



HAL
open science

Étude et implantation de quelques algorithmes en algèbre différentielle

François Boulier

► **To cite this version:**

François Boulier. Étude et implantation de quelques algorithmes en algèbre différentielle. Modélisation et simulation. Université des Sciences et Technologie de Lille - Lille I, 1994. Français. NNT : . tel-00137866

HAL Id: tel-00137866

<https://theses.hal.science/tel-00137866>

Submitted on 22 Mar 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le titre de

DOCTEUR en INFORMATIQUE

par

François Boulier

Étude et implantation de
quelques algorithmes en algèbre
différentielle

Thèse soutenue le 27 juin 1994 devant la commission d'examen

Membres du jury :	MM.	M. LATTEUX	Président
		M. GIUSTI	Rapporteur
		D. LAZARD	Rapporteur
		G. CARRÀ-FERRO	Examineur
		A. GALLIGO	Examineur
		F. OLLIVIER	Examineur
		S. DIOP	Examineur
		R. BKOUCHE	Examineur
		M. PETITOT	Examineur
		G. JACOB	Examineur

Remerciements

L'équipe de Calcul Formel forme un groupe actif au sein du LIFL. Ce dynamisme est bien sûr dû à la personnalité de ses membres mais aussi, je pense, à ses thèmes de recherche qui plongent leurs racines dans trois disciplines : informatique, automatique et mathématiques. Cette pluridisciplinarité rend les thèses de calcul formel difficiles aux étudiants en informatique qui les abordent et qui n'y sont pas toujours préparés mais elle les rend du même coup très intéressantes, parce que liées à des problèmes physiques réels et à des théories mathématiques qui ont une histoire. Je suis heureux que cette thèse, dont la dernière année a exigé de moi une véritable année, ait été consacrée à l'étude d'un tel sujet.

Mes premiers remerciements iront donc à Gérard Jacob, Professeur au LIFL, pour m'avoir proposé un travail d'une telle nature et m'avoir accordé une grande liberté dans sa réalisation.

Plus que tout autre, je remercie Michel Petitot. Nous avons passé ensemble des heures, que je ne compte plus, à faire et à refaire les preuves de théorèmes, d'algorithmes et, bien que je ne l'aie jamais mentionné dans les pages qui suivent, les résultats qui y sont exposés lui reviennent pour une bonne part. Michel Petitot est une figure marquante du LIFL, quelqu'un avec qui on parle spontanément. Sa présence dans ce jury m'a fait vraiment plaisir.

Je remercie Michel Latteux, Professeur au LIFL, d'avoir présidé ce jury de thèse. C'est en me rappelant la rigueur et le souci du détail des démonstrations exposées dans ses cours de Licence, Maîtrise, DEA d'Informatique que j'ai essayé de rédiger ce mémoire.

Les rapporteurs de cette thèse sont Daniel Lazard, Professeur à l'Université Paris VI et François Ollivier, Chargé de Recherches CNRS à l'Ecole Polytechnique, co-rapporteur avec Marc Giusti, Directeur de Recherches CNRS et Maître de Conférences à l'Ecole Polytechnique. Ce travail doit beaucoup aux remarques dont ils m'ont fait part. En particulier, je remercie Daniel Lazard et François Ollivier pour les théorèmes qu'ils m'ont communiqués en fin de thèse, qui élargissent la portée de mes résultats mais qui symbolisent aussi tout le temps qu'ils m'ont consacré. Je me permets également de remercier Marc Giusti pour la petite phrase d'encouragement, dite peu après un exposé ... de débutant et qui m'a aidé à garder le moral pendant les années qui

l'ont suivie.

Je remercie Giuseppa Carrà-Ferro, Professeur à l'Université de Catania, pour avoir accepté de participer à ce jury et avoir bien voulu faire l'effort de lire cette thèse, rédigée en Français.

Je remercie André Galligo, Professeur à l'Université de Nice, de m'avoir fait l'honneur de participer à ce jury.

Je remercie Sette Diop, Chargé de Recherches CNRS au LAGEP d'avoir participé à ce jury. Sa thèse, qui fut le point départ de la mienne, m'a permis d'aborder en douceur certains textes ardu.

Je remercie Rudolf Bkouche, Professeur à l'UFR de Mathématiques, de m'avoir fait l'honneur de participer à ce jury.

J'exprime ma profonde gratitude à Nour-Eddine Oussous pour le tact exceptionnel dont il fait preuve en permanence et qui assure la cohésion de l'équipe. Il a aidé cette thèse à faire ses premiers pas et me rappelle encore souvent à la réalité.

J'adresse un grand merci aux autres membres de l'équipe et à la quinzaine de turbulents occupants du bureau 214. Je souhaite bon courage à tous ceux qui doivent encore rédiger et soutenir une thèse. Je demande merci aux trop populaires cafetière et imprimante du bureau. Merci au labo, merci aux copains extérieurs au labo, merci enfin à tous les membres de ma famille qui ont été pour moi, pendant près de quatre ans, un soutien et une source d'énergie indispensables.

Villeneuve d'Ascq, le 12 juillet 1994.

Table des matières

Introduction	5
1 Notions d'algèbre différentielle	9
1.1 Polynômes différentiels	10
1.1.1 Relations d'ordre admissibles	10
1.1.2 Initiaux — séparants	14
1.2 Idéaux différentiels	14
1.2.1 Idéaux résiduels	15
1.3 Réduction	16
1.3.1 Réduction par un polynôme	16
1.3.2 Ensembles auto-réduits	20
2 Modèles d'un système d'équations et d'inéquations	25
2.1 Décomposition d'un idéal radiciel	25
2.2 Extensions différentielles	28
2.3 Modèles différentiels et modèles algébriques	29
2.3.1 Modèles différentiels	29
2.3.2 Modèles algébriques	30
2.4 Relations entre les modèles	31
2.4.1 Ensembles auto-réduits et cohérents	31
2.4.2 Systèmes réguliers	35
3 Les algorithmes d'élimination de Seidenberg	37
3.1 L'algorithme en algèbre différentielle ordinaire	38
3.1.1 Préliminaires	38
3.1.2 Définition des systèmes terminaux	41
3.1.3 Génération des systèmes terminaux	41
3.1.4 Élimination dans les systèmes terminaux	42
3.1.5 Preuve d'arrêt	43
3.1.6 Preuves de correction	46
3.1.7 Optimisations	48
3.1.8 Exemple d'application	49
3.2 L'algorithme en algèbre différentielle partielle	50
3.2.1 Définition des systèmes terminaux	50
3.2.2 Génération des systèmes terminaux	51

3.2.3	Preuves	52
3.2.4	Elimination dans les systèmes terminaux — Preuves	54
4	Bases de Gröbner — Ensembles caractéristiques	57
4.1	Bases de Gröbner algébriques	58
4.1.1	Préliminaires	58
4.1.2	Bases de Gröbner	60
4.1.3	Théorèmes utiles	61
4.2	Bases de Gröbner différentielles	63
4.2.1	Méthode de Ollivier	64
4.2.2	Méthode de Mansfield	65
4.3	Ensembles caractéristiques d'idéaux	65
5	Représentation des modèles différentiels d'un système	68
5.1	Génération des systèmes réguliers	69
5.1.1	Preuve d'arrêt	71
5.1.2	Preuves de correction	71
5.1.3	Système principal associé à Σ	72
5.2	Calcul des bases de Gröbner	73
5.3	Ce que l'algorithme fait	74
5.3.1	Décision du vide	74
5.3.2	Test d'appartenance au radical d'un idéal différentiel	74
5.3.3	Test d'appartenance à un idéal différentiel premier	74
5.3.4	Calcul d'un ensemble caractéristique	76
6	Implantations — Applications — Comparaisons	80
6.1	Les algorithmes	80
6.1.1	Algorithme Rosenfeld-Gröbner	80
6.1.2	Calcul d'ensemble caractéristique	83
6.2	Les logiciels	84
6.3	Une application à l'automatique non linéaire	89
6.4	Calcul des conditions de compatibilité	91
6.5	Décision du vide	93
6.6	Contradictions algébriques cachées	95
	Conclusion	97
A	Exemples de sessions	99
A.1	Une application à l'automatique non linéaire	99
A.2	Equations d'Euler	107
A.3	Génération de commentaires	110
	Bibliographie	118
	Index	121

Introduction

Le but de cette thèse est de rendre effectifs certains théorèmes et d'implanter efficacement certains algorithmes en algèbre différentielle, en vue d'une application à l'automatique non linéaire. Nous présentons trois résultats originaux. Le premier est un algorithme qui décrit les modèles d'un système d'équations et d'inéquations polynomiales en algèbre différentielle ordinaire comme en algèbre différentielle partielle. L'algorithme décide du vide et donc de l'appartenance au radical d'un idéal différentiel de type fini. Notre deuxième résultat est une méthode qui calcule un ensemble caractéristique d'un idéal différentiel premier donné par une famille génératrice. Nous donnons enfin de nouvelles preuves des algorithmes d'élimination de SEIDENBERG. Les algorithmes que nous décrivons sont effectifs : ils n'utilisent que l'addition, la multiplication, les dérivations et le test d'égalité à zéro dans le corps de base des polynômes.

En octobre 1990, mon directeur de thèse m'a proposé — dans le cadre d'un doctorat en informatique — d'implanter efficacement certains algorithmes d'algèbre différentielle en vue d'une application à l'automatique non linéaire. L'algorithme d'élimination de SEIDENBERG en algèbre différentielle ordinaire, par lequel cette étude a commencé, illustre parfaitement les difficultés auxquelles nous nous sommes heurtés. L'explosion combinatoire est la première puisque des entrées très simples parviennent à saturer la mémoire d'une station de travail : cet algorithme n'est praticable que jusqu'à un certain point. La seconde est la nature même des résultats produits. A la différence des algorithmes de bases de GRÖBNER par exemple, les informations fournies par l'algorithme de SEIDENBERG ne sont pas standard, leur interprétation n'est pas aisée.

Réaliser le travail demandé exigeait d'optimiser ces algorithmes, conçus sans souci d'efficacité par des algébristes du milieu du siècle, puis d'en donner des implantations spécialisées, capables d'en interpréter les résultats, au moins en partie. Nos outils : d'une part des ouvrages d'algèbre différentielle pour la compréhension des algorithmes et les preuves de correction des optimisations que nous leur apportions ; d'autre part des travaux sur les systèmes de réécriture pour les problèmes de canonicité et les preuves d'arrêt. Citons les livres de RITT [Ri] et de KOLCHIN [Ko]. Le premier est moins complet, mais plus accessible aux néophytes que le second. Nous avons consulté avec profit des articles intermédiaires [Ro] : leur contenu figure bien dans [Ko], mais sous une forme généralisée, plus abrupte. Les textes de JOUANNAUD, DERSHOWITZ [JD] et COMON [Co] nous ont beaucoup éclairé en matière de réécriture.

Introduisons maintenant plus techniquement notre sujet.

Généralisant un théorème de 1956, dû à SEIDENBERG [Sel], un mathématicien américain, ROSENFELD, donne en 1959 une condition suffisante [Ro] [Ko], pour qu'un système d'équations polynomiales différentielles admette un modèle (une solution) différentielle s'il admet un modèle purement algébrique. Ce théorème est le fil conducteur de cette thèse.

Pour l'illustrer, considérons le système différentiel suivant, en deux indéterminées u et v , où les deux dérivations δ_x et δ_y sont notées en indice :

$$\Sigma \begin{cases} u_x = 0 \\ u_y - v = 0 \\ v_x = 1 \end{cases}$$

Le système Σ ne vérifie pas la condition de ROSENFELD puisqu'il admet un modèle algébrique mais pas de modèles différentiels. En effet, si on dérive la première équation par rapport à x , la deuxième par rapport à y et si on soustrait les polynômes obtenus,

$$u_{xy} - (u_{xy} - v_x) = v_x$$

on obtient une relation qui contredit la troisième équation du système : Σ n'admet pas de modèles différentiels. Par contre, si on oublie que les équations sont différentielles et si on les interprète comme des équations purement algébriques, en quatre indéterminées u_x , u_y , v et v_x , le système admet un modèle algébrique trivial.

Fondé sur le lemme de ROSENFELD, notre premier résultat est un algorithme qui décrit les modèles d'un système d'équations polynomiales différentielles en n'employant que l'addition, la multiplication, les dérivations et le test d'égalité à zéro dans le corps de base des équations. Partant d'un système Σ , l'algorithme construit une famille (Ω_i) de systèmes ayant mêmes modèles que Σ , dont les éléments vérifient la condition de ROSENFELD. Un calcul de bases de GRÖBNER algébrique classique permet alors, et de détecter les contradictions algébriques cachées, et de décrire les modèles de chaque Ω_i .

Cet algorithme, que nous nommons ROSENFELD–GRÖBNER, a trois applications.

Il décide du vide et donc, d'après un théorème célèbre de HILBERT, de l'appartenance au radical d'un idéal différentiel de type fini.

Lorsque l'idéal différentiel $[\Sigma]$ engendré par Σ est premier, la première base de GRÖBNER non contradictoire produite par notre algorithme permet d'extraire un ensemble caractéristique (au sens de RITT) de I . Nous améliorons ainsi un résultat de OLLIVIER [Oll].

Enfin, l'algorithme ROSENFELD–GRÖBNER rend effectives de nombreuses propriétés structurelles des systèmes dynamiques en automatique non linéaire (observabilité, rang de sortie, etc . . .). Sous l'impulsion de FLIESS [F] notamment, de nombreux chercheurs ont montré comment ces propriétés pouvaient être lues dans les ensembles caractéristiques des idéaux différentiels premiers engendrés par les équations des systèmes. Nous améliorons un peu ces techniques puisque nous montrons que ces informations peuvent être lues directement dans la base de GRÖBNER de laquelle l'ensemble caractéristique est extrait.

L'originalité de notre travail, c'est de répondre aux questions ci-dessus par un algo-

rithme qui produit un résultat fini, en un nombre fini d'étapes de calcul et qui n'emploie que l'addition, la multiplication, les dérivations et le test d'égalité à zéro dans le corps de base des équations. Pour mieux situer notre apport, décrivons en quelques mots les principales méthodes existantes.

RITT a donné [Ri] une méthode pour décomposer le radical d'un idéal différentiel en une intersection d'idéaux différentiels premiers \mathcal{P}_i , en fournissant un ensemble caractéristique pour chacun des \mathcal{P}_i . Cet algorithme, qui en "fait plus" que le nôtre, présente l'inconvénient de n'être que partiellement effectif puisqu'il procède à des factorisations suivant des tours d'extensions algébriques du corps des coefficients. A notre connaissance, il n'a jamais été implanté.

OLLIVIER [O11] et CARRÀ-FERRO [Ca] ont indépendamment tenté de généraliser à l'algèbre différentielle les bases de GRÖBNER inventées par BUCHBERGER [Bu] pour l'étude des idéaux de polynômes en algèbre commutative. Ces bases de GRÖBNER différentielles sont d'agréables représentants des idéaux différentiels qu'elles engendrent, trop agréables peut-être, car l'appartenance à un idéal différentiel quelconque est un problème indécidable [GMO], l'appartenance à un idéal de type fini est un problème ouvert, et les bases de GRÖBNER différentielles ainsi définies sont en général infinies. Plus récemment, MANSFIELD [M1] a donné une définition de bases de GRÖBNER différentielles, moins proche de celles de BUCHBERGER et moins satisfaisante que celles de OLLIVIER et CARRÀ-FERRO. L'algorithme proposé par MANSFIELD calcule une base finie en un nombre fini d'étapes de calculs mais impose des restrictions sur la famille génératrice de l'idéal considéré.

Plus généraux sont les algorithmes d'élimination de SEIDENBERG [Se1]. Ils décident si un système d'équations différentielles admet des modèles différentiels en n'utilisant, comme ROSENFELD-GRÖBNER, que les opérations du corps de base des équations. Ils présentent deux inconvénients : d'une part, ils permettent difficilement de trouver les relations qui lient entr'elles les grandeurs décrites par un système dynamique et d'autre part, leur comportement est explosif parce qu'ils calculent des projections ensemblistes successives de la variété algébrique différentielle associée au système. DIOP [Di1] [Di2] a étudié l'algorithme d'élimination en algèbre différentielle ordinaire et en a donné une application à l'automatique.

Dans cette thèse, nous donnons aussi de nouvelles preuves des algorithmes d'élimination de SEIDENBERG. En algèbre différentielle partielle, nous reformulons la preuve grâce au lemme de ROSENFELD, ce qui intègre l'algorithme dans le giron des théorèmes classiques : le théorème original de SEIDENBERG, très technique, était marginal. En algèbre différentielle ordinaire, suite aux travaux de DIOP, nous fournissons une preuve très simple du mécanisme d'élimination, fondée sur un lemme de RITT qui est une version faible du lemme de ROSENFELD.

Nous ne sommes encore parvenus à déterminer, ni la complexité de l'algorithme ROSENFELD-GRÖBNER, ni celle des versions que nous donnons des algorithmes d'élimination. En algèbre différentielle ordinaire, GRIGOR'EV [G] a donné une optimisation de l'algorithme d'élimination qui fait appel à une méthode de LAZARD [L1], qui est d'une complexité presque optimale [L3]; l'algorithme de GRIGOR'EV, qui a une complexité triplement exponentielle (en temps), fournit la meilleure borne connue pour le

problème de la décision du vide en algèbre différentielle ordinaire. Nous pensons qu'il doit être possible de l'améliorer.

Nous avons implanté en langage C plusieurs versions expérimentales de l'algorithme ROSENFELD–GRÖBNER et plusieurs versions de l'algorithme d'élimination de SEIDENBERG en algèbre différentielle ordinaire. Nous employons la librairie de grands nombres PARI et le logiciel de calcul de bases de GRÖBNER GB de FAUGERE [FGLM]. Pour générer le code C, nous avons employé un petit langage, nommé CL, dont nous avons réalisé un compilateur, et qui s'apparente en fait à un préprocesseur de C, agréable d'emploi. Le but original de cette thèse était l'implantation efficace de l'algorithme d'élimination dont DIOP [Di1] [Di2] avait établi l'utilité peu auparavant. CL était un outil dans cette tâche. Il apparaît aujourd'hui que de tels projets d'implantation étaient prématurés.

Les deux premiers chapitres de cette thèse sont des chapitres d'introduction. Le premier présente quelques notions fondamentales élémentaires, ainsi qu'une sorte de division euclidienne de polynômes, étendue à l'algèbre différentielle et quelques résultats apparentés. Nous avons voulu que le chapitre soit d'un abord simple.

Le deuxième chapitre présente deux versions du Nullstellensatz de HILBERT, l'une dans le cas différentiel, l'autre dans le cas algébrique. Nous y donnons également le lemme de ROSENFELD qui les relie.

Le troisième chapitre est consacré aux algorithmes d'élimination de SEIDENBERG.

Notre algorithme utilise les bases de GRÖBNER algébriques. Nous les présentons dans le quatrième chapitre et nous établissons un parallèle succinct entre ces dernières, les bases de GRÖBNER différentielles de CARRÀ-FERRO [Ca], OLLIVIER [O11], celles de MANSFIELD [M1], et les ensembles caractéristiques d'idéaux différentiels premiers.

Le cinquième chapitre décrit notre algorithme et ses applications : décision du vide, test d'appartenance au radical d'un idéal différentiel de type fini, à un idéal différentiel premier de type fini et calcul d'un ensemble caractéristique d'un idéal différentiel premier donné par une famille génératrice finie.

Le dernier chapitre est consacré aux implantations et aux applications. Nous donnons des traces d'exécutions de nos programmes en annexe.

Chapitre 1

Notions d'algèbre différentielle

Le chapitre I introduit une sorte de division euclidienne, employée dans la plupart des algorithmes en algèbre différentielle, ainsi que quelques résultats connexes. Auparavant, nous présentons un certain nombre de notions fondamentales.

Sauf mention du contraire, les corps et les anneaux que nous considérerons seront toujours de caractéristique nulle.

On appelle *dérivation* sur un anneau A , toute fonction δ de A dans A , qui à un élément u de A , associe un élément $\dot{u} = \delta u$, et qui satisfait les axiomes deux suivants, quels que soient les deux éléments u et v de A :

- $\delta(u + v) = \dot{u} + \dot{v}$,
- $\delta(uv) = \dot{u}v + u\dot{v}$.

Si A comporte plusieurs dérivations, celles-ci sont supposées commuter entre elles.

Un *anneau* (respectivement *corps*, *algèbre*) différentiel est un anneau (respectivement *corps*, *algèbre*) muni de dérivations. Un anneau (respectivement *corps*, *algèbre*) différentiel ne comportant qu'une seule dérivations est dit *ordinaire*. Dans le cas contraire, il est dit *partiel* et nous indiquerons par m le nombre de ses dérivations : $\delta_1, \dots, \delta_m$.

Soient u et v deux éléments d'un corps différentiel K ; considérant que $\delta(v(u/v)) = \dot{u} = v\delta(u/v) + \dot{v}(u/v)$, on obtient : $\delta(u/v) = (\dot{u}v - u\dot{v})/v^2$. De $\delta 0 = \delta(0 + 0) = 2\delta 0$ et de $\delta 1 = \delta(1 \cdot 1) = 2\delta 1$, on déduit que la dérivée des entiers et de leurs fractions vaut nécessairement zéro. On appelle *constante* d'un anneau différentiel A , tout élément de A dont la dérivée est nulle. L'ensemble des constantes d'un anneau différentiel A forme un sous-anneau de A .

Notons Θ le monoïde commutatif libre engendré par les dérivations $\delta_1, \dots, \delta_m$. Les éléments de Θ sont appelés *opérateurs de dérivation* et correspondent chacun à un certain produit de puissances de dérivations $\theta = \delta_1^{\alpha_1} \dots \delta_m^{\alpha_m}$ (où les α_i sont des entiers positifs ou nuls). Si u est un élément d'un anneau différentiel A , l'expression θu désigne l'élément de A obtenu en dérivant u par δ_1 un nombre α_1 de fois etc . . . par δ_m un nombre α_m de fois. Soient $\theta = \delta_1^{\alpha_1} \dots \delta_m^{\alpha_m}$ et $\phi = \delta_1^{\beta_1} \dots \delta_m^{\beta_m}$ deux opérateurs de dérivation, $\theta\phi$ désigne l'opérateur $\delta_1^{\alpha_1+\beta_1} \dots \delta_m^{\alpha_m+\beta_m}$. La somme des exposants α_i d'un opérateur θ est appelée *ordre* de l'opérateur. Le seul élément du monoïde qui soit d'ordre nul est

l'opérateur *identité* (l'élément neutre de Θ). Un opérateur est dit *propre* s'il est d'ordre strictement positif.

Soient K un corps et E un sous-ensemble d'un anneau A , qui contient K . L'expression $K[E]$ (respectivement $K(E)$) désigne le plus petit anneau (respectivement corps) contenant K et E . Soient K un corps différentiel et E un sous-ensemble d'un anneau différentiel A , qui contient K . L'expression $K\{E\}$ (respectivement $K\langle E\rangle$) désigne le plus petit anneau différentiel (respectivement corps différentiel) contenant K et E . Notons ΘE le plus petit sous-ensemble de A qui contienne E et qui soit stable par dérivation. L'anneau $K\{E\}$ coïncide avec $K[\Theta E]$; le corps $K\langle E\rangle$ coïncide avec $K(\Theta E)$.

1.1 Polynômes différentiels

Soient K un corps différentiel et X un alphabet. $K\{X\}$ désigne l'anneau différentiel des polynômes différentiels à coefficients dans K , construit sur l'alphabet X . Les indéterminées qui figurent dans l'écriture des polynômes de $K\{X\}$ sont donc des dérivées des éléments de X .

Terminologie Tout polynôme différentiel de $K\{X\}$ est aussi un polynôme (que nous qualifierons d'*algébrique* par opposition à *différentiel*) de $K[\Theta X]$, c'est-à-dire un polynôme non différentiel dont les indéterminées sont des éléments de ΘX . Pour éviter toute confusion, nous conviendrons dans le texte qui suit de réserver le terme *indéterminée* pour désigner les éléments de ΘX . Nous emploierons le terme *lettre* pour désigner les éléments de X .

Soit x_i une lettre. Nous notons $x_{i,\theta}$ ou même $x_{i,(\alpha_1, \dots, \alpha_m)}$ l'indéterminée θx_i . En algèbre différentielle ordinaire, \dot{x}_i et \ddot{x}_i et $x_i^{(t)}$ désignent les dérivées première, deuxième, et $t^{\text{ème}}$ de x_i . Supposons par exemple le nombre m de dérivations égal à 1, on a :

$$\delta(x^2y + z + 1) = 2x\dot{x}y + x^2\dot{y} + \dot{z}.$$

Supposons $m = 2$, on a :

$$\delta_1\delta_2x^2 = \delta_1(2xx_{(0,1)}) = 2x_{(1,0)}x_{(0,1)} + 2xx_{(1,1)}.$$

1.1.1 Relations d'ordre admissibles

Il est utile de pouvoir considérer un polynôme différentiel p d'un anneau $K\{X\}$ comme un polynôme en une indéterminée u (nous supposons que p n'appartient pas à K), à coefficients dans l'anneau $K[\Theta X \setminus \{u\}]$. On privilégie ainsi une indéterminée en se donnant une relation d'ordre totale \mathcal{R} sur ΘX et on simplifie théorèmes et algorithmes en imposant que \mathcal{R} soit compatible avec l'action du monoïde Θ sur $K\{X\}$. La relation \mathcal{R} induit naturellement un préordre sur l'anneau de polynômes :

Définition 1 Soit X un alphabet. Une relation d'ordre sur ΘX est dite admissible si elle est compatible avec la structure de monoïde de Θ , c'est-à-dire si elle vérifie pour tous les éléments u et v de ΘX , et pour toute dérivation δ :

- $u < \delta u$,
- $u < v \quad \Rightarrow \quad \delta u < \delta v$.

Définition 2 Soit p un polynôme de $K\{X\}$, n'appartenant pas à K . Supposons ΘX ordonné suivant une relation d'ordre totale admissible. On appelle indéterminée principale de p la plus grande indéterminée u apparaissant avec un coefficient non nul dans l'écriture de p .

Définition 3 Soient p et q deux polynômes de $K\{X\}$, ni p , ni q n'appartenant à K . Nous dirons que p est inférieur à q si l'indéterminée principale de p est inférieure à celle de q . Si p et q ont même indéterminée principale u , nous dirons que p est inférieur à q si le degré en u de p est inférieur au degré en u de q .

Deux polynômes p et q ayant même indéterminée principale et même degré en cette indéterminée seront dits de même rang, ce que nous noterons : $p \simeq q$

Les relations d'ordre admissibles ont de bonnes propriétés : d'une part, aucun polynôme ne contient de dérivées de son indéterminée principale, d'autre part, si u est l'indéterminée principale d'un polynôme p , alors θu est l'indéterminée principale de θp (pour tout opérateur de dérivation θ). La troisième bonne propriété des relations d'ordre admissibles est la plus importante, puisqu'elle sous-tend les preuves de nombreux algorithmes et théorèmes : les ordres admissibles sont *artinien*s (lemme 3).

Définition 4 Une relation d'ordre définie sur un ensemble E est dite artinienne, si toute suite strictement décroissante d'éléments de E est finie.

Nous donnons tout de suite le lemme suivant, bien qu'il ne nous serve pas avant la section 1.3.2. La preuve est classique.

Lemme 1 Une relation d'ordre définie sur un ensemble E est artinienne si et seulement si tout sous-ensemble de E admet un élément minimal pour cette relation.

Le lemme auxiliaire ci-dessous va nous permettre d'établir que les relations d'ordre admissibles sont artiniennes, ainsi que quelques autres théorèmes importants. Nous l'emploierons également au sujet des bases de GRÖBNER, au chapitre IV.

Lemme 2 Soit m un entier positif. Dans \mathbb{N}^m , toute suite (u_n) , de terme courant $u_i = ((\alpha_i)_1, \dots, (\alpha_i)_m)$ qui vérifie la relation (1) suivante est artinienne.

$$(1) \quad j < i \quad \Rightarrow \quad \exists h, \quad 1 \leq h \leq m, \quad (\alpha_i)_h < (\alpha_j)_h$$

Preuve Par récurrence sur m . Le cas initial ($m = 1$) est immédiat. Le cas général : de toute suite infinie d'entiers naturels, on peut extraire une sous-suite infinie croissante. Il est donc possible d'extraire de (u_n) une sous-suite (v_n) infinie, croissante pour la $m^{\text{ème}}$ composante :

$$\forall i < j, \quad (\alpha_i)_m \leq (\alpha_j)_m,$$

vérifiant donc la relation (1) pour les $m - 1$ autres composantes :

$$j < i \quad \Rightarrow \quad \exists h, \quad 1 \leq h \leq m - 1, \quad (\alpha_i)_h < (\alpha_j)_h.$$

□

Lemme 3 *Soit X un alphabet fini. Tout ordre admissible sur ΘX est artинien.*

Preuve Voir [Ko], lemme 15, page 49. Supposons l'existence d'une suite infinie strictement décroissante (u_n) dans ΘX , et montrons la contradiction. Comme le cardinal de X est fini, on peut extraire de (u_n) une sous-suite (v_n) infinie et uniquement composée des dérivées d'une même lettre x de X (théorème de RAMSEY). Notons $v_i = \delta_1^{(\alpha_i)1} \dots \delta_m^{(\alpha_i)m} x$ le $i^{\text{ème}}$ terme de (v_n) . Les axiomes des ordres admissibles impliquent que si l'indice j est inférieur à i , alors v_i ne peut pas être une dérivée de v_j . En d'autres termes, la suite vérifie la relation :

$$j < i \quad \Rightarrow \quad \exists h, \quad 1 \leq h \leq m, \quad (\alpha_i)_h < (\alpha_j)_h.$$

D'après le lemme 2, (v_n) est finie. Cette contradiction prouve le lemme. □

Lemme 4 *Soit X un alphabet fini. Le préordre induit dans $K\{X\}$ par une relation d'ordre admissible sur ΘX est artинien.*

Preuve C'est une conséquence immédiate du lemme 3. □

Les axiomes de la définition 1 ne déterminent pas complètement les relations d'ordre admissibles. Voici trois exemples en algèbre différentielle ordinaire pour $m = 1$ et $X = \{x, y\}$:

1. L'indéterminée u est supérieure à v si u est une dérivée de x et v une dérivée de y ou si u et v sont deux dérivées d'une même lettre et si l'ordre de u est supérieur à celui de v :

$$\dots > \ddot{x} > \dot{x} > x > \dots > \ddot{y} > \dot{y} > y$$

2. L'indéterminée u est supérieure à v si l'ordre de u est supérieur à celui de v ou s'il existe un certain entier t tel que $u = x^{(t)}$ et $v = y^{(t)}$:

$$\dots > \ddot{x} > \ddot{y} > \dot{x} > \dot{y} > x > y$$

3. Nous laissons au lecteur le soin de formuler la définition précise de la relation d'ordre admissible qui suit :

$$\dots > x^{(5)} > \ddot{y} > x^{(4)} > \dot{y} > x^{(3)} > y > \ddot{x} > \dot{x} > x$$

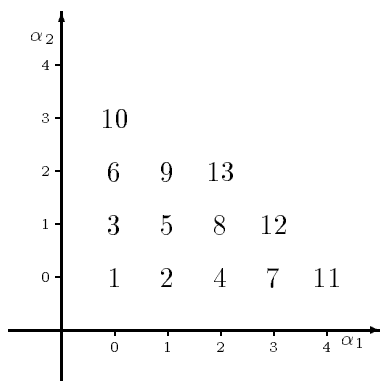
La relation illustrée par le premier exemple ci-dessus est utilisée par l'algorithme d'élimination en algèbre différentielle ordinaire; on appelle *ordre d'élimination* entre deux indéterminées x et y , un ordre admissible tel que, quels que soient les opérateurs de dérivation θ et ϕ , on ait: $\theta x > \phi y$. Les autres ordres sont dits *alternés*. En algèbre différentielle partielle, pour obtenir un ordre admissible, il faut également ordonner entre elles les dérivées d'une même lettre de X . Considérons par exemple le polynôme suivant :

$$p = x_{(1,0)} + x_{(0,1)}.$$

Voici trois façons de lui choisir une indéterminée principale :

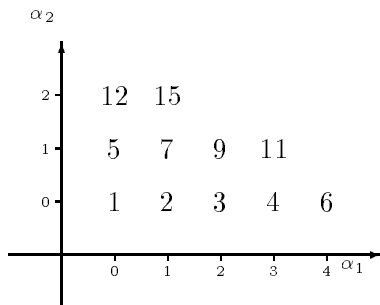
- En adoptant l'ordre lexicographique sur les composantes des opérateurs de dérivation. $x_{(1,0)}$ est alors indéterminée principale de p .
- En ordonnant les opérateurs suivant la valeur de la fonction de Cantor, qui à tout couple d'entiers (α_1, α_2) , associe: $((\alpha_1 + \alpha_2) \cdot (\alpha_1 + \alpha_2 + 1))/2 + \alpha_2 + 1$. L'indéterminée $x_{(1,0)}$ est alors indéterminée principale de p . La fonction est donnée pour $m = 2$; elle s'étend sans difficultés à $m > 2$.

Nous appellerons par la suite *ordre de Cantor*, cette relation.



- En associant aux composantes des opérateurs de dérivation, des coefficients c_i , linéairement indépendants sur \mathbb{Q} , et ordonner les $x_{(\alpha_1, \dots, \alpha_m)}$ suivant la valeur de la somme des $c_i \cdot \alpha_i$.

Nous appellerons par la suite *ordre de Seidenberg*, la relation fournie par $c_1 = \pi$ et $c_2 = 1$ (voir [Se1], §11, page 49). $x_{(0,1)}$ est alors indéterminée principale de p .



Voici un exemple assez étrange d'ordre admissible alterné en algèbre différentielle partielle :

$$\begin{aligned} \cdots > x_{(0,2)} > x_{(1,1)} > x_{(2,0)} > y_{(2,0)} > y_{(1,1)} > y_{(0,2)} > x_{(0,1)} > x_{(1,0)} > \\ & y_{(1,0)} > y_{(0,1)} > x_{(0,0)} > y_{(0,0)}. \end{aligned}$$

1.1.2 Initiaux — séparants

Soient u l'indéterminée principale d'un polynôme p de $K\{X\}$, n'appartenant pas à K , et d le degré de p en u . L'*initial* de p , noté I_p , est le coefficient de u^d dans p . Le *séparant* de p , noté S_p , est le polynôme $\partial p / \partial u$.

Lemme 5 Soient p un polynôme de $K\{X\}$, n'appartenant pas à K , d'indéterminée principale u et θ un opérateur de dérivation propre. Le polynôme θp est de degré 1 en son indéterminée principale θu . Son initial est aussi le séparant de p :

$$p = I_p \cdot u^d + R \quad \text{donne} \quad \theta p = S_p \cdot \theta u + R_\theta.$$

Preuve Par récurrence sur θ . Le cas général : supposons $\theta = \delta \phi$ et supposons (hypothèse de récurrence) que le polynôme $\phi p = S_p \cdot \phi u + R_\phi$ ait ϕu pour indéterminée principale. Nous avons alors :

$$\theta p = (\delta S_p) \cdot \phi u + S_p \cdot \theta u + \delta R_\phi.$$

Vu les propriétés des ordres admissibles, l'indéterminée principale de θp est θu . Montrons le cas initial, où θ est d'ordre 1. Soit p le polynôme suivant :

$$p = A_d \cdot u^d + \cdots + A_1 \cdot u + A_0.$$

Quelle que soit la dérivation δ , on a :

$$\begin{aligned} \delta p &= (d \cdot A_d \cdot u^{d-1} \cdot \delta u + \cdots + A_1 \cdot \delta u) + (\delta A_d) \cdot u^d + \cdots + (\delta A_1) \cdot u + (\delta A_0) \\ \delta p &= S_p \cdot \delta u + R. \end{aligned}$$

Vu les propriétés des ordres admissibles, l'indéterminée principale de δp est δu , son initial est S_p . \square

1.2 Idéaux différentiels

Le seul but de cette section est de fournir des notations pour la section 1.3. Les idéaux de polynômes sont étudiés de façon plus approfondie au chapitre suivant.

Rappelons qu'on appelle *idéal* d'un anneau A , tout sous-groupe additif de l'anneau, stable par multiplication par un élément quelconque de A . L'ensemble des idéaux de A est fermé par intersection. Soit E un sous-ensemble de $K\{X\}$. On note (E) l'*idéal engendré* par E , qui est égal à l'intersection des idéaux contenant E .

Soient I un idéal, a et b deux éléments de $K\{X\}$. Nous notons $a \equiv b \pmod{I}$ pour signifier que la différence $a - b$ est dans I .

Un *idéal différentiel* d'un anneau différentiel A est un idéal de A , clos par dérivation. L'ensemble des idéaux différentiels de A est fermé par intersection. Soit E un sous-ensemble de $K\{X\}$. On note $[E]$ l'*idéal différentiel engendré* par E , qui est égal au plus petit idéal différentiel contenant E (l'intersection de tous les idéaux différentiels contenant E).

Lemme 6 *Soit E un sous-ensemble de $K\{X\}$. On a : $[E] = (\Theta E)$.*

Voir [Ri] I, §7. Ce lemme indique bien une des difficultés algorithmiques rencontrées. Décider de l'appartenance à un idéal différentiel engendré par un unique polynôme p , c'est décider de l'appartenance à un idéal engendré par une famille infinie de polynômes : p et ses dérivées. En algèbre différentielle ordinaire par exemple, considérons deux polynômes p et q de $K\{X\}$. Dire que q appartient à $[p]$, c'est dire qu'il existe des polynômes A_i et un entier t , dépendant de q , tels que :

$$q = A_0 \cdot p + A_1 \cdot \dot{p} + \cdots + A_t \cdot p^{(t)}.$$

Les tests d'appartenance sont donc nettement plus complexes en algèbre différentielle qu'en algèbre commutative. On notera à cet égard que l'appartenance à un idéal différentiel quelconque est un problème indécidable et que l'appartenance à un idéal différentiel de type fini est un problème ouvert (voir [GMO]).

1.2.1 Idéaux résiduels

Définition 5 *Soit I un idéal et E un sous-ensemble de $K\{X\}$. On appelle résiduel de I par E , l'ensemble :*

$$I : E = \{p \in K\{X\} \mid \exists a \in E, a \cdot p \in I\}.$$

Lemme 7 *Le résiduel d'un idéal différentiel par une famille multiplicativement stable, est un idéal différentiel.*

Preuve Soient I un idéal différentiel, E une famille multiplicativement stable de $K\{X\}$ et p et q deux éléments de $I : E$. Il existe donc deux éléments a et b de E tels que $a \cdot p$ et $b \cdot q$ appartiennent à I . Comme I est un idéal, $(a \cdot b \cdot (p + q))$ appartient à I , et comme E est multiplicativement stable, $(p + q)$ appartient à $I : E$. De même, quel que soit l'élément r de $K\{X\}$, le polynôme $(a \cdot p \cdot r)$ appartient à I et $I : E$ est un idéal. Établissons pour conclure que $I : E$ est stable par dérivation : $a \cdot \delta(a \cdot p)$ est égal à $a^2 \cdot \delta p + a \cdot (\delta a) \cdot p$. Le deuxième terme de la somme est dans l'idéal, donc le premier y est aussi. Comme a^2 appartient à E , on conclut que $I : E$ est un idéal différentiel. \square

Dans la pratique, nous aurons souvent besoin de considérer des idéaux engendrés par une famille finie de polynômes $A = A_1, \dots, A_r$ résiduels par les initiaux ou les séparants des éléments de A . Nous noterons alors H_A le produit des initiaux et des séparants des

A_ℓ et H_A^∞ l'ensemble de tous les produits de puissances des initiaux et des séparants des A_ℓ (c'est-à-dire la plus petite partie de $K\{X\}$ qui soit multiplicativement stable, et qui contienne 1 et H_A). Le résiduel de l'idéal différentiel engendré par A , par les initiaux et les séparants des A_ℓ sera noté :

$$[A] : H_A^\infty.$$

1.3 Réduction

Nous définissons la notion de polynôme réduit par rapport à un autre, nous décrivons ensuite un algorithme de réduction (un équivalent différentiel de la division euclidienne); enfin nous étudions les ensembles de polynômes auto-réduits et avec eux, la notion d'ensemble caractéristique.

