Une implémentation générique et efficace des bases involutives

Simina Stefania MARIS,

INRIA Sophia Antipolis,
Projet CAFE,
2004, Route des Lucioles, BP 93,
06902 Sophia Antipolis cedex,
France.

smaris@sophia.inria.fr

Juillet 2003

Table des matières

Chapitre 1

Introduction

1.1 Systèmes d'équations aux dérivées partielles

Dans toute la suite du mémoire, nous utiliserons les notations suivantes:

- $x = (x_1, \ldots, x_n)$ un ensemble de n variables indépendantes,
- $y(x) = (y_1(x), \ldots, y_m(x))$ un ensemble de m variables dépendantes,
- $\mu = (\mu_1, \ldots, \mu_n) \in \mathbb{Z}_+^n$ un multi-indice d'entiers positifs ou nuls,
- $|\mu| = \sum_{i=1}^n \mu_i$ la longueur μ,
- $\partial_\mu y_j(x) \triangleq \dfrac{\partial^{|\mu|} y_j(x)}{\partial x_1^{\mu_1} \ldots \partial x_n^{\mu_n}}$.

Considérons le système des équations aux dérivées partielles (EDP) suivant

$$f_k(x, \partial_\mu y(x)) = 0, \quad 0 \leq |\mu| \leq q, \quad k = 1, \ldots, l, \tag{1.1}$$

où les fonctions y_i et f_k sont des fonctions analytiques réelles.

La théorie formelle des équations aux dérivées partielles, développée par C. Riquier et M. Janet dans les années 1890-1930 [16, 17, 18, 24, 25], a pour but de répondre aux deux questions suivantes:

1. Est-ce que le système (1.1) est imcompatible, c'est-à-dire, est-il possible d'obtenir des conséquences incompatibles du système (1.1) (e.g. $1 = 0$) par des manipulations algébriques élémentaires et des dérivations?
2. Quelles sont les conditions initiales supplémentaires à ajouter à ce système pour avoir une solution analytique réelle unique du système (1.1) localement au voisinage d'un point $x^0 = (x_1^0, \ldots, x_n^0) \in \mathbb{R}^n$?

Pour répondre à la deuxième question, nous devons résoudre le problème suivant: De combien de fonctions arbitraires de $n, n-1, \ldots, 1$ variables indépendantes x_i et de combien de constantes arbitraires dépend une solution générale analytique réelle du système (1.1)? Illustrons ce problème par plusieurs exemples.

Exemple 1.1.1. 1. Le système d'EDP suivant

$$\begin{cases} \dfrac{\partial y(x)}{\partial x_1} + 1 = 0, \\[2ex] \dfrac{\partial y(x)}{\partial x_2} + y(x) = 0, \end{cases}$$

est incompatible car nous avons:

$$\frac{\partial}{\partial x_2}\left(\frac{\partial y(x)}{\partial x_1}+1\right)-\frac{\partial}{\partial x_1}\left(\frac{\partial y(x)}{\partial x_2}+y(x)\right)+\left(\frac{\partial y(x)}{\partial x_1}+1\right)=1\neq 0.$$

2. La solution générale du système d'EDP suivant

$$\begin{cases}\dfrac{\partial y(x)}{\partial x_1}=0,\\[2ex] \dfrac{\partial y(x)}{\partial x_2}=0,\end{cases}$$

est $y(x_1, x_2) = c$, où c est une constante réelle. Cette solution dépend d'une seule constante arbitraire.

3. La solution générale du système d'EDP suivant

$$\begin{cases}\dfrac{\partial^3 y(x)}{\partial^2 x_1\,\partial x_2}=0,\\[2ex] \dfrac{\partial^3 y(x)}{\partial x_1\,\partial^2 x_2}=0,\end{cases}$$

est $y(x_1, x_2) = c\,x_1\,x_2 + f(x_1) + g(x_2)$. Cette solution dépend d'une constante arbitraire $c \in \mathbb{R}$ et deux fonctions réelles analytiques arbitraires f et g d'une seule variable indépendante x_i.

