines bornées des points définies par la RUR. Il s'agit alors de trouver un élément bien séparant afin de pouvoir substituer 0 à ε dans la RUR calculée. Le nombre de points retournés par l'algorithme LRB devient alors comparable au nombre de cellules produites par l'algorithme de Décomposition Cylindrique Algébrique. Ce phénomène s'explique simplement sur nos exemples : certaines solutions du système zéro-dimensionnel à coefficients dans $K(\varepsilon)$ tendent vers un seul et même point lorsque ε tend vers 0.
Dans le tableau 1.8, nous comparons les temps de calcul de **HA3** et de l'algorithme de Décomposition Cylindrique Algébrique. Il est clair que ces résultats ne sont pas satisfaisants : nous ne bénéficions plus du fait que la sortie de nos algorithmes est de taille inférieure à celle de l'algorithme de Décomposition Cylindrique Algébrique.

1.7 Conclusions

Les résultats obtenus dans le cas des hypersurfaces contenant au plus un nombre fini de points singuliers sont encourageants. En effet, la taille de la sortie de l'algorithme **HA3** est bien inférieur au nombre de cellules renvoyées par l'algorithme de Décomposition Cylindrique Algébrique : les résultats théoriques sont expérimentalement vérifiés. L'impact sur les temps de calcul est sensible : l'algorithme **HA3** résoud dans ces cas-là des problèmes qui sont inaccessibles à l'algorithme de Décomposition Cylindrique Algébrique.

Hypersurface	**HA3**	**CAD**
Birk.3-1	3,2	0,21
Birk.3-2	3,1	0,11
Birk.3-3	20,6	4,44
Birk.3-4	20,9	3,45
Birk.3-5	56,2	11,1
Birk.3-6	50,2	15,7
Birk.3-7	59,5	18,2
Birk.3-8	962,8	0,2
Birk.3-9	1072,7	0,15
Birk.3-10	6474,6	1,86
Birk.3-11	9424,6	1,26
Birk.3-12	6033,5	1,88
Birk.3-13	∞	0,74
Birk.3-14	∞	1,12
Birk.3-15	∞	2,1

TAB. 1.8 – *Algorithmes HA3 et CAD: comparaison des temps de calcul dans les cas singuliers*

Nos efforts ont principalement porté sur le cas des hypersurfaces contenant une infinité de singularités pour lequel un certain nombre d'outils ont été implantés. Sur les exemples considérés qui sont extraits du problème d'interpolation de Birkhoff (voir chapitre 4 de la partie II), il est clair que l'algorithme **HA3** n'est pas satisfaisant, l'étape bloquante étant le calcul certifié d'une Représentation Univariée Rationnelle à coefficients infinitésimaux.

Les outils de calcul développés pour le calcul de Représentations Univariées Rationnelles à coefficients infinitésimaux traitent l'infinitésimal comme un paramètre alors qu'on pourrait espérer négliger les puissances élevées en ε et tronquer les calculs. Le tableau 1.5 montre que, sur les exemples considérés, l'implantation d'une arithmétique infinitésimale ne serait que d'un faible secours: les «degrés utiles» en ε ne sont pas nécessairement faibles. Une étude plus poussée serait nécessaire sur ce point.

De plus, en observant le tableau 1.7, on remarque que sur les exemples inaccessibles à **HA3** le degré de l'idéal zéro-dimensionnel à coefficients dans $K(\varepsilon)$ est grand comparativement à la somme des degrés des Représentations Univariées Rationnelles représentant les limites des racines bornées de l'idéal étudié.
Ainsi, comme dans l'algorithme décrit dans le chapitre 6 de la partie I, l'algorithme **HA3** souffrirait d'un mauvais contrôle des données intermédiaires. Dans le Chapitre suivant, nous étudions une alternative aux méthodes de déformation pour résoudre le cas des hypersurfaces contenant une infinité de points singuliers. Nous verrons que ceci nous amène à considérer le cas des systèmes polynomiaux.

Chapitre 2

Systèmes d'équations

Résumé

Ces travaux ont effectués en collaboration avec P. Aubry et F. Rouillier. La caractérisation des points critiques de la fonction distance est généralisée au cas des systèmes polynomiaux (voir [83]). Les singularités d'une variété V sont étudiées comme une variété à part entière, de dimension inférieure, sans déformation infinitésimale. Les propriétés d'ensembles triangulaires extraits de bases de Gröbner lexicographiques permettent de faciliter le calcul des points critiques et améliorent l'algorithme proposé. Puis, nous montrons comment substituer les calculs de bases de Gröbner par les décompositions en ensembles triangulaires au sens de Kalkbrener dans nos algorithmes. Nous simplifions ensuite les algorithmes obtenus dans le cas de la position Shape Lemma et dans le cas des variétés réelles compactes.

Sommaire

Introduction

Soit une hypersurface définie par $P = 0$ (où $P \in K[X_1, \ldots, X_n]$ est sans carrés) contenant une infinité de points singuliers. D'après la preuve du lemme 1.2 du chapitre précédent, on peut trouver un point A tel que l'ensemble des solutions du système $P = 0$, $\overrightarrow{grad}_M\ (P) // \overrightarrow{AM}$ est la réunion d'un ensemble fini de points et du lieu singulier de V. Ce lieu singulier – qui est une variété algébrique de dimension strictement inférieure à celle de l'hypersurface – est défini par un système d'équations polynomiales. Si on sait en donner au moins un point par composante semi-algébriquement connexe sans introduire d'infinitésimaux, on sait alors résoudre le cas des hypersurfaces sans infinitésimaux. Ainsi, pour pouvoir résoudre le cas des hypersurfaces contenant une infinité de singularités, il faut pouvoir caractériser les points critiques de la fonction distance sur une variété algébrique définie par un système d'équations polynomiales.

Dans [30], les auteurs proposent un nouvel algorithme pour décider du vide d'ensembles semi-algébriques sans faire la somme des carrés des systèmes polynomiaux à résoudre. En s'inspirant des techniques proposées dans [15], ils proposent de traiter le cas des variétés contenant une infinité de singularités en utilisant le fait que la dimension du lieu singulier est inférieure à celle de la variété considérée. Néanmoins, l'algorithme proposé nécessite l'introduction récursive d'infinitésimaux. Ceci est lié au fait que les auteurs continuent d'utiliser la fonction de projection sur un axe de coordonnée.

Dans [9, 10, 11], les auteurs proposent un algorithme utilisant la fonction de projection sur un axe. Les méthodes d'élimination algébrique utilisées sont basées sur des techniques d'évaluation et sur la représentation des données par des programmes d'évaluation. En revanche, les auteurs ne proposent pas de solutions pour donner au moins un point par composantes semi-algébriquement connexes en toute généralité.

Dans ce chapitre, nous proposons un algorithme *ne faisant intervenir aucune déformation infinitésimale* et qui trouve au moins un point par composante semi-algébriquement connexe sur une variété algébrique réelle *quelconque*. Ce chapitre est basé sur [8], travail effectué en collaboration avec P. Aubry et F. Rouillier. Les algorithmes proposés relèvent de la méthode des points critiques. Ils consistent à résoudre un problème d'optimisation de la fonction distance à un point restreinte à la variété algébrique réelle que l'on désire étudier.

Dans la première section, nous décrivons un algorithme calculant au moins un point par composante semi-algébriquement connexe sur une variété algébrique réelle. Cet algorithme est basé sur une caractérisation algébrique d'un ensemble contenant les points critiques de la fonction distance à un point arbitrairement choisi, restreinte à une variété algébrique réelle. Cet ensemble est de dimension strictement inférieure à celle de la variété que l'on désire étudier.

Dans la deuxième section de ce chapitre, nous optimisons l'algorithme. En particulier, nous utilisons intensivement des développements concernant la résolution algébrique des systèmes polynomiaux (théorie des ensembles triangulaires et décomposition en premiers).

Les méthodes d'élimination utilisées dans les algorithmes obtenus sont fondées sur les bases de Gröbner. Dans la troisième section de ce chapitre, nous montrons comment substituer ces calculs par des décompositions en ensembles triangulaires. Ce travail a été effectué en collaboration avec P. Aubry et F. Rouillier. La difficulté repose dans le fait que, contrairement aux bases de Gröbner, les ensembles triangulaires sont une représentation «paresseuse» des variétés algébriques. Ainsi, la perte d'information engendrée rend délicate l'adaptation de nos algorithmes à ces techniques de calculs récentes.

Dans la quatrième section de ce chapitre, nous étudions deux cas particuliers importants. Le premier est le cas des variétés équi-dimensionnelles en position générales. Nous montrons comment ramener l'étude de ces variétés à celle d'une hypersurface dans C^{d+1} où d est la dimension de la variété étudiée. Le deuxième cas est celui des variétés algébriques réelles compactes. Nous montrons alors comment obtenir des algorithmes donnant au moins un point par composante semi-algébriquement connexe et compacte en utilisant la fonction de projection sur un axe et les résultats des sections précédentes. D'un point de vue plus général, cette étude annexe est utile dans la mesure où, comme J.-C. Faugère nous l'a fait remarquer, pour certains problèmes physiques seules ces composantes intéressent l'utilisateur. Dans [11], les auteurs proposent un algorithme spécifiquement dédié au cas compact. Comparativement, nos algorithmes sont plus généraux : aucune hypothèse, ni sur la variété étudiée, ni sur le système d'équations qui la définit n'est requise. En particulier nous apportons une réponse au problème que pose la présence de singularités. Notons néanmoins que, comme dans [11], les points retournés par nos algorithmes peuvent appartenir à des composantes non compactes et qu'aucun test de compacité n'est donné.

Enfin, la dernière section de ce chapitre est consacrée à la validation expérimentale de nos algorithmes. Nous présentons dans un premier temps les implantations utilisées. Nous comparons ensuite, sur des exemples d'hypersurfaces contenant une infinité de singularités, la stratégie proposée dans ce chapitre pour gérer la présence de singularités à celle étudiée dans le chapitre précédent et qui consiste à déformer l'hypersurface. Nous en déduisons que notre approche permet de diminuer significativement la taille des données intermédiaires (degré des systèmes zéro-dimensionnels étudiés) apparaissant en cours de calcul et que l'impact sur les temps de calcul est sensible. Puis, nous comparons ces algorithmes à l'algorithme de Décomposition Cylindrique Algébrique : nous avons constaté que la taille de la sortie de nos algorithmes est sensiblement inférieure à celle de l'algorithme de Décomposition Cylindrique Algébrique. En terme de temps de calcul, les répercussions sont sensibles : nous pouvons traiter des problèmes qui sont inabordables par l'algorithme de Collins. Enfin, nous procédons à l'étude plus spécifique des algorithmes proposés dans les troisièmes et quatrièmes sections de ce chapitre.
Dans ce chapitre, on note K un corps ordonné, R sa cloture réelle et C sa cloture algébrique.

2.1 L'algorithme

Si $(P_1,\ldots,P_s)$ est une famille de polynômes dans $K[X_1,\ldots,X_n]$, on note $V(P_1,\ldots,P_s) \subset C^n$ la variété algébrique définie par le système d'équations polynomiales :

$$P_1 = \ldots = P_s = 0$$

et $I = \langle P_1,\ldots,P_s \rangle$ l'idéal de $K[X_1,\ldots,X_n]$ engendré par cette famille de polynômes.

Reprenons le deuxième exemple page?? du chapitre précédent. On étudie donc l'hypersurface définie par :

$$x^2 - y^2z^2 + z^3 = 0$$

En choisissant le point $A = (1,2,3)$ on obtient le système suivant :

$$\begin{cases} x^2 - y^2z^2 + z^3 = 0 \\ 2xy - 4x + 2yz^2x - 2yz^2 = 0 \\ -2yz^3 + 3yz^2 + 2y^3z - 4y^2z + 6z^2 = 0 \\ 2xz - 6x + 2y^2zx - 2y^2z - 3z^2x + 3z^2 = 0 \end{cases}$$

On a vu que ce système est de dimension 1 et de degré 1. Il contient l'ensemble des singularités de l'hypersurface :

$$\begin{cases} z = 0 \\ x = 0 \end{cases}$$

qui est de dimension strictement inférieure à celle de l'hypersurface. Par ailleurs, si on effectue une décomposition équi-dimensionnelle du système, on obtient (comme prévu par la preuve du lemme 1.2 du chapitre précédent) en plus de l'ensemble des singularités l'ensemble zéro-dimensionnel de degré 15 suivant (décrit par une base de Gröbner lexicographique) :

```
[5398746452967737165536*x-39839127175867630680*z^14+26004917309531866 7844*z^13_
```

D'après la section 1.6 du chapitre précédent, on a vu que les déformations infinitésimales (plus précisément les calculs de Représentations Univariées Rationnelles à coefficients infinitésimaux) doivent être évités. On voit sur cet exemple qu'en caractérisant les points critiques de la fonction distance à un point arbitrairement choisi, sur une variété algébrique définie par un ensemble fini de polynômes, on est ramené à l'étude d'une sous-variété de dimension inférieure (l'ensemble des singularités de la variété). C'est cette idée que l'on retrouve antérieurement dans [30] que nous allons essayer de mettre en œuvre.

Notation 2.1 *Soit $S = \{P_1,\ldots,P_s\}$ un ensemble de polynômes dans $K[X_1,\ldots,X_n]$ tel que $V = V(S)$ soit une variété de dimension d. Etant donné un point $A \in C^n$, on définit l'ensemble algébrique suivant :*

$$\mathcal{C}(V,A) = \{M \in V \mid \text{rank}(\overrightarrow{grad}_M\ (P_1),\ldots,\ \overrightarrow{grad}_M\ (P_s),\ \overrightarrow{AM} \leq n-d)\}$$

La construction et l'étude de $\mathcal{C}(V,A)$ n'a d'intérêt que si :

– $\mathcal{C}(V,A)$ intersecte chaque composante semi-algébriquement connexe de $V \bigcap R^n$,

- $\mathcal{C}(V,A)$ est strictement plus petite que V et en particulier qu'elle soit de dimension strictement inférieure à celle de V.

Sous ces conditions, il apparait clairement que nous pourrons obtenir un algorithme calculant au moins un point par composante semi-algébriquement connexe de $V \bigcap R^n$, sans faire la moindre déformation infinitésimale. Malheureusement, les conditions ci-dessus ne sont pas vraies en général, comme le montrent les exemples ci-dessous.

- **Exemple 1 :** Considérons la variété algébrique V définie par
$$P = (x^2 + y^2 - 1)^2 = 0$$
Il est aisé de s'apercevoir que quelque soit le point A choisi la variété algébrique définie par
$$P(M) = 0, \quad \overrightarrow{AM} \;//\; \overrightarrow{grad}_M\,(P)$$
est de dimension 1. En effet, l'ensemble des points qui vérifient :
$$P(M) = 0 \quad \overrightarrow{grad}_M\,(P) = \overrightarrow{0}$$
est égal à V. Notons que dans ce cas l'idéal $\langle P \rangle$ n'est pas radical.
Enfin, remarquons que l'ensemble des singularités de V est vide.
- **Exemple 2 :** Le cas des idéaux radicaux n'est pas exempt de difficultés lui aussi. Considérons la variété algébrique V définie par
$$\begin{cases} P_1 = (X_1^2 + X_2^2 - 1)(X_1 - 2) = 0 \\ P_2 = (X_1 - 2)X_3 = 0 \end{cases}$$
L'ensemble $V \bigcap R^3$ est la réunion du plan défini par l'équation $X_1 = 2$ et du cercle défini par les équations $X_1^2 + X_2^2 - 1 = 0$ et $X_3 = 0$. Il est facile de constater que chaque point du cercle est régulier et vérifie
$$rank\left(\begin{bmatrix} 3\,X_1{}^2 - 4\,X_1 + X_2{}^2 - 1 & 2\,X_2\,(X_1 - 2) & 0 \\ X_3 & 0 & X_1 - 2 \end{bmatrix}\right) = 2\,.$$
Ainsi, $\mathcal{C}(V,A)$ n'intersecte pas chaque composante semi-algébriquement connexe de $V \bigcap R^n$ puisqu'il ne peut contenir aucun point du cercle.

Dans l'exemple 2 ci-dessus, V est composée de composantes irréductibles de dimensions différentes. Les points critiques de la fonction distance qui ne se trouvent pas dans la composante principale de dimension d ne se trouveront donc pas dans $\mathcal{C}(V,A)$ puisque ces tels points qui ne sont pas singuliers vérifient :
$$\text{rank}(\overrightarrow{grad}_M\,(P_1), \ldots, \overrightarrow{grad}_M\,(P_s), \overrightarrow{AM}) > n - d.$$
Par ailleurs, soit $M \in V$ tel que :
$$\dim(\overrightarrow{grad}_M\,(P_1), \ldots, \overrightarrow{grad}_M\,(P_s)) < n - d.$$

On a alors $M \in \mathcal{C}(V,A)$. Ceci arrive en particulier lorsque M est un point singulier d'une composante irréductible de dimension d dans V.

Notation 2.2 *Soit V une variété algébrique de dimension d, $\{P_1,\ldots,P_s\}$ une famille de polynômes dans $K[X_1,\ldots,X_n]$ telle que $I(V) = \langle P_1,\ldots,P_s\rangle$. On note* Sing$(V)$ *la variété :*

$$\mathrm{Sing}(V) = \{M \in V \mid \mathrm{rank}(\overrightarrow{grad}_M\,(P_1),\ldots,\overrightarrow{grad}_M\,(P_s)) < n-d\}.$$

Le théorème suivant permet d'obtenir un algorithme.

Théorème 2.1 *Soit V une variété algébrique équi-dimensionnelle de dimension d et $\{P_1,\ldots,P_s\} \subset K[X_1,\ldots,X_n]$ tel que $I(V) = \langle P_1,\ldots,P_s\rangle$.*
Il existe D un entier positif suffisamment grand, tel qu'il existe au moins un point A dans $\{1,\ldots,D\}^n$ vérifiant :

1. *$\mathcal{C}(V,A)$ intersecte chaque composante semi-algébriquement connexe de $V \bigcap R^n$,*
2. *$\mathcal{C}(V,A) = \mathrm{Sing}(V) \bigcup V_0$, où V_0 est un ensemble fini de points dans C^n.*

De plus, $dim(\mathcal{C}(V,A)) < dim(V)$.

Preuve :

1. Soit A un point quelconque de R^n et $\mathcal{D}$ une composante semi-algébriquement connexe de $V \bigcap R^n$. Si $\mathcal{D} \bigcap \mathrm{Sing}(V) \neq \emptyset$, il est clair que $\mathcal{C}(V,A) \bigcap \mathcal{D} \neq \emptyset$ car
$$\mathrm{Sing}(V) \subset \mathcal{C}(V,A).$$
Supposons maintenant que $\mathcal{D} \bigcap \mathrm{Sing}(V) = \emptyset$ et soit $M \in \mathcal{D}$ à distance minimale de A. Soit $\mathcal{S}(A,r)$ la boule de centre A et de rayon $r = d(A,M)$. Puisque le point M est à distance minimale de A, la boule $\mathcal{S}$ et la variété V sont tangeantes en M et
$$\overrightarrow{AM} \in \mathrm{Vect}(\overrightarrow{grad}_M\,(P_1),\ldots,\overrightarrow{grad}_M\,(P_s)).$$
Puisque V est équi-dimensionnelle de dimension d, et puisque $\langle P_1,\ldots,P_s\rangle$ est radical, l'espace vectoriel $\mathrm{Vect}(\overrightarrow{grad}_M\,(P_1),\ldots,\overrightarrow{grad}_M\,(P_s))$ est de dimension inférieure ou égale à $n-d$ et donc $M \in \mathcal{C}(V,A)$.
2. Afin de montrer le deuxième point du théorème, nous allons montrer que pour tout point A choisi en dehors d'une variété algébrique de C^n de dimension strictement inférieure à n, $\mathcal{C}(V,A)$ s'écrit comme la réunion de Sing(V) et de V_0 (où V_0 est un ensemble fini de points).
   (a) On suppose dans un premier temps que $s = n-d$. Soient $\lambda_1,\ldots,\lambda_{n-d}$ $n-d$ nouvelles variables, et $Q_1,\ldots,Q_n$ dans $K[X_1,\ldots,X_n,\lambda_1,\ldots,\lambda_{n-d}]$ définis par :
   $$Q_j = \sum_{i=1}^{n-d} \lambda_i \frac{\partial P_i}{\partial X_j} - X_j$$
   Soit $\mathcal{H}$ le sous-ensemble de C^{2n-d}
   $$\mathcal{H} = \{(M,\lambda_1,\ldots,\lambda_s) \in C^{2n-d} \mid M \in V(P_1,\ldots,P_{n-d}) \setminus \mathrm{Sing}(V(P_1,\ldots,P_{n-d}))\}.$$

On considère l'application :

$$F: \begin{array}{ccc} \mathcal{H} & \longrightarrow & C^n \\ (M,\lambda_1,\ldots,\lambda_{n-d}) & \longmapsto & (Q_1(M,\lambda_1,\ldots,\lambda_{n-d}),\ldots,Q_n(M,\lambda_1,\ldots,\lambda_{n-d})) \end{array}$$

Comme $V(P_1,\ldots,P_{n-d})$ est équi-dimensionnelle de dimension d et que l'idéal $\langle P_1,\ldots,P_{n-d}\rangle$ est radical, les valeurs critiques de F sont les points $\chi \in \mathcal{H}$ tels que :

$$\dim(\mathrm{Vect}(\overrightarrow{grad}_\chi\,(P_1),\ldots,\overrightarrow{grad}_\chi\,(P_{n-d}),\overrightarrow{grad}_\chi\,(Q_1),\ldots,\overrightarrow{grad}_\chi\,(Q_n))) < 2n-d.$$

Soit $Jac(P_1,\ldots,P_{n-d},Q_1,\ldots,Q_n)$ la matrice jacobienne $(2n-d,2n-d)$ associée aux polynômes $P_1,\ldots,P_{n-d},Q_1,\ldots,Q_n$ et notons $\mathcal{J}$ son déterminant. L'ensemble des valeurs critiques de F est :

$$\mathcal{B} = \{A = (a_1,\ldots,a_n) \in C^n \mid \mathcal{H} \bigcap V(Q_1-a_1,\ldots,Q_n-a_n) \bigcap V(\mathcal{J}) \neq \emptyset\}.$$

D'après le théorème de Sard (voir [73]), on en déduit que $\mathcal{B}$ est un ensemble constructible dans C^n de dimension strictement inférieure à n. Il existe donc un entier D tel que l'ensemble $\{0,\ldots,D\}^n$ contienne un point qui ne soit pas dans $\mathcal{B}$. Soit $A \notin \mathcal{B}$. Pour tout $M \in \mathcal{H} \bigcap V(Q_1-a_1,\ldots,Q_n-a_n)$, $\mathcal{J}(M) \neq 0$ (autrement dit, $Jac(P_1,\ldots,P_{n-d},Q_1,\ldots,Q_n)$ est de rang $2n-d$ en M), ce qui implique que $\mathcal{H} \bigcap V(Q_1-a_1,\ldots,Q_n-a_n)$ est un ensemble fini de points. Si on considère la projection π définie par :

$$\pi: \begin{array}{ccc} C^{2n-d} & \longrightarrow & C^n \\ (x_1,\ldots,x_n,\ell_{i_1},\ldots,\ell_{i_{n-d}}) & \longmapsto & (x_1,\ldots,x_n) \end{array},$$

on peut voir que $\mathcal{C}(V,A) = \mathrm{Sing}(V) \bigcup \pi(\mathcal{H} \cap V(Q_1-a_1,\ldots,Q_n-a_n))$, ce qui implique que $\mathcal{C}(V,A) = \mathrm{Sing}(V) \cup V_0$, où V_0 est un ensemble fini de points.

(b) On considère le cas général. Comme $V(P_1,\ldots,P_s)$ est équi-dimensionnelle de dimension d et que $\langle P_1,\ldots,P_s\rangle$ est radical, pour tout point régulier $M \in V(P_1,\ldots,P_s)$, il existe une famille d'indices $\mathcal{I} = \{i_1,\ldots,i_{n-d}\} \subset \{1,\ldots,s\}$ telle que :

$$\dim(Vect(\overrightarrow{grad}_M\,(P_{i_1}),\ldots,\overrightarrow{grad}_M\,(P_{i_{n-d}})) = n-d$$

Pour tout $\mathcal{I} \subset \{1,\ldots,s\}$ tel que $\sharp\mathcal{I} = n-d$, on pose :

$$Q_j^{\mathcal{I}} = \sum_{i\in\mathcal{I}} \lambda_i \frac{\partial P_i}{\partial X_j} - X_j$$

(où $\lambda_{i_1},\ldots,\lambda_{i_{n-d}}$ sont des nouvelles variables) et

$$\mathcal{H}^{\mathcal{I}} = \{(M,\lambda_{i_1},\ldots,\lambda_{i_{n-d}} \mid M \in V(P_1,\ldots,P_s) \setminus \mathrm{Sing}(V(P_1,\ldots,P_s)),$$

$$\dim(\mathrm{Vect}(\overrightarrow{grad}_M\,(P_{i_1}),\ldots,\overrightarrow{grad}_M\,(P_{i_{n-d}})) = n-d\}.$$

On considère l'application :

$$\begin{array}{rccc} F^{\mathcal{I}}: & \mathcal{H}^{\mathcal{I}} & \longrightarrow & C^n \\ & (M,\lambda_{i_1},\ldots,\lambda_{i_{n-d}}) & \longmapsto & (Q_1^{\mathcal{I}}(M,\lambda_{i_1},\ldots,\lambda_{i_{n-d}}),\ldots,Q_n^{\mathcal{I}}(M,\lambda_{i_1},\ldots,\lambda_{i_{n-d}})) \end{array}$$

De manière analogue au cas $s = n - d$, on définit $\mathcal{J}^{\mathcal{I}}$, le déterminant de la matrice $(2n - d,2n - d)$ jacobienne associée à la famille de polynômes $(P_{i_1},\ldots,P_{i_{n-d}},Q_1^{\mathcal{I}},\ldots,Q_n^{\mathcal{I}})$. L'ensemble des valeurs critiques de $F^{\mathcal{I}}$ s'écrit alors :

$$\mathcal{B}^{\mathcal{I}} = \{A = (a_1,\ldots,a_n) \in C^n \mid \mathcal{H}^{\mathcal{I}} \cap V(Q_1^{\mathcal{I}} - a_1,\ldots,Q_n^{\mathcal{I}} - a_n) \cap V(\mathcal{J}^{\mathcal{I}}) \neq \emptyset\}$$

et $\mathcal{B}^{\mathcal{I}}$ est un ensemble constructible de C^n de dimension strictement inférieure à n. Ainsi, $\mathcal{B} = \bigcup_{\mathcal{I}\subset\{1,\ldots,s\},\sharp\mathcal{I}=n-d} \mathcal{B}^{\mathcal{I}}$ est un ensemble constructible de dimension strictement inférieure à n. Si $A \notin \mathcal{B}$, alors, pour tout $\mathcal{I} \subset \{1,\ldots,s\}$ tel que $\sharp\mathcal{I} = n - d$, $\mathcal{H}^{\mathcal{I}} \bigcap V(Q_1^{\mathcal{I}} - a_1,\ldots,Q_n^{\mathcal{I}} - a_n)$ est un ensemble fini de points (éventuellement vide). En remarquant que

$$\mathcal{C}(V,A) = \mathrm{Sing}(V) \bigcup_{\mathcal{I}\subset\{1,\ldots,s\},\sharp\mathcal{I}=n-d} \pi(\mathcal{H}^{\mathcal{I}} \bigcap V(Q_1^{\mathcal{I}} - a_1,\ldots,Q_n^{\mathcal{I}} - a_n))$$

on conclut que $\mathcal{C}(V,A)$ est la réunion de $\mathrm{Sing}(V)$ et d'un ensemble fini de points.

■

Soit V une variété équi-dimensionnelle. Etant donnée une famille de générateurs de $I(V)$ on peut choisir un point A et calculer $\mathcal{C}(V,A)$ tel que $\dim(\mathcal{C}(V,A)) < \dim(V)$. D'après le Théorème 2.1, une décomposition équi-dimensionnelle de $\mathcal{C}(V,A)$ donne une composante zéro-dimensionnelle V_0 et plusieurs autres composantes équi-dimensionnelles de dimension positive. On peut alors appliquer le théorème 2.1 à chacune de ces composantes.

L'algorithme que nous proposons consiste à appliquer pas à pas le processus décrit ci-dessus en calculant à chaque étape des décompositions équi-dimensionnelles des variétés intermédiaires obtenues. A la fin, nous obtenons un ensemble de systèmes zéro-dimensionnels contenant au moins un point par composante semi-algébriquement connexe dans la variété $V \bigcap R^n$.

Notation 2.3 *Pour $A \in C^n$, $\mathcal{Q} = \{Q_1,\ldots,Q_s\} \subset K[X_1,\ldots,X_n]^s$, et $d \in \mathbb{N}$, $0 \leq d < n$, on définit $\Delta_{A,d}(\mathcal{Q})$ comme étant l'ensemble de tous les mineurs d'ordre $(n-d+1,n-d+1)$ de la matrice*

$$\left[\left[\frac{\partial Q_i}{\partial X_j} \right]_{(i=1,\ldots,n,j=1,\ldots,s)} \middle| \overrightarrow{AM} \right]$$

D'après les résultats ci-dessus, les routines de base nécessaires pour l'implantation d'un tel algorithme qui calcule cet ensemble de systèmes zéro-dimensionnels sont les suivantes :

- **EquiDim** : prend en entrée un système d'équations polynomiales S et retourne une liste de systèmes de générateurs $\mathcal{P}_d,\ldots,\mathcal{P}_0$ engendrant des idéaux radicaux et équi-dimensionnels, tels que $V(S) = V(\mathcal{P}_d) \bigcup \ldots \bigcup V(\mathcal{P}_0)$.
- **Dim** : prend en entrée un système de générateurs d'un idéal et calcule la dimension de la variété associée,

- **Minors** : prend en entrée une famille finie de polynômes $\mathcal{Q}$, un entier d et un point A, et calcule $\Delta_{\mathcal{Q},d,A}(\mathcal{Q})$).

Nous obtenons l'algorithme **SA1** (**S**ystèmes **A**lgorithme **1**).

Algorithme SA1

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S)\bigcap R^n$.

1. list := EquiDim(S), result := [],
2. Choisir un point A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), poser d := Dim(S) et enlever S de list,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - (*) Q = Minors(S,d,A) $\bigcup S$
     - u = Dim(Q)
     - Si $u = d$ choisir un autre point A et aller au pas (*).
     - list := list $\bigcup$ EquiDim(Q)
4. Retourner result.

2.2 Optimisations

Dans cette section, nous décrivons les optimisations que nous avons apportées à l'algorithme **SA1**. Soit $\mathcal{G} \subset K[X_1,\ldots,X_n]$ une base de Gröbner contenant s polynômes et engendrant un idéal équi-dimensionnel et radical de dimension d. D'aprés les résultats de la section précédente, le nombre de déterminants qui sont calculés par l'algorithme **SA1** est

$$\left(\begin{array}{c} s \\ n-d \end{array}\right)\left(\begin{array}{c} n \\ n-d+1 \end{array}\right).$$

Il est clair qu'un tel facteur combinatoire n'a que peu d'incidences dans le cas des hypersurfaces, mais il devient limitant sur des problèmes significatifs, de co-dimension plus grande que 1. Les améliorations décrites ci-dessous ont donc pour but de faciliter la résolution de ces cas.

Dans le premier paragraphe de cette section, nous allons montrer comment, en donnant un peu plus de propriétés à $\mathcal{G}$, nous allons pouvoir en extraire une famille de $n-d$ polynômes représentant «presque tous les points» de $V(\mathcal{G})$ et permettant de ne calculer que d déterminants. Nous verrons en particulier que cette famille est un ensemble triangulaire de polynômes.

Dans le deuxième paragraphe de cette section, nous montrons comment les optimisations s'appliquent aux décompositions en idéaux premiers. Puis, nous montrons comment ces décompositions permettent :

- de réduire considérablement la taille de la sortie de nos algorithmes,

– d'éviter des calculs intuiles et donc de réduire les temps de calcul.

En effet en travaillant sur des composantes irréductibles, on évite l'étude des points qui appartiennent à l'intersection de deux de ces composantes.

2.2.1 L'apport des ensembles triangulaires

Soit $\mathcal{G} \subset K[X_1,\ldots,X_n]$ une **base de Gröbner lexicographique réduite** engendrant un idéal radical équi-dimensionnel de dimension d pour l'ordre $X_1 < \ldots < X_n$. Pour $p \in K[X_1,\ldots,X_n]$, on note mvar(p) (variable principale de p) la plus grande variable apparaissant dans p pour l'ordre $X_1 < \ldots < X_n$.

Soit $\mathcal{T} = (t_{d+1},\ldots,t_n)$ un ensemble triangulaire extrait de $\mathcal{G}$ tel que :

- $\forall g \in \mathcal{G}$ il existe $i \in \{d+1,\ldots,n\}$ vérifiant
  - (i) $\mathrm{mvar}(t_i) = \mathrm{mvar}(g)$,
  - (ii) $\deg(t_i,\mathrm{mvar}(t_i)) \leq \deg(g,\mathrm{mvar}(t_i))$,
- $\forall i \in \{d+1,\ldots,n\}$ il n'existe pas de polynômes $g \in \mathcal{G}$ de même variable principale que t_i et de monôme dominant inférieur à celui de t_i pour l'ordre lexicographique.

Notons qu'un tel ensemble triangulaire est unique. On note **ExtractTriangular** une routine qui prend en entrée une base de Gröbner lexicographique réduite et qui retourne l'ensemble triangulaire extrait de la base d'entrée et vérifiant les propriétés ci-dessus.

Dans la suite, on suppose que la base de Gröbner lexicographique réduite $\mathcal{G}$ est telle que :

- l'ensemble triangulaire $\mathcal{T} \subset \mathcal{G}$ extrait de $\mathcal{G}$ par **ExtractTriangular** est **régulier** et **séparable**,
- $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.

Notons que de telles suppositions impliquent que $\langle \mathcal{G} \rangle$ est équi-dimensionnel car il est saturé d'un ensemble triangulaire régulier (voir chapitre 5 partie I ou [5]), et que $\langle \mathcal{G} \rangle$ est un idéal radical car il est saturé d'ensemble triangulaire $\mathcal{T}$ séparable (voir chapitre 5 partie I ou [5]).