Définition 6 Soient p et q deux polynômes de $K\{X\}$, p n'appartenant pas à K . Le polynôme q est dit partiellement réduit par rapport à p si aucune dérivée propre de l'indéterminée principale de p n'apparaît dans l'écriture de q .

Le polynôme q est dit réduit par rapport à p s'il est partiellement réduit par rapport à p et s'il est de degré inférieur à celui de p , en l'indéterminée principale de p .

Par extension, nous dirons d'un polynôme p de $K\{X\}$, qu'il est *réduit par rapport à un sous-ensemble E* de $K\{X\}$, s'il est réduit par rapport à chaque élément de E .

Remarque Il ne faut pas confondre les relations “est réduit par rapport à” et “est inférieur à” : soient p, q deux polynômes de $K\{X\}$. Clairement, si p est inférieur à q alors p est réduit par rapport à q , mais la réciproque n'est pas vraie : la relation “est réduit par rapport à” n'est ni transitive (par exemple, pour tout ordre admissible, $(x+z)$ est réduit par rapport à y et y est réduit par rapport à z , mais $(x+z)$ n'est pas réduit par rapport à z) ni artinienne (x est réduit par rapport à y , qui est réduit par rapport à x etc ...). Des conséquences de la non transitivité de cette relation sont abordées dans la section 4.3.

1.3.1 Réduction par un polynôme

Soient p et q deux polynômes de $K\{X\}$, p n'appartenant pas à K . Nous donnons les spécifications d'une classe d'algorithmes qui calculent par soustractions successives un polynôme r , réduit par rapport à p , vérifiant :

$$(1) \quad I_p^\alpha \cdot S_p^\beta \cdot q \equiv r \pmod{[p]}$$

où α et β sont des entiers positifs ou nuls.

Description Appelons u l'indéterminée principale de p et d le degré de p en u . D'après les axiomes des ordres admissibles, il existe un polynôme T_p , réduit par rapport à p tel que l'on ait :

$$p = I_p \cdot u^d + T_p.$$

Qui plus est, pour tout opérateur de dérivation propre θ , le lemme 5 (page 14) nous indique qu'il existe un polynôme $T_{\theta,p}$, réduit par rapport à θp tel que l'on ait :

$$\theta p = S_p \cdot \theta u + T_{\theta,p}.$$

Réduire par p , c'est utiliser les deux égalités ci-dessus comme des règles de réécriture :

$$(R1) \quad u^d \rightarrow \frac{-T_p}{I_p}$$

$$(R2) \quad \theta u \rightarrow \frac{-T_{\theta,p}}{S_p}.$$

En pratique, les algorithmes procèdent à des multiplications par l'initial et le séparant de p , afin d'éviter des manipulations de fractions et constituent ainsi une sorte de pseudo-division euclidienne, étendue à l'algèbre différentielle.

Soit q le polynôme à réduire par p . Appelons w la plus grande dérivée de u , non nécessairement propre, qui apparaisse dans l'écriture de q . Si w est égale à u , on applique la règle (R1). Le reste r est de degré en u inférieur à celui de p et vérifie un cas particulier de (1) :

$$(2) \quad I_p^\alpha \cdot q = A \cdot p + r.$$

Si w est une dérivée propre de u ($w = \theta u$), on applique la règle (R2). Le reste r est de degré en w inférieur à celui de θp , mais comme ce dernier est égal à 1, le polynôme r ne contient pas w , et vérifie :

$$(3) \quad S_p^\alpha \cdot q = A \cdot \theta p + r.$$

En répétant cette opération un nombre fini (au plus égal à l'ordre de θ) de fois, l'algorithme produit un polynôme r , partiellement réduit par rapport à p , que nous appellerons *reste partiel* de la division de q par p . Celui-ci satisfait la relation :

$$(4) \quad S_p^\beta \cdot q \equiv r \pmod{[p]}.$$

Pour que r soit réduit par rapport à p , il suffit d'appliquer (R1) une fois de plus.

Remarque Sur la façon de déterminer w , son degré d et le coefficient de w^d dans q .

Supposons que u soit la dérivée d'une certaine lettre x de l'alphabet X . En algèbre différentielle ordinaire, si quels que soient les opérateurs θ , ϕ et l'indéterminée y de X ($y \neq x$), on a : $\theta x > \phi y$ (par exemple si X est ordonné suivant un ordre d'élimination), alors w est nécessairement égale à l'indéterminée principale v de q . La détermination de w de d et du coefficient de w^d est immédiate dans ce cas, mais sans ces hypothèses, w peut se distinguer de v à n'importe quelle étape de la réduction. C'est ce que montrent

les exemples suivants.

En algèbre différentielle ordinaire par exemple, pour l'ordre $\dots > \ddot{y} > \dot{x} > \dot{y} > x > y$, calculons $(\dot{x} + x) \text{ rem } yx$ (normalement égal à 0). Initialement, $v = w = \dot{x}$. L'algorithme commence par diviser $(\dot{x} + x)$ par la dérivée première de yx , égale à $(y\dot{x} + x\dot{y})$ et produit un premier reste: $(-x\dot{y} + yx)$. Dès après la première étape, $w = x$ n'est plus l'indéterminée principale $v = \dot{y}$ du dividende. En algèbre différentielle partielle, pour $m = 2$, on a : $(x_{(0,2)} + x_{(1,0)} + x_{(0,1)}) \text{ rem } x_{(0,1)} = x_{(1,0)}$. Si on adopte l'ordre de Cantor sur $\Theta\{x\}$, dès la première soustraction, $x_{(1,0)}$, qui n'est pas une dérivée de $u = x_{(1,0)}$, devient l'indéterminée principale du dividende.

Pour s'assurer que v soit toujours égale à w , on peut en algèbre différentielle ordinaire, réordonner q avant calcul, suivant un ordre d'élimination *ad hoc*. En algèbre différentielle partielle, il n'existe pas d'ordre admissible qui, quel que soit q , assure que w soit toujours indéterminée principale. On peut alors réordonner q suivant un ordre non admissible, qui privilégie les dérivées de u .

L'algorithme ci-dessous suppose que v et w coïncident en permanence.

```

fonction    rem
paramètres  $p \in K\{X\}, p \notin K$ 
               $q \in K\{X\}$ 
résultat    $r = q \text{ rem } p$ 
{  $u$  et  $v$  désignent les indéterminées principales respectives de  $p$  et de  $r$  }
début
     $r := q$ 
    tant que  $v > u$  faire
        soit  $\theta$  tel que  $v = \theta u$ 
         $r :=$  pseudo-reste euclidien de  $r$  par  $\theta p$ 
    fin tant que
     $r :=$  pseudo-reste euclidien de  $r$  par  $p$ 
fin

```

```

fonction    pseudo-reste euclidien
paramètres  $p \in K\{X\}, p \notin K$ 
               $q \in K\{X\}$ 
résultat    $r =$  pseudo-reste euclidien de  $q$  par  $p$ 
{  $p$  et  $q$  ont même indéterminée principale :  $u$  }
début
     $r := q$ 
    tant que  $\deg_u r > \deg_u p$  faire
         $d := \deg_u r - \deg_u p$ 
         $r := I_p \cdot r - I_r \cdot u^d \cdot p$ 
    fin tant que
fin

```

A propos de l'algorithme

- En algèbre différentielle ordinaire, le reste partiel de la division de q par p est

égal au reste de la division de q par la dérivée première de $p : q \text{ rem } \dot{p}$. Rappelons que dans tous les cas, ce reste vérifie un cas particulier de (1) :

$$S_p^\beta \cdot q \equiv r \pmod{[p]}$$

avec r partiellement réduit par rapport à p .

- Si les polynômes p et q ont même indéterminée principale u , le reste vérifie une version simplifiée de la relation (1) :

$$I_p^\alpha \cdot q \equiv r \pmod{[p]}$$

avec r réduit par rapport à p .

- Que le reste de la division de q par p soit nul n'implique pas que q appartienne à l'idéal différentiel $[p]$ engendré par p . Par exemple, $x \text{ rem } yx$ vaut zéro, bien que x n'appartienne pas à $[yx]$. Par contre, q appartient alors à $[p] : H_p^\infty$, où H_p^∞ désigne l'ensemble de tous les produits de puissances de l'initial et du séparant de p .

$$q \text{ rem } p = 0 \quad \Rightarrow \quad q \in [p] : H_p^\infty.$$

- Soit $r = q \text{ rem } p$. En général, r n'est pas équivalent à q , modulo $[p] : H_p^\infty$. On démontre facilement par contre l'équivalence :

$$q \in [p] : H_p^\infty \quad \Leftrightarrow \quad r \in [p] : H_p^\infty.$$

- Le reste de la division euclidienne classique (polynômes algébriques à coefficients dans un corps) est unique. Ici, tel n'est pas le cas : il y a en général une infinité de polynômes qui vérifient (1). En particulier, il se peut fort bien que certains soient nuls et d'autres non. En appliquant par exemple l'algorithme ci-dessus, on a :

$$(x\dot{x} + x) \text{ rem } x^2 = 0.$$

En effet, l'algorithme commence par diviser $(x\dot{x} + x)$ par la dérivée première de x^2 (à un facteur constant près : $x\dot{x}$), sans chercher à déterminer si le séparant de x^2 est un facteur de $(x\dot{x} + x)$:

$$x \cdot (x\dot{x} + x) - x \cdot (x\dot{x}) = x^2.$$

Il calcule ensuite le reste de la division de x^2 par lui-même, et produit bien évidemment zéro :

$$x^2 - x^2 = 0.$$

Un algorithme de pseudo-division qui chercherait à éviter des multiplications inutiles calculerait par contre un reste non nul :

$$(x\dot{x} + x) \text{ rem } x^2 = x.$$

Par la séquence :

$$(x\dot{x} + x) - (x\dot{x}) = x,$$

$$x \text{ rem } x^2 = x.$$

1.3.2 Ensembles auto-réduits

Les ensembles auto-réduits (voir [Ko], I, §9) correspondent aux *chaînes*, introduites par RITT (voir [Ri], I, §4). La principale différence entre ces deux notions est qu'une *chaîne* est finie par définition, alors qu'on démontre que tout ensemble auto-réduit est fini (théorème 1). Les théorèmes démontrés dans cette section seront utilisés pour prouver l'arrêt des algorithmes utilisés dans les prochains chapitres. Ils permettent également de prouver certains théorèmes fondamentaux, comme le théorème de la base finie de RITT-RAUDENBUSH (voir [Ri], I, §12).

Définition 7 *Un sous-ensemble E de $K\{X\}$ est dit auto-réduit si aucun élément de E n'appartient à K et si chaque élément de E est réduit par rapport aux autres.*

A propos des ensembles auto-réduits

- Les polynômes d'un ensemble auto-réduit ont des indéterminées principales différentes.
- Un ensemble peut fort bien être auto-réduit pour un certain ordre, mais pas pour un autre. L'ensemble suivant est auto-réduit pour l'ordre de Cantor,

$$E = \{x_{(4,0)} + x_{(2,2)}, x_{(1,0)} \cdot x_{(3,2)}\}$$

les indéterminées principales sont $x_{(4,0)}$ et $x_{(3,2)}$, mais par pour l'ordre de Seidenberg, pour lequel l'indéterminée principale du second polynôme, $x_{(3,2)}$, est une dérivée de celle du premier, $x_{(2,2)}$.

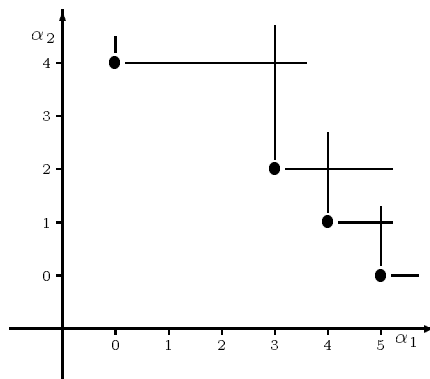
- Pour $m = 2$, les propriétés des ordres admissibles impliquent que les indéterminées principales des éléments d'un ensemble auto-réduit engendrent un "escalier". Considérons par exemple,

$$E = \{x_{(0,4)} + x_{(2,2)}, x_{(1,0)} \cdot x_{(3,2)}, x_{(4,1)}, x_{(5,0)}^3\}.$$

Pour l'ordre de Cantor, les indéterminées principales des polynômes sont

$$\{x_{(0,4)}, x_{(3,2)}, x_{(4,1)}, x_{(5,0)}\}$$

qui engendrent l'"escalier" suivant :



Théorème 1 *Soit X un alphabet fini. Tout sous-ensemble auto-réduit de $K\{X\}$ est fini.*

Preuve Supposons l'existence d'un ensemble auto-réduit infini A , et montrons la contradiction. L'ensemble B des indéterminées principales des éléments de A est infini (voir remarque ci-dessus). Comme X est fini, B contient au moins un sous-ensemble infini C d'indéterminées qui soient toutes dérivées d'une même lettre x de X . D'après la définition de la réduction, toute paire $u = \delta_1^{\alpha_1} \cdots \delta_m^{\alpha_m} x$ et $v = \delta_1^{\beta_1} \cdots \delta_m^{\beta_m} x$ d'éléments de C , vérifie :

$$\exists h, \quad 1 \leq h \leq m, \quad \alpha_h < \beta_h.$$

D'après le lemme 2 (page 11), C est fini. \square

- Pour $m = 2$ et pour une seule indéterminée, on peut borner le cardinal d'un ensemble auto-réduit dès qu'on en connaît un élément, comme le montrent les diagrammes en escalier. La borne est égale à l'ordre de l'indéterminée principale de l'élément. Ce résultat n'est pas valable pour $m = 3$. L'ensemble suivant est auto-réduit, pour toute valeur de h :

$$\{x_{(1,1,1)}, x_{(h,0,0)}, x_{(h-1,1,0)}, \dots, x_{(0,h,0)}\}.$$

Par la suite nous noterons les ensembles auto-réduits sous la forme d'une séquence $A = A_1, A_2, \dots, A_r$, ordonnée par ordre croissant : $A_1 < A_2 < \dots < A_r$.

Définition 8 *Soient $A = A_1, \dots, A_r$ et $B = B_1, \dots, B_s$ deux sous-ensembles auto-réduits de $K\{X\}$. Nous dirons que A est inférieur à B si A et B remplissent l'une des deux conditions suivantes :*

1. $A_j \simeq B_j$ et $A_i < B_i$, $1 \leq j < i \leq \min(r, s)$,
2. $A_j \simeq B_j$ et $r > s$, $1 \leq j \leq \min(r, s)$.

Si r est égal à s et si pour toute valeur de i , comprise entre 1 et s , le polynôme A_i a même rang que B_i , nous dirons que les ensembles A et B ont même rang.

Il s'agit d'une sorte de préordre lexicographique pour lequel les ensembles les plus volumineux sont les plus petits.

Théorème 2 *Soit E un sous-ensemble de $K\{X\}$. Nous supposons qu'aucun élément non nul de E n'appartient à K . Un sous-ensemble auto-réduit A de E est un ensemble caractéristique de E s'il vérifie l'une des deux conditions équivalentes suivantes :*

1. A est un sous-ensemble auto-réduit minimal de E .
2. E ne comporte aucun élément non nul réduit par rapport à A .

Preuve 1) \Rightarrow 2) Supposons l'existence d'un polynôme p dans E , réduit par rapport à A et montrons qu'il existe un sous-ensemble auto-réduit de E , inférieur à A . Si p est supérieur à tous les éléments de A , la séquence A_1, \dots, A_r, p forme un ensemble auto-réduit (voir la remarque qui suit la définition 6 (page 16)) et inférieur à A par la condition 2) de la définition 8. Dans le cas contraire, soient A_1, \dots, A_s les éléments de A inférieurs à p . La séquence A_1, \dots, A_s, p constitue un ensemble auto-réduit de E , inférieur à A par la condition 1) de la définition 8.

2) \Rightarrow 1) Supposons la séquence A inférieure à la séquence B et montrons qu'il existe dans A un polynôme réduit par rapport à B . Si A est inférieure à B par la condition 1) de la définition 8 pour un certain indice i , alors le polynôme A_i est réduit par rapport à B . Si A est inférieure à B par la condition 2) de la définition 8, alors le polynôme A_{s+1} est réduit par rapport à B , où s désigne le cardinal de B . \square

Corollaire *Soit A un ensemble caractéristique d'un sous-ensemble E de $K\{X\}$ et p un polynôme de $K\{X\}$, réduit par rapport à A . Tout ensemble caractéristique de $E \cup \{p\}$ est inférieur à A .*

Les preuves d'arrêt des algorithmes étudiés dans les prochains chapitres utilisent le corollaire ci-dessus conjointement au fait que les initiaux et les séparants des éléments d'un ensemble caractéristique A sont réduits par rapport à A . De même, soit $A_\ell = I_\ell \cdot u^d + R_\ell$ l'un des éléments de A , les polynômes R_ℓ et $(d \cdot A_\ell - u \cdot S_\ell)$ sont réduits par rapport à A . Comme ces polynômes sont inférieurs à A_ℓ , si on substitue l'un d'eux à A_ℓ dans A , on obtient un ensemble auto-réduit inférieur à A .

Théorème 3 *Tout ensemble de polynômes admet un ensemble caractéristique. En d'autres termes, toute suite strictement décroissante d'ensembles auto-réduits est finie.*

Preuve Voir [Ko], I, §10, page 81. Soient E inclus dans $K\{X\}$ et F l'ensemble des sous-ensembles auto-réduits de E . Considérons la suite $(F_n) = F_1 \supset F_2 \supset \dots$ de sous-ensembles de F , définie comme suit : F_1 est l'ensemble de tous les éléments A_1, \dots, A_r de F , pour lesquels A_1 est minimal (l'existence d'un élément minimal est due aux lemmes 1 et 4). F_i est l'ensemble de tous les éléments $A_1, \dots, A_i, \dots, A_r$ ($r \geq i$) de F_{i-1} , pour lesquels A_i est minimal. D'après le théorème 1, il existe un indice h tel que, pour tout i supérieur à h , F_i est vide. Tout élément de F_h est un ensemble caractéristique de E . \square

Un ensemble E peut admettre plusieurs ensembles caractéristiques qui sont bien sûr, tous de même rang. Si E est fini (et si on connaît ses éléments), le preuve ci-dessus nous donne même un algorithme de calcul d'un ensemble caractéristique de E .

Réduction par un ensemble auto-réduit

Soient q un polynôme et $A = A_1, \dots, A_r$ un sous-ensemble auto-réduit de $K\{X\}$. Nous souhaitons étendre l'algorithme de la section 1.3.1 (page 16), pour calculer un polynôme $r = q \text{ rem } A$, réduit par rapport à A (c'est-à-dire par rapport à chaque A_i),

vérifiant :

$$H_A^\alpha \cdot q \equiv r \pmod{[A]},$$

où H_A^α désigne un certain produit de puissances d'initiaux et de séparants des A_i .

RITT décrit ([Ri], I, §6, page 6) un algorithme efficace en algèbre différentielle ordinaire, lorsque ΘX est ordonné suivant un ordre d'élimination : le reste r peut se calculer en réduisant successivement q par A_r , puis par A_{r-1} , etc ... jusqu'à A_1 .

$$r = ((q \text{ rem } A_r) \cdots \text{ rem } A_1).$$

Avec de telles conditions en effet, on a :

$$(i < j \text{ et } q \text{ réduit par rapport à } A_j) \Rightarrow (q \text{ rem } A_i) \text{ réduit par rapport à } A_j,$$

mais cette propriété n'est pas vraie pour un ordre admissible quelconque, comme le montrent les exemples suivants :

Exemple 1 en algèbre différentielle ordinaire, pour l'ordre $\cdots > \dot{x} > \dot{y} > x > y$, les polynômes $A_1 = x + y$ et $A_2 = \dot{y}$ forment un ensemble auto-réduit. Le polynôme \dot{x} est réduit par rapport à A_2 , mais $(\dot{x} \text{ rem } A_1) = -\dot{y}$ ne l'est plus.

Exemple 2 en algèbre différentielle partielle, pour l'ordre de Cantor, les polynômes $A_1 = x_{(4,1)} + x_{(1,4)}$ et $A_2 = x_{(3,3)}$ forment un ensemble auto-réduit. Le polynôme $x_{(6,1)}$ est réduit par rapport à A_2 , mais $(x_{(6,1)} \text{ rem } A_1) = -x_{(3,4)}$ ne l'est plus.

KOLCHIN fournit un algorithme ([Ko], I, §9, page 77) plus général que celui de RITT, mais aussi beaucoup plus lent et pénible à mettre en œuvre. Soit à réduire q par $A = A_1, \dots, A_r$, la méthode de KOLCHIN consiste à réduire d'abord partiellement q par A , puis à conclure la réduction par une suite de calculs de restes algébriques¹ :

premièrement calculer $q' = q \text{ rem-partiel } A$. Ce polynôme, partiellement réduit par rapport à A (c'est-à-dire par rapport à chaque A_i) vérifie :

$$S_A^\alpha \cdot q \equiv q' \pmod{[A]}$$

où S_A est le produit des séparants des A_i , et α un entier naturel.

deuxièmement calculer $r = ((q' \text{ rem } A_r) \cdots \text{ rem } A_1)$.

L'algorithme de réduction partielle de q par A consiste à répéter le procédé suivant :

1. déterminer la plus grande indéterminée, que nous noterons v , qui apparaisse dans l'écriture de q , et qui soit également une dérivée propre de l'indéterminée principale d'un A_i .

1. il semble que ce calcul en deux étapes engendre moins de réductions élémentaires qu'un calcul qui mélangerait réductions partielles et réductions algébriques.

2. déterminer le plus grand indice ℓ tel que v soit la dérivée θ de l'indéterminée principale de A_ℓ .
3. réduire q par θA_ℓ .

On remarquera que lors de l'étape 3, q n'est pas réduit par A_ℓ , mais par θA_ℓ . On prouve aisément l'arrêt de la phase de réduction partielle grâce au lemme 3 (page 12), puisqu'après chaque réduction, l'indéterminée v décroît.

Il est possible d'optimiser légèrement l'algorithme de KOLCHIN en partageant l'ensemble auto-réduit A :

$$A = \underbrace{A_1, \dots, A_i, \dots}_{B_1}, \dots, \underbrace{A_j, \dots, A_r}_{B_s},$$

deux polynômes A_i et A_j ($i < j$) de A appartiennent à un même ensemble B_ℓ si il existe un opérateur θ tel que θA_i soit supérieur à A_j . Il est clair que les A_i contenus dans un ensemble B_ℓ ont des indices consécutifs dans A . On peut démontrer assez facilement que réduire par A , c'est réduire (à la façon de KOLCHIN) successivement par B_s , puis par B_{s-1} , etc ... jusqu'à B_1 :

$$r = ((q \text{ rem } B_s) \cdots \text{ rem } B_1).$$

Cette méthode devient efficace lorsque certains ensembles B_ℓ sont réduits à un seul polynôme, où réduire par B_ℓ , c'est tout simplement appliquer l'algorithme de la section 1.3.1 (page 16). De cette façon, en algèbre différentielle ordinaire et pour un ordre d'élimination, on obtient l'algorithme de RITT.

Chapitre 2

Modèles d'un système d'équations et d'inéquations

Ce chapitre établit deux versions “faibles” (une dans le cas différentiel, une dans le cas algébrique) du théorème des zéros de HILBERT : “Décider si un système d'équations et d'inéquations polynomiales différentielles admet des modèles, c'est décider si l'inéquation appartient au radical de l'idéal engendré par les équations”. Il se trouve que pour certains systèmes d'équations et d'inéquations, l'existence de modèles algébriques implique l'existence de modèles différentiels. Nous donnons une condition suffisante, due à ROSENFELD, pour qu'un système d'équations et d'inéquations ait cette propriété.

2.1 Décomposition d'un idéal radiciel

Le théorème des zéros de HILBERT se décompose en deux théorèmes. Le premier affirme que le radical d'un idéal (différentiel) est un idéal (différentiel) ; le second que tout idéal (différentiel) radiciel est une intersection d'idéaux (différentiels) premiers.

Dans cette section, nous donnons des démonstrations de ces deux théorèmes.

Définition 9 *Un idéal I (algébrique ou différentiel) de $K\{X\}$ est dit premier s'il est inclus strictement dans $K\{X\}$ et s'il vérifie pour tous polynômes p et q de $K\{X\}$:*

$$p \cdot q \in I \quad \Rightarrow \quad (p \in I \text{ ou } q \in I).$$

Remarque un ensemble de générateurs peut fort bien engendrer un idéal différentiel qui soit premier mais un idéal algébrique qui ne le soit pas : l'idéal $[\dot{x}^2, x]$ est premier (il est égal à $[x]$), alors que (\dot{x}^2, x) ne l'est pas. La réciproque est vraie également : l'idéal $(\dot{x}^2 + x)$ est premier (puisque $\dot{x}^2 + x$ est algébriquement irréductible sur \mathbb{Q}) mais $[\dot{x}^2 + x]$ ne l'est pas. En effet, la dérivée de $\dot{x}^2 + x$ est égale à $\dot{x} \cdot (2\ddot{x} + 1)$, mais — et nous aurons là une belle occasion d'appliquer l'algorithme d'élimination en algèbre différentielle ordinaire (voir la section 3.1.8 (page 49)) — ni \dot{x} , ni $2\ddot{x} + 1$ n'appartiennent à $[\dot{x}^2 + x]$.

L'intersection de deux idéaux premiers forme un idéal *radiciel*:

Définition 10 Un idéal I (algébrique ou différentiel) de $K\{X\}$ est dit radiciel s'il vérifie, pour tout polynôme p de $K\{X\}$:

$$(\exists n \in \mathbb{N}, \quad p^n \in I) \quad \Rightarrow \quad p \in I.$$

Les idéaux radiciels sont clos par intersection. Soit E un sous-ensemble de $K\{X\}$. On note $\{E\}$ le plus petit idéal différentiel radiciel contenant E .

Lemme 8 Soit E un sous-ensemble de $K\{X\}$. Le plus petit idéal différentiel radiciel contenant E est égal au radical du plus petit idéal différentiel contenant E . En d'autres termes, $\{E\} = \sqrt{[E]}$.

Preuve Comme $\{E\}$ contient $\sqrt{[E]}$, il suffit de montrer que $\sqrt{[E]}$ est un idéal différentiel:

Soient p et q deux éléments de $K\{X\}$ tels que p^α appartienne à $[E]$. Il est clair que $(p \cdot q)^\alpha$ appartient à $[E]$, donc que $\sqrt{[E]}$ est stable par produit par un élément quelconque de $K\{X\}$. Soient p et q deux éléments de $K\{X\}$ tels que p^α et q^β appartiennent à $[E]$. $(p+q)^{\alpha+\beta-1}$ est une somme de produits de puissances $p^i \cdot q^j$ où $i \geq \alpha$ ou $j \geq \beta$. Chaque terme appartient à $[E]$ et, d'après ce qui précède, leur somme également. $\sqrt{[E]}$ est donc un idéal.

Soient p un élément de $K\{X\}$, tel que p^α appartienne à $[E]$ et δ une dérivation. Montrons qu'il existe alors un entier β tel que $(\delta p)^\beta$ appartienne à $[E]$.

$$\begin{aligned} p^\alpha &\in [E] \quad \text{donc (en dérivant par } \delta): \\ p^{\alpha-1} \cdot \delta p &\in [E] \quad \text{et} \\ (\alpha-1) \cdot p^{\alpha-2} \cdot (\delta p)^2 + p^{\alpha-1} \cdot \delta^2 p &\in [E]. \end{aligned}$$

En multipliant le dernier polynôme par δp , on obtient

$$(\alpha-1) \cdot p^{\alpha-2} \cdot (\delta p)^3 + p^{\alpha-1} \cdot (\delta p) \cdot \delta^2 p \in [E].$$

Le deuxième terme de la somme contient en facteur un élément de $[E]$ (égal à $p^{\alpha-1} \cdot \delta p$). Le premier terme est donc lui-aussi dans l'idéal:

$$p^{\alpha-2} \cdot (\delta p)^3 \in [E].$$

En répétant ce procédé α fois, on obtient:

$$(\delta p)^{2\alpha-1} \in [E].$$

$\sqrt{[E]}$ est donc un idéal différentiel. \square

A propos du lemme

- Le lemme n'est pas vrai en caractéristique non nulle or, que le radical d'un idéal soit un idéal est une conséquence du théorème des zéros. Les algorithmes donnés dans cette thèse ne pourront donc pas se généraliser en caractéristique différente de zéro. SEIDENBERG a considérablement étudié ce problème (voir [Se1], [Se2] et une remarque de KOLCHIN [Ko], II, page 86). Il n'est pas vrai non plus dans un anneau différentiel A quelconque: il faut que A soit un anneau de RITT, c'est-à-dire qu'il contienne \mathbb{Q} .

Les deux lemmes qui suivent sont destinés à démontrer le théorème 4.

Lemme 9 *Soient p et q deux polynômes de $K\{X\}$. Quels que soient les opérateurs de dérivation θ et ϕ on a $\theta p \cdot \phi q \in \{p \cdot q\}$.*

Preuve Voir [Ri] I, §10. Supposons inductivement que $\theta = \delta\psi$ et que le polynôme $(\psi p \cdot \phi q)$ appartienne à l'idéal.

$$\begin{aligned} \psi p \cdot \phi q &\in \{p \cdot q\} && \text{donc (en dérivant par } \delta): \\ \theta p \cdot \phi q + \psi p \cdot \delta\phi q &\in \{p \cdot q\}. \end{aligned}$$

En multipliant le dernier polynôme par ϕq , on obtient

$$\theta p \cdot (\phi q)^2 + \psi p \cdot \phi q \cdot \delta\phi q \in \{p \cdot q\}$$

et donc aussi (après multiplication par θp):

$$(\theta p \cdot \phi q)^2 + \theta p \cdot \psi p \cdot \phi q \cdot \delta\phi q \in \{p \cdot q\}.$$

Le deuxième terme de la somme contient en facteur un élément de $\{p \cdot q\}$ (égal à $\psi p \cdot \phi q$). Le premier terme est donc lui-aussi dans l'idéal. Comme ce dernier est radiciel, on peut supprimer l'exposant :

$$\theta p \cdot \phi q \in \{p \cdot q\}.$$

□

Lemme 10 *Soient E un sous-ensemble, p et q deux polynômes de $K\{X\}$. Notons par $\{E + p\}$ le plus petit idéal différentiel radiciel contenant E et p . On a :*

$$\{E + p\} \cap \{E + q\} = \{E + p \cdot q\}.$$

Preuve Voir [Ri] I, §11. Le deuxième terme de l'égalité est manifestement inclus dans le premier. Supposons l'existence d'un polynôme r dans les idéaux $\{E + p\}$ et $\{E + q\}$; montrons qu'il appartient à $\{E + p \cdot q\}$. D'après le lemme 8, r admet les deux écritures :

$$\begin{aligned} r^\alpha &= A_0 \cdot \theta_0 p + A_1 \cdot \theta_1 p + \cdots + A_s \cdot \theta_s p + C && (C \in [E]) \\ r^\beta &= B_0 \cdot \phi_0 q + B_1 \cdot \phi_1 q + \cdots + B_t \cdot \phi_t q + D && (D \in [E]) \end{aligned}$$

où s et t sont deux entiers, les A_i et les B_j sont des polynômes de $K\{X\}$, les θ_i et les ϕ_j sont des opérateurs de dérivation. Nous avons donc :

$$r^{\alpha+\beta} = \sum_{i=0}^s \sum_{j=0}^t A_i \cdot B_j \cdot (\theta_i p) \cdot (\phi_j q) + F \quad (F \in [E]).$$

D'après le lemme 9, chaque produit $(\theta_i p)(\phi_j q)$ appartient à $\{p \cdot q\}$. Ainsi $r^{\alpha+\beta}$ appartient à $\{E + p \cdot q\}$ de même que r puisque l'idéal est radiciel. \square

Théorème 4 *Tout idéal différentiel radiciel est une intersection d'idéaux différentiels premiers.*

Preuve Voir [Se2], page 178. Soit I un idéal différentiel radiciel de $K\{X\}$. La preuve consiste à montrer que, quel que soit l'élément q de $K\{X\}$, n'appartenant pas à I , il existe un idéal différentiel premier P qui contienne I mais pas q .

Appelons S l'ensemble des idéaux différentiels radiciels qui contiennent I mais pas q . L'ensemble S est non vide puisqu'il contient au moins I . D'après le lemme de ZORN, il contient un élément maximal P . Supposons que P ne soit pas premier. Il existe alors deux éléments u et v dans $K\{X\}$, tels que ni u , ni v n'appartienne à P , mais dont le produit soit dans P .

Les deux idéaux différentiels radiciels $\{P + u\}$ et $\{P + v\}$ contiennent P strictement. Comme P est maximal dans S , chacun de ces deux idéaux contient q .

Nous avons supposé que P contenait $u \cdot v$. D'après le lemme 10, $\{P + u\} \cap \{P + v\}$ est égal à P , donc P contient q . Cette contradiction prouve que P est premier. \square

2.2 Extensions différentielles

Soient A et B deux anneaux. On appelle *morphisme d'anneaux*, toute application $\Phi : A \rightarrow B$ qui vérifie, pour tous éléments a et b de A :

$$\Phi(a + b) = \Phi(a) + \Phi(b), \quad \text{et} \quad \Phi(a \cdot b) = \Phi(a) \cdot \Phi(b).$$

Supposons que A et B soient des anneaux différentiels, munis d'un même¹ ensemble de dérivations. Φ est un *morphisme d'anneaux différentiels* si Φ vérifie de plus, pour toute dérivation δ :

$$\Phi(\delta a) = \delta(\Phi a).$$

Lemme 11 *Soient A et B deux anneaux différentiels. Le noyau $\ker \Phi$ d'un morphisme d'anneaux différentiels $\Phi : A \rightarrow B$ est un idéal différentiel. Si de plus, B est intègre, $\ker \Phi$ est un idéal différentiel premier.*

1. il s'agit bien sûr d'un abus de langage. Nous supposons plus précisément qu'il existe une bijection entre l'ensemble des dérivations de A et celui de B . Nous notons d'un même symbole les dérivations de A et celles qui leur sont associées dans B .

Preuve Il découle immédiatement de la définition des morphismes d'anneaux que $\ker \Phi$ est un idéal. Soient a dans $\ker \Phi$ et δ une dérivation quelconque. Nous avons $\Phi(\delta a) = \delta(\Phi a) = \delta 0$. Nous avons montré au début du chapitre 1 que $\delta 0 = 0$; $\ker \Phi$ est donc un idéal différentiel. Enfin, supposons qu'un produit $a \cdot b$ soit dans $\ker \Phi$. Nous avons alors $\Phi(a \cdot b) = (\Phi a) \cdot (\Phi b) = 0$. Si B est intègre, $\Phi(a)$ ou $\Phi(b)$ est nul, a ou b appartient à $\ker \Phi$ et $\ker \Phi$ est premier. \square

Soient K un corps différentiel de caractéristique nulle, X un alphabet et P un idéal différentiel de $K\{X\}$. L'ensemble quotient $K\{X\}/P$ est muni canoniquement d'une structure d'anneau différentiel (voir [Ri], II, §6). Si P est premier, l'anneau obtenu est intègre et son corps des fractions L peut être muni d'une structure de corps différentiel. L constitue alors une extension de corps différentielle de K .

2.3 Modèles différentiels et modèles algébriques

2.3.1 Modèles différentiels

Soient L un sur-corps différentiel de K et X un alphabet. Toute application

$$\begin{aligned} \Phi : \quad X &\rightarrow L \\ x_i &\mapsto \alpha_i \end{aligned}$$

se prolonge de manière unique en un morphisme d'anneaux différentiels de $K\{X\}$ dans L , qui injecte K dans L . Par abus de langage, nous nommons également Φ le morphisme :

$$\begin{aligned} \Phi : \quad K\{X\} &\rightarrow L \\ p &\mapsto p(\alpha_i). \end{aligned}$$

Définition 11 *On appelle modèle différentiel d'un système d'équations et d'inéquations de $K\{X\}$ (nous notons les inéquations sous la forme d'un unique polynôme, égal à leur produit) :*

$$\Sigma \quad \left\{ \begin{array}{l} p_1 = 0 \\ \vdots \\ p_r = 0 \\ q \neq 0 \end{array} \right.$$

toute application Φ de X dans un sur-corps différentiel L de K qui annule les équations, mais pas l'inéquation :

$$\begin{aligned} \Phi : \quad X &\rightarrow L \\ x_i &\mapsto \alpha_i. \end{aligned}$$

Le théorème suivant est un Nullstellensatz différentiel, une version différentielle (faible) du théorème des zéros de HILBERT.

Théorème 5 *Un système d'équations et d'inéquations de $K\{X\}$*

$$\Sigma \quad \begin{cases} p_1 = 0 \\ \vdots \\ p_r = 0 \\ q \neq 0 \end{cases}$$

admet un modèle différentiel si et seulement si l'inéquation n'appartient pas au radical de l'idéal différentiel $[p_1, \dots, p_r]$ engendré par les équations.

Preuve Supposons que Σ admette un modèle Φ . Le noyau de Φ contient les p_i . D'après les lemmes 8 et 11, $\ker \Phi$ contient le radical de l'idéal différentiel engendré par les p_i . Comme q n'appartient pas au noyau du modèle, q n'appartient pas à cet idéal.

Supposons que q n'appartienne pas au radical de l'idéal différentiel engendré par les p_i . D'après le théorème 4, il existe un idéal différentiel premier P qui contient les p_i , mais pas q . On obtient un modèle de Σ en prenant pour sur-corps différentiel L de K , le corps des fractions de $K\{X\}/P$ et pour Φ , l'application qui aux x_i , associe leur image par le morphisme canonique de $K\{X\}$ dans L . \square

2.3.2 Modèles algébriques

Soient L un sur-corps de K et X un alphabet. Toute application

$$\begin{aligned} \Phi : \quad \Theta X &\rightarrow L \\ \theta x_i &\mapsto \alpha_{i,\theta} \end{aligned}$$

se prolonge de manière unique en un morphisme d'anneaux de $K[\Theta X]$ dans L qui injecte K dans L :

$$\begin{aligned} \Phi : \quad K[\Theta X] &\rightarrow L \\ p &\mapsto p(\alpha_{i,\theta}). \end{aligned}$$

Définition 12 *On appelle modèle algébrique d'un système d'équations et d'inéquations de $K\{X\}$*

$$\Sigma \quad \begin{cases} p_1 = 0 \\ \vdots \\ p_r = 0 \\ q \neq 0 \end{cases}$$

toute application Φ de ΘX dans un sur-corps L de K qui annule les équations, mais pas l'inéquation :

$$\begin{aligned} \Phi : \quad \Theta X &\rightarrow L \\ \theta x_i &\mapsto \alpha_{i,\theta}. \end{aligned}$$

Le théorème suivant est la version non différentielle du théorème 5. Il s'agit du Nullstellensatz algébrique classique, dont la preuve est contenue dans celle du théorème 5.

Théorème 6 *Un système d'équations et d'inéquations de $K\{X\}$*

$$\Sigma \quad \begin{cases} p_1 = 0 \\ \vdots \\ p_r = 0 \\ q \neq 0 \end{cases}$$

admet un modèle algébrique si et seulement si l'inéquation n'appartient pas au radical de l'idéal (p_1, \dots, p_r) engendré par les équations.