Notons que l'intégration explicite d'un système d'EDP est généralement impossible. Ainsi, la théorie de C. Riquier et M. Janet ne cherche pas à intégrer explicitement un système d'EDP mais essaie plutôt de calculer la dimension de l'espace de ses solutions (solutions en séries formelles ou analytiques). La dimension de l'espace des solutions d'un système d'EDP est souvent appelée son *degré de généralité* (*strength* par A. Einstein [7]). Pour cela, en se basant sur les travaux antérieurs de C. Riquier, M. Janet a développé des algorithmes généraux permettant de rendre un système d'EDP *formellement intégrable* (ou *complètement intégrable* [16, 18]). Un système d'EDP est dit formellement intégrable s'il est possible de déterminer quelles sont les dérivées que l'on doit fixer à chaque ordre afin de pouvoir développer une solution en série. Ceci est possible si, à partir des équations du système, on peut connaître toutes les relations différentielles pour un ordre donné, c'est-à-dire si l'on n'obtient pas d'équations d'ordre plus bas par combinaisons différentielles des équations du système. Si un système est formellement intégrable, alors un développement en série peut être obtenu en fixant des valeurs numériques pour un certain nombre de dérivées bien déterminées (ce qui donne le degré de généralité du système d'EDP). Les travaux de C. Riquier et M. Janet montrent alors que cette série formelle, solution du système d'EDP, est localement analytique au voisinage du point de développement en série [16, 17, 18, 24, 25].

Du point de vue analytique, les résultats de C. Riquier et M. Janet [16, 18, 24, 25] généralisent le célèbre théorème de A. L. Cauchy (1842-1843):

Théorème 1.1.1. *Soit le système des EDP du premier ordre:*

$$\frac{\partial y(x)}{\partial x_1}=f\left(x, y(x), \frac{\partial y(x)}{\partial x_2},\ldots,\frac{\partial y(x)}{\partial x_n}\right). \tag{1.2}$$

Soit $x^0 = (x_1^0, \ldots, x_n^0) \in \mathbb{R}^n$ et ϕ une fonction réelle analytique dans un voisinage de $\underline{x}^0 = (x_2^0, \ldots, x_n^0)$. Alors, si f est une fonction réelle analytique dans un voisinage du point

$$\left(x^0, \phi(\underline{x}_0), \left(\frac{\partial \phi}{\partial x_2}\right)(\underline{x}^0), \ldots, \left(\frac{\partial \phi}{\partial x_n}\right)(\underline{x}^0)\right) \in \mathbb{R}^{2n},$$

il existe une fonction réelle analytique y, définie dans un voisinage de x^0, satisfaisant le système (1.2) ainsi que les conditions aux frontières suivantes:

$$y(x_1^0, x_2, \ldots, x_n) = \phi(x_2, \ldots, x_n).$$

De manière plus générale, le théorème de S. Kowalewsky (et G. Darboux (1875)), généralisation du théorème de A. L. Cauchy, est aussi un cas particulier des résultats obtenus par C. Riquier et M. Janet [16, 18, 24, 25]. Rappelons le théorème de S. Kowalewsky.

Théorème 1.1.2. *Soit le système d'EDP suivant*

$$\frac{\partial^{r_i} y_i(x)}{\partial x_1^{r_i}} = f_i\left(x, y_j(x), \frac{\partial y_j(x)}{\partial x_1}, \ldots, \frac{\partial y_j(x)}{\partial x_n}, \ldots\right), \quad 1 \leq i \leq m, \tag{1.3}$$

où f_i contient seulement les dérivées de y_j d'ordre plus petit ou égal à r_j et ne contient aucune des m dérivées $\dfrac{\partial^{r_k} y_k(x)}{\partial x_1^{r_k}}$, $1 \leq k \leq m$. Soit $x^0 = (x_1^0, \ldots, x_n^0) \in \mathbb{R}^n$ et $\sum_{i=1}^{m} r_i$ fonctions réelles analytiques $(\phi_{ks_k})_{1 \leq k \leq m, 0 \leq s_k \leq r_k - 1}$ dans un voisinage de $\underline{x}^0 = (x_2^0, \ldots, x_n^0)$. Alors, si les f_i sont des fonctions réelles analytiques dans un voisinage du point

$$\left(x^0, \phi_{j0}(\underline{x}_0), \left(\frac{\partial \phi_{j0}}{\partial x_1}\right)(\underline{x}^0), \ldots, \left(\frac{\partial \phi_{j0}}{\partial x_n}\right)(\underline{x}^0), \ldots\right),$$

il existe des fonctions réelles analytiques $y = (y_1, \ldots, y_m)$, définies dans un voisinage de x^0, satisfaisant (1.3) ainsi que les $\sum_{i=1}^{m} r_i$ conditions aux frontières suivantes:

$$\begin{cases} y_i(x_1^0, x_2, \ldots, x_n) = \phi_{i0}(x_2, \ldots, x_n), \\ \left(\dfrac{\partial y_i}{\partial x_i}\right)(x_1^0, x_2, \ldots, x_n) = \phi_{i1}(x_2, \ldots, x_n), \\ \vdots & 1 \leq i \leq m. \\ \left(\dfrac{\partial^{r_i - 1} y_i}{\partial x_i^{r_i - 1}}\right)(x_1^0, x_2, \ldots, x_n) = \phi_{i(r_i - 1)}(x_2, \ldots, x_n). \end{cases}$$