Soit $M = (x_1,\ldots,x_n)$, A un point de K^n, $d = \dim(V(\mathcal{G}))$, et considérons pour $j = 1,\ldots,d$ la liste des mineurs d'ordre $(n-d+1)$ extraite de $\Delta_{A,d}(\mathcal{T})$:

$$\Gamma_A(\mathcal{T}) = \{\Gamma_A^{(j)}(\mathcal{T}) = det(\mathcal{M}_A^{(j)}), j = 1,\ldots,d\}$$

où

$$\mathcal{M}_A^{(j)} = \left[\begin{array}{c|c} \left[\frac{\partial t_i}{\partial X_j}\right]_{i=d+1,\ldots,n} & X_j - a_j \\ \hline & X_{d+1} - a_{d+1} \\ \mathcal{U}_{\mathcal{T}} = \left[\frac{\partial t_i}{\partial X_k}\right]_{k=d+1,\ldots,n}^{i=d+1,\ldots,n} & \vdots \\ & X_n - a_n \end{array}\right]$$

Sans nuire à la généralité, on peut supposer que $\mathrm{mvar}(t_i) = X_i$, ce qui rend les mineurs $\Gamma_A^{(j)}(\mathcal{T})$ faciles à calculer puisque $\mathcal{U}_{\mathcal{T}}$ est triangulaire supérieure. Nous allons montrer que nous pouvons substituer le calcul de $\Delta_{A,d}(\mathcal{G})$ par celui de $\Gamma_A(\mathcal{T})$ dans notre algorithme :

Proposition 2.1 *Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite et $\mathcal{T}$ un ensemble triangulaire vérifiant les hypothèses ci-dessus. Soit $\mathcal{D}(V(\mathcal{G}),A) = V(\mathcal{G}) \bigcap V(\Gamma_A(\mathcal{T}))$, $d =$* $\dim(\mathcal{G})$ *et* $\mathrm{Sep}(\mathcal{T}) = \prod_{i=d+1}^{n} \frac{\partial t_i}{X_i}$*. Si A est un point de K^n tel que* $\dim(\mathcal{C}(V(\mathcal{G}),A)) <$ $\dim(V(\mathcal{G}))$ *alors on a :*
- $\mathcal{C}(V(\mathcal{G}),A) \subset \mathcal{D}(V(\mathcal{G}),A)$,
- $(\mathcal{D}(V(\mathcal{G}),A) \setminus V(\mathrm{Sep}(\mathcal{T}))) \subset V_0$,
- $\dim(\mathcal{D}(V(\mathcal{G}),A) \bigcap V(\mathrm{Sep}(\mathcal{T}))) < \dim(V(\mathcal{G}))$.

En particulier, $\dim(\mathcal{D}(V(\mathcal{G}),A)) < \dim(V(\mathcal{G}))$ *et $\mathcal{D}(V(\mathcal{G}),A)$ s'intersecte avec toute composante semi-algébriquement connexe de $V(\mathcal{G})$.*

Preuve :
- Puisque $\mathcal{T} \subset \mathcal{G}$, $\Gamma_A(\mathcal{T}) \subset \Delta_{A,d}(\mathcal{T}) \subset \Delta_{A,d}(\mathcal{G})$, alors :
$$\mathcal{C}(V(\mathcal{G}),A) = V(\mathcal{G}) \bigcap V(\Delta_{A,d}(\mathcal{G})) \subset V(\mathcal{G}) \bigcap V(\Delta_{A,d}(\mathcal{T})) \subset V(\mathcal{G}) \bigcap V(\Gamma_A(\mathcal{T})).$$
- Soit $M \in \mathcal{D}(V(\mathcal{G}),A) \setminus V(Sep(\mathcal{T}))$. On a $det(\mathcal{U}_{\mathcal{T}}(M)) \neq 0$ donc
$$\mathrm{rank}(\overrightarrow{grad}_M\ (t_{d+1}), \ldots, \overrightarrow{grad}_M\ (t_n)) \geq n - d,$$
et par conséquent $\mathrm{rank}(\overrightarrow{grad}_M\ (g_1), \ldots, \overrightarrow{grad}_M\ (g_s)) \geq n - d$. D'une part,
$$\Gamma_A^{(i)}(\mathcal{T})(M) = 0\ ,\ \forall i = 1, \ldots, d$$
et donc $rank(\overrightarrow{grad}_M\ (t_{d+1}), \ldots, \overrightarrow{grad}_M\ (t_n), \overrightarrow{AM}) = n - d$. D'autre part, la dimenseion de $V(\mathcal{G})$ est d, donc
$$\mathrm{rank}(\overrightarrow{grad}_N\ (g_1), \ldots, \overrightarrow{grad}_N\ (g_s)) \leq n - d\ ,\ \forall N \in V(\mathcal{G})$$
et ainsi $M \notin Sing(V(\mathcal{G}))$. De plus,
$$Vect(\overrightarrow{grad}_M\ (g_1), \ldots, \overrightarrow{grad}_M\ (g_s)) = Vect(\overrightarrow{grad}_M\ (t_{d+1}), \ldots, \overrightarrow{grad}_M\ (t_n))$$
ce qui montre que $M \in V_0 = \mathcal{C}(V(\mathcal{G}),A) \setminus \mathrm{Sing}(V(\mathcal{G}))$.
- D'après la Proposition 5.1 du chapitre 5 de la partie I, on a
$$\dim(V(\mathrm{Sep}(\mathcal{T})) \bigcap V(\mathcal{G})) < \dim(V(\mathcal{G})).$$

■

Ainsi, dans les hypothèses où la Proposition ci-dessus s'applique, seuls d déterminants doivent être calculés pour caractériser $\mathcal{D}(V(\mathcal{G},A))$. L'algorithme induit requiert ainsi une routine ayant plus de propriétés qu'une décomposition équi-dimensionnelle. En effet, il n'est pas toujours possible d'extraire un ensemble triangulaire $\mathcal{T}$ régulier et séparable d'une base de Gröbner lexicographique réduite $\mathcal{G}$ telle que $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$. Pour s'en convaincre il suffit de considérer l'exemple $\langle x,yz \rangle$ pour l'ordre $x < y < z$.

On note **LexTriSetEquiDim** une routine qui prend en entrée un système d'équations polynomiales S et qui retourne un ensemble de bases de Gröbner lexicographiques réduites $\mathcal{G}_1, \ldots, \mathcal{G}_m$ telles que pour tout $i \in \{1, \ldots, m\}$:
- $\mathcal{T}_i :=$ ExtractTriangular($\mathcal{G}_i$) est un ensemble triangulaire régulier et séparable;

- $\mathrm{sat}(\mathcal{T}_i) = \mathcal{G}_i$;
- $V(S) = V(\mathcal{G}_1) \bigcup \ldots \bigcup V(\mathcal{G}_m)$.

Remarque 2.1 *Une manière de concevoir une telle routine est d'implanter les algorithmes décrits dans [72, 5] et dont les sorties sont des ensembles triangulaires réguliers et séparables puis de calculer les saturés de ces ensembles triangulaires. Bien évidemment, ce n'est pas forcément la manière la plus judicieuse. Dans l'Annexe A, nous décrivons une implantation d'une telle routine qui n'utilise que des calculs de bases de Gröbner (ceux-ci étant réputés plus efficaces).*

Nous obtenons l'algorithme **SA2** (**S**ystèmes **A**lgorithme **2**).

Algorithme SA2

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \ldots, X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S) \bigcap R^n$.

1. list := LexTriSetEquiDim(S), result := [],
2. Choisir un point A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), et enlever S de list, poser $d = \mathrm{Dim}(S)$,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - $\mathcal{T}$ = ExtractTriangular(S).
     - (*) $Q = \Gamma_A(\mathcal{T}) \bigcup S$ et poser $u = \mathrm{Dim}(Q)$
     - Si $u = d$ choisir un autre point A et aller au pas (*).
     - list := list $\bigcup$ LexTriSetEquiDim(Q),
4. Retourner result.

Une fois les déterminants calculés, une étape d'élimination algébrique supplémentaire est nécessaire. Afin de rendre ces calculs plus faciles, on peut réduire modulo l'ensemble triangulaire les déterminants calculés à l'étape (*) de l'algorithme.

Notons $\mathrm{prem}(p,q,X)$ le pseudo-reste classique de deux polynômes p et q par rapport à la variable X. Si $p \in K[X_1, \ldots, X_n]$, sa forme réduite $\mathrm{prem}(p,\mathcal{T})$ peut être calculée par la procédure récursive suivante :

- si $\mathcal{T} = \emptyset$, alors $\mathrm{prem}(p,\mathcal{T}) = p$.
- sinon, si X_i est la plus grande variable apparaissant dans un polynôme $t \in \mathcal{T}$,

$$\mathrm{prem}(p,\mathcal{T}) = \mathrm{prem}(\mathrm{prem}(p,t,X_i),\mathcal{T} \setminus \{t\}).$$

En particulier, ceci implique qu'il existe des polynômes $q_{d+1}, \ldots q_n$ et des entiers positifs $i_{d+1}, \ldots, i_n$ tels que :

$$\mathrm{prem}(p,\mathcal{T}) = q_{d+1}t_{d+1} + \ldots + q_n t_n + h_{d+1}^{i_{d+1}} \ldots h_n^{i_n} p.$$

Ainsi, $V(\mathcal{G}) \bigcap V(\text{prem}(p,\mathcal{T})) = V(\mathcal{G}) \bigcap (V(p) \cup V(h_{d+1} \dots h_n))$. Par conséquent, on a :

$$\dim(V(\mathcal{G}) \bigcap V(p)) < \dim(V(\mathcal{G})) \Longrightarrow \dim(V(\mathcal{G}) \bigcap V(\text{prem}(p,\mathcal{T})) < \dim(V(\mathcal{G})).$$

2.2.2 L'apport d'une décomposition en premiers

Dans l'algorithme **SA1**, les points calculés dans $\mathcal{C}(V,A)$ ne sont pas tous les points critiques de la fonction distance mais peuvent aussi être des points singuliers de V et en particulier appartenir aux intersections des composantes irréductibles de V. Par exemple, si V est la réunion de deux sphères S_1 et S_2 telles que $S_1 \bigcap S_2$ est un cercle, la dimension de $\mathcal{C}(V,A)$ est 1 lorsque le point A est bien choisi. En revanche, $\mathcal{C}(S_1,A)$ et $\mathcal{C}(S_2,A)$ sont tous les deux des ensembles finis de points et pour un choix suffisamment générique du point A, on a $\mathcal{C}(S_1,A) \bigcup \mathcal{C}(S_2,A) \bigcup (S_1 \cap S_2) = \mathcal{C}(V,A)$. On constate sur cet exemple simple que l'usage de la décomposition en composantes irréductibles permet d'éviter un calcul inutile. De manière plus générale, l'usage de décompositions en idéaux premiers permet de ne pas avoir à considérer les points singuliers qui sont intersections de composantes irréductibles. Ainsi, des étapes de calcul inutiles sont évitées et la taille de la sortie de l'algorithme est réduite.
Le théorème suivant montre que la Proposition 2.1 s'applique aux décompositions en idéaux premiers.

Théorème 2.2 *([6, 5]) Soit $\mathcal{T} = (t_{d+1}, \dots, t_n) \subset \mathcal{G}$ un ensemble de polynômes tels que*

$$\forall (t_i,t_j) \in \mathcal{T} \times \mathcal{T} \ \text{mvar}(t_i) \neq \text{mvar}(t_j),$$

et $\forall g \in \mathcal{G}$, $\forall i \in \{d+1, \dots, n\}$ tels que $\text{mvar}(t_i) = \text{mvar}(g)$ ([5]) :

$$\deg(t_i,\text{mvar}(t_i)) \leq \deg(g,\text{mvar}(t_i)).$$

On note

- *h_i le coefficient dominant de t_i (lorsque t_i est considéré comme un polynôme univarié en sa variable principale) et $\mathcal{H}(\mathcal{T}) = \{h_{d+1}, \dots, h_n\}$.*
- *$W(\mathcal{T}) = \{M \in V(\mathcal{T}) \setminus V(\Pi_{i=d+1}^{n} h_i)\}$,*
- *$\text{sat}(\mathcal{T}) = \{p \in K[X_1, \dots, X_n] \mid \exists m \in N, \exists h \in \langle \mathcal{H}(\mathcal{T}) \rangle, h^m p \in \langle \mathcal{T} \rangle\}$.*

On a alors :

1. *$\mathcal{T}$ est un ensemble triangulaire régulier et séparable;*
2. *$\text{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$;*
3. *$\overline{W(\mathcal{T})} = V(\mathcal{G})$.*

Soit **LexPrimeDecomposition** une routine prenant en entrée un système d'équations polynomiales S dans $K[X_1, \dots, X_n]$ et retournant un système de générateurs de chaque idéal premier associé à $\sqrt{\langle S \rangle}$ sous la forme de base de Gröbner lexicographique réduite. On propose alors l'algorithme **SA3** (**S**ystèmes **A**lgorithme **3**) :

Algorithme SA3

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S)\bigcap R^n$.

1. list := LexPrimeDecomposition(S), result := [],
2. Choisir un point A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), et enlever S de list, poser $d = \mathrm{Dim}(S)$,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - $\mathcal{T} = \mathrm{ExtractTriangular}(S)$.
     - (*) $Q = \Gamma_A(\mathcal{T})\bigcup S$ et poser $u = \mathrm{Dim}(Q)$
     - Si $u = d$ choisir un autre point A et aller au pas (*).
     - list := list $\bigcup$ LexPrimeDecomposition(Q),
4. Retourner result.

2.3 Calculer avec les ensembles triangulaires

En collaboration avec P. Aubry et F. Rouillier, nous avons adapté nos algorithmes à l'usage des algorithmes de décomposition en ensembles triangulaires réguliers et séparables.

2.3.1 Problématique

Soit $\mathcal{G} \in K[X_1,\ldots,X_n]$ une base de Gröbner lexicographique réduite et $\mathcal{T} \subset \mathcal{G}$ un ensemble triangulaire régulier séparable vérifiant $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$. Dans l'algorithme **SA2**, on calcule des décompositions équi-dimensionnelles de $\mathcal{G}\bigcup\Gamma_A(\mathcal{T})$ (avec $A \in K^n$).

Comme nous le rappelons dans le chapitre 5, les ensembles triangulaires ne sont pas des systèmes de générateurs de leur saturé. Or, il n'existe pas d'algorithmes prenant en entrée un ensemble triangulaire $\mathcal{T}$, une famille de polynômes $S \subset K[X_1,\ldots,X_n]$ (équivalente à $\Gamma_A(\mathcal{T})$ dans notre cadre) et qui renvoie un décomposition en ensembles triangulaires (au sens de Lazard ou au sens de Kalkbrener) de $V(\mathrm{sat}(\mathcal{T}))\bigcap V(S)$ sans calculer un système de générateurs de $\mathrm{sat}(\mathcal{T})$. La routine **decompose** (rappelée dans le chapitre 5 de la partie I et décrite dans [72]) permet néanmois d'obtenir facilement un algorithme.
En effet, si $\dim(\overline{W(\mathcal{T})} \cap V(S)) < \dim(\overline{W(\mathcal{T})})$ et si

$$[\mathcal{T}_1,\ldots,\mathcal{T}_\ell] = \mathrm{decompose}(\mathcal{T},S),$$

alors

$$\forall i \in \{1,\ldots,\ell\} \quad \dim(\overline{W(\mathcal{T}_i)}) < \dim(\overline{W(\mathcal{T})}).$$

En reprenant les notations de la section 2.2 de ce chapitre, on peut donc utiliser l'opération **decompose** de manière à calculer une décomposition de $W(\mathcal{T})\bigcap V(\Gamma_A(\mathcal{T}))$. On note **Lazard** la routine qui prend en entrée un système d'équations polynomiales S et qui

en renvoie une décomposition en ensembles triangulaires réguliers séparables au sens de Lazard.
Nous obtenons alors l'algorithme **LTSA1** (**L**azard **T**riangulaire **S**ystèmes **A**lgorithme **1**), qui d'après la proposition 2.1, étant donné un système d'équations polynomiales S construit des ensembles triangulaires réguliers zéro-dimensionnels contenant au moins un point dans chaque composante semi-algébriquement connexe de $V(S) \bigcap R^n$:

Algorithme LTSA1

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \dots, X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S) \bigcap R^n$.

1. list := Lazard(S), result := [],
2. Choisir A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - $\mathcal{T}$:= first(list), et enlever $\mathcal{T}$ de list, poser $d = n - \sharp\mathcal{T}$,
   - Si $d = 0$ alors result := result $\bigcup \mathcal{T}$,
   - Sinon
     - Choisir un point A dans K^n,
     - (*) newlist := decompose($\mathcal{T}$,$\Gamma_A(\mathcal{T})$),
     - Si il existe $\mathcal{T}' \in$ newlist tel que $n - \sharp\mathcal{T}' = n - \sharp\mathcal{T}$, alors choisir un autre point A et retourner au pas (*),
     - list := list $\bigcup$ newlist,
4. Retourner result.

Les implantations dans le système de Calcul Formel Axiom de l'algorithme de décomposition en ensembles triangulaires de Lazard s'avèrent moins efficaces en pratique, que les implantations de l'algorithme de décomposition en ensembles triangulaires de Kalkbrener dans le système de Calcul Formel Axiom (voir [7]).

Dans le but d'aboutir à une utilisation plus efficace des ensembles triangulaires dans les calculs, nous avons essayé d'adapter nos algorithmes aux décompositions de Kalkbrener. On note **Kalkbrener** la routine qui prend en entrée un système d'équations polynomiales et renvoie en sortie une décomposition de Kalkbrener en ensembles triangulaires réguliers et séparables.

Premier problème : représentation des solutions

Le fait que toutes les solutions ne sont pas représentées par les décompositions au sens de Kalkbrener constitue une première difficulté.

Exemple 2.1 *Considérons l'ensemble triangulaire suivant :*

$$\mathcal{T} = \left\{ \begin{array}{l} xz - y^2 \\ y^2 + x^2 \end{array} \right.$$

La composante $\overline{W(\mathcal{T})}$ *est de dimension* 1, *et* $\overline{W(\mathcal{T})} \bigcap R^3$ *est un point isolé défini par les équations*

$$\begin{cases} z = 0 \\ y = 0 \\ x = 0 \end{cases}$$

Dans l'exemple ci-dessus, le lieu réel de la variété est inclus dans $\overline{W(\mathcal{T})} \setminus W(\mathcal{T})$. Ceci montre que dans un cadre de résolution non probabiliste et en l'état actuel des outils développés, les décompositions au sens de Lazard semblent être une sortie nécessaire. Il faudra donc montrer comment calculer de telles décompositions en utilisant exclusivement **Kalkbrener**.

Deuxième problème : atteindre les composantes zéro-dimensionnelles

Soit $\mathcal{T}$ un ensemble triangulaire régulier et séparable extrait d'une base de Gröbner lexicographique au cours de l'algorithme **SA2**. Au pas suivant de l'algorithme, le point A est tel que $\dim(\mathcal{C}(\overline{W(\mathcal{T})},A)) < \dim(\overline{W(\mathcal{T})})$.

De manière générale, étant donné une famille de polynômes S, telle que :

$$\dim(\overline{W(\mathcal{T})} \bigcap V(S)) < \dim(\overline{W(\mathcal{T})})$$

il peut arriver que :

$$\dim(V(\mathcal{T}) \bigcap V(S)) \geq \dim(\overline{W(\mathcal{T})}).$$

Exemple 2.2 *Considérons le système d'équations polynomiales* S *suivant :*

$$\begin{cases} xz - y \\ y - x \end{cases}$$

Une décomposition au sens de Kalkbrener renvoie deux ensembles triangulaires

$$\mathcal{T}_1 = \begin{cases} xz - y \\ y - x \end{cases} \qquad \mathcal{T}_2 = \begin{cases} y \\ x \end{cases}$$

(dont les saturés sont $\mathrm{sat}(\mathcal{T}_1) = \langle z - 1, y - x \rangle$ *et* $\mathrm{sat}(\mathcal{T}_2) = \langle y, x \rangle$*). Puisque* $\mathcal{T}_1$ *est un ensemble triangulaire régulier, on a :*

$$\dim(\overline{W(\mathcal{T}_1)} \bigcap V(x)) < \dim(\overline{W(\mathcal{T}_1)})$$

Or, il est aisé de constater que :

$$\dim(V(\mathcal{T}_1) \bigcap V(x)) = \dim(V(S)) = \dim(\overline{W(\mathcal{T}_1)})$$

Ceci montre qu'on ne pourra pas calculer simplement $\mathcal{D}(\overline{W(\mathcal{T})},A)$ en utilisant exclusivement la routine **Kalkbrener**.

2.3.2 L'algorithme

Décompositions au sens de Lazard

Le lemme suivant induit l'algorithme que nous cherchons :

Lemme 2.1 *Soit S un système d'équations polynomiales et $\mathcal{T} \in$ **Kalkbrener**(S). Alors, on a :*

$$\forall h \in \mathcal{H}(\mathcal{T}) \quad \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} \subset \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle + \langle h \rangle}.$$

et l'inclusion est stricte.

Preuve : On a :
- $\langle S \rangle \subset \sqrt{\langle S \rangle} \subset \mathrm{sat}(\mathcal{T})$ (car $\mathcal{T} \in$ **Kalkbrener**(S)),
- $\langle \mathcal{T} \rangle \subset \mathrm{sat}(\mathcal{T})$,

Donc $\langle S \rangle + \langle \mathcal{T} \rangle \subset \mathrm{sat}(\mathcal{T})$, et comme $\mathrm{sat}(\mathcal{T}) = \sqrt{\mathrm{sat}(\mathcal{T})}$, on a :

$$\sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} \subset \mathrm{sat}(\mathcal{T}).$$

Puisque $\mathcal{T}$ est un ensemble triangulaire régulier et séparable, h n'appartient à aucun des idéaux premiers associés à $\mathrm{sat}(\mathcal{T})$, donc

$$h \notin \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle}$$

ce qui implique que

$$\sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} \subset \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} + \langle h \rangle$$

(où l'inclusion est stricte). Comme $\sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} + \langle h \rangle \subset \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle + \langle h \rangle}$, on en déduit que

$$\sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} \subset \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle + \langle h \rangle}$$

où l'inclusion est stricte. ■

Il est alors clair que l'algorithme **LDK** (**L**azard **D**ecomposition via **K**alkbrener) ci-dessous construit des suites strictement croissantes d'idéaux. Donc cet algorithme termine et la correction de sa sortie est évidente.

Algorithme LDK

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \ldots, X_n]$.
- **Sortie :** Une décomposition de Lazard en ensembles triangulaires réguliers séparables de $V(S)$.

1. list := Kalkbrener(S), et result := list,
2. tant que list $\neq$ [] faire
   - newlist := [],
   - pour tout $\mathcal{T}$ dans list faire newlist := newlist $\cup$ $[\bigcup_{h \in \mathcal{H}(\mathcal{T})}$ Kalkbrener$(\mathcal{T} \cup \{h\} \cup \{S\})]$,
   - list := newlist et result := result $\bigcup$ list.
3. Retourner result.

On note $\mathcal{S}(\mathcal{T})$ l'ensemble des séparants d'un ensemble triangulaire $\mathcal{T}$. Nous avons mis en évidence dans la section précédente l'intérêt de pouvoir calculer $\overline{W(\mathcal{T})} \bigcap V(s)$ (pour $s \in \mathcal{S}(\mathcal{T})$). Le lemme suivant se démontre de la même manière que le lemme 2.1.

Lemme 2.2 *Soit S un système d'équations polynomiales et $\mathcal{T} \in$ **Kalkbrener**(S). Alors, on a :*

$$\forall s \in \mathcal{S}(\mathcal{T}) \quad \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle} \subset \sqrt{\langle S \rangle + \langle \mathcal{T} \rangle + \langle s \rangle}.$$

et l'inclusion est stricte.

Remarque 2.2 *Il suffit que la routine* **Kalkbrener** *renvoie des ensembles triangulaires réguliers pour que lemme 2.1 soit vérifié. En revanche, l'hypothèse de séparabilité est indispensable dans le lemme 2.2 ci-dessus.*

Notons que sous les hypothèses du lemme ci-dessus, si s_i est un séparant de $\mathcal{T}$, l'opération Kalkbrener$(S \cup \mathcal{T} \cup \{s_i\})$ renvoie une liste d'ensembles triangulaires contenant celle que Kalkbrener$(S \cup \mathcal{T} \cup \{h_i\})$ aurait renvoyé car h_i est un facteur du résultant associé au couple (s_i,t_i).

L'algorithme **SLDK** (**S**eparants **L**azard **D**ecomposition **K**alkbrener) décrit ci-dessous prend en entrée un système d'équations polynomiales S et renvoie une famille d'ensembles triangulaires réguliers séparables $\mathcal{T}_1, \ldots ,\mathcal{T}_\ell$ telle que :

$$V(S) = (W(\mathcal{T}_1) \setminus V(\mathrm{Sep}(\mathcal{T}_1))) \bigcup \ldots \bigcup (W(\mathcal{T}_\ell) \setminus V(\mathrm{Sep}(\mathcal{T}_\ell)))\,.$$

Algorithme SLDK

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \ldots ,X_n]$.
- **Sortie :** une famille d'ensembles triangulaires réguliers séparables $\mathcal{T}_1, \ldots ,\mathcal{T}_\ell$ telle que :

$$V(S) = (W(\mathcal{T}_1) \setminus V(\mathrm{Sep}(\mathcal{T}_1))) \bigcup \ldots \bigcup (W(\mathcal{T}_\ell) \setminus V(\mathrm{Sep}(\mathcal{T}_\ell)))\,.$$

1. list := LDK(S), et result := list,
2. tant que list $\neq$ [] faire
   - newlist := [],
   - pour tout $\mathcal{T}$ dans list faire newlist := newlist $\cup$ $[\bigcup_{s \in \mathcal{S}(\mathcal{T})} \mathrm{LDK}(\mathcal{T} \cup \{s\} \cup \{S\})]$,
   - list := newlist et result := result $\bigcup$ list.
3. Retourner result.

Atteindre les composantes zéro-dimensionnelles

Soit $S \subset K[X_1, \ldots ,X_n]$ un système d'équations polynomiales, $\mathcal{T} \in$ LDK(S) un ensemble triangulaire régulier séparable, et A un point de C^n. On suppose que A est choisi tel que $\dim(\mathcal{D}(\overline{W(\mathcal{T})},A)) < \dim(\overline{W(\mathcal{T})})$. Dans la suite nous allons montrer comment calculer un ensemble fini de points contenant la composante zéro-dimensionnelle de $\mathcal{D}(\overline{W(\mathcal{T})},A)$.

Pour cela, on utilise la routine **QuasiKalkbrener** (voir chapitre 5 de la partie I).

Lemme 2.3 *Soit* $[\mathcal{T}_1,\ldots,\mathcal{T}_\ell] = \mathrm{QuasiKalkbrener}([\mathcal{T}\cup\Gamma_A(\mathcal{T})],[\mathcal{H}(\mathcal{T})\bigcup\mathcal{S}(\mathcal{T})])$. *Alors, la variété*

$$\overline{W(\mathcal{T}_1)}\bigcup\ldots\bigcup\overline{W(\mathcal{T}_\ell)}$$

est un ensemble fini de points contenant la composante de dimension zéro de $\mathcal{D}(\overline{W(\mathcal{T})},A)$.

Preuve : D'après les spécifications de la routine **QuasiKalkbrener**, on a :

$$\overline{W(\mathcal{T}_1)}\bigcup\ldots\bigcup\overline{W(\mathcal{T}_\ell)} = \overline{\Big(W(\mathcal{T})\bigcap V(\Gamma_A(\mathcal{T}))\Big)\setminus V(\mathrm{Sep}(\mathcal{T}))}.$$

La proposition 2.1 permet alors de conclure. ∎

Le lemme ci-dessus permet en un sens d'étudier $(W(\mathcal{T})\bigcap V(\Gamma_A(\mathcal{T})))\setminus V(\mathrm{Sep}(\mathcal{T}))$. L'étude des points de $\overline{W(\mathcal{T})}\bigcap\left(\bigcup_{h\in\mathcal{H}(\mathcal{T})}V(h)\cup V(\mathrm{Sep}(\mathcal{T}))\right)$ est assurée puisque ces points sont représentés par un autre ensemble triangulaire dans SLDK(S). On obtient alors l'algorithme **KTSA1** (**K**albrener **T**riangulaire **S**ystèmes **A**lgorithme **1**).

Algorithme KTSA1

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S)\bigcap R^n$.

1. list := SLDK(S), result := [],
2. Tant que list $\neq$ [] faire
   - $\mathcal{T} := first$(list) et enlever $\mathcal{T}$ de list,
   - Si $\sharp\mathcal{T} = n$ alors result := result $\bigcup\mathcal{T}$,
   - sinon choisir un point A dans K^n
   - (*) newlist := QuasiKalkbrener($[\mathcal{T}\bigcup\Gamma_A(\mathcal{T})],[\mathcal{H}(\mathcal{T})\bigcup\mathcal{S}(\mathcal{T})]$),
   - Si il existe $\mathcal{T}'\in$ newlist tel que $\sharp\mathcal{T}'\neq n$ alors choisir un autre point A et retourner au pas (*).
   - result := result $\bigcup$ newlist.
3. Retourner result.

Remarque 2.3

- *Nous avons montré comment calculer des décompositions en ensembles triangulaires au sens de Lazard en n'utilisant que des décompositions au sens de Kalkbrener. Cette méthode souffre de la redondance de certains calculs, même si sur de nombreux exemples elle s'est avérée avantageuse. Par ailleurs, dans la description de l'algorithme* **LDK** *nous n'avons pas fait état des heuristiques utilisées pour éviter ces calculs inutiles.*
- *Enfin, nous avons mis en évidence l'intérêt de l'algorithme* **SLDK** *dont les spécifications sont un peu particulières. Les choix algorithmiques que nous avons effectués, en privilégiant l'usage de* **Kalkbrener** *comme routine de base n'ont pas été comparés avec d'autres stratégies (voir [33] par exemple).*

2.4 Quelques cas particuliers importants

Dans cette section, nous étudions comment adapter les algorithmes **SA2** et **SA3** pour traiter plus efficacement deux cas particuliers :

- le cas de la position générale;
- et le cas des variétés algébriques réelles compactes.

2.4.1 La position générale

Le cas de la position générale est important car il est considéré comme étant un cas générique souvent rencontré «en pratique». Soit V une variété équi-dimensionnelle de dimension d en position générale. Algorithmiquement, le cas de la position générale peut être détecté quand la base de Gröbner $\mathcal{G}$ lexicographique réduite définissant l'idéal radical associé à V est un ensemble triangulaire de la forme :

$$\mathcal{G}\begin{cases} X_n + q_n(X_1,\ldots,X_{d+1}) \\ \vdots \\ X_{d+2} + q_{d+2}(X_1,\ldots,X_{d+1}) \\ p(X_1,\ldots,X_{d+1}) \end{cases}$$

Par abus de langage, on dira que $\mathcal{G}$ est en position générale. Ainsi, il est clair que donner au moins un point par composante semi-algébriquement connexe de $V(\mathcal{G})\bigcap R^n$ revient à donner au moins un point par composante semi-algébriquement connexe de $V(p)\bigcap R^{d+1}$. Pour cela, l'algorithme **SA1** peut être utilisé. Nous obtenons l'algorithme **SLSA** (**S**hape **L**emma **S**ystème **A**lgorithme). Cet algorithme utilise la routine **LexEquiDim** qui prend en entrée un système d'équations polynomiales S et retourne une liste de bases de Gröbner lexicographiques réduites $\mathcal{G}_d,\ldots,\mathcal{G}_0$ engendrant des idéaux radicaux et équi-dimensionnels, tels que $V(S) = V(\mathcal{G}_d)\bigcup\ldots\bigcup V(\mathcal{G}_0)$.

Algorithme SLSA

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \ldots, X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S) \bigcap R^n$.

1. list := LexEquiDim(S), result := [],
2. Choisir un point A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), poser d := Dim(S) et enlever S de list,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Si S est en position générale, alors faire result := result $\bigcup$ (SA1(last(p)) $\cup$ S).
   - Sinon
     - (*) Q = Minors(S,d,A) $\bigcup S$
     - u = Dim(Q)
     - Si $u = d$ choisir un autre point A et aller au pas (*).
     - list := list $\bigcup$ LexEquiDim(Q)
4. Retourner result.

2.4.2 Variétés algébriques réelles compactes

Dans ce paragraphe, nous montrons comment dans le cas des variétés algébriques réelles compactes, les fonctions de projection sur un axe peuvent être utilisées dans des algorithmes inspirés de ceux présentés ci-dessus. En particulier, nous donnons un algorithme qui trouve au moins un point réel sur chaque composante semi-algébriquement connexe compacte d'une variété algébrique réelle. Insistons sur le fait que cet algorithme peut retourner des points appartenant à des composantes semi-algébriquement connexes et non compactes. Notons que des travaux similaires sont exposés dans [11].

Un premier algorithme

Soit $\overrightarrow{U} = (u_1, \ldots, u_n) \in K^n$ un vecteur non nul et Π_U l'application régulière

$$\begin{array}{cccc} \Pi_U : & C^n & \longrightarrow & C \\ & M = (x_1, \ldots, x_n) & \longmapsto & u_1x_1 + \ldots + u_nx_n \end{array}$$

Théorème 2.3 *Soit V une variété algébrique équi-dimensionnelle de dimension d et $S = \{P_1, \ldots, P_s\}$ des polynômes de $K[X_1, \ldots, X_n]$ tels que $I(V) = \langle P_1, \ldots, P_s \rangle$. Etant donné un vecteur $\overrightarrow{U} \in K^n$, on définit l'ensemble algébrique :*

$$\mathcal{C}(V, \overrightarrow{U}) = \{M \in V, \mathrm{rank}(\overrightarrow{grad}_M\ (P_1), \ldots, \overrightarrow{grad}_M\ (P_s), \overrightarrow{U}) \leq n - d\}.$$

Il existe D un entier positif suffisamment grand, tel qu'il existe au moins un vecteur $\overrightarrow{U}$ dans $\{1 \ldots D\}^n$ vérifiant :

1. *$\mathcal{C}(V, \overrightarrow{U})$ s'intersecte avec chaque composante* **compacte** *et semi-algébriquement connexe de $V \bigcap R^n$,*

2. $\mathcal{C}(V, \overrightarrow{U}) = \mathrm{Sing}(V) \bigcup V_0$ *où* V_0 *est un ensemble fini de points.*

De plus, $\dim(\mathcal{C}(V, \overrightarrow{U})) < \dim(V)$.

La démonstration de ce théorème est similaire à celle du théorème 2.1.

Pour $\overrightarrow{U} \in C^n$, $\mathcal{Q} = \{Q_1, \ldots, Q_s\} \subset K[X_1, \ldots, X_n]^s$, et $d \in \mathbb{N}$, $0 \leq d < n$, on définit $\Delta_{\overrightarrow{U},d}(\mathcal{Q})$ comme étant l'ensemble de tous les mineurs d'ordre $(n-d+1, n-d+1)$ de la matrice

$$\left[\left[\frac{\partial Q_i}{\partial X_j} \right]_{(i=1,\ldots,n,j=1,\ldots,s)} \middle| \overrightarrow{U} \right]$$

On déduit du théorème ci-dessus un premier algorithme **CSA1** (**C**ompact **S**ystèmes **A**lgorithme **1**) qui donne au moins un point par composante compacte et semi-algébriquement connexe sur une variété algébrique réelle, qui utilise les mêmes routines que celles de l'algorithme **SA1**.