2.4 Relations entre les modèles

D'après les théorèmes 5 et 6, un système d'équations n'admet pas de modèles si une combinaison linéaire des équations est égale à l'unité. Tout algorithme de recherche de modèles d'un système Σ va donc chercher à enrichir Σ par de nouvelles équations, appartenant au radical de l'idéal engendré par les équations précédemment calculées. L'algorithme de réduction est un outil efficace pour ce genre de travail. En algèbre différentielle partielle du moins, il n'est pas le seul. Considérons le système d'équations de $K\{X\}$, en algèbre différentielle partielle ($m = 2$):

$$\Sigma \quad \begin{cases} A_1 & \left\{ \begin{array}{l} x_{(1,2)} \\ x_{(2,1)} + y_{(0,0)} \\ y_{(0,1)} + 1 \end{array} \right. & \begin{array}{l} = 0 \\ = 0 \\ = 0. \end{array} \end{cases}$$

Supposons ΘX ordonné par un ordre d'élimination ($\theta x > \phi y$, pour tous opérateurs θ et ϕ). L'ensemble $A = A_1, A_2, A_3$ est alors auto-réduit, mais Σ n'admet pas de modèles. En effet, le polynôme $\Delta_{12} = \delta_2 A_2 - \delta_1 A_1 = y_{(0,1)}$, appartient au radical de l'idéal différentiel engendré par les A_i . On peut le rajouter aux équations sans changer les modèles différentiels de Σ , ainsi que $\Delta_{12} \text{ rem } A = 1$.

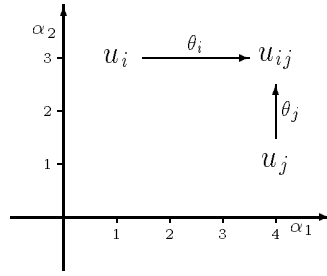
Le procédé illustré par l'exemple ci-dessus, calcule ce que nous appellerons un Δ -polynôme à partir des équations. Nous dirons qu'un système est *cohérent* si aucun Δ -polynôme ainsi calculé "n'apporte quelque chose" aux équations. Ces notions sont précisées dans la section suivante.

2.4.1 Ensembles auto-réduits et cohérents

Soit $A = A_1, \dots, A_r$ un sous-ensemble auto-réduit de $K\{X\}$. Nous notons respectivement u_i et S_i les indéterminées principales et les séparants des A_i . Les expressions H_A et H_A^∞ désignent respectivement le produit de tous les initiaux et séparants des A_i et l'ensemble de tous les produits de puissances de ces initiaux et séparants.

Pour chaque couple (i, j) , notons $u_{ij} = \theta_i u_i = \theta_j u_j$ la plus petite indéterminée qui soit à la fois une dérivée de u_i et de u_j , quand elle existe (c'est-à-dire quand u_i et u_j sont des dérivées d'une même lettre x de X). Par exemple pour $m = 2$, prenons

$u_i = x_{(1,3)}$ et $u_j = x_{(4,1)}$, on obtient $\theta_i = \delta_1^3$, $\theta_j = \delta_2^2$ et $u_{ij} = x_{(4,3)}$:



Comme A est auto-réduit, θ_i et θ_j sont des opérateurs propres. En vertu du lemme 5 (page 14), le polynôme $\Delta_{ij} = (S_j \cdot \theta_i A_i - S_i \cdot \theta_j A_j)$ est d'indéterminée principale strictement inférieure à u_{ij} . Nous appellerons Δ -polynôme de A tout polynôme Δ_{ij} ainsi calculé à partir des A_i .

Définition 13 Soit $A = A_1, \dots, A_r$ un sous-ensemble auto-réduit de $K\{X\}$. Nous dirons que A est cohérent si pour tout Δ -polynôme Δ_{ij} de A , il existe un entier α tel que $H_A^\alpha \cdot \Delta_{ij}$ soit une combinaison linéaire des A_ℓ et de leurs dérivées d'indéterminées principales strictement inférieures à u_{ij} . En d'autres termes, si

$$H_A^\alpha \cdot \Delta_{ij} \in (\theta A_\ell) \quad (\text{où } \theta u_\ell < u_{ij}, \quad 1 \leq \ell \leq r)$$

dit encore autrement si

$$\Delta_{ij} \in (\theta A_\ell) : H_A^\infty \quad (\text{où } \theta u_\ell < u_{ij}, \quad 1 \leq \ell \leq r).$$

A propos des ensembles auto-réduits et cohérents

- Un polynôme Δ_{ij} s'obtient en dérivant des polynômes A_ℓ "jusqu'à u_{ij} ". Dans un système cohérent, Δ_{ij} s'obtient également (à un produit d'initiaux et de séparants d'éléments de A près) en cessant les dérivations *avant* d'atteindre u_{ij} .
- En algèbre différentielle ordinaire, tout ensemble auto-réduit est cohérent, puisque deux indéterminées principales différentes sont nécessairement les dérivées de lettres de X différentes.
- Quand le reste de la division de chaque Δ_{ij} par A est nul, on est certain que A est cohérent. Appelons R_{ij} le polynôme $\Delta_{ij} \text{ rem } A$. Comme le processus de réduction n'emploie que des dérivées des A_ℓ , d'indéterminées principales strictement inférieures à u_{ij} , on a en effet :

$$H_A^\alpha \cdot \Delta_{ij} \equiv R_{ij} \pmod{(\theta A_\ell)} \quad (\theta u_\ell < u_{ij}).$$

- Décider de la non cohérence d'un système n'est par contre *a priori* pas trivial, puisqu'il s'agit de montrer que, quel que soit l'entier α , $H_A^\alpha \cdot \Delta_{ij} \notin (\theta A_\ell)$ ($\theta u_\ell < u_{ij}$). Il est toutefois manifeste que le système donné en exemple dans l'introduction de la section 2.4 n'est pas cohérent.

Le lemme suivant est dû à ROSENFELD ([Ro], lemme, page 397).

Lemme 12 *Soit A un sous-ensemble auto-réduit de $K\{X\}$. Si A est cohérent, alors tout polynôme q de $[A] : H_A^\infty$, partiellement réduit par rapport à A , appartient également à $(A) : H_A^\infty$.*

Preuve Supposons A cohérent et considérons un polynôme q appartenant à $[A] : H_A^\infty$, partiellement réduit vis-à-vis de A . Le polynôme vérifie pour un certain entier α :

$$(1) \quad H_A^\alpha \cdot q = \sum_i \sum_\theta C_{i,\theta} \cdot \theta A_i.$$

La preuve est une récurrence sur la plus grande indéterminée v apparaissant dans (1), qui soit une dérivée propre d'un u_i . Le lemme 3 (page 12) nous autorise à procéder à une telle récurrence.

Cas initial: Aucune dérivée propre, d'aucun u_i n'apparaît dans (1). le terme de droite de l'expression ne contient *a fortiori*, aucune dérivée propre d'aucun A_i , et $H_A^\alpha \cdot q$ appartient à l'idéal (A) . Cas général: supposons l'existence de v . Le polynôme $H_A^\alpha \cdot q$ appartient à l'idéal (θA_ℓ) ($\theta u_\ell \leq v$). La preuve consiste à montrer que pour un certain β , le polynôme $H_A^\beta \cdot q$ appartient à l'idéal (θA_ℓ) ($\theta u_\ell < v$), que nous noterons \mathcal{A}_v par la suite. Le cas général se subdivise en deux sous-cas.

Premier cas $v = \phi u_i$ est la dérivée d'un unique u_i . La preuve est celle du lemme 13. D'après le lemme 5, on a

$$\phi A_i = S_i \cdot v + R_i$$

En appliquant sur les termes de l'expression (1) la substitution :

$$v \rightarrow \frac{\phi A_i - R_i}{S_i}.$$

puis en multipliant les deux termes de l'égalité par une puissance appropriée de S_i , pour effacer les fractions, on obtient une expression :

$$H_A^\beta \cdot q = D \cdot \phi A_i + \sum_j \sum_\theta E_{j,\theta} \cdot \theta A_j \quad (\text{avec } \theta A_j \neq \phi A_i)$$

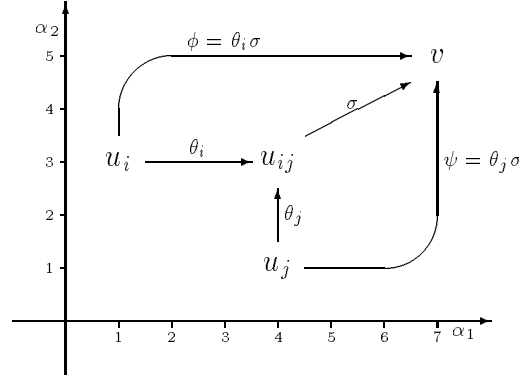
dans laquelle, ni les $E_{j,\theta}$, ni bien sûr $H_A^\beta \cdot q$ ne contiennent v . Le coefficient D est donc nécessairement nul. Comme chaque θA_j (différent de ϕA_i) est inférieur à ϕA_i , le polynôme $H_A^\beta \cdot q$ appartient à \mathcal{A}_v .

Deuxième cas $v = \phi_{i_1} u_{i_1} = \dots = \phi_{i_h} u_{i_h}$ est la dérivée de h indéterminées u_i . En appliquant une substitution similaire à celle du cas précédent et en exprimant v en fonction de $\phi_{i_1} A_{i_1}$ (y compris les indéterminées principales des $\phi_{i_\ell} A_{i_\ell}$, ($2 \leq \ell \leq i_h$)). Nous obtenons comme précédemment une expression :

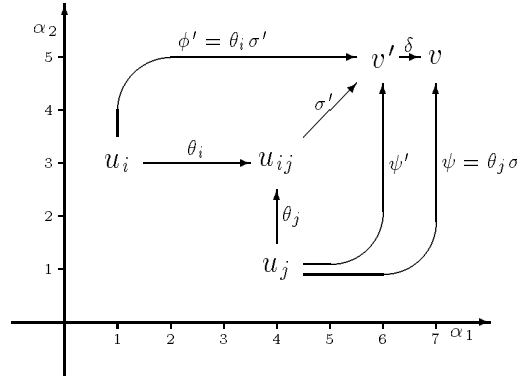
$$\begin{aligned} H_A^\beta \cdot q = D \cdot \phi_{i_1} A_{i_1} + \sum_{\ell=2}^h F_\ell \cdot (S_{i_1} \cdot \phi_{i_\ell} A_{i_\ell} - S_{i_\ell} \cdot \phi_{i_1} A_{i_1}) \\ + \sum_j \sum_\theta E_{j,\theta} \cdot \theta A_j \quad (\text{avec } \theta A_j \neq \phi_{i_1} A_{i_1}) \end{aligned}$$

dans laquelle ni les $E_{j,\theta}$, ni les F_ℓ , ni $H_A^\beta \cdot q$ ne contiennent v . Le coefficient D est nécessairement nul. Chaque θA_j (différent de $\phi_{i_1} A_i$) est inférieure à $\phi_{i_1} A_{i_1}$. Reste à établir que chaque $(S_{i_1} \cdot \phi_{i_\ell} A_{i_\ell} - S_{i_\ell} \cdot \phi_{i_1} A_{i_1})$ appartient à \mathcal{A}_v .

Supposons donc $v = \phi u_i = \psi u_j$. Comme A est auto-réduit, et que v est la dérivée de deux u_ℓ , il existe un opérateur (non nécessairement propre) σ , tel que $\sigma u_{ij} = v$, ce qu'illustre le schéma suivant :



Montrons par récurrence sur σ que $T_\sigma = (S_j \cdot \phi u_i - S_i \cdot \psi u_j)$ appartient à $\mathcal{A}_v : H_A^\infty$. Cas initial: σ est l'opérateur identité (en d'autres termes, $v = u_{ij}$) et on a $T_\sigma = \Delta_{ij}$. Par hypothèse, A est cohérent, ce qui signifie que T_σ appartient à $(\theta A_\ell) : H_A^\infty$ ($\theta u_\ell < u_{ij}$), c'est-à-dire à $\mathcal{A}_v : H_A^\infty$. Cas général: σ se décompose en $\sigma = \delta \sigma'$ et pareillement, $\phi = \delta \phi'$ et $\psi = \delta \psi'$. Nous notons v' l'indéterminée $\sigma' u_{ij}$, ce qu'illustre le schéma suivant :



Soit $T_{\sigma'} = (S_j \cdot \phi' A_i - S_i \cdot \psi' A_j)$. L'hypothèse de récurrence est que $H_A^\beta \cdot T_{\sigma'}$ appartient à l'idéal (θA_ℓ) ($\theta u_\ell < v'$). D'après une propriété des ordres admissibles (à savoir: $u < v \Rightarrow \delta u < \delta v$), la dérivée du polynôme ci-dessus $\delta(H_A^\beta \cdot T_{\sigma'}) = \delta(H_A^\beta) \cdot T_{\sigma'} + H_A^\beta \cdot \delta T_{\sigma'}$ appartient à l'idéal (θA_ℓ) ($\theta u_\ell < \delta v' = v$), c'est-à-dire à \mathcal{A}_v . Le premier terme de la somme est dans \mathcal{A}_v , donc le deuxième y est aussi. Les polynômes $\phi' A_i$ et $\psi' A_j$ sont dans \mathcal{A}_v . Comme T_σ est égal à $\delta T_{\sigma'} - (\delta S_j) \cdot \phi' A_i - (\delta S_i) \cdot \psi' A_j$, on conclut que $H_A^\beta \cdot T_\sigma$ est dans \mathcal{A}_v . \square

A propos du lemme 12

- Le lemme est une généralisation du lemme 13 (page 38) (utilisé pour prouver la correction de l'algorithme d'élimination en algèbre différentielle ordinaire), qui ne s'applique qu'à des ensembles A formés d'un unique polynôme, et donc nécessairement auto-réduits et cohérents.
- SEIDENBERG en connaissait une version plus faible, plus difficile à lire (voir [Se1], III, th. 6), utilisée pour prouver la correction d'un algorithme d'élimination en algèbre différentielle partielle. Ces travaux de SEIDENBERG sont cités par ROSENFELD dans [Ro].
- Le lemme que nous avons donné ne contient qu'une implication : si A est cohérent, alors les idéaux $[A]$ et (A) vérifient une certaine propriété. ROSENFELD donne également l'implication inverse, plus proche de la définition implicitement adoptée par SEIDENBERG. KOLCHIN, qui a généralisé le lemme de ROSENFELD (voir [Ko], III, lemme 5, page 137), procède au même changement que nous. Nous n'avons pas besoin d'une version aussi élaborée que celle de KOLCHIN.

2.4.2 Systèmes réguliers

Soit A un ensemble auto-réduit. RITT qualifie de *réguliers* les zéros de A qui n'annulent pas H_A (voir [Ri], II, §5). C'est cette dénomination qui nous a suggéré la définition suivante :

Définition 14 Nous dirons d'un système d'équations et d'inéquations Σ de $K\{X\}$, qu'il est régulier s'il est de la forme suivante :

$$\Sigma \left\{ \begin{array}{l} A_1 = 0 \\ \vdots \\ A_t = 0 \\ H_A \neq 0 \\ q \neq 0 \end{array} \right\} \begin{array}{l} \text{sous-ensemble auto-réduit et cohérent de } K\{X\} \\ \text{le produit des initiaux et séparants des } A_i \\ q \in K\{X\} \text{ est partiellement réduit vis-à-vis des } A_i \end{array}$$

Théorème 7 Un système d'équations et d'inéquations régulier Σ de $K\{X\}$ admet un modèle différentiel si et seulement s'il admet un modèle algébrique.

Preuve Tout modèle différentiel fournit un modèle algébrique. Supposons que Σ n'admette pas de modèle différentiel et montrons qu'il n'admet pas de modèle algébrique. D'après le théorème 5 (page 29), une puissance α du polynôme $(H_A \cdot q)$ appartient à l'idéal différentiel engendré par les équations :

$$(H_A \cdot q)^\alpha \in [A_1, \dots, A_t].$$

Le polynôme $(H_A \cdot q)^\alpha$ est partiellement réduit vis-à-vis des A_i . D'après le lemme 12, il existe un entier β tel que :

$$H_A^\beta \cdot (H_A \cdot q)^\alpha \in (A_1, \dots, A_t).$$

Une puissance de $(H_A \cdot q)$ appartient à l'idéal algébrique engendré par les A_i . D'après le théorème 6, le système Σ n'admet pas de modèle algébrique. \square

Chapitre 3

Les algorithmes d'élimination de Seidenberg

Les algorithmes d'élimination permettent de décider si un système d'équations et d'inéquations polynomiales différentielles admet ou non un modèle différentiel. SEIDENBERG en déduit d'ailleurs une preuve constructive du Nullstellensatz (notre théorème 5). Dans ce chapitre, nous séparons l'étude de l'algorithme en algèbre différentielle ordinaire, de celle de l'algorithme en algèbre différentielle partielle. Dans les deux cas, nous donnons une version simplifiée de la phase de génération des "systèmes terminaux". Grâce au lemme de ROSENFELD, nous étendons un peu le domaine d'application de l'algorithme aux dérivées partielles (SEIDENBERG ne considérait que des ordres admissibles bornés).

Soit Σ un système d'équations et d'inéquations de $K\{X\}$. Éliminer une lettre x de X c'est trouver une condition nécessaire et suffisante vérifiée par les indéterminées de l'alphabet $X \setminus \{x\}$, que nous noterons Y dans la suite de ce chapitre, pour que Σ admette un modèle différentiel.

Plus concrètement, éliminer une lettre x de X c'est produire une famille finie (Λ_j) de systèmes d'équations et d'inéquations de $K\{Y\}$ qui vérifie: Σ admet un modèle différentiel $\Phi_1 : X \rightarrow L_1$ si et seulement si l'un au moins des Λ_j admet un modèle différentiel $\Phi_2 : Y \rightarrow L_2$. Les modèles Φ_1 et Φ_2 sont différents, ne serait-ce que parce que X et Y sont différents. En pratique L_1 est un sur-corps différentiel de L_2 .

Soit Σ un système quelconque de $K\{X\}$. Il n'est en général pas possible d'éliminer x , c'est-à-dire de construire (Λ_j) en une réécriture. Nous appelons *système terminal* tout système de $K\{X\}$ duquel x peut être éliminée en une réécriture. Les algorithmes de SEIDENBERG (voir [Se1]) fonctionnent en deux temps. Ils commencent par transformer Σ en une famille finie (Ω_i) de systèmes terminaux ayant mêmes modèles différentiels que Σ : plus précisément, Σ admet un modèle différentiel Φ_1 si et seulement si Φ_1 est modèle différentiel de l'un au moins des Ω_i . Les algorithmes éliminent ensuite x des systèmes de cette famille.

A notre connaissance, l'algorithme d'élimination en algèbre différentielle partielle n'a jamais été étudié depuis les travaux de SEIDENBERG. Avant nous par contre,

DIOP [Di1] avait donné une version simplifiée de l'algorithme d'élimination dans le cas ordinaire. Ces travaux ont constitué le point de départ de notre étude. Plus récemment, WANG [Wa2] a décrit des variantes de l'algorithme de 1956 qui incluent des factorisations des équations dans le corps de base, ou qui triangularisent le système pour éviter de procéder aux éliminations, souvent coûteuses. Ces variantes, qui peuvent s'avérer efficaces en pratique, ne diffèrent pas dans leur principe de la méthode originale de SEIDENBERG.

3.1 L'algorithme en algèbre différentielle ordinaire

En algèbre différentielle ordinaire, $K\{X\}$ n'est muni que d'une seule dérivation : δ . Nous supposons ΘX ordonné suivant un ordre d'élimination qui privilégie x , l'indéterminée à éliminer (l'alphabet Y est ordonné de façon quelconque) :

$$\dots > \ddot{x} > \dot{x} > x > \Theta Y$$

Par l'expression *polynôme en x* , nous entendons un polynôme de $K\{X\}$ dans l'écriture duquel x apparaît avec un coefficient non nul, par *l'ordre de p* , noté $\text{ord } p$, nous entendons l'ordre de p en la lettre x . Si p appartient à $K\{Y\}$, nous dirons qu'il est d'ordre -1 . L'ordre de p désigne sans ambiguïté l'indéterminée principale du polynôme. Soit p d'ordre $m \geq 0$. Le degré de p , noté $\text{deg } p$, est le plus grand entier d tel que $x^{(m)^d}$ apparaisse dans l'écriture de p .

3.1.1 Préliminaires

Ces préliminaires ont pour but d'introduire deux propositions, qui seront appliquées en section 3.1.6 pour prouver la correction du mécanisme d'élimination de l'algorithme. Elles sont fondées sur les deux lemmes suivants, dus à RITT (voir [Ri], II, §12). Le lemme 13 constitue une version faible du lemme de ROSENFELD. Ces préliminaires peuvent être passés en première lecture.

Lemme 13 *Soient p un polynôme de $K\{X\}$, d'ordre $m \geq 0$, et q un polynôme partiellement réduit par rapport à p , appartenant à l'idéal différentiel $[p]$ engendré par p . Il existe alors un entier α tel que $S_p^\alpha \cdot q$ soit divisible par p . En d'autres termes*

$$(q \in [p] \text{ et } \text{ord } q \leq \text{ord } p) \quad \Rightarrow \quad \exists \alpha \in \mathbb{N}, \quad S_p^\alpha \cdot q \in (p).$$

Preuve Dire que q appartient à $[p]$, c'est dire qu'il existe un entier s et des polynômes A_i tels que

$$(1) \quad q = A_0 \cdot p + A_1 \cdot p^{(1)} + \dots + A_t \cdot p^{(s)}.$$

Nous supposons q non nul. Soit $x^{(m+t)}$ la dérivée de x d'ordre maximal qui apparaisse dans le terme de droite de l'égalité (1). Si t est nul alors q est égal à $(A_0 \cdot p)$ et le lemme est prouvé. Supposons donc t strictement positif. La preuve consiste à établir

qu'il existe un entier α tel que $S_p^\alpha \cdot q$ admette une écriture qui ne fasse appel qu'à des dérivées de x d'ordre inférieur à $m + t$. D'après le lemme 5 (page 14),

$$p^{(t)} = S_p \cdot x^{(m+t)} + R_t \quad (\text{ord } R_t < m + t).$$

Appliquons dans (1) la substitution

$$x^{(m+t)} \rightarrow \frac{p^{(t)} - R_t}{S_p}.$$

Le polynôme q , qui est d'ordre strictement inférieur à $m + t$, reste inchangé. Éliminons les fractions en multipliant (1) par une puissance appropriée du séparant de p . L'égalité devient :

$$S_p^\alpha \cdot q = B_0 \cdot p + B_1 \cdot p^{(1)} + \cdots + B_t \cdot p^{(t)}.$$

Les coefficients B_i sont tous d'ordre strictement inférieur à $m + t$, ainsi bien sûr que les polynômes $p^{(j)}$, pour $j < u$. Comme $x^{(m+t)}$ n'apparaît pas dans l'écriture de $S_p^\alpha \cdot q$, le coefficient B_t est nécessairement nul. \square

Exemple Soient $p = x\dot{x}$, sa dérivée $\delta p = x\ddot{x} + \dot{x}^2$ et $q = \dot{x}^3$. On a :

$$q \in [p] \quad \text{par} \quad \dot{x}^3 = \dot{x} \cdot \delta p - \ddot{x} \cdot p$$

Ici, $x^{(m+t)}$ est égal à \ddot{x} . Appliquons la substitution

$$\ddot{x} \rightarrow \frac{\delta p - \dot{x}^2}{x}.$$

Notons que δp reste inchangé par la substitution. Multiplions par $S_p = x$:

$$\dot{x}^3 = \dot{x} \cdot \delta p - \frac{\delta p - \dot{x}^2}{x} \cdot p,$$

$$x \cdot \dot{x}^3 = \dot{x}^2 \cdot p + (x \cdot \dot{x} - p) \cdot \delta p.$$

Le terme $B_u = (x \cdot \dot{x} - p)$ est effectivement nul.

Lemme 14 Soit p un polynôme premier de $K\{X\}$, d'ordre $m \geq 0$. Tout polynôme q d'ordre inférieur ou égal à m , appartenant à l'idéal différentiel $[p]$ engendré par p est divisible par p .

Preuve Soit q un polynôme satisfaisant les hypothèses du lemme. D'après le lemme précédent, il existe un entier α tel que :

$$S_p^\alpha \cdot q = A_0 \cdot p.$$

Aucun polynôme ne divise son séparant. Comme p est premier, p divise q . \square

Proposition 1 Soit p un polynôme de $K\{X\}$ d'ordre $m \geq 0$. Tout polynôme appartenant au radical de l'idéal différentiel engendré par p est d'ordre supérieur ou égal à celui de p . En d'autres termes, p est partiellement réduit vis-à-vis de tout polynôme de cet idéal.

Preuve p contient un facteur premier p_0 , de même ordre que lui. L'idéal $\sqrt{[p]}$ est inclus dans $\sqrt{[p_0]}$. D'après les lemmes 8 (page 26) et 14, tout polynôme de $\sqrt{[p_0]}$ est d'ordre supérieur ou égal à celui de p_0 . \square

Proposition 2 *Soit p un polynôme de $K\{X\}$ d'ordre $m \geq 0$. Soit q un polynôme de même ordre que p et tel que le produit $I_p \cdot q$ appartienne au radical de l'idéal différentiel engendré par p . Le polynôme $I_p \cdot q^{\deg p}$ est alors divisible par p . En d'autres termes :*

$$(I_p \cdot q \in \sqrt{[p]} \text{ et } \text{ord} q = \text{ord} p) \Rightarrow I_p \cdot q^{\deg p} \in (p).$$

Preuve Soit q un polynôme vérifiant :

$$I_p \cdot q \in \sqrt{[p]}.$$

Décomposons p en facteurs premiers :

$$p = p_0^{\alpha_0} \cdot p_1^{\alpha_1} \cdots p_t^{\alpha_t}.$$

Parmi ces facteurs, certains, d'indices 0 à s , ont même ordre que p . Les autres, d'indices $s + 1$ à t sont d'ordre inférieur. Appliquons le lemme 14 :

$$I_p \cdot q \in (p_i) \quad (0 \leq i \leq s).$$

L'initial de p est d'ordre strictement inférieur à celui des p_i ($0 \leq i \leq s$), il n'est donc pas divisible par eux. Comme les p_i sont premiers, nous avons :

$$q \in (p_i) \quad (0 \leq i \leq s),$$

$$q^{\alpha_0 + \cdots + \alpha_s} \in (p_0^{\alpha_0} \cdot p_1^{\alpha_1} \cdots p_s^{\alpha_s}).$$

On conclut la démonstration en remarquant que le degré de p est supérieur ou égal à la somme des exposants α d'indices 0 à s et que l'initial de p contient les polynômes $p_{s+1}^{\alpha_{s+1}}$ à $p_t^{\alpha_t}$ en facteurs :

$$q^{\deg p} \in (p_0^{\alpha_0} \cdot p_1^{\alpha_1} \cdots p_s^{\alpha_s}),$$

$$I_p \cdot q^{\deg p} \in (p).$$

\square

Il serait possible de faire de la proposition 1 une conséquence de la proposition 2 mais, comme elles correspondent à deux cas distincts de l'algorithme d'élimination, nous préférons séparer les preuves, afin de mieux mettre en évidence les points délicats de l'algorithme.

Corollaire *Soient p et q deux polynômes de $K\{X\}$, ayant même ordre $m \geq 0$. Si $I_p \cdot q$ appartient au radical de l'idéal différentiel engendré par p , alors le polynôme $q^{\deg p} \text{ rem } p$ est nul.*

$$I_p \cdot q \in \sqrt{[p]} \Rightarrow q^{\deg p} \text{ rem } p = 0.$$

Preuve Le reste $r = q^{\deg p} \text{ rem } p$ vérifie, en accord avec spécifications de l'algorithme de la section 1.3.1

$$I_p^\alpha \cdot q^{\deg p} \equiv r \pmod{(p)}.$$

Comme $q^{\deg p}$ est de degré supérieur ou égal à celui de p , l'exposant α est supérieur ou égal à 1. Supposons que $I_p \cdot q$ appartienne à $\sqrt{(p)}$. D'après la proposition 2, r appartient à l'idéal (p) . Le reste r est de degré inférieur à celui de p , donc r est nul. \square

3.1.2 Définition des systèmes terminaux

Les systèmes terminaux sont des systèmes dans lesquels les équations en x forment un ensemble auto-réduit. Parce que nous sommes en algèbre différentielle ordinaire, cet ensemble est constitué d'un unique polynôme A_1 . L'initial de A_1 doit figurer parmi les inéquations, et les autres inéquations doivent être partiellement réduites par rapport à A_1 . Les systèmes terminaux sont donc de deux types. Le cas général :

$$\Omega \quad \begin{cases} A_1 = 0 & \text{une seule équation en } x \\ I_1 \neq 0 \\ r_i = 0 & \text{les autres équations, appartenant à } K\{Y\} \\ q_j \neq 0 & \text{inéquations partiellement réduites par rapport à } A_1 \end{cases}$$

et sa version dégénérée, qui ne compte pas d'équations en x :

$$\Omega \quad \begin{cases} r_i = 0 \\ q_j \neq 0. \end{cases}$$

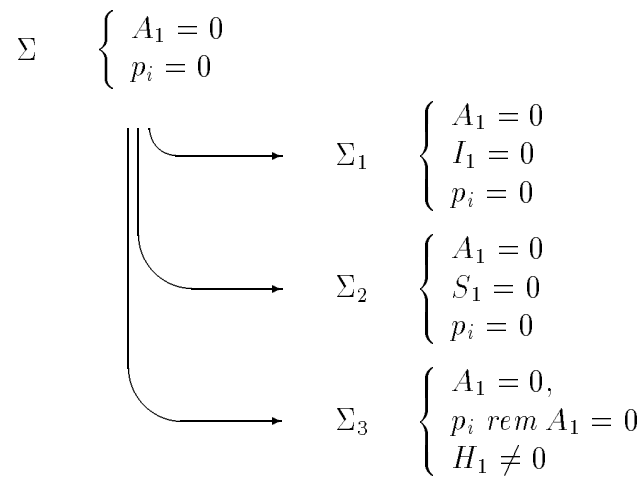
3.1.3 Génération des systèmes terminaux

La méthode que nous donnons ici est plus simple que celle de SEIDENBERG : elle engendre moins de scindages et évite le recours à un sous-algorithme (dit de *préparation*¹). Avant nous, DIOP avait déjà donné une description des réécritures de l'algorithme d'élimination sans le sous-algorithme de préparation (voir [Di1]). La preuve d'arrêt que nous exposons est différente de celle de DIOP.

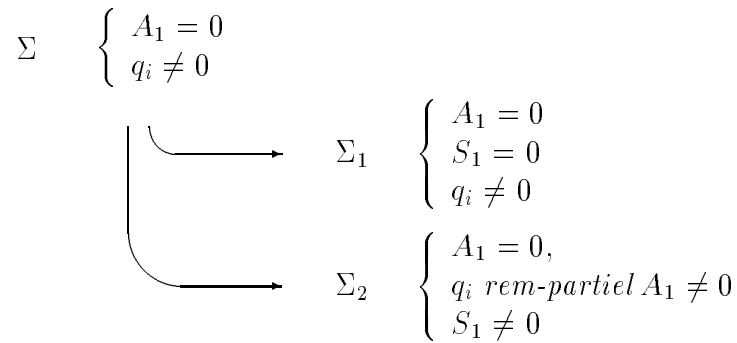
Soit Σ un système quelconque de $K\{X\}$. Nous notons A_1 l'une au choix de ses plus petites équations en x . Cette équation constitue un ensemble caractéristique des équations en x de Σ . Nous donnons l'algorithme sous la forme d'un jeu de règles numérotées, qui portent en commentaire les conditions sous lesquelles elles s'appliquent. A tout système non terminal Σ , s'applique l'une des règles ci-dessous.

règle 11 Elle s'applique si Σ compte des équations en x autres que A_1 (appelons-les p_i). Parce que nous sommes en algèbre différentielle ordinaire, il est possible de réduire ces équations par A_1 . L'algorithme substitue donc $(p_i \text{ rem } A_1)$ à p_i sous la contrainte $H_1 \neq 0$ où H_1 désigne le produit de l'initial I_1 et du séparant S_1 de p_1 .

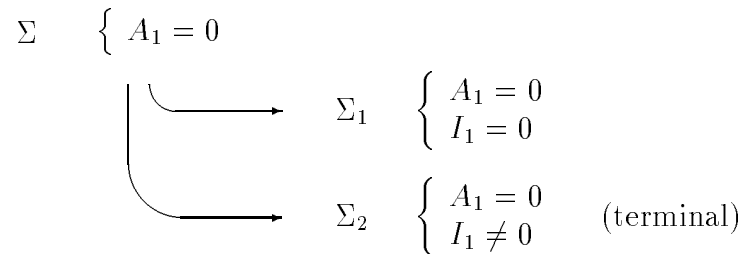
1. sans rapport avec les *préparations* décrites dans [Ko], IV, §13, page 183.



règle 12 Elle s'applique si la seule équation en x de Σ est A_1 et si certaines inéquations q_i ne sont pas partiellement réduites par rapport à A_1 . Dans ce cas, l'algorithme les réduit partiellement, sous la contrainte $S_1 \neq 0$, où S_1 désigne le séparant de A_1 .



règle 13 Elle s'applique si la seule équation en x de Σ est A_1 , si toutes les inéquations q_i sont partiellement réduites par rapport à A_1 mais si l'initial de A_1 ne figure pas parmi les inéquations. Dans ce cas, l'algorithme procède à un scindage sur I_1 .



3.1.4 Elimination dans les systèmes terminaux

Soit Ω un système terminal. Nous notons q le produit de ses inéquations en x , q_j ses autres inéquations et r_i ses équations appartenant à $K\{Y\}$.

Définition 15 On appelle coefficients dans $K\{Y\}$ d'un polynôme p de $K\{X\}$ les polynômes $\text{coeff}_j(p)$ appartenant à $K\{Y\}$, dont l'annulation est nécessaire et suffisante pour spécialiser p en zéro.

Exemple Supposons que A, B, C et D appartiennent à $K\{Y\}$. Ces polynômes sont alors les coefficients dans $K\{Y\}$ du polynôme $p = (A \cdot x + B) \cdot \dot{x} + C \cdot x^3 + D$

règle 21 Le système terminal Ω ne compte pas d'équations en x (il s'agit du cas dégénéré). Notons ℓ le nombre de coefficients dans $K\{Y\}$ de l'inéquation q . L'algorithme réécrit Ω en ℓ systèmes (un par coefficient).

$$\Omega \quad \left\{ \begin{array}{l} r_i = 0 \\ q \neq 0 \\ q_j \neq 0 \end{array} \right. \longrightarrow \Lambda_h \quad \left\{ \begin{array}{l} r_i = 0 \\ \text{coeff}_h(q) \neq 0 \\ q_j \neq 0 \end{array} \right. \quad h \in [1, \ell]$$

règle 22 Le système terminal Ω compte une équation en x , i.e. A_1 , d'ordre strictement supérieur à celui de l'inéquation q . Soit m le nombre de coefficients dans $K\{Y\}$ de I_1 et n le nombre de coefficients dans $K\{Y\}$ de q . L'algorithme réécrit Ω en $m \times n$ systèmes.

$$\Omega \quad \left\{ \begin{array}{l} A_1 = 0 \\ I_1 \neq 0 \\ q \neq 0 \\ r_i = 0 \\ q_j \neq 0 \end{array} \right. \longrightarrow \Lambda_{h,\ell} \quad \left\{ \begin{array}{l} \text{coeff}_h(I_1) \neq 0 \\ \text{coeff}_\ell(q) \neq 0 \\ r_i = 0 \\ q_j \neq 0 \end{array} \right. \quad \begin{array}{l} h \in [1, m] \\ \ell \in [1, n] \end{array}$$

règle 23 Ω compte une équation en x , i.e. A_1 , de même ordre que l'inéquation q . Soit m le nombre de coefficients dans $K\{Y\}$ de I_1 et n le nombre de coefficients dans $K\{Y\}$ du polynôme $(q^{\deg A_1} \text{ rem } A_1)$. L'algorithme réécrit Ω en $m \times n$ systèmes.

$$\Omega \quad \left\{ \begin{array}{l} A_1 = 0 \\ I_1 \neq 0 \\ q \neq 0 \\ r_i = 0 \\ q_j \neq 0 \end{array} \right. \longrightarrow \Lambda_{h,\ell} \quad \left\{ \begin{array}{l} \text{coeff}_h(I_1) \neq 0 \\ \text{coeff}_\ell(q^{\deg A_1} \text{ rem } A_1) \neq 0 \\ r_i = 0 \\ q_j \neq 0 \end{array} \right. \quad \begin{array}{l} h \in [1, m] \\ \ell \in [1, n] \end{array}$$

3.1.5 Preuve d'arrêt

Dans les règles de la section 3.1.3, chaque fois que l'on pose l'initial ou le séparant d'un polynôme p égal à zéro, on peut transformer p en un polynôme plus petit, sans changer les modèles différentiels du système : soient m l'ordre de p et d son degré. Les systèmes

$$\Sigma \quad \left\{ \begin{array}{l} p = 0 \\ I_p = 0 \end{array} \right. \quad \text{et} \quad \Sigma' \quad \left\{ \begin{array}{l} p - I_p \cdot x^{(m)d} = 0 \\ I_p = 0 \end{array} \right.$$

ont mêmes modèles, de même que les systèmes

$$\Sigma \quad \begin{cases} p = 0 \\ S_p = 0 \end{cases} \quad \text{et} \quad \Sigma' \quad \begin{cases} d \cdot p - S_p \cdot x^{(m)} = 0 \\ S_p = 0. \end{cases}$$

Avec de telles modifications, lors de chaque réécriture $\Sigma \rightarrow \Sigma_i$ de la section 3.1.3, l'algorithme réécrit au moins une équation en x de Σ en un nombre fini, éventuellement nul d'équations en x plus petites².

Bien que les polynômes de $K\{X\}$ soient ordonnés suivant un préordre artinien, l'arrêt de l'algorithme n'est pas immédiat, parce que les réécritures ne traitent qu'une lettre à la fois. Certaines réécritures $\Sigma \rightarrow \Sigma'$, peuvent faire augmenter le nombre d'équations en x , et il se peut que la plus petite équation en x de Σ' soit supérieure à celle de Σ , par exemple, après un scindage sur l'initial de cette plus petite équation :

$$\Sigma \quad \begin{cases} p_1 = a \cdot x + b = 0 \\ p_2 = c \cdot \dot{x}^3 + \dots = 0 \end{cases} \quad \text{donne} \quad \Sigma' \quad \begin{cases} a = 0 \\ b = 0 \\ p_2 = c \cdot \dot{x}^3 + \dots = 0. \end{cases}$$

La plus petite équation en x de Σ est p_1 . Celle de Σ' est p_2 ($p_2 > p_1$). Pour prouver l'arrêt, nous allons utiliser le lemme de KÖNIG (voir [Kön], Satz 6.6).

Le lemme de König

Un *graphe orienté* est un couple (S, B) , où S est l'ensemble des *sommets* et $B \subset S \times S$ l'ensemble des *arcs*. Soit a un arc de B , nous appellerons respectivement *origine* et *extrémité* de a la première et la deuxième composante de a . Soient s et t deux sommets de S , s'il existe un arc ayant pour s pour origine et t pour extrémité, on dira que t est un *successeur* de s . Un *chemin* est une suite d'arcs a_1, a_2, \dots telle que l'extrémité de chaque arc coïncide avec l'origine de l'arc qui le suit. On dira enfin qu'un sommet b est *accessible* depuis un sommet a s'il existe un chemin fini $a, a_2, \dots, a_{n-1}, b$.

Un *arbre* est un graphe orienté (S, B, r) où r est un sommet distingué appelé *racine* de l'arbre, vérifiant les axiomes suivants :

- Aucun arc n'arrive en r .
- Il arrive exactement un arc sur chaque sommet différent de r .
- Chaque sommet est accessible depuis la racine.

On appelle *feuille* tout sommet sans successeurs. Un arbre (S, B, r) est dit *infini* si B est infini. Il est dit *localement fini* si chaque sommet de S n'admet qu'un nombre fini de successeurs. Le théorème suivant est dû à KÖNIG.