Du point de vue algorithmique, pour les systèmes d'EDP linéaires, l'algorithme rendant un système formellement intégrable correspond à calculer ce que, de nos jours, nous appelons une *base de Janet* de l'idéal/module différentiel associé au système. Nous donnerons l'algorithme de Janet dans la prochaine section. Pour les systèmes d'EDP linéaires à coefficients constants, le calcul d'une base de Janet est très similaire au calcul d'une *base de Gröbner* pour un idéal/module polynomial [5, 12] . Notons que le concept de base de Gröbner pour les systèmes polynomiaux a été développé dans la thèse de B. Buchberger en 1965 [5]. Ainsi, il semble bien que, dès les années 1920, M. Janet ait développé un algorithme plus général que le celui des bases de Gröbner pour les systèmes d'EDP linéaires à coefficients variables, mais aussi pour certaines classes de systèmes d'EDP non-linéaires (e.g. systèmes quasi-linéaires ou *orthonomes* au sens de C. Riquier [24, 25]). Notons aussi que la classe des systèmes d'EDP non-linéaires autorisée par l'algorithme de M. Janet est proche de celle pour laquelle les méthodes effectives de l'algèbre différentielle ont été développées (méthodes dues à J. F. Ritt et E. R. Kolchin [19, 26] dans les années 1940-1960). Finalement, dans les années 1970, le concept de système formellement intégrable a été à nouveau développé par D. Spencer et son école dans une approche mathématique moderne (e.g. jets, variétés fibrées, cohomologie de Spencer, coordonnées δ-régulières, système involutifs ...) [13, 14, 15, 21, 22, 29]. Cette approche moderne de la théorie formelle des équations aux dérivées partielles a permis de comprendre en profondeur les liens entre les travaux de M. Janet [17] sur les systèmes d'EDP *en involution* et les travaux de E. Cartan sur les *systèmes extérieurs* [6].

1.2 Les travaux de M. Janet

Avant de donner l'algorithme de M. Janet qui complète un système linéaire d'EDP en un système formellement intégrable, nous devons introduire les concepts de *variables multiplicatrices et non-multiplicatrices*, de *cônes involutifs* et de *systèmes de monômes complets*.

1.2.1 Variables (non-)multiplicatrices: un exemple intuitif

Soit U un ensemble fini de monômes de $\mathbb{M} = \{x^d = x_1^{d_1} \dots x_n^{d_n} \mid |d| = d_1 + \dots + d_n \geq 0\}$. Considérons l'ordre sur les variables indépendantes $x_n > x_{n-1} > \dots > x_1$ (dans la suite, il nous arrivera aussi d'utiliser l'ordre $x_1 > x_2 > \dots > x_n$: cela revient à utiliser l'ordre précédent en permutant les variables x_i).

La variable x_i est appelée *variable multiplicatrice* pour $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \in U$ s'il n'existe aucun monôme dans U ayant un degré α_n en x_n, ..., α_{i+1} en x_{i+1} et un degré strictement supérieur à α_i en x_i. Si x_i n'est pas une variable multiplicatrice pour x^α, alors x_i est dite *variable non-multiplicatrice* pour $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \in U$. Nous noterons par $M_J(x^\alpha)$ l'ensemble des variables multiplicatrices du monôme $x^\alpha \in U$ et par $NM_J(x^\alpha)$ l'ensemble des variables non-multiplicatrices du monôme $x^\alpha \in U$.

Le *cône* d'un monôme $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \in U$ est l'ensemble défini par:

$$C(x^\alpha) = x^\alpha \, \mathbb{M} = \{x_1^{\alpha_1+\beta_1} x_2^{\alpha_2+\beta_2} \dots x_n^{\alpha_n+\beta_n} \mid \beta_i \geq 0, \, 1 \leq i \leq n\}.$$

Le *cône involutif* d'un monôme $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \in U$ est l'ensemble défini par:

$$C_J(x^\alpha) = \{x_1^{\alpha_1+\beta_1} x_2^{\alpha_2+\beta_2} \dots x_n^{\alpha_n+\beta_n} \mid \beta_i = 0 \;\text{ si } x_i \in NM_J(x^\alpha), \, \beta_i \geq 0 \;\text{ si } x_i \in M_J(x^\alpha)\}.$$

Un ensemble fini de monômes U est dit *complet* si tout multiple d'un monôme de U appartient à un unique cône involutif d'un élément de U, c'est-à-dire, nous avons:

$$C(U) \triangleq \cup_{u \in U} u \, \mathbb{M} = C_J(U) \triangleq \cup_{u \in U} u \, C_J(u).$$