Algorithme CSA1

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1, \ldots, X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe et compacte de $V(S) \bigcap R^n$.

1. list := EquiDim(S), result := [],
2. Choisir un vecteur $\overrightarrow{U}$ dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), poser d := Dim(S) et enlever S de list,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - (*) $Q = \Delta_{\overrightarrow{U},d} \bigcup S$
     - $u = \mathrm{Dim}(Q)$
     - Si $u = d$ choisir un autre vecteur $\overrightarrow{U}$ et aller au pas (*).
     - list := list $\bigcup$ EquiDim(Q)
4. Retourner result.

Un deuxième algorithme

Le fait que nous ayons considéré dans la section précédente les familles de fonctions Π_U est pleinement justifié. En effet, étant donnée une variété algébrique réelle, il n'existe pas toujours une variable telle que le lieu des points critiques de la fonction de projection sur l'axe de coordonnée de cette variable est de dimension strictement inférieure à celle de la variété. Pour s'en convaincre, il suffit de considérer la variété de R^3 qui est la réunion de trois cercles dessinés autour de chacun des axes de coordonnée. Dans cette section, on montre que si la routine de décomposition équi-dimensionnelle est **LexTriSetEquiDim**, pour $\mathcal{G} \in$ LexTriSetEquiDim, il existera *toujours* une variable X telle que $\mathcal{C}(V(\mathcal{G}), \overrightarrow{e_X})$ soit de dimension strictement inférieure à celle de V.

Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite pour l'ordre $X_1 < \ldots < X_n$ engendrant un idéal radical et équi-dimensionnel de dimension d, tel que la routine **ExtractTriangular** définie Section 2.2.1 de ce chapitre en extrait un ensemble triangulaire $\mathcal{T} = (t_{d+1}, \ldots, t_n)$ vérifiant :

- $\mathcal{T}$ est un ensemble triangulaire régulier,
- $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.

Sans nuire à la généralité, on suppose que $\mathrm{mvar}(t_i) = X_i$ (dans le cas contraire, on peut renuméroter les variables). Pour toute variable X_i, on note $\overrightarrow{e_i}$ le vecteur dont toutes les coordonnées sont nulles sauf la i-ième qui est égale à 1. Nous allons montrer ci-dessous que pour tout $i \in \{1, \ldots, d\}$, la fonction de projection sur le vecteur $\overrightarrow{e_i}$ convient.

Notation 2.4 *Soit* $\overrightarrow{U} = (u_1, \ldots, u_n)$ *un vecteuur de* K^n, $d = \dim(V(\mathcal{G}))$, *et considérons pour* $j = 1, \ldots, d$ *la liste des mineurs d'ordre* $(n-d+1)$ *extraite de* $\Delta_{\overrightarrow{U},d}(\mathcal{T})$ *:*

$$\Gamma_{\overrightarrow{U}}(\mathcal{T}) = \{\Gamma_A^{(j)}(\mathcal{T}) = det(\mathcal{M}_{\overrightarrow{U}}^{(j)}), j = 1, \ldots, d\}$$

où

$$\mathcal{M}_{\overrightarrow{U}}^{(j)} = \left[\begin{array}{c|c} \left[\frac{\partial t_i}{\partial X_j}\right]_{i=d+1,\ldots,n} & u_j \\ \hline \mathcal{U}_{\mathcal{T}} = \left[\frac{\partial t_i}{\partial X_j}\right]_{j=d+1,\ldots,n}^{i=d+1,\ldots,n} & \begin{array}{c} u_{d+1} \\ \vdots \\ u_n \end{array} \end{array}\right]$$

On note $\mathcal{D}(V(\mathcal{G}), \overrightarrow{U}) = V(\mathcal{G}) \cap V(\Gamma_{\overrightarrow{U}}(\mathcal{T}))$.

Proposition 2.2 *Pour tout* $i \in \{1, \ldots, d\}$, *on a :*

- $\mathcal{C}(V(\mathcal{G}), \overrightarrow{e_i}) \subset \mathcal{D}(V(\mathcal{G}), \overrightarrow{e_i})$,
- $\dim(\mathcal{D}(V(\mathcal{G}), \overrightarrow{e_i})) < \dim(V(\mathcal{G}))$.

Preuve : Soit $i \in \{1, \ldots, n\}$ et considérons la matrice

$$\mathcal{M}_{\Pi_i}^{(j)} = \left[\begin{array}{c|c} \begin{array}{c} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{array} & \left[\frac{\partial t_i}{\partial X_j}\right]_{i=d+1\ldots n} \\ \hline \begin{array}{c} 0 \\ \vdots \\ \vdots \\ 0 \end{array} & \mathcal{U}_{\mathcal{T}} = \left[\frac{\partial t_i}{\partial X_j}\right]_{j=d+1,\ldots,n}^{i=d+1\ldots n} \end{array}\right]$$

Alors, si on enlève les $i-1$-ième premières lignes ainsi que celles allant de la $i+1$-ième à la d-ième, on obtient une matrice carrée triangulaire supérieure dont la diagonale contient les éléments de $\mathrm{Sep}(\mathcal{T})$. Le produit de ces éléments est donc un des d déterminants à calculer. Or, d'après le théorème 5.3, si on note s_j un élément quelconque de $\mathrm{Sep}(\mathcal{T})$, on a :

$$\dim(V(\mathcal{G})\bigcap V(s_i)) < \dim(V(\mathcal{G})).$$

Comme on a $\mathcal{C}(V(\mathcal{G}),\ \overrightarrow{e_i}) \subset \bigcup_{j=d+1}^{n} V(\mathcal{G})\bigcap V(s_j)$, le lemme est démontré. ∎

Remarque 2.4 *D'un point de vue calculatoire, la preuve du résultat ci-dessus est importante. En effet, tout calcul de déterminant est supprimé : il suffit d'intersecter $V(\mathcal{G})$ avec les séparants de l'ensemble triangulaire extrait de $\mathcal{G}$.*

On en déduit l'algorithme **CSA2** (**C**ompact **S**ystèmes **A**lgorithme **2**) qui utilise les mêmes routines que celles utilisées dans **Algorithme SA2**.

Algorithme CSA2

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** Une liste de systèmes zéro-dimensionnels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe et compacte de $V(S)\bigcap R^n$.

1. list := LexTriSetEquiDim(S), result := [],
2. Tant que list $\neq \emptyset$ faire
   - S := first(list), et enlever S de list, poser $d = \mathrm{Dim}(S)$,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - $\mathcal{T} = \mathrm{ExtractTriangular}(S)$.
     - $Q = \mathrm{Sep}(\mathcal{T})\bigcup S$ et poser $u = \mathrm{Dim}(Q)$
     - list := list $\bigcup$ LexPrimeDecomposition(Q),
3. Retourner result.

2.5 Validation expérimentale

2.5.1 Méthodologie, algorithmes de base et logiciels

Dans le but de tester les algorithmes **SA1** et **SA2** nous avons implanté un algorithme calculant une décomposition équi-dimensionnelle radicale de systèmes polynomiaux. L'algorithme prend en entrée une base de Gröbner lexicographique et s'inspire des méthodes de scindage présentées dans [72, 5] pour renvoyer une famille de bases de Gröbner lexicographiques dont on peut extraire un ensemble triangulaire régulier et séparable tel que son saturé est l'idéal engendré par la base dont il est extrait. Cet algorithme est décrit dans l'Annexe A du document. Les logiciels utilisés sont Gb pour le calcul de bases de Gröbner lexicographiques et Maple. Même si notre implantation de cet algorithme offre de meilleures performances que les implantations d'algorithmes ayant les mêmes spécifications et qui sont disponibles dans le logiciel Magma par exemple, notre méthode n'est pas optimale : nous avons vu dans les sections précédentes que les systèmes engendrés en cours d'algorithmes contiennent des composantes de différentes dimension, ce qui rend difficile le premier calcul de base de Gröbner.

De plus, il est clair que tout progrès significatif dans le calcul de telles décompositions induit un progrès équivalent pour nos algorithmes. Le logiciel F7 [1] de J.-C. Faugère – en cours d'implantation – permettra d'améliorer significativement les performances de nos algorithmes, tant du point de vue de la taille de la sortie que du point de vue des temps de calcul. Enfin, nous n'avons pas implanté de décomposition en idéaux premiers. Nous ne testerons donc pas l'algorithme **SA3**, considérant comme acquis que la somme des degrés des systèmes zéro-dimensionnels retournés par cet algorithme sera toujours inférieure ou égal à la somme des degrés des systèmes zéro-dimensionnels retournés par les algorithmes **SA2** et **SA1**.

La résolution des systèmes zéro-dimensionnels se fait en calculant des Représentations Univariées Rationnelles. Pour cela, nous utilisons le logiciel RS (implanté en C par F. Rouillier). Le comptage et l'isolation des racines réelles se fait alors avec l'algorithme d'Uspensky amélioré (voir [86]) implanté dans RS. Les logiciels Gb et RS sont liés à Maple par une connexion fichiers.

Les calculs de décomposition cylindrique algébrique ont été effectués avec le logiciel **QEPCAD** qui est implanté en C par H. Hong *et al.* (voir [29]). Ce logiciel est à notre connaissance l'un des plus rapides pour le calcul de décompositions cylindriques algébriques.

Les implantations de l'algorithme **KTSA1** sont basées sur les implantations de P. Aubry (voir [5]) de l'algorithme de décomposition en ensembles triangulaires réguliers séparables au sens de Kalkbrener.
Tous ces calculs ont été effectués sur un PC Pentium II 400 MHz avec 512 Mo de RAM de l'UMS MEDICIS [3]. Les temps de calculs sont donnés en secondes.
Les calculs ont systématiquement été stoppés au bout de 12 heures de calculs. Aussi, le symbole ∞ dans les tableaux ci-dessous signifie en réalité *arrêté après 12 heures de calcul.*

2.5.2 Déformer ou ne pas déformer ?

Dans un premier temps, dans le cas des hypersurfaces, on compare la stratégie proposée dans ce Chapitre à la stratégie de déformation infinitésimale étudiée dans le Chapitre précédent.

Dans le tableau 2.1, nous donnons dans la colonne **Algorithme HA3** les degrés des systèmes zéro-dimensionnels étudiés et les temps de calcul de l'algorithme décrit dans le chapitre précédent. Dans la colonne **Algorithme SA2**, nous donnons la somme des degrés des systèmes zéro-dimensionnels étudiés et les temps de calcul de l'algorithme **SA2** décrit dans ce chapitre.

On constate que la taille des données intermédiaires (degré du système zéro-dimensionnel à coefficients infinitésimaux) apparaissant dans l'algorithme **HA3** est bien plus grande sur nos exemples que celles (degré des systèmes zéro-dimensionnels) de l'algorithme **SA2** décrit dans ce chapitre. Cette amélioration du contrôle de la taille des données intermédiaires a un impact sensible sur les temps de calcul, puisque des problèmes dont la résolution nécessite plusieurs heures de calcul avec l'algorithme **HA3** sont résolus par l'algorithme

Hypersurface	Algorithme HA3		Algorithme SA2	
Birk.3-1	16	3,2	12	0,08
Birk.3-2	16	3,1	7	0,13
Birk.3-3	34	20,6	25	0,37
Birk.3-4	36	24,9	16	0,18
Birk.3-5	40	56,2	31	0,46
Birk.3-6	52	50,2	37	0,86
Birk.3-7	52	59,5	38	0,72
Birk.3-8	130	962,8	45	7,11
Birk.3-9	132	1072,7	47	7,88
Birk.3-10	136	6474,6	48	8,04
Birk.3-11	138	9424,6	50	8,88
Birk.3-12	138	6033,5	50	10,01
Birk.3-13	252	∞	32	9,26
Birk.3-14	264	∞	60	66,75
Birk.3-15	272	∞	60	82,68

TAB. 2.1 – *Algorithmes HA3 et SA2 : comparaison des sorties et des temps de calcul*

SA2 dans des temps de calcul de l'ordre de la dizaine de secondes. Le phénomène est encore plus visible sur les 3 dernières hypersurfaces (Birk.3-13 à Birk.3-15) dont la résolution n'est pas accessible à l'algorithme **HA3** mais que l'algorithme **SA2** résoud facilement.

Dans le tableau 2.2, nous comparons le nombre de cellules ainsi que les temps de calcul d'une Décomposition Cylindrique Algébrique (colonne **CAD**) avec la somme des degrés des idéaux zéro-dimensionnels ainsi que les temps de calcul de l'algorithme **SA2** sur les exemples considérés ci-dessus.

On constate que la sortie de l'algorithme de Décomposition Cylindrique Algébrique reste de taille inférieure à celle de l'algorithme **SA2**. Il en est de même des temps de calcul, mais les différences ne sont plus aussi sensibles que ce que nous avions constaté en comparant l'algorithme **HA3** et la Décomposition Cylindrique Algébrique sur les mêmes exemples (voir chapitre précédent). Nous devons pousser nos comparatifs sur des exemples plus significatifs : les méthodes de points critiques sont conçues pour traiter des problèmes de plus de trois variables.

2.5.3 Algorithme SA1 / Algorithme SA2

Nous comparons les algorithmes **SA1** et **SA2**. Dans le tableau 2.3, nous donnons les temps de calcul respectivement obtenus par ces algorithmes. Dans la première colonne, on donnons le temps passé dans l'étape de décomposition des systèmes polynomiaux. Dans la seconde colonne, on donne le temps passé dans le calcul des déterminants. Le signe ? est introduit lorsque l'étape précédente (calculs de déterminants ou décomposition de systèmes polynomiaux) n'a pas terminé.

Sachant que pour les exemples *Vermeer*, *Wang, Euler, Neural, Butcher, Buchberger*, et *DiscPb* les bases de Gröbner manipulées sont des ensembles triangulaires, il est normal

Hypersurface	CAD		Algorithme SA2	
Birk.3-1	13	0,21	12	0,08
Birk.3-2	12	0,11	7	0,13
Birk.3-3	12	4,44	25	0,37
Birk.3-4	9	3,45	16	0,18
Birk.3-5	12	11,1	31	0,46
Birk.3-6	12	15,7	37	0,86
Birk.3-7	12	18,2	38	0,72
Birk.3-8	10	0,2	45	7,11
Birk.3-9	10	0,15	47	7,88
Birk.3-10	21	1,86	48	8,04
Birk.3-11	21	1,26	50	8,88
Birk.3-12	21	1,88	50	10,01
Birk.3-13	10	0,74	32	9,26
Birk.3-14	14	1,12	60	66,75
Birk.3-15	41	2,1	60	82,68

TAB. 2.2 – *Algorithmes SA2 et CAD : comparaison des sorties et des temps de calcul*

System	Dimension/Degree	Nb Vars	Algorithme SA1		Algorithme SA2	
Vermeer	1,26	5	0.01	0	0.01	0
Wang	1,114	13	0.12	0	0.12	0
Euler	3,2	10	0.01	0	0.01	0
Neural	1,24	4	0.43	0	0.43	0
Butcher	3,3	8	1.7	0	1.7	0
Buchberger	4,6	8	0	0	0	0
DiscPb	2,3	4	0.02	0	0.02	0
Donati	1,10	4	0.04	26	0.04	0
Hairer2	2,25	13	23.03	∞	23.03	0
Prodecco	2,2	5	284	26	284	0
F633	2,32	10	?	∞	∞	?
F744	1,40	12	24.06	∞	24.06	0.02
F855	1,52	14	5654	∞	5654	173

TAB. 2.3 – *Algorithmes SA1 et SA2 : comparaison des temps de calcul*

System	Algorithme SA2 + ZDS	QEPCAD
Vermeer	84	65976
Wang	132	∞
Euler	10	failed(872043)
Neural	133	205
Butcher	15	∞
Buchberger	32	failed(991324)
DiscPb	28	∞
Donati	61	10

TAB. 2.4 – *Algorithmes SA2 et CAD : comparaison de la taille des sorties*

que les temps de calcul des algorithmes **SA1** et **SA2** coincident. Pour les autres exemples, plus difficiles, les bases de Gröbner manipulées ne sont plus des ensembles triangulaires. Il apparait alors clairement que les optimisations dont bénéficie l'algorithme **SA2** permettent d'obtenir des temps de calculs significativement meilleurs que ceux obtenus par l'algorithme **SA1**. Pour certains exemples, le calcul de tous les déterminants est même une étape bloquante de l'algorithme (*F633, F744, F855*).

2.5.4 Algorithme SA2 / CAD

Taille de la sortie

On note **ZDS** la routine de résolution réelle des systèmes zéro-dimensionnels produits par l'algorithme **SA2**, c'est-à-dire que nous incluons les temps de comptages et d'isolation des racines réelles des premiers polynômes des Représentations Univariées Rationnelles calculées. Ces Représentations Univariées Rationnelles sont calculées avec les logiciels AGb (écrit en C++ par J.-C. Faugère et calculant des bases de Gröbner pour l'ordre DRL) et le logiciel RS (écrit en C par F. Rouillier et calculant les RUR ainsi que le nombre de racines réelles de leur premier polynôme). Dans le tableau 2.4, on donne le nombre de points calculés par l'algorithme **SA2** (c'est-à-dire la somme des degrés des systèmes zéro-dimensionnels) et par **QEPCAD** sur les exemples pour lesquels au moins une de ces méthodes termine. Quand **QEPCAD** est stoppé après 12 heures d'attente, le signe ∞ est inscrit dans le tableau. Si le calcul échoue à cause d'un nombre trop important de cellules, on inscrit failed(n), où n est la borne inférieure du nombre de cellules prédit par **QEPCAD**.

Ces résultats sont cohérents avec les prédictions «théoriques» : la sortie de l'algorithme de Décomposition Cylindrique Algébrique est supérieure en taille à celle de l'algorithme **SA2**.
On peut aussi noter qu'aucune de ces implantations ne résoud les exemples *Hairer2*, *Prodecco*, *F633*, *F744* et *F855*, même si l'algorithme **SA2** a réussi à calculer les systèmes zéro-dimensionnels. Ces systèmes sont trop difficiles pour le calcul d'une base de Gröbner avec AGb. Il est néanmoins important de faire les remarques suivantes :

- notre implantation, basée sur Gb, est loin d'être optimale puisque les scindages sont effectués après le calcul d'une base de Gröbner lexicographique. Or, d'après les

System	Algorithme SA1 + ZDS	Algorithme SA2 + ZDS	QEPCAD
Vermeer	62.36	3.32	43
Wang	1.37	1.37	∞
Euler	0.01	0.01	failed(*872043*)
Neural	1.02	1.02	0.9
Butcher	1.7	1.7	∞
Buchberger	< 0.01	< 0.01	failed(*991324*)
DiscPb	0.2	0.2	∞
Donati	11609	10	0.6

TAB. 2.5 – *Algorithmes SA1, SA2 et CAD : comparaison des temps de calcul*

résultats de la section 2.2 de ce chapitre, il est clair que les systèmes obtenus après calcul de déterminants sont scindables et que donc le calcul de bases lexicographiques de tels systèmes est difficile. Dès lors, il apparait évident que des techniques de scindages mieux conçues permettront résoudre ces problèmes.

– J.-C. Faugère a récemment effectué des progrès significatifs dans le calcul de bases de Gr'öbner en produisant des algorithmes dont les implantations sont plus efficaces de plusieurs ordres de grandeur que toutes ses conccurentes. L'utilisation de telles implantations dans nos algorithmes permettra sans nul doute d'accroitre le nombre de problèmes pouvant être traités.

Temps de calcul

Le tableau 2.5 montre que les deux algorithmes **SA1 + ZDS** et **SA2 + ZDS** ont un meilleur comportement en pratique que **QEPCAD**. En effet, sur 8 exemples, le logiciel **QEPCAD** n'en résoud que 3 alors que les algorithmes **SA1** et **SA2** les résolvent tous. Par ailleurs, on peut constater que sur les exemples où les bases de Gröbner manipulées ne sont pas des ensembles triangulaires (*Donati* et *Vermeer*) **SA2** est bien meilleur que **SA1**.

2.5.5 Ensembles triangulaires

Nous étudions les algorithmes obtenus dans la section 2.3 de ce chapitre. Dans le tableau 2.6, nous donnons les temps de calcul ainsi que les degrés des systèmes zéro-dimensionnels étudiés, obtenus en utilisant les décompositions en ensembles triangulaires de Kalkbrener implantées en Axiom par P. Aubry (voir [5]).

On constate que l'implantation de l'algorithme **KTSA1** permet de résoudre autant de problèmes dans notre jeu de tests que l'implantation de l'algorithme **SA2**. Signalons quand même que pour les exemples *Prodecco* et *F855*, les implantations de P. Aubry en Axiom ne permettent pas d'obtenir une décomposition en ensembles triangulaires réguliers au sens de Kalkbrener alors que Gb ‘fournit une base de Gröbner lexicographique.

System	Algorithme KTSA1		Algorithme SA2	
Vermeer	64	25	84	3,32
Wang	288	102,8	132	1,37
Euler	14	10,8	10	0,01
Neural	69	9,9	133	1,02
Butcher			15	1,7
Buchberger			32	<0,01
DiscPb	30	7,6	28	0,2
Donati	61	0,29	61	10
Hairer2	*NI*	∞	*NI*	∞
Prodecco	*NI*	∞	*NI*	∞
F633	*NI*	∞	*NI*	∞
F744	*NI*	∞	*NI*	∞
F855	*NI*	∞	*NI*	∞

TAB. 2.6 – *Algorithmes KTSA1 et SA2: comparaison des sorties et des temps de calcul*

System	Algorithme CSA2 + ZDS	Algorithme SA2 + ZDS
Vermeer	<1	3,32
Donati	<1	10
Hairer2	28,57	∞
Prodecco	934	∞
F633	175	∞
F744	90	∞
F855	7223	∞

TAB. 2.7 – *Algorithmes CSA2 et SA2: comparaison des temps de calcul*

2.5.6 Etudier les composantes compactes

Dans ce paragraphe, nous donnons les résultats obtenus lorsqu'on utilise la fonction de projection sur un axe. Les tests ne sont effectués que sur les exemples dont la taille du résultat et les temps de calcul sont significatifs. Le tableau 2.7 compare les temps de calcul des algorithmes **CSA2** et **SA2**.

Le tableau 2.8 compare la sortie de l'algorithme **CSA2** à celle de **SA2**.

Il apparait que, lorsque c'est possible, il faudra utiliser la fonction de projection sur un axe à la place de la fonction distance à un point. Les temps de calcul et la sortie de

System	Algorithme CSA2 + ZDS	Algorithme SA2 + ZDS
Vermeer	24	84
Donati	61	61
Hairer2	0	*NI*
Prodecco	36	*NI*
F633	12	*NI*
F744	69	*NI*
F855	76	*NI*

TAB. 2.8 – *Algorithmes CSA2 et SA2: comparaison des sorties*

l'algorithme **CSA2** sont significativement inférieurs à ceux de l'algorithme **SA2**.

2.6 Conclusions

Dans le cas des hypersurfaces contenant une infinité de points singuliers, la stratégie proposée dans ce Chapitre semble avantageuse devant la stratégie proposée dans le Chapitre précédent, basée sur une déformation infinitésimale.

De plus, nos implantations des algorithmes **SA1** et **SA2** permettent de résoudre des problèmes qui sont inaccessibles à l'une des meilleures implantations de l'algorithme de Décomposition Cylindrique Algébrique (rappelons néanmoins que cet algorithme a des spécifications plus générales que celles des algorithmes **SA1** et **SA2**). Les récents progrès effectués par J.-C. Faugère dans le calcul de décompositions en idéaux premiers permettent d'augmenter significativement la taille des problèmes pouvant être traités par nos algorithmes.

Nous avons constaté qu'en pratique, la taille de la sortie des algorithmes **SA1** et **SA2** est bien inférieure à celle de l'algorithme de Décomposition Cylindrique Algébrique. Néanmoins, du point de vue de la complexité théorique, nous n'avons pas pu donner de bornes satisfaisantes et valables dans tous les cas, sur la taille de la sortie des algorithmes **SA1** et **SA2**, ni sur le nombre d'opérations qu'ils effectuent. Dans le cas où la variété étudiée est équi-dimensionnelle et ne contient pas de singularités, on peut calculer les déterminants avec les polynômes de départ. Il est alors possible de donner une borne simplement exponentielle en le nombre de variables sur la taille de la sortie si on suppose que le premier choix du point A est le bon.

Dans les cas où la variété étudiée a un grand nombre de variables comparativement à sa dimension, nous avons constaté que l'algorithme **SA2** n'a pas un bon comportement en pratique. En effet, dans ce cas les déterminants calculés ont un degré et un nombre de monômes élevés. Dans le chapitre suivant, nous montrons comment remédier à ce problème dans certains cas.

Chapitre 3

Vers les systèmes d'équations et d'inéquations

Résumé

Ces travaux ont été effectués en collaboration avec P. Aubry et F. Rouillier. L'étude de variétés algébriques de dimension d est ramenée à l'étude d'ensembles constructibles dans R^{d+1} en utilisant les propriétés des ensembles triangulaires réguliers et séparables et les résultats du chapitre 3 de la partie I. Le premier algorithme que nous proposons décide uniquement du vide d'une variété algébrique réelle dans certains cas favorables. Le deuxième algorithme proposé calcule au moins un point par composante semi-algébriquement connexe d'une variété réelle. L'étude des ensembles constructibles se fait par l'introduction d'un infinitésimal [12, 13, 87, 82] dans les algorithmes du chapitre précédent. Nous donnons des outils permettant d'effectuer une décomposition équi-dimensionnelle de systèmes polynomiaux à coefficients dans $K(\varepsilon)$. Une dernière section de validation expérimentale confirme l'efficacité de ces nouveaux algorithmes sur les exemples considérés.

Sommaire

Introduction

Considérons la variété algébrique $V \subset \mathbb{C}^3$ définie par le système d'équations $S \subset \mathbb{Q}[x,y,z]$ ci-dessous :

$$(S) \quad \begin{cases} xz - 1 = 0 \\ y - x = 0 \end{cases}$$

Si nous déroulons l'algorithme **SA2**, en choisissant le point $A = (0,0,0)$ à l'origine, le déterminant D de la matrice

$$\begin{bmatrix} -1 & z & x \\ 1 & 0 & y \\ 0 & x & z \end{bmatrix}$$

est calculé, et on trouve $D = x^2 - z^2 + xy$. Le système d'équations polynomiales

$$\begin{cases} xz - 1 = 0 \\ y - x = 0 \\ x^2 - z^2 + xy = 0 \end{cases}$$

est de dimension 0 et de degré 4. En effet, les solutions s'écrivent plus simplement sous la forme :

$$\begin{cases} z - 2x^3 = 0 \\ y - x = 0 \\ 2x^4 - 1 = 0 \end{cases}$$

Bien évidemment, ce système admet deux racines réelles, chacune d'entre elle appartenant à l'une des deux composantes connexes de $V \bigcap \mathbb{R}^3$.

Remarquons dans un premier temps que S est un ensemble triangulaire régulier et séparable pour l'ordre $x < y < z$. Ainsi, comme $\dim(V(y - x) \cap V(x)) < \dim(V(y - x))$ et comme $xz - 1$ est de degré 1 en sa variable principale z, on peut espérer pouvoir décider si $V \bigcap \mathbb{R}^3$ est vide en étudiant uniquement $V' = V(y - x)$. Notons qu'alors si on choisit le point $A = (0,1)$, l'algorithme **SA2** calcule le polynôme $x + y - 1$ et engendre un système zéro-dimensionnel de degré 1. On constate ainsi une décroissance des degrés des déterminants calculés et des systèmes zéro-dimensionnels engendrés.

Maintenant que nous savons que $V \bigcap \mathbb{R}^3 \neq \emptyset$, continuons nos investigations et imposons-nous de trouver au moins un point par composante connexe sur cette variété réelle. Pour cela, il est évident (en faisant un dessin) qu'il suffit d'étudier $V(S) \cap V(x)$ d'une part et d'autre part de donner au moins un point dans chaque cellule de l'ensemble constructible défini par :

$$y - x = 0, \quad x \neq 0.$$

car, au-dessus de chacune de ces cellules, les racines réelles de $xz - 1$ varient continument (voir chapitre 3 partie I). Soit $e \in \mathbb{R} \setminus \{0\}$, il est clair sur cet exemple qu'il suffit de considérer les points (e,e) et $(-e, -e)$ pour obtenir un point dans chaque cellule de l'ensemble constructible. En substituant dans le système d'équations de départ, on se retrouve alors à résoudre deux systèmes zéro-dimensionnels de degrés 1, au lieu d'un seul système de degré 4. Là encore, on note une décroissance des degrés des systèmes zéro-dimensionnels que nous avons produits.

Plus généralement, notons que les deux stratégies que nous venons de décrire sur cet exemple simple permettent de travailler avec $d+1$ variables lorsqu'on étudie une variété équi-dimensionnelle de dimension d. Nous espérons ainsi résoudre le problème de la taille des déterminants calculés par l'algorithme **SA2** dans les cas où la dimension de la variété est petite devant le nombre de variables.

Soit K un corps réel, R sa clôture réelle et C sa clôture algébrique. Dans la première section de ce chapitre nous montrons comment ramener l'étude d'une variété algébrique réelle de R^n de dimension d à l'étude d'un ensemble constructible dans R^{d+1}, défini par une équation et plusieurs inéquations. Pour cela, nous utilisons les propriétés des ensembles triangulaires réguliers et séparables ainsi que quelques résultats du chapitre 3 de la partie I. Cette étude fait l'objet de la première section de ce chapitre.

Dans la deuxième section, nous décrivons les deux algorithmes obtenus à partir des résultats précédents. Le premier algorithme – que l'on appellera **SA4** – décide du vide dans certains cas favorables et généralise l'algorithme **SLSA** décrit dans le chapitre précédent. Le deuxième algorithme – que l'on appellera **SA5** – est une généralisation de l'algorithme **SA4** et renvoie au moins un point par composante connexe de la variété étudiée. Il utilise une routine permettant de trouver au moins un point par composante semi-algébriquement connexe dans un ensemble constructible défini par une seule équation et plusieurs inéquations. Pour cela nous restons dans le cadre de la méthode des points critiques et nous décrivons une telle routine basée sur les résultats de [13, 12, 87] qui introduisent une déformation infinitésimale et l'algorithme proposé dans [82] qui calcule les conditions de signe vérifiées par une famille de polynômes. Nous modifions ce dernier pour qu'il renvoie au moins un point par composante semi-algébriquement connexe dans un ensemble constructible défini par une seule équation et plusieurs inéquations.

Dans la troisième section, nous montrons comment optimiser l'algorithme **SA5**. En particulier, nous montrons comment calculer une décomposition équi-dimensionnelle d'un système d'équations polynomiales à coefficients dans $K[\varepsilon]$ en effectuant tous les calculs dans K. Puis, nous montrons comment utiliser les résultats du chapitre 1 de la partie I pour calculer une Représentation Univariée Rationnelle à coefficients infinitésimaux.

Enfin, dans la quatrième section, nous analysons le comportement pratique de cet algorithme. Nous le comparons en particulier à **SA2** et montrons que pour les problèmes de grande co-dimension que nous avons considérés, les algorithmes décrits dans ce chapitre sont bien plus efficaces que l'algorithme **SA2** du chapitre précédent.

Ces travaux ont été effectués en collaboration avec P. Aubry et F. Rouillier.

3.1 Préliminaires

Dans cette section, nous allons montrer comment ramener l'étude d'une variété équi-dimensionnelle de dimension d à celle d'un ensemble constructible de R^{d+1}. Pour cela, nous allons utiliser les propriétés des ensembles triangulaires réguliers séparables et les relier à certains résultats du chapitre 3 de la partie I.

Soit $\mathcal{G} \subset K[X_1,\ldots,X_n]$ une base de Gröbner lexicographique pour l'ordre $X_1 < \ldots < X_n$ engendrant un idéal radical et équi-dimensionnel, et $\mathcal{T} = (t_{d+1},\ldots,t_n)$ un ensemble triangulaire extrait de $\mathcal{G}$. Dans tout ce chapitre, on supposera que :

- $\mathcal{T}$ est un ensemble triangulaire régulier et séparable.
- $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.

Sans nuire à la généralité, on peut supposer que pour tout $i \in \{d+1,\ldots,n\}$, la variable principale de t_i est X_i. Pour tout $i \in \{d+1,\ldots,n\}$, on note

- h_i le coefficient dominant (que l'on appelle initial) de t_i vu comme un polynôme univarié en X_i,
- et s_i la dérivée (que l'on appelle séparant) de t_i par rapport à X_i.

Pour tout couple de polynômes (p,q) dans $K[X_1,\ldots,X_n]$, on note $\mathrm{res}(p,q,X_i)$ le résultant de p et de q vus comme polynômes univariés en la variable X_i. Soit p un polynôme quelconque de $K[X_1,\ldots,X_n]$. On note $\mathrm{mvar}(p)$ la plus grande variable apparaissant dans p pour l'ordre $X_1 < \ldots < X_n$. On définit le résultant de p modulo l'ensemble triangulaire $\mathcal{T}$ (que l'on notera $\mathrm{res}(p,\mathcal{T})$) par la procédure récursive suivante :

- si $\forall t \in \mathcal{T}$ on a $\mathrm{mvar}(t) \neq \mathrm{mvar}(p)$, alors $\mathrm{res}(p,\mathcal{T}) = p$,
- sinon, si $\exists t \in \mathcal{T}$ tel que $\mathrm{mvar}(t) = \mathrm{mvar}(p)$, alors

$$\mathrm{res}(p,\mathcal{T}) = \mathrm{res}(\mathrm{res}(p,t,\mathrm{mvar}(t)),\mathcal{T} \setminus \{t\}).$$

Lemme 3.1 *Soit $\mathcal{T}$ un ensemble triangulaire régulier séparable, p un polynôme quelconque. et M un point quelconque de $W(\mathcal{T})$. Alors, on a :*

$$\mathrm{res}(p,\mathcal{T})_{(M)} \neq 0 \Longrightarrow p(M) \neq 0.$$

Preuve : Sans nuire à la généralité, on peut supposer que les polynômes $t_{d+1},\ldots,t_n$ de $\mathcal{T}$ sont de variables principales $X_{d+1},\ldots,X_n$. Soit $M \in W(\mathcal{T})$ tel que $\mathrm{res}(p,\mathcal{T})_{(M)} \neq 0$. On considère la suite de polynômes

$$\begin{cases} R_n = \mathrm{res}(p,t_n,X_n) \\ R_{n-1} = \mathrm{res}(R_n,t_{n-1},X_{n-1}) \\ \vdots \\ R_{d+2} = \mathrm{res}(R_{d+3},t_{d+2},X_{d+2}) \\ R = R_{d+1} = \mathrm{res}(p,\mathcal{T}) = \mathrm{res}(R_{d+2},t_{d+1},X_{d+1}) \end{cases}$$

Puisque $M \in W(\mathcal{T})$, M n'annule aucun des initiaux de $t_{d+1},\ldots,t_n$. Donc, d'après la Proposition 2.2 du chapitre 2 de la partie I, $R_{d+2}(m_1,\ldots,m_d,X_{d+1})$ et $t_{d+1}(m_1,\ldots,m_d,X_{d+1})$ ont un pgcd trivial, ce qui implique que $R_{d+2}(M) \neq 0$. Il suffit alors d'itérer cet argument jusqu'à obtenir $p(M) \neq 0$. ■

Pour $i \in \{d+1,\ldots,n\}$, on note $\widetilde{s}_i = \mathrm{res}(s_i,\mathcal{T})$ et Π_{d+1} la projection de C^n dans C^{d+1} qui à un point $(x_1,\ldots,x_n)$ associe le point $(x_1,\ldots,x_{d+1})$.