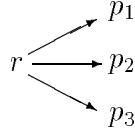
Lemme 15 *Tout arbre infini et localement fini comporte un chemin infini.*

Preuve Soit s un sommet de S et notons $T(s)$ le sous-arbre de (S, B, r) de racine s . Soient t_1, \dots, t_n les successeurs s . Si $T(s)$ est infini, l'un des arbres $T(t_i)$ l'est aussi. Il existe donc une suite infinie (s_n) de sommets avec $s_0 = r$ et telle que s_{i+1} soit successeur de s_i . Cette suite définit un chemin infini dans (S, B, r) . \square

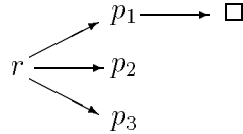
². A une exception près, la réécriture $\Sigma \rightarrow \Sigma_2$ de la règle 12. Cette exception est sans importance puisque le système Σ_2 ainsi produit relève de la règle 13.

Application à l'algorithme

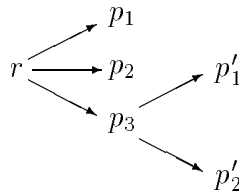
Supposons que la phase de génération de systèmes terminaux ne s'arrête pas, et cherchons la contradiction. En enchaînant les réécritures, l'algorithme produit une suite infinie (Σ_n) de systèmes d'équations et d'inéquations. A cette suite, nous pouvons associer un arbre ayant des équations en x pour sommets. Initialement, l'arbre contient une racine conventionnelle r et autant de feuilles que Σ_0 admet d'équations en x . Par exemple :



La réécriture de Σ_0 en Σ_1 substitue à au moins une équation en x de Σ_0 , un nombre fini éventuellement nul d'équations en x plus petites. Si cette réécriture supprime une équation (mettons p_1) nous convenons de rajouter à l'arbre, un arc ayant p_1 pour origine et un sommet conventionnel \square pour extrémité :



Dans le cas contraire (mettons que l'algorithme réécrive p_3 en p'_1 et en p'_2), nous rajoutons à l'arbre deux arcs, ayant p_3 pour origine et p'_1 et p'_2 pour extrémité :



Clairement, à l'étape i , l'ensemble des équations en x de Σ_i est donné par le feuillage de l'arbre (exceptées bien sûr, les feuilles conventionnelles \square). Par conséquent, notre construction ne modifie que les feuilles et produit bien un arbre localement fini. Si la suite (Σ_n) est infinie, l'arbre l'est aussi. Comme tout sommet est strictement supérieur à ses successeurs, d'après le lemme de KÖNIG, il est possible d'extraire de l'arbre une suite de polynômes infinie et strictement décroissante. Le lemme 4 (page 12) nous assure que cela est impossible. Cette contradiction prouve l'arrêt de l'algorithme.

3.1.6 Preuves de correction

Preuve de correction des scindages

La preuve de la correction des scindages de la section 3.1.3 est quasiment triviale. Soient Σ un système et p un polynôme quelconque de $K\{X\}$. Considérons les systèmes Σ_1 , obtenu en rajoutant l'équation $p = 0$ à Σ et Σ_2 , obtenu en rajoutant à Σ l'inéquation $p \neq 0$

$$\Sigma_1 : \Sigma + p = 0, \quad \text{et} \quad \Sigma_2 : \Sigma + p \neq 0.$$

Il est clair que tout modèle (algébrique ou différentiel) de Σ est, soit un modèle de Σ_1 , soit un modèle de Σ_2 . Inversement, tout modèle (algébrique ou différentiel) de Σ_1 (resp. de Σ_2) est modèle de Σ . Montrons enfin que les deux systèmes suivants ont mêmes modèles différentiels.

$$\Sigma_1 \quad \left\{ \begin{array}{l} p_1 = 0 \\ p_2 = 0 \\ H_1 \neq 0 \end{array} \right. \quad \text{et} \quad \Sigma_2 \quad \left\{ \begin{array}{l} p_1 = 0 \\ r_2 = p_2 \text{ rem } p_1 = 0 \\ H_1 \neq 0. \end{array} \right.$$

Preuve D'après les spécifications de l'algorithme de réduction page 16, nous avons :

$$H_1^\alpha \cdot p_2 \equiv r_2 \pmod{[p_1]}.$$

En d'autres termes, r_2 appartient à l'idéal différentiel $[p_1, p_2]$ et p_2 appartient au radical de l'idéal différentiel $[p_1, r_2]$.

Soit Φ un modèle différentiel de Σ_1 . D'après le théorème 5, Φ annule r_2 et constitue donc un modèle différentiel de Σ_2 . Réciproquement, soit $\Phi : K\{X\} \rightarrow L$ un modèle différentiel de Σ_2 . Comme Φ est un morphisme d'anneaux, $\Phi(H_1 \cdot p_2)$ est égal à $\Phi(H_1) \cdot \Phi(p_2)$. Le modèle n'annule pas H_1 . Comme L est intègre, Φ annule p_2 et constitue un modèle différentiel de Σ_1 . \square

Preuves du mécanisme d'élimination

Nous devons montrer qu'un système terminal Ω admet un modèle différentiel Ψ si et seulement si l'un des systèmes Λ_j obtenus après élimination de la lettre x , admet un modèle différentiel Φ . Notons y_i les éléments de Y ; on a $X = \{y_1, \dots, y_n, x\}$. Soit Ψ un modèle différentiel de Ω

$$\begin{array}{lcl} \Psi : & X & \rightarrow L \\ & y_i & \mapsto \alpha_i \\ & x & \mapsto \beta. \end{array}$$

L'implication de gauche à droite se montre sans difficultés. Pour les trois règles de la section 3.1.4, la restriction Φ de Ψ à l'alphabet Y constitue manifestement un modèle différentiel de l'un au moins des systèmes obtenus après élimination de x

$$\begin{array}{lcl} \Phi : & Y & \rightarrow L \\ & y_i & \mapsto \alpha_i. \end{array}$$

C'est l'implication de droite à gauche qui pose problème. Nous la prouvons séparément pour les trois règles de la section 3.1.4. Soit Φ , donné ci-dessus, un modèle différentiel d'un Λ_j . Trouver Ψ connaissant Φ , c'est trouver un modèle différentiel à $\Phi\Omega$, où $\Phi\Omega$ désigne le système de $L\{x\}$, obtenu en spécialisant Ω au point $y_1 = \alpha_1, \dots, y_n = \alpha_n$.

Preuve de la règle 21. Le système $\Phi\Omega$ ne comporte qu'une inéquation

$$\Phi\Omega \quad \left\{ \begin{array}{l} \Phi q \neq 0. \end{array} \right.$$

Φq est non identiquement nulle puisque l'un au moins des coefficients dans $K\{Y\}$ de q n'appartient pas au noyau de Φ . En d'autres termes, Φq n'appartient pas à l'idéal (0). D'après le théorème 5, $\Phi\Omega$ admet un modèle différentiel. \square

Remarque en algèbre différentielle, à la différence de l'algèbre commutative, il existe des corps de caractéristique nulle dans lesquels certaines inéquations n'admettent pas de solutions. L'inéquation suivante par exemple, n'a aucune solution dans \mathbb{Q} :

$$\dot{x} \neq 0.$$

RITT a montré ([Ri] §22 page 32) que toute inéquation admettait une solution dans un corps différentiel L si L contenait au moins un élément non constant (de dérivée non nulle).

Preuve de la règle 22. A_1 est d'ordre supérieur à q . Le système $\Phi\Omega$ comporte une équation et deux inéquations

$$\Phi\Omega \quad \left\{ \begin{array}{l} \Phi A_1 = 0 \\ \Phi I_1 \neq 0 \\ \Phi q \neq 0. \end{array} \right.$$

Comme ΦI_1 est non identiquement nul, A_1 et ΦA_1 ont même indéterminée principale et sont de même degré en cette indéterminée. Le polynôme $\Phi I_1 \cdot \Phi q$ est donc d'ordre strictement inférieur à ΦA_1 . D'après la proposition 1 (page 39), $\Phi I_1 \cdot \Phi q$ ne peut pas appartenir au radical de l'idéal différentiel engendré par ΦA_1 . D'après le théorème 5, $\Phi\Omega$ admet un modèle différentiel. \square

Preuve de la règle 23. A_1 a même ordre que q . Le système $\Phi\Omega$ comporte une équation et deux inéquations

$$\Phi\Omega \quad \left\{ \begin{array}{l} \Phi p = 0 \\ \Phi I_1 \neq 0 \\ \Phi q \neq 0. \end{array} \right.$$

Comme ΦI_1 est non identiquement nul, A_1 et ΦA_1 ont même indéterminée principale et sont de même degré en cette indéterminée. $\Phi I_1 \cdot \Phi q$ peut être d'ordre strictement inférieur à celui de ΦA_1 . Dans ce cas, l'argumentation donnée pour la règle 22 nous assure de l'existence d'un modèle différentiel. Supposons maintenant que $\Phi I_1 \cdot \Phi q$ ait même ordre que ΦA_1 . L'un des coefficients de ($q^{\deg A_1} \text{ rem } A_1$) n'appartient pas au

noyau de Φ , donc $\Phi(q^{\deg A_1} \text{rem } A_1)$ est non nul et, parce que l'initial de A_1 n'appartient pas non plus au noyau de Φ , les polynômes $\Phi(q^{\deg A_1} \text{rem } A_1)$ et $(\Phi q)^{\deg(\Phi A_1)} \text{rem } (\Phi A_1)$ sont égaux ce qu'illustre le diagramme suivant :

$$\begin{array}{ccc}
 q^{\deg A_1}, A_1 & \xrightarrow{\Phi} & \Phi(q^{\deg A_1}), \Phi A_1 \\
 \text{rem} \downarrow \swarrow & & \downarrow \swarrow \text{rem} \\
 q^{\deg A_1} \text{rem } A_1 & \xrightarrow{\Phi} & \Phi(q^{\deg A_1} \text{rem } A_1)
 \end{array}$$

D'après le corollaire de la proposition 2 (page 40), $\Phi I_1 \cdot \Phi q$ ne peut pas appartenir au radical de l'idéal différentiel engendré par ΦA_1 . D'après le théorème 5, $\Phi \Omega$ admet un modèle différentiel. \square

3.1.7 Optimisations

règle 11 si l'équation A_1 a même ordre que q , vu les spécifications de l'algorithme de réduction, le scindage sur le séparant de A_1 est inutile.

règle 11 il est possible de rajouter à Σ_2 l'inéquation $I_1 \neq 0$. Cela assure que les modèles différentiels étudiés dans les trois branches sont distincts ... du moins jusqu'à la prochaine élimination (la projection de deux ensembles disjoints n'est pas nécessairement disjointe).

règle 13 Le scindage sur I_1 est inutile si A_1 comporte un monôme d'ordre strictement supérieur à celui des inéquations, ayant un coefficient dans $K\{Y\}$ qui ne puisse pas se spécialiser en zéro : un coefficient appartenant à K , ou figurant parmi les inéquations. Voir la preuve de la correction de la règle 22. Par exemple

$$\Sigma \quad \begin{cases} y_1 \cdot \ddot{x} + \dot{x} + y_2 \cdot x + 1 = 0 \\ y_3 \cdot x \neq 0 \end{cases}$$

admet un modèle différentiel si et seulement si $y_3 \neq 0$ (le scindage sur y_1 est inutile). En effet, l'équation ne peut pas se spécialiser en un polynôme d'ordre inférieur à 1 et l'inéquation ne peut pas se spécialiser en un polynôme d'ordre supérieur à zéro.

règle 13 Cas particulier de la remarque précédente : le scindage sur l'initial de A_1 est inutile si le système ne comporte pas d'inéquations en x et si l'équation n'a pas de monôme de degré zéro en x .

règles 21 et 22 Il n'est bien sûr, ni nécessaire, ni souhaitable de multiplier entre elles les inéquations en x pour calculer les coefficients de q .

règle 23 La preuve de la règle 23 montre qu'on ne peut pas se contenter d'un test de division exacte entre $I_1 \cdot q^{\deg A_1}$ et A_1 dans $K\{X\}$, comme pourrait le laisser croire la proposition 2. Par exemple le système

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

n'a pas de solution pour $y = -1$, bien que $(x^2 + y)$ ne soit pas divisible par $(y \cdot x - 1)$. C'est dans $L\{x\}$ qu'il faut procéder au test. Pour les mêmes raisons, une factorisation dans $K\{X\}$ n'évite pas d'élever q à la puissance du degré de A_1 , ainsi que le montre le système

$$\Sigma \quad \begin{cases} x^2 + 2x + y = 0 \\ x + 1 \neq 0 \end{cases}$$

qui n'admet pas de solutions pour $y = 1$.

3.1.8 Exemple d'application

En application d'une remarque de la section 2.1 (page 25), montrons que ni \dot{x} , ni $2\ddot{x} + 1$ n'appartiennent à $[\dot{x}^2 + x]$. Il suffit de montrer que les deux polynômes n'appartiennent pas au radical de cet idéal, c'est-à-dire que les systèmes

$$\Sigma_1 \quad \begin{cases} \dot{x}^2 + x = 0 \\ \dot{x} \neq 0 \end{cases} \quad \text{et} \quad \Sigma_2 \quad \begin{cases} \dot{x}^2 + x = 0 \\ 2\ddot{x} + 1 \neq 0 \end{cases}$$

admettent des modèles différentiels. Nous appliquons l'algorithme d'élimination à la lettre.

Comme l'initial de l'équation est constant dans Σ_1 , tout scindage est inutile. Le système est terminal, il comporte une équation, de même ordre que l'inéquation. D'après la règle 23, Σ_1 admet des solutions si et seulement si l'un au moins des coefficients dans \mathbb{Q} du polynôme $(\dot{x})^2 \text{ rem } (\dot{x}^2 + x) = -x$ est non identiquement nul, ce qui est manifestement le cas. \square

Le système Σ_2 n'est pas terminal. Appliquons la règle 12 et transformons-le en deux systèmes

$$\Sigma_3 \quad \begin{cases} \dot{x} = 0 & \text{le séparant de } \dot{x}^2 + x \\ x = 0 & \dot{x}^2 + x \text{ après simplification} \\ 2\ddot{x} + 1 \neq 0 \end{cases}$$

$$\Sigma_4 \quad \begin{cases} \dot{x}^2 + x = 0 \\ \dot{x} \neq 0 & \text{le séparant de } \dot{x}^2 + x \\ 0 \neq 0 & \text{c'est-à-dire } (2\ddot{x} + 1) \text{ rem } (\dot{x}^2 + x) \end{cases}$$

Σ_4 n'a visiblement pas de solutions. Σ_3 relève de la règle 11. Les scindages sur l'initial

et sur le séparant de la plus petite équation sont inutiles. Nous obtenons le système Σ_5 .

$$\Sigma_5 \quad \begin{cases} 0 & = & 0 \\ x & = & 0 \\ 2\ddot{x} + 1 & \neq & 0 \end{cases} \quad \text{c'est-à-dire } (\dot{x} \text{ rem } x)$$

Appliquons le règle 12 sur Σ_5 . Nous obtenons Σ_6 (le scindage sur le séparant est inutile)

$$\Sigma_6 \quad \begin{cases} x & = & 0 \\ 1 & \neq & 0 \end{cases} \quad \text{c'est-à-dire } (2\ddot{x} + 1) \text{ rem } x$$

Le système terminal Σ_6 relève de la règle 22. Il admet des solutions si et seulement si l'un au moins des coefficients dans \mathbb{Q} de l'inéquation est différent de 0, ce qui est manifestement le cas. \square

3.2 L'algorithme en algèbre différentielle partielle

Nous supposons ΘX ordonné suivant un ordre d'élimination qui privilégie x , l'indéterminée à éliminer (l'alphabet Y est ordonné de façon quelconque) :

$$\Theta\{x\} > \Theta Y$$

Par l'expression *polynôme en x* , nous entendons un polynôme de $K\{X\}$ dans l'écriture duquel x apparaît avec un coefficient non nul.

3.2.1 Définition des systèmes terminaux

Les systèmes terminaux sont des systèmes dans lesquels les équations en x forment un ensemble auto-réduit $A = A_1, \dots, A_t$. Les initiaux et les séparants des éléments de A doivent figurer parmi les inéquations. Celles-ci doivent être partiellement réduites par rapport à A et tous les Δ -polynômes qu'il est possible de calculer à partir éléments de A doivent, après réduction par A , appartenir à $K\{Y\}$ et figurer parmi les équations du système. Les systèmes terminaux sont de deux types. Le cas général :

$$\Omega \quad \begin{cases} A_1 = 0, \dots, A_t = 0 & \text{équations en } x \\ r_i = 0 & \text{équations dans } K\{Y\} \\ H_A \neq 0 & \text{les initiaux et séparants des } A_\ell \\ q_j \neq 0 & \text{partiellement réduits par rapport aux } A_\ell \end{cases}$$

et sa version dégénérée, qui ne compte pas d'équations en x :

$$\Omega \quad \begin{cases} r_i = 0 & \text{équations dans } K\{Y\} \\ q_j \neq 0 & \text{partiellement réduits par rapport aux } A_\ell \end{cases}$$

3.2.2 Génération des systèmes terminaux

Soit Σ un système quelconque de $K\{X\}$. Il est possible d'extraire un ensemble caractéristique $B = B_1, \dots, B_s$ de l'ensemble des équations de Σ , toutes lettres confondues. Nous notons $A = A_1, \dots, A_t$ le sous-ensemble de B des équations en x de B . L'ensemble A n'est pas nécessairement un ensemble caractéristique des équations en x de Σ . Il existe deux façons de calculer de nouvelles équations r à partir des anciennes :

- soit en réduisant par B l'une des équations en x de Σ , n'appartenant pas à A :

$$r = p_i \text{ rem } A$$

- soit en réduisant par B un Δ -polynôme calculé à partir des éléments de A :

$$r = \Delta_{ij} \text{ rem } B \quad (i, j \in [1, t]).$$

Nous désignons par R l'ensemble de toutes les nouvelles équations qu'il est ainsi possible de calculer (B étant fixé). Informellement, l'algorithme consiste à enrichir Σ tant que R contient au moins une équation en x . Lorsque R est inclus dans $K\{Y\}$, l'algorithme procède à un scindage sur les initiaux et les séparants des éléments de A . L'un des systèmes ainsi obtenus est terminal; on applique à nouveau le traitement ci-dessus sur les autres. Nous donnons l'algorithme sous la forme d'un jeu de règles numérotées, qui portent en commentaire les conditions sous lesquelles elles s'appliquent.

règle 11 si R contient au moins une équation en x , l'algorithme enrichit Σ avec R .

règle 12 si R ne contient aucune équation en x ($R \subset K\{Y\}$) L'algorithme enrichit Σ avec R et procède à un scindage sur les initiaux I_ℓ et les séparants S_ℓ des éléments de A et engendre $2t + 1$ systèmes. Dans le système ci-dessous, nous désignons par p_i les équations en x de Σ qui ne sont pas dans A , et par r_j les équations de Σ qui appartiennent à $K\{Y\}$, *y compris* les éléments de B appartenant à $K\{Y\}$, et les nouvelles équations qui sont dans R :

$$\begin{array}{l}
\Sigma \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ p_i = 0 \\ r_j = 0 \\ q_h \neq 0 \end{array} \right. \\
\left. \begin{array}{l} \longrightarrow \Sigma_1 \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ I_1 = 0 \\ p_i = 0 \\ r_j = 0 \\ q_h \neq 0 \end{array} \right. \\ \text{etc ...} \\ \longrightarrow \Sigma_{2t} \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ S_t = 0 \\ p_i = 0 \\ r_j = 0 \\ q_h \neq 0 \end{array} \right. \\ \longrightarrow \Sigma_{2t+1} \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ p_i = 0 \\ r_j = 0 \\ H_A \neq 0 \\ q_h \neq 0 \end{array} \right. \end{array} \right.
\end{array}$$

Pour rendre terminal Σ_{2t+1} , l'algorithme supprime les équations p_i et réduit partiellement par A les inéquations q_j :

$$\Sigma_{2t+1} \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ p_i = 0 \\ r_j = 0 \\ H_A \neq 0 \\ q_h \neq 0 \end{array} \right. \longrightarrow \Omega \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ r_j = 0 \\ H_A \neq 0 \\ \bar{q}_h = q_h \text{ rem-partiel } A \neq 0 \end{array} \right.$$

Remarque comme il a été dit précédemment, il est possible de modifier légèrement la règle 12, afin de disjointre les modèles des Σ_i : soit $\ell \in [1, 2t]$ un indice, il suffit de rajouter à Σ_ℓ , comme inéquations, les initiaux et les séparants dont l'annulation a déjà été considérée, dans les systèmes Σ_1 à $\Sigma_{\ell-1}$. On peut également supprimer le monôme de tête de l'équation $A_\ell = 0$, dans le système Σ_ℓ (où l'on pose $I_\ell = 0$), et réécrire $A_\ell = 0$ en $(\deg_{u_\ell} A_\ell \cdot A_\ell - u_\ell \cdot S_\ell) = 0$ dans $\Sigma_{2\ell}$ (où l'on pose $S_\ell = 0$).

3.2.3 Preuves

Preuve d'arrêt

Les systèmes dont les équations comportent un élément non nul de K , n'admettent bien sûr pas de modèles. Nous supposons donc qu'aucun élément de R , autre que zéro,

n'est dans K . Les polynômes produits par réduction par A , de même que les initiaux et les séparants des éléments de l'ensemble caractéristique sont réduits par rapport à A . D'après le corollaire du théorème 2 (page 21), les deux règles font décroître A strictement. D'après le théorème 3 (page 22), il n'existe pas de suite infinie strictement décroissante d'ensembles auto-réduits. L'algorithme s'arrête donc après un nombre fini de réécritures.

A propos de la réduction par B

- En réduisant les nouvelles équations r par un ensemble caractéristique de l'ensemble de toutes les équations de Σ , nous sommes assurés que les seules équations qui ne font pas décroître B sont celles qui appartiennent à K et qui produisent donc des systèmes sans modèles. De cette façon, nous sommes certains que les réécritures terminent.

Par contre, si nous réduisons les nouvelles équations par un ensemble caractéristique A de l'ensemble des équations en x de Σ , nous produirions des équations qui ne font pas décroître A , parce qu'elles appartiennent à $K\{Y\}$, sans pour autant générer de systèmes sans solutions. De plus, après scindage sur les initiaux et séparants de A , l'ensemble caractéristique pourrait grandir : nous avons donné un exemple d'une telle situation en algèbre différentielle ordinaire, dans la section 3.1.5 (page 43).

- Il n'est pas non plus possible de transposer en algèbre différentielle partielle la méthode que nous avons employée en algèbre différentielle ordinaire : l'algorithme s'arrêterait parce que les nouvelles relations calculées ne s'ajoutaient pas au système, mais étaient substituées à au moins une ancienne équation, plus grande qu'elles. Ici, ce n'est plus le cas à cause des Δ -polynômes, qui ne sont pas inférieurs aux A_ℓ , mais réduits par rapport à A (voir note page 16) et qui s'ajoutent aux équations du système. Il est assez facile d'ailleurs, de conduire dans une boucle sans fin un algorithme inspiré de celui de la section 3.1. Considérons le système suivant :

$$\Sigma_0 \quad \begin{cases} A_1 = x_{(1,0)} = 0 \\ A_2 = x_{(0,1)} + y \cdot x = 0. \end{cases}$$

Σ est auto-réduit, mais on peut calculer un Δ -polynôme entre A_1 et A_2 , le réduire par A , et le rajouter au système :

$$\Sigma_1 \quad \begin{cases} A_0 = y_{(1,0)} \cdot x = 0 \\ A_1 = x_{(1,0)} = 0 \\ A_2 = x_{(0,1)} + y \cdot x = 0. \end{cases}$$

Pour réduire A_1 et A_2 par A_0 , procédons à un scindage sur I_0 . Nous obtenons

deux systèmes :

$$\Sigma_2 \begin{cases} I_0 = y_{(1,0)} = 0 \\ A_1 = x_{(1,0)} = 0 \\ A_2 = x_{(0,1)} + y \cdot x = 0 \end{cases} \quad \text{et} \quad \Sigma_3 \begin{cases} A_0 = y_{(1,0)} \cdot x = 0 \\ I_0 = y_{(1,0)} \neq 0. \end{cases}$$

Comme nous ne considérons que les équations en x , le système Σ_2 se comporte comme Σ et l'algorithme boucle.

Preuves de correction

Nous nous contenterons de prouver que la réécriture de Σ_{2t+1} en Ω préserve les modèles différentiels de Σ_{2t+1} . Notons r_i le reste par A du polynôme p_i . D'après les spécifications de l'algorithme de réduction, nous avons pour toute équation p_i et toute inéquation q_j de Σ_{2t+1} :

$$H_A^\alpha \cdot p_i \equiv r_i \pmod{[A]}$$

$$S_A^\beta \cdot q_h \equiv \bar{q}_h \pmod{[A]}$$

où S_A désigne le produit des séparants des éléments A_ℓ de A , où H_A désigne le produit des initiaux et des séparants des A_ℓ et où α et β sont certains entiers, dépendant de p_i et de q_h .

Tout morphisme d'anneaux différentiels qui annule A annule $(S_A^\beta \cdot q_h - \bar{q}_h)$ (lemme 11 (page 28)) et donc tout morphisme d'anneaux différentiels qui n'annule pas l'un des deux termes de la différence, n'annule pas l'autre non plus. Rappelons qu'un modèle différentiel est un morphisme d'anneaux différentiels à image dans un domaine intègre. Les modèles différentiels de Σ_{2t+1} n'annulent ni q_j , ni S_A (qui apparaît en facteur dans H_A). Ils sont donc des modèles différentiels de Ω . Réciproquement, les modèles différentiels de Ω n'annulent pas q_j . Ils annulent $(H_A^\alpha \cdot p_i)$ (puisqu'ils annulent A et r_i), mais pas H_A . Ils annulent donc p_i et sont des modèles différentiels de Σ_{2t+1} .

3.2.4 Elimination dans les systèmes terminaux — Preuves

Soit Ω un système terminal. Nous notons q le produit des inéquations en x et respectivement r_i et q_j les équations et les inéquations du système, qui appartiennent à $K\{Y\}$.

Le cas dégénéré

Le système Ω est dégénéré s'il ne comporte par d'équations en x . Soit ℓ le nombre de coefficients dans $K\{Y\}$ (définition 15 (page 42)) de l'inéquation q . L'algorithme réécrit Ω en ℓ systèmes (un par coefficient).

$$\Omega \begin{cases} r_i = 0 \\ q \neq 0 \\ q_j \neq 0 \end{cases} \longrightarrow \Lambda_h \begin{cases} r_i = 0 \\ \text{coeff}_h(q) \neq 0 \\ q_j \neq 0. \end{cases} \quad h \in [1, \ell]$$

Preuve Montrons que Ω admet un modèle différentiel Ψ si et seulement si l'un des Λ_h admet un modèle différentiel Φ . L'implication de gauche à droite est claire ; montrons sa réciproque. Soit Φ un modèle différentiel d'un Λ_h .

$$\begin{aligned} \Phi : \quad Y &\rightarrow L \\ y_i &\mapsto \alpha_i. \end{aligned}$$

Trouver Ψ connaissant Φ , c'est résoudre le système $\Phi\Omega$, obtenu en spécialisant Ω au point $y_1 = \alpha_1, \dots, y_n = \alpha_n$. Le système $\Phi\Omega$ se résume à une seule inéquation non identiquement nulle : $\Phi q \neq 0$. Comme Φq n'appartient pas à l'idéal zéro, d'après le théorème 5, $\Phi\Omega$ admet un modèle différentiel. \square

Le cas général

Le système Ω contient un ensemble auto-réduit $A = A_1, \dots, A_t$ d'équations en x . Cet ensemble n'est pas cohérent, mais les Δ -polynômes réduits par A , calculés à partir des éléments A_ℓ de A figurent parmi les équations r_i .

$$\Omega \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ r_i = 0 \\ H_A \neq 0 \\ q \neq 0 \\ q_j \neq 0. \end{array} \right.$$

Grâce au théorème 7 (page 35), nous allons montrer que l'élimination de la lettre x peut s'effectuer par une succession d'éliminations algébriques des indéterminées appartenant à $\Theta\{x\}$, présentes dans Ω .

Soit m le nombre de coefficients dans $K\{Y\}$ du produit H_A des initiaux I_A et des séparants S_A des éléments de A , et soit n le nombre de coefficients dans $K\{Y\}$ de l'inéquation q . Considérons les $m \times n$ systèmes suivants :

$$\Lambda_{h,\ell} \quad \left\{ \begin{array}{l} \text{coeff}_h(H_A) \neq 0 \\ \text{coeff}_\ell(q) \neq 0 \\ r_i = 0 \\ q_j \neq 0. \end{array} \right. \quad \begin{array}{l} h \in [1, m] \\ \ell \in [1, n] \end{array}$$

Nous allons montrer que si un système $\Lambda_{h,\ell}$ admet un modèle différentiel Φ , et si le système $\Phi\Omega$, obtenu en spécialisant Ω au point $y_1 = \alpha_1, \dots, y_n = \alpha_n$ admet un modèle algébrique, alors Ω admet un modèle différentiel. Soit Φ un modèle différentiel de $\Lambda_{h,\ell}$:

$$\begin{aligned} \Phi : \quad Y &\rightarrow L \\ y_i &\mapsto \alpha_i. \end{aligned}$$

Le système spécialisé est de la forme suivante :

$$\Phi\Omega \quad \left\{ \begin{array}{l} \Phi A_1 = 0, \dots, \Phi A_t = 0 \\ \Phi H_A \neq 0 \\ \Phi q \neq 0. \end{array} \right.$$

Comme ΦI_A est différent de zéro, l'ensemble $\Phi A = \Phi A_1, \dots, \Phi A_t$ est auto-réduit et équivalent³ à A .

Montrons que ΦA est cohérent. Comme ΦI_A et ΦS_A sont différents de zéro, les images par Φ des Δ -polynômes calculés entre éléments de A sont égales aux Δ -polynômes calculés entre éléments de ΦA , de même que les restes respectifs de ces polynômes par A et par ΦA , ce qu'illustre le diagramme suivant :

$$\begin{array}{ccc}
 A_i, A_j & \xrightarrow{\Phi} & \Phi A_i, \Phi A_j \\
 \downarrow \swarrow & & \downarrow \swarrow \\
 \Delta_{ij} & \xrightarrow{\Phi} & \Phi \Delta_{ij} \\
 \downarrow \swarrow & & \downarrow \swarrow \\
 \Delta_{ij} \text{ rem } A & \xrightarrow{\Phi} & \Phi \Delta_{ij} \text{ rem } \Phi A
 \end{array}$$

Comme les Δ -polynômes réduits par A figurent parmi les équations de $\Lambda_{h,\ell}$, le système $\Phi \Omega$ est cohérent. D'après le théorème 7, $\Phi \Omega$ admet un modèle différentiel s'il admet un modèle algébrique. \square

A propos de cette preuve

- La preuve que nous donnons est un tout petit peu plus générale que celle de SEIDENBERG, qui ne traitait que des ordres d'élimination bornés. Surtout, notre preuve est plus "lisible" que celle de SEIDENBERG, probablement parce qu'en 1954, l'auteur ne disposait pas encore de la notion de système cohérent introduite par ROSENFELD en 1959, et du lemme qui l'accompagne. Les théorèmes employés par SEIDENBERG sont les théorèmes 6 et 7, page 51 et 52, dans [Se1].

3. pour tout ℓ , les polynômes A_ℓ et ΦA_ℓ ont même indéterminée principale et même degré en cette indéterminée.

Chapitre 4

Bases de Gröbner — Ensembles caractéristiques

Ce chapitre a deux raisons d'être : d'une part, présenter les bases de GRÖBNER que notre algorithme emploie au chapitre suivant et faire d'autre part, un tour d'horizon des tentatives effectuées pour représenter les idéaux différentiels.

Les bases de GRÖBNER, introduites par BUCHBERGER en 1970 (voir [Bu]) pour l'étude des idéaux de polynômes en algèbre commutative, constituent de très bons représentants des idéaux auxquels elles sont associées. En effet, tout idéal d'une algèbre de type fini admet une base de GRÖBNER finie, canonique une fois fixé l'ordre admissible et que l'on sait calculer. Qui plus est, les bases de GRÖBNER constituent des bases, c'est-à-dire des familles génératrices, des idéaux qu'elles représentent. Elles fournissent également des algorithmes d'appartenance aux idéaux, d'égalité et d'inclusion entre idéaux.

En algèbre différentielle, les choses sont moins simples et on peut distinguer deux types de tentatives effectuées pour représenter les idéaux différentiels.

La première est la définition de *bases de GRÖBNER différentielles*, malheureusement infinies lorsqu'elles sont trop proches de leur homologue en algèbre commutative — il s'agit des méthodes de CARRÀ-FERRO (voir [Ca]) et de OLLIVIER (voir [Oll], IV, page 75) — ou bien finies, mais nettement moins satisfaisantes dans le cas de MANSFIELD (voir [M1]).

La seconde consiste à représenter l'idéal différentiel I par un de ses ensembles caractéristiques or, ce type de représentant n'est satisfaisant que lorsque I est premier. OLLIVIER a donné un algorithme qui les calcule sous certaines conditions (voir [Oll], IV, page 89) ; nous étendons ce résultat dans le prochain chapitre.

La première section décrit ce que sont les bases de GRÖBNER en algèbre commutative. La deuxième relate très brièvement les tentatives effectuées par OLLIVIER et MANSFIELD pour les adapter à l'algèbre différentielle. La dernière section est consacrée aux ensembles caractéristiques d'idéaux différentiels.

4.1 Bases de Gröbner algébriques

Alors que l'étude des ensembles caractéristiques conduit à se représenter les polynômes construits sur un alphabet comme des polynômes d'une indéterminée distinguée, à coefficients dans l'anneau construit sur le reste de l'alphabet, le formalisme des bases de GRÖBNER induit naturellement une représentation de ces polynômes en une combinaison linéaire de monômes, à coefficients dans un corps. Ainsi, de nombreuses notions introduites au chapitre I vont admettre un "équivalent" un peu "déformé" dans cette section. Citons en particulier un algorithme de réduction différent et des relations d'ordre admissibles, non plus sur les indéterminées, mais sur les monômes. La preuve que ces relations sont artiniennes est très semblable à celle du lemme 3 (page 12); l'arrêt des algorithmes de cette section en est pareillement la conséquence. Ces similitudes ne doivent pas masquer de profondes différences, comme nous le verrons dans la section 4.3.

4.1.1 Préliminaires

Soit $X = \{x_1, \dots, x_n\}$ un alphabet. Un *monôme* est un élément du monoïde *commutatif* libre \mathcal{M} contenant X . Un monôme peut donc être représenté par un élément de \mathbb{N}^n . Nous notons multiplicativement la loi interne du monoïde. Son élément neutre est noté ϵ . Soient u et w deux monômes de \mathcal{M} . Nous dirons que u est un *facteur* de w s'il existe un monôme v tel que w soit égal au produit uv .

Définition 16 Soit \mathcal{M} l'ensemble des monômes construits sur un alphabet X . On appelle *monoïdéal* tout sous-ensemble de \mathcal{M} , stable par multiplication par des monômes arbitraires.

Qualifions d'*auto-réduit* tout sous-ensemble de \mathcal{M} , dont aucun monôme n'est facteur d'un autre. Nous avons alors :

Lemme 16 (lemme de Dickson) Tout monoïdéal construit sur un alphabet fini X admet une famille génératrice auto-réduite finie et unique.

Preuve Montrons d'abord l'unicité. Soient A et B deux familles génératrices auto-réduites d'un monoïdéal M . Comme A engendre M et que M contient B , tout élément b_i de B admet un élément a_j de A en facteur : $b_i = a_j w_j$. Pour des raisons similaires, a_j admet en facteur un élément b_ℓ de B : $a_j = b_\ell w_\ell$; et par conséquent, b_i est égal à $b_\ell w_j w_\ell$. Comme B est auto-réduite, $w_j w_\ell$ est égal au monôme vide et $b_i = a_j = b_\ell$. Les familles génératrices A et B sont donc égales.

La finitude: montrons que tout ensemble auto-réduit de monômes M construit sur un alphabet fini $X = \{x_1, \dots, x_n\}$ est fini. Soient $m_1 = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ et $m_2 = x_1^{\beta_1} \cdots x_n^{\beta_n}$ deux éléments de M . Comme M est auto-réduit, m_2 ne peut être facteur de m_1 . Les deux monômes vérifient donc :

$$\exists h, \quad 1 \leq h \leq n, \quad \alpha_h < \beta_h.$$

D'après le lemme 2 (page 11), M est fini¹. \square

Définition 17 Soit \mathcal{M} l'ensemble des monômes construits sur l'alphabet X . Une relation d'ordre sur \mathcal{M} est dite admissible si elle est compatible avec la structure de monoïde de \mathcal{M} , c'est-à-dire si elle vérifie pour tous monômes u, v et w de \mathcal{M} :

- $u \leq v \Rightarrow uw \leq vw$,
- $\epsilon \leq u$.

ROBBIANO a donné [Rob] une caractérisation des ordres définis ci-dessus. Parmi ces derniers, nous aurons besoin de l'ordre lexicographique: nous dirons que \mathcal{M} est ordonné suivant l'ordre lexicographique $x_1 < \dots < x_n$ si, quels que soient les monômes $m_1 = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ et $m_2 = x_1^{\beta_1} \dots x_n^{\beta_n}$ on a :

$$m_1 < m_2 \Leftrightarrow \exists i \in [1, n], \quad \forall j \in [i + 1, n], \quad \alpha_i < \beta_i \quad \text{et} \quad \alpha_j = \beta_j.$$

Lemme 17 Soit \mathcal{M} l'ensemble des monômes construits sur un alphabet fini. Toute relation d'ordre admissible dans \mathcal{M} est artinienne.

Preuve Soit (m_n) une suite strictement décroissante dans \mathcal{M} . Montrons que (m_n) est finie. Notons $m_i = x_1^{(\alpha_i)_1} \dots x_n^{(\alpha_i)_n}$ le terme courant de la suite. Les axiomes des ordres admissibles impliquent que si l'indice j est inférieur à i , alors m_j ne peut pas être facteur de m_i . En d'autres termes, la suite vérifie la relation :

$$j < i \Rightarrow \exists h, \quad 1 \leq h \leq n, \quad (\alpha_i)_h < (\alpha_j)_h.$$

D'après le lemme 2 (page 11), M est fini. \square

Soit K un corps de caractéristique nulle et X un alphabet. Tout polynôme de $K[X]$ est une combinaison linéaire de monômes. Supposons \mathcal{M} ordonné suivant un ordre admissible total. Nous appelons *monôme de tête* d'un polynôme p de $K[X]$, n'appartenant pas à K , le plus grand monôme apparaissant dans l'écriture de p , avec un coefficient non nul ; nous le notons : $lm(p)$. Le monôme de tête d'un élément de K , autre que zéro, est ϵ . Zéro n'a pas de monôme de tête. Par extension, si E est un sous-ensemble de $K[X]$, nous notons $lm(E)$ l'ensemble des monômes de tête de ses éléments.

Lemme 18 Soit I un idéal de $K[X]$. Si l'ensemble \mathcal{M} des monômes construits sur X est ordonné suivant un ordre admissible, alors $lm(I)$ est un monoïdéal.

Nous en laissons la preuve au lecteur. Les relations admissibles sur \mathcal{M} induisent un préordre artinien dans $K[X]$:

Définition 18 Soient p et q deux polynômes de $K[X]$. Nous dirons que q est inférieur à p si le monôme de tête de q est inférieur à celui de p .

1. Nous attirons l'attention du lecteur sur le fait que dans ce chapitre, nous appliquons le lemme 2 sur des suites de monômes. Dans le chapitre I, nous l'appliquons sur des suites d'opérateurs de dérivation.