Exemple 1.2.1. Considérons l'ensemble des monômes $U = \{x_1^2, \, x_2^2\}$ et l'ordre sur les variables $x_1 > x_2$. Dans notre cas, l'ensemble de tous les monômes en deux variables, $(\mathbb{M},\cdot)$, est isomorphe à $(\mathbb{N}^2,+)$. Donc, nous pouvons représenter l'ensemble U dans $\mathbb{N}^2$ à l'aide des exposants de ses monômes $U' = \{(2,0),\ (0,2)\}$. Le cône de l'élément $x_1^2 \in U$ vaut alors

$$C(x_1^2) = \{x_1^{2+a} \cdot x_2^b \mid a,b \in \mathbb{N}\} \tag{1.4}$$

ou, si l'on travaille avec les multi-indices, nous avons alors:

$$C'((2,0)) = \{(2+a,b) \mid a,b \in \mathbb{N}\}.$$

Le cône de l'élément $x_2^2 \in U$ s'écrit sous la forme

$$C(x_2^2) = \{x_1^a \cdot x_2^{2+b} \mid a,b \in \mathbb{N}\}, \tag{1.5}$$

ou, de manière équivalente:

$$C'((0,2)) = \{(a,2+b) \mid a,b \in \mathbb{N}\}.$$

Nous observons que l'intersection de ces deux cônes n'est pas vide: tout élément de la forme $x_1^{2+a} \cdot x_2^{2+b}$ (ou $(2a,2+b)$) est un multiple commun des deux monômes de U. La complétion involutive réalise une partition de $\mathbb{N}^2$ telle que chaque monôme de l'espace $\mathbb{M}$ se trouve dans un unique cône involutif.

Nous écrivons les variables multiplicatrices des monômes de U dans un tableau et nous mettons $\bullet$ pour les variables non-multiplicatrices du monôme. Pour U, nous avons:

$$\left\{ \begin{array}{l} x_1^2 \\ x_2^2 \end{array} \right. \begin{array}{|cc|} \hline 1 & 2 \\ \hline \bullet & 2 \\ \hline \end{array}$$

Ainsi, x_1 et x_2 sont deux variables multiplicatrices de x_1^2 alors que seule x_2 est une variable multiplicatrice pour x_2^2. Alors, les cônes involutifs de ces deux mônomes sont:

$$\left\{ \begin{array}{l} C_J(x_1^2) = \{x_1^{2+a} \cdot x_2^b \mid a,b \in \mathbb{N}\}, \\ \\ C_J(x_2^2) = \{x_2^{2+b} \mid b \in \mathbb{N}\}. \end{array} \right.$$

La seule prolongation non-multiplicatrice des mônomes de U vaut $x_1\,(x_2^2) = x_1\,x_2^2 \in C(x_2^2)$. Cependant, le monôme $x_1\,x_2^2$ n'appartient à aucun des deux cônes involutifs définis par (1.4) et (1.5). Donc, U n'est pas un ensemble complet de monômes au sens de la définition précédente.

Afin de compléter l'ensemble U, nous ajoutons le monôme $x_1\,x_2^2$ à l'ensemble U pour obtenir $V = \{x_1^2, x_1\,x_2^2, x_2^2\}$. Nous obtenons alors les variables multiplicatrices et non-multiplicatrices suivantes:

$$\left\{ \begin{array}{l} x_1^2 \\ x_1\,x_2^2 \\ x_2^2 \end{array} \right. \begin{array}{|cc|} \hline 1 & 2 \\ \hline \bullet & 2 \\ \hline \bullet & 2 \\ \hline \end{array}$$

Maintenant, les nouveaux cônes involutifs sont:

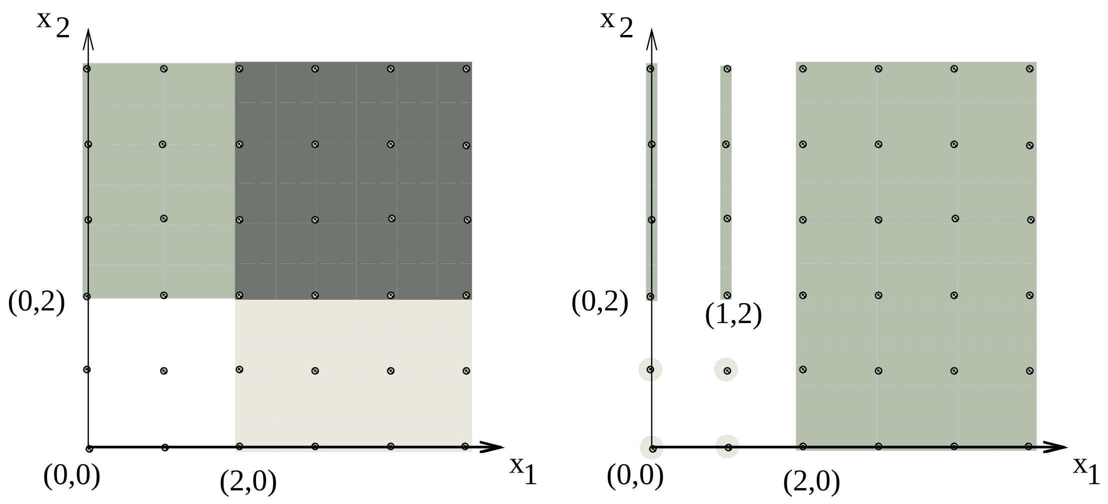
$$\left\{ \begin{array}{l} C_J(x_1^2) = \{x_1^{2+a} \cdot x_2^b \mid a,b \in \mathbb{N}\}, \\ \\ C_J(x_1\,x_2^2) = \{x_1 \cdot x_2^{2+b} \mid a,b \in \mathbb{N}\}, \\ \\ C_J(x_2^2) = \{x_2^{2+b} \mid b \in \mathbb{N}\}. \end{array} \right.$$

L'ensemble V est complet car la prolongation non-multiplicatrice de $x_1\,x_2^2$ par rapport à sa variable non-multiplicatrice x_1 vaut $x_1^2\,x_2^2 = (x_2^2)\,x_1^2 \in C_J(x_1^2)$.

Les monômes ne figurant dans aucun des cônes involutifs $C_J(x_1^2)$, $C_J(x_1\,x_2^2)$ et $C_J(x_2^2)$ sont $x_1\,x_2$, x_1,x_2 et 1.

Dans la figure A, nous avons representé les cônes $C(x_1^2)$ et $C(x_2^2)$ des monômes de U et, dans la figure B, les cônes involutifs de l'ensemble complet V.

A. Intersection des cônes B. Cônes involutifs



1.2.2 L'algorithme de M. Janet

Dans les années 1920, M. Janet a donné un algorithme général transformant un système linéaire et homogène d'EDP en une *base de Janet*, c'est-à-dire en un système *autoréduit* d'EDP linéaires dont toutes les *conditions d'intégrabilité* sont réduites à zéro modulo les équations du système et dont les monômes correspondant aux dérivées des termes de tête des équations forment un ensemble complet. Présentons l'algorithme de M. Janet [16, 18].

Considérons un système d'EDP linéaire et homogène S. Afin d'autoréduire le système S, nous avons besoin de définir le concept de système de côtes.

Définition 1.2.1. – Un *système de côtes* sur les variables dépendantes $x = (x_1, \ldots ,x_n)$ et non-dépendantes $y = (y_1(x), \ldots ,y_m(x))$ est la donnée d'une matrice $C \in \mathbb{Z}_+^{s\times(n+m)}$ d'entiers positifs de la forme:

$$\begin{array}{|c|c|c|c|c||c|c|c|c|} \hline x_1 & x_2 & \ldots & x_{n-1} & x_n & y_1 & y_2 & \ldots & y_m \\ \hline a_{11} & a_{12} & \ldots & a_{1(n-1)} & a_{1n} & b_{11} & b_{12} & \ldots & b_{1m} \\ a_{21} & a_{22} & \ldots & a_{2(n-1)} & a_{2n} & b_{21} & b_{22} & \ldots & b_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{s1} & a_{s2} & \ldots & a_{s(n-1)} & a_{sn} & b_{s1} & b_{s2} & \ldots & b_{sm} \\ \hline \end{array} \tag{1.6}$$

Divisons la matrice C en deux sous-matrices $C = (A, B)$, où $A \in \mathbb{Z}_+^{s\times n}$ et $B \in \mathbb{Z}_+^{s\times m}$.