Considérons :

- S une famille finie de points de R^n qui intersecte chaque composante semi-algébriquement connexe de $V(\mathcal{G}) \cap V(\mathrm{Sep}(\mathcal{T})) \cap R^n$,

– $\widetilde{\mathcal{F}}$ une famille de points réels de R^{d+1} intersectant chaque composante semi-algébriquement connexe de l'ensemble constructible défini par :

$$t_{d+1} = 0, \quad \widetilde{s}_{d+2} \neq 0, \ldots, \widetilde{s}_n \neq 0.,$$

et $\mathcal{F} = \Pi_{d+1}^{-1}(\widetilde{\mathcal{F}})$.

Remarque 3.1 *Notons que pour tout point $M \in R^n$, et pour tout $i \in \{d+1, \ldots, n\}$, on a :*

$$\mathrm{res}(\widetilde{s}_i, \mathcal{T})(M) \neq 0 \Longrightarrow h_i(M) \neq 0,$$

car h_i est un facteur du résultant associé au couple de polynômes (s_i, t_i) vus comme univarié en $\mathrm{mvar}(t_i)$.

Proposition 3.1 *La famille finie de points $S \bigcup \mathcal{F}$ intersecte chaque composante semi-algébriquement connexe de $V(\mathcal{G}) \bigcap R^n$.*

Preuve : Soit D une composante semi-algébriquement connexe de $V(\mathcal{G}) \bigcap R^n$.
Si $D \subset V(\mathcal{G}) \bigcap V(\mathrm{Sep}(\mathcal{T}))$, alors il est clair qu'il existe $M \in S$, tel que $M = (x_1, \ldots, x_n) \in D$. Supposons maintenant qu'il existe $M \in D$ tel que $M \notin V(\mathcal{G}) \bigcap V(\mathrm{Sep}(\mathcal{T}))$. On note $\widetilde{M} = \Pi_{d+1}(M) = (x_1, \ldots, x_{d+1})$ et $\widetilde{D}$ la composante semi-algébriquement connexe de l'ensemble constructible de R^{d+1} défini par :

$$t_{d+1} = 0, \quad \widetilde{s}_{d+2} \neq 0, \ldots, \widetilde{s}_n \neq 0$$

contenant $\widetilde{M}$. Dans la suite, nous allons montrer que pour tout $\widetilde{N} \in \widetilde{D}$, il existe $N \in R^n$ tel que $\Pi_{d+1}(N) = \widetilde{N}$ et $N \in D$. Soit donc N un point quelconque de $\widetilde{D}$. Puisque $\widetilde{D}$ est semi-algébriquement connexe, il existe un chemin γ semi-algébriquement continu de $]0,1[$ dans $\widetilde{D}$ tel que $\gamma(0) = \widetilde{M}$ et $\gamma(1) = \widetilde{N}$. D'après le théorème 3.2 du chapitre 3 de la partie I, il existe des fonctions semi-algébriquement continues $\xi_1, \ldots, \xi_\ell$ de $\gamma([0,1])$ dans R telles que l'ensemble des racines réelles de $t_{d+2}(\gamma(t), X_{d+2})$ est exactement l'ensemble $\{\xi_1(\gamma(t)), \ldots, \xi_\ell(\gamma(t))\}$. Puisque pour tout t, on a $\widetilde{s_{d+2}}(\gamma(t)) \neq 0$, toutes ces racines sont de multiplicité 1 et il existe un unique indice $i \in \{1, \ldots, \ell\}$ tel que $X_{d+2}(M) = \xi_i(\gamma(0))$. On note γ_{d+2} le chemin semi-algébriquement continu de $[0,1]$ dans R^{d+2} qui à t associe le point $(\gamma(t), \xi_i(\gamma(t)))$. D'après le lemme 3.1, on a pour tout $t \in [0,1]$ et tout $i \in \{d+2, \ldots, n\}$, $s_i(\gamma_{d+2}(t)) \neq 0$. Avec les mêmes arguments, on montre ainsi par récurrence que l'on peut construire un chemin γ_n de $[0,1]$ dans D tel que $\gamma_n(0) = M$ et $\gamma_n(1) = N$. ∎

Ainsi donner au moins un point par composante connexe de $V(\mathcal{G}) \bigcap R^n$ revient à :

– (i) donner au moins un point par composante semi-algébriquement connexe de

$$V(\mathcal{G}) \bigcap V(\mathrm{Sep}(\mathcal{T})) \bigcap R^n,$$

– (ii) donner au moins un point par composante semi-algébriquement connexe de

$$t_{d+1} = 0, \quad \widetilde{s}_{d+2} \neq 0, \ldots, \widetilde{s}_n \neq 0.$$

La tâche (i) peut être effectuée à l'aide de l'algorithme **SA2** du chapitre précédent. En revanche, la tâche (ii) nécessite de pouvoir donner au moins un point par composante semi-algébriquement connexe dans un ensemble constructible. Dans la section suivante, nous montrons comment utiliser **SA2** (ou **SA3**) et les résultats de [87] pour obtenir un tel algorithme.

3.2 Les algorithmes

Dans cette section, nous décrivons les deux algorithmes obtenus à partir des résultats précédents. Le premier (**SA4**) décide uniquement du vide d'une variété algébrique réelle dans certains cas favorables, le second (**SA5**) généralise **SA4** dans tous les cas et calcule au moins un point par composante connexe sur une variété algébrique réelle.

3.2.1 Décider du vide dans certains cas

Soit $\mathcal{G} \subset K[X_1,\ldots,X_n]$ une base de Gröbner lexicographique réduite et $\mathcal{T} = (t_{d+1},\ldots,t_n)$ un ensemble triangulaire régulier séparable extrait de $\mathcal{G}$. Sans nuire à la généralité, on suppose que $\forall i \in \{d+1,\ldots,n\}$ $\text{mvar}(t_i) = X_i$. D'après les résultats ci-dessus, pour donner au moins un point par composante semi-algébriquement connexe de $V(\mathcal{G})$, il faut pouvoir donner au moins un point par composante semi-algébriquement connexe de l'ensemble constructible $\mathcal{S}$ défini par :

$$t_{d+1} = 0, \quad \widetilde{s}_{d+2} \neq 0,\ldots,\widetilde{s}_n \neq 0$$

et donner au moins un point par composante semi-algébriquement connexe des variétés algébriques réelles $V(\mathcal{G}) \bigcap V(s_i)$. D'après la proposition 3.2, il est nécessaire d'étudier l'hypersurface réelle définie par $t_{d+1} = 0$. Dans la suite de ce paragraphe, nous montrons comment *dans certains cas «fréquents»*, la seule étude de cette hypersurface permet de décider du vide de la variété algébrique réelle que l'on veut étudier.

Définition 3.1 *On dit que $\mathcal{T}$ est en position k-favorable si et seulement si il existe $k \in \{d+1,\ldots,n\}$ tel que*

$$\forall i > k \quad \deg(t_i,X_i) = 1.$$

Lemme 3.2 *Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite et $\mathcal{T}$ un ensemble triangulaire régulier séparable extrait de $\mathcal{G}$. On suppose que $\mathcal{T}$ est en position k-favorable. On note $\mathcal{G}_k = \{g \in \mathcal{G} \mid g \in K[X_1,\ldots,X_k]\}$. On a :*

1. *si $V(\mathcal{G}_k) \bigcap R^k = \emptyset$ alors $V(\mathcal{G}) \bigcap R^n = \emptyset$,*
2. *si $\Big(\mathcal{D}(V(\mathcal{G}_k,A_k)) \setminus \Big(\text{Sing}(V(\mathcal{G}_k)) \bigcap V(\prod_{i=d+1}^n h_i)\Big)\Big) \bigcap R^k \neq \emptyset$, alors $V(\mathcal{G}) \bigcap R^n \neq \emptyset$.*

Preuve : Si $\mathcal{T}$ est en position k-favorable, on peut supposer sans nuire à la généralité que $t_j = h_jX_j + q_j$ avec h_j , $q_j \in K[X_1,\ldots,X_k]$, $\forall j = k+1\ldots n$.

1. C'est évident.
2. Supposons que $V(\mathcal{G}_k) \bigcap R^k \neq \emptyset$ et soit $M \in \mathcal{D}(V(\mathcal{G}_k,A_k)) \bigcap R^k$ où A_k est un point de K^k.
   - Si $M = (x_1,\ldots,x_k) \notin V(h_{k+1})$, il existe une unique valeur $y \in R$ telle que $M' = (x_1,\ldots,x_k,y) \in V(\mathcal{T}_{k+1})$. De plus, si $M \notin V(\prod_{j=d+1}^{k+1} h_j)$ alors, d'après le théorème 5.3, $M' \in V(\mathcal{G}_{k+1}) \bigcap R^{k+1}$.

– Supposons maintenant que $M \in V(h_{k+1})$ et $M \notin \mathrm{Sing}(V(\mathcal{G}_k))$. Puisque

$$\dim(V(h_{k+1}) \bigcap V(\mathcal{G}_k)) < \dim(V(\mathcal{G}_k))$$

et puisque M est un point régulier de $V(\mathcal{G}_k)$, il existe un voisinage U de M tel que $U \subset V(\mathcal{G}_k) \bigcap R^k$ contient un point N vérifiant $h_{k+1}(N) \neq 0$ et donc d'après l'item précédent $N \in V(\mathcal{G}_{k+1}) \bigcap R^{k+1}$.

Ainsi, on montre par récurrence que si

$$\left(\mathcal{D}(V(\mathcal{G}_k,A_k)) \setminus \left(\mathrm{Sing}(V(\mathcal{G}_k)) \bigcap V(\prod_{i=d+1}^{n} h_i)\right)\right) \bigcap R^k \neq \emptyset,$$

alors $V(\mathcal{G}) \bigcap R^n \neq \emptyset$.

■

Notons que les cas où

$$(\mathcal{D}(V(\mathcal{G}_k),A_k) \setminus (\mathrm{Sing}(V(\mathcal{G}_k)) \cap V(\prod_{i=d+1}^{n} h_i))) \bigcap R^k = \emptyset$$

$$\text{et } \ (\mathcal{D}(V(\mathcal{G}_k),A_k) \bigcap \mathrm{Sing}(V(\mathcal{G}_k)) \bigcap V(\prod_{i=d+1}^{n} h_i)) \bigcap R^k \neq \emptyset,$$

sont rares. On propose un algorithme spécifique essentiellement basé sur **SA2** et optimisé pour décider du vide uniquement.

Dans la suite, on note $\Delta(\mathcal{G}_k)$ la liste des mineurs d'ordre $k-d$ de la matrice jacobienne associée à $\mathcal{G}_k$ et on définit les nouvelles routines :

– **LexTriSetEquiDim :** une routine qui prend en entrée un système d'équations polynomiales S et qui renvoie en sortie une liste de bases de Gröbner lexicographiques réduites engendrant les idéaux premiers associés à $\sqrt{\langle S \rangle}$.
– **ZeroDimTest :** qui prend en entrée un système zéro-dimensionnel S et renvoie *true* si $V(S) \bigcap R^n = \emptyset$, sinon elle renvoie *false*.
– **cleaningStep :** qui prend en entrée un système zéro-dimensionnel S et un polynôme p, et renvoie une liste de solutions réelles de S qui n'annulent pas p.

Algorithme SA4

- **Entrée :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Sortie :** *true* si $V(S)\bigcap R^n = \emptyset$, sinon *false*.

1. list := LexTriSetEquiDim(S), result := [],
2. Choisir un point A dans K^n.
3. Tant que list $\neq \emptyset$ faire
   - (*)S := first(list), et enlever S de list, poser $d = \mathrm{Dim}(S)$,
   - Si $d = 0$ et si ZeroDimTest(S) = false alors retourner *false*,
   - $\mathcal{T} = ExtractTriangular(S)$.
   - Si $\mathcal{T}$ est en position k-favorable alors
     - newlist := SA2($\Gamma_A(\mathcal{T}_k) \cup S_k$),
     - Pour S' dans newlist, enlever S' si ZeroDimTest(S) = true,
     - Si newlist := $\emptyset$ alors affecter true à result et retourner au pas (*),
     - Pour S' dans newlist, enlever S' de newlist et si cleaningStep(S',$\Delta(\mathcal{G}_k)$) $\neq \emptyset$ alors retourner *false*,
     - Pour S' dans newlist, enlever S' de newlist et si cleaningStep($S', \prod_{i=d+1}^{n} h_i$) $\neq \emptyset$ alors retourner *false*,
   - result := result $\bigcup$ SA2(S),
4. Retourner result.

L'algorithme ci-dessus donne au moins un point par composante semi-algébriquement connexe si $\forall i \in \{k+1,\ldots,n\},\ h_i \in K$. Dans les autres cas, ceci ne peut être garanti que si :

$$(V(\mathcal{G}_k)\bigcap V(\prod_{i=d+1}^{n} h_i))\bigcap R^k = \emptyset.$$

3.2.2 Donner au moins un point par composante connexe

Résolution des systèmes d'équations et d'inéquations polynomiales

Pour donner au moins un point par composante semi-algébriquement connexe dans un ensemble constructible défini par une seule équation et plusieurs inéquations, nous choisissons de rester dans le cadre de la méthode des points critiques. Nous rappelons ci-dessous les résultats de [13, 12, 87] et nous appauvrissons l'algorithme décrit dans [82] (qui détermine toutes les conditions de signe vérifiées par une famille de polynômes) de manière à retourner au moins un point par composante connexe dans un ensemble constructible défini par une équation et plusieurs inéquations.

Considérons un ensemble semi-algébrique $\mathcal{S}$ défini par :

$$P_1 = \ldots = P_k = 0 \quad Q_1 > 0,\ldots,Q_s > 0$$

où les polynômes $P_1,\ldots,P_k,Q_1,\ldots,Q_s$ sont dans $K[X_1,\ldots,X_n]$. Dans [87], le résultat suivant est démontré.

Proposition 3.2 *[87] Soit D une composante semi-algébriquement connexe de l'ensemble semi-algébrique défini par $\mathcal{S}$. Il existe une famille d'indices $\{i_1,\ldots,i_\ell\}$ telle que la variété* *algébrique définie par :*

$$P_1 = \ldots = P_k = 0, \quad Q_{i_1} - \varepsilon = \ldots = Q_{i_\ell} - \varepsilon = 0$$

admette une composante semi-algébriquement connexe $C_{R\langle\varepsilon\rangle}$ strictement incluse dans l'extension de D à $R\langle\varepsilon\rangle^n$.

Ainsi, pour donner au moins un point par composante semi-algébriquement connexe de $\mathcal{S}$, il suffit de donner au moins un point par composante semi-algébriquement connexe sur les variétés algébriques réelles de $R\langle\varepsilon\rangle^n$ définies par :

$$P_1 = \ldots = P_k = 0, \quad Q_{i_1} - \varepsilon = \ldots = Q_{i_\ell} - \varepsilon = 0$$

pour chaque famille $\{i_1, \ldots ,i_\ell\} \subset \{1, \ldots ,s\}$. Ces points sont donnés sous la forme de Représentations Univariées Rationnelles à coefficients infinitésimaux

$$\begin{cases} f(\varepsilon,T) = 0 \\ g_0(\varepsilon,T)X_1 - g_1(\varepsilon,T) \\ \vdots \\ g_0(\varepsilon,T)X_n - g_n(\varepsilon,T) \end{cases}$$

Il reste alors à tester le signe de Q_j (pour $j \notin \{i_1, \ldots ,i_\ell\}$) en les points définis par cette Représentation Univariée Rationnelle. Notons que pour donner un point dans chaque composante semi-algébriquement connexe de l'ensemble semi-algébrique, il faut trouver un rationnel e suffisamment petit, tel que :

- le nombre de racines réelles de $f(\varepsilon,T)$ ne varie pas lorsque ε parcourt l'intervalle $]0,e]$,
- le signe des polynômes $Q_j(\frac{g_1(\varepsilon,T)}{g_0(\varepsilon,T)}, \ldots ,\frac{g_1(\varepsilon,T)}{g_0(\varepsilon,T)})$ (pour $j \notin \{i_1, \ldots ,i_\ell\}$) en les racines de $f(\varepsilon,T)$ ne varie pas lorsque ε parcourt l'intervalle $]0,e]$.

Ceci peut se faire en calculant une décomposition cylindrique algébrique adaptée à la famille de polynômes

$$(f(\varepsilon,T),(Q_j^h(g_0(\varepsilon,T),g_1(\varepsilon,T), \ldots ,g_n(\varepsilon,T))_{j \notin \{i_1,\ldots,i_\ell\}}))$$

(où $Q_j^h(h,X_1, \ldots ,X_n)$ est le polynôme obtenu en homogénéisant Q_j par la variable h). Puisqu'à ce stade de l'algorithme, on travaille avec des polynômes de deux variables, les optimisations de l'opérateur projection décrites dans [70] sont applicables. On note ε-**Substitution** la routine qui trouve un rationnel e vérifiant les propriétés ci-dessus.
On en déduit un premier algorithme qui donne au moins un point par composante semi-algébriquement connexe d'un ensemble semi-algébrique. On note :

- ε-**SA2** : une routine qui prend en entrée un système d'équations polynomiales $S_\varepsilon \subset K(\varepsilon)[X_1, \ldots ,X_n]$) et renvoie un ensemble de systèmes zéro-dimensionnels contenant au moins un point par composante semi-algébriquement connexe dans $V(S_\varepsilon) \bigcap R\langle\varepsilon\rangle^n$
- ε-**RUR** : une routine qui prend en entrée un système zéro-dimensionnel à coefficients dans $K(\varepsilon)$ et renvoie une Représentation Univariée Rationnelle représentant les solutions du système d'entrée.

Algorithme ESA

- **Entrée :** Deux listes de polynômes $[P_1,\dots,P_k]$ et $[Q_1,\dots,Q_s]$ représentant un ensemble semi-algébrique $\mathcal{S}$ défini par plusieurs équations $P_1 = \dots = P_k = 0$ et plusieurs inégalités $Q_1 > 0,\dots,Q_s > 0$.
- **Sortie :** Une liste de Représentations Univariées Rationnelles représentant au moins un point par composante semi-algébriquement connexe dans $\mathcal{S}$.

1. Construire tous les systèmes du type
$$P_1 = \dots = P_k = Q_{i_1} - \varepsilon = \dots = Q_{i_\ell} - \varepsilon = 0$$
où $\{i_1,\dots,i_\ell\} \subset \{1,\dots,s\}$ et les mettre dans la variable list. result := []
2. Tant que list $\neq$ [] faire
   - S_ε := first(list), enlever S_ε de list,
   - newlist := ε-SA2(S_ε),
   - Pour tout S dans newlist faire
     - rur$_\varepsilon$:= ε-RUR(newlist),
     - remplacer ε par e := ε-Substitution(rur$_\varepsilon$,$(Q_j)_{j\notin\{i_1,\dots,i_\ell\}}$). Soit rur$_e$ la RUR obtenue.
     - faire result := result $\cup$ rur$_e$.
3. Retourner result.

L'algorithme SA5

Dans ce paragraphe, nous donnons un algorithme qui retourne au moins un point par composante connexe d'une variété algébrique réelle quelconque définie par un système d'équations polynomiales. Cet algorithme se déduit directement des propositions 3.1 et 3.2 de ce chapitre. Il utilise une déformation infinitésimale pour l'étude des systèmes d'équations et d'inéquations apparaissant au cours des calculs.

Algorithme SA5

- **Entrée :** Un système d'équations polynomiales S dans $K[X_1,\dots,X_n]$.
- **Sortie :** Une liste de Représentations Univariées Rationnelles représentant au moins un point par composante semi-algébriquement connexe sur $V(S)\bigcap R^n$.

1. list := LexTriSetEquiDim(S) et faire result := [],
2. Tant que list $\neq$ [] faire
   - S := first(list), enlever S de list et faire newlist := [].
   - $\mathcal{T} = (t_{d+1},\dots,t_n)$:= ExtractTriangular(S) et calculer $\widetilde{s}_{d+2},\dots,\widetilde{s}_n$.
   - newlist := newlist $\bigcup$ ESA($[t_{d+1}]$,$[\pm\widetilde{s}_{d+2},\dots,\pm\widetilde{s}_n]$)
   - Pour tout S' dans newlist, faire result := result $\bigcup RUR(S \cup S')$.
   - Pour $i \in \{d+2,\dots,n\}$ faire list := list $\bigcup$ LexTriSetEquiDim($S \cup \{s_i\}$),
3. Retourner result.

D'un point de vue pratique, il est important de noter les points suivants :

- Concernant l'usage de l'infinitésimal, on peut remarquer que celui-ci est de nature différente de celle de l'infinitésimal introduit dans le chapitre 1 de la partie II : en

effet, dans cet algorithme, l'infinitésimal n'est pas introduit pour désingulariser une variété.
- D'après le Lemme 3.1 et la Proposition 3.1 de ce chapitre, on doit calculer le résultant des séparants de l'ensemble triangulaire modulo l'ensemble triangulaire. Il est bien évidemment plus efficace de calculer les discriminants puis de considérer les initiaux.

Remarque 3.2 *Lorsqu'on est en dimension* 1*, on ne travaille plus qu'avec des polynômes de* 2 *variables. Dans ces cas-là, il est inutile d'introduire un infinitésimal: on peut directement appliquer l'algorithme de Décomposition Cylindrique Algébrique qui a un bon comportement pour des problèmes de deux variables.*

L'algorithme ci-dessus ne doit pas être opposé aux algorithmes **SA2/SA3** qui ont les mêmes spécifications. En effet, l'algorithme **SA5** est conçu pour réduire la taille des déterminants calculés par les algorithmes **SA2/SA3** lorsque la dimension est inférieure à la co-dimension. Dans les cas où c'est la co-dimension qui est inférieure à la dimension, ce sont les algorithmes **SA2/SA3** qui s'avèrent les plus performants : en effet, dans ces cas-là les déterminants calculés par **SA5** sont plus gros que ceux calculés par les algorithmes **SA2/SA3**.

3.3 Optimisations

Dans cette section, nous présentons deux optimisations qui peuvent être appliquées aux algorithmes décrits ci-dessus faisant intervenir des infinitésimaux.

3.3.1 Décomposition de systèmes à coefficients dans $K(\varepsilon)$

Dans cette section, nous allons montrer comment, étant donné un système d'équations polynomiales S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$, on calcule une famille de bases de Gröbner lexicographiques $\mathcal{G}_{1,\varepsilon},\ldots,\mathcal{G}_{\ell,\varepsilon}$, à coefficients dans $K[\varepsilon]$ telles que :
- $\sqrt{S_\varepsilon} = \langle \mathcal{G}_{1,\varepsilon}\rangle \bigcap \ldots \bigcap \langle \mathcal{G}_{\ell,\varepsilon}\rangle$,
- Pour tout i dans $\{1,\ldots,\ell\}$, l'idéal $\langle \mathcal{G}_{i,\varepsilon}\rangle$ est équi-dimensionnel,
- Pour tout i dans $\{1,\ldots,\ell\}$, on peut extraire de $\mathcal{G}_{i,\varepsilon}$ un ensemble triangulaire régulier séparable $\mathcal{T}_{i,\varepsilon}$ tel que $\mathrm{sat}(\mathcal{T}_{i,\varepsilon}) = \langle \mathcal{G}_{i,\varepsilon}\rangle$.

Considérons la famille de bases de Gröbner réduites $\mathcal{G}_1,\ldots,\mathcal{G}_s$ obtenue par **LexTriSetEquiDim** appliquée au système d'équations polynomiales S_E obtenu en remplaçant l'infinitésimal ε par la variable E pour l'ordre $E < X_1 < \ldots < X_n$. On a donc dans $K[E,X_1,\ldots,X_n]$:

$$\sqrt{S_E} = \langle \mathcal{G}_1\rangle \bigcap \ldots \bigcap \langle \mathcal{G}_s\rangle$$

On note Π_ε l'application de spécialisation de la variable E en l'infinitésimal ε. D'après le lemme 1.7 du chapitre 1 de la partie II, on a pour tout $i \in \{1,\ldots,s\}$, $\langle \Pi_\varepsilon(\mathcal{G}_i)\rangle = \Pi_\varepsilon(\langle \mathcal{G}_i\rangle)$ et $\Pi_\varepsilon(\mathcal{G}_i)$ est une base de Gröbner de $\Pi_\varepsilon(\langle \mathcal{G}_i\rangle)$. Sans nuire à la généralité, on suppose qu'il existe $\ell \in \mathbb{N}$, tel que pour $\{1,\ldots,\ell\} \subset \{1,\ldots,s\}$ est l'ensemble des indices i tel que

$\Pi_\varepsilon(\mathcal{G}_i)$ ne contienne aucun élément de $R\langle\varepsilon\rangle$. Pour $i \in \{1,\ldots,s\}$, on note $\mathcal{G}_{i,\varepsilon} = \Pi_\varepsilon(\mathcal{G}_i)$. Notons qu'on a bien évidemment :

$$\sqrt{S_\varepsilon} = \langle\mathcal{G}_{1,\varepsilon}\rangle \bigcap \ldots \bigcap \langle\mathcal{G}_{\ell,\varepsilon}\rangle.$$

Soit $e \in R$, on note Π_e l'application de spécialisation de la variable E en e, et

$$\mathcal{E} = \{e \in R \mid \langle\Pi_e(\mathcal{G}_i)\rangle = \Pi_e(\langle\mathcal{G}_i\rangle)\}$$

Lemme 3.3 *L'ensemble des réels e inclus dans $\mathcal{E}$ tels que $\langle\Pi_e(\mathcal{G}_i)\rangle$ n'est pas radical est fini.*

Preuve : D'après le théorème de Sard, l'ensemble des valeurs critiques de la projection sur l'axe de coordonnées de la variable E restreinte aux points réguliers de $V(\mathcal{G}_i)$ est un nombre fini de points. Pour toute valeur e de E n'appartenant pas à cet ensemble fini de points la jacobienne associée à $\Pi_e(\mathcal{G}_i)$ est de rang maximal, donc pour de tels e, l'idéal $\langle\Pi_e(\mathcal{G}_i)\rangle$ est radical. ∎

D'après ce lemme, les bases de Gröbner $\mathcal{G}_{i,\varepsilon}$ engendrent des idéaux radicaux. Pour $i \in \{1,\ldots,\ell\}$, soit $\mathcal{T}_i$ un ensemble triangulaire régulier et séparable extrait de $\mathcal{G}_i$. On note $\mathcal{T}_{i,e} = \Pi_e(\mathcal{T}_i)$.

Lemme 3.4 *L'ensemble des valeurs e incluses dans $\mathcal{E}$ telles que $\mathcal{T}_{i,e}$ n'est pas un ensemble triangulaire régulier séparable est fini.*

Preuve : D'après les définitions de régularité et de séparabilité (voir chapitre 5 de la partie I), tester si un ensemble triangulaire est régulier et séparable revient à tester la non-appartenance de polynômes à des idéaux. Donc l'ensemble des valeurs e telles que $\mathcal{T}_{i,e}$ est un ensemble triangulaire régulier séparable est un ouvert constructible pour la topologie de Zariski. Donc, son complémentaire est soit vide, soit un ensemble fini de points. ∎

D'après le lemme précédent, $\mathcal{T}_{i,\varepsilon} \subset \mathcal{G}_{i,\varepsilon}$ est un ensemble triangulaire régulier et séparable. Des arguments similaires à ceux de la preuve ci-dessus montrent que :

$$\mathrm{sat}(\mathcal{T}_{i,\varepsilon}) = \langle\Pi_\varepsilon(\mathrm{sat}(\mathcal{T}_i))\rangle = \langle\mathcal{G}_{i,\varepsilon}\rangle.$$

On obtient l'algorithme ci-dessous :

Algorithme ε-EDD

- **Entrée :** Un système d'équations polynomiales S_ε dans $K[\varepsilon][X_1,\ldots,X_n]$.
- **Sortie :** Une famille de bases de Gröbner lexicographiques $\mathcal{G}_1,\ldots,\mathcal{G}_\ell$ telles que
  - $\sqrt{S_\varepsilon} = \langle \mathcal{G}_{1,\varepsilon}\rangle \bigcap \cdots \bigcap \langle \mathcal{G}_{\ell,\varepsilon}\rangle$,
  - Pour tout i dans $\{1,\ldots,\ell\}$, l'idéal $\langle \mathcal{G}_{i,\varepsilon}\rangle$ est équi-dimensionnel,
  - Pour tout i dans $\{1,\ldots,\ell\}$, on peut extraire de $\mathcal{G}_{i,\varepsilon}$ un ensemble triangulaire régulier séparable $\mathcal{T}_{i,\varepsilon}$ tel que $\mathrm{sat}(\mathcal{T}_{i,\varepsilon}) = \langle \mathcal{G}_{i,\varepsilon}\rangle$.

1. Remplacer ε par une variable E dans S_ε. On obtient S_E.
2. list := LexTriSetequiDim(S_E) (où E est choisie comme étant la plus petite des variables).
3. Réduire chaque élément de list et les mettre dans la variable result.
4. Retourner result.

3.3.2 Phase de nettoyage

Soit $p \in K[X_1,\ldots,X_n]$ et $\mathcal{T}$ un ensemble triangulaire régulier et séparable. Dans le chapitre 5 de la partie I, nous avons rappelé la définition de pseudo-reste d'un polynôme p modulo un ensemble triangulaire régulier et séparable ainsi que la propriété suivante :

$$p \in \mathrm{sat}(\mathcal{T}) \Longleftrightarrow \mathrm{prem}(p,\mathcal{T}) = 0.$$

Dans l'algorithme **SA5**, on étudie des ensembles semi-algébriques de la forme

$$t = 0, \quad q_1 \neq 0,\ldots,q_s \neq 0$$

Après chaque appel à ε-**EDD**, on obtient des bases de Gröbner lexicographiques dont on peut extraire un ensemble triangulaire régulier et séparable. On peut alors tester pour chacun de ces ensembles triangulaires si il existe i tel que $\mathrm{prem}(q_i,\mathcal{T}) = 0$ auquel cas la composante dont est extrait $\mathcal{T}$ n'a pas lieu d'être étudiée.
Il n'y a pas de raison pour qu'en toute généralité cette étape de nettoyage apporte un gain significatif en pratique. En revanche, il convient de remarquer que l'algorithme **SA5** étudie les projections des composantes irréductibles de la variété que l'on désire étudier. Il est donc fréquent que les inéquations introduites par l'algorithme **SA5** éliminent certaines composantes obtenues par ε-**EDD**.

3.3.3 Calcul de Représentations Univariées Rationnelles à coefficients infinitésimaux

Dans le chapitre 1 de la partie II, nous donnons un algorithme permettant de calculer des Représentations Univariées Rationnelles de systèmes de n équations polynomiales en n variables, à coefficients dans $K(\varepsilon)$ et définissant un idéal radical. Nous ne pouvons donc pas appliquer directement cet algorithme aux systèmes zéro-dimensionnels à coefficients dans $K(\varepsilon)$ renvoyés par ε-**SA2**, qui peuvent contenir plus de n équations.

Soit $S_\varepsilon \subset K(\varepsilon)[X_1,\ldots,X_n]$ un de ces systèmes et $[\mathcal{G}_1,\ldots,\mathcal{G}_\ell] = \varepsilon$-**EDD**. Donc chacune des bases de Gröbner $\mathcal{G}_1,\ldots,\mathcal{G}_\ell$ engendre un idéal radical de $K(\varepsilon)[X_1,\ldots,X_n]$. De plus, pour tout $i \in \{1,\ldots,\ell\}$, on peut extraire de $\mathcal{G}_i$ un ensemble triangulaire $\mathcal{T}_i$ de $K(\varepsilon)[X_1,\ldots,X_n]$ tel que $\mathrm{sat}(\mathcal{T}_i) = \langle \mathcal{G}_i \rangle$ (dans $K(\varepsilon)[X_1,\ldots,X_n]$) et tel que les polynômes de $\mathcal{T}_i$ ont un initial appartenant à $K(\varepsilon)$ (voir [5] ou chapitre 5 de la partie I). Dans ce cas, il est évident que $\mathrm{sat}(\mathcal{T}_i) = \langle \mathcal{T}_i \rangle$ dans $K(\varepsilon)[X_1,\ldots,X_n]$ (et que cet idéal est radical). Comme l'ensemble triangulaire $\mathcal{T}_i$ contient n polynômes, l'algorithme ε-**RUR** décrit dans le chapitre 1 de la partie I peut être appliqué en prenant en entrée l'ensemble triangulaire $\mathcal{T}_i$.

3.4 Validation expérimentale

3.4.1 Méthodologie

Nous comparons les algorithmes **SA4** et **SA5** proposés dans ce chapitre à l'algorithme **SA2** proposé dans le chapitre précédent. Nous utilisons pour ces trois algorithmes les mêmes implantations de décomposition de systèmes polynomiaux que nous décrivons dans l'Annexe A de ce document. Le calcul de Représentation Univariée Rationnelle à coefficients infinitésimaux se fait avec les outils proposés dans le chapitre 1 de la partie II. Les degrés des systèmes zéro-dimensionnels produits d'une part par l'algorithme **SA2** puis par les algorithmes **SA4** et **SA5** seront comparés. Pour effectuer ces comparatifs, on n'a considéré que les systèmes polynomiaux que l'algorithme **SA2** résoud en un temps significatif ($>$ 1 sec.) ou ne peut résoudre. Tous les calculs ci-dessous ont été effectués sur un PC Pentium II 400 MHz avec 512 Mo de RAM de l'UMS MEDICIS [3]. Les temps sont donnés en secondes.

3.4.2 Algorithmes SA4/SA2

Dans le tableau 3.1, nous comparons les **sommes** des degrés des systèmes zéro-dimensionnels engemdrés par les algorithmes **SA2** et **SA4**. Les exemples signalés par la présence d'un astérisque * sont des exemples pour lesquels l'algorithme **SA4** trouve au moins un point par composante connexe. Lorsque nous inscrivons la mention *NI* dans une case, ceci signifie que l'algorithme correspondant n'a pas pu résoudre le problème. Il est difficile de tirer des conclusions d'un tel tableau : il n'y a pas de différence entre les degrés des systèmes zéro-dimensionnels produits par les algorithmes **SA2** et **SA4** lorsque l'algorithme **SA2** termine. Nous avons donc complété ces expérimentations en utilisant un prototype du logiciel FGb (écrit en C par J.-C. Faugère), disponible à l'URL `https://calfor.lip6.fr`. Nous avons constaté que le degré des systèmes zéro-dimensionnels produits par l'algorithme **SA4** est bien inférieur à celui des degrés des systèmes zéro-dimensionnels produits par l'algorithme **SA2**. A titre d'exemple, l'algorithme **SA2** produit un idéal de degré 322 alors que l'algorithme **SA4** en produit un de degré 67.