Définition 19 Soient p et q deux éléments de $K[X]$. Nous dirons que q est réduit par rapport à p si le monôme de tête de p n'est facteur d'aucun monôme apparaissant dans l'écriture de q .

Soient p un polynôme et E un sous-ensemble de $K[X]$. Par extension, nous dirons que p est réduit par rapport à E s'il est réduit par rapport à chaque élément de E .

Remarque il ne faut pas confondre les relations “est réduit par rapport à” et “est inférieur à” : soient p et q deux polynômes de $K[X]$. Clairement, si p est inférieur à q alors p est réduit par rapport à q , mais la réciproque n'est pas vraie : la relation “est réduit par rapport à” n'est pas transitive.

Algorithme de réduction Soient p et q deux polynômes de $K[X]$. Il est toujours possible de calculer un polynôme $r = q \text{ rem } p$ vérifiant :

$$\begin{aligned} r &\equiv q \pmod{(p)} \\ r &\text{ réduit par rapport à } p. \end{aligned}$$

Nous notons w le plus grand monôme apparaissant dans l'écriture de q avec un coefficient $k \in K$ et dont $u = \text{lm}(p)$ soit un facteur : $w = uv$. Réduire q par p c'est réécrire q en $(q - k \cdot v \cdot p)$ tant² qu'un tel monôme w existe. w décroît strictement à chaque itération. D'après le lemme 17 l'algorithme s'arrête.

On étend sans difficultés cet algorithme à un ensemble fini de polynômes : soient q un polynôme et E un sous-ensemble de $K[X]$. Il est toujours possible de calculer un polynôme $r = q \text{ rem } E$ vérifiant :

$$\begin{aligned} r &\equiv q \pmod{(E)} \\ r &\text{ réduit par rapport à } E. \end{aligned}$$

4.1.2 Bases de Gröbner

Théorème 8 Soit I un idéal de $K[X]$. Un sous-ensemble B de I est une base de GRÖBNER de I s'il vérifie l'une des deux conditions équivalentes suivantes :

1. $\text{lm}(B)$ engendre $\text{lm}(I \setminus \{0\})$.
2. Pour tout polynôme q de $K[X]$, il existe un unique polynôme r , réduit par rapport à B et équivalent à q modulo I .

Preuve 1) \Rightarrow 2) L'existence d'un algorithme de réduction implique l'existence de r . Montrons l'unicité : soient r_1 et r_2 deux polynômes réduits par rapport à B et équivalents à q modulo I . La différence $(r_1 - r_2)$ est dans I et est réduite par rapport à B : si $\text{lm}(B)$ engendre $\text{lm}(I \setminus \{0\})$, alors $(r_1 - r_2)$ est nul.

2. nous supposons p unitaire.

2) \Rightarrow 1) D'après la condition 2), tout polynôme p de I est réduit à zéro par B . D'après les spécifications de l'algorithme de réduction, il existe un élément dans la base dont le monôme de tête divise le monôme de tête de p . \square

L'idéal (0) admet l'ensemble vide pour (unique) base de GRÖBNER. Nous supposons dans ce qui suit que les idéaux sont différents de (0) .

Définition 20 *Soit B une base de GRÖBNER d'un idéal I de $K[X]$. Nous dirons que B est complètement réduite si chaque élément de B est unitaire et est réduit par rapport aux autres.*

Il est donc clair que la base de GRÖBNER complètement réduite associée à l'idéal unité est le singleton : $\{1\}$.

Théorème 9 *Soit X un alphabet fini. Tout idéal I de $K[X]$ admet une base de GRÖBNER complètement réduite unique³ et finie.*

Preuve Montrons l'unicité. Tout idéal I admet au moins une base de GRÖBNER complètement réduite. Soient B_1 et B_2 deux bases de GRÖBNER complètement réduites de I . Les ensembles $\ell m(B_1)$ et $\ell m(B_2)$ sont auto-réduits et engendrent tous deux $\ell m(I)$. D'après le lemme 16 ces deux ensembles sont égaux. Montrons que deux polynômes $p_1 \in B_1$ et $p_2 \in B_2$ ayant mêmes monômes de tête u , sont égaux. Le polynôme $q = (p_1 - p_2)$ est dans I . Comme les bases sont complètement réduites et comme tous les monômes non vides m apparaissant dans l'écriture de q sont, soit des monômes de p_1 , soit des monômes de p_2 et sont distincts de u , le polynôme q est réduit par rapport à chacune des bases. D'après le théorème 8, q est nul.

La finitude est une conséquence immédiate du lemme 16. \square

Synthèse *Une base de GRÖBNER B d'un idéal I constitue une base de I et fournit un algorithme d'appartenance à I . Les bases de GRÖBNER de deux idéaux I et J permettent donc de tester l'égalité et l'inclusion des idéaux. Lorsqu'elle est complètement réduite, B est un représentant canonique de I , une fois fixé l'ordre admissible.*

BUCHBERGER a donné en 1965 (voir [Bu]) un algorithme qui la calcule.

4.1.3 Théorèmes utiles

A partir de cette section, les bases de GRÖBNER que nous considérerons seront toujours complètement réduites.

Dans le prochain chapitre, nous aurons besoin des théorèmes suivants. Le premier est dû à TRINKS (voir [Tr1]). J'ignore à qui nous sommes redevables du second.

Théorème 10 *Soient I un idéal de $K[x_1, \dots, x_n]$ et B sa base de GRÖBNER pour l'ordre lexicographique $x_1 < \dots < x_n$. Quel que soit l'entier i inférieur ou égal à n , $B \cap$*

3. l'ordre admissible étant fixé.

$K[x_1, \dots, x_i]$ est la base de GRÖBNER complètement réduite pour l'ordre lexicographique $x_1 < \dots < x_i$ de l'idéal $I \cap K[x_1, \dots, x_i]$. En d'autres termes,

$$(B \cap K[x_1, \dots, x_i]) = (B) \cap K[x_1, \dots, x_i].$$

Preuve Tout sous-ensemble d'une base de GRÖBNER complètement réduite est complètement réduit. Supposons B calculée pour l'ordre lexicographique $x_1 < \dots < x_n$. Si le monôme de tête d'un polynôme p de B appartient à $K[x_1, \dots, x_i]$, alors p lui-même appartient à $K[x_1, \dots, x_i]$. Ainsi, tout polynôme q de $K[x_1, \dots, x_i]$ qui appartient aussi à (B) , est réduit à zéro par $B \cap K[x_1, \dots, x_i]$, donc $B \cap K[x_1, \dots, x_i]$ engendre l'idéal $(B) \cap K[x_1, \dots, x_i]$. \square

Théorème 11 Soient $A = A_1, \dots, A_t$ et $H = H_1, \dots, H_s$ deux sous-ensembles de $K[x_1, \dots, x_n]$. Soit G un ensemble de s polynômes construits à partir de H et de s nouvelles indéterminées z_ℓ :

$$G_i = H_i \cdot z_i - 1.$$

Appelons B la base de GRÖBNER de l'idéal $I = (A \cup G)$ engendré par A et G dans $K[x_1, \dots, x_n, z_1, \dots, z_s]$ pour un ordre lexicographique du type $x_i < z_j$ (où $1 \leq i \leq n$ et $1 \leq j \leq s$).

L'intersection de B et de l'anneau $K[x_1, \dots, x_n]$ est une base de GRÖBNER de l'idéal $(A) : H^\infty$. En d'autres termes :

$$(B \cap K[x_1, \dots, x_n]) = (A) : H^\infty.$$

Preuve L'inclusion de droite à gauche (\supset) est simple : soit p un élément de $(A) : H^\infty$. Comme I contient A , il existe un certain $h = H_1^{\alpha_1} \dots H_s^{\alpha_s}$ tel que $(h \cdot p)$ soit dans I . Soit $\ell = z_1^{\alpha_1} \dots z_s^{\alpha_s}$. Le polynôme $(\ell \cdot h \cdot p)$ appartient à I . Comme chaque produit $(z_i \cdot H_i)$ est équivalent à 1 modulo I , le polynôme p est dans I .

Prouvons l'inclusion inverse (\subset) par la technique de RITT (celle du lemme 13 (page 38)). Soit p appartenant à I et à $K[x_1, \dots, x_n]$. Le polynôme p admet une écriture :

$$(1) \quad p = \sum_{i=1}^t C_i \cdot A_i + \sum_{j=1}^s D_j \cdot G_j.$$

Soit z_t l'indéterminée z d'indice maximal qui apparaisse dans le terme de droite de l'égalité (1). Si t est nul, alors p est dans (A) et le théorème est prouvé. La preuve consiste à établir qu'il existe un entier α tel que $(H_t^\alpha \cdot p)$ admette une écriture du type (1) dans laquelle n'apparaissent que des indéterminées z d'indice strictement inférieur à t .

Appliquons dans (1) la substitution

$$z_t \longrightarrow \frac{G_t - 1}{H_t}.$$

Le polynôme p , qui appartient à $K[x_1, \dots, x_n]$, reste inchangé. G_t n'est pas non plus modifié par la substitution. Eliminons les fractions en multipliant (1) par une puissance

appropriée de H_t . L'égalité devient :

$$H_t^\alpha \cdot p = \sum_{i=1}^t E_i \cdot A_i + \sum_{j < u} F_j \cdot G_j + K \cdot G_t.$$

L'indéterminée z_t n'apparaît plus que dans l'écriture de G_t . Le coefficient K est nécessairement nul. \square

4.2 Bases de Gröbner différentielles

Les idéaux différentiels de $K\{X\}$ sont des idéaux de $K[\Theta X]$, mais, comme ΘX est un alphabet infini, il est clair que tous les idéaux non triviaux de $K\{X\}$ admettent des bases de GRÖBNER algébriques infinies.

CARRÀ-FERRO (voir [Ca]) et OLLIVIER (voir [Oll], IV, page 75) ont tenté indépendamment de définir une notion de base de GRÖBNER différentielle en étendant le mécanisme de réduction défini dans ce chapitre, un peu comme l'algorithme de la section 1.3.1 (page 16) étend la division euclidienne classique. Ces tentatives ont abouti à un demi-succès : s'il est possible de définir de telles bases, celles-ci sont généralement infinies, même pour certains idéaux "très simples" tels que $[x^2]$ en algèbre différentielle ordinaire.

Bien que RITT montre (voir [Ri], I, §15) qu'il existe des idéaux différentiels qui n'admettent pas de famille génératrice finie, on peut difficilement invoquer cette raison puisqu'algorithmeiquement, les idéaux manipulés sont presque toujours donnés par une famille génératrice finie. En fait, trouver un algorithme qui décide de l'appartenance à un idéal différentiel de type fini est un problème ouvert (voir [GMO]).

Notons qu'il existe en algèbre différentielle un *théorème de la base finie* (voir [Ri], I, §10) dû à RITT et RAUDENBUSH, mais c'est un *faux-ami* du célèbre théorème de la base de HILBERT, en algèbre commutative :

Définition 21 *Soit I un idéal différentiel de $K\{X\}$. Un sous-ensemble B de I est une base de I (au sens de RITT et RAUDENBUSH) si I est inclus dans le radical de l'idéal différentiel $[B]$ engendré par B .*

Théorème 12 (théorème de la base finie) *Soit X un alphabet fini. Tout idéal différentiel I de $K\{X\}$ admet une base finie (au sens de la définition 21).*

Une conséquence importante de ce théorème est que tout idéal différentiel radiciel est une intersection *finie* d'idéaux différentiels premiers (pour les preuves, voir [Ri] I, §12 et [Se2]).

Pour remédier aux problèmes de finitude rencontrés par OLLIVIER et CARRÀ-FERRO, MANSFIELD a donné (voir [M1]) une autre définition des bases de GRÖBNER différentielles. Sa méthode emploie l'algorithme de réduction du chapitre I mais, si

l'auteur obtient effectivement un algorithme qui s'arrête dans tous les cas, le résultat obtenu n'est pas aussi satisfaisant que celui de ses prédécesseurs.

Nous présentons un très bref aperçu des travaux de OLLIVIER puis de ceux de MANSFIELD.

4.2.1 Méthode de Ollivier

Soit X un alphabet. Si l'ensemble \mathcal{M} des monômes construits sur l'alphabet ΘX est totalement ordonné, on peut définir la *dérivée* θ d'un monôme m , comme le monôme de tête du polynôme θm . Les dérivées de ϵ sont égales à ϵ . De cette façon, il est possible de définir une notion d'idéal différentiel de monômes :

Définition 22 *Soit \mathcal{M} l'ensemble des monômes construits sur l'alphabet ΘX et soit \mathcal{M} ordonné suivant une relation d'ordre totale. On appelle monoïdéal différentiel tout monoïdéal de \mathcal{M} , stable par dérivation.*

On remarquera que cette notion n'est pas intrinsèque, mais dépend de la relation qui ordonne les monômes. La définition ci-dessous étend la définition 17. Nous laissons le lecteur en déduire un algorithme de réduction :

Définition 23 *Soit \mathcal{M} l'ensemble des monômes construits sur l'alphabet ΘX . Une relation d'ordre sur \mathcal{M} est dite admissible si elle vérifie pour tous monômes u, v et w de \mathcal{M} , et pour toute dérivation δ :*

$$- u \leq v \quad \Rightarrow \quad uw \leq vw,$$

$$- \epsilon \leq u,$$

$$- u \leq v \quad \Rightarrow \quad \delta u \leq \delta v,$$

$$- u < \delta u.$$

Théorème 13 *Soit I un idéal différentiel de $K\{X\}$. Un sous-ensemble B de I est une base de GRÖBNER différentielle de I s'il vérifie l'une des deux conditions équivalentes suivantes :*

1. $\ell m(B)$ engendre $\ell m(I \setminus \{0\})$ en tant que monoïdéal différentiel.
2. Pour tout polynôme q de $K\{X\}$, il existe un unique polynôme r , réduit par rapport à B et équivalent à q modulo I .

Voir la preuve dans [O11], théorème 1, page 80. OLLIVIER donne un algorithme (voir [O11]) qui calcule la base de GRÖBNER différentielle complètement réduite d'un idéal différentiel, si celle-ci est finie.

4.2.2 Méthode de Mansfield

MANSFIELD a repris et étendu l'algorithme inventé par CARRÀ–FERRO mais définit les bases de GRÖBNER différentielles de la manière suivante (voir [M1], II, page 32) :

Définition 24 *Soit I un idéal différentiel de $K[X]$. Une base de GRÖBNER B de I est une famille génératrice de I qui réduit à zéro (par l'algorithme de la section 1.3.1 (page 16)) tous les éléments de I .*

L'emploi de cet algorithme de réduction a l'avantage de ne pas engendrer de bases infinies mais présente deux inconvénients :

- La base B calculée par l'algorithme de MANSFIELD est incorrecte s'il existe un produit d'initiaux et de séparants des éléments de B (c'est-à-dire un élément de H_B^∞) qui appartienne à I , ce qui ne peut pas être testé algorithmiquement (voir [M1], II, pages 18 et 49).
- Qu'un polynôme p de $K\{X\}$ soit réduit à zéro par la base B n'implique pas que p appartienne à I .

Par exemple, le singleton $\{\dot{x}^2 + x\}$ constitue la base de GRÖBNER différentielle de l'idéal différentiel $I = [\dot{x}^2 + x]$ (voir [M1], theorem one, page 43). La base réduit $2\ddot{x} + 1$ à zéro, bien que ce polynôme n'appartienne pas à l'idéal I (pour une preuve de ce dernier point voir la section 3.1.8 (page 49)).

4.3 Ensembles caractéristiques d'idéaux

Un ensemble caractéristique A d'un idéal différentiel I réduit⁴ à zéro tous les éléments de I .

Preuve Soient p un polynôme de I et $r = p \text{ rem } A$. Supposons r non nul et montrons la contradiction. D'après les spécifications de l'algorithme de réduction, on a $r \equiv H_A^\alpha \cdot p \pmod{I}$. Comme p est dans l'idéal, r y est aussi et est réduit par rapport à A . D'après le théorème 2 (page 21), cela ne se peut pas. Cette contradiction prouve que le reste est nul. \square

Si l'idéal différentiel est premier, l'ensemble caractéristique A en fournit même un algorithme d'appartenance :

Lemme 19 *Soient A un ensemble caractéristique d'un idéal différentiel premier I de $K\{X\}$ et p un polynôme quelconque de $K\{X\}$. On a :*

$$p \in I \quad \Leftrightarrow \quad p \text{ rem } A = 0.$$

4. dans cette section, l'algorithme de réduction utilisé est celui du chapitre I.

Preuve Voir [Ri], II, §5. Nous avons déjà démontré l'implication de gauche à droite. Montrons l'implication inverse et supposons $p \text{ rem } A$ nul. Les initiaux et séparants des éléments de A sont réduits par rapport à A . D'après le théorème 2 (page 21), ils n'appartiennent pas à I . Comme le produit $(H_A^\alpha \cdot p)$ appartient à I et que cet idéal est premier, p appartient à I . \square

En théorie, le lemme ci-dessus peut également être utilisé pour décider de l'appartenance à un idéal différentiel radiciel. RITT donne un algorithme qui décompose un idéal différentiel radiciel I de $K\{X\}$ en une intersection d'idéaux différentiels premiers (voir théorème 12) en calculant un ensemble caractéristique pour chacun de ces idéaux. Du lemme 19 on déduit alors facilement un test d'appartenance à I . L'algorithme de RITT est malheureusement peu praticable puisqu'il utilise des factorisations de polynômes suivant des tours d'extensions de corps de K . Qui plus est, les propriétés des ensembles caractéristiques ne permettent pas de déterminer la plus petite famille d'idéaux différentiels premiers dont I est l'intersection⁵. C'est ce que confirme le paragraphe en fin de section.

Même si l'on ne s'intéresse qu'à des idéaux différentiels premiers, les ensembles caractéristiques n'ont pas toutes les bonnes propriétés des bases de GRÖBNER algébriques.

OLLIVIER a donné une méthode (voir [O11] IV, page 89) qui les calcule à partir d'une famille génératrice de l'idéal, à condition de savoir à l'avance que I est premier, d'être capable de tester l'appartenance à I (ce point pouvant être résolu grâce aux algorithmes de Nullstellensatz décrits dans cette thèse), et de calculer des inverses dans des extensions algébriques du corps de base K . L'algorithme que nous donnons dans le prochain chapitre est plus général.

Les principales différences entre bases de GRÖBNER algébriques et ensembles caractéristiques sont les suivantes :

Les ensembles caractéristiques ne sont pas des bases

Un exemple suffit à le montrer : l'idéal différentiel $I = [y \cdot x + z, \dot{y}, \dot{x}]$ est premier puisque ses générateurs forment un ensemble *orthonomique* ($H_A = 1$) auto-réduit pour un certain ordre admissible (voir [Ri], VIII, §10), ou encore parce que les règles suivantes forment une base de GRÖBNER différentielle et que les monômes de tête sont de simples lettres (voir [O11]) :

$$\begin{cases} z & \rightarrow & -y \cdot x \\ \dot{y} & \rightarrow & 0 \\ \dot{x} & \rightarrow & 0. \end{cases}$$

Lorsque les indéterminées sont ordonnées suivant un ordre d'élimination : $\theta z < \phi y < \psi x$ pour tous opérateurs de dérivation θ , ϕ et ψ , l'idéal I admet $A = \{\dot{z}, \dot{y}, y \cdot x + z\}$ pour ensemble caractéristique, or \dot{x} n'appartient pas à $[A]$.

5. celle qui correspond aux composantes irréductibles de la variété algébrique différentielle associée à I

Ils fournissent un test d'égalité mais pas d'inclusion entre idéaux

Lemme 20 *Deux idéaux différentiels premiers I et J de $K\{X\}$, d'ensembles caractéristiques respectifs A et B sont égaux si et seulement si tous les éléments de A sont réduits à zéro par B et si tous les éléments de B sont réduits à zéro par A .*

Preuve si tous les éléments de A réduits à zéro par B , comme J est premier, A est dans J (lemme 19). Comme B est un ensemble caractéristique de J , l'ensemble auto-réduit A est supérieur ou équivalent à B . Par une argumentation similaire, on montre que B est dans I et donc que A et B sont équivalents. Les deux idéaux admettent alors indifféremment A ou B pour ensemble caractéristique. Comme I et J sont premiers, d'après le lemme 19, ces deux idéaux sont égaux. \square

Par contre et de façon un peu surprenante, si tous les éléments de A sont réduits à zéro par B , on ne peut pas conclure que I est inclus dans J . Considérons à nouveau l'idéal différentiel premier $I = [y \cdot x + z, \dot{y}, \dot{x}]$ donné en exemple dans le paragraphe ci-dessus. Pour l'ordre d'élimination : $\theta z < \phi y < \psi x$ (pour tous opérateurs de dérivations θ, ϕ et ψ), I admet $A = \{\dot{z}, \dot{y}, y \cdot x + z\}$ pour ensemble caractéristique. L'idéal différentiel $J = [z, y]$ est lui-aussi premier et admet $B = \{z, y\}$ pour ensemble caractéristique or, bien que tous les éléments de A soient réduits à zéro par B , le polynôme \dot{x} n'appartient pas à J .

Cette bizarrerie est due au fait que les ensembles caractéristiques ne sont pas des familles génératrices des idéaux et que la relation de réduction n'est pas transitive. KOLCHIN indique (voir [Ko], IV, §9, problem 3) que ce problème de l'inclusion de deux idéaux différentiels premiers dont on ne connaît que les ensembles caractéristiques est loin d'être résolu.

On dispose tout de même d'un *critère* d'inclusion, dont nous laissons la preuve au lecteur :

Lemme 21 *Soient I et J deux idéaux différentiels premiers de $K\{X\}$, d'ensembles caractéristiques respectifs A et B . Si tous les éléments de A sont réduits à zéro par B et si H_A n'est pas réduit à zéro par B , alors I est inclus dans J .*

Chapitre 5

Représentation des modèles différentiels d'un système

Dans ce chapitre, nous exposons le principal résultat de cette thèse : un algorithme original qui calcule une représentation des modèles différentiels d'un système d'équations et d'inéquations polynomiales différentielles. Nous en déduisons un algorithme qui calcule un ensemble caractéristique d'un idéal différentiel premier donné par une famille génératrice finie.

L'algorithme est une application du lemme de ROSENFELD. A partir d'un système d'équations et d'inéquations polynomiales différentielles Σ de $K\{X\}$, il construit une famille finie (Ω_i) de systèmes réguliers ayant mêmes modèles différentiels que Σ . Plus précisément : Σ admet un modèle différentiel Φ si et seulement si Φ est modèle différentiel de l'un au moins des Ω_i .

Les systèmes réguliers admettent un tel modèle si et seulement si ils admettent un modèle algébrique (théorème 7). Nous représentons les modèles de chaque Ω_i par une base de GRÖBNER algébrique. Ce calcul détecte les systèmes réguliers sans modèles : l'algorithme décide donc du vide.

La représentation que nous calculons est assez lourde : une famille de bases de GRÖBNER mais, l'algorithme fonctionne en algèbre différentielle partielle aussi bien qu'ordinaire, pour des idéaux différentiels non nécessairement premiers, engendrés par des équations non nécessairement explicites. Il n'emploie enfin que l'addition, la multiplication et le test d'égalité à zéro dans le corps de base K .

Supposons que Σ ne comporte pas d'inéquations. Les bases de GRÖBNER calculées à partir de Σ permettent de tester après coup l'appartenance au radical de l'idéal différentiel $[\Sigma]$, même si ce test n'est pas élémentaire pour un idéal radiciel quelconque. Dans le cas d'un idéal que l'on sait premier, ce test se limite à un simple calcul de réduction par une seule base, aisément identifiable. Cette propriété a d'intéressantes applications en automatique, comme nous le verrons au chapitre suivant.

Nous désignons l'algorithme décrit dans les deux sections suivantes sous le nom : ROSENFELD-GRÖBNER.

5.1 Génération des systèmes réguliers

Soit Σ un système quelconque de $K\{X\}$. L'ensemble des équations de Σ est fini, il est donc possible d'en extraire un ensemble caractéristique A . Nous notons $A = A_1, \dots, A_t$ l'ensemble caractéristique de Σ , étant bien entendu que cet ensemble évolue avec Σ . Si le système admet plusieurs ensembles caractéristiques, A désigne l'un d'entre eux au choix.

$$\Sigma \quad \left\{ \begin{array}{ll} A_1 = 0, \dots, A_t = 0 & \text{ensemble caractéristique des équations} \\ p_i = 0 & \text{les autres équations} \\ q_j \neq 0 & \text{les inéquations} \end{array} \right.$$

Il existe deux façons de calculer de nouvelles équations r à partir des anciennes :

- soit en réduisant par A l'un des polynômes p_i :

$$r = p \text{ rem } A,$$

- soit en réduisant par A un Δ -polynôme calculé à partir des éléments de A :

$$r = \Delta_{ij} \text{ rem } A \quad (i, j \in [1, t]).$$

Nous notons R l'ensemble de toutes les nouvelles équations qu'il est ainsi possible de calculer (A étant fixé).

Informellement, l'algorithme consiste à enrichir Σ tant que c'est possible. Lorsque le système ne peut plus être complété, l'algorithme procède à un scindage sur les initiaux et les séparants des éléments de A . L'un des systèmes ainsi obtenus est régulier au sens de la définition 14 (page 35); on applique à nouveau le traitement ci-dessus sur les autres.

Nous donnons l'algorithme sous la forme d'un jeu de règles numérotées, qui portent en commentaire les conditions sous lesquelles elles s'appliquent.

règle 1 si R contient au moins un polynôme non identiquement nul, l'algorithme enrichit Σ avec R .

règle 2 si toutes les nouvelles équations sont nulles (si $R = \{0\}$), l'algorithme procède à un scindage sur les initiaux et les séparants des éléments de A , notés H_ℓ ($1 \leq \ell \leq 2t$) et engendre $2t + 1$ systèmes. Les systèmes générés admettent des modèles différentiels

disjoints :

$$\begin{array}{l}
 \Sigma \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ p_i = 0 \\ q_j \neq 0 \end{array} \right. \\
 \left. \begin{array}{l} \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{array} \right. \begin{array}{l} \Sigma_1 \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_1 = 0 \\ p_i = 0 \\ q_j \neq 0 \end{array} \right. \\ \text{etc ...} \\ \Sigma_{2t} \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_{2t} = 0, H_{2t-1} \neq 0, \dots, H_1 \neq 0 \\ p_i = 0 \\ q_j \neq 0 \end{array} \right. \\ \Sigma_{2t+1} \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_A \neq 0 \\ p_i = 0 \\ q_j \neq 0. \end{array} \right. \end{array}
 \end{array}$$

Pour rendre Σ_{2t+1} régulier, l'algorithme supprime les équations p_i et réduit partiellement par A les inéquations q_j . Le système Ω obtenu est régulier au sens de la définition 14 (page 35) :

$$\Sigma_{2t+1} \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_A \neq 0 \\ p_i = 0 \\ q_j \neq 0 \end{array} \right. \longrightarrow \Omega \quad \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_A \neq 0 \\ \bar{q}_j = q_j \text{ rem-partiel } A \neq 0. \end{array} \right.$$

A propos de ces règles

- On peut optimiser la règle 2 en supprimant le monôme de tête de l'équation $A_\ell = 0$, dans le système Σ_ℓ (celui où l'on pose $I_\ell = 0$), et réécrire $A_\ell = 0$ en $(\deg_{u_\ell} A_\ell \cdot A_\ell - u_\ell \cdot S_\ell) = 0$ dans $\Sigma_{2\ell}$ (où l'on pose $S_\ell = 0$).
- Le choix de l'ensemble caractéristique A du système courant Σ peut influencer sur le comportement de l'algorithme. Considérons par exemple le système suivant :

$$\Sigma \quad \left\{ \begin{array}{l} x = 0 \\ y \cdot x = 0. \end{array} \right.$$

Nous supposons que l'indéterminée principale de la deuxième équation est égale à x . Le système admet deux ensembles caractéristiques : x et $(y \cdot x)$. Si nous choisissons le premier, nous obtenons, après réduction en 0 de $(y \cdot x)$, un système régulier :

$$\Omega \quad \left\{ \begin{array}{l} x = 0. \end{array} \right.$$

Si nous choisissons le second, la règle 2 nous conduit à un scindage sur y , après réduction en 0 de x :

$$\Sigma_1 \quad \begin{cases} x = 0 \\ y = 0 \end{cases} \quad \text{et} \quad \Sigma_2 \quad \begin{cases} y \cdot x = 0 \\ y \neq 0. \end{cases}$$

Σ_1 et Σ_2 sont réguliers.

5.1.1 Preuve d'arrêt

Les systèmes dont les équations comportent un élément non nul de K , n'admettent bien sûr pas de modèles. Nous supposons donc qu'aucun élément de R , autre que zéro n'est dans K .

Les polynômes produits par réduction par A , de même que les initiaux et les séparants des éléments de l'ensemble caractéristique sont réduits par rapport à A . D'après le corollaire du théorème 2 (page 21), les deux règles font décroître A strictement. D'après le théorème 3 (page 22), il n'existe pas de suite infinie strictement décroissante d'ensembles auto-réduits : l'algorithme s'arrête après un nombre fini de réécritures.

5.1.2 Preuves de correction

La règle 1 enrichit Σ avec des équations qui appartiennent à l'idéal différentiel engendré par les équations de Σ . D'après le théorème 5, cette réécriture ne modifie pas les modèles différentiels de Σ .

Après application de la règle 2, il est clair que Σ admet un modèle (algébrique ou différentiel) Φ , si et seulement si Φ est modèle de l'un des Σ_i . Montrons que la réécriture de Σ_{2t+1} en Ω préserve les modèles différentiels de Σ_{2t+1} . Cette réécriture s'applique quand $R = \{0\}$. D'après les spécifications de l'algorithme de réduction, nous avons donc pour toute équation p_i et toute inéquation q_j de Σ_{2t+1} :

$$H_A^\alpha \cdot p_i \in [A],$$

$$S_A^\beta \cdot q_j \equiv \bar{q}_j \pmod{[A]},$$

où S_A désigne le produit des séparants des éléments A_ℓ de A , où H_A désigne le produit des initiaux et des séparants des A_ℓ et où α et β sont certains entiers, dépendant de p_i et de q_j .

Tout morphisme d'anneaux différentiels qui annule A annule $(S_A^\beta \cdot q_h - \bar{q}_h)$ (lemme 11 (page 28)) et donc tout morphisme d'anneaux différentiels qui n'annule pas l'un des deux termes de la différence, n'annule pas l'autre non plus. Rappelons qu'un modèle différentiel est un morphisme d'anneaux différentiels à image dans un domaine intègre. Les modèles différentiels de Σ_{2t+1} n'annulent ni q_j , ni S_A (qui apparaît en facteur dans H_A). Ils sont donc des modèles différentiels de Ω . Réciproquement, les modèles différentiels de Ω n'annulent pas q_j . Ils annulent $(H_A^\alpha \cdot p_i)$ (puisqu'ils annulent A), mais pas H_A . Ils annulent donc p_i et sont des modèles différentiels de Σ_{2t+1} .

5.1.3 Système principal associé à Σ

Parmi les systèmes réguliers produits par l'algorithme ROSENFELD–GRÖBNER à partir d'une famille génératrice d'un idéal différentiel, il en existe un, "plus général" que les autres. Il s'agit du système dans lequel seules les relations de l'idéal $[\Sigma]$ sont supposées satisfaites :

Définition 25 Soient Σ un système d'équations polynomiales différentielles non contradictoire et (Ω_i) la famille de systèmes réguliers produite par l'algorithme ROSENFELD–GRÖBNER pour un quelconque ordre admissible sur ΘX . On appelle système principal associé à Σ l'unique système Ω_ℓ vérifiant

- Ω_ℓ admet des modèles différentiels.
- les équations du système Ω_ℓ appartiennent au radical de l'idéal différentiel $[\Sigma]$.
- les équations du système Ω_ℓ réduisent à zéro (par l'algorithme de la section 1.3.1) les éléments d'une famille génératrice de l'idéal différentiel $[\Sigma]$.

Montrons que l'algorithme ROSENFELD–GRÖBNER produit un unique système vérifiant les deux premières conditions données dans la définition. Soit I le radical de l'idéal différentiel engendré par le système de départ.

Cas initial : les équations du système appartiennent bien à I .

Cas général : soit Σ un système dont les équations sont dans I . La règle 1 enrichit Σ avec des équations qui sont dans $[\Sigma]$, donc dans I . On conclut la démonstration en montrant que, parmi les systèmes Σ_i non contradictoires engendrés par la règle 2, celui dont l'indice est maximal est le seul dont les équations soient dans l'idéal : notons H_i ($1 \leq i \leq 2t$) les initiaux et les séparants des éléments de A . Si aucun H_i n'appartient à I , alors les systèmes Σ_1 à Σ_{2t} contiennent une équation qui n'est pas dans I et le système principal est Ω , obtenu à partir de Σ_{2t+1} . Dans le cas contraire, soit ℓ minimal tel que H_ℓ soit dans I . Les systèmes $\Sigma_{\ell+1}$ à Σ_{2t+1} sont sans modèle différentiel puisqu'ils contiennent l'inéquation $H_\ell \neq 0$, les systèmes Σ_1 à $\Sigma_{\ell-1}$ contiennent une équation qui n'est pas dans l'idéal : Σ_ℓ est le système non contradictoire dont l'indice est maximal. Toutes ses équations sont dans I .

Au vu des réécritures, on s'assure aisément que le système principal réduit à zéro une famille génératrice de l'idéal différentiel $[\Sigma]$. A noter également : si l'optimisation de la règle 2 proposée en fin de section 5.1 est implantée, cette famille n'est pas nécessairement Σ .

Il est possible d'implanter les réécritures de la règle 2 de l'algorithme ROSENFELD–GRÖBNER de telle sorte que le système principal soit le premier des systèmes non contradictoires produits : il suffit de parcourir l'arbre des scindages suivant une méthode droite-racine-gauche.

5.2 Calcul des bases de Gröbner

Soit Ω un système régulier. Ω admet un modèle différentiel si et seulement s'il admet un modèle algébrique. Parce que nous cherchons nos modèles dans des corps, dont tout élément non nul est inversible, les modèles algébriques de Ω sont en bijection rationnelle avec ceux du système Ω' obtenu en transformant chaque inéquation en équation, grâce à de nouvelles indéterminées z_j :

$$\Omega \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_1 \neq 0 \\ \vdots \\ H_{2t} \neq 0 \\ q_j \neq 0 \end{array} \right. \quad \text{donne} \quad \Omega' \left\{ \begin{array}{l} A_1 = 0, \dots, A_t = 0 \\ H_1 \cdot z_1 = 1 \\ \vdots \\ H_{2t} \cdot z_{2t} = 1 \\ q_j \cdot z_j = 1 \end{array} \right.$$

Notons x_i les indéterminées qui apparaissent dans l'écriture des équations et des inéquations de Ω et q le produit des inéquations q_j .

On peut décider de l'existence de modèles algébriques pour Ω en calculant une base de GRÖBNER pour n'importe quel ordre admissible (définition 17 (page 59)) sur les monômes de l'anneau $K[x_i, z_j]$.

Si par contre, on calcule la base B pour un ordre d'élimination où les monômes indépendants des z_j sont plus petits que ceux qui en dépendent¹ alors l'ensemble $B \cap K[x_i]$ des polynômes de B dans l'écriture desquels les indéterminées z_j n'apparaissent pas forme une base de GRÖBNER de l'idéal $(A) : (H_A \cdot q)^\infty$ (théorème 11 (page 62)). cet idéal ne décrit pas exactement les modèles de Ω mais leur adhérence de ZARISKI, c'est-à-dire le plus petit ensemble algébrique² contenant les modèles du système.

Définition 26 Soit Ω un système régulier de $K\{X\}$. Soient A l'ensemble de ses équations, H_A et q ses inéquations. Nous appelons base de GRÖBNER associée à Ω toute base de GRÖBNER de l'idéal $(A) : (H_A \cdot q)^\infty$ de $K\{X\}$.

Nous appellerons *base de GRÖBNER principale associée à Σ* la base de GRÖBNER associée au système principal calculé à partir de Σ par l'algorithme ROSENFELD-GRÖBNER.

A propos des calculs de bases de Gröbner

- Soit Ω un système régulier et A l'ensemble de ses équations. Les séparants des éléments A_ℓ de A contiennent en facteur les racines multiples des A_ℓ . Le calcul de la base de $(A) : H_A^\infty$ va ainsi “simplifier” les A_ℓ et les rendre *square-free*.

Nous avons observé sur des exemples que les bases de GRÖBNER non contradictoires calculées à partir d'une famille d'équations p_1, \dots, p_n sont les mêmes que celles obtenues après avoir élevé les générateurs p_i à diverses puissances. Ainsi, il semble que les bases calculées soient “plus canoniques” que ce chapitre ne le laisse penser.

1. c'est-à-dire un ordre lexicographique du type : $x_i < z_j$.

2. qui puisse ne se décrire que par des *équations* polynomiales.

5.3 Ce que l'algorithme fait

Dans cette section, nous décrivons un certain nombre d'utilisations possibles de l'algorithme ROSENFELD–GRÖBNER.

5.3.1 Décision du vide

Le calcul des bases de GRÖBNER détecte les systèmes réguliers sans modèles. Ainsi :

Proposition 3 *Soit Σ un système d'équations et d'inéquations polynomiales différentielles de $K\{X\}$. Le système Σ est sans modèles différentiels si et seulement si les bases de GRÖBNER calculées par l'algorithme ROSENFELD–GRÖBNER pour un quelconque ordre admissible sur ΘX sont toutes égales au singleton $\{1\}$.*

5.3.2 Test d'appartenance au radical d'un idéal différentiel

La proposition suivante montre comment réutiliser les bases de GRÖBNER calculées pour décider après-coup de l'appartenance au radical de l'idéal différentiel engendré par les équations de départ.

Proposition 4 *Soit Σ un sous-ensemble fini de $K\{X\}$. Soit (Ω_i) ($i \in I$) la famille de systèmes réguliers produite par l'algorithme ROSENFELD–GRÖBNER à partir de Σ pour un quelconque ordre admissible sur ΘX . Soient A_i et B_i respectivement l'ensemble des équations de Ω_i et la base de GRÖBNER associée à Ω_i .*

Un polynôme p de $K\{X\}$ appartient au radical de l'idéal différentiel $[\Sigma]$ engendré par Σ si et seulement si, pour tout i ($i \in I$), une puissance du polynôme (p rem-partiel A_i) appartient à l'idéal (B_i) . En d'autres termes :

$$p \in \sqrt{[\Sigma]} \quad \Leftrightarrow \quad \forall i \in I, \quad \exists \alpha \in \mathbb{N}, \quad (p \text{ rem-partiel } A_i)^\alpha \in (B_i).$$

Preuve D'après le théorème 5, p appartient au radical de l'idéal $[\Sigma]$ si et seulement si le système obtenu en rajoutant l'inéquation $p \neq 0$ à Σ n'admet aucun modèle différentiel c'est-à-dire, si et seulement si pour tout i ($1 \leq i \leq s$), le système Ω'_i obtenu en rajoutant l'inéquation $p \neq 0$ à Ω_i n'admet aucun modèle différentiel.

Les Ω'_i ne sont pas nécessairement réguliers, puisque p n'est pas nécessairement partiellement réduit par rapport aux A_i mais, comme H_{A_i} figure parmi les inéquations des Ω'_i , on ne change pas les modèles différentiels de ces systèmes en réduisant partiellement p par A_i .

Les systèmes réguliers ainsi obtenus sont sans modèles si et seulement si les bases de GRÖBNER calculées en rajoutant l'équation $p \cdot z = 1$ aux bases B_i sont égales au singleton $\{1\}$. \square

5.3.3 Test d'appartenance à un idéal différentiel premier

L'algorithme du paragraphe ci-dessus se spécialise très agréablement lorsque l'idéal différentiel engendré par les équations du système de départ est premier (proposition 5).

Nous commençons par établir un lemme qui ressemble au lemme 19 (page 65) mais qui est en réalité un peu plus fort : A n'est pas nécessairement un ensemble caractéristique de l'idéal mais l'ensemble des équations du système principal associé à un système Σ .