- La *côte associée à l'opérateur différentiel* ∂_μ est $A\,\mu^T \in \mathbb{Z}_+^s$, avec $\mu = (\mu_1, \ldots ,\mu_n)$ est un multi-indice.
- La *côte associée à la fonction* $\partial_\mu y_i(x)$ est $A\,\mu^T + B_i \in \mathbb{Z}_+^s$, où B_i est la ième colonne de la matrice B.
- Nous dirons que $\partial_\mu y_i(x)$ est *plus haut* que $\partial_\nu y_j(x)$ si la première des différences non-nulles

$$(b_{i1} - \sum_{k=1}^{n} a_{1k}\,\mu_k) - (b_{j1} - \sum_{k=1}^{n} a_{1k}\,\nu_k), \ldots ,(b_{in} - \sum_{k=1}^{n} a_{nk}\,\mu_k) - (b_{jn} - \sum_{k=1}^{n} a_{nk}\,\nu_k)$$

TAB. 1.1 – *calcul de la base de Janet associée à un système S*

Données: le système S linéaire et homogène d'EDP,
une relation d'ordre monomial
Sorties: la base de Janet associée au système S

1. Autoréduction
2. Complétion de l'ensemble des monômes de tête des équations de S
3. Détermination des conditions d'intégrabilité:
nous déterminons toutes les paires des termes de tête u_i et u_j des équations e_i et e_j de S telles que

$$\frac{\partial u_i(x)}{\partial x_{i_k}} = \frac{\partial^{r_1+r_2+\ldots+r_l} u_j(x)}{\partial x_{j_1}^{r_1} \partial x_{j_2}^{r_2} \ldots \partial x_{j_l}^{r_l}},$$

où x_{i_k} est une variable non-multiplicatrice de u_i
et $x_{j_1},\ldots,x_{j_l}$ sont des variables multiplicatrices de u_j.

Alors, nous calculons les conditions d'intégrabilité suivantes:

$$c_{ij} = coef(v_j) \cdot \frac{\partial e_i}{\partial x_{i_k}} - coef(v_i) \cdot \frac{\partial^{r_1+r_2+\ldots+r_l} e_j}{\partial x_{j_1}^{r_1} \partial x_{j_2}^{r_2} \ldots \partial x_{j_l}^{r_l}}$$

4. Réduction de toutes les conditions d'intégrabilité c_{ij}
5. Fin de l'algorithme
- si tous les c_{ij} sont nuls, nous retournons le système S,
- sinon, nous ajoutons les conditions d'intégrabilité non nulles au système S et nous retournons en **1**.

est positive. Nous notons alors $\partial_\mu y_i(x) > \partial_\nu y_j(x)$.

Lemme 1.2.1. Nous avons les propriétés suivantes:
- $1 > \partial_\mu$, $\forall\, \mu \in \mathbb{Z}_+^n \backslash \{(0,\ldots,0)\}$,
- $\partial_\mu y_i(x) > y_i(x)$,
- $\partial_\mu y_i(x) > \partial_\nu y_j(x) \Rightarrow \partial_{\mu+\sigma} y_i(x) > \partial_{\nu+\sigma} y_j(x)$.

Dans la suite, nous prendrons le système de côtes suivant:

x_1	x_2	$\ldots$	x_{n-1}	x_n	y_1	y_2	$\ldots$	y_m
1	1	$\ldots$	1	1	0	0	$\ldots$	0
0	0	$\ldots$	0	0	1	2	$\ldots$	m
0	0	$\ldots$	0	1	0	0	$\ldots$	0
0	0	$\ldots$	1	0	0	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	0	$\ldots$	0	0	0	0	$\ldots$	0

(1.7)

Lemme 1.2.2. Considérons le système de côtes défini par la matrice (1.7). Alors,
- la côte de $\partial_\mu\, y_i(x)$ est $(|\mu|, i, \mu_n, \mu_{n-1}, \ldots, \mu_1)$,
- $\partial_\mu\, y_i(x) > \partial_\nu\, y_j(x)$ si et seulement si la première différence non-nulle

$$|\mu| - |\nu|,\ i-j,\ \mu_n - \nu_n,\ \ldots,\ \mu_1 - \nu_1$$

est positive.

- Nous avons l'orde suivant $x_n > x_{n-1} > \ldots > x_1$ sur les variables indépendantes.
- La relation d'ordre $>$ est un ordre total sur les fonctions $\partial_\mu y_i(x)$.

Considérons un système linéaire et homogène d'EDP S. En utilisant l'ordre total sur les fonctions $\partial_\mu y_i(x)$ défini par le système de côte (1.7) dans chacune des équation de S, nous pouvons tirer dans le membre de gauche le terme $\partial_\mu y_i(x)$ le plus haut (terme de tête de l'équation). Maintenant, il se peut que le terme de tête d'une équation ou une de ses dérivées soit égal à un terme d'une autre équation. Nous pouvons alors *réduire* cette dernière équation par rapport à la première grâce à des dérivations, multiplications de coefficients et substitutions. Ainsi, par réductions successives, nous *autoréduisons* le système S. En particulier, nous pouvons toujours supposer que le système est sous une *forme résolue* ou *orthonome*, c'est-à-dire:

- les premiers membres des équations sont tous distincts,
- les seconds membres des équations ne contiennent que des éléments plus bas que les premiers membres.