Dans le tableau 3.2, les temps de calcul des algorithmes **SA2** et **SA4** sont comparés. En terme de temps de calcul, la différence entre l'algorithme **SA2** et l'algorithme **SA4** s'explique par la taille des déterminants calculés. On constate que les systèmes zéro-dimensionnels engendrés par l'algorithme **SA4** sont bien plus simples à résoudre et ceci

System	Algorithme SA2	Algorithme SA4
Vermeer	84	84
Donati	61	61
Hairer2	*NI*	1
Prodecco	*NI*	18
F633	*NI*	67
F744	*NI*	55
F855	*NI*	64

TAB. 3.1 – *Algorithmes SA2 et SA4 : comparaison des sorties*

System	Algorithme SA2 + ZDS	Algorithme SA4 +ZDS
Vermeer	3.32	<1
Donati	10	10
Hairer2	∞	23.03
Prodecco	∞	286
F633	∞	5700
F744	∞	40
F855	∞	5664

TAB. 3.2 – *Algorithmes SA2 et SA4 : comparaison des temps de calcul*

induit des progrès significatifs.

3.4.3 Algorithmes SA5/SA2

Dans le tableau 3.3, nous comparons la somme des degrés des systèmes zéro-dimensionnels produits par les deux algorithmes **SA2** et **SA5**. Sur les deux seuls systèmes que l'algorithme **SA2** parvient à résoudre, la taille de la sortie de l'algorithme **SA5** est inférieure à celle de l'algorithme **SA2**. Sur les exemples significatifs, ceci reste vrai (mais il faut alors utiliser FGb pour s'en apercevoir).

Dans le tableau 3.4, nous comparons les temps de calcul des algorithmes **SA2** et **SA5**. L'algorithme **SA5** semble plus efficace que l'algorithme **SA2** sur les problèmes de dimension 1. Pour les systèmes polynomiaux de dimension strictement supérieure à 1, l'algorithme **SA5** subit un facteur combinatoire (nombre de systèmes à coefficients infinitésimaux à étudier) qui sont «prédits» dans les théorèmes de complexité de [13].

System	Algorithme SA2 + ZDS	Algorithme SA5 + ZDS
Vermeer	84	36
Donati	61	
Hairer2	*NI*	451
Prodecco	*NI*	36
F633	*NI*	*NI*
F744	*NI*	113
F855	*NI*	142

TAB. 3.3 – *Algorithmes SA2 et SA5 : comparaison des sorties*

System	Algorithme SA2 + ZDS	Algorithme SA5 + ZDS
Vermeer	3,32	2,8
Hairer2	*NI*	5684
Prodecco	*NI*	938,6
F633	*NI*	∞
F744	*NI*	198
F855	*NI*	8464

TAB. 3.4 – *Algorithmes SA2 et SA5 : comparaison des temps de calcul*

L'algorithme **SA5** parvient néanmoins à résoudre le système Hairer2 qui est inaccessible à l'algorithme **SA2** (même en utilisant le logiciel FGb). L'algorithme **SA5** ne parvient pas à traiter le système *F633*. Le calcul de Représentation Univariée Rationnelle n'est pas l'étape bloquante de l'algorithme : c'est l'exploitation de ces RUR (voir section 3.2.2) qui est délicate et empêche la résolution.

A titre indicatif, pour l'exemple *F633*, le degré maximal du système zéro-dimensionnel à coefficients infinitésimaux est 31. Il fallait étudier un ensemble constructible défini par une équation et seulement 5 inéquations (alors que le problème de départ contient 10 variables et que la variété est de dimension 2). Au total, ce sont 45 systèmes à coefficients infinitésimaux qui ont été étudiés. Sur ces 45 systèmes, 40 étaient de dimension zéro et de degré 2 ou 3. Les 5 autres systèmes restants à étudier étaient de dimension 1. Leur étude par l'algorithme ε-**SA2** a engendré 2 systèmes de degré 31 et 2 systèmes de degré 12 et un système de degré 11.

3.5 Conclusions

Nous avons montré comment, dans certains cas favorables, on pouvait simplifier l'algorithme **SA2** pour obtenir un algorithme (**SA4**) qui ne décide que du vide d'une variété algébrique réelle. Sur les exemples considérés, l'algorithme obtenu offre des performances meilleures de plusieurs ordres de grandeur que celles de l'algorithme **SA2**. Le principe de l'algorithme **SA4** a été généralisé et on a ainsi obtenu un algorithme (**SA5**) dont le comportement est lui aussi meilleur que celui de l'algorithme **SA2**, au moins sur les exemples considérés.

Ces conslusions sont néanmoins partielles. En effet, la plupart des exemples considérés sont de dimension 1 ce qui est «avantageux» pour l'algorithme **SA5**. Pour les 2 seuls exemples de dimension 2 (Hairer2 et F633), l'algorithme **SA5** semble pâtir d'un facteur combinatoire (nombre de systèmes à coefficients infinitésimaux à résoudre). L'étape bloquante sur un seul de ces exemples est la résolution de systèmes d'équations et d'inéquations polynomiales, et plus particulièrement la taille des Représentations Univariées Rationnelles, dont l'exploitation est devenue problématique sur l'exemple F633.

Ainsi, même si ces résultats semblent encourageants, la résolution des systèmes d'équations et d'inéquations doit faire l'objet d'une étude plus approfondie, en particulier pour améliorer le calcul de Représentations Univariées Rationnelles à coefficients infinitésimaux d'une part et la routine ε-Substitution d'autre part. Les améliorations qui seraient apportées à ces deux étapes de calcul permettraient d'améliorer le comportement en pratique de l'algorithme **SA5**.

Par ailleurs, il faut encore mieux identifier la classe d'exemples pour laquelle les algorithmes **SA4** et **SA5** seront plus efficaces que l'algorithme **SA2**, en particulier en résolvant des applications.

Chapitre 4

Problème d'interpolation de Birkhoff

Résumé

Soit f une fonction de $\mathbb{R}$ dans $\mathbb{R}$, on note $f^{(j)}$ sa j-ième dérivée partielle. Soit $x_1,\ldots,x_n$ (avec $x_1 < \ldots < x_n$) un ensemble ordonné de points réels tels que pour une famille $\mathcal{I}$ de couples $(i,j) \in \{1,\ldots,n\} \times \{0,\ldots,r\}$ les valeurs de $f^{(j)}(x_i) = f_{i,j}$ soient connues. Le problème qui consiste à trouver un polynôme $P \in \mathbb{R}[X]$ de degré borné par r tel que, pour $(i,j) \in \mathcal{I}$, le polynôme P vérifie $P^{(j)}(x_i) = f_{i,j}$ s'appelle le problème d'interpolation de Birkhoff.

Dans [48], L. Gonzalez-Vega montre que le problème qui consiste à déterminer les listes de couples $\mathcal{I}$ pour lesquels le problème d'interpolation de Birkhoff est résoluble revient à étudier un ensemble d'hypersurfaces. Il s'agit de vérifier pour chacune d'entre elles si elles contiennent au moins un point réel dont aucune des coordonnées n'est nulle.

Après avoir rappelé les résultats de [48], nous montrons comment utiliser les algorithmes décrits dans ce document pour résoudre ce problème. Puis nous donnons les résultats obtenus dans le cas $n = 5$ et $r = 4$.

Sommaire

Introduction

Le problème qui consiste à interpoler une fonction inconnue $f : R \longrightarrow R$ par un polynôme univarié en connaissant les valeurs de f et de certaines de ses dérivées en des points de R est un problème classique d'Analyse numérique et en Théorie de l'Approximation.

Deux cas d'Interpolation classiques ont été étudiés et résolus : il s'agit de la Formule d'Interpolation de Lagrange et du Problème d'Interpolation de Hermite. Dans le premier cas, les valeurs de f en les points $x_0 < \ldots < x_n$ sont connues et la Formule d'Interpolation de Lagrange montre l'existence d'un unique polynôme de degré inférieur ou égal à n interpolant f en les x_i.

Le problème d'Interpolation de Hermite généralise le cas précédent en incluant des informations sur les dérivées de f. Soit $x_1 < \ldots < x_n$ des points donnés et $\nu_1, \ldots, \nu_n$ des entiers positifs : le Problème d'Interpolation de Hermite est résolu en prouvant qu'il existe un unique polynôme P de degré inférieur ou égale à $\nu_1 + \ldots + \nu_n - 1$ tel que pour tous $k \in \{1, \ldots, n\}$ et $j \in \{0, \ldots, \nu_k - 1\}$ l'égalité suivante est vérifiée :

$$f^{(j)}(x_k) = P^{(j)}(x_k).$$

Les problèmes d'interpolation peuvent être présentés de manière générale en décrivant les conditions d'interpolation en termes de *matrices d'incidence* : de telles matrices contiennent l'information connue sur f.

Définition 4.1 *Soit n et r deux entiers tels que $n \geq 1$ et $r \geq 0$. La matrice*

$$\mathcal{E} = \begin{pmatrix} e_{1,0} & \ldots & e_{1,r} \\ \vdots & & \vdots \\ e_{n,0} & \ldots & e_{n,r} \end{pmatrix}$$

est appelée matrice d'incidence si $e_{i,j} \in \{0,1\}$ pour tout couple (i,j).

Pour une matrice d'incidence, on note $|\mathcal{E}|$ le nombre de 1 dans $\mathcal{E}$:

$$|\mathcal{E}| = \sum_{i,j} e_{i,j}$$

Dans le cas où $|\mathcal{E}|$ est égal au nombre de colonnes dans $\mathcal{E}$, on dira que $\mathcal{E}$ est une *matrice d'incidence normale.*

Soit $\chi = \{x_1, \ldots, x_n\}$ un ensemble de nombres réels tels que $x_1 < \ldots < x_n$ et $\mathcal{F}$ une matrice de nombre réels donnés ayant le même nombre de rangées et de colonnes que $\mathcal{E}$ et dont on notera les éléments $f_{i,j}$. Le problème de déterminer un polynôme P dans $R[x_1, \ldots, x_n]$ de degré plus petit que r qui interpole $\mathcal{F}$ en $(\chi,\mathcal{E})$ c'est-à-dire qui vérifie les conditions :

$$P^{(j)}(x_i) = f_{i,j} \quad \text{ssi} \quad e_{i,j} = 1$$

est connu sous le nom de *Problème d'Interpolation de Birkhoff.*

Définition 4.2 *Une matrice d'incidence normale $\mathcal{E}$ ayant n rangées et $r+1$ colonnes est dite* équilibrée *si pour tout choix de noeuds $x_1 < \ldots < x_n$ et d'une matrice $\mathcal{F}$ il existe un unique polynôme P de degré inférieur ou égal à r qui interpole $\mathcal{F}$ en $(\chi,\mathcal{E})$.*

Un premier exemple de matrice d'incidence *équilibrée* est celle qui correspond à la Formule d'Interpolation de Lagrange (avec $r = n - 1$) :

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix}$$

Un second exemple vient du Problème d'Interpolation de Lagrange: pour tout choix d'entiers positifs $\nu_1,\ldots,\nu_n$, la matrice ayant n rangées et $N = \nu_1 + \ldots + \nu_n$ colonnes telle que dans la i-ième rangée, il y ait ν_i 1 est équilibrée.

Dans tout ce chapitre, nous nous référons à [48]. Dans la première section, nous rappelons [48] comment caractériser les matrices d'incidence normales pour le Problème d'Interpolation de Birkhoff. En particulier, dans [48], L. Gonzalz-Vega montre que si l'on sait résoudre un certain problème d'élimination des quantificateurs, on sait décider si une matrice d'incidence donnée est équilibrée. La deuxième section de ce chapitre rappelle comment L. Gonzalez-Vega simplifie ce problème en le ramenant à l'étude d'un ensemble semi-algébrique dont on doit décider s'il est vide ou pas. Dans la troisième section, nous décrivons les stratégies de résolution que nous allons utiliser pour exprimer cet ensemble algébrique plus simplement. Puis, nous rappellerons les algorithmes que nous avons utilisés pour résoudre ce problème dans le cas $n = 5$ et $r = 4$, dont une résolution partielle et non automatisée e'st déjà proposée dans [80]. Enfin, nous donnons les résultats obtenus.

4.1 Caractérisation des matrices équilibrées

Dans cette section, on montre comment déterminer si une matrice d'incidence donnée est équilibrée en utilisant des techniques de Calcul Formel. Ceci revient en particulier à déterminer si un système d'équations linéaires (dont la matrice dépend de plusieurs paramètres qu'on appellera *noeuds*) a une unique solution. Soit $a_0,\ldots,a_r$ les indéterminées et $P_r(x)$ le polynôme générique de degré r

$$P_r(x) = a_r x^r + \ldots + a_0.$$

Alors, une matrice d'incidence $\mathcal{E}$ ayant n rangées et r colonnes est *équilibrée* si pour tout ensemble $\chi = \{x_1,\ldots,x_n\}$ de nombres réels tels que $x_1 < \ldots < x_n$ et pour toute matrice de nombres réels $\mathcal{F}$ (ayant n rangées et $r+1$ colonnes) le système d'équations :

$$P_r^{(j)}(x_i) = f_{i,j} \quad \text{ssi} \quad e_{i,j} = 1$$

a une unique solution. Dans la suite, on note $\mathcal{M}_{\mathcal{E}}$ la matrice associée au système linéaire qui caractérise le polynôme d'interpolation de χ et $\mathcal{E}$.
La proposition ci-dessous montre comment on réduit le problème à l'étude des matrices d'incidence normales : les matrices $\mathcal{M}_{\mathcal{E}}$ à étudier sont toujours des matrices carrées.

Proposition 4.1 *Soit $\mathcal{E}$ une matrice d'incidence équilibrée ayant n rangées et $r+1$ colonnes. Alors, $\mathcal{E}$ est normale, c'est-à-dire que*

$$|\mathcal{E}| = r + 1.$$

Preuve : La matrice d'incidence est équilibrée si pour tout ensemble $\chi = \{x_1, \ldots, x_n\}$ de nombres réels tels que $x_1 < \ldots < x_n$ et pour toute matrice de nombres réels $\mathcal{F}$ (ayant n rangées et $r+1$ colonnes) le système d'équations linéaires :

$$P_r^{(j)}(x_i) = f_{i,j} \quad \text{ssi} \quad e_{i,j} = 1$$

a une unique solution. La matrice associée $\mathcal{M}_{\mathcal{E}}$ à ce système a $|\mathcal{E}|$ rangées et $r+1$ colonnes. On en déduit que $\mathcal{E}$ est normale puisque l'application linéaire associée à $\mathcal{M}_{\mathcal{E}}$ est bijective.

■

Exemple 4.1 *Soit $\mathcal{E}$ la matrice d'incidence normale définie par :*

$$\mathcal{E} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Alors la matrice $\mathcal{M}_{\mathcal{E}}$ associée à $\mathcal{E}$ est

$$\mathcal{M}_{\mathcal{E}} = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 0 & 1 & 2x_2 & 3x_2^2 & 4x_2^3 & 5x_2^4 \\ 0 & 1 & 2x_3 & 3x_3^2 & 4x_3^3 & 5x_3^4 \\ 0 & 0 & 2 & 6x_2 & 12x_2^2 & 20x_2^3 \\ 0 & 0 & 0 & 6 & 24x_1 & 60x_1^2 \end{pmatrix}$$

Exemple 4.2 *Soit $\mathcal{E}_{\mathcal{L}}$ la matrice d'incidence normale correspondante à la Formule d'Interpolation de Lagrange*

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix}$$

Alors la matrice d'incidence $\mathcal{M}_{\mathcal{E}_{\mathcal{L}}}$ associée à $\mathcal{E}_{\mathcal{L}}$ est

$$\mathcal{M}_{\mathcal{E}} = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} \end{pmatrix}$$

Soit $\mathcal{E}_{\mathcal{H}}$ la matrice d'incidence normale correspondante au Problème d'Interpolation de Hermite associé à $\nu_1, \ldots, \nu_n$ ($N = \nu_1 + \ldots + \nu_n$). Alors la matrice $\mathcal{M}_{\mathcal{E}_{\mathcal{H}}}$ a la structure suivante :

$$\mathcal{M}_{\mathcal{E}_{\mathcal{H}}} = \begin{pmatrix} \mathcal{P}_1 \\ \mathcal{P}_2 \\ \vdots \\ \mathcal{P}_n \end{pmatrix} \mathcal{P}_j = \left(\frac{\partial^k}{\partial x_j} \left[\begin{array}{cccccc} 1 & x_j & x_j^2 & x_j^3 & \ldots & x_j^{N-1} \end{array} \right] \right)_{0 \le k \le \nu_k - 1}$$

La proposition réduit le problème de déterminer si une matrice d'incidence normale est équilibrée à un problème d'élimination des quantificateurs sur les réels.

Proposition 4.2 *Soit $\mathcal{E}$ une matrice d'incidence normale. Alors $\mathcal{E}$ est équilibrée si et seulement si le déterminant de $\mathcal{M}_{\mathcal{E}}$ ne s'annule pas pour tout ensemble de nombres réels $\chi = \{x_1, \ldots, x_n\}$ tels que $x_1 < \ldots < x_n$.*

Preuve : Si $\mathcal{E}$ a n rangées et $r+1$ colonnes alors $\mathcal{E}$ est équilibrée si et seulement si le système d'équations linéaires

$$\mathcal{M}_{\mathcal{E}} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_r \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_r \end{pmatrix}$$

a une unique solution pour tout choix de $\chi = \{x_1, \ldots, x_n\}$ avec $x_1 < \ldots < x_n$ et pour tout choix de $\mathcal{F} = \{f_0, \ldots, f_r\}$. Ceci est clairement équivalent à la non annulation du déterminant de $\mathcal{M}_{\mathcal{E}}$ pour tout ensemble de nombre réels $\chi = \{x_1, \ldots, x_n\}$ tels que $x_1 < \ldots < x_n$. ■

Pour une matrice d'incidence normale $\mathcal{E}$ donnée, on note $\mathcal{D}_{\mathcal{E}}$ le déterminant de $\mathcal{M}_{\mathcal{E}}$. D'après la proposition précédente, déterminer si une matrice d'incidence normale $\mathcal{E}$ est équilibrée est un problème qui se réduit à trouver un ensemble de nombres réels $\chi = \{x_1, \ldots, x_n\}$ tels que :

$$x_1 < \ldots < x_n \quad \text{et} \quad \mathcal{D}_{\mathcal{E}}(x_1, \ldots, x_n) = 0.$$

4.2 Simplification du problème

Dans cette section, la structure de la matrice $\mathcal{M}_{\mathcal{E}}$ est étudiée pour obtenir plus d'information sur le polynôme $\mathcal{D}_{\mathcal{E}}$.

Proposition 4.3 *Soit $\mathcal{E}$ une matrice d'incidence ayant n rangées et $r+1$ colonnes, qui n'est pas nécessairement normale. Alors les équations du système linéaire associé au Problème d'Interpolation de Birkhoff pour $\mathcal{E}$ peuvent être ordonnées de manière à ce que la matrice $\mathcal{M}_{\mathcal{E}}$ ait la structure suivante :*

$$\mathcal{M}_{\mathcal{E}} = \begin{pmatrix} B_0 \\ B_1 \\ \vdots \\ B_r \end{pmatrix}$$

où chaque sous-matrice B_k s'écrit comme :

$$B_k = \begin{pmatrix} 0 & \ldots & k!\binom{k}{k} & k!\binom{k+1}{k} x_{i(1,k)} & \ldots & k!\binom{r}{k} x_{i(1,k)}^{r-k} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & k!\binom{k}{k} & k!\binom{k+1}{k} x_{i(e_k,k)} & \ldots & k!\binom{r}{k} x_{i(e_k,k)}^{r-k} \end{pmatrix}$$

où e_k est le nombre de 1 dans la k-ième colonne de $\mathcal{E}$ et $i(u,k)$ est l'élément de $\{1,\ldots,n\}$ tel que le u-ième 1 dans la k-ième colonne de $\mathcal{E}$ soit à la $i(u,k)$-ième ligne de $\mathcal{E}$.

Preuve : Il est suffisant de regarder les équations du système linéaire considéré en regardant les 1 dans $\mathcal{E}$ par colonnes, c'est-à-dire, d'abord les équations correspondantes à P pusi celles de $P^{(1)}$, et ainsi de suite... ∎

Dans la suite, nous montrons comment ramener ce problème d'élimination des quantificateurs à un problème plus algébrique de théorie existentielle des réels.

Proposition 4.4 *Soit $\mathcal{E}$ une matrice d'incidence normale ayant n rangées et $r+1$ colonnes. Si $\ell_{i,j}$ $(1 \leq i < j \leq n)$ est le nombre de colonnes dans $\mathcal{E}$ telles que*

$$\mathcal{E}_{i,k} = 1, \quad \text{et} \quad \mathcal{E}_{j,k} = 1$$

alors $(x_i - x_j)^{\ell_{i,j}}$ divise le polynôme $\mathcal{D}_{\mathcal{E}}$.

Preuve : D'après la proposition 4.3, pour chaque triplet (i,j,k) tel que $1 \leq i < j \leq n$ et $0 \leq k \leq n$ vérifiant

$$\mathcal{E}_{i,k} = 1, \quad \text{et} \quad \mathcal{E}_{j,k} = 1$$

il existe dans $\mathcal{M}_{\mathcal{E}}$ deux colonnes correspondantes à $P^{(k)}(x_i)$ et $P^{(k)}(x_j)$. En soustrayant la première à la deuxième, il est alors clair que $(x_i - x_j)$ divise $\mathcal{D}_{\mathcal{E}}$. On peut appliquer ceci $\ell_{i,j}$ fois, et donc $(x_i - x_j)^{\ell_{i,j}}$ divise $\mathcal{D}_{\mathcal{E}}$. ∎

La proposition ci-dessus montre que l'on peut simplifier le polynôme $\mathcal{D}_{\mathcal{E}}$ en le divisant par $(x_i - x_j)^{\ell_{i,j}}$, mais comme le montre l'exemple ci-dessous $\ell_{i,j}$ n'est pas la puissance maximale de $(x_i - x_j)$ divisant $\mathcal{D}_{\mathcal{E}}$.

Exemple 4.3 *Soit $\mathcal{E}$ la matrice d'incidence de l'exemple 4.1. Dans cet exemple le polynôme $\mathcal{D}_{\mathcal{E}}$ se factorise de la manière suivante :*

$$\mathcal{D}_{\mathcal{E}} = -36(x_2 - x_3)^2(x_1 - x_2)^4(6x_1^2 - 12x_1x_3 - x_2^2 + 2x_2x_3 + 5x_3^2).$$

Or, dans ce cas $\ell_{1,2} = 1$ et $\ell_{2,4} = 1$.

Ceci nous amène à introduire l'entier $L_{i,j}$ comme étant la plus grande puissance de $(x_i - x_j)$ qui divise $\mathcal{D}_{\mathcal{E}}$.

Définition 4.3 *Soit $\mathcal{E}$ une matrice d'incidence normale. L'indicateur d'équilibre de $\mathcal{E}$ est le polynôme à coefficients entiers :*

$$\widetilde{\mathcal{D}}_{\mathcal{E}} = \frac{\mathcal{D}_{\mathcal{E}}}{\prod_{1 \leq i < j \leq n}(x_j - x_i)^{L_{i,j}}}$$

On peut alors travailler avec l'*indicateur d'équilibre* d'une matrice d'incidence normale $\mathcal{E}$ pour étudier si elle est équilibrée. Revenons à l'exemple précédent. L'*indicateur d'équilibre* de $\mathcal{E}$ est alors :

$$\widetilde{\mathcal{D}}_{\mathcal{E}} = -36(6x_1^2 - 12x_1x_3 - x_2^2 + 2x_2x_3 + 5x_3^2).$$

Il est facile de voir qu'on a

$$\widetilde{\mathcal{D}}_{\mathcal{E}} = -36(6(x_1 - x_3)^2 - (x_2 - x_3)^2)$$

En effectuant la substitution :

$$x_2 - x_1 = t_1^2 \quad x_3 - x_2 = t_2^2 \Longrightarrow x_3 - x_1 = t_1^2 + t_2^2$$

on obtient

$$\widetilde{\mathcal{D}}_{\mathcal{E}}(t_1,t_2) = -36(5t_1^4 + 12t_1^2t_2^2 + 6t_2^4)$$

ce qui nous permet alors de conclure que $\mathcal{E}$ est équilibrée puisque pour tout $x_1 < x_2 < x_3$ le polynôme $\widetilde{\mathcal{D}}_{\mathcal{E}}$ est strictement négatif (ce qui implique que $\mathcal{D}_{\mathcal{E}}$ est strictement négatif).

Exemple 4.4 *Les déterminants des matrices $\mathcal{M}_{\mathcal{E}_\mathcal{L}}$ (issus de l'Interpolation de Lagrange) et des matrices $\mathcal{M}_{\mathcal{E}_\mathcal{H}}$ (issus de l'Interpolation de Hermite) sont*

$$\mathcal{D}_{\mathcal{E}_\mathcal{L}} = \prod_{i<j}(x_j - x_i) \quad \mathcal{D}_{\mathcal{E}_\mathcal{H}} = \prod_{k=1}^{n} \prod_{\lambda=0}^{\nu_k - 1} \lambda! \prod_{i<j}(x_j - x_i)^{\nu_j \nu_i}$$

Les indicateurs d'équilibre correspondants sont égaux à des nombres entiers non nuls.

Pour déterminer si une matrice d'incidence normale est équilibrée, on va suivre la méthode appliquée pour résoudre l'exemple 4.3.

4.3 Stratégies de résolution

4.3.1 Premiers ingrédients

Proposition 4.5 *Soit $\mathcal{E}$ une matrice d'incidence normale ayant n rangées et $r+1$ colonnes. Soit $t_1,\ldots,t_{n-1}$ de nouvelles variables. Le polynôme*

$$\mathcal{H}_{\mathcal{E}} = \widetilde{\mathcal{D}}_{\mathcal{E}}(x_1, x_1 + t_1^2, x_1 + t_1^2 + t_2^2, \ldots, x_1 + \sum_{i=1}^{n-1} t_i^2)$$

est un polynôme dans $Z[t_1,\ldots,t_{n-1}]$.

Preuve : On raisonne par l'absurde. Soit $d > 0$ le degré de $\mathcal{H}_{\mathcal{E}}$ en x_1. On peut alors écrire $\mathcal{H}_{\mathcal{E}}$ sous la forme :

$$\mathcal{H}_{\mathcal{E}} = a_d(t_1,\ldots,t_{n-1})x_1^d + \ldots + a_1(t_1,\ldots,t_{n-1})x_1 + a_0(t_1,\ldots,t_{n-1})$$

où les a_i sont des polynômes à coefficients entiers et a_d est non identiquement nul. Ce dernier point implique l'existence d'un $(n-1)$-uplet $(\tau_1,\ldots,\tau_{n-1}) \in C^{n-1}$ tel que $a_d(\tau_1,\ldots,\tau_{n-1}) \neq 0$ et l'existence d'un complexe $\alpha \in C$ tel que

$$\mathcal{H}_{\mathcal{E}}(\tau_1,\ldots,\tau_{n-1},\alpha) = 0.$$

Donc, il existe $\beta \in C$, tel que $\mathcal{H}_{\mathcal{E}}(\tau_1,\ldots,\tau_{n-1},\alpha+\beta) \neq 0$ Alors en adaptant les énoncés au cas complexe, le Problème d'Interpolation de Birkhoff a une unique solution $Q(x)$ pour $\mathcal{E}$ et les noeuds :

$$\alpha+\beta,\alpha+\beta+t_1^2,\ldots,\alpha+\beta+\sum_{i=1}^{n} t_i^2.$$

Or, ceci implique que $P(x) = Q(x+\beta)$ est l'unique solution du Problème d'Interpolation de Birkhoff pour $\mathcal{E}$ et les noeuds :

$$\alpha,\alpha+t_1^2,\ldots,\alpha+\sum_{i=1}^{n} t_i^2.$$

et que $\mathcal{H}_{\mathcal{E}}(\tau_1,\ldots,\tau_{n-1},\alpha) \neq 0$. ■

Proposition 4.6 *Soit $\mathcal{E}$ une matrice d'incidence normale ayant n rangées et $r+1$ colonnes. Alors le polynôme $\mathcal{H}_{\mathcal{E}}$ est homogène et son degré est strictement borné par $2nr$.*

Preuve : Soit λ une nouvelle variable. Alors

$$\mathcal{H}_{\mathcal{E}}(\lambda t_1,\ldots,\lambda t_{n-1}) = \widetilde{\mathcal{D}}_{\mathcal{E}}(0,\lambda^2 t_1^2,\lambda^2(t_1^2+t_2)^2,\ldots,\lambda^2\sum_{i=1}^{n-1} t_i^2) = \sum_{j=0}^{r} a_j(t_1,\ldots,t_{n-1})\lambda^j = R(t_1,\ldots,t_{n-1},\lambda)$$

Si $\mathcal{H}_{\mathcal{E}}$ est nul ou constant, il n'y a rien à prouver. On montre dans la suite qu'il existe au moins un a_j qui est non identiquement nul. Si il existe $a_i(t_1,\ldots,t_{n-1})$ et $a_k(t_1,\ldots,t_{n-1})$ qui sont différents de zéro, alors il existe un $(n-1)$-uplet $(\alpha_1,\ldots,\alpha_{n-1}) \in R^{n-1}$ tel que :

$$\mathcal{H}_{\mathcal{E}}(\alpha_1,\ldots,\alpha_{n-1}).a_i(\alpha_1,\ldots,\alpha_{n-1}).a_k(\alpha_1,\ldots,\alpha_{n-1}) \neq 0.$$

Alors le polynôme $R(\alpha_1,\ldots,\alpha_{n-1},\lambda)$ a une solution λ_0 non nulle et non nécessairement réelle. Puisque $\mathcal{H}_{\mathcal{E}}(\alpha_1,\ldots,\alpha_{n-1}) \neq 0$, le Problème d'Interpolation de Birkhoff pour $\mathcal{E}$ a une unique solution pour les noeuds

$$\chi_1 = \{0,t_1^2,\ldots,\sum_{i=1}^{n-1} t_i^2\}$$

qui est aussi une solution du Problème d'Interpolation de Birkhoff pour $\mathcal{E}$ et les noeuds :

$$\chi_2 = \{0,\lambda_0^2 t_1^2,\ldots,\lambda_0^2\sum_{i=1}^{n-1} t_i^2\}$$

puisque si $P(x)$ résoud l'interpolation en $(\chi_1,\mathcal{E})$ et $\mathcal{F}$ alors $P(x/\lambda_0^2)$ résoud l'interpolation en $(\chi_2,\mathcal{E})$ et $\mathcal{F}$. Ce dernier point implique que $R(\alpha_1,\ldots,\alpha_{n-1},\lambda_0) \neq 0$, ce qui est absurde.

On a donc prouvé qu'il existe un entier m tel que

$$\mathcal{H}_{\mathcal{E}}(\lambda t_1, \ldots, \lambda t_{n-1}) = \lambda^m R(t_1, \ldots, t_{n-1})$$

ce qui implique que $\mathcal{H}_{\mathcal{E}}$ est homogène.

On reprend les notations introduites dans la proposition 4.3, il est clair que le degré de $\mathcal{D}_{\mathcal{E}}$ est borné par

$$\sum_{k=0}^{r-1}(r-k)e_k.$$

Or, la somme des e_k est égale à n, donc la borne ci-dessus est égale à

$$nr - \sum_{k=0}^{r-1} ke_k$$

qui est strictement inférieure à nr. ■

Corollaire 4.1 *Soit $\mathcal{E}$ une matrice d'incidence normale. Le polynôme $\mathcal{H}_{\mathcal{E}}$ a des racines réelles $(t_1, \ldots, t_{n-1})$ telles que $t_1 \ldots t_{n-1} \neq 0$ si et seulement si la matrice $\mathcal{E}$ est équilibrée.*

Preuve : Si $\mathcal{E}$ n'est pas équilibrée alors alors il existe $\chi = \{x_1, \ldots, x_n\} \subset R^n$ tel que $x_1 < \ldots < x_n$ et $\widetilde{\mathcal{D}}_{\mathcal{E}}(x_1, \ldots, x_n) = 0$. En posant $t_i = (x_{i+1} - x_i)^{1/2}$, on a $(t_1, \ldots, t_{n-1})$ qui est une racine réelle de $\mathcal{H}_{\mathcal{E}}$ avec bien sur pour tout i $t_i \neq 0$.

Réciproquement, soit $(t_1, \ldots, t_{n-1})$ une racine réelle de $\mathcal{H}_{\mathcal{E}}$ dont aucune des coordonnées n'est nulle. Alors, en posant $x_1 = 0$ et $x_{i+1} = x_i + t_i^2$, on a $\widetilde{\mathcal{D}}_{\mathcal{E}}(x_1, \ldots, x_n) = 0$ et $\mathcal{E}$ n'est pas équilibrée. ■

Puisque $\mathcal{H}$ est homogène et puisque on en cherche des solutions réelles dont aucune des coordonnées n'est nulle, le résultat suivant est immédiat.

Corollaire 4.2 *Soit $\mathcal{E}$ une matrice d'incidence normale. Le polynôme $\mathcal{H}_{\mathcal{E}}$ a des racines réelles de la forme $(1, t_2, \ldots, t_{n-1})$ telles que $t_2 \ldots t_{n-1} \neq 0$ si et seulement si la matrice $\mathcal{E}$ est équilibrée.*

Ainsi, si on fixe n et r la résolution du Problème d'Interpolation de Birkhoff est équivalente à décider si des hypersurfaces contiennent des points réels dont aucune des coordonnées réelles. Ceci montre d'une part que le Problème d'Interpolation de Birkhoff pour n et r fixés est *décidable* et permet de donner des bornes de complexité pour ce problème.

4.3.2 Résultats supplémentaires

On va donc travailler avec n et r fixés. On note :

- $\mathcal{IN}_{n,r}$ l'ensemble des matrices d'incidence normale ayant n rangées et $r+1$ colonnes,
- $\mathcal{P}_{n,r}$ l'ensemble des matrices d'incidence normale équilibrées ayant n rangées et $r+1$ colonnes,

– S_n le n-ième groupe symétrique.

Si $\sigma \in S_n$ et $\mathcal{E} \in \mathcal{IN}_{n,r}$, alors $\sigma(\mathcal{E})$ représentera la matrice d'incidence normale obtenue en permutant les rangées de $\mathcal{E}$ selon σ. Une première remarque importante est que l'image d'une matrice d'incidence normale équilibrée par σ n'est pas forcément équilibrée.