Lemme 22 *Soit Σ un sous-ensemble fini de $K\{X\}$ tel que l'idéal différentiel $[\Sigma]$ soit premier. Soit Ω le système principal (définition 25 (page 72)) associé à Σ produit par l'algorithme ROSENFELD–GRÖBNER pour un quelconque ordre admissible sur ΘX .*

$$\Omega \quad \begin{cases} A = 0 \\ H_A \neq 0 \\ q \neq 0 \end{cases}$$

On a alors :

$$[\Sigma] = [A] : (H_A \cdot q)^\infty.$$

Preuve Comme A réduit à zéro une famille génératrice de l'idéal $[\Sigma]$, on a l'inclusion (\subset) de gauche à droite.

Pour montrer l'inclusion inverse (\supset) , montrons que $(H_A \cdot q)$ n'appartient pas à $[\Sigma]$.

Si ce produit était dans l'idéal alors une de ses puissances serait dans $[A]$ (nous avons démontré $[\Sigma] \subset [A] : (H_A \cdot q)^\infty$ ci-dessus) et Ω serait sans modèles différentiels or, un calcul de base de GRÖBNER a établi le contraire.

Concluons la démonstration du lemme : soit p un polynôme tel que $((H_A \cdot q)^\alpha \cdot p)$ soit dans $[A]$. D'après la définition des systèmes principaux, A est dans $[\Sigma]$, donc le produit $((H_A \cdot q)^\alpha \cdot p)$ y est aussi. Comme $[\Sigma]$ est premier et comme $(H_A \cdot q)$ ne lui appartient pas, c'est p qui est dans l'idéal. \square

Proposition 5 *Soient Σ un sous-ensemble fini de $K\{X\}$ tel que l'idéal différentiel $[\Sigma]$ soit premier, Ω le système principal associé à Σ produit par l'algorithme ROSENFELD–GRÖBNER pour un quelconque ordre admissible sur ΘX et B la base de GRÖBNER qui lui est associée.*

$$\Omega \quad \begin{cases} A = 0 \\ H_A \neq 0 \\ q \neq 0. \end{cases}$$

Un polynôme p de $K\{X\}$ appartient à l'idéal différentiel $[\Sigma]$ si et seulement si le polynôme $(p \text{ rem-partiel } A)$ appartient à l'idéal (B) . En d'autres termes :

$$p \in [\Sigma] \quad \Leftrightarrow \quad (p \text{ rem-partiel } A) \in (B).$$

Preuve L'implication de gauche à droite est simple (elle n'exige ni que $[\Sigma]$ soit premier, ni que Ω soit le système principal).

Soit p un polynôme de $[\Sigma]$. Comme A réduit à zéro une famille génératrice de l'idéal, p appartient à $[A] : (H_A \cdot q)^\infty$. Le polynôme r égal à $(p \text{ rem-partiel } A)$ appartient aussi à $[A] : (H_A \cdot q)^\infty$ et est partiellement réduit par rapport à A . D'après le lemme 12 (page 33) de ROSENFELD, il est dans l'idéal $(A) : (H_A \cdot q)^\infty$, c'est-à-dire dans (B) (théorème 11 (page 62)).

L'implication de droite à gauche. L'idéal (B) , égal à $(A) : (H_A \cdot q)^\infty$, est inclus dans $[A] : (H_A \cdot q)^\infty$. D'après le lemme précédent, B est incluse dans $[\Sigma]$. \square

5.3.4 Calcul d'un ensemble caractéristique

Nous montrons comment calculer un ensemble caractéristique à partir d'une famille génératrice finie Σ d'un idéal différentiel premier $[\Sigma]$. Il s'agit bien d'un ensemble caractéristique de RITT (théorème 2 (page 21)) et non de WU [Wu]; voir à ce propos [H]. Il s'agit du premier algorithme "général" qui calcule ces ensembles caractéristiques en algèbre différentielle (voir [Di3]).

Définition 27 Soit p un polynôme de $K\{X\}$ d'indéterminée principale u et de degré d en cette indéterminée. Nous appelons terme principal de p le terme $lt(p) = u^d$.

Tous les ensembles caractéristiques d'un ensemble E ont même ensemble de termes principaux.

Définition 28 Soit I un idéal de $K\{X\}$. Nous dirons que u^d est un terme principal non dégénéré de I , ce que nous notons $u^d \in lt^*(I)$, si u^d est le terme principal d'un polynôme p de I dont l'initial n'appartient pas à I .

En algèbre différentielle ordinaire, pour l'ordre d'élimination $\theta x < \phi y$ (pour tous opérateurs de dérivation θ et ϕ), considérons l'idéal: $I = [x, \dot{y}^3 + 1]$. Les termes y^2 et \dot{y} ne font pas partie des termes principaux non dégénérés de I , bien que les polynômes xy^2 et $x\dot{y}$ par exemple appartiennent à l'idéal.

Définition 29 Soit I un idéal de $K\{X\}$. Nous dirons que u^d est un terme principal minimal de I , ce que nous notons $u^d \in ltm(I)$, si u^d un terme principal non dégénéré de I vérifiant: si v^e est un terme principal non dégénéré de I alors e est supérieur ou égal à d ; si v^e est un terme principal non dégénéré de I alors u n'est pas une dérivée propre de v . En d'autres termes:

$$u^d \in ltm(I) \quad \text{si} \quad \begin{cases} u^d \in lt^*(I) & \text{et} \\ u^e \in lt^*(I) & \Rightarrow e \geq d \quad \text{et} \\ v^e \in lt^*(I) & \Rightarrow \forall \theta \in \Theta, \quad u \neq \theta v \quad \text{ou} \quad \theta = 1. \end{cases}$$

Les termes principaux minimaux de l'idéal I donné en exemple ci-dessus sont x et \dot{y}^3 .

Voici les grandes lignes de la preuve de l'algorithme.

Nous établissons pour commencer que, dans le cas d'un idéal différentiel premier $[\Sigma]$, l'ensemble des termes principaux minimaux de l'idéal est égal à l'ensemble des termes principaux des éléments des ensembles caractéristiques de $[\Sigma]$ (lemme 24) et fait partie de l'ensemble des termes principaux de la base de GRÖBNER principale calculée par l'algorithme ROSENFELD-GRÖBNER à partir de Σ (proposition 6).

Il est facile d'extraire de B un ensemble de polynômes ayant mêmes termes principaux que tout ensemble caractéristique de l'idéal. Pour terminer, nous montrons qu'on peut construire un ensemble auto-réduit à partir de cet ensemble, sans en changer les termes principaux.

Le lemme auxiliaire suivant sera utilisé à plusieurs reprises:

Lemme 23 Soit J un idéal différentiel premier de $K\{X\}$. Soit q un polynôme de I , d'indéterminée principale u , dont l'initial n'appartient pas à J . Si $P = \{p_1, \dots, p_s\}$ est une famille d'éléments de J , dont ni les initiaux, ni les séparants, n'appartiennent à J et dont aucune dérivée n'ait u pour indéterminée principale alors, le terme principal de $q \text{ rem } P$ est égal au terme principal de q .

Preuve Le reste $r = q \text{ rem } P$ est égal à $r = H_P^\alpha \cdot q - T$ où :

$$(1) \quad T = \sum_{i=1}^s \sum_{\theta} C_{i,\theta} \cdot \theta p_i.$$

Au vu de l'algorithme de réduction, les termes principaux de T et de H_P^α sont inférieurs³ ou égaux au terme principal de q . Comme aucune dérivée, d'aucun p_i n'admet u pour indéterminée principale les polynômes $(H_P^\alpha \cdot q)$ et q ont même indéterminée principale. Pour la même raison, dans la partie droite de l'égalité (1), le terme principal de q ne peut figurer que dans l'écriture des coefficients $C_{i,\theta}$. Ainsi, ou bien T est de terme principal inférieur à celui de q et le lemme est prouvé, ou bien T a même terme principal que q et l'initial de T appartient à $[P]$, donc à J .

Supposons égaux les termes principaux de q et de T et, pour conclure la démonstration, montrons que la différence $(I_T - I_{H_P^\alpha \cdot q})$ est non nulle. Il suffit d'établir que l'initial de $(H_P^\alpha \cdot q)$ n'appartient pas à J : ni les initiaux et séparants des p_i , ni I_q n'appartiennent à l'idéal. Comme celui-ci est premier, l'initial du produit $(H_P^\alpha \cdot q)$ ne lui appartient pas. \square

Lemme 24 Soient J un idéal différentiel premier de $K\{X\}$ et C un ensemble caractéristique de J . L'ensemble des termes principaux de C est égal à l'ensemble des termes principaux minimaux de J . En d'autres termes :

$$lt(C) = ltm(J).$$

Preuve L'inclusion (\supset) de droite à gauche. Soit u^d un terme principal minimal de J et p un polynôme de l'idéal de la forme :

$$p = I_p \cdot u^d + R_p \quad (\text{avec } I_p \notin J).$$

Considérons $r = p \text{ rem } C$. On a $r \equiv H_C^\alpha \cdot p \pmod{J}$. D'après le lemme 19 (page 65), comme p appartient à l'idéal, le reste r est nul or, si nous supposons que u^d n'est le terme principal d'aucun élément de C , d'après le lemme 23, p et r ont même terme principal. Cette contradiction prouve l'inclusion.

L'inclusion (\subset) de gauche à droite. Soit u^d le terme principal d'un polynôme p de C . L'initial de p n'appartient pas à l'idéal donc $lt(p) \in lt^*(J)$. Nous avons démontré $ltm(J) \subset lt(C)$. Comme tout ensemble caractéristique est auto-réduit, u n'est la dérivée d'aucun élément de $lt^*(J)$ et d est minimal. \square

Appelons x_i les indéterminées qui apparaissent dans l'écriture des équations et des inéquations du système principal et z_j les indéterminées artificielles, introduites pour

3. nous devons être plus précis que d'habitude : H_P^α désigne ici le produit de puissances des initiaux et des séparants des p_i qui ont *effectivement* servi aux réductions.

convertir les inéquations en équations lors des calculs de bases de GRÖBNER. Jusqu'à présent, nous supposons que le calcul de la base de GRÖBNER principale B s'effectuait suivant un ordre d'élimination où les monômes indépendants des z_j sont plus petits que ceux qui en dépendent⁴. Pour la proposition suivante, nous faisons une hypothèse supplémentaire sur cet ordre lexicographique: nous supposons que les indéterminées x_i sont ordonnées entre elles en respectant l'ordre admissible sur ΘX pour lequel le système principal a été calculé.

Proposition 6 *Soit Σ un sous-ensemble fini de $K\{X\}$ tel que l'idéal différentiel $[\Sigma]$ soit premier et Ω le système principal associé à Σ produit par l'algorithme ROSENFELD–GRÖBNER pour un ordre admissible \mathcal{R}_1 sur ΘX .*

Si la base de GRÖBNER B associée à Ω est calculée pour un ordre lexicographique \mathcal{R}_2 qui respecte \mathcal{R}_1 alors l'ensemble des termes principaux minimaux de l'idéal $[\Sigma]$ est inclus dans l'ensemble des termes principaux des éléments de B :

$$\text{ltm}([\Sigma]) \subset \text{lt}(B).$$

Preuve Soient u^d un terme principal minimal de $[\Sigma]$ et p un polynôme de $[\Sigma]$ dont l'initial n'appartient pas à l'idéal.

Nous pouvons supposer que p appartient à l'idéal (B) : d'après le lemme 22 (page 75), les initiaux et les séparants des équations A du système principal Ω n'appartiennent pas à $[\Sigma]$ donc, d'après le lemme 23, le reste partiel de p par A a même terme principal que p .

Vu les spécifications de \mathcal{R}_2 , le monôme de tête de p s'écrit $\text{lm}(p) = m_p \cdot u^d$ et nous pouvons supposer que m_p est sous forme normale modulo la base B .

D'après la définition des bases de GRÖBNER (théorème 8 (page 60)), B contient un élément b dont le monôme de tête $\text{lm}(b) = m_b \cdot u^e$ divise $\text{lm}(p)$. Comme m_p est sous forme normale, le degré e est strictement positif. L'initial de b n'appartient pas à (B) . D'après le lemme 12 (page 33) de ROSENFELD, il n'appartient pas non plus à $[\Sigma]$. D'après la définition des termes principaux (la minimalité de d), le degré e est égal à d . \square

D'après le lemme 24 et la proposition 6, si on extrait de B la plus longue suite de polynômes b_1, \dots, b_s dont les indéterminées principales u_1, \dots, u_s forment un ensemble auto-réduit d'indéterminées, on obtient un ensemble de polynômes qui n'est pas nécessairement auto-réduit au sens de RITT (définition 7 (page 20)), mais dont les termes principaux sont ceux d'un ensemble caractéristique de l'idéal différentiel premier $[\Sigma]$.

Il ne nous manque que le lemme suivant pour conclure la construction.

Lemme 25 *Soit Σ un sous-ensemble fini de $K\{X\}$ tel que l'idéal différentiel $[\Sigma]$ soit premier, Ω le système principal associé à Σ produit par l'algorithme ROSENFELD–GRÖBNER pour un ordre admissible \mathcal{R}_1 sur ΘX et B la base de GRÖBNER associée à Ω , calculée pour un ordre lexicographique \mathcal{R}_2 qui respecte \mathcal{R}_1 .*

4. c'est-à-dire un ordre lexicographique quelconque du type $x_i < z_j$.

Les initiaux et les séparants des éléments b de B dont le terme principal est un terme principal minimal de l'idéal $[\Sigma]$ n'appartiennent pas à l'idéal $[\Sigma]$. En d'autres termes :

$$lt(b) \in ltm([\Sigma]) \quad \Rightarrow \quad (I_b \notin [\Sigma] \quad \text{et} \quad S_b \notin [\Sigma]).$$

Preuve D'après le lemme 12 de ROSENFELD, il suffit de montrer que ces polynômes n'appartiennent pas à l'idéal (B) engendré par la base de GRÖBNER.

Le cas des initiaux est immédiat.

Les séparants : soit $b = I_b \cdot u^d + R_b$ un élément de B dont le terme principal u^d est un terme principal minimal de l'idéal $[\Sigma]$. Si d est égal à 1, le séparant S_b de b est égal à I_b sinon on a $S_b = d \cdot I_b \cdot u^{d-1} + T_b$. L'initial du séparant n'appartient pas à l'idéal, si S_b appartenait à l'idéal, le terme u^d ne serait pas un terme principal minimal de $[\Sigma]$ (la minimalité de d). \square

Du lemme ci-dessus et du lemme 23 il ressort que si on réduit entre eux, par l'algorithme de RITT (section 1.3.1), les éléments de B dont le terme principal est un terme principal minimal de l'idéal différentiel $[\Sigma]$, on obtient un ensemble auto-réduit au sens de RITT (définition 7 (page 20)) dont l'ensemble des termes principaux est égal à l'ensemble des termes principaux minimaux de l'idéal : un ensemble caractéristique de l'idéal.

Chapitre 6

Implantations — Applications — Comparaisons

6.1 Les algorithmes

6.1.1 Algorithme Rosenfeld-Gröbner

Nous en donnons deux versions.

Les programmes principaux sont *rosenfeld1* et *rosenfeld2*. Ils correspondent à deux stratégies différentes de calcul de systèmes réguliers à partir d'un système d'équations et d'inéquations passé en paramètre. Seuls sont produits les systèmes réguliers qui admettent des modèles différentiels.

Les deux programmes font appel aux trois fonctions auxiliaires qui suivent. Les lettres grecques servent d'identificateurs aux systèmes d'équations et d'inéquations, les majuscules aux simples ensembles d'équations ou d'inéquations.

La fonction *trivialementSansModèles* (Λ) retourne *vrai* si un élément non nul de K figure parmi les équations, ou si 0 figure parmi les inéquations de Λ . Elle retourne *faux* sinon.

La fonction *scindagesSurH_A* (Λ) retourne une *liste de systèmes* \mathcal{F} , obtenus par scindages sur les initiaux et les séparants des éléments d'un ensemble caractéristique A des équations de Λ . L'ensemble des autres équations de Λ est appelé E ; l'ensemble des inéquations est I . La partition (A, E) de l'ensemble des équations est fournie par la procédure appelante.

En gérant \mathcal{F} comme une liste plutôt que comme un ensemble, on assure que l'arbre des scindages est parcouru suivant une méthode *droite-racine-gauche*: les systèmes les plus généraux (ceux dans lesquels un maximum de H_i sont supposés non nuls) se trouvent en tête de liste. Les éléments de \mathcal{F} admettent des modèles différentiels disjoints.

fonction *scindagesSurH_A* (A, E, I) **retourne** \mathcal{F}
locales Ω, L
début

```

 $\mathcal{F} := \square$ 
 $L := \emptyset$ 
pour tout  $H \in H_A$  faire
     $\Omega_{\text{équations}} := A \cup E \cup \{H\}$ 
     $\Omega_{\text{inéquations}} := I \cup L$ 
     $\mathcal{F} := \text{cons}(\Omega, \mathcal{F})$ 
     $L := L \cup \{H\}$ 
fin
retourner  $\mathcal{F}$ 
fin

```

La procédure *produireSiNonContradictoire* (Λ) produit en sortie le système Λ s'il admet des modèles différentiels. Appelons x_i les indéterminées effectivement présentes dans les équations et les inéquations de Λ , et z_j les nouvelles indéterminées introduites pour convertir les inéquations en équations. L'algorithme calcule une base de GRÖBNER pour un quelconque ordre admissible sur les monômes de $K[x_i, z_j]$.

```

procédure produireSiNonContradictoire ( $\Lambda$ )
locales     $B, F$ 
début
     $F := \Lambda_{\text{équations}}$ 
    pour tout  $H \in \Lambda_{\text{inéquations}}$  faire
        créer une nouvelle indéterminée  $z_j$ 
         $F := F \cup \{H \cdot z_j - 1\}$ 
    fin
     $B := \text{Gröbner}(F)$ 
    si  $B \neq \{1\}$  alors produire ( $\Omega$ ) fin
fin

```

rosenfeld1 (Λ, \mathcal{R}) procède aux scindages le plus rapidement possible. \mathcal{R} désigne un ordre admissible sur ΘX . Soit A un ensemble caractéristique des équations de Λ . Le programme substitue aux autres équations leur reste par A sous la contrainte $H_A \neq 0$. Cette stratégie semble plus efficace que celle de *rosenfeld2*.

Note Nous ne prouverons pas que *rosenfeld1* produit un système principal. Et pour cause : comme cet algorithme substitue leur reste aux polynômes qu'il réduit et que la réduction n'est pas transitive, il se peut que le "système principal" ne réduise pas à zéro la famille génératrice Σ . Il est toutefois facile de modifier l'algorithme pour qu'il engendre ce système.

```

programme rosenfeld1 ( $\Lambda, \mathcal{R}$ )
locales     $A, R, \Omega, \mathcal{F}$ 
début
    si non trivialementSansModèles ( $\Lambda$ ) alors
         $A :=$  un ensemble caractéristique de  $\Lambda_{\text{équations}}$ 
         $R := (\Lambda_{\text{équations}} \setminus A \cup \Delta\text{-polynômes}(A)) \text{ rem } A$ 
        si  $R = \emptyset$  ou  $R = \{0\}$  alors
             $\Omega_{\text{équations}} := A$ 

```

```

     $\Omega_{\text{inéquations}} := (\Lambda_{\text{inéquations rem-partiel } A}) \cup H_A$ 
    produireSiNonContradictoire ( $\Omega$ )
sinon
     $\Omega_{\text{équations}} := A \cup R$ 
     $\Omega_{\text{inéquations}} := \Lambda_{\text{inéquations}} \cup H_A$ 
    rosenfeld1 ( $\Omega$ )
fin
 $\mathcal{F} := \text{scindagesSur}H_A (A, \Lambda_{\text{équations}} \setminus A, \Lambda_{\text{inéquations}})$ 
appliquer rosenfeld1 sur tous les systèmes de  $\mathcal{F}$ 
fin
fin

```

rosenfeld2 (Λ, \mathcal{R}) retarde le plus longtemps possible les scindages sur H_A . Ce faisant, il augmente (inconsidérément?) la masse des équations à traiter après les scindages. \mathcal{R} désigne un ordre admissible sur ΘX .

Cet algorithme correspond exactement aux réécritures données en début de chapitre.

Si le système de départ Σ est un système d'équation alors, le premier système régulier non contradictoire produit est le système principal associé à Σ .

```

programme rosenfeld2 ( $\Lambda, \mathcal{R}$ )
locales       $A, R, \Omega, \mathcal{F}$ 
début
    répéter
         $A :=$  un ensemble caractéristique de  $\Lambda_{\text{équations}}$ 
         $R := (\Lambda_{\text{équations}} \setminus A \cup \Delta\text{-polynômes}(A)) \text{ rem } A$ 
         $\Lambda_{\text{équations}} := \Lambda_{\text{équations}} \cup R$ 
    jusqu'à trivialementSansModèles ( $\Lambda$ ) ou  $R = \emptyset$  ou  $R = \{0\}$ 
    si non trivialementSansModèles ( $\Lambda$ ) alors
         $\Omega_{\text{équations}} := A$ 
         $\Omega_{\text{inéquations}} := (\Lambda_{\text{inéquations rem-partiel } A}) \cup H_A$ 
        produireSiNonContradictoire ( $\Omega$ )
         $\mathcal{F} := \text{scindagesSur}H_A (A, \Lambda_{\text{équations}} \setminus A, \Lambda_{\text{inéquations}})$ 
    appliquer rosenfeld2 sur tous les systèmes de  $\mathcal{F}$ 
    fin
fin

```

Dans la pratique, un certain nombre d'optimisations heuristiques peuvent être apportées aux algorithmes ci-dessus. Voici celles que nous avons implantées :

- Un ensemble d'équations peut admettre plusieurs ensembles caractéristiques. On diminue le nombre de scindages en choisissant de préférence ceux dont un maximum d'équations admettent des initiaux dans K , ou dont l'initial et le séparant figurent parmi les inéquations (parce qu'elles ont fait partie de précédents ensembles caractéristiques, dans le cas de *rosenfeld1*).
- Dans le cas de *rosenfeld1*, il n'est pas toujours utile de procéder à un scindage sur tous les polynômes de H_A . On peut se restreindre aux initiaux et aux séparants des éléments de A qui ont effectivement servi aux réductions.

- Le nombre de scindages diminue d’une manière impressionnante lorsqu’on simplifie les équations par des factorisations élémentaires : la seule équation $x^{10} = 0$ par exemple engendre de nombreux scindages, dus aux extractions répétées de séparants. Elle se simplifie agréablement en $x = 0$.

De telles équations apparaissent très fréquemment après quelques calculs.

- Une autre manière de “couper” rapidement des “branches mortes” dans l’arbre de systèmes engendrés par l’algorithme consiste à tenter des divisions par les inéquations : $p \cdot q = 0$, $q \neq 0$ peut par exemple se simplifier en $p = 0$, $q \neq 0$.

Lorsqu’il arrive que l’algorithme enchaîne un certain nombre de réductions algébriques, on ralentit ainsi (de façon maladroite) la croissance des coefficients des restes successifs, connue au moins depuis l’algorithme des sous-résultants (voir [Col]).

Cette optimisation anticipe en partie le calcul de bases de GRÖBNER effectué dans *produireSiNonContradictoire*.

6.1.2 Calcul d’ensemble caractéristique

Le programme *ensembleCaractéristique* (Σ, \mathcal{R}_1) produit un ensemble caractéristique de l’idéal différentiel $[\Sigma]$ si celui-ci est premier, pour l’ordre admissible \mathcal{R}_1 .

programme *ensembleCaractéristique* (Σ, \mathcal{R}_1)

locales $B, C, F, \Omega, \mathcal{R}_2$

début

— *Calcul du système principal Ω associé à Σ .*

$\Omega :=$ le premier système régulier produit par *rosenfeld2* (Σ, \mathcal{R}_1)

— *Appelons x_i les indéterminées effectivement présentes dans les équations et les inéquations de Ω . Les instructions suivantes calculent la base de GRÖBNER principale associée à Ω .*

$F := \Omega$ _équations

pour tout $H \in \Omega$ _inéquations **faire**

créer une nouvelle indéterminée z_j

$F := F \cup \{H \cdot z_j - 1\}$

fin

$\mathcal{R}_2 :=$ un ordre lexicographique qui respecte \mathcal{R}_1 :

$(\forall i, j \ x_i <_{\mathcal{R}_2} z_j, \text{ et } \forall i, j \ x_i <_{\mathcal{R}_1} x_j \Rightarrow x_i <_{\mathcal{R}_2} x_j)$

$B :=$ **Gröbner** $(F, \mathcal{R}_2) \cap K[x_i]$

— *Calcul de l’ensemble caractéristique C de $[\Sigma]$ pour l’ordre \mathcal{R}_1 .*

$C := \emptyset$

pour tout $b \in B$ (par ordre croissant) **faire**

 soit u l’indéterminée principale de b

 soient v_1, \dots, v_s les indéterminées principales des éléments de C

```

    si  $u \neq v_s$  et si  $u$  n'est la dérivée propre d'aucun  $v_i$  alors
       $C := C \cup \{b \text{ rem } C\}$ 
    fin
  fin
produire ( $C$ )
fin

```

6.2 Les logiciels

Les logiciels que nous avons développés ont été écrits en langage C, bien que d'une façon détournée, comme nous le verrons. Ils sont implantés sur stations SUN-4 et RS-6000. Pourquoi le langage C? Le but premier de ma thèse était l'implantation efficace de l'algorithme d'élimination de SEIDENBERG en algèbre différentielle ordinaire, dont DIOP avait peu auparavant établi l'utilité en automatique; or FAUGERE a montré que l'implantation directe en langage C d'un logiciel de bases de GRÖBNER donnait un outil cent fois plus rapide que son homologue rédigé sous AXIOM. Plus tard, la complexité rédhibitoire des algorithmes d'élimination nous est apparue clairement, nous avons mis en veilleuse nos projets d'implantation sophistiquée et nous nous sommes tournés vers des domaines d'investigation plus théoriques.

Aujourd'hui, les programmes *rosen1* et *rosen2* qui sont l'implantation des deux versions de l'algorithme ROSENFELD-GRÖBNER décrites dans la section précédente, sont obtenus par compilation des unités données dans le tableau donné page suivante. Un exécutable occupe sur le disque dur un volume d'à-peu près un million d'octets: les deux-tiers sont consommés par PARI.

Pour obtenir par exemple, un exécutable à partir de *rosen1*, il suffit de procéder à l'édition des liens des "fichiers objets" correspondant aux unités de compilation situées sous *rosen1: rosen1.o, sppolsys.o, ..., clker.o*.

Ces unités sont hiérarchisées. Les couches les plus basses sont des couches de base. Les couches supérieures sont dédiées à l'algèbre différentielle.

clker est le gestionnaire de la mémoire, chargé des "ramasse-miettes" (en Anglais *garbage collector*). Décrivons-en rapidement le fonctionnement.

La mémoire est partagée en plusieurs zones. Chacune est dédiée à un type d'objet particulier. Actuellement, le système en gère quatre: une zone pour les doublets (ou paires pointées), une pour les grands nombres, une pour les symboles et une pour certains pointeurs particuliers, dont les pointeurs de fonctions.

Les doublets sont gérés en liste libre: *clker* dispose d'une liste des doublets disponibles et alloue des emplacements à la demande des fonctions des couches supérieures. Lors des saturations (lorsque la liste est vide), le système déclenche un ramasse-miettes de type Mark & Sweep (voir [Kn]) et construit une nouvelle liste libre. *clker* provoque l'erreur *saturation de la zone des doublets* si la masse des

doublets récupérés est inférieure à quinze pour cents de la masse totale. L'appel au ramasse-miettes est invisible pour les autres couches.

GB	rosen1 rosen2 calcul	seiden pseiden
	sppolsys	
	lppolsys	
	ppolsys1 ppolsys2	
	alphabet	
PARI	usuel operateurs box	
	clsys system	
	clker	

Les trois autres zones sont gérées de façon homogène, ce qui rend le nombre de zones aisément extensible. *clker* ne sait rien (en théorie) de leur contenu, ni de la manière dont leur valeur est manipulée par les couches supérieures. Elles sont nettoyées lorsqu'elles sont saturées ; leur contenu est alors tassé. Les objets inutiles sont détectés de la manière suivante :

Un élément d'une de ces zones contient toujours au moins deux informations : la longueur de la zone, et un pointeur vers un doublet appelé *ancree* de l'élément dans la zone des doublets. Lors de sa création, chaque élément reçoit une ancre en direction de laquelle il dispose d'un pointeur et dont les deux champs (*car* et *cdr* pour ne pas les nommer) pointent en retour vers l'élément. Les fonctions des couches supérieures manipulent toujours l'élément via son ancre. On détecte ainsi très facilement les éléments qui ne sont plus utilisés : ce sont ceux dont l'ancre fait partie de la liste libre des doublets, c'est-à-dire ceux dont l'ancre comporte un pointeur *cdr* qui pointe ailleurs que sur eux.

Cette technique astucieuse permet un ramasse-miettes asynchrone avec le Mark & Sweep des doublets. Elle m'a été suggérée par la lecture d'une documentation

LE_LISP (voir [De]). La documentation ne dit pas à quoi l'ancre sert, mais je ne vois pas quel autre usage elle pourrait avoir.

clsys, system contiennent le code des primitives de bas niveau : routines élémentaires d'entrée-sortie, arithmétiques ... sans lesquelles un exécutable ne pourrait pas fonctionner.

C'est *clsys* qui gère les entiers : les entiers de valeur absolue inférieure à 32768 sont codés comme des entiers machine (une autre technique de LE_LISP). Les grands entiers sont sous-traités à la bibliothèque PARI. Un entier de valeur absolue supérieure à 32768 consomme au minimum seize octets dans la zone des entiers (un pointeur vers l'ancre, douze octets pour la gestion interne de PARI).

La taille des différentes zones est habituellement fournie en paramètre au compilateur C ; elle varie d'une exécution à l'autre. Pour traiter les exemples de ce chapitre, nous avons compilé les programmes *rosen1* et *rosen2* avec un peu plus d'un million cinq cent mille (1500000) paires pointées et de deux cent cinquante mille (250000) octets pour les entiers. Les autres zones sont négligeables.

usuel contient de nombreuses fonctions usuelles : opérations classiques sur les listes, routines standard d'entrée-sortie *read, print* etc ...

opérateurs contient le code d'une fonction très classique qui reçoit en entrée une table d'opérateurs et qui lit sur l'entrée standard un flot de caractères. Le flot est analysé et transformé en arbre n-aire en fonction de la table. Toutes les entrées évoluées (lecture de polynômes, de systèmes d'équations et d'inéquations, etc ...) sont programmées très simplement grâce à cette fonction.

box contient le code d'un paragrapheur de sorties rudimentaire. Le paragrapheur intercepte les sorties à l'insu des routines qui les produisent, ce qui le rend d'un emploi agréable.

alphabet implante la gestion des ordres admissibles sur ΘX . Un ordre admissible est donné sous la forme d'une liste de couples (*groupe · fonction*), précédée par le nombre des dérivations. Nous notons la liste à la LISP :

$$(\text{nombre de dérivations } (g_1 \cdot f_1) (g_2 \cdot f_2) \cdots (g_n \cdot f_n)).$$

Un *groupe* est soit une lettre (ordre d'élimination), soit une liste de lettres (ordre alterné entre les membres du groupe). La *fonction* définit un ordre admissible sur les opérateurs de dérivations. Elle est paramétrée par deux opérateurs de dérivations et retourne *vrai* si le premier est inférieur au second :

une indéterminée θx sera dite inférieure à une indéterminée ϕy suivant l'alphabet A si le groupe dont x fait partie apparaît dans A après le groupe dont y fait partie, ou si les deux groupes sont identiques et $f(\theta, \phi)$ vaut *vrai*, où f est la fonction associée au groupe.

Des primitives d'entrée-sortie permettent à l'utilisateur de manipuler agréablement les alphabets.

ppolsys1, **ppolsys2** implantent les polynômes différentiels à coefficients dans \mathbb{Z} .

Les polynômes qui appartiennent à \mathbb{Z} sont représentés par de simples entiers, les autres sous forme récursive. Un polynôme p de $\mathbb{Z}\{X\}$, d'indéterminée principale θx est représenté sous forme creuse par une liste :

$$((x \theta) (\text{degré}_n \cdot \text{coeff}_n) \cdots (\text{degré}_1 \cdot \text{coeff}_1)).$$

Les couples $(\text{degré}_i \cdot \text{coeff}_i)$ sont rangés par degré décroissant avec i . Les coeff_i sont des polynômes différentiels non nuls, d'indéterminée principale inférieure à θx . Cette représentation permet d'accéder rapidement à l'indéterminée principale d'un polynôme, à son degré et à son initial.

L'implantation est répartie sur deux unités parce que, lorsqu'elles sont réunies, elles saturent la mémoire des optimiseurs du compilateur C.

lppolsys implante les opérations sur les listes de polynômes différentiels : extraction de tous les ensembles caractéristiques d'une liste, réduction par un ensemble auto-réduit suivant la méthode décrite en fin de premier chapitre, calcul des Δ -polynômes d'un ensemble auto-réduit.

Les fonctions qui y sont implantées fournissent des informations complémentaires aux fonctions appelantes des couches supérieures : lors d'une réduction, quels sont exactement les polynômes qui ont servi aux réductions, etc ...

sppolsys implante les systèmes d'équations et d'inéquations. Un système d'équations et d'inéquations, c'est plus qu'une liste d'équations et une liste d'inéquations. C'est (aujourd'hui) un quadruplet de listes :

$$(\text{équations}, \text{inéquations}, \text{équations sûres}, \text{informations})$$

Les *équations sûres* sont celles dont l'initial et le séparant figurent parmi les inéquations (éventuellement sous formes réduites, et donc difficilement reconnaissables). Il s'agit d'une information utile puisqu'elle permet de choisir un "bon" ensemble caractéristique de la liste des équations, parmi ceux fournis par *lppolsys*. Le champ *informations* contient de nombreuses informations, qui y sont rangées par les diverses routines de l'application : où sommes-nous dans l'arbre des scindages ? d'où vient tel polynôme ? fait-il partie de l'ensemble caractéristique des équations ? etc ...

sppolsys contient également des routines élaborées d'entrée-sortie qui exploitent tout ou partie du champ *information*.

Disposer de ces informations permet d'analyser le comportement des réécritures. En effet, quoi de plus illisible qu'un volumineux arbre de systèmes d'équations et d'inéquations livré sous forme brute. Nous donnerons un exemple de trace en annexe A.3 (page 110).

rosen1, **rosen2** contiennent chacune un programme principal (fonction *main* en C) et correspondent aux deux versions de ROSENFELD-GRÖBNER décrites au chapitre précédent.

Les calculs de bases de GRÖBNER se font sous le logiciel *GB*. En attendant de disposer d'une librairie *GB* que nous pourrions relier à nos exécutables, nous utilisons un petit programme LEX¹ qui convertit les sorties de nos algorithmes en scripts *GB*, qui sont ensuite évalués par le logiciel de FAUGERE. Nous donnerons un exemple de tels scripts en annexe A.1 (page 99).

calcul est une calculatrice symbolique qui utilise pleinement les ressources de l'unité *opérateurs*. Nous donnons un exemple d'utilisation de la calculatrice en annexe A.2 (page 107).

seiden, *pseiden* contiennent chacune un programme principal et correspondent à deux versions de l'algorithme d'élimination de SEIDENBERG en algèbre différentielle ordinaire : *seiden* implante les réécritures que nous avons décrites au chapitre III tandis que *pseiden* implante les réécritures données dans l'article [Se1] de 1956.

seiden et *pseiden* sont des réalisations anciennes qui n'utilisent pas pleinement le découpage actuel des unités de compilation. C'est la raison pour laquelle elles n'accèdent pas aux unités *lppolsys* et *sppolsys*.

Les unités décrites précédemment n'ont pas été rédigées directement en C, mais par l'intermédiaire d'un petit langage nommé CL. CL ressemble à première vue à un mini-LISP (même parenthésage, primitives *cons*, *car*, *csr*, *map*, lambda-expressions etc ...) mais correspond davantage à un préprocesseur du langage C.

Le système de portée des fonctions et des variables et le découpage d'une application en unités compilées séparément est exactement le même en CL qu'en C : une variable ou une fonction CL locale à une unité de compilation est traduite en une variable ou une fonction *static* du langage C, etc ...

Le développement en CL d'une application complète, comme celles décrites dans la section ci-dessus, est bien moins pénible qu'en C. Les exécutables ainsi obtenus sont bien sûr moins rapides que leurs homologues écrits directement dans ce langage mais, comme le découpage des unités de compilation est exactement celui du langage C, il est tout-à-fait possible au programmeur de réécrire finement certaines unités stratégiques (dans notre cas *alphabet*, *ppolsys1* et *ppolsys2*) après que toute l'application a atteint une maturité suffisante et que les interfaces entre unités ont acquis une certaine stabilité.

Sur ce point, qu'en est-il de nos implantations ? Les algorithmes que nous avons décrits et dont nous donnerons des applications ultérieurement sont encore expérimentaux, les interfaces entre unités ne sont pas encore nettement dessinées : ce serait donc une perte de temps que d'optimiser aujourd'hui les modules *alphabet*, *ppolsys1* ou *ppolsys2*. Certains choix sont d'ailleurs particulièrement inefficaces en temps d'exécution : la comparaison de deux indéterminées, qui est une opération très couramment utilisée lors des opérations sur les polynômes, se fait en parcourant la structure *alphabet*. Nous conservons cette méthode à l'heure actuelle parce qu'elle est très souple : elle permet de faire évoluer l'ordre admissible au cours des calculs. A ce propos, j'ajoute que si nous le pouvions, nous rendrions la structure encore plus malléable : nous ne spécifierions

1. LEX est un utilitaire UNIX standard d'analyse lexicale.

que la partie de l'ordre admissible utile pour déduire des informations de structure sur un système dynamique par exemple, et nous laisserions les algorithmes choisir le reste au mieux, afin d'éviter les explosions combinatoires.

CL est une idée intéressante, probablement prématurée mais le comportement explosif des algorithmes ne nous est réellement apparue que sur le tard. Il s'agit d'un logiciel encore expérimental, non évalué vis-à-vis de l'existant mais agréable d'emploi. Son principal défaut? Le développement d'algorithmes en dehors des logiciels de calcul formel classiques rend l'usage de bibliothèques standard (de factorisation de polynômes par exemple) difficile.

6.3 Une application à l'automatique non linéaire

Nous ne nous intéresserons ici qu'au problème de *l'observabilité*, sachant que d'autres applications de nos méthodes à l'automatique non linéaire existent (calcul du rang de sortie, inversion entrée-sortie, etc ...). Nous illustrons le problème sur un exemple.

Considérons un mobile en déplacement rectiligne uniforme sur un plan muni d'un repère cartésien (l'exemple est tiré de [FG]). Un radar situé sur l'origine du repère mesure la distance r du mobile à l'origine (ou plutôt son carré):

$$\Sigma \quad \left\{ \begin{array}{l} V_x = \dot{X} \\ V_y = \dot{Y} \\ \dot{V}_x = 0 \\ \dot{V}_y = 0 \\ R = X^2 + Y^2. \end{array} \right.$$

La question que nous nous posons au sujet de ce système se formule intuitivement ainsi: *connaissant la distance r , pouvons nous déduire les deux composantes (v_x, v_y) de la vitesse du mobile?*

La réponse est non, et s'obtient aisément par de simples considérations géométriques, mais l'exemple est tout de même intéressant.