Les termes de tête des équations d'un système sous forme résolue ainsi que leurs dérivées sont appelés *dérivées principales*, les autres sont appelés *dérivées paramétriques*.

Définition 1.2.2. Un système d'EDP sous forme résolue est *formellement intégrable* si, par dérivations et éliminations, nous n'obtenons jamais des relations entre les dérivées paramétriques.

L'algorithme de M. Janet permet d'obtenir en sortie un système linéaire d'EDP formellement intégrable et complet ayant les mêmes solutions que le système S en entrée. Le décompte du nombre des dérivées paramétriques à un ordre fixé permet d'obtenir le degré de généralité du système S. Pour cela, nous avons besoin de définir *l'ensemble des monômes complémentaires.*

Définition 1.2.3. Soit $U = \{\alpha, \beta, \ldots, \zeta\}$ un ensemble fini de monômes. Nous définissons l'ensemble des monômes complémentaires $V = \cup_{i=1}^n V^{(i)}$ de U, où:

$$\left\{\begin{array}{l} V^{(n)} = \{x_n^k \mid k < \max\{\alpha_n,\beta_n,\ldots,\zeta_n\}, k \notin \{\alpha_n,\beta_n,\ldots,\zeta_n\}\}, \\ \\ V^{(i)} = \{x_n^{\theta_n} \ldots x_{i+1}^{\theta_{i+1}} x_i^k \mid \theta = (\theta_1,\ldots,\theta_n) \in U, k < \max\{\alpha_i,\beta_i,\ldots,\zeta_i\}, k \notin \{\alpha_i,\beta_i,\ldots,\zeta_i\}\}. \end{array}\right.$$

Nous associons à tout monôme de V les variables multiplicatrices suivantes:

- $x_1, \ldots, x_{n-1}$ à tout monôme de $V^{(n)}$,
- $x_1, \ldots, x_{i-1}$ à tout monôme u de $V^{(i)}$ ainsi que celles des variables $x_{i+1}, \ldots, x_n$ qui sont multiplicatrices pour u par rapport à U.

Pour un système formellement intégrable dont l'ensemble des premiers membres forme un ensemble complet pour chacune des variables dépendantes $y_i(x)$, l'union des ensembles complémentaires correspond aux conditions initiales qu'il faut fixer afin d'obtenir localement une solution analytique du système. Plus précisément, nous avons le théorème fondamental suivant dû à C. Riquier et M. Janet [16, 18, 24, 25].

Théorème 1.2.1. *Soit S un système analytique réel d'EDP (linéaire) sous forme résolue et complètement intégrable. Pour chacune des variables dépendantes $y_i(x)$, soit U_i l'ensemble des monômes correspondants aux dérivées de $y_i(x)$ apparaissant dans les premiers membres de S. Soient $\widetilde{U}_i$ le complété de U_i et V_i l'ensemble complémentaire de $\widetilde{U}_i$. Alors, il existe une unique solution $y(x) = (y_1(x), \ldots, y_m(x))$ du système S qui est analytique réelle au voisinage d'un point régulier $x_0 = (x_1^0, \ldots, x_n^0)$ et qui satisfait les conditions* *intiales suivantes*

$$\forall\, i = 1 \dots m,\, \forall\, x^{\mu} \in V_i : \quad (\partial_{\mu} y_i)(x)_{\{x_j = x_j^0 \;|\; x_j \in NM_J(x^{\mu})\}} = f_{i\mu}(x_k, x_k \in M_J(x^{\mu})),$$

où les $f_{i\mu}$ sont des fonctions analytiques réelles données au voisinage de x_0.

1.2.3 Exemple de M. Janet

Afin de simplifier les notations, nous noterons par $\partial_1^{\alpha_1}\, \partial_2^{\alpha_2}\, \partial_3^{\alpha_3}$ l'opérateur différentiel:

$$\frac{\partial^{\alpha_1+\alpha_2+\alpha_3}}{\partial\, x_1^{\alpha_1}\, \partial\, x_2^{\alpha_2}\, \partial\, x_3^{\alpha_3}}.$$

Illustrons l'algorithme de M. Janet sur le système d'équations aux dérivées partielles:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) - x_2\, \partial_1^2\, y(x) = 0, \\ \partial_2^2\, y(x) = 0. \end{array}\right. \tag{1.8}$$

Etudions l'intégrabilité formelle du système d'EDP linéaire à coefficients variants (2.0.2) afin d'étudier ses solutions analytiques au voisinage du point $x_0 = (x_1^0, x_2^0, x_3^0)$. Pour cela, nous devons déterminer le degré de généralité du système (2.0.2).