Définition 4.4 *Deux matrices équilibrées $\mathcal{E}_1$ et $\mathcal{E}_2$ sont dites équivalentes si et seulement si il existe une permutation $\sigma \in S_n$ telle que $\sigma(\mathcal{E}_1) = \mathcal{E}_2$. On notera alors $\mathcal{E}_1 \equiv \mathcal{E}_2$.*

Il est aisé de montrer que $\equiv$ est une relation d'équivalence dans $\mathcal{P}_{n,r}$. L'ensemble des classes d'équivalence sera noté $\mathcal{P}_{n,r}/\equiv$. Dans [48], l'auteur énonce les trois résultats suivants :

Théorème 4.1 *Soit n un entier supérieur ou égal à 3. Le nombre de matrices d'incidence normale ayant n rangées et 3 colonnes qui sont équilibrées est égal à :*

$$m_{n,2} = \binom{n}{1} + 12\binom{n}{2} + 15\binom{n}{3}.$$

Proposition 4.7 *Soit n en entier supérieur ou égal à 1. Le nombre de matrices d'incidence normale ayant n rangées et 2 colonnes qui sont équilibrées est égale à :*

$$m_{n,1} = \binom{n}{1} + 3\binom{n}{2}.$$

Théorème 4.2 *Soit n et r deux entiers supérieurs ou égaux à 1. Pour toute classe d'équivalence $\mathcal{C}$ dans $\mathcal{P}_{r+1,r}/\equiv$ d'une matrice $\mathcal{E}$ d'incidence normale équilibrée, on note :*

- *$r_{\mathcal{C}}$ le nombre de rangées non nulles dans $\mathcal{E}$,*
- *$k_{\mathcal{C}}$ le nombre de matrices différentes dans $\mathcal{C}$ après en avoir effacé les rangées nulles.*

Alors, le nombre d'éléments de $\mathcal{P}_{n,r}$ est égal à :

$$m_{n,r} = \sum_{\mathcal{C} \in \mathcal{P}_{r+1,r}/\equiv} k_{\mathcal{C}}.\binom{n}{r_{\mathcal{C}}}.$$

4.4 Les algorithmes utilisés

Dans cette section, nous rappelons brièvement les algorithmes utilisés pour la résolution du Problème d'Interpolation de Birkhoff avec n et r fixés.

4.4.1 Etude de l'hypersurface

On va commencer par occulter le fait que on cherche à déterminer si une hypersurface admet des points réels dont aucune des coordonnées n'est nulle pour étudier directement ces hypersurfaces, sans prendre en compte, dans un premier temps, cette spécificité du problème. Pour étudier les hypersurfaces nous avons utilisé l'algorithme décrit dans le Chapitre 2 que nous rappelons ci-dessous.

Algorithm 3

- **Input :** Un système S d'équations polynomiales dans $K[X_1,\ldots,X_n]$.
- **Output :** Une liste de systèmes zéro-dimensionels tel que l'ensemble de leurs solutions est inclus dans $V(S)$ et contient au moins un point par composante semi-algébriquement connexe de $V(S)\bigcap R^n$.

1. list := LexPrimeDecomposition(S), result := [],
2. Choisir $A \notin V(S)$.
3. Tant que list $\neq \emptyset$ faire
   - S := first(list), et enlever S de list, poser $d = \mathrm{Dim}(S)$,
   - Si $d = 0$ alors result := result $\bigcup S$,
   - Sinon
     - $\mathcal{T} = ExtractTriangular(S)$.
     - (*) $Q = \Gamma_A(\mathcal{T}) \bigcup S$ et poser $u = \mathrm{Dim}(Q)$
     - Si $u = d$ choisir un autre point $A \notin S$ et aller au pas (*).
     - $d := u$; list := list $\bigcup$ LexPrimeDecomposition(Q),
4. Retourner result.

Notons que nous obtenons en sortie une liste de Représentations Univariées Rationnelles

$$(f(T),g_0(T),g_1(T),\ldots,g_n(T)).$$

Ainsi, pour détecter si une racine réelle trouvée a au moins une de ses coordonnées nulles, il suffit de calculer le $\gcd(f(T),g_i(T))$ pour $i \in \{1,\ldots,n\}$.

Trois cas peuvent alors se produire :

- l'hypersurface est vide du point de vue réel et on peut conclure que la matrice d'incidence normale associée au polynôme définissant cette hypersurface est équilibrée,
- l'hypersurface contient au moins un point réel dont toutes les coordonnées sont non nulles et on peut conclure que la matrice d'incidence normale associée au polynôme définissant cette hypersurface n'est pas équilibrée,
- on n'a trouvé sur l'hypersurface que des points réels ayant au moins une de ses coordonnées nulles; dans ce cas on ne peut rien conclure a priori.

Ce dernier cas fait l'objet des deux paragraphes suivants.

4.4.2 Racines à coordonnées nulles : premier cas

Soit une hypersurface définie par $P = 0$. Dans cette section, on suppose que :

- tous les points réels trouvés sur cette hypersurface ont au moins une coordonnée nulle,
- parmi ces points, il en existe un (que l'on appelle M) tel que le vecteur gradient en M de $V(P)$, $\overrightarrow{grad}_M\,(P)$, est non nul.

On va montrer le résultat suivant :

Lemme 4.1 *Soit $P \in \mathbb{R}[t_1,\ldots,t_n]$ et $V(P)$ l'hypersurface définie par $P = 0$. Soit $M =$*

$(\mu_1,\dots,\mu_n) \in V(P)\cap \mathbb{R}^n$ tel que $\overrightarrow{grad}_M\ (P) = (g_1,\dots,g_n) \neq \overrightarrow{0}$, et $I = \{i_1,\dots,i_k\} \subset \{1,\dots,n\}$ l'ensemble des indices pour lesquels μ_i est nul.

Si $\overrightarrow{grad}_M\ (P)$ n'est co-linéaire à aucun des axes de coordonnées $(t_i)_{i\in I}$, alors il existe un point M' dans $V(P)\cap \mathbb{R}^n$ dont aucune des coordonnées n'est nulle.

Preuve :

- On note $\overrightarrow{e_i}$ le vecteur dont toutes les coordonnées sont non nulles sauf la i-ième. On suppose dans un premier temps que $I = \{j\}$ (la j-ième coordonnée de M est nulle et c'est la seule). Puisque $\overrightarrow{grad}_M\ (P)$ et le vecteur $\overrightarrow{e_j}$ ne sont pas parallèles, et puisque $\overrightarrow{grad}_M\ (P) \neq \overrightarrow{0}$, d'après [34] p. 59, il existe
  - un vecteur $\overrightarrow{u}$ de l'espace tangent à $V(P)\cap\mathbb{R}^n$ tel que $\overrightarrow{u}\ .\ \overrightarrow{e_j} = u_j \neq 0$
  - et $\gamma \subset V(P)\cap\mathbb{R}^n$ une courbe passant par M et dont le vecteur tangent en M est $\overrightarrow{u}$.

  Dans un voisinage de M, tout point $N = (x_1,\dots,x_n)$ de γ s'écrit :

  $$\forall i \in \{1,\dots,n\} \quad x_i = \mu_i + \lambda u_i + o(\lambda)u_i.$$

  Donc, il existe un rationnel λ_0 suffisamment petit tel que $x_j = \lambda_0 u_j + o(\lambda_0)u_j$ soit non nul. Le point N correspondant a sa j-ième coordonnée non nulle.
- On conclut de manière analogue en choisissant $\overrightarrow{u}$ tel que $\forall i \in I \quad \overrightarrow{u}\ .\ \overrightarrow{e_i} = u_i \neq 0$, ce qui est possible car $\overrightarrow{grad}_M\ (P)$ n'est co-linéaire à aucun des vecteurs $\overrightarrow{e_i}$ (pour $i \in I$) et qu'il est non nul.

■

On se donne une Représentation Univariée Rationnelle

$$(f(T),g_0(T),g_1(T),\dots,g_n(T))$$

Il est possible de savoir si un des points définis à des coordonnées nulles en calculant le pgcd de f et de chacun des g_i. Notons que ce calcul peut être fait, dans un premier temps, modulo un nombre premier. Dans ce cas, si il est non nul, on sait que ce pgcd est non nul sur les entiers. Si il est nul, on refait le calcul sur les entiers.

Si ce pgcd est non trivial, on obtient un polynôme univarié dont on va tout de suite vérifier si il a des racines réelles. Si il n'en a pas, cela veut dire que la RUR ne définit pas de points réels ayant au moins une coordonnée nulle. Si il en a, il faut déterminer quelles sont les coordonnées nulles des vecteurs gradients en ces points.

Pour cela, on homogénéise les polynômes représentant ces coordonnées. Puis, on remplace la variable d'homogénéisation par g_0 et chacun des X_i par g_i. On obtient un polynôme univarié dont on peut alors calculer le pgcd avec f.

4.4.3 Racines à coordonnées nulles : deuxième cas

Cette section traite du cas où les hypothèses du lemme précédent ne sont pas vérifiées. Deux stratégies sont applicables :

- ou bien on applique les stratégies basées sur les déformations infinitésimales inspirées du Chapitre??,
- ou bien on introduit l'équation $TX_1 \dots X_n - 1 = 0$ (où T est une nouvelle variable) dans les systèmes d'équations que nous allons étudier.

4.5 La résolution en pratique : le cas $r = 4$, $n = 5$

Le traitement du cas $r = 4$, $n = 5$ nécessite l'étude de 53130 hypersurfaces. Sur les 53130 hypersurfaces, 42925 d'entre elles sont définies par un polynôme constant. Dans le but de réduire le nombre d'appels aux algorithmes **HA3** et **SA2** toutes les variables (sauf une) des polynômes sont spécialisées en des valeurs non nulles afin d'obtenir des polynômes univariés dont on teste si ils admettent une racine réelle différente de 0 (ces tests sont effectués par l'implantation de l'algorithme d'Uspensky dans RS, F. Rouillier).

Nous avons utilisé les implantations des algorithmes **HA3** et **SA2** faisant appel aux logiciels Gb, AGb (dédiés au calcul de base de Gröbner, écrit par J.-C. Faugère) et RS (dédié au calcul de Représentations Univariées Rationnelles et au comptage des racines réelles de polynômes univariés, écrit par F. Rouillier).

Ainsi, il n'a fallu au total qu'un millier d'appels aux algorithmes **HA3** et **SA2** pour traiter ce cas. Parmi les hypersurfaces étudiées, seules 102 d'entre elles contiennent une infinité de points singuliers. Nous en donnons la liste ci-dessous. Pour toutes les hypersurfaces contenant au plus un nombre fini de points singuliers, les algorithmes **HA3** ou **SA2** ont permis de montrer, soit que leur lieu réel est vide, soit qu'il contient au moins un point dont aucune des coordonnées n'est nulle.

```
[1+2*t2^2-2*t3^2-2*t4^2+t2^4-2*t2^2*t3^2-2*t2^2*t4^2+3*t3^4+6*t3^2*t4^2+3*t4^4,
3+6*t2^2-2*t3^2-2*t4^2+3*t2^4-2*t2^2*t3^2-2*t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-1-2*t2^2+t3^2+t4^2-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
2*t3^2+2*t2^2+1+t2^4+2*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2-2*t4^2+t3^4+3*t4^4,
6*t2^2+3*t2^4+6*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+3*t3^4+t4^4+6*t3^2-2*t4^2+3,
-2*t2^2-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4-2*t3^2+t4^2-1,
-t3^6-t4^6-3*t3^2*t4^4-3*t3^4*t4^2-t2^2*t4^4-t2^2*t3^4+t2^4*t4^2+t2^4*t3^2-3*t2^2-3*t2^4+t3^2+t4^2-t3^4-
t4^4-1-t2^6-2*t2^2*t3^2*t4^2-2*t3^2*t4^2+2*t2^2*t4^2+2*t2^2*t3^2,
-t3^6-t4^6-1-t3^2*t4^4+t3^4*t4^2-t2^2*t4^4-3*t2^2*t3^4+t2^4*t4^2-3*t2^4*t3^2-t2^6+2*t2^2*t3^2*t4^2-
3*t2^4-3*t3^2-3*t2^2-t4^4+t4^2-3*t3^4+2*t2^2*t4^2-6*t2^2*t3^2+2*t3^2*t4^2,
-t3^6-t4^6-9*t2^4-3-9*t2^2-3*t3^2*t4^4-3*t3^4*t4^2-3*t2^2*t4^4-3*t2^2*t3^4+3*t2^4*t4^2+3*t2^4*t3^2-
3*t4^4-3*t3^4+3*t4^2+3*t3^2-3*t2^6-6*t2^2*t3^2*t4^2-6*t3^2*t4^2+6*t2^2*t4^2+6*t2^2*t3^2,
-3*t3^6-t4^6-3*t3^2*t4^4+3*t3^4*t4^2-3*t2^2*t4^4-9*t2^2*t3^4+3*t2^4*t4^2-9*t2^4*t3^2-3-9*t2^2-9*t3^2-
9*t2^4-9*t3^4-18*t2^2*t3^2+6*t2^2*t4^2+6*t3^2*t4^2-3*t2^6+3*t4^2-3*t4^4+6*t2^2*t3^2*t4^2,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
-6*t2^2*t3^2*t4^2-t2^8-2*t2^6-t2^4+2*t2^6*t4^2+2*t2^6*t3^2-t4^4-t3^4-3*t2^4*t4^4-3*t2^4*t3^4-3*t2^2*t4^4-
3*t2^2*t3^4-6*t2^4*t3^2*t4^2+3*t2^4*t4^2+3*t2^4*t3^2-2*t3^2*t4^2+t2^2*t4^2+t2^2*t3^2,
-2*t3^6-3*t3^2*t4^4+3*t3^4*t4^2-3*t2^2*t4^4-6*t2^2*t3^4+3*t2^4*t4^2-6*t2^4*t3^2-2*t2^6-t2^4-2*t2^2*t3^2+
t2^2*t4^2+t3^2*t4^2-t2^8-t3^4-t4^4+6*t2^2*t3^2*t4^2+2*t2^6*t4^2-4*t2^6*t3^2-3*t2^4*t4^4-6*t2^4*t3^4+
6*t2^4*t3^2*t4^2-t3^8-6*t2^2*t3^2*t4^4-3*t3^4*t4^4+2*t3^6*t4^2+6*t2^2*t3^4*t4^2-4*t2^2*t3^6,
2*t2^2*t3^2*t4^2-t2^8+2*t2^6-t3^4-3*t2^4+t2^2*t4^4+3*t2^4*t4^2-t4^4-3*t2^2*t4^2-2*t3^2*t4^2-2*t2^6*t4^2-
2*t2^6*t3^2-t2^4*t4^4-t2^4*t3^4+t2^2*t3^4+3*t2^4*t3^2-3*t2^2*t3^2-2*t2^4*t3^2*t4^2,
2*t3^6+t3^2*t4^4+3*t3^4*t4^2+t2^2*t4^4+6*t2^2*t3^4+3*t2^4*t4^2+6*t2^4*t3^2+2*t2^6-3*t2^4-6*t2^2*t3^2-
3*t2^2*t4^2-3*t3^2*t4^2-t2^8-3*t3^4-t4^4+6*t2^2*t3^2*t4^2-2*t2^6*t4^2-4*t2^6*t3^2-t2^4*t4^4-6*t2^4*t3^4-
```

```
6*t2^4*t3^2*t4^2-t3^8-2*t2^2*t3^2*t4^4-t3^4*t4^4-2*t3^6*t4^2-6*t2^2*t3^4*t4^2-4*t2^2*t3^6,
-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
-1-2*t2^2+t3^2+t4^2-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
-t4^4+t3^2*t4^2+3*t3^4*t4^2-2*t2^2*t4^4+2*t2^2*t3^2*t4^2-t3^4-2*t3^6-2*t2^2*t3^4-3*t3^2*t4^4-t2^4*t4^4-
t2^4*t3^4+t2^4*t3^2*t4^2-t3^8-3*t2^2*t3^2*t4^4-3*t3^4*t4^4+2*t3^6*t4^2+3*t2^2*t3^4*t4^2-2*t2^2*t3^6,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-3*t3^2*t4^2+3*t3^4*t4^2-t4^4+2*t3^6-2*t2^2*t4^4+t3^2*t4^4-6*t2^2*t3^2*t4^2-t2^4*t4^4-3*t2^4*t3^4-
3*t2^4*t3^2*t4^2-t3^8+t2^2*t3^2*t4^4-t3^4*t4^4-2*t3^6*t4^2+3*t2^2*t3^4*t4^2+2*t2^2*t3^6-6*t2^2*t3^4-3*t3^4,
-t3^6-3-4*t3^4*t4^2+t2^2*t3^4-4*t2^4*t4^2-t2^4*t3^2-3*t2^6+t3^4-9*t2^4-t3^2-9*t2^2-4*t4^2+4*t2^2*t3^2*t4^2-
2*t2^2*t3^2-8*t2^2*t4^2+4*t3^2*t4^2,
t3^6+1+2*t3^4*t4^2-t2^2*t3^4+2*t2^4*t4^2+t2^4*t3^2+t2^6-t3^4+3*t2^4+t3^2+3*t2^2+2*t4^2-2*t2^2*t3^2*t4^2+
2*t2^2*t3^2+4*t2^2*t4^2-2*t3^2*t4^2,
-3*t3^6-1-4*t3^4*t4^2+3*t2^2*t3^4-4*t2^4*t4^2-3*t2^4*t3^2-t2^6+3*t3^4-3*t2^4-3*t3^2-3*t2^2-4*t4^2+
4*t2^2*t3^2*t4^2-6*t2^2*t3^2-8*t2^2*t4^2+4*t3^2*t4^2,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-1-2*t2^2+t3^2+t4^2-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
3+6*t2^2-2*t3^2-2*t4^2+3*t2^4-2*t2^2*t3^2-2*t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t3^6+t4^6+3*t3^2*t4^4+3*t3^4*t4^2-t2^2*t4^4-t2^2*t3^4+t2^4*t4^2+t2^4*t3^2+t2^6+2*t2^2*t4^2-2*t3^2*t4^2+
2*t2^2*t3^2-t3^4+t3^2+t4^2-2*t2^2*t3^2*t4^2-t4^4+3*t2^2+3*t2^4+1,
-3*t3^6-3*t4^6-9*t3^2*t4^4-9*t3^4*t4^2+3*t2^2*t4^4+3*t2^2*t3^4-3*t2^4*t4^2-3*t2^4*t3^2-t2^6-6*t2^2*t4^2+
6*t3^2*t4^2-6*t2^2*t3^2+3*t3^4-3*t3^2-3*t4^2+6*t2^2*t3^2*t4^2+3*t4^4-3*t2^2-3*t2^4-1,
1+2*t2^2-2*t3^2-2*t4^2+t2^4-2*t2^2*t3^2-2*t2^2*t4^2+3*t3^4+6*t3^2*t4^2+3*t4^4,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
-2*t2^2-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4-2*t3^2+t4^2-1,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
-2*t2^2-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4-2*t3^2+t4^2-1,
6*t2^2+3*t2^4+6*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+3*t3^4+t4^4+6*t3^2-2*t4^2+3,
t3^6+t4^6+1-t3^2*t4^4+t3^4*t4^2-t2^2*t4^4+3*t2^2*t3^4+t2^4*t4^2+3*t2^4*t3^2+t2^6+2*t2^2*t3^2*t4^2+
3*t3^4+3*t2^4+3*t3^2+3*t2^2+t4^2-t4^4+6*t2^2*t3^2+2*t3^2*t4^2+2*t2^2*t4^2,
-t3^6-3*t4^6-1+3*t3^2*t4^4-3*t3^4*t4^2+3*t2^2*t4^4-3*t2^2*t3^4-3*t2^4*t4^2-3*t2^4*t3^2-t2^6-
6*t2^2*t3^2*t4^2-3*t3^4-3*t2^4-3*t3^2-3*t2^2-3*t4^2+3*t4^4-6*t2^2*t3^2-6*t3^2*t4^2-6*t2^2*t4^2,
2*t3^2+2*t2^2+1+t2^4+2*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2-2*t4^2+t3^4+3*t4^4,
3*t3^6+3*t4^6+9*t3^2*t4^4+9*t3^4*t4^2+t2^2*t4^4+t2^2*t3^4-t2^4*t4^2-t2^4*t3^2+t2^6+4*t4^4+4*t3^4+
4*t2^4+2*t2^2*t3^2*t4^2+8*t3^2*t4^2-4*t2^2*t4^2-4*t2^2*t3^2,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t3^6+3*t4^6+t3^2*t4^4-t3^4*t4^2+t2^2*t4^4+3*t2^2*t3^4-t2^4*t4^2+3*t2^4*t3^2+t2^6+4*t3^4+4*t4^4+4*t2^4-
2*t2^2*t3^2*t4^2+8*t2^2*t3^2-4*t2^2*t4^2-4*t3^2*t4^2,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t3^6+3*t4^6+t3^2*t4^4-t3^4*t4^2+4*t2^2*t4^4+4*t2^2*t3^4-4*t2^2*t3^2*t4^2+4*t3^4+4*t4^4-4*t3^2*t4^2,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
1+2*t2^2-t3^2-t4^2+t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
1+2*t3^2+2*t2^2+t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4-t4^2,
t3^6+t4^6+3*t3^2*t4^4+3*t3^4*t4^2+t2^2*t4^4+t2^2*t3^4-t2^4*t4^2-t2^4*t3^2+2*t4^4+2*t3^4+2*t2^4+t2^6+
2*t2^2*t3^2*t4^2+4*t3^2*t4^2-2*t2^2*t4^2-2*t2^2*t3^2,
t3^6+t4^6+t3^2*t4^4-t3^4*t4^2+t2^2*t4^4+3*t2^2*t3^4-t2^4*t4^2+3*t2^4*t3^2+2*t4^4+t2^6-2*t2^2*t3^2*t4^2+
2*t3^4+2*t2^4-2*t3^2*t4^2-2*t2^2*t4^2+4*t2^2*t3^2,
t3^6+t4^6+2*t3^4+t3^2*t4^4-t3^4*t4^2+2*t2^2*t4^4+2*t2^2*t3^4+2*t4^4-2*t3^2*t4^2-2*t2^2*t3^2*t4^2,
t3^6+t4^6+4*t4^4+4*t3^4+3*t3^2*t4^4+3*t3^4*t4^2+3*t2^2*t4^4+3*t2^2*t3^4-3*t2^4*t4^2-3*t2^4*t3^2+4*t2^4+
3*t2^6+8*t3^2*t4^2-4*t2^2*t4^2-4*t2^2*t3^2+6*t2^2*t3^2*t4^2,
3*t3^6+t4^6+3*t3^2*t4^4-3*t3^4*t4^2+3*t2^2*t4^4+9*t2^2*t3^4-3*t2^4*t4^2+9*t2^4*t3^2+4*t4^4+4*t3^4+
4*t2^4+3*t2^6-6*t2^2*t3^2*t4^2+8*t2^2*t3^2-4*t3^2*t4^2-4*t2^2*t4^2,
3*t3^6+t4^6+4*t3^4+4*t4^4+3*t3^2*t4^4-3*t3^4*t4^2+4*t2^2*t4^4+4*t2^2*t3^4-4*t2^2*t3^2*t4^2-4*t3^2*t4^2,
-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4,
t2^4-2*t2^2*t3^2-2*t2^2*t4^2+3*t3^4+6*t3^2*t4^2+3*t4^4,
3*t2^4-2*t2^2*t3^2-2*t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
t2^4+2*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+t3^4+3*t4^4,
3*t2^4+6*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+3*t3^4+t4^4,
-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4,
-t3^6-t4^6-3*t3^2*t4^4-3*t3^4*t4^2-t2^2*t4^4-t2^2*t3^4+t2^4*t4^2+t2^4*t3^2-t2^6-2*t2^2*t3^2*t4^2,
```

```
-t3^6-t4^6-t3^2*t4^4+t3^4*t4^2-t2^2*t4^4-3*t2^2*t3^4+t2^4*t4^2-3*t2^4*t3^2-t2^6+2*t2^2*t3^2*t4^2,
-t3^6-t4^6-3*t3^2*t4^4-3*t3^4*t4^2-3*t2^2*t4^4-3*t2^2*t3^4+3*t2^4*t4^2+3*t2^4*t3^2-3*t2^6-6*t2^2*t3^2*t4^2,
-3*t3^6-t4^6-3*t3^2*t4^4+3*t3^4*t4^2-3*t2^2*t4^4-9*t2^2*t3^4+3*t2^4*t4^2-9*t2^4*t3^2-
3*t2^6+6*t2^2*t3^2*t4^2,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
-t2^4*t4^4-t2^4*t3^4+t2^4*t3^2*t4^2-t3^8-3*t2^2*t3^2*t4^4-3*t3^4*t4^4+2*t3^6*t4^2+
3*t2^2*t3^4*t4^2-2*t2^2*t3^6,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-t2^4*t4^4-3*t2^4*t3^4-3*t2^4*t3^2*t4^2-t3^8+t2^2*t3^2*t4^4-t3^4*t4^4-2*t3^6*t4^2+
3*t2^2*t3^4*t4^2+2*t2^2*t3^6,
-t3^6+t2^2*t3^4-t2^4*t3^2-3*t2^6-4*t2^4*t4^2-4*t3^4*t4^2+4*t2^2*t3^2*t4^2,
t3^6-t2^2*t3^4+t2^4*t3^2+t2^6+2*t2^4*t4^2+2*t3^4*t4^2-2*t2^2*t3^2*t4^2,
-3*t3^6+3*t2^2*t3^4-3*t2^4*t3^2-t2^6-4*t2^4*t4^2-4*t3^4*t4^2+4*t2^2*t3^2*t4^2,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
-t2^4+t2^2*t3^2+t2^2*t4^2-t3^4-2*t3^2*t4^2-t4^4,
3*t2^4-2*t2^2*t3^2-2*t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t3^6+t4^6+3*t3^2*t4^4+3*t3^4*t4^2-t2^2*t4^4-t2^2*t3^4+t2^4*t4^2+t2^4*t3^2+t2^6-2*t2^2*t3^2*t4^2,
-3*t3^6-3*t4^6-9*t3^2*t4^4-9*t3^4*t4^2+3*t2^2*t4^4+3*t2^2*t3^4-3*t2^4*t4^2-3*t2^4*t3^2-
t2^6+6*t2^2*t3^2*t4^2,
t2^4-2*t2^2*t3^2-2*t2^2*t4^2+3*t3^4+6*t3^2*t4^2+3*t4^4,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
-t2^4-2*t2^2*t3^2+t2^2*t4^2+t3^2*t4^2-t3^4-t4^4,
3*t2^4+6*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+3*t3^4+t4^4,
t3^6+t4^6-t3^2*t4^4+t3^4*t4^2-t2^2*t4^4+3*t2^2*t3^4+t2^4*t4^2+3*t2^4*t3^2+t2^6+2*t2^2*t3^2*t4^2,
-t3^6-3*t4^6+3*t3^2*t4^4-3*t3^4*t4^2+3*t2^2*t4^4-3*t2^2*t3^4-3*t2^4*t4^2-3*t2^4*t3^2-t2^6-6*t2^2*t3^2*t4^2,
t2^4+2*t2^2*t3^2-2*t2^2*t4^2-2*t3^2*t4^2+t3^4+3*t4^4,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t3^6+3*t4^6+t3^2*t4^4-t3^4*t4^2+4*t2^2*t4^4+4*t2^2*t3^4-4*t2^2*t3^2*t4^2,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t2^4-t2^2*t3^2-t2^2*t4^2+t3^4+2*t3^2*t4^2+t4^4,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
t2^4+2*t2^2*t3^2-t2^2*t4^2-t3^2*t4^2+t3^4+t4^4,
t3^6+t4^6+t3^2*t4^4-t3^4*t4^2+2*t2^2*t4^4+2*t2^2*t3^4-2*t2^2*t3^2*t4^2,
3*t3^6+t4^6+3*t3^2*t4^4-3*t3^4*t4^2+4*t2^2*t4^4+4*t2^2*t3^4-4*t2^2*t3^2*t4^2]
```

Comme on l'a vu dans les Chapitres 1 et 2, l'algorithme **HA3** ne permet pas d'étudier ces hypersurfaces. En revanche, l'algorithme **SA2** permet d'aboutir en 4930 secondes sur un PC 400 MHz avec 512 Mo de RAM de l'UMS Medicis [3]. L'étude de la plupart de ces hypersurfaces est immédiate. Seule une hypersurface a nécessité 'à peu près 1 heure de calcul.

Ainsi, globalement, le cas $r = 4$ et $n = 5$ nécessite approximativement 5 heures de calcul pour être complètement traité. Seules 19092 matrices (sur les 53130 de départ) sont équilibrées. Ces matrices sont disponibles au format Maple à l'adresse :
`http://www-calfor.lip6.,fr/~safey/applications.html`.

Enfin, signalons que pour toutes les hypersurfaces lisses, il n'a pas été nécessaire d'utiliser les résultats du paragraphe 4.4.2 de ce chapitre mais que pour l'étude des hypersurfaces contenant une infinité de singularité, il a été nécessaire d'introduire l'équation $Tt_1t_2t_3 - 1 = 0$.

Annexe A

Décomposition de systèmes polynomiaux

Dans les Chapitres 2 et 3 de la Partie II, nous utilisons une routine de décomposition de systèmes polynomiaux nommée **LexTriSetEquiDim**. Cette routine prend en entrée un système d'équations polynomiales S et retourne une famille de bases de Gröbner $\mathcal{G}_1, \ldots, \mathcal{G}_\ell$ telle que :

- $V(S) = V(\mathcal{G}_1) \bigcup \ldots \bigcup V(\mathcal{G}_\ell)$;
- pour tout $i \in \{1, \ldots, \ell\}$, l'idéal $\langle \mathcal{G}_i \rangle$ est équi-dimensionnel et radical;
- pour tout $i \in \{1, \ldots, \ell\}$, on peut extraire de $\mathcal{G}_i$ un ensemble triangulaire régulier et séparable $\mathcal{T}_i$ tel que $\mathrm{sat}(\mathcal{T}_i) = \langle \mathcal{G}_i \rangle$.

Dans cette annexe, nous décrivons comment à partir des résultats de [72, 5], nous avons obtenu un algorithme ayant les mêmes spécifications que celle de **LexTriSetEquiDim**.

A.1 Description de l'algorithme

Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite de $K[X_1, \ldots, X_n]$ pour l'ordre $X_1 < \ldots < X_n$. Soit $p \in K[X_1, \ldots, X_n]$, on note :

- $\mathrm{mvar}(p)$ – et on appelle variable principale de p – la plus grande variable apparaissant dans p pour l'ordre lexicographique;
- $\mathrm{mdeg}(p)$ – et on appelle degré principal de p – le degré de p en sa variable principale.

Le résultat suivant est bien connu.

Lemme A.1 *Soit $\mathcal{G}$ une base de Gröbner lexicographique réduite. Alors, pour tout $g \in \mathcal{G}$ l'initial de g (où g est vu comme un polynôme univarié en* $\mathrm{mvar}(g)$*) n'appartient pas à $\langle \mathcal{G} \rangle$.*

Pour $i \in \{1, \ldots, n\}$, on note $\mathcal{G}_{\leq X_i}$ la famille de polynômes $\mathcal{G} \bigcap K[X_1, \ldots, X_i]$. Soit $\mathcal{T} = (t_{d+1}, \ldots, t_n)$ l'ensemble triangulaire extrait de $\mathcal{G}$ tel que :

$$\forall p \in \mathcal{G} \mid \mathrm{mvar}(p) = \mathrm{mvar}(t_i), \quad \mathrm{mdeg}(p) \geq \mathrm{mdeg}(t_i),$$

et **ExtractTriangular** une routine prenant en entrée $\mathcal{G}$ et renvoyant un tel ensemble triangulaire. On note :

- h_i le coefficient dominant de $t_i \in \mathcal{T}$ vu comme un polynôme univarié en $\mathrm{mvar}(t_i)$;
- s_i la dérivée partielle de $t_i \in \mathcal{T}$ par rapport à sa variable principale.

On note **SplitbyInitials** la routine, décrite ci-dessous, qui prend en entrée une base de Gröbner $\mathcal{G} \subset K[X_1, \ldots, X_n]$ lexicographique réduite telle que :

- $\mathcal{T}_{\leq X_{n-1}} = \mathrm{ExtractTriangular}(\mathcal{G}_{\leq X_{n-1}})$ est régulier séparable
- $\mathrm{sat}(\mathcal{T}_{\leq X_{n-1}}) = \langle \mathcal{G}_{\leq X_{n-1}} \rangle$

Elle renvoie :

- $\mathcal{G}$ si pour tout idéal premier $\mathcal{P}$ associé à $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$, h_n est inversible dans l'anneau quotient $K[X_1, \ldots, X_{n-1}]/\mathcal{P}$;
- sinon deux bases de Gröbner lexicographiques réduites $\mathcal{G}_1$ et $\mathcal{G}_2$ telles que :
  - $\langle \mathcal{G}_1 \rangle = \langle \mathcal{G} \rangle + \langle h_n \rangle$;
  - $\langle \mathcal{G}_2 \rangle = (\langle \mathcal{G} \rangle + \langle T h_n - 1 \rangle) \bigcap K[X_1, \ldots, X_n]$ où T est une nouvelle variable.

On note **GrobnerLex** une routine qui prend en entrée une liste de polynômes et calcule une base de Gröbner lexicographique réduite de l'idéal engendrée par ces polynômes.

Algorithme SplitbyInitials

- **Entrée :** Une base de Gröbner $\mathcal{G} \subset K[X_1, \ldots, X_n]$ lexicographique réduite telle que :
  - $\mathcal{T}_{\leq X_{n-1}} = \mathrm{ExtractTriangular}(\mathcal{G}_{\leq X_{n-1}})$ est régulier séparable
  - $\mathrm{sat}(\mathcal{T}_{\leq X_{n-1}}) = \langle \mathcal{G}_{\leq X_{n-1}} \rangle$.
- **Sortie :**
  - $\mathcal{G}$ si pour tout idéal premier $\mathcal{P}$ associé à $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$, h_n est inversible dans l'anneau quotient $K[X_1, \ldots, X_{n-1}]/\mathcal{P}$;
  - sinon deux bases de Gröbner lexicographiques réduites $\mathcal{G}_1$ et $\mathcal{G}_2$ telles que :
    - $\langle \mathcal{G}_1 \rangle = \langle \mathcal{G} \rangle + \langle h_n \rangle$;
    - $\langle \mathcal{G}_2 \rangle = (\langle \mathcal{G} \rangle + \langle T h_n - 1 \rangle) \bigcap K[X_1, \ldots, X_n]$ où T est une nouvelle variable.

1. Poser $\mathcal{T} := \mathrm{ExtractTriangular}(\mathcal{G})$ et calculer $d := \mathrm{Dim}(V(h_n) \bigcap V(\mathcal{G}_{\leq X_{n-1}}))$
2. Si $d < \mathrm{Dim}(V(\mathcal{G}_{\leq X_{n-1}}))$ alors retourner $\mathcal{G}$
3. sinon
   - $\mathcal{G}_1 := \mathrm{GrobnerLex}(\mathcal{G}, h_n)$
   - $\mathcal{G}_2 := \mathrm{GrobnerLex}(\mathcal{G}, T h_n - 1) \bigcap K[X_1, \ldots, X_n]$ (avec $T > X_n > \ldots > X_1$)
4. retourner $[\mathcal{G}_1, \mathcal{G}_2]$.