Les équations du système engendrent un idéal différentiel premier de l'anneau de polynômes différentiel ordinaire $K\{R, X, Y, V_x, V_y\}$, puisqu'il existe un ordre admissible sur $\Theta\{R, X, Y, V_x, V_y\}$ pour lequel Σ soit orthonomique et auto-réduit (voir [Ro], page 399). Notons r, x, y, v_x et v_y l'image des lettres R, X, Y, V_x, V_y dans le corps des fractions L de l'anneau $K\{R, X, Y, V_x, V_y\}/[\Sigma]$ par le morphisme canonique. En d'autres termes, posons:

$$L = K\langle r, x, y, v_x, v_y \rangle.$$

Si on admet que les grandeurs physiques décrites par le système sont bien représentées par les "quantités formelles" ci-dessus alors, d'après les théories de FLIESS (voir [F]), le problème de l'observabilité de v_x (respectivement v_y) relativement à r se reformule en:

La grandeur v_x est localement observable relativement à r si v_x est algébrique (mais

non différentielle) sur le corps $K\langle r \rangle$. La grandeur v_x est globalement observable relativement à r si v_x appartient au corps $K\langle r \rangle$.

En pratique, on ne dispose pas du corps $K\langle r \rangle$ mais d'un système de générateurs et de relations qui le définissent. Ainsi, décider si v_x est localement observable relativement à r , c'est décider si l'idéal différentiel premier $[\Sigma]$ contient une relation du type suivant :

$$(\mathcal{T}) \quad P = \sum_{i=1}^d C_i \cdot V_x^i, \quad \text{avec} \quad (P \in [\Sigma] \cap K\{R\}[V_x] \quad \text{et} \quad I_P = C_d \notin [\Sigma]).$$

Décider si v_x est globalement observable relativement à r , c'est décider si l'idéal $[\Sigma]$ contient une relation du type (\mathcal{T}) , de degré en V_x égal à 1.

Si nous reprenons les notations introduites dans la section 5.3.4, il s'agit de trouver (s'il existe) le terme principal minimal de $[\Sigma]$, dont l'indéterminée principale est V_x pour tout ordre du type :

$$(\mathcal{O}) \quad \text{autres indéterminées} \dots > \ddot{V}_x > \dot{V}_x > V_x > \dots > \ddot{R} > \dot{R} > R$$

Le problème est résolu par la donnée d'un ensemble caractéristique de l'idéal pour l'ordre (\mathcal{O}) , ainsi que le montrent [DF], [Ol3], [FG] et le lemme 24 (page 77).

Il l'est également par le calcul de la base de GRÖBNER principale B associée à Σ pour le même ordre. D'après la proposition 6 (page 78) en effet, v_x est localement observable relativement à r si et seulement si la base de GRÖBNER principale B calculée par l'algorithme ROSENFELD–GRÖBNER, pour un ordre admissible du type (\mathcal{O}) comporte un polynôme de type (\mathcal{T}) . La grandeur v_x est globalement observable relativement à r si et seulement si B comporte un polynôme de type (\mathcal{T}) de degré 1 en V_x .

Voici la base B , calculée par *rosenfeld1* et *GB* pour l'ordre admissible :

$$\dots > \dot{Y} > Y > \dots > \dot{X} > X > \dots > \dot{V}_y > V_y > \dots > \dot{V}_x > V_x > \dots > \dot{R} > R$$

$$B \quad \left\{ \begin{array}{l} R^{(3)} \\ \dot{V}_x \\ 2V_y^2 + 2V_x^2 - \ddot{R} \\ 2X^2\ddot{R} - 4XV_x\dot{R} + 4V_x^2R - 2\ddot{R}R + \dot{R}^2 \\ 2YV_x^2 - Y\ddot{R} - 2XV_yV_x + V_y\dot{R} \\ 2YV_y + 2XV_x - \dot{R} \\ YX\ddot{R} - YV_x\dot{R} - XV_y\dot{R} + 2V_yV_xR \\ 2YXV_x - Y\dot{R} - 2X^2V_y + 2V_yR \\ Y^2 + X^2 - R \end{array} \right.$$

On constate que v_x n'est pas observable relativement à r .

Mesures Une trace des calculs est donnée en annexe A.1 (page 99). Les calculs ont été effectués sur une station de travail RS6000. Les temps de calculs sont fournis par `/bin/time`. Le premier est le temps *réel*, c'est-à-dire celui passé par l'utilisateur entre

le moment où la commande est entrée et celui où les résultats s'affichent à l'écran. Le second est le temps *CPU* effectivement consommé par le programme².

nombre de systèmes	réguliers	contradictaires	total
<i>rosenfeld1</i>	11	34	95
bases de GRÖBNER	non contradictoires	contradictaires	
<i>GB</i>	11	0	
temps de calcul	temps réel (sec.)	temps CPU (sec.)	
<i>rosenfeld1</i>	1.7	1.5	
<i>GB</i>	3.3	0.7	

Remarque Nous avons fourni aux programmes un ordre admissible sur toutes les indéterminées or n'importe quel ordre de type (\mathcal{O}) nous satisfaisait. Pour autant, tous ces ordres ne produisent pas des comportements équivalents. Par exemple, l'ordre suivant

$$\dots > \dot{V}_y > V_y > \dots > \dot{X} > X > \dots > \dot{Y} > Y > \dots > \dot{V}_x > V_x > \dots > \dot{R} > R$$

provoque une explosion combinatoire.

De tels phénomènes montrent qu'il serait très agréable de disposer de logiciels capables de faire évoluer l'ordre admissible au cours des calculs. Cela ralentirait bien sûr considérablement les manipulations de polynômes mais rendrait les logiciels plus faciles à utiliser pour le commun des mortels : la question *v_x est-elle observable relativement à r ?* est nettement plus intuitive que *la base calculée pour tel ou tel ordre contient-elle un polynôme de tel ou tel type ?*

6.4 Calcul des conditions de compatibilité

Considérons le système suivant aux dérivées partielles, dû à JANET, traité dans [P], [M1] et [M2] :

$$\Sigma \begin{cases} u_{zz} - yu_{xx} = 0 \\ u_{yy} = 0. \end{cases}$$

Les dérivations sont notées suivant une notation de jets : u_{yyz} se lit $\frac{\partial^3 u}{\partial y^2 \partial z}$.

Nous souhaitons déterminer les conditions nécessaires (non suffisantes) que doivent vérifier les deux équations (appelons-les v et w) pour que Σ admette des modèles différentiels. Pour traiter ce problème avec notre logiciel, il suffit de réécrire Σ en :

$$\Sigma' \begin{cases} v = u_{zz} - yu_{xx} \\ w = u_{yy} \\ y_x = 0 \\ y_y = 1 \\ y_z = 0 \end{cases}$$

2. pour les initiés, il s'agit de la somme des temps *user* et *system*.

et de calculer une représentation des modèles différentiels pour l'ordre suivant (qui illustre au passage la souplesse de notre gestion des ordres admissibles) :

- $\dots \theta u > u > \dots > \theta v > \theta w > v > w > \dots > \theta y > y$
- $\theta u > \phi u$ si $\theta > \phi$ pour l'ordre lexicographique donné par $\delta_x > \delta_y > \delta_z$ (idem pour la lettre y)
- $\theta v > \phi v$ si l'ordre de θ est supérieur à l'ordre de ϕ (idem pour la lettre w)
- $\theta v > \phi v$ si les deux opérateurs ont même ordre et si $\theta > \phi$ pour l'ordre lexicographique donné par $\delta_x > \delta_y > \delta_z$ (idem pour la lettre w)

L'algorithme ROSENFELD-GRÖBNER ne produit qu'un seul système régulier, qui est nécessairement principal :

$$\Omega \left\{ \begin{array}{l} y_z = 0 \\ y_y - 1 = 0 \\ y_x = 0 \\ yw_{xxy} + v_{yyy} - w_{yzz} + 3w_{xx} = 0 \\ y^3w_{xxxxx} + y^2v_{xxxxxy} - 3y^2w_{xxxxzz} - 2yv_{xxyzz} + 3yw_{xxxxzz} + v_{yyzzzz} \\ \quad - w_{zzzzzz} - 2yv_{xxxxxy} + 2v_{xxyzz} + 2v_{xxxx} = 0 \\ 2u_{zzzz} - y^4w_{xxxx} - y^3v_{xxyy} + 2y^3w_{xxxz} + y^2v_{yyzz} - y^2w_{zzzz} + 2y^2v_{xxy} \\ \quad - 2yv_{xx} - 2v_{zz} = 0 \\ 2yu_{yzz} - 2u_{zz} + y^3w_{xx} + y^2v_{yy} - y^2w_{zz} - 2yv_y + 2v = 0 \\ u_{yy} - w = 0 \\ yu_{xx} - u_{zz} + v = 0 \\ y \neq 0 \end{array} \right.$$

Nous ne donnons pas la base de GRÖBNER, qui est un peu volumineuse (15 polynômes). On constate que les deux équations doivent satisfaire une condition de compatibilité d'ordre 3 et une autre, d'ordre 6.

Mesures Les calculs ont été effectués sur une station de travail RS6000. Les temps de calculs sont fournis par `/bin/time`. Le premier est le temps *réel*, c'est-à-dire celui passé par l'utilisateur entre le moment où la commande est entrée et celui où les résultats s'affichent à l'écran. Le second est le temps *CPU* effectivement consommé par le programme.

nombre de systèmes	réguliers	contradictaires	total
<i>rosenfeld1</i>	1	4	10
bases de GRÖBNER	non contradictaires	contradictaires	
<i>GB</i>	1	0	
temps de calcul	temps réel (sec.)	temps CPU (sec.)	
<i>rosenfeld1</i>	5.2	4.1	
<i>GB</i>	2.3	0.8	

Remarque Dans [M2] (page 26), MANSFIELD indique que son algorithme résout le problème en une minute environ, en utilisant MAPLE V.

6.5 Décision du vide

L'algorithme ROSENFELD–GRÖBNER décide si un polynôme appartient au radical d'un idéal différentiel de type fini (proposition 3 (page 74)) en n'utilisant que l'addition, la multiplication et le test d'égalité à zéro dans le corps de base des équations. Dans ce domaine, les seuls concurrents que nous lui connaissons sont les algorithmes d'élimination de SEIDENBERG et leurs optimisations (voir [G]).

Nous n'avons pas la preuve que l'algorithme de ROSENFELD–GRÖBNER est plus efficace que ceux de SEIDENBERG, bien que nous pensions que ce soit le cas. L'exemple suivant illustre cependant assez bien notre principal argument sur ce sujet.

Le système aux dérivées partielles suivant, que nous empruntons à [P], décrit les équations d'EULER d'un fluide incompressible en deux dimensions :

$$\Sigma \begin{cases} v_t^1 + v^1 v_1^1 + v^2 v_2^1 + p_1 \\ v_t^2 + v^1 v_1^2 + v^2 v_2^2 + p_2 \\ v_1^1 + v_2^2. \end{cases}$$

La pression est représentée par la lettre p . Les lettres v^1 et v^2 désignent les deux composantes de la vitesse du fluide. Les trois dérivations (une par rapport au temps et une pour chaque dimension) sont notées suivant une notation de jets : v_{22t}^1 se lit $\delta_2 \delta_2 \delta_t v^1$.

L'application de l'algorithme ROSENFELD–GRÖBNER sur le système Σ pour l'ordre admissible alterné \mathcal{R} suivant : quelles que soient les lettres x et y de l'alphabet $\{p, v^1, v^2\}$, on a $\theta x > \phi y$ si

1. l'ordre de l'opérateur θ est supérieur à celui de ϕ ou si,
2. les deux opérateurs ont même ordre et θ est supérieur à ϕ pour l'ordre lexicographique donné par $\delta_t > \delta_1 > \delta_2$ ou enfin si,
3. les deux opérateurs sont identiques et x est supérieur à y pour l'ordre $p > v^1 > v^2$,

engendre presque instantanément un unique système régulier :

$$\Omega \begin{cases} v_1^1 + v_2^2 \\ v_t^2 + v^1 v_1^2 + p_2 + v^2 v_2^2 \\ v_t^1 + p_1 + v^2 v_2^1 - v^1 v_2^2 \\ p_{11} + p_{22} + 2v_2^1 v_1^2 + 2(v_2^2)^2. \end{cases}$$

Les idéaux différentiels $[\Omega]$ et $[\Sigma]$ sont égaux. Comme Ω est orthonomique ($H_A = 1$), l'idéal différentiel engendré par les équations d'EULER est premier et les équations de Ω constituent un ensemble caractéristique de cet idéal pour l'ordre \mathcal{R} (voir [Ro], remarque 3, page 399 ou [Ko], IV, §9, lemme 2, page 167).

Comme chacune des lettres de l'alphabet admet au moins une dérivée parmi les indéterminées principales des éléments d'un ensemble caractéristique de l'idéal pour un ordre admissible qui vérifie : $\theta x > \phi y$ si $\text{ord } \theta > \text{ord } \phi$, le degré de transcendance différentiel du système est nul (voir [Ko], II, §12, theorem 6, page 115); en d'autres

termes, tout élément du corps des fractions de $\mathbb{Q}\{p, v^1, v^2\}/[\Sigma]$ vérifie au moins une relation polynomiale différentielle à coefficients dans \mathbb{Q} (voir sur ce sujet [Ko], II, §10, page 108 ou [Ol1], I, page 18, pour une entrée en matière moins abrupte).

Par conséquent, l'idéal $[\Sigma]$ contient des polynômes qui ne lient que v^1 et certaines de ses dérivées (même chose pour v^2 et p) et nous avons en effet obtenu à la calculatrice symbolique (nous donnons une trace des calculs en annexe A.2 (page 107)) une relation d'ordre 5 en v^1 , conséquence des relations initiales, en nous inspirant de la méthode suggérée par POMMARET :

$$\begin{aligned} & ((v^1 v_2^1 v_{1122}^1 + v^1 v_2^1 v_{2222}^1 - v^1 v_{22}^1 v_{112}^1 - v^1 v_{22}^1 v_{222}^1) v_{11122}^1 + (-v^1 v_2^1 v_{1112}^1 - v_2^1 v_{112t}^1 - v^1 v_2^1 v_{1222}^1 - \\ & v_2^1 v_{222t}^1 + (v^1 v_{22}^1 - 2(v_2^1)^2) v_{111}^1 + 2v_2^1 v_1^1 v_{112}^1 + v_{22}^1 v_{11t}^1 + (v^1 v_{22}^1 - 2(v_2^1)^2) v_{122}^1 + 2v_2^1 v_1^1 v_{222}^1 + \\ & v_{22}^1 v_{22t}^1 + (v_2^1 v_{12}^1 - v_1^1 v_{22}^1) v_{11}^1 + v_2^1 v_{22}^1 v_{12}^1 - v_1^1 (v_{22}^1)^2 v_{11222}^1 + (v_2^1 v_{1122}^1 + v_2^1 v_{2222}^1 - v_{22}^1 v_{112}^1 - \\ & v_{22}^1 v_{222}^1) v_{1122t}^1 + (v^1 v_2^1 v_{1122}^1 + v^1 v_2^1 v_{2222}^1 - v^1 v_{22}^1 v_{112}^1 - v^1 v_{22}^1 v_{222}^1) v_{12222}^1 + (-v^1 v_2^1 v_{1112}^1 - \\ & v_2^1 v_{112t}^1 - v^1 v_2^1 v_{1222}^1 - v_2^1 v_{222t}^1 + (v^1 v_{22}^1 - 2(v_2^1)^2) v_{111}^1 + 2v_2^1 v_1^1 v_{112}^1 + v_{22}^1 v_{11t}^1 + (v^1 v_{22}^1 - \\ & 2(v_2^1)^2) v_{122}^1 + 2v_2^1 v_1^1 v_{222}^1 + v_{22}^1 v_{22t}^1 + (v_2^1 v_{12}^1 - v_1^1 v_{22}^1) v_{11}^1 + v_2^1 v_{22}^1 v_{12}^1 - v_1^1 (v_{22}^1)^2 v_{22222}^1 + (v_2^1 v_{1122}^1 + \\ & v_2^1 v_{2222}^1 - v_{22}^1 v_{112}^1 - v_{22}^1 v_{222}^1) v_{2222t}^1 + (3(v_2^1)^2 v_{1122}^1 + 3(v_2^1)^2 v_{2222}^1 + (v^1 v_{22}^1 - 3v_2^1 v_{22}^1) v_{112}^1 + \\ & v^1 (v_{22}^1)^2 - 3v_2^1 v_{22}^1 v_{222}^1) v_{1112}^1 - 3v_2^1 v_1^1 (v_{1122}^1)^2 + (3(v_2^1)^2 v_{1222}^1 - 6v_2^1 v_1^1 v_{2222}^1 + (-v^1 v_{22}^1 + \\ & 3v_2^1 v_{22}^1) v_{111}^1 + (-3v_2^1 v_{12}^1 + 3v_1^1 v_{22}^1) v_{112}^1 - v_{22}^1 v_{11t}^1 + (-v^1 v_{22}^1 - v_2^1 v_{11}^1 + 2v_2^1 v_{22}^1) v_{122}^1 + (-v_{22t}^1 + \\ & v_1^1 v_{11}^1 - 3v_2^1 v_{12}^1 + 4v_1^1 v_{22}^1) v_{222}^1) v_{1122}^1 + (v_{22}^1 v_{112}^1 + (v_{22}^1)^2) v_{112t}^1 + (3(v_2^1)^2 v_{2222}^1 + (v^1 v_{22}^1 - \\ & 3v_2^1 v_{22}^1) v_{112}^1 + v^1 (v_{22}^1)^2 - 3v_2^1 v_{22}^1 v_{222}^1) v_{1222}^1 - 3v_2^1 v_1^1 (v_{2222}^1)^2 + ((-v^1 v_{22}^1 + 3v_2^1 v_{22}^1) v_{111}^1 + \\ & (-3v_2^1 v_{12}^1 + 3v_1^1 v_{22}^1) v_{112}^1 - v_{22}^1 v_{11t}^1 + (-v^1 v_{22}^1 - v_2^1 v_{11}^1 + 2v_2^1 v_{22}^1) v_{122}^1 + (-v_{22t}^1 + v_1^1 v_{11}^1 - \\ & 3v_2^1 v_{12}^1 + 4v_1^1 v_{22}^1) v_{222}^1) v_{2222}^1 + (v_{22}^1 v_{112}^1 + (v_{22}^1)^2) v_{222t}^1 + ((2v_2^1 v_{22}^1 - 3(v_2^1)^2) v_{112}^1 + 2v_2^1 (v_{22}^1)^2 - \\ & 3(v_2^1)^2 v_{222}^1) v_{111}^1 + (-2v_1^1 v_{22}^1 + 3v_2^1 v_{12}^1) (v_{112}^1)^2 + ((2v_2^1 v_{22}^1 + v_{22}^1 v_{11}^1 - 2(v_2^1)^2) v_{122}^1 - 4v_1^1 (v_{22}^1)^2 + \\ & (-v_{12}^1 v_{11}^1 + 5v_{22}^1 v_{12}^1) v_{222}^1) v_{112}^1 + (2v_2^1 (v_{22}^1)^2 + (v_{22}^1 v_{11}^1 - 2(v_2^1)^2) v_{222}^1) v_{122}^1 - 2v_1^1 (v_{22}^1)^3 + \\ & (-v_{12}^1 v_{11}^1 + 2v_{22}^1 v_{12}^1) (v_{22}^1)^2) \end{aligned}$$

Par des calculs similaires, il est facile d'obtenir une relation en v^2 par contre, nous ne sommes pas parvenus à calculer la relation qui ne lie que la pression et certaines de ses dérivées : les polynômes que nous avons calculés saturent la mémoire d'une station de travail.

En théorie, l'algorithme ROSENFELD–GRÖBNER résout le problème : il suffit de calculer la base de GRÖBNER principale associée à Σ pour un ordre d'élimination du type : $\theta p < \Theta\{v^1, v^2\}$. Toutefois, s'il est si difficile de calculer *une* relation en p à la calculatrice, il est clair qu'un système de réécriture brutal qui tente de donner une description de l'idéal $[\Sigma] \cap K\{p\}$ ne peut qu'échouer.

Les algorithmes d'élimination de SEIDENBERG sont restreints à des ordres d'élimination sur tout l'alphabet, quelle que soit l'optimisation qu'on leur apporte : ils sont donc incapables de décider de l'appartenance au radical d'idéaux différentiels aussi complexes que celui-ci.

L'algorithme ROSENFELD–GRÖBNER vérifie par contre très facilement que la relation $P(v^1)$ ci-dessus est conséquence des équations de départ : il suffit d'ajouter aux équations d'EULER l'inéquation $P(v^1) \neq 0$. L'algorithme détecte la contradiction en un peu moins de deux secondes, pour l'ordre \mathcal{R} (source d'information : `/bin/time`).

Cet exemple n'est pas un cas isolé. Il ne nous est quasiment jamais arrivé de traiter

un exemple pour lequel l'ordre le plus efficace soit un ordre d'élimination. On doit pouvoir rapprocher ce comportement du phénomène constaté dans les calculs de bases de GRÖBNER (voir [FGLM]) où les calculs de bases suivant le degré total sont bien plus rapides que suivant un ordre lexicographique. Similairement (malheureusement), les descriptions des modèles différentiels obtenues suivant un ordre alterné sont moins riches en informations que celles obtenues suivant un ordre d'élimination.

6.6 Contradictions algébriques cachées

Dans l'exemple suivant, nous avons "trafiqué" un système d'équations au dérivées partielles afin de masquer le fait que $v = 0$ est une conséquence des équations de Σ . Les lettres sont u et v , les dérivations sont δ_x , δ_y et δ_z .

$$\Sigma \begin{cases} u_y^2 u_x^2 = 2u_y u_x - 1 \\ u_{xy} = v \\ v_x = u_x v_z \\ v_y = u_y v_z \\ u_z^3 = u_x u_y. \end{cases}$$

L'algorithme *rosenfeld1* produit deux systèmes réguliers pour l'ordre d'élimination

- $\theta u > \phi v$ pour tous opérateurs de dérivation θ et ϕ .
- $\theta u > \phi u$ si $\theta > \phi$ pour l'ordre lexicographique donné par $\delta_x > \delta_y > \delta_z$

$$\Omega_1 \begin{cases} v_z v_{yz} - v_{zz} v_y = 0 \\ (v_z^2 v_y v_{yy} + (-v_{zz} + v v_z) v_y^3) v_x - v_z^4 v_{yy} + (v_z^2 v_{zz} - v v_z^3) v_y^2 = 0 \\ u_z^3 - 1 = 0 \\ v_z u_y - v_y = 0 \\ (v_z^2 v_y v_{yy} + (-v_{zz} + v v_z) v_y^3) u_x - v_z^3 v_{yy} + (v_z v_{zz} - v v_z^2) v_y^2 = 0 \\ v_z \neq 0 \\ v_z v_y v_{yy} - v_y^2 v_{yz} + v v_y^3 \neq 0 \\ v_y \neq 0 \\ v_y^2 v_x - v_z^2 v_y \neq 0 \\ u_z \neq 0 \\ v_z^2 v_y v_{yy} + (-v_{zz} + v v_z) v_y^3 \neq 0 \end{cases} \quad \Omega_2 \begin{cases} v = 0 \\ u_z^3 - 1 = 0 \\ u_{yy} = 0 \\ u_y u_x - 1 = 0 \\ u_z \neq 0 \\ u_y \neq 0 \end{cases}$$

GB détecte une contradiction algébrique cachée dans Ω_1 : le système principal est le second système. Voici la base de GRÖBNER qui lui est associée :

$$B \begin{cases} v \\ u_z - 1 \\ u_{yy} \\ u_x u_y - 1. \end{cases}$$

Mesures Les calculs ont été effectués sur une station de travail RS6000. Les temps de calculs sont fournis par `/bin/time`. Le premier est le temps *réel*, c'est-à-dire celui passé par l'utilisateur entre le moment où la commande est entrée et celui où les résultats s'affichent à l'écran. Le second est le temps *CPU* effectivement consommé par le programme.

nombre de systèmes	réguliers	contradictaires	total
<i>rosenfeld1</i>	2	26	55
bases de GRÖBNER	non contradictaires	contradictaires	
<i>GB</i>	1	1	
temps de calcul	temps réel (sec.)	temps CPU (sec.)	
<i>rosenfeld1</i>	13.3	5.2	
<i>GB</i>	3.4	0.2	

Conclusion

En introduction, nous avons présenté l'algorithme ROSENFELD–GRÖBNER comme le principal résultat de cette thèse et nous en avons donné trois applications : décider si un système d'équations polynomiales différentielles admet des modèles, calculer un ensemble caractéristique d'un idéal différentiel premier de type fini et rendre effectives certaines propriétés structurelles des systèmes dynamiques en automatique non linéaire. Chacun de ces trois domaines appelle des études complémentaires spécifiques que nous allons exposer en conclusion.

Les algorithmes qui décident si un système d'équations est contradictoire fournissent des réponses de type oui ou non. Il est donc essentiel de comparer les performances en temps de calcul et en volume de mémoire consommé de ROSENFELD–GRÖBNER avec celles de ses concurrents, notamment avec celles des algorithmes d'élimination de SEIDENBERG. Le premier argument en notre faveur est le suivant : les algorithmes d'élimination sont restreints aux seuls ordres d'élimination sur l'alphabet des indéterminées et l'expérience montre que pour la plupart des systèmes, il existe un ordre qui n'est pas un ordre d'élimination, pour lequel les calculs sont élémentaires. Sur ce sujet, l'exemple des équations d'EULER est particulièrement frappant. Le second argument est que les bases de GRÖBNER calculées par ROSENFELD–GRÖBNER peuvent être réutilisées pour décider après-coup de l'appartenance au radical d'un idéal différentiel de type fini. Toutefois, GRIGOR'EV a donné en 1987 une version de l'algorithme d'élimination de SEIDENBERG en algèbre différentielle ordinaire avec une complexité en temps triplement exponentielle pour le pire des cas. Cette borne est la meilleure connue ; nous souhaitons l'améliorer, soit en établissant que ROSENFELD–GRÖBNER est meilleur, soit en concevant un nouvel algorithme inspiré conjointement des idées de GRIGOR'EV et de celles de ROSENFELD.

Dans le domaine du calcul d'ensembles caractéristiques d'idéaux de polynômes différentiels, nous disposons de résultats complémentaires, non exposés dans cette thèse, qui nous ont été communiqués par MM. LAZARD et OLLIVIER.

Le théorème dont M. LAZARD nous a fait part semble permettre l'extraction d'un ensemble caractéristique d'un idéal différentiel premier donné par une base, au sens de RITT et RAUDENBUSH. Ce résultat est d'autant plus satisfaisant que tout idéal différentiel premier admet une base finie. Une autre conséquence de ce théorème est de simplifier le test d'appartenance au radical d'un idéal différentiel quelconque à partir des bases de GRÖBNER calculées par notre algorithme. D'après M. OLLIVIER, il est également possible de décomposer le radical de tout idéal différentiel de type fini en

une intersection d'idéaux différentiels radiciels définis chacun par un ensemble caractéristique; cette décomposition fournit ensuite une méthode simple pour tester l'appartenance au radical de l'idéal différentiel considéré. Nous souhaitons évidemment développer ces résultats dans un avenir très proche.

Les problèmes d'automatique non linéaire auxquels nous nous intéressons consistent le plus souvent en la recherche de relations qui lient entr'elles certaines grandeurs décrites par un système dynamique non linéaire. La méthode couramment utilisée pour répondre à ce type de question consiste à calculer d'abord un ensemble caractéristique ou une base de GRÖBNER, puis à lire la relation voulue dans l'ensemble de polynômes calculé. Cette méthode provoque visiblement une grande quantité de calculs inutiles. Nous pensons qu'il doit être possible de traiter des systèmes plus importants que ceux que nous avons donnés en exemple dans cette thèse en implantant des versions spécialisées de ROSENFELD-GRÖBNER.

Annexe A

Exemples de sessions

A.1 Une application à l'automatique non linéaire

Nous donnons ici la trace des calculs effectués sur les équations d'un mobile en mouvement rectiligne uniforme étudiées en section 6.3 (page 89). Le programme *rosenfeld1* commence par demander le numéro de fonction de sortie : nous demandons ici au logiciel de n'imprimer que les systèmes réguliers (code 2). Le programme demande ensuite le nombre de dérivations (une) et l'alphabet. On spécifie l'ordre admissible voulu en jouant sur le parenthésage de l'alphabet. Le logiciel lit ensuite les équations et les inéquations du système initial.

```
(1 :tous 2 :terminaux 3 :tous mais sans commentaires)
numero de fonction : 2
nombre de derivations : 1
alphabet : (Y X Vy Vx R)
% ordre : ... > Y' > Y > ... > X' > X > ... > Vy' > Vy > ... > Vx' >
Vx > ... > R' > R
(Entrez les equations (= 0) et les inequations (# 0))
(terminez par <exit;>)
Vx = X';
Vy = Y';
Vx' = 0;
Vy' = 0;
R = X^2 + Y^2;
exit;
```

Voici les systèmes réguliers produits par l'algorithme. Chaque système est précédé par sa position dans l'arbre des scindages, notée suivant une méthode classique. Les impressions se terminent par la liste des indéterminées apparaissant dans l'ensemble des systèmes imprimés et par quelques informations statistiques.

```
% 1/1 1/2 1/2 1/3 2/2 1/3 1/4
R''' = 0;
Vx' = 0;
(2.Vy^2 + 2.Vx^2 - R'') = 0;
```

```

(2.R'' .X^2 - 4.R'.Vx.X + 4.R.Vx^2 - 2.R.R'' + R'^2) = 0;
(2.Vy.Y + 2.Vx.X - R') = 0;
Vx # 0;
(Vy^2 + Vx^2) # 0;
Y # 0;
(R'' .X - R'.Vx) # 0;
R'' # 0;
Vy # 0;

```

```

% 1/1 1/2 1/2 1/3 2/2 3/3 1/1 1/3 1/5

```

```

R''' = 0;
Vx = 0;
(2.Vy^2 - R'') = 0;
(2.R'' .X^2 - 2.R.R'' + R'^2) = 0;
(2.Vy.Y - R') = 0;
R' # 0;
(Vy^2 + Vx^2) # 0;
(R'' .X - R'.Vx) # 0;
Y # 0;
X # 0;
R'' # 0;
Vy # 0;

```

```

% 1/1 1/2 1/2 2/3 1/2 1/3 1/2 1/3

```

```

(2.R.R'' - R'^2) = 0;
Vx' = 0;
(4.R.Vy^2 + 4.R.Vx^2 - R'^2) = 0;
(R' .X - 2.R.Vx) = 0;
(R' .Y - 2.R.Vy) = 0;
R # 0;
Vx # 0;
Y # 0;
(Vy^2 + Vx^2) # 0;
Vy # 0;
R' # 0;

```

```

% 1/1 1/2 1/2 2/3 2/2 1/4 1/1 1/3

```

```

(2.R.R'' - R'^2) = 0;
Vx = 0;
(4.R.Vy^2 - R'^2) = 0;
X = 0;
(2.Vy.Y - R') = 0;
Y # 0;
(Vy^2 + Vx^2) # 0;
R' # 0;
Vy # 0;
R # 0;

```

```

% 1/1 1/2 1/2 3/3 1/2 1/2 3/3 1/1 1/2
R'' = 0;
Vx' = 0;
(Vy^2 + Vx^2) = 0;
(4.R'.Vx.X - 4.R.Vx^2 - R'^2) = 0;
(4.R'.Vy.Y + 4.R.Vx^2 - R'^2) = 0;
Vx # 0;
Y # 0;
R' # 0;
Vy # 0;

```

```

% 1/1 1/2 1/2 3/3 1/2 2/2 2/2 1/1 2/2 1/1 1/2
R = 0;
Vx' = 0;
(Vy^2 + Vx^2) = 0;
(X' - Vx) = 0;
(Vy.Y + Vx.X) = 0;
Y # 0;
Vy # 0;

```

```

% 1/1 1/2 2/2 1/2 1/2 1/3
R''' = 0;
(2.Vx^2 - R'') = 0;
Vy = 0;
(2.Vx.X - R') = 0;
(2.R''.Y^2 - 2.R.R'' + R'^2) = 0;
Vx # 0;
Y # 0;
R'' # 0;

```

```

% 1/1 1/2 2/2 2/2 1/1 1/1
R' = 0;
Vx = 0;
Vy = 0;
X' = 0;
(Y^2 + X^2 - R) = 0;
Y # 0;

```

```

% 1/1 2/2 1/2 1/2 1/3 1/2
(2.R.R'' - R'^2) = 0;
(4.R.Vx^2 - R'^2) = 0;
Vy = 0;
(2.Vx.X - R') = 0;
Y = 0;
Vx # 0;
X # 0;
R # 0;
R' # 0;

```

```

% 1/1 2/2 1/2 2/2 1/1 1/1
R' = 0;
Vx = 0;
Vy = 0;
(X^2 - R) = 0;
Y = 0;
X # 0;

% 1/1 2/2 2/2 1/1 1/1 1/1
R = 0;
Vx = 0;
Vy = 0;
X = 0;
Y = 0;

[Y, X', X, Vy, Vx', Vx, R''', R'', R', R]
% 34 insatisfiable[s]
% 38 non autoreduit[s]
% 0 autoreduit[s]
% 12 autoreduit[s] et coherent[s]
% 11 termina[l|ux]

```

Un programme LEX reprend ensuite les systèmes ci-dessus et les transforme en un script *GB*. L'indéterminée $x^{(r)}$ est notée $x.r$. En algèbre différentielle partielle, le codage des opérateurs de dérivation est malheureusement moins élégant. Voici le script *GB* produit par le programme LEX :

```

corps := INT;
)displayPF off
)type off
v1 := [z1,z2,z3,z4,z5,z6,z7,z8,z9,z10,z11,z12,z13,z14,z15,z16,z17,z18,
z19,z20,z21,z22,z23,z24,z25,z26,z27,z28,z29,z30,z31,z32,z33,z34,z35,
z36,z37,z38,z39,Y, X.1, X, Vy, Vx.1, Vx, R.3, R.2, R.1, R];
poly := DMP (v1,corps);

p1:poly := R.3;
p2:poly := Vx.1;
p3:poly := (2*Vy**2 + 2*Vx**2 - R.2);
p4:poly := (2*R.2*X**2 - 4*R.1*Vx*X + 4*R*Vx**2 - 2*R*R.2 + R.1**2);
p5:poly := (2*Vy*Y + 2*Vx*X - R.1);
p6:poly := Vx*z1 + 1;
p7:poly := (Vy**2 + Vx**2)*z2 + 1;
p8:poly := Y*z3 + 1;
p9:poly := (R.2*X - R.1*Vx)*z4 + 1;
p10:poly := R.2*z5 + 1;
p11:poly := Vy*z6 + 1;
liste1:List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11];

```



```

base1 := sugar (liste1);
p1:poly := R.3;
p2:poly := Vx;
p3:poly := (2*Vy**2 - R.2);
p4:poly := (2*R.2*X**2 - 2*R*R.2 + R.1**2);
p5:poly := (2*Vy*Y - R.1);
p6:poly := R.1*z7 + 1;
p7:poly := (Vy**2 + Vx**2)*z8 + 1;
p8:poly := (R.2*X - R.1*Vx)*z9 + 1;
p9:poly := Y*z10 + 1;
p10:poly := X*z11 + 1;
p11:poly := R.2*z12 + 1;
p12:poly := Vy*z13 + 1;
liste2>List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12];
base2 := sugar (liste2);
p1:poly := (2*R*R.2 - R.1**2);
p2:poly := Vx.1;
p3:poly := (4*R*Vy**2 + 4*R*Vx**2 - R.1**2);
p4:poly := (R.1*X - 2*R*Vx);
p5:poly := (R.1*Y - 2*R*Vy);
p6:poly := R*z14 + 1;
p7:poly := Vx*z15 + 1;
p8:poly := Y*z16 + 1;
p9:poly := (Vy**2 + Vx**2)*z17 + 1;
p10:poly := Vy*z18 + 1;
p11:poly := R.1*z19 + 1;
liste3>List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11];
base3 := sugar (liste3);
p1:poly := (2*R*R.2 - R.1**2);
p2:poly := Vx;
p3:poly := (4*R*Vy**2 - R.1**2);
p4:poly := X;
p5:poly := (2*Vy*Y - R.1);
p6:poly := Y*z20 + 1;
p7:poly := (Vy**2 + Vx**2)*z21 + 1;
p8:poly := R.1*z22 + 1;
p9:poly := Vy*z23 + 1;
p10:poly := R*z24 + 1;
liste4>List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9,p10];
base4 := sugar (liste4);
p1:poly := R.2;
p2:poly := Vx.1;
p3:poly := (Vy**2 + Vx**2);
p4:poly := (4*R.1*Vx*X - 4*R*Vx**2 - R.1**2);
p5:poly := (4*R.1*Vy*Y + 4*R*Vx**2 - R.1**2);
p6:poly := Vx*z25 + 1;
p7:poly := Y*z26 + 1;
p8:poly := R.1*z27 + 1;

```

```

p9:poly := Vy*z28 + 1;
liste5:List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9];
base5 := sugar (liste5);
p1:poly := R;
p2:poly := Vx.1;
p3:poly := (Vy**2 + Vx**2);
p4:poly := (X.1 - Vx);
p5:poly := (Vy*Y + Vx*X);
p6:poly := Y*z29 + 1;
p7:poly := Vy*z30 + 1;
liste6:List (poly) := [p1,p2,p3,p4,p5,p6,p7];
base6 := sugar (liste6);
p1:poly := R.3;
p2:poly := (2*Vx**2 - R.2);
p3:poly := Vy;
p4:poly := (2*Vx*X - R.1);
p5:poly := (2*R.2*Y**2 - 2*R*R.2 + R.1**2);
p6:poly := Vx*z31 + 1;
p7:poly := Y*z32 + 1;
p8:poly := R.2*z33 + 1;
liste7:List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8];
base7 := sugar (liste7);
p1:poly := R.1;
p2:poly := Vx;
p3:poly := Vy;
p4:poly := X.1;
p5:poly := (Y**2 + X**2 - R);
p6:poly := Y*z34 + 1;
liste8:List (poly) := [p1,p2,p3,p4,p5,p6];
base8 := sugar (liste8);
p1:poly := (2*R*R.2 - R.1**2);
p2:poly := (4*R*Vx**2 - R.1**2);
p3:poly := Vy;
p4:poly := (2*Vx*X - R.1);
p5:poly := Y;
p6:poly := Vx*z35 + 1;
p7:poly := X*z36 + 1;
p8:poly := R*z37 + 1;
p9:poly := R.1*z38 + 1;
liste9:List (poly) := [p1,p2,p3,p4,p5,p6,p7,p8,p9];
base9 := sugar (liste9);
p1:poly := R.1;
p2:poly := Vx;
p3:poly := Vy;
p4:poly := (X**2 - R);
p5:poly := Y;
p6:poly := X*z39 + 1;
liste10:List (poly) := [p1,p2,p3,p4,p5,p6];

```

```

base10 := sugar (liste10);
p1:poly := R;
p2:poly := Vx;
p3:poly := Vy;
p4:poly := X;
p5:poly := Y;
liste11:List (poly) := [p1,p2,p3,p4,p5];
base11 := sugar (liste11);

```

```

)axiom on gb.out
base (base1);
base (base2);
base (base3);
base (base4);
base (base5);
base (base6);
base (base7);
base (base8);
base (base9);
base (base10);
base (base11);
)quit

```

Le logiciel *GB* produit un fichier *gb.out* sous une forme peu lisible. Un deuxième programme LEX réécrit les bases de GRÖBNER sous la forme plus agréable que voici :

```

[Y^2 + X^2 - R,
2.Y.X.Vx - Y.R' - 2.X^2.Vy + 2.Vy.R,
Y.X.R'' - Y.Vx.R' - X.Vy.R' + 2.Vy.Vx.R,
2.Y.Vy + 2.X.Vx - R',
2.Y.Vx^2 - Y.R'' - 2.X.Vy.Vx + Vy.R',
2.X^2.R'' - 4.X.Vx.R' + 4.Vx^2.R - 2.R''.R + R'^2,
2.Vy^2 + 2.Vx^2 - R'',
Vx',
R''']

```

```

[Y^2 + X^2 - R,
2.Y.Vy - R',
Y.R'' - Vy.R',
Y.R' + 2.X^2.Vy - 2.Vy.R,
2.X^2.R'' - 2.R''.R + R'^2,
2.Vy^2 - R'',
Vx,
R''']

```

```

[Y^2 + X^2 - R,
2.Y.Vy + 2.X.Vx - R',
Y.Vx - X.Vy,

```

$$\begin{aligned}
& Y.R'' - Vy.R', \\
& Y.R' - 2.Vy.R, \\
& X.R'' - Vx.R', \\
& X.R' - 2.Vx.R, \\
& 2.Vy^2 + 2.Vx^2 - R'', \\
& Vx', \\
& 2.R'' . R - R'^2]
\end{aligned}$$

$$\begin{aligned}
& [Y^2 - R, \\
& 2.Y.Vy - R', \\
& Y.R'' - Vy.R', \\
& Y.R' - 2.Vy.R, \\
& X, \\
& 2.Vy^2 - R'', \\
& Vx, \\
& 2.R'' . R - R'^2]
\end{aligned}$$

$$\begin{aligned}
& [Y^2 + X^2 - R, \\
& 2.Y.X.Vx - Y.R' - 2.X^2.Vy + 2.Vy.R, \\
& 2.Y.Vy + 2.X.Vx - R', \\
& 2.Y.Vx^2 - 2.X.Vy.Vx + Vy.R', \\
& Y.Vx.R' + X.Vy.R' - 2.Vy.Vx.R, \\
& Y.R'^2 + 4.X^2.Vy.R' - 4.X.Vy.Vx.R - 2.Vy.R'.R, \\
& 4.X.Vx.R' - 4.Vx^2.R - R'^2, \\
& Vy^2 + Vx^2, \\
& Vx', \\
& R'']
\end{aligned}$$

$$\begin{aligned}
& [Y^2 + X^2, \\
& Y.Vy + X.Vx, \\
& Y.Vx - X.Vy, \\
& X' - Vx, \\
& Vy^2 + Vx^2, \\
& Vx', \\
& R]
\end{aligned}$$

$$\begin{aligned}
& [Y^2 + X^2 - R, \\
& 2.X.Vx - R', \\
& X.R'' - Vx.R', \\
& Vy, \\
& 2.Vx^2 - R'', \\
& R''']
\end{aligned}$$

$$\begin{aligned}
& [Y^2 + X^2 - R, \\
& X', \\
& Vy, \\
& Vx, \\
& R']
\end{aligned}$$

```
[Y,
X^2 - R,
2.X.Vx - R',
X.R'' - Vx.R',
X.R' - 2.Vx.R,
Vy,
2.Vx^2 - R'',
2.R''.R - R'^2]
```

```
[Y,
X^2 - R,
Vy,
Vx,
R']
```

```
[Y,
X,
Vy,
Vx,
R]
```

A.2 Equations d'Euler

La trace de session que nous donnons ci-dessous montre comment nous avons obtenu à la calculatrice symbolique, l'équation différentielle en v^1 de la section 6.5 (page 93), conséquence des équations d'EULER d'un fluide incompressible en deux dimensions.