Lemme A.2

1. *Si* $\mathrm{SplitbyInitials}(\mathcal{G}) = [\mathcal{G}]$ *alors* $\mathcal{T}$ *est un ensemble triangulaire régulier et* $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.
2. *Si* $\mathrm{SplitbyInitials}(\mathcal{G}) = [\mathcal{G}_1, \mathcal{G}_2]$, *alors* $\mathcal{G} \subset \mathcal{G}_1$ *et* $\mathcal{G} \subset \mathcal{G}_2$ *et ces inclusions sont strictes.*

Preuve :

1. Puisque $\langle \mathcal{G}_{\leq X_{n-1}} \rangle = \mathrm{sat}(\mathcal{T}_{\leq X_{n-1}})$, l'idéal $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$ est équi-dimensionnel, et donc tous les idéaux premiers associés à $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$ sont de même dimension. Ainsi, pour tout $p \in K[X_1,\ldots,X_{n-1}]$, $\dim(V(\mathcal{G}_{\leq X_{n-1}}) \cap V(p)) < \dim(V(\mathcal{G}_{\leq X_{n-1}}))$ implique que pour tout idéal premier $\mathcal{P}$ associé à $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$, $p \notin \mathcal{P}$. Ceci est en particulier vrai pour h_n et il est alors clair que si h_n n'appartient à aucun idéal premier associé à $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$, $\mathcal{T}$ est un ensemble triangulaire régulier et $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.
2. Le fait que l'inclusion $\langle \mathcal{G} \rangle \subset \langle \mathcal{G}_1 \rangle$ soit stricte se déduit directement du Lemme A.1. Il nous reste à montrer que l'inclusion $\mathcal{G} \subset \mathcal{G}_2$ est stricte. Pour cela, on raisonne par l'absurde : supposons que cette inclusion n'est pas stricte. Ceci implique que $h_n \in \mathcal{G}$, et comme $h_n \in K[X_1,\ldots,X_{n-1}]$, on aurait $h_n \in \mathcal{G}_{\leq X_{n-1}}$. Ceci impliquerait alors que h_n appartienne à tous les idéaux premiers associés à $\mathcal{G}_{\leq X_{n-1}}$ et que donc $\dim(V(\mathcal{G}_{\leq X_{n-1}}) \cap V(h_n)) = \dim(V(\mathcal{G}_{\leq X_{n-1}}))$, ce qui est contraire aux hypothèses.

∎

On note **SplitbySeparants** la routine, décrite ci-dessous, qui prend en entrée une base de Gröbner $\mathcal{G} \subset K[X_1,\ldots,X_n]$ lexicographique réduite telle que :

- $\mathcal{T}_{\leq X_{n-1}} = \mathrm{ExtractTriangular}(\mathcal{G}_{\leq X_{n-1}})$ est régulier séparable
- $\mathrm{sat}(\mathcal{T}_{\leq X_{n-1}}) = \langle \mathcal{G}_{\leq X_{n-1}} \rangle$ (ceci implique que l'idéal $\langle \mathcal{G}_{\leq X_{n-1}} \rangle$ est radical).

Elle renvoie :

- $\mathcal{G}$ si pour tout idéal premier $\mathcal{P}$ associé à $\langle \mathcal{G} \rangle$, s_n est inversible dans l'anneau quotient $K[X_1,\ldots,X_n]/\mathcal{P}$;
- sinon deux bases de Gröbner lexicographiques réduites $\mathcal{G}_1$ et $\mathcal{G}_2$ telles que :
  - $\langle \mathcal{G}_1 \rangle = \langle \mathcal{G} \rangle + \langle s_n \rangle$;
  - $\langle \mathcal{G}_2 \rangle = (\langle \mathcal{G} \rangle + \langle T s_n - 1 \rangle) \bigcap K[X_1,\ldots,X_n]$ où T est une nouvelle variable.

Algorithme SplitbySeparants

- **Entrée :** Une base de Gröbner $\mathcal{G} \subset K[X_1,\ldots,X_n]$ lexicographique réduite telle que :
  - $\mathcal{T} = \mathrm{ExtractTriangular}(\mathcal{G}_{\leq X_{n-1}})$ est régulier séparable
  - $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G}_{\leq X_{n-1}} \rangle$.
- **Sortie :**
  - $\mathcal{G}$ si pour tout idéal premier $\mathcal{P}$ associé à $\langle \mathcal{G} \rangle$, s_n est inversible dans l'anneau quotient $K[X_1,\ldots,X_n]/\mathcal{P}$;
  - sinon deux bases de Gröbner lexicographiques réduites $\mathcal{G}_1$ et $\mathcal{G}_2$ telles que :
    - $\langle \mathcal{G}_1 \rangle = \langle \mathcal{G} \rangle + \langle s_n \rangle$;
    - $\langle \mathcal{G}_2 \rangle = (\langle \mathcal{G} \rangle + \langle T s_n - 1 \rangle) \bigcap K[X_1,\ldots,X_n]$ où T est une nouvelle variable.

1. $\mathcal{T} := \mathrm{ExtractTriangular}(\mathcal{G})$ et calculer $d := \mathrm{Dim}(V(s_n) \bigcap V(\mathcal{G}_{\leq X_{n-1}}))$
2. Si $d < \mathrm{Dim}(V(\mathcal{G}_{\leq X_{n-1}}))$ alors retourner $\mathcal{G}$
3. sinon
   - $\mathcal{G}_1 := \mathrm{GrobnerLex}(\mathcal{G}, s_n)$
   - $\mathcal{G}_2 := \mathrm{GrobnerLex}(\mathcal{G}, T s_n - 1) \bigcap K[X_1,\ldots,X_n]$ (avec $T > X_n > \ldots > X_1$)
4. retourner $[\mathcal{G}_1, \mathcal{G}_2]$.

Lemme A.3

1. *Si* SplitbySeparants($\mathcal{G}$) = [$\mathcal{G}$] *alors* $\mathcal{T}$ *est un ensemble triangulaire régulier séparable et* sat($\mathcal{T}$) = $\langle\mathcal{G}\rangle$.
2. *Si* SplitbySeparants($\mathcal{G}$) = [$\mathcal{G}_1$,$\mathcal{G}_2$], *alors* $\mathcal{G} \subset \mathcal{G}_1$ *et* $\mathcal{G} \subset \mathcal{G}_2$ *et ces inclusions sont strictes.*

Preuve : La preuve est en tout point similaire à celle du lemme précédent. ∎

De la même manière que pour la routine **SplitbyInitials**, lorsque **SplitbySeparants** engendre un scindage, les deux idéaux codés par les deux bases de Gröbner retournées contiennent strictement celui qui est codé par $\mathcal{G}$.

Nous décrivons l'algorithme **LexTriSetEquiDim** ci-dessous. Il prend en entrée un système d'équations polynomiales S dans $K[X_1, \dots ,X_n]$ et nous allons démontrer ci-dessous qu'il renvoie une famille de bases de Gröbner $\mathcal{G}_1, \dots ,\mathcal{G}_\ell$ telles que :

- $V(S) = V(\mathcal{G}_1) \bigcup \dots \bigcup V(\mathcal{G}_\ell)$;
- pour tout $i \in \{1, \dots ,\ell\}$, $\langle\mathcal{G}_i\rangle$ est radical, équi-dimensionnel et sat($\mathcal{T}_i$) = $\langle\mathcal{G}_i\rangle$ où $\mathcal{T}_i$ = ExtractTriangular($\mathcal{G}_i$).

Algorithme LexTriSetEquiDim

- **Entrée :** un système d'équations polynomiales S dans $K[X_1, \dots ,X_n]$.
- **Sortie :** une famille de bases de Gröbner $\mathcal{G}_1, \dots ,\mathcal{G}_\ell$ telles que :
  - $V(S) = V(\mathcal{G}_1) \bigcup \dots \bigcup V(\mathcal{G}_\ell)$;
  - pour tout $i \in \{1, \dots ,\ell\}$, $\langle\mathcal{G}_i\rangle$ est radical, équi-dimensionnel et sat($\mathcal{T}_i$) = $\langle\mathcal{G}_i\rangle$ où $\mathcal{T}_i$ = ExtractTriangular($\mathcal{G}_i$).

1. Calculer la base de Gröbner lexicographique réduite $\mathcal{G}$ de $\langle S\rangle$ pour l'ordre $X_1 < \dots < X_n$
2. Faire toStudy := [$\mathcal{G}$] et toReturn := []
3. Tant que toStudy ≠ [] faire
   - (*)$\mathcal{G}$:= first(toStudy), enlever $\mathcal{G}$ de toStudy et faire $i := 1$
   - Tant que $i < n$ faire
     - list := SplitbyInitials($\mathcal{G}_{\leq X_i}$)
     - si list = [$\mathcal{G}_{\leq X_i}$] alors $i := i + 1$
     - sinon calculer $\mathcal{G}'_1$ = GrobnerLex($\mathcal{G}_1 \cup \mathcal{G}$), $\mathcal{G}'_2$ = GrobnerLex($\mathcal{G}_2 \cup \mathcal{G}$), faire toStudy := toStudy $\bigcup\{\mathcal{G}'_1,\mathcal{G}'_2\}$ et revenir au pas (*)
   - Tant que $i < n$ faire
     - list := SplitbySeparants($\mathcal{G}_{\leq X_i}$)
     - si list = [$\mathcal{G}_{\leq X_i}$] alors $i := i + 1$
     - sinon calculer $\mathcal{G}'_1$ = GrobnerLex($\mathcal{G}_1 \cup \mathcal{G}$), $\mathcal{G}'_2$ = GrobnerLex($\mathcal{G}_2 \cup \mathcal{G}$), faire toStudy := toStudy $\bigcup\{\mathcal{G}'_1,\mathcal{G}'_2\}$ et revenir au pas (*)
   - Faire toReturn := toReturn $\bigcup\{\mathcal{G}\}$
4. Retourner toReturn.

La terminaison de l'algorithme ci-dessous découle alors des Lemmes A.2, A.3. La correction de l'algorithme est une conséquence évidente du résultat ci-dessous.

Proposition A.1 *Soit* $\mathcal{G} \in$ LexTriSetEquiDim(S) *et* $\mathcal{T}$ = ExtractTriangular($\mathcal{G}$). *Alors* *on a :*

1. $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$;
2. $\mathcal{T}$ *est régulier et séparable.*

Preuve :

1. Montrons dans un premier temps que $\langle \mathcal{G} \rangle \subset \mathrm{sat}(\mathcal{T})$. Soit $p \in \langle \mathcal{G} \rangle$ et $r = \mathrm{prem}(p,\mathcal{T})$. Il est alors clair qu'il existe $\delta_{d+1},\ldots,\delta_n$ dans $\mathbb{N}$ et $q_{d+1},\ldots,q_n$ dans $K[X_1,\ldots,X_n]$ tels que :
$$r = h_{d+1}^{\delta_{d+1}} \ldots h_n^{\delta_n} p + q_{d+1}t_{d+1} + \ldots + q_n t_n$$
Ainsi, on a $r \in \langle \mathcal{G} \rangle$ et donc il existe $g \in \mathcal{G}$ tel que $\mathrm{lm}(g)$ divise $\mathrm{lm}(r)$. Posons $v = \mathrm{mvar}(g)$. Par construction de $\mathcal{T}$, la variable v est une variable algébrique de $\mathcal{T}$ et $\mathrm{mdeg}(\mathcal{T} \cap K[X_1,\ldots,v]) \leq \mathrm{mdeg}(g)$. Supposons que $r \neq 0$. Le polynôme r est réduit par rapport à $\mathcal{T}$ donc son monôme de tête l'est aussi. Par conséquent $\mathrm{lm}(g)$ est lui aussi réduit par rapport à $\mathcal{T}$ ce qui implique que $\mathrm{mdeg}(\mathcal{T} \cap K[X_1,\ldots,v]) > \mathrm{mdeg}(g)$ et est absurde. On en déduit que $r = 0$ et donc $p \in \mathrm{sat}(\mathcal{T})$ car $\{p \in K[X_1,\ldots,X_n] \mid \mathrm{prem}(p,\mathcal{T}) = 0\} \subset \mathrm{sat}(\mathcal{T})$ (voir [5]). On a donc $\langle \mathcal{G} \rangle \subset \mathrm{sat}(\mathcal{T})$.
Il nous faut maintenant montrer que $\mathrm{sat}(\mathcal{T}) \subset \langle \mathcal{G} \rangle$. Comme $\mathcal{T} \subset \mathcal{G}$, on a :
$$\mathrm{sat}(\mathcal{T}) \subset \{p \in K[X_1,\ldots,X_n] \mid \exists n \in \mathbb{N} \quad (h_{d+1} \ldots h_n)^n \, p \in \mathcal{G}\}$$
Comme par construction, on a $\{p \in K[X_1,\ldots,X_n] \mid \exists n \in \mathbb{N} \quad (h_{d+1} \ldots h_n)^n \, p \in \mathcal{G}\} = \langle \mathcal{G} \rangle$, on en conclut que $\mathrm{sat}(\mathcal{T}) \subset \langle \mathcal{G} \rangle$.
2. Ceci est une conséquence directe du point précédent et des Lemmes A.2 et A.3.

■

A.2 Implantations

Les algorithmes proposés ci-dessus ont été implantés en Maple, les calculs de bases de Gröbner étant faits par le logiciel Gb (implanté en C++ par J.-C. Faugère). Nous avons apporté quelques modifications à l'algorithme **LexTriSetEquiDim** proposé ci-dessus. En intégrant de la factorisation de polynômes, nous pouvons éviter certains appels aux routine **SplitbyInitials** et **SplitbySeparants**. Pour s'en convaincre, il suffit de considérer l'exemple :
$$\begin{cases} xz = 0 \\ xy = 0 \end{cases}$$
Par ailleurs, certains critères d'irréductibilité ont été implantés afin d'éviter des appels intutiles aux routines **SplitbyInitials** et **SplitbySeparants** en particulier dans les situations où les polynômes de l'ensemble triangulaire extrait sont de degrés principaux égaux à 1.

Ainsi, pour la plupart de nos exemples, nous avons constaté que peu de calculs sont à faire après le premier calcul de base de Gröbner, et pour tous nos exemples, lorsqu'on a pu calculé la première base de Gröbner, on a pu décomposer le système d'équations polynomiales.

Sans pouvoir parler d'efficacité – comparativement aux implantations de J.-C. Faugère [1] – nous avons constaté que notre implantation avait des performances meilleures que la décomposition en idéaux premiers de Magma [24]. Il est difficile de comparer notre implantation à la stratégie qui consiste à calculer une décomposition en ensembles triangulaires réguliers et séparables au sens de Kalkbrener, puis à calculer une base de Gröbner des saturés des ensembles triangulaires retournés. En effet, à notre connaissance, toutes les implantations existantes de tels algorithmes sont écrites en Maple ou Axiom [5] ce qui limite la taille des problèmes qu'elles peuvent résoudre. Aussi, alors que notre implantation permet de «résoudre» (au sens algébrique du terme) des problèmes que les implantations de l'algorithme de Kalkbrener ne peuvent traiter directement, nous avons constaté qu'à partir d'une base de Gröbner lexicographique, ces implantations traitent au moins autant de problèmes que la nôtre.

Annexe B

Exemples de systèmes polynomiaux

Dans cette annexe, nous donnons les exemples de systèmes polynomiaux et d'hypersurfaces que nous avons utilisés pour analyser le comportement en pratique des algorithmes exposés dans ce document. La plupart de ces exemples sont issus de la batterie de systèmes tests du projet européen FRISCO (voir [2]) auquel nous avons participé. D'autres exemples sont extraits d'articles scientifiques ([21, 38]). Ils ont déjà été utilisés pour illustrer les performances d'algorithmes d'élimination algébrique (voir [7, 72, 5, 1]). Nous pouvons donc considérer qu'ils sont reconnus, au moins à l'échelle européenne, comme *représentatifs* des problèmes posés à un solveur en dimension positive.

Nous avons donc choisi de tester nos algorithmes sur ces systèmes, puisque notre objectif est de fournir des algorithmes *efficaces en pratique* pouvant être utilisés sur des applications concrètes, et qui donc ne sont pas forcément génériques, au sens géométrique du terme.

En ce qui concerne les hypersurfaces, nous n'en avons pas trouvé dans la littérature. Nous avons donc commencé par construire les hypersurfaces nommées **Hyp1, Hyp2, Hyp3, Hyp4,** et **Hyp5**. Puis, en suivant les indications de F. Rouillier, nous nous sommes intéressés au Problème d'Interpolation de Birkhoff (voir [48, 80] et Chapitre 4). Celui-ci nous a permis de tester nos algorithmes sur des *milliers* d'hypersurfaces. Nous ne donnons dans ce chapitre que quelques dizaines d'entre elles sélectionnées parmi les plus *pathologiques* (elles contiennent une infinité de points singuliers).

B.1 Hypersurfaces

Afin d'illustrer expérimentalement les algorithmes présentées dans le Chapitre 1 et l'article [84], nous avions construit les 5 hypersurfaces suivantes :

Hyp1 :

```
2*u6^2*u5*u4*u3*u2+4*u6^2*u5*u4*u3+4*u6^2*u5*u4*u2+4*u6^2*u5*u3*u2-1
```

Hyp2 :

```
36*u5^2*u4^2*u3^2*u2^2+88*u5^2*u4^2*u3^2*u2+32*u5^2*u4^2*u3^2+_
32*u5^2*u4^2*u3*u2^3+152*u5^2*u4^2*u3*u2^2-1
```

Hyp3 :

```
36*u5^2*u4^2*u3^2*u2^2+88*u5^2*u4^2*u3^2*u2+32*u5^2*u4^2*u3^2+_
32*u5^2*u4^2*u3*u2^3+152*u5^2*u4^2*u3*u2^2+64*u5^2*u4^2*u3*u2_
+64*u5^2*u4^2*u2^3+32*u5^2*u4^2*u2^2+32*u5^2*u4*u3^3*u2^2-1
```

Hyp4 :

```
552*u2*u3^2*u4+62208*u2+1492992*u3+2799360*u4-3*u2^2*u4^2_
-7842*u2*u3*u4+420*u2*u3*u4^2-314*u2^2*u3*u4+3*u2^2*u3^2*u4_
-62208*u2^2+429*u4^3+20736*u3^2-4*u2^3*u3^2-1157*u2^2*u3^2_
-18801*u2^2*u3-83520*u2*u3^2+39744*u2*u3+3*u2*u4^2+864*u2*u4_
+17280*u3^2*u4+60912*u4^2-864*u2^2*u4-207*u2^3*u3_
+1152*u3^2*u4^2+156*u4^3*u3+18540*u3*u4^2-554688*u3*u4_
+8*u2*u3^2*u4^2+2*u2^3*u3*u4-2*u2*u3*u4^3+u4^4-8957952
```

Hyp5 :

```
110*u5^2*u4*u3+190*u5*u4^2*u3+80*u4^3*u3+80*u5^2*u3^2+_
270*u5*u4*u3^2+160*u4^2*u3^2+80*u5*u3^3+80*u4*u3^3-_
32*u4*u3^2*u2-32*u3^3*u2-80*u5^2*u2^2-128*u5*u4*u2^2-_
160*u5*u3*u2^2-112*u4*u3*u2^2-64*u3^2*u2^2-80*u5*u2^3-_
32*u3*u2^3+60*u5^2*u4+220*u5*u4^2+160*u4^3+67*u5*u4*u3+_
136*u4^2*u3-24*u5*u3^2-88*u4*u3^2-64*u3^3-100*u5^2*u2+_
32*u5*u4*u2+96*u4^2*u2-228*u5*u3*u2-108*u4*u3*u2-_
120*u3^2*u2+20*u5*u2^2+96*u4*u2^2-56*u3*u2^2+110*u5*u4+_
80*u4^2+48*u4*u3-32*u3^2+30*u5*u2+48*u4*u2-20*u3*u2
```

Dans le cadre de l'étude du problème d'interpolation de Birkhoff, nous avions étudié à peu près un millier d'hypersurfaces de 3 variables. Sur ce millier d'hypersurfaces, seules 102 ne contenaient pas une infinité de points singuliers, rendant ainsi leur étude plus délicate. Nous en avons sélectionné une dizaine (qui sont les plus dures à étudier).

Birk.3-1 :

```
1+2*p2^2-2*p3^2-2*p4^2+p2^4-2*p2^2*p3^2-2*p2^2*p4^2+3*p3^4+
6*p3^2*p4^2+3*p4^4
```

Birk.3-2 :

```
p2^4+2*p2^2*p3^2-p2^2*p4^2-p3^2*p4^2+p3^4+p4^4
```

Birk.3-3 :

```
3*p3^6+3*p4^6+9*p3^2*p4^4+9*p3^4*p4^2+p2^2*p4^4+p2^2*p3^4-
p2^4*p4^2-p2^4*p3^2+p2^6+4*p4^4+4*p3^4+4*p2^4+2*p2^2*p3^2*p4^2+
8*p3^2*p4^2-4*p2^2*p4^2-4*p2^2*p3^2
```

Birk.3-4 :

```
-3*p3^6-p4^6-3*p3^2*p4^4+3*p3^4*p4^2-3*p2^2*p4^4-9*p2^2*p3^4+
3*p2^4*p4^2-9*p2^4*p3^2-3-9*p2^2-9*p3^2-9*p2^4-9*p3^4-
18*p2^2*p3^2+6*p2^2*p4^2+6*p3^2*p4^2-3*p2^6+3*p4^2-3*p4^4+
6*p2^2*p3^2*p4^2
```

Birk.3-5 :

```
2*p2^2*p3^2*p4^2-p2^8+2*p2^6-p3^4-3*p2^4+p2^2*p4^4+3*p2^4*p4^2-
```

```
p4^4-3*p2^2*p4^2-2*p3^2*p4^2-2*p2^6*p4^2-2*p2^6*p3^2-p2^4*p4^4-
p2^4*p3^4+p2^2*p3^4+3*p2^4*p3^2-3*p2^2*p3^2-2*p2^4*p3^2*p4^2
```

Birk.3-6 :

```
-6*p2^2*p3^2*p4^2-p2^8-2*p2^6-p2^4+2*p2^6*p4^2+2*p2^6*p3^2-
p4^4-p3^4-3*p2^4*p4^4-3*p2^4*p3^4-3*p2^2*p4^4-3*p2^2*p3^4-
6*p2^4*p3^2*p4^2+3*p2^4*p4^2+3*p2^4*p3^2-2*p3^2*p4^2+p2^2*p4^2+
p2^2*p3^2
```

Birk.3-7 :

```
-p2^4-p3^4-p4^4-3*p2^2*p4^4-6*p2^2*p3^4+3*p2^4*p4^2-6*p2^4*p3^2-
p2^8-2*p3^6-2*p2^6-p3^8-2*p2^2*p3^2+p2^2*p4^2+p3^2*p4^2-
3*p3^2*p4^4+3*p3^4*p4^2+6*p2^2*p3^2*p4^2+6*p2^4*p3^2*p4^2-
6*p2^2*p3^2*p4^4+6*p2^2*p3^4*p4^2+2*p2^6*p4^2-4*p2^6*p3^2-
3*p2^4*p4^4-6*p2^4*p3^4-3*p3^4*p4^4+2*p3^6*p4^2-4*p2^2*p3^6
```

Birk.3-8 :

```
p3^6+3*p4^6+p3^2*p4^4-p3^4*p4^2+4*p2^2*p4^4+4*p2^2*p3^4-
4*p2^2*p3^2*p4^2+4*p3^4+4*p4^4-4*p3^2*p4^2
```

Birk.3-9 :

```
p3^6+p4^6+2*p3^4+p3^2*p4^4-p3^4*p4^2+2*p2^2*p4^4+2*p2^2*p3^4+
2*p4^4-2*p3^2*p4^2-2*p2^2*p3^2*p4^2
```

Birk.3-10 :

```
-p3^6-3-4*p3^4*p4^2+p2^2*p3^4-4*p2^4*p4^2-p2^4*p3^2-3*p2^6+
p3^4-9*p2^4-p3^2-9*p2^2-4*p4^2+4*p2^2*p3^2*p4^2-2*p2^2*p3^2-
8*p2^2*p4^2+4*p3^2*p4^2
```

Birk.3-11 :

```
p3^6+1+2*p3^4*p4^2-p2^2*p3^4+2*p2^4*p4^2+p2^4*p3^2+p2^6-p3^4+
3*p2^4+p3^2+3*p2^2+2*p4^2-2*p2^2*p3^2*p4^2+2*p2^2*p3^2+
4*p2^2*p4^2-2*p3^2*p4^2
```

Birk.3-12 :

```
-3*p3^6-1-4*p3^4*p4^2+3*p2^2*p3^4-4*p2^4*p4^2-3*p2^4*p3^2-
p2^6+3*p3^4-3*p2^4-3*p3^2-3*p2^2-4*p4^2+4*p2^2*p3^2*p4^2-
6*p2^2*p3^2-8*p2^2*p4^2+4*p3^2*p4^2
```

Birk.3-13 :

```
-p2^4*p4^4-p2^4*p3^4+p2^4*p3^2*p4^2-p3^8-3*p2^2*p3^2*p4^4-
3*p3^4*p4^4+2*p3^6*p4^2+3*p2^2*p3^4*p4^2-2*p2^2*p3^6
```

Birk.3-14 :

```
-p4^4+p3^2*p4^2+3*p3^4*p4^2-2*p2^2*p4^4+2*p2^2*p3^2*p4^2-
p3^4-2*p3^6-2*p2^2*p3^4-3*p3^2*p4^4-p2^4*p4^4-p2^4*p3^4+
p2^4*p3^2*p4^2-p3^8-3*p2^2*p3^2*p4^4-3*p3^4*p4^4+2*p3^6*p4^2+
3*p2^2*p3^4*p4^2-2*p2^2*p3^6
```

Birk.3-15 :

```
-3*p3^2*p4^2+3*p3^4*p4^2-p4^4+2*p3^6-2*p2^2*p4^4+p3^2*p4^4-
```

```
6*p2^2*p3^2*p4^2-p2^4*p4^4-3*p2^4*p3^4-3*p2^4*p3^2*p4^2-p3^8+
p2^2*p3^2*p4^4-p3^4*p4^4-2*p3^6*p4^2+3*p2^2*p3^4*p4^2+
2*p2^2*p3^6-6*p2^2*p3^4-3*p3^4
```

B.2 Systèmes venus du monde académique

Système Euler :

```
[2 * a3 - a1,
2 * a4 - a1,
2 * a5 - a2,
2 * a6 - a2,
-2*a7*a3+a3^2+2*a7*a4-a4^2+2*a8*a5-a5^2,
-2*a7*a3+a3^2+2*a8*a6-a6^2,
a2 * a9 + a1 * a10 - a1 * a2,
a2 * a10 - a1 * a9,
-2*a7*a3+a3^2+2*a7*a9-a9^2+2*a8*a10-a10^2]
```

- 10 variables
- Dimension 3
- Degré 2

Système Buchberger :

```
[t-b-d,
x+y+z+t-a-c-d,
x*z+y*z+x*t+z*t-a*c-a*d-c*d,
x*z*t-a*c*d]
```

- 8 variables
- Dimension 4
- Degré 6

Système Donati-Traverso :

```
[x**31-x**6-x-y,
x**8-z,
x**10-t]
```

- 4 variables
- Dimension 1
- Degré 10

Système Butcher :

```
[B1 + y + z - t - w ,
2*z*u + 2*y*v + 2*t*w - 2*w^2 - w - 1,
3*z*u^2 + 3*y*v^2 - 3*t*w^2 + 3*w^3 + 3*w^2 - t + 4*w ,
6*x*z*v - 6*t*w^2 + 6*w^3 - 3*t*w + 6*w^2 - t + 4*w ,
4*z*u^3 + 4*y*v^3 + 4*t*w^3- 4*w^4 - 6*w^3 + 4*t*w - 10*w^2 - w- 1 ,
8*x*z*u*v + 8*t*w^3 - 8*w^4 + 4*t*w^2 - 12*w^3 + 4*t*w - 14*w^2 - 3*w - 1,
12*x*z*v^2 + 12*t*w^3 - 12*w^4 + 12*t*w^2 - 18*w^3 + 8*t*w - 14*w^2 -w-1,
-24*t*w^3 + 24*w^4 - 24*t*w^2 + 36*w^3 - 8*t*w + 26*w^2 + 7*w + 1]
```

- 8 variables
- Dimension 3
- Degré 3

Système Prodecco :

```
[5079655570*z3*z1+6235363358*z4*z1+914136786*z3*z2+1754030184*z4*z2-5566186923*z3*z2*z1+_
1220100665*z4*z2*z1+4821149906*z4*z3*z1-7139895402*z4*z3*z2-12926589043*z4*z3*z2*z1+_
7139895402*z4^2*z1-4821149906*z3^2*z2+5566186923*z4*z1^2-1220100665*z3*z2^2+_
6463294521*z4^2*z1^2+6463294521*z3^2*z2^2,
2479672620*z3*z1+4168821089*z4*z1+956199576*z3*z2+2093013784*z4*z2-5536375634*z3*z2*z1+_
2503259887*z4*z2*z1+2178194447*z4*z3*z1-5507365228*z4*z3*z2-16274081227*z4*z3*z2*z1+_
5507365228*z4^2*z1-2178194447*z3^2*z2+5536375634*z4*z1^2-2503259887*z3*z2^2+_
8137040613*z4^2*z1^2+8137040613*z3^2*z2^2,
1350302534*z3*z1+3703379139*z4*z1+1034675187*z3*z2+1694653994*z4*z2-4225842203*z3*z2*z1+_
2388470402*z4*z2*z1+807978279*z4*z3*z1-2317809870*z4*z3*z2-7056340638*z4*z3*z2*z1+_
2317809870*z4^2*z1-807978279*z3^2*z2+4225842203*z4*z1^2-2388470402*z3*z2^2+_
3528170319*z4^2*z1^2+3528170319*z3^2*z2^2,
1628792420*z3*z1+4297692981*z4*z1+2255973223*z3*z2+5541365097*z4*z2-4310350388*z3*z2*z1+_
5605751431*z4*z2*z1+237992575*z4*z3*z1-640051014*z4*z3*z2-1379872605*z4*z3*z2*z1+_
640051014*z4^2*z1-237992575*z3^2*z2+4310350388*z4*z1^2-5605751431*z3*z2^2+_
689936302*z4^2*z1^2+689936302*z3^2*z2^2]
```

- 5 variables
- Dimension 2
- Degré 2

Système Vermeer :

```
[x^2-2*x*u+u^2+y^2-2*y*v+v^2-1,
v^2-u^3,
2*v*x-2*v*u+3*u^2*y-3*u^2*v,
6*w^2*u^2*v-3*w*u^2-2*w*v+1]
```

- 5 variables
- Dimension 1
- Degré 26

Système discPb :

```
[-80*a1*a3*a2^2*a4+18*a1^3*a3*a2*a4-4*a1^3*a3^3-128*a2^2*a4^2+16*a2^4*a4-4*a2^3*a3^2-_
27*a1^4*a4^2+256*a4^3-27*a3^4-6*a1^2*a3^2*a4-192*a1*a3*a4^2+18*a1*a3^3*a2+_
```

```
144*a2*a1^2*a4^2+a2^2*a1^2*a3^2-4*a2^3*a1^2*a4+144*a4*a3^2*a2,
-80*a3*a2^2*a4+54*a1^2*a3*a2*a4-12*a1^2*a3^3-108*a1^3*a4^2-12*a1*a3^2*a4-192*a3*a4^2+_
18*a3^3*a2+288*a2*a1*a4^2+2*a2^2*a1*a3^2-8*a2^3*a1*a4,
-160*a3*a1*a4*a2+18*a3*a1^3*a4-256*a2*a4^2+64*a2^3*a4-12*a3^2*a2^2+18*a1*a3^3+_
144*a1^2*a4^2+2*a3^2*a2*a1^2-12*a2^2*a1^2*a4+144*a3^2*a4,
-80*a1*a2^2*a4+18*a1^3*a2*a4-12*a1^3*a3^2-8*a2^3*a3-108*a3^3-12*a1^2*a4*a3-192*a1*a4^2+_
54*a3^2*a2*a1+2*a1^2*a2^2*a3+288*a3*a2*a4,
-80*a2^2*a1*a3+18*a2*a1^3*a3-256*a2^2*a4+16*a2^4-54*a1^4*a4+768*a4^2-6*a1^2*a3^2-_
384*a1*a3*a4+288*a2*a1^2*a4-4*a2^3*a1^2+144*a3^2*a2]
```

- 4 variables
- Dimension 2
- Degré 4

Système Neural :

```
[x^2*z+y^2*z-z*a+1,
x^2*y+y*z^2-y*a+1,
x*y^2+x*z^2-x*a+1]
```

- 4 variables
- Dimension 1
- Degré 24

Système Wang :

```
[y^2-2*x1^2*x2^2*x3^2*x4*x5*x6-x1^2*x2^2*x3^2*x4^2*x5-x1^2*x2^2*x3^2*x4*x6^2-x1^2*x2^2*x3^2*x4^2*x6-_
x1^2*x2^2*x3^2*x5^2*x6-x1^2*x2^2*x3^2*x5^2*x4-x1^2*x2^2*x3^2*x5*x6^2,
2*x2^2*x3^2*x4*x5*x6+x2^2*x3^2*x4^2*x5+x2^2*x3^2*x4*x6^2+x2^2*x3^2*x4^2*x6+x2^2*x3^2*x5^2*x6+_
x2^2*x3^2*x5^2*x4+x2^2*x3^2*x5*x6^2+a2*x4,
2*x1^2*x3^2*x4*x5*x6+x1^2*x3^2*x4^2*x5+x1^2*x3^2*x4*x6^2+x1^2*x3^2*x4^2*x6+x1^2*x3^2*x5^2*x6+_
x1^2*x3^2*x5^2*x4+x1^2*x3^2*x5*x6^2+a1*x6,
2*x1^2*x2^2*x4*x5*x6+x1^2*x2^2*x4^2*x5+x1^2*x2^2*x4*x6^2+x1^2*x2^2*x4^2*x6+x1^2*x2^2*x5^2*x6+_
x1^2*x2^2*x5^2*x4+x1^2*x2^2*x5*x6^2+a3*x5,
2*x6*x5*x3^2*x2^2+2*x2^2*x3^2*x4*x5+x6^2*x3^2*x2^2+2*x2^2*x3^2*x4*x6+x5^2*x3^2*x2^2+a2,
2*x1^2*x2^2*x5*x6+2*x1^2*x2^2*x4*x5+x6^2*x2^2*x1^2+2*x6*x4*x2^2*x1^2+x4^2*x2^2*x1^2+a3,
2*x1^2*x3^2*x4*x6+x4^2*x3^2*x1^2+2*x1^2*x3^2*x5*x6+2*x5*x4*x3^2*x1^2+x5^2*x3^2*x1^2+a1,
x2^2*x6-1,
x1^2*x4-1,
x3^2*x5-1]
```

- 10 variables
- Dimension 1
- Degré 114

Système Hairer2 :

```
[B1+B2+B3+B4-1,
 2*B2*C2 + 2*B3*C3 + 2*B4*C4 - 1,
 3*B2*C2^2 +3*B3*C3^2 +3*B4*C4^2 -1,
 6*B3*A32*C2 +6*B4*A42*C2 +6*B4*A43*C3 -1,
 4*B2*C2^3 +4*B3*C3^3 +4*B4*C4^3 -1,
 8*B3*C3*A32*C2 +8*B4*C4*A42*C2 +8*B4*C4*A43*C3 -1,
 12*B3*A32*C2^2 +12*B4*A42*C2^2 +12*B4*A43*C3^2 -1,
 24*B4*A43*A32*C2 -1,
 -A21+C2,
 -A31-A32+C3,
 -A41-A42-A43+C4]
```

- 13 variables
- Dimension 2
- Degré 25

B.3 Systèmes venus du monde industriel

Les systèmes suivants sont extraits de [38]. Dans les systèmes **F633, F744,** et **F855**, les auteurs recherchent les nombres complexes de module 1 qui vérifient ces équations. Chronologiquement, nous avons d'abord oculté le sens de ces systèmes afin de tester si nos algorithmes étaient capables de résoudre de tels problèmes contenant autant de variables. Puis, dans un deuxième temps, nous avons construit à partir de ces systèmes, les systèmes **RF633, RF744,** et **RF855** qui sont la réécriture des systèmes précédents, de manière à en chercher les racines réelles.

Système F633 :

```
[2*u6 + 2*u5 + 2*u4 + 2*u3 + 2*u2 + 1,
2*U6 + 2*U5 + 2*U4 + 2*U3 + 2*U2 + 1,
1-4*u2*U3-4*u2*U4-4*u2*U5-4*u2*U6+4*u3*U2-4*u3*U4-4*u3*U5-4*u3*U6+4*u4*U2+4*u4*U3-4*u4*U5-_
4*u4*U6+4*u5*U2+4*u5*U3+4*u5*U4-4*u5*U6+4*u6*U2+4*u6*U3+4*u6*U4+4*u6*U5+2*u2+2*u3+2*u4+2*u5+2*u6,
1-4*U2*u3-4*U2*u4-4*U2*u5-4*U2*u6+4*U3*u2-4*U3*u4-4*U3*u5-4*U3*u6+4*U4*u2+4*U4*u3-4*U4*u5-_
4*U4*u6+4*U5*u2+4*U5*u3+4*U5*u4-4*U5*u6+4*U6*u2+4*U6*u3+4*U6*u4+4*U6*u5+2*U2+2*U3+2*U4+2*U5+2*U6,
U2*u2 -1,
U3*u3 -1,
U4*u4 -1,
U5*u5 -1,
U6*u6 -1]
```

- 10 variables
- Dimension 2
- Degré 32

Système F744 :

```
[2*u7 + 2*u6 + 2*u5 + 2*u4 + 2*u3 + 2*u2 + 1,
2*U7 + 2*U6 + 2*U5 + 2*U4 + 2*U3 + 2*U2 + 1,
8*U6*u7 + 8*U5*u7 + 8*U4*u7 + 8*U3*u7 + 8*U2*u7 + 8*U6*u6 + 8*U5*u6 + 8*U4*u6 + 8*U3*u6 + 8*U2*u6 + _
8*U5*u5 + 8*U4*u5 + 8*U3*u5 + 8*U2*u5 + 8*U4*u4 + 8*U3*u4 + 8*U2*u4 + 8*U3*u3 + 8*U2*u3 + 8*U2*u2 -17,
8*U7*u6 + 8*U6*u6 + 8*U7*u5 + 8*U6*u5 + 8*U5*u5 + 8*U7*u4 + 8*U6*u4 + 8*U5*u4 + 8*U4*u4 + 8*U7*u3 + _
8*U6*u3 + 8*U5*u3 + 8*U4*u3 + 8*U3*u3 + 8*U7*u2 + 8*U6*u2 + 8*U5*u2 + 8*U4*u2 + 8*U3*u2 + 8*U2*u2 -17,
16*U5*U3*u4 + 16*U5*U2*u4 + 16*U5*U2*u3 + 16*U4*U2*u3 + 8*U5*u4 + 8*U5*u3 + 8*U4*u3 + 8*U5*u2 + 8*U4*u2 +_
 8*U3*u2 + 18*U5 + 18*U4 + 18*U3 + 18*U2 + 11,
16*U4*u5*u3 + 16*U4*u5*u2 + 16*U3*u5*u2 + 16*U3*u4*u2 + 8*U4*u5 + 8*U3*u5 + 8*U2*u5 + 8*U3*u4 + 8*U2*u4 + _
8*U2*u3 + 18*u5 + 18*u4 + 18*u3 + 18*u2 + 11,
U2*u2 -1,
U3*u3 -1,
U4*u4 -1,
U5*u5 -1,
U6*u6 -1,
U7*u7 -1]
```

- 12 variables
- Dimension 1

– Degré 40

<u>Système F855 :</u>

```
[2*x8 + 2*x7 + 2*x6 + 2*x5 + 2*x4 + 2*x3 + 2*x2 + 1,
2*X8 + 2*X7 + 2*X6 + 2*X5 + 2*X4 + 2*X3 + 2*X2 + 1,
X2*x2 -1,
X3*x3 -1,
X4*x4 -1,
X5*x5 -1,
X6*x6 -1,
X7*x7 -1,
X8*x8 -1,
-4*X7*x8 -4*X6*x8 -4*X5*x8 -4*X4*x8 -4*X3*x8 -4*X2*x8 + 4*X8*x7 -4*X6*x7 -4*X5*x7 -_
4*X4*x7 -4*X3*x7-4*X2*x7 + 4*X8*x6 + 4*X7*x6 -4*X5*x6 -4*X4*x6 -4*X3*x6 -4*X2*x6 + _
4*X8*x5 + 4*X7*x5 + 4*X6*x5 -4*X4*x5 -4*X3*x5 -4*X2*x5 + 4*X8*x4 + 4*X7*x4 + 4*X6*x4 + _
4*X5*x4 -4*X3*x4 -4*X2*x4 + 4*X8*x3 + 4*X7*x3 + 4*X6*x3 + 4*X5*x3 + 4*X4*x3 -4*X2*x3 + _
4*X8*x2 + 4*X7*x2 + 4*X6*x2 + 4*X5*x2 + 4*X4*x2 + 4*X3*x2 + 2*X8 + 2*X7 + 2*X6 + 2*X5 + _
2*X4 + 2*X3 + 2*X2 + 1,
4*X7*x8 + 4*X6*x8 + 4*X5*x8 + 4*X4*x8 + 4*X3*x8 + 4*X2*x8 - 4*X8*x7 + 4*X6*x7 + 4*X5*x7 + _
4*X4*x7 + 4*X3*x7 + 4*X2*x7 - 4*X8*x6 -4*X7*x6 + 4*X5*x6 + 4*X4*x6 + 4*X3*x6 + 4*X2*x6 - _
4*X8*x5 -4*X7*x5 -4*X6*x5 + 4*X4*x5 + 4*X3*x5 + 4*X2*x5 - 4*X8*x4 -4*X7*x4 -4*X6*x4 -_
4*X5*x4 + 4*X3*x4 + 4*X2*x4 - 4*X8*x3 -4*X7*x3 -4*X6*x3 -4*X5*x3 -4*X4*x3 + 4*X2*x3 - _
4*X8*x2 -4*X7*x2 -4*X6*x2 -4*X5*x2 -4*X4*x2 -4*X3*x2 + 2*x8 + 2*x7 + 2*x6 + 2*x5 + 2*x4 + _
2*x3 + 2*x2 + 1,
16*X6*X4*x5 + 16*X6*X3*x5 + 16*X6*X2*x5 + 16*X6*X3*x4 + 16*X5*X3*x4 + 16*X6*X2*x4 + _
16*X5*X2*x4 + 16*X6*X2*x3 + 16*X5*X2*x3 + 16*X4*X2*x3 + 8*X6*x5 + 8*X6*x4 + 8*X5*x4 + _
8*X6*x3 + 8*X5*x3 + 8*X4*x3 + 8*X6*x2 + 8*X5*x2 + 8*X4*x2 + 8*X3*x2 + 26*X6 + 26*X5 + _
26*X4 + 26*X3 + 26*X2 + 15,
16*X5*x6*x4 + 16*X5*x6*x3 + 16*X4*x6*x3 + 16*X4*x5*x3 + 16*X5*x6*x2 + 16*X4*x6*x2 + _
16*X3*x6*x2 + 16*X4*x5*x2 + 16*X3*x5*x2 + 16*X3*x4*x2 + 8*X5*x6 + 8*X4*x6 + 8*X3*x6 + _
8*X2*x6 + 8*X4*x5 + 8*X3*x5 + 8*X2*x5 + 8*X3*x4 + 8*X2*x4 + 8*X2*x3 + 26*x6 + 26*x5 + _
26*x4 + 26*x3 + 26*x2 + 15,
-2*X7*X5*x8*x6 -2*X7*X4*x8*x6 -2*X7*X3*x8*x6 -2*X7*X2*x8*x6 - 2*X7*X4*x8*x5 -_
2*X6*X4*x8*x5 -2*X7*X3*x8*x5 -2*X6*X3*x8*x5 - 2*X7*X2*x8*x5 -2*X6*X2*x8*x5 + _
2*X8*X6*x7*x5 -2*X6*X4*x7*x5 - 2*X6*X3*x7*x5 -2*X6*X2*x7*x5 -2*X7*X3*x8*x4 -_
2*X6*X3*x8*x4 - 2*X5*X3*x8*x4 -2*X7*X2*x8*x4 -2*X6*X2*x8*x4 -2*X5*X2*x8*x4 + _
2*X8*X6*x7*x4 + 2*X8*X5*x7*x4 -2*X6*X3*x7*x4 -2*X5*X3*x7*x4 - 2*X6*X2*x7*x4 -_
2*X5*X2*x7*x4 + 2*X8*X5*x6*x4 + 2*X7*X5*x6*x4 - 2*X5*X3*x6*x4 -2*X5*X2*x6*x4 -_
2*X7*X2*x8*x3 -2*X6*X2*x8*x3 - 2*X5*X2*x8*x3 -2*X4*X2*x8*x3 + 2*X8*X6*x7*x3 + _
2*X8*X5*x7*x3 + 2*X8*X4*x7*x3 -2*X6*X2*x7*x3 -2*X5*X2*x7*x3 -2*X4*X2*x7*x3 + _
2*X8*X5*x6*x3 + 2*X7*X5*x6*x3 + 2*X8*X4*x6*x3 + 2*X7*X4*x6*x3 - 2*X5*X2*x6*x3 -_
2*X4*X2*x6*x3 + 2*X8*X4*x5*x3 + 2*X7*X4*x5*x3 + 2*X6*X4*x5*x3 -2*X4*X2*x5*x3 + _
2*X8*X6*x7*x2 + 2*X8*X5*x7*x2 + 2*X8*X4*x7*x2 + 2*X8*X3*x7*x2 + 2*X8*X5*x6*x2 + _
2*X7*X5*x6*x2 + 2*X8*X4*x6*x2 + 2*X7*X4*x6*x2 + 2*X8*X3*x6*x2 + 2*X7*X3*x6*x2 + _
2*X8*X4*x5*x2 + 2*X7*X4*x5*x2 + 2*X6*X4*x5*x2 + 2*X8*X3*x5*x2 + 2*X7*X3*x5*x2 + _
2*X6*X3*x5*x2 + 2*X8*X3*x4*x2 + 2*X7*X3*x4*x2 + 2*X6*X3*x4*x2 + 2*X5*X3*x4*x2 + _
X8*X6*x7 + X8*X5*x7 + X8*X4*x7 + X8*X3*x7 + X8*X2*x7 + X8*X5*x6 + X7*X5*x6 + X8*X4*x6 + _
X7*X4*x6 + X8*X3*x6 + X7*X3*x6 + X8*X2*x6 + X7*X2*x6  - X7*x8*x6 + X8*X4*x5 + X7*X4*x5 + _
X6*X4*x5 + X8*X3*x5 + X7*X3*x5 + X6*X3*x5 + X8*X2*x5 + X7*X2*x5 + X6*X2*x5  - X7*x8*x5  - _
X6*x8*x5  - X6*x7*x5 + X8*X3*x4 + X7*X3*x4 + X6*X3*x4 + X5*X3*x4 + X8*X2*x4 + X7*X2*x4 + _
X6*X2*x4 + X5*X2*x4  - X7*x8*x4  - X6*x8*x4  - X5*x8*x4  - X6*x7*x4  - X5*x7*x4  - X5*x6*x4 + _
X8*X2*x3 + X7*X2*x3 + X6*X2*x3 + X5*X2*x3 + X4*X2*x3  - X7*x8*x3  - X6*x8*x3  - X5*x8*x3  - _
X4*x8*x3  - X6*x7*x3  - X5*x7*x3  - X4*x7*x3  - X5*x6*x3  - X4*x6*x3  - X4*x5*x3  - X7*x8*x2  - _
X6*x8*x2  - X5*x8*x2  - X4*x8*x2  - X3*x8*x2  - X6*x7*x2  - X5*x7*x2  - X4*x7*x2  - X3*x7*x2  - _
X5*x6*x2  - X4*x6*x2  - X3*x6*x2  - X4*x5*x2  - X3*x5*x2  - X3*x4*x2,
2*X7*X5*x8*x6 + 2*X7*X4*x8*x6 + 2*X7*X3*x8*x6 + 2*X7*X2*x8*x6 + 2*X7*X4*x8*x5 + _
2*X6*X4*x8*x5 + 2*X7*X3*x8*x5 + 2*X6*X3*x8*x5 + 2*X7*X2*x8*x5 + 2*X6*X2*x8*x5 -_
2*X8*X6*x7*x5 + 2*X6*X4*x7*x5 + 2*X6*X3*x7*x5 + 2*X6*X2*x7*x5 + 2*X7*X3*x8*x4 + _
2*X6*X3*x8*x4 + 2*X5*X3*x8*x4 + 2*X7*X2*x8*x4 + 2*X6*X2*x8*x4 + 2*X5*X2*x8*x4 - _
2*X8*X6*x7*x4 -2*X8*X5*x7*x4 + 2*X6*X3*x7*x4 + 2*X5*X3*x7*x4 + 2*X6*X2*x7*x4 + _
2*X5*X2*x7*x4 -2*X8*X5*x6*x4 -2*X7*X5*x6*x4 + 2*X5*X3*x6*x4 + 2*X5*X2*x6*x4 + _
2*X7*X2*x8*x3 + 2*X6*X2*x8*x3 + 2*X5*X2*x8*x3 + 2*X4*X2*x8*x3 -2*X8*X6*x7*x3 -_
2*X8*X5*x7*x3 - 2*X8*X4*x7*x3 + 2*X6*X2*x7*x3 + 2*X5*X2*x7*x3 + 2*X4*X2*x7*x3 - _
2*X8*X5*x6*x3 -2*X7*X5*x6*x3 -2*X8*X4*x6*x3 -2*X7*X4*x6*x3 + 2*X5*X2*x6*x3 + _
```

```
2*X4*X2*x6*x3 -2*X8*X4*x5*x3 -2*X7*X4*x5*x3 - 2*X6*X4*x5*x3 + _
2*X4*X2*x5*x3 -2*X8*X6*x7*x2 -2*X8*X5*x7*x2 - 2*X8*X4*x7*x2 -2*X8*X3*x7*x2 -_
2*X8*X5*x6*x2 -2*X7*X5*x6*x2 - 2*X8*X4*x6*x2 -2*X7*X4*x6*x2 -2*X8*X3*x6*x2 -_
2*X7*X3*x6*x2 - 2*X8*X4*x5*x2 -2*X7*X4*x5*x2 -2*X6*X4*x5*x2 -2*X8*X3*x5*x2 - _
2*X7*X3*x5*x2 -2*X6*X3*x5*x2 -2*X8*X3*x4*x2 -2*X7*X3*x4*x2 - 2*X6*X3*x4*x2 -_
2*X5*X3*x4*x2  - X8*X6*x7  - X8*X5*x7  - X8*X4*x7  - X8*X3*x7  - X8*X2*x7  - X8*X5*x6  - _
X7*X5*x6  - X8*X4*x6  - X7*X4*x6  - X8*X3*x6  - X7*X3*x6  - X8*X2*x6  - X7*X2*x6 + X7*x8*x6  - _
X8*X4*x5  - X7*X4*x5  - X6*X4*x5  - X8*X3*x5  - X7*X3*x5  - X6*X3*x5  - X8*X2*x5  - X7*X2*x5  - _
X6*X2*x5 + X7*x8*x5 + X6*x8*x5 + X6*x7*x5  - X8*X3*x4  - X7*X3*x4  - X6*X3*x4  - X5*X3*x4  - _
X8*X2*x4  - X7*X2*x4  - X6*X2*x4  - X5*X2*x4 + X7*x8*x4 + X6*x8*x4 + X5*x8*x4 + X6*x7*x4 + _
X5*x7*x4 + X5*x6*x4  - X8*X2*x3  - X7*X2*x3  - X6*X2*x3  - X5*X2*x3  - X4*X2*x3 + X7*x8*x3 + _
X6*x8*x3 + X5*x8*x3 + X4*x8*x3 + X6*x7*x3 + X5*x7*x3 + X4*x7*x3 + X5*x6*x3 + X4*x6*x3 + _
X4*x5*x3 + X7*x8*x2 + X6*x8*x2 + X5*x8*x2 + X4*x8*x2 + X3*x8*x2 + X6*x7*x2 + X5*x7*x2 + _
X4*x7*x2 + X3*x7*x2 + X5*x6*x2 + X4*x6*x2 + X3*x6*x2 + X4*x5*x2 + X3*x5*x2 + X3*x4*x2]
```

- 12 variables
- Dimension 1
- Degré 52

Bibliographie

[1] J.C. Faugère's Home Page `http://www-calfor.lip6.fr/~jcf`

[2] The FRISCO Test-suite, `http://www-sop.inria.fr/saga/POL`

[3] UMS Medicis, `http://medicis.polytechnique.fr`

[4] M. E. Alonso, E. Becker, M.-F. Roy. T. Wormann, *Zeroes, Multiplicities and Idempotents for Zero Dimensional Systems,* in Algorithms in Algebraic Geometry and Applications, Laureano Gonzalez Vega and Tomas Recio Eds., 6-16, Birkhauser (1996).

[5] P. Aubry, *Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom*, Thèse de doctorat, Université de Paris VI, 1999.

[6] P. Aubry, D. Lazard, M. Moreno Maza, *On the theories of triangular sets,* in Journal of Symbolic Computation, 1999.

[7] P. Aubry, M. Moreno Maza, *Triangular sets for solving polynomial systems : a comparison of four implementations,* in Journal of Symbolic computation.

[8] P. Aubry, F. Rouillier, M. Safey El Din, *Real Solving for positive dimensional systems,* Rapport de Recherche du Laboratoire d'Informatique de Paris VI, 2000.

[9] B. Bank, M. Giusti, J. Heintz, and M. Mbakop *Polar Varieties, real equation solving and data structures : The hypersurface case*, Journal of Complexity 13, No 1, 5-27, (1997), Best Paper Award J. Complexity 1997.

[10] B. Bank, M. Giusti, J. Heintz, R. Mandel, and M. Mbakop *Polar Varieties and Efficient Real Equation Solving : The Hypersurface Case*, Proceedings of the 3rd Conference Approximation and Optimization in the Carribean, in : Aportaciones Matemàticas, Mexican Society of Mathematics, J. Bustamante, M. A. Jimenez, et al. (eds) (1998).

[11] B. Bank, M. Giusti, J. Heintz, and M. Mbakop *Polar Varieties and Efficient Real Elimination*, in Mathematische Zeitschrift, 2000.

[12] S. Basu, R. Pollack, M.-F. Roy, *A New Algorithm to Find a Point in Every Cell Defined by a Family of Polynomials,* in Quantifier Elimination and Cylindri-

cal Algebraic Decomposition, Texts and Monographs in Symbolic Computation, B. Caviness and J. Johnson, Eds. 341-349, Springer-Verlag, Wien, New York (1998).

[13] S. Basu, R. Pollack, M.-F. Roy, *On the combinatorial and algebraic complexity of Quantifier elimination.* J. Assoc. Comput. Machin., 43, 1002–1045, (1996).

[14] S. Basu, R. Pollack, M.-F. Roy, *Algorithms in real algebraic geometry*, in preparation.

[15] E. Becker, R. Neuhaus, *Computation of Real Radicals for polynomial ideals*, in Computational Algebraic Geometry, Progress in Math., vol. 109, 1-20, Birkhaüser, 1993.

[16] E. Becker, T. Wörmann, *On the trace formula for quadratic forms*, Proc. RAG-SQUAD (1992).

[17] T. Becker, V. Weispfenning, *Gröbner Bases: A Computational Appraoch to Commutative Algebra*, Springer-Verlag, 1993.

[18] R. Benedetti, J.-J. Risler, *Real algebraic and semi-algebraic sets,* Hermann, 1990.

[19] D. Bini, V. Pan, *Polynomial and matrix computation, Volume I*, Birkhaüser.

[20] J. Bochnak, M. Coste, M.-F. Roy, *Real algebraic geometry,* Springer-Verlag (1999).

[21] W. Boege, R. Gebauer, H. Kredel, *Some examples for solving systems of algebraic equations by calculating Gröbner bases,* Academic Press (London) Ltd, 1986.

[22] W.S. Brown, J.F. Traub, *On Euclid's Algorithm and the theory of Subresultants*, J.A.C.M., 18:505-524, 1971.

[23] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensiolanen Polynomideal*, Thèse de Doctorat, Innsbruck, Autriche, 1965.

[24] J.J. Cannon, C. Playoust, *An Introduction to Algebraic Programming with MAGMA,* University of Sydney, 1997.

[25] J. Canny , *Some Algebraic and Geometric Computations in PSPACE,* Proc. Twentieth ACM Symp. on Theory of Computing, 460-467, (1988).

[26] J. Canny , *A toolkit for nonlinear algebra,* Goldberg, Ken (ed.) et al., Algorithmic foundations of robotics, Proceedings of the workshop on the algorithmic foundations of robotics, WAFR'94, held in San Francisco, CA, USA, 17-19 February, 1994. Wellesley, MA: A. K. Peters.

[27] G.E. Collins, *Subresultants and Reduced Polynomial Remainder Sequences*, J.A.C.M., 14:128-142, 1967.

[28] G. E. Collins, *Quantifier elimination for real closed fields by cylindrical algebraic decomposition,* Springer Lecture Notes in Computer Science 33, 515- 532, (1975).

[29] G. E. Collins and H. Hong, *Partial Cylindrical Algebraic Decomposition for Quantifier Elimination*, Journal of Symbolic Computation, vol. 12, 299-328, 1991.

[30] P. Conti, C. Traverso, *Algorithms for the real radical,* unpublished manuscript

[31] M. Coste, *Introduction à la géométrie semi-algébrique*, Polycopié, Institut de Recherche Mathématique de Rennes.

[32] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms,* Springer-Verlag (1991)

[33] S. Dellière, *Triangularisation de systèmes constructibles – Application à l'évaluation dynamique*, Thèse de l'Université de Limoges, 1999.

[34] M. Demazure, *Catastrophes et bifurcations*, Ed. Ellipses, 1989.

[35] D. Duval, *Rational puiseux expansions,* Composition Math. 70, vol 2:119-154, 1989.

[36] J.C. Faugère *Résolution de systèmes d'équations algébriques*, Thèse de Doctorat, Université de Paris VI, 1994.

[37] J.C. Faugère *A new efficient Algorithm for computing Gröbner bases*, in Journal of Pure and Applied Algebra, 1999.

[38] J.C. Faugère, F. Moreau de Saint-Martin, F.Rouillier, *Design of regular nonseparable bidimensional wavelets using Groebner basis techniques*, in IEEE SP Transactions Special Issue on Theory and Applications of Filter Banks and Wavelets (1997).

[39] G. Gallo, B. Mishra, *Efficient algorithms and bounds for Wu-Ritt characteristic sets*, In Proceedings MEGA'90, 119-142, 1990.

[40] G. Gallo, B. Mishra, *Wu-Ritt characteristic sets and their complexity*, In J.E. Goodman, R. Pollack, W. Steiger, editor, Discrete and Computational Geometry : Papers from the DIMACS Special Year, vol. 6 of DIMACS Series in Discrete Mathematicsc and Theoritical Computer Science, pages 111-136, American Mathematical Society and Association for Computing Machinery, 1991.

[41] J. Von Zur Gathen, J. Gerhardt, *Modern Computer Algebra,*Cambridge University Press, 1999.

[42] P.Gianni, *Properties of Gröbner basis under specializations,* Lecture Notes in Computer Science, Vol. 378, 293-297 (1987)

[43] M. Giusti, J. Heintz, *La détermination des points isolés et de la dimension d'une variété algébrique réelle peut se faire en temps polynomial*, Computational Algebraic Geometry and Commutative Algebra, Eds D. Eisenbud and L. Robbiano, 1993.

[44] M. Giusti, J. Heintz, K. Hägele, J. E. Morais, J. L. Montaña, L. M. Pardo, *Lower Bounds for Diophantine Approximations*, Journal of Pure and Applied Algebra, 117&118, 277-317 (1997).

[45] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo, *Straight-line programs in geometric elimination theory*, Journal of Pure and Applied Algebra, 124, No 1-3, 101-146 (1998).

[46] M. Giusti, J. Heintz, J. E. Morais, L. M. Pardo, *When polynomial equation systems can be "solved" fast?*, in Proc. 11th International Symposium Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-11, Paris 1995, G. Cohen, M. Giusti and T. Mora, eds, Springer LNCS, 948, 205-231 (1995).

[47] M. Giusti, G. Lecerf, B. Salvy, *A Gröbner free alternative for solving polynomial systems*, in Journal of Complexity 2000.

[48] L. Gonzalez-Vega, *Applying quantifier elimination to the Birkhoff Interpolation Problem,* Journal of Symbolic Computation, 1996.

[49] L. Gonzalez-Vega, H. Lombardi, T. Recio, M.-F. Roy, *Spécialisation de la suite de sturm et sous-résultants*, Informatique théorique et applications, 24:561-588, 1990.

[50] L. Gonzalez-Vega, H. Lombardi, T. Recio, M.-F. Roy, *Spécialisation de la suite de sturm*, Informatique théorique et applications, 28:1-24, 1994.

[51] L. Gonzalez-Vega, F. Rouillier, M.-F. Roy, G. Trujillo, *Symbolic Recipes for Real Solutions,* In: Sometapas of computer algebra, A. Cohen ed. Springer, 121-167, (1999).

[52] H.-G. Gräbe, *Triangular systems and factorized Gröbner bases*, In G. Cohen, M. Giusti and T. Mora editors, Proceddings AAECC'11, volume 948 of Lecture Notes in Computer Science, 248-261, Paris, France, 1995, Springer.

[53] D. Grigor'ev, N. Vorobjov , *Solving Systems of Polynomial Inequalities in Subexponential Time,* J. Symbolic Comput., 5:37–64, (1988).

[54] W. Habicht, *Eine Verallgemeinerung des Sturmschen Wurzelzählverfahrens*, Comm. Math. Helvetici, 21:99-116, 1948.

[55] J. Heintz, M.-F. Roy, P. Solernó , *On the Complexity of Semi-Algebraic Sets,* Proc. IFIP 89, San Francisco. North-Holland 293-298 (1989).

[56] J. Heintz, M.-F. Roy, P. Solerno, *On thetheoretical and practical complexity of the existential theory of the reals,* Comput. J. 36, No.5, 427–431 (1993).

[57] C. Hermite, *Remarques sur le théorème de Sturm*, C. R. Acad. Sci. Paris, 36:52-54, 1853.

[58] H. Hong, *Comparison of Several Decision Algorithms for the Existential Theory of the Reals*, Research report, RISC, 1991.

[59] M. Kalkbrener, *Solving systems of algebraic equations by using Gröbner bases*, Lecture Notes in Computer Science, Vol. 378, 293-297 (1987)

[60] M. KALKBRENER, *Three contributions to elimination theory,* Thèse de Doctorat, Johannes Kepler University, Linz, 1991.

[61] M. KALKBRENER, *A generalized euclidean algorithm for computing triangular representations of algebraic varieties*, Journal of Symbolic Computation, 15:143-167, 1993

[62] L. KRONECKER, *Grundzüge einer arithmetischen Theorie de algebraischen Grössen*, J. reine angew. Math. 1882.

[63] M. LAUER, *Computing by Homomorphic Images*, In Computer Algebra, Symbolic and Algebraic Computation, Edited by B. Buchberger, G.E. Collins and R. Loos, Berlin, Springer-Verlag.

[64] D. LAZARD, *Solving zero-dimensional algebraic systems*, Journal of Symbolic Computation, 15:117-132, 1992.

[65] D. LAZARD, *A new method for solving algebraic systems of positive dimension,* in Discrete Applied Mathematics, 1991.

[66] T. LICKTEIG, M.-F. ROY, *Cauchy index computation*, Calcolo, 33:337-351, 1996.

[67] Z. LIGATSIKAS, R. RIOBOO AND M.F. ROY, (1993). *Generic Computation of the Real Closure of an Ordered Field.* Proceedings of IMACS 93 (Lille, may 1993), 203–208.

[68] R. LOOS, *Generalized polynomial remainder sequence,* In Computer Algebra, Symbolic and Algebraic Computation, Edited by B. Buchberger, G.E. Collins and R. Loos, Berlin, Springer-Verlag.

[69] E.W. MAYR, A.R. MEYER, *The complexity of the word problems for commutative semi-groups and ideals*, Advances in Mathematics, 46:305-329, 1982.

[70] S. MC CALLUM, *An improved projection operator for Cylindrical Algebraic Decomposition*, Thèse de Doctorat, Université de Wisconsin-Madison, 1984.

[71] H.M. MÖLLER, *On decomposing systems of polynomial equations with finitely many solutions,* Applicable Algebra in Engineering, Communications and Computing, 1993.

[72] M. MORENO MAZA, *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Equations Algébriques,* Thèse de Doctorat, Université of Paris VI, 1997.

[73] D. MUMFORD *Algebraic Geometry I, Complex projective varieties*, Berlin, Heildelberg, New York : Springer Verlag (1976).

[74] J. RENEGAR *On the computational complexity and geometry of the first order theory of the reals,* J. of Symbolic Comput.13(3):255-352, (1992).

[75] R. RIOBOO, *Quelques aspects du calcul exact avec les nombres réels*, Thèse de Doctorat, Université de Paris 6, 1992.

[76] R. RIOBOO, *Algebraic Closure of an Ordered Field, Implementation inAxiom*, Proc. of Issac'92 (Berkeley, july 1992).

[77] R. Rioboo, *Computing with infinitesimals*, manuscript.

[78] J.F. Ritt, *Differential equations from an algebraic standpoint*, vol 14, American Mathematical Society Colloquium Publications, New York, 1932.

[79] J.F. Ritt, *Differential Algebra*, Dover Publications, Inc., New-York, 1966.

[80] F. Rouillier, *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux,* Thèse de Doctorat, University de Rennes I (1996).

[81] F. Rouillier, *Solving Zero-Dimensional Systems through the Rational Univariate Representation,* AAECC Journal.9 : 433-461 (1999).

[82] F. Rouillier, M.-F. Roy, *Introduction aux algorithmes de la géométrie algébrique réelle*, Calcul Formel, Applications, en préparation.

[83] F. Rouillier, M.-F. Roy, M. Safey El Din, *Testing emptiness of real hypersurfaces, real algebraic sets and semi-algebraic sets* FRISCO Report Month 23 (1998).

[84] F. Rouillier, M.-F. Roy, M. Safey El Din, *Finding at least one point in each connected component of a real algebraic set defined by a single equation,* to appear in Journal of Complexity, 1999.

[85] F. Rouillier, M. Safey El Din, E. Schost, *Solving the Birkhoff Interpolation Problem via the Critical Point Method : an experimental study,* rapport technique du LIP6, soumis aux Proceedings de la Conférence ADG'2000, Novembre 2000.

[86] F. Rouillier, P. Zimmermann, *Uspensky's algorithm : improvements and applications,* in preparation (1999).

[87] M.-F. Roy, *Basic algorithms in real algebraic geometry: from Sturm theorem to the existential theory of reals,* Lectures on Real Geometry in memoriam of Mario Raimondo, Expositions in Mathematics 23, 1- 67. Berlin, New York: de Gruyter (1996).

[88] E. Schost, *Computing parametric geometric resolutions*, manuscript, 2000.

[89] A. Seidenberg, *A new decision method for elementary algebra,* Annals of Mathematics, 60:365–374, (1954).

[90] C. Sturm, *Mémoire sur la résolutin des équations numériques*, Inst. France Sc. Math. Phys., 6, 1835.

[91] A. Tarski, *A Decision method for elementary algebra and geometry,* University of California Press (1951).

[92] W. Trinks, *On improving approximate results of Buchberger'salgorithm by Newton's method*, LNCS, 1984.

[93] J. Uspensky, *Theory of equations*, Mc Graw-Hill, New-York, 1948.

[94] A. Valibouze, *Théorie de Galois constructive*, Mémoire d'Habilitation à diriger des recherches, Université Paris VI, 1994.

[95] M. Vincent, *Sur la résolution des équations numériques*, Journal de Mathématiques pures et appliquées 1, 341-372, 1836.

[96] R. J. Walker, *Algebraic Curves*, Princeton University Press (1950).

[97] D. Wang, *Some improvements on Wu's Method for solving systems of algebraic equations*, In Wu Wen-Tsün and Cheng Min-De, editors, Proc. of the Int. Workshop on Math. Mechanisation, Beijing, China, 1992. Institute of Systems Science, Academia Sinica.

[98] D. Wang, *An elimination method based on Seidenberg's theory and its applications*, In F. Eyssette, A. Galligo, editors, Computational Algebraic Geometry, 301-328, Birkhaüser Boston, 1993.

[99] D. Wang, *An implementation of the characteristic set method in Maple*, In J. Pfalzgraf D. Wang, editors, Automated Practical Reasonning : algebraic approaches, 187-201, Springer, Wien, 1995.

[100] D. Wang, *Decomposing polynomial systems into simple systems*, Journal of Symbolic Computation, 1998.

[101] D. Wang, *Computing Triangular Systems and Regular Systems,* Journal of Symbolic Computation, 2000.

[102] F. Winkler, *A p-adic approach to the computation of Gröbner bases*, Journal of Symbolic Computation, 1988.

[103] W.T. Wu, *On zeros of algebraic equations – an application of Ritt principle,* Kexue Tongbao, 31:1-5, 1986.

[104] W.T. Wu, *A zero structure theorem for polynomial equations solving*, Research Preprints, 1:2-12, 1987.

[105] R. Zippel, *Effective polynomial computation* Kluwer Academic Publishers.