La calculatrice symbolique (*calcul.cl*) est un interpréteur de commandes construit au-dessus des unités de compilation décrites en début de chapitre VI, afin de permettre une utilisation interactive des fonctions compilées. Lorsqu'on souhaite utiliser une fonction en mode interprété, on ajoute à la table d'opérateurs de la calculatrice un nouveau mot-clef défini par :

- un symbole (par exemple `red`, `delta`, `{...}`, etc ...)
- un mode (préfixe, infixe, postfixe) avec en plus un mode d'associativité pour les opérateurs infixes : par exemple, l'opérateur binaire `-` est implicitement parenthésé à gauche ($a - b - c$ se lit $(a - b) - c$),
- une priorité,
- un pointeur sur la fonction à appeler pour évaluer l'opérateur.

C'est l'unité de compilation *opérateurs.cl* qui est chargée de la lecture des expressions avec opérateurs. Elle construit un arbre qui est ensuite évalué par la calculatrice. Voici une description des opérateurs utilisés dans la trace qui suit :

alf opérateur préfixe paramétré par une liste. Construit l'alphabet. On peut changer l'alphabet en cours de calcul.

jets opérateur préfixe paramétré par une liste. Définit une notation de jets qui permet de noter agréablement les indéterminées.

{...} opérateur postfixe paramétré par une expression qui s'évalue en un polynôme. Les deux accolades encadrent une suite de dérivations.

delta opérateur infix. Calcule le Δ -polynôme engendré par ses deux opérandes.

red opérateur infix. Appelle l'algorithme de réduction.

rmf opérateur infix. Son deuxième opérande est une liste de polynômes par lesquels la fonction appelée tente de diviser le premier opérande.

Nous donnons à la fois les commandes entrées et les réponses du logiciel. Les deux premières commandes font que l'ordre admissible utilisé lors des calculs sera un ordre d'élimination :

$$\theta p > \phi v^2 > \psi v^1$$

pour toutes valeurs des opérateurs de dérivations θ , ϕ et ψ . Pour les lettres p (de même que pour v^2), on aura $\theta p > \phi p$ si θ est supérieur à ϕ pour l'ordre lexicographique (donné par la commande **jets**):

$$\delta_1 > \delta_2 > \delta_t$$

Pour la lettre v^1 , on aura $\theta v^1 > \phi v^1$ si l'ordre de θ est supérieur à celui de ϕ ou si les deux opérateurs ont même ordre et si θ est supérieur à ϕ pour l'ordre lexicographique sur les dérivations donné ci-dessus.

Les trois polynômes p_1 , p_2 et p_3 sont les trois équations d'EULER. On voit que la réduction par p_3 substitue $-v_1^1$ à v_2^2 . Le polynôme p_4 , qui n'est pas utile pour le calcul, montre comment éliminer la pression des deux premières équations. p_4 comporte un terme embêtant en v_{1t}^2 qu'on peut supprimer par dérivation par δ_2 , puis par réduction par p_3 . Le polynôme ainsi obtenu se nomme p_5 et a la forme

$$A \cdot v_{11}^2 + B \cdot v^2 + C$$

où A , B et C appartiennent à $\mathbb{Q}\{v^1\}$. En dérivant p_4 par δ_2 une fois de plus qu'il n'avait fallu pour obtenir p_5 , on obtient p_6 qui a la "même forme" que p_5 . Une simple réduction algébrique donne p_7 , d'indéterminée principale v^2 et de degré 1 en cette indéterminée. En dérivant p_7 par δ_2 puis en réduisant par p_3 , on obtient un nouveau polynôme du premier degré en v^2 . Une simple réduction algébrique donne la relation cherchée. Le polynôme obtenu est d'ordre 5 en v^1 . Ce polynôme est divisible par v_2^1 (l'initial de p_7 et de p_8). Comme l'idéal engendré par les équations d'EULER est premier et que cet initial ne lui appartient pas, p_9 est bien une conséquence des équations initiales.

nombre de derivations : 3

```
alf [p,v2,v1:ord];
-- ordre : ... > p' > p (lex) > ... > v2' > v2 (lex) > ... > v1' > v1 (ord)
t
```

```
jets [1,2,t];
t
```

```
p1 = v1{t} + v1 * v1{1} + v2 * v1{2} + p{1};
(p{1} + v1{2}.v2 + v1.v1{1} + v1{t})
```

```
p2 = v2{t} + v1 * v2{1} + v2 * v2{2} + p{2};
(p{2} + v1.v2{1} + v2.v2{2} + v2{t})
```

```
p3 = v1{1} + v2{2};
(v2{2} + v1{1})
```

```
p4 = (p1 delta p2) red p3;
(- v1.v2{11} - v2{1t} + (v1{11} + v1{22}).v2 + v1.v1{12} + v1{2t})
```

```
p5 = (p1 delta p2){2} red p3;
(- v1{2}.v2{11} + (v1{112} + v1{222}).v2 + v1.v1{111} + v1{11t} +
v1.v1{122} + v1{22t} - v1{1}.v1{11} + v1{2}.v1{12} - v1{1}.v1{22})
```

```
p6 = (p1 delta p2){22} red p3;
(- v1{22}.v2{11} + (v1{1122} + v1{2222}).v2 + v1.v1{1112} + v1{112t} +
v1.v1{1222} + v1{222t} + 2.v1{2}.v1{111} - 2.v1{1}.v1{112} +
2.v1{2}.v1{122} - 2.v1{1}.v1{222} - v1{12}.v1{11})
```

```
p7 = p6 red p5;
((- v1{2}.v1{1122} - v1{2}.v1{2222} + v1{22}.v1{112} + v1{22}.v1{222}).v2 -
v1.v1{2}.v1{1112} - v1{2}.v1{112t} - v1.v1{2}.v1{1222} - v1{2}.v1{222t} +
(v1.v1{22} - 2.v1{2}^2).v1{111} + 2.v1{2}.v1{1}.v1{112} + v1{22}.v1{11t} +
(v1.v1{22} - 2.v1{2}^2).v1{122} + 2.v1{2}.v1{1}.v1{222} + v1{22}.v1{22t} +
(v1{2}.v1{12} - v1{1}.v1{22}).v1{11} + v1{2}.v1{22}.v1{12} - v1{1}.v1{22}^2)
```

```
p8 = p7{2} red p3;
((- v1{2}.v1{11222} - v1{2}.v1{22222} + v1{222}.v1{112} + v1{222}^2).v2 -
v1.v1{2}.v1{11122} - v1{2}.v1{1122t} - v1.v1{2}.v1{12222} -
v1{2}.v1{2222t} - 3.v1{2}^2.v1{1112} + 3.v1{2}.v1{1}.v1{1122} -
3.v1{2}^2.v1{1222} + 3.v1{2}.v1{1}.v1{2222} + (v1.v1{222} -
3.v1{2}.v1{22}).v1{111} + 3.v1{2}.v1{12}.v1{112} + v1{222}.v1{11t} +
(v1.v1{222} + v1{2}.v1{11} - 2.v1{2}.v1{22}).v1{122} + (v1{22t} -
v1{1}.v1{11} + 3.v1{2}.v1{12} - v1{1}.v1{22}).v1{222})
```

```
p9 = (p8 red p7) rmf [v1{2}];
-- quotient : v1{2}
```

```
((v1.v1{2}.v1{1122} + v1.v1{2}.v1{2222} - v1.v1{22}.v1{112} -
v1.v1{22}.v1{222}).v1{11122} + (- v1.v1{2}.v1{1112} - v1{2}.v1{112t} -
v1.v1{2}.v1{1222} - v1{2}.v1{222t} + (v1.v1{22} - 2.v1{2}^2).v1{111} +
2.v1{2}.v1{1}.v1{112} + v1{22}.v1{11t} + (v1.v1{22} - 2.v1{2}^2).v1{122} +
2.v1{2}.v1{1}.v1{222} + v1{22}.v1{22t} + (v1{2}.v1{12} -
```

```

v1{1}.v1{22}.v1{11} + v1{2}.v1{22}.v1{12} - v1{1}.v1{22}^2).v1{11222} +
(v1{2}.v1{1122} + v1{2}.v1{2222} - v1{22}.v1{112} -
v1{22}.v1{222}).v1{1122t} + (v1.v1{2}.v1{1122} + v1.v1{2}.v1{2222} -
v1.v1{22}.v1{112} - v1.v1{22}.v1{222}).v1{12222} + (- v1.v1{2}.v1{1112} -
v1{2}.v1{112t} - v1.v1{2}.v1{1222} - v1{2}.v1{222t} + (v1.v1{22} -
2.v1{2}^2).v1{111} + 2.v1{2}.v1{1}.v1{112} + v1{22}.v1{11t} + (v1.v1{22} -
2.v1{2}^2).v1{122} + 2.v1{2}.v1{1}.v1{222} + v1{22}.v1{22t} +
(v1{2}.v1{12} - v1{1}.v1{22}).v1{11} + v1{2}.v1{22}.v1{12} -
v1{1}.v1{22}^2).v1{22222} + (v1{2}.v1{1122} + v1{2}.v1{2222} -
v1{22}.v1{112} - v1{22}.v1{222}).v1{2222t} + (3.v1{2}^2.v1{1122} +
3.v1{2}^2.v1{2222} + (v1.v1{222} - 3.v1{2}.v1{22}).v1{112} + v1.v1{222}^2 -
3.v1{2}.v1{22}.v1{222}).v1{1112} - 3.v1{2}.v1{1}.v1{1122}^2 +
(3.v1{2}^2.v1{1222} - 6.v1{2}.v1{1}.v1{2222} + (- v1.v1{222} +
3.v1{2}.v1{22}).v1{111} + (- 3.v1{2}.v1{12} + 3.v1{1}.v1{22}).v1{112} -
v1{222}.v1{11t} + (- v1.v1{222} - v1{2}.v1{11} + 2.v1{2}.v1{22}).v1{122} +
(- v1{22t} + v1{1}.v1{11} - 3.v1{2}.v1{12} +
4.v1{1}.v1{22}).v1{222}).v1{1122} + (v1{222}.v1{112} + v1{222}^2).v1{112t} +
(3.v1{2}^2.v1{2222} + (v1.v1{222} - 3.v1{2}.v1{22}).v1{112} + v1.v1{222}^2 -
3.v1{2}.v1{22}.v1{222}).v1{1222} - 3.v1{2}.v1{1}.v1{2222}^2 +
((- v1.v1{222} + 3.v1{2}.v1{22}).v1{111} + (- 3.v1{2}.v1{12} +
3.v1{1}.v1{22}).v1{112} - v1{222}.v1{11t} + (- v1.v1{222} - v1{2}.v1{11} +
2.v1{2}.v1{22}).v1{122} + (- v1{22t} + v1{1}.v1{11} - 3.v1{2}.v1{12} +
4.v1{1}.v1{22}).v1{222}).v1{2222} + (v1{222}.v1{112} + v1{222}^2).v1{222t} +
((2.v1{2}.v1{222} - 3.v1{22}^2).v1{112} + 2.v1{2}.v1{222}^2 -
3.v1{22}^2.v1{222}).v1{111} + (- 2.v1{1}.v1{222} +
3.v1{22}.v1{12}).v1{112}^2 + ((2.v1{2}.v1{222} +
v1{22}.v1{11} - 2.v1{22}^2).v1{122} - 4.v1{1}.v1{222}^2 + (- v1{12}.v1{11} +
5.v1{22}.v1{12}).v1{222}).v1{112} + (2.v1{2}.v1{222}^2 + (v1{22}.v1{11} -
2.v1{22}^2).v1{222}).v1{122} - 2.v1{1}.v1{222}^3 + (- v1{12}.v1{11} +
2.v1{22}.v1{12}).v1{222}^2)

```

```

lt p9;
v1{11122}

```

A.3 Génération de commentaires

La trace que nous donnons ici montre les commentaires que le programme *rosenfeld1* génère lorsqu'on lui demande d'imprimer tous les systèmes. Il s'agit d'un exemple en algèbre différentielle partielle. L'anneau de polynômes est $\mathbb{Z}\{u, v\}$. Les dérivations sont notées suivant une notation de jets. L'ordre admissible est un ordre d'élimination: $\theta u > \phi v$ pour tous opérateurs de dérivation θ et ϕ ; l'indéterminée θu sera supérieure à ϕv si θ est supérieur à ϕ pour l'ordre lexicographique donné par la notation de jets: $\delta_x > \delta_y > \delta_z$.

```

(1 :tous 2 :terminaux 3 :tous mais sans commentaires)
numero de fonction : 1

```



```

nombre de derivations : 3
alphabet : (u v)
% ordre : ... > u' > u (lex) > ... > v' > v (lex)
notation de jets (nil sinon) : (x y z)
(Entrez les equations (= 0) et les inequations (# 0))
(terminez par <exit;>)
u{xy} = v;
u{x} * u{y} = 1;
v{x} = u{x} * v{z};
v{y} = u{y} * v{z};
u{z} = 1;
exit;

```

Chacun des systèmes engendrés par l'algorithme est précédé par son emplacement dans l'arbre des scindages, par l'ensemble caractéristique qui a été extrait de ses équations (les équations de l'ensemble caractéristique choisi sont notées alternativement A_1, A_2, \dots et B_1, B_2, \dots) et par une description du scindage auquel il correspond.

Les commentaires générés pour ce système-ci sont assez intéressants, parce que l'arbre des scindages est de petite taille. Ce type de commentaires est malheureusement trop peu élaboré pour permettre l'étude d'un système plus important.

```

% 1/1 (NON_AUTOREDUIT)
A1 = (u{z} - 1)
A2 = (v{z}.u{y} - v{y})
A3 = (v{z}.u{x} - v{x})
% ----- SYSTEME -----
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
(v{z}.u{x} - v{x}) = 0;
(u{y}.u{x} - 1) = 0;
(u{xy} - v) = 0;

% 1/1 1/3 (NON_AUTOREDUIT)
ini A3 != 0
ini A2 != 0
REDUCTION (v{y}.v{x} - v{z}^2) = (u{y}.u{x} - 1) REDUIT PAR A3, A2
REDUCTION (v{z}.v{xy} - v{yz}.v{x} - v.v{z}^2) = (u{xy} - v) REDUIT PAR A3
B1 = (v{y}.v{x} - v{z}^2)
B2 = (u{z} - 1)
B3 = (v{z}.u{y} - v{y})
% ----- SYSTEME -----
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
(v{z}.u{x} - v{x}) = 0;
(v{y}.v{x} - v{z}^2) = 0;
(v{z}.v{xy} - v{yz}.v{x} - v.v{z}^2) = 0;
v{z} # 0;

```

```

% 1/1 1/3 1/2 (AUTOREDUIT)
ini B1 != 0
REDUCTION (v{z}.v{y}.u{x} - v{z}^2) = (v{z}.u{x} - v{x}) REDUIT PAR B1
REDUCTION (- v{z}^3.v{yy} + v{z}^2.v{y}.v{yz} - v.v{z}^2.v{y}^2) =
(v{z}.v{xy} - v{yz}.v{x} - v.v{z}^2) REDUIT PAR B1
QUOTIENT (v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = (v{z}^3.v{yy} -
v{z}^2.v{y}.v{yz} + v.v{z}^2.v{y}^2) DIVISE PAR v{z}
QUOTIENT (v{y}.u{x} - v{z}) = (v{z}.v{y}.u{x} - v{z}^2) DIVISE PAR v{z}
A1 = (v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2)
A2 = (v{y}.v{x} - v{z}^2)
A3 = (u{z} - 1)
A4 = (v{z}.u{y} - v{y})
A5 = (v{y}.u{x} - v{z})
% ----- SYSTEME -----
(v{y}.v{x} - v{z}^2) = 0;
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
(v{y}.u{x} - v{z}) = 0;
(v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = 0;
v{z} # 0;
v{y} # 0;

```

```

% 1/1 1/3 1/2 1/3 (NON_AUTOREDUIT)
ini A5 != 0
ini A1 != 0
SYZYGIE ENTRE A1 et A2 REDUITE PAR A5, A4, A2, A1 RESULTAT :
(2.v.v{z}^5.v{y}^5.v{yz} + (- 2.v.v{z}^4.v{zz} + 2.v^2.v{z}^5).v{y}^6)
SYZYGIE ENTRE A4 et A5 REDUITE PAR A5, A4, A2, A1 RESULTAT :
(- 2.v{z}^2.v{y}^4.v{yz} + 2.v{z}.v{zz}.v{y}^5)
SYZYGIE ENTRE A3 et A5 REDUITE PAR A5, A4, A2, A1 RESULTAT :
(- v{z}.v{yz} + v{zz}.v{y})
SYZYGIE ENTRE A3 et A4 REDUITE PAR A5, A4, A2, A1 RESULTAT :
(v{z}.v{yz} - v{zz}.v{y})
QUOTIENT (v{z}.v{y}^4.v{yz} - v{zz}.v{y}^5) = (v{z}^2.v{y}^4.v{yz} -
v{z}.v{zz}.v{y}^5) DIVISE PAR v{z}
QUOTIENT (v{z}.v{yz} - v{zz}.v{y}) = (v{z}.v{y}^4.v{yz} - v{zz}.v{y}^5)
DIVISE PAR v{y}
QUOTIENT (v.v{z}.v{y}^5.v{yz} + (- v.v{zz} + v^2.v{z}).v{y}^6) =
(v.v{z}^5.v{y}^5.v{yz} + (- v.v{z}^4.v{zz} + v^2.v{z}^5).v{y}^6) DIVISE
PAR v{z}
QUOTIENT (v.v{z}.v{yz} + (- v.v{zz} + v^2.v{z}).v{y}) =
(v.v{z}.v{y}^5.v{yz} + (- v.v{zz} + v^2.v{z}).v{y}^6) DIVISE PAR v{y}
B1 = (v{z}.v{yz} - v{zz}.v{y})
B2 = (v{y}.v{x} - v{z}^2)
B3 = (u{z} - 1)

```

```

B4 = (v{z}.u{y} - v{y})
B5 = (v{y}.u{x} - v{z})
% ----- SYSTEME -----
(v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = 0;
(v{y}.v{x} - v{z}^2) = 0;
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
(v{y}.u{x} - v{z}) = 0;
(v.v{z}.v{yz} + (- v.v{zz} + v^2.v{z}).v{y}) = 0;
(v{z}.v{yz} - v{zz}.v{y}) = 0;
v{y} # 0;
v{z} # 0;

```

```

% 1/1 1/3 1/2 1/3 1/2 (NON_AUTOREDUIT)
ini B1 != 0
REDUCTION (v{z}^2.v{yy} + (- v{zz} + v.v{z}).v{y}^2) = (v{z}.v{yy} -
v{y}.v{yz} + v.v{y}^2) REDUIT PAR B1
REDUCTION v^2.v{z}^2.v{y} = (v.v{z}.v{yz} + (- v.v{zz} +
v^2.v{z}).v{y}) REDUIT PAR B1
QUOTIENT v.v{y} = v.v{z}.v{y} DIVISE PAR v{z}
QUOTIENT v = v.v{y} DIVISE PAR v{y}
A1 = v
A2 = (u{z} - 1)

```

```

% ----- SYSTEME -----
(v{z}.v{yz} - v{zz}.v{y}) = 0;
(v{y}.v{x} - v{z}^2) = 0;
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
(v{y}.u{x} - v{z}) = 0;
(v{z}^2.v{yy} + (- v{zz} + v.v{z}).v{y}^2) = 0;
v = 0;
v{y} # 0;
v{z} # 0;

```

```

% 1/1 1/3 1/2 1/3 1/2 1/1 (AUTOREDUIT_COHERENT)
REDUCTION 0 = (v{z}.v{yz} - v{zz}.v{y}) REDUIT PAR A1
REDUCTION 0 = (v{y}.v{x} - v{z}^2) REDUIT PAR A1
REDUCTION 0 = (v{z}.u{y} - v{y}) REDUIT PAR A1
REDUCTION 0 = (v{y}.u{x} - v{z}) REDUIT PAR A1
REDUCTION 0 = (v{z}^2.v{yy} + (- v{zz} + v.v{z}).v{y}^2) REDUIT PAR A1
B1 = v
B2 = (u{z} - 1)
% ----- SYSTEME -----
v = 0;
(u{z} - 1) = 0;
v{z} # 0;

```

v{y} # 0;

% 1/1 1/3 1/2 1/3 1/2 1/1 1/1 (SANS_SOLUTIONS)

% ----- SYSTEME -----

v = 0;

(u{z} - 1) = 0;

0 # 0;

% 1/1 1/3 1/2 1/3 2/2 (SANS_SOLUTIONS)

ini B1 = 0

QUOTIENT 1 = v{z} DIVISE PAR v{z}

QUOTIENT v{zz} = v{zz}.v{y} DIVISE PAR v{y}

% ----- SYSTEME -----

(v.v{z}.v{yz} + (- v.v{zz} + v^2.v{z}).v{y}) = 0;

(v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = 0;

(v{z}.v{yz} - v{zz}.v{y}) = 0;

(v{y}.v{x} - v{z}^2) = 0;

(u{z} - 1) = 0;

(v{z}.u{y} - v{y}) = 0;

(v{y}.u{x} - v{z}) = 0;

v{zz} = 0;

1 = 0;

v{z} # 0;

v{y} # 0;

% 1/1 1/3 1/2 2/3 (SANS_SOLUTIONS)

ini A5 != 0

ini A1 = 0

QUOTIENT 1 = v{z} DIVISE PAR v{z}

QUOTIENT (v{yz} - v.v{y}) = (v{y}.v{yz} - v.v{y}^2) DIVISE PAR v{y}

% ----- SYSTEME -----

(v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = 0;

(u{z} - 1) = 0;

(v{y}.u{x} - v{z}) = 0;

(v{z}.u{y} - v{y}) = 0;

(v{y}.v{x} - v{z}^2) = 0;

(v{yz} - v.v{y}) = 0;

1 = 0;

v{z} # 0;

v{y} # 0;

% 1/1 1/3 1/2 3/3 (SANS_SOLUTIONS)

ini A5 = 0

QUOTIENT 1 = v{y} DIVISE PAR v{y}

```

QUOTIENT 1 = v{z} DIVISE PAR v{z}
% ----- SYSTEME -----
(u{z} - 1) = 0;
(v{z}.v{yy} - v{y}.v{yz} + v.v{y}^2) = 0;
(v{y}.v{x} - v{z}^2) = 0;
(v{z}.u{y} - v{y}) = 0;
1 = 0;
v{y} # 0;
v{z} # 0;

```

```

% 1/1 1/3 2/2 (SANS_SOLUTIONS)
ini B1 = 0
QUOTIENT 1 = v{z} DIVISE PAR v{z}
% ----- SYSTEME -----
(v{z}.v{xy} - v{yz}.v{x} - v.v{z}^2) = 0;
(v{z}.u{x} - v{x}) = 0;
(v{y}.v{x} - v{z}^2) = 0;
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
v{y} = 0;
1 = 0;
v{z} # 0;

```

```

% 1/1 2/3 (SANS_SOLUTIONS)
ini A3 != 0
ini A2 = 0
QUOTIENT 1 = v{z} DIVISE PAR v{z}
% ----- SYSTEME -----
(u{xy} - v) = 0;
(u{y}.u{x} - 1) = 0;
(u{z} - 1) = 0;
(v{z}.u{x} - v{x}) = 0;
v{y} = 0;
1 = 0;
v{z} # 0;

```

```

% 1/1 3/3 (NON_AUTOREDUIT)
ini A3 = 0
B1 = v{z}
B2 = v{x}
B3 = (u{z} - 1)
B4 = (u{y}.u{x} - 1)
% ----- SYSTEME -----
(u{xy} - v) = 0;
(u{y}.u{x} - 1) = 0;

```

```
(u{z} - 1) = 0;
(v{z}.u{y} - v{y}) = 0;
v{x} = 0;
v{z} = 0;
```

```
% 1/1 3/3 1/2 (AUTOREDUIT)
```

```
ini B4 != 0
```

```
REDUCTION (- u{yy} - v.u{y}^2) = (u{xy} - v) REDUIT PAR B4
```

```
REDUCTION - v{y} = (v{z}.u{y} - v{y}) REDUIT PAR B1
```

```
A1 = v{z}
```

```
A2 = v{y}
```

```
A3 = v{x}
```

```
A4 = (u{z} - 1)
```

```
A5 = (u{yy} + v.u{y}^2)
```

```
A6 = (u{y}.u{x} - 1)
```

```
% ----- SYSTEME -----
```

```
v{z} = 0;
```

```
v{x} = 0;
```

```
(u{z} - 1) = 0;
```

```
(u{y}.u{x} - 1) = 0;
```

```
(u{yy} + v.u{y}^2) = 0;
```

```
v{y} = 0;
```

```
u{y} # 0;
```

```
% 1/1 3/3 1/2 1/1 (NON_AUTOREDUIT)
```

```
SYZYGIE ENTRE A2 et A3 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT : 0
```

```
SYZYGIE ENTRE A1 et A3 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT : 0
```

```
SYZYGIE ENTRE A1 et A2 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT : 0
```

```
SYZYGIE ENTRE A5 et A6 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT :
```

```
2.v^2.u{y}^4
```

```
SYZYGIE ENTRE A4 et A6 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT : 0
```

```
SYZYGIE ENTRE A4 et A5 REDUITE PAR A6, A5, A4, A3, A2, A1 RESULTAT : 0
```

```
QUOTIENT v = v.u{y} DIVISE PAR u{y}
```

```
B1 = v
```

```
B2 = (u{z} - 1)
```

```
B3 = (u{y}.u{x} - 1)
```

```
% ----- SYSTEME -----
```

```
v{z} = 0;
```

```
v{y} = 0;
```

```
v{x} = 0;
```

```
(u{z} - 1) = 0;
```

```
(u{yy} + v.u{y}^2) = 0;
```

```
(u{y}.u{x} - 1) = 0;
```

```
v = 0;
```

```
u{y} # 0;
```

```

% 1/1 3/3 1/2 1/1 1/1 (AUTOREDUIT_COHERENT)
REDUCTION 0 = v{z} REDUIT PAR B1
REDUCTION 0 = v{y} REDUIT PAR B1
REDUCTION 0 = v{x} REDUIT PAR B1
REDUCTION u{yy} = (u{yy} + v.u{y}^2) REDUIT PAR B1
A1 = v
A2 = (u{z} - 1)
A3 = u{yy}
A4 = (u{y}.u{x} - 1)
% ----- SYSTEME -----
v = 0;
(u{z} - 1) = 0;
(u{y}.u{x} - 1) = 0;
u{yy} = 0;
u{y} # 0;

```

```

% 1/1 3/3 1/2 1/1 1/1 1/1 (TERMINAL)
% ----- SYSTEME -----
v = 0;
(u{z} - 1) = 0;
u{yy} = 0;
(u{y}.u{x} - 1) = 0;
u{y} # 0;

```

```

% 1/1 3/3 2/2 (SANS_SOLUTIONS)
ini B4 = 0
% ----- SYSTEME -----
(v{z}.u{y} - v{y}) = 0;
(u{xy} - v) = 0;
v{z} = 0;
v{x} = 0;
(u{z} - 1) = 0;
u{y} = 0;
1 = 0;

```

```

[u{xy}, u{x}, u{yy}, u{y}, u{z}, v{xy}, v{x}, v{yy}, v{yz}, v{y},
v{zz}, v{z}, v]
% 7 insatisfiable[s]
% 6 non autoreduit[s]
% 2 autoreduit[s]
% 2 autoreduit[s] et coherent[s]
% 1 termina[l|ux]

```

Bibliographie

- [Bu] B. Buchberger.— *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal (German)* (Ph. D. Thesis. Math. Inst. Univ. of Innsbruck, Austria 1965, and *Aequationes Math.* **4/3** (1970), 374–383)
- [Ca] G. Carrà-Ferro.— *Gröbner bases and differential ideals* (notes de AAEECC5, Menorca, Spain, Springer Verlag (1987), 129–140)
- [Co] H. Comon.— *Résolution de Contraintes dans les Algèbres de Termes* (CNRS and LRI, Univ. Paris–Sud (1992))
- [Col] G. E. Collins.— Subresultants and reduced polynomial remainder sequences (*J. Assoc. Comput. Mach.* **14** (1967), 128–142)
- [De] M. Devin.— *Le portage du système LE_LISP — Mode d'emploi* (Rapport Technique de l'INRIA **50**)
- [Di1] S. Diop.— *Théorie de l'élimination et principe du modèle interne en automatique* (Thèse de doctorat, Univ. Paris-Sud (1989))
- [Di2] S. Diop.— *Elimination in Control Theory* (*Math. Control Signals Systems* **4** (1991), 17–32)
- [Di3] S. Diop.— *Differential-algebraic decision methods and some applications to system theory* (*Theoretical Computer Science* **98** (1992), 137–161)
- [DD] C. Dicrescenzo, D. Duval.— *Algebraic extensions and algebraic closure in Scratchpad* (*Symbolic and algebraic computation. Gianni P. ed. Springer Lecture Notes in Computer Science* **358** (1989), 440–446)
- [DF] S. Diop, M. Fliess.— *On nonlinear observability* (C. Commault et al eds. *Proc. of 1st European Control Conf.* Hermès, Paris (1991), 152–157)
- [F] M. Fliess.— *Automatique et corps différentiels* (*Forum Math. I*, 227–238)
- [FG] M. Fliess, S. T. Glad.— *An Algebraic Approach to Linear and Nonlinear Control* (*Essays on Control: Perspectives in the Theory and its Applications*, Birkhäuser (1993))
- [FLMR] M. Fliess, J. Lévine, Ph. Martin, P. Rouchon *On differentially flat nonlinear systems* (*Proc. IFAC-Symposium NOLCOS'92, Bordeaux* (1992), 408–412)

- [FGLM] J. C. Faugère, P. Gianni, D. Lazard, T. Mora.— *Efficient computation of Gröbner bases by change of orderings* (Journal of Symb. Comp. **16** (1993), 329–344)
- [G] D. Y. Grigor'ev.— *Complexity of quantifier elimination in the theory of ordinary differential equations* (Eurocal'87, LNCS **378**, 11–25)
- [GMO] G. Gallo, B. Mishra, F. Ollivier.— *Some Constructions in Rings of Differential Polynomials* (Lecture Notes in C. Sc. Vol **539** (AAECC-9), 171–182)
- [H] L. C. G. J. M. Habets.— *Characteristic Sets in Commutative Algebra: an overview* (Memorandum COSOR 92-24, Eindhoven University of Technology, Dept. of Math. and Comp. Sc. (1992))
- [JD] N. Dershowitz, J.P. Jouannaud.— *Rewrite Systems* (volume B of Handbook of Theoretical Computer Science, chapter 6 (1990), 244–320, Elsevier Science Publishers)
- [Kn] D. E. Knuth.— *The art of computer programming* (volume 1, Addison-Wesley (1966))
- [Ko] E. R. Kolchin.— *Differential Algebra and Algebraic Groups* (Academic Press, New York (1950))
- [Kön] D. König.— *Theorie der endlichen und unendlichen Graphen* (Chelsea publ. Co. New York (1950))
- [L1] D. Lazard.— *Résolution des systèmes d'équations algébriques* (Theoretical Computer Science **15** (1981), 77–110)
- [L2] D. Lazard.— *A new method for solving algebraic systems of positive dimension* (Discr. App. Math. **33** (1991), 147–160)
- [L3] D. Lazard.— *Systems of algebraic equations (algorithms and complexity)* (Proc. of Cortona Conference (1991), Eisenbud and Robbiano eds. Cambridge Univ. Press (1993))
- [M1] E. Mansfield.— *Differential Gröbner Bases* (PhD Thesis, University of Sydney (1991))
- [M2] E. Mansfield.— *diffgrob2: A symbolic algebra package for analysing systems of PDE using MAPLE* (Technical Report (1993), University of Exeter, Dept of Math.)
- [O11] F. Ollivier.— *Le problème de l'identifiabilité structurelle globale: approche théorique, méthodes effectives et bornes de complexité.* (Thèse de doctorat, Ecole Polytechnique (1990))
- [O12] F. Ollivier.— *Generalized standard bases with applications to control* (Proc. European Control Conference, ECC'91 (1991), 170–176)

- [Ol3] F. Ollivier.– *Some theoretical problems in effective differential algebra and their relation to control theory* (Actes de Nolcos'92, Bordeaux (1992), 301–306)
- [P] J. F. Pommaret.– *New perspectives in control theory for partial differential equations* (IMA Journal of Mathematics Control & Information **9** (1992), 305–330)
- [Ri] J. F. Ritt.– *Differential Algebra* (Amer. Math. Soc, New York (1950))
- [Ro] A. Rosenfeld.– *Specializations in differential algebra* (Trans. Amer. Math. Soc. **90** (1959), 394–407)
- [Rob] L. Robbiano.– *Term orderings on the polynomial rings* (Proceedings of Euro-cal'85 (Linz) Lect. Notes in Comp. Science 204 (1985), 513–517)
- [Se1] A. Seidenberg.– *An elimination theory for differential algebra* (Univ. California Publ. Math. (N.S.) (1956), 31–38)
- [Se2] A. Seidenberg.– *Some basic theorems in differential algebra (characteristic p arbitrary)* (Trans. Amer. Math. Soc. **73** (1952), 174–190)
- [Tr1] W. L. Trinks.– *Über B. Buchbergers Verfahren Systeme algebraischer Gleichungen zu lösen* (J. Number Theory **10** (1978), 475–488)
- [Wa2] D. Wang.– *An elimination method for differential polynomial systems I* (preprint, LIFIA–IMAG, Grenoble, (1994))
- [Wu] Wu Wen–Tsün.– *On the foundation of algebraic differential geometry* (Mechanization of Mathematics, research preprints, Institute of System Science, Academica Sinica, Beijing **3** (1987), 1–26)

Index

- (E) , 14
- $A = A_1, A_2, \dots, A_r$, 21
- $I : E$, 14
- $K\{E\}$, 10
- Y , 37
- $[A] : H_A^\infty$, 15
- $[E]$, 14
- $K\{X\}$, 10
- Δ -polynôme, 31
- $K\langle E \rangle$, 10
- ΘE , 9
- Θ , 9
- \mathcal{M} , 58
- lt , 76
- ltm , 76
- lt^* , 76
- $\sqrt{[E]}$, 26
- lm , 59
- $\{E\}$, 25
- $a \equiv b \pmod{I}$, 14
- $p \simeq q$, 11
- rem , 16, 22
- rem (bases de Gröbner), 60
- élimination, 37
- élimination (ordre d'), 13

- admissible (ordre sur les indéterminées), 10
- admissible (ordre sur les monômes différentiels), 64
- admissible (ordre sur les monômes), 59
- alterné (ordre), 13
- arbre, 44
- artinien (ordre), 11
- auto-réduit (ensemble de monômes), 58
- auto-réduit (ensemble de polynômes), 20

- base (d'un idéal), 57
- base (de GRÖBNER associée à un système régulier), 73
- base (de GRÖBNER principale), 73
- base (théorème de la base finie), 63

- Cantor (ordre de), 13
- CL, 88
- cohérent (ensemble auto-réduit et cohérent), 32
- constante, 9

- dérivation, 9
- Dickson (lemme de), 58
- différentiel (anneau), 9

- ensemble caractéristique, 21
- ensemble caractéristique (algorithme), 83

- facteur (d'un monôme), 58

- Gröbner (bases de), 57
- Gröbner (bases différentielles), 63

- idéal, 14
- idéal différentiel, 15
- idéal différentiel premier, 25
- idéal différentiel radiciel, 25
- indéterminée, 10
- initial, 14

- König (lemme de), 44

- lettre, 10
- lexicographique (ordre sur les monômes), 59
- localement fini (arbre), 44

- modèle algébrique, 30
- modèle différentiel, 29
- monôme, 58
- monoïdéal, 58

monoidéal différentiel, 64
morphisme (d'anneaux), 28

Nullstellensatz algébrique, 30
Nullstellensatz différentiel, 29

observabilité, 89
opérateur de dérivation, 9
ordinaire (anneau), 9
ordre (d'un opérateur de dérivation), 9
orthonormé (ensemble), 66

partiel (anneau), 9
préordre (ensembles auto-réduits), 21
préordre (polynômes), 11
préordre (polynômes, bases de Gröbner),
59
principal (système), 72
principale (indéterminée), 11
propre (opérateur de dérivation), 9

réduction, 16, 22
réduction (bases de Gröbner), 59
réduction partielle, 16
réduite (base complètement), 61
régulier (système), 35
régulier (zéro), 35
résiduel (d'un idéal), 15
rang (ensembles auto-réduits de même
rang), 21
rang (polynômes de même rang), 11
reste partiel, 17
Ritt (lemme de), 38
Rosenfeld (lemme de), 33
Rosenfeld–Gröbner, 68
rosenfeld1 (algorithme), 81
rosenfeld2 (algorithme), 82

séparant, 14
Seidenberg (ordre de), 13

tête (monôme de), 59
terme principal (d'un polynôme), 76
terme principal minimal (d'un idéal), 76
terme principal non dégénéré (d'un idéal),
76
transcendance (degré différentiel de), 93