Nous associons à chaque dérivée entrant dans les équations de (2.0.2) le monôme correspondant: par exemple, à la dérivée $\partial_{13} y(x) = \dfrac{\partial^2 y(x)}{\partial x_1 \partial x_3} = y_{13}(x)$ correspond le monôme $x_1\, x_3$. Par rapport au système de côte donné par (1.7), nous obtenons la forme resolue suivante de (2.0.2):

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|} \hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline \end{array}$$

Les monômes de tête sont donc x_3^2 et x_2^2. Complétons l'ensemble de monôme $U = \{x_3^2, x_2^2\}$. La prolongation non-multiplicatrice $x_3\, x_2^2$ de x_2^2 n'appartient à aucun des cônes involutifs des éléments de U. Nous devons donc ajouter $x_3\, x_2^2$ à U afin d'obtenir un ensemble complet de monômes. Cela revient à ajouter la nouvelle équation $\partial_3\, \partial_2^2\, y(x) = 0$ au système (2.0.2). Nous obtenons le système suivant:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0, \\ \partial_2^2\, \partial_3\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|} \hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline 1 & 2 & \bullet \\ \hline \end{array}$$

Maintenant, il nous faut calculer la condition d'intégrabilité: pour cela, il nous faut recommencer le calcul précédent avec les équations à la place des monômes de tête. Ainsi, en utilisant le fait que $x_3\, (x_3\, x_2^2) = x_3^2\, x_2^2 = x_2^2\, (x_3^2)$, alors, en dérivant la première équation deux fois par rapport à x_3, nous obtenons

$$\partial_2^2 \partial_3^2 y(x) = x_2\, \partial_1^2 \partial_2^2 y(x) + 2\, \partial_1^2 \partial_2 y(x),$$

qui se réduit à l'aide des deux dernières équations en une nouvelle équation:

$$\partial_1^2 \partial_2 y(x) = 0.$$

Cette condition d'intégrabilité n'est pas nulle et donc le système (2.0.2) n'est pas formellement intégrable. Il nous faut donc ajouter cette nouvelle équation au système (2.0.2) et recommencer de nouveau avec:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0, \\ \partial_2^2\, \partial_3\, y(x) = 0, \\ \partial_1^2\, \partial_2\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|}\hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline \end{array} \tag{1.9}$$

Le calcul des prolongations non-multiplicatrices donne le système complet suivant:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0, \\ \partial_1^2\, \partial_2\, y(x) = 0, \\ \partial_2^2\, \partial_3\, y(x) = 0, \\ \partial_1^2\, \partial_2\, \partial_3\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|}\hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline \end{array}$$

De nouveau, en dérivant par rapport aux variables non-multiplicatrices $\bullet$ les équations et en réduisant, nous obtenons une nouvelle équation:

$$\partial_1^4 y(x) = 0.$$

Le système n'ést pas formellement intégrable. Donc, nous devons ajouter cette nouvelle équation au système (1.9) et nous obtenons le nouveau système suivant:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0, \\ \partial_1^2 \partial_2\, y(x) = 0, \\ \partial_2^2\, \partial_3\, y(x) = 0, \\ \partial_1^2\, \partial_2\, \partial_3\, y(x) = 0, \\ \partial_1^4\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|}\hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline \end{array}$$

De nouveau, nous complétons le système précédent puis nous dérivons les équations par rapport à leurs variables non-multiplicatrices. Après réduction, nous obtenons le système complet formellement intégrable suivant:

$$\left\{\begin{array}{l} \partial_3^2\, y(x) = x_2\, \partial_1^2\, y(x), \\ \partial_2^2\, y(x) = 0, \\ \partial_1^2\, \partial_2\, y(x) = 0, \\ \partial_2^2\, \partial_3\, y(x) = 0, \\ \partial_1^2\, \partial_2\, \partial_3\, y(x) = 0, \\ \partial_1^4\, y(x) = 0, \\ \partial_1^4\, \partial_3\, y(x) = 0. \end{array}\right. \quad \begin{array}{|ccc|}\hline 1 & 2 & 3 \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & 2 & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline 1 & \bullet & \bullet \\ \hline \end{array}$$

Les monômes associés aux premiers membres du système précédent sont:

$$U = \{x_3^2,\, x_2^2\, x_3,\, x_1^2\, x_2\, x_3,\, x_1^4\, x_3,\, x_2^2,\, x_1^2\, x_2,\, x_1^4\}.$$

Calculons les monômes complémentaires V de U:

- Pour $i = 3$, les exposants en x_3 des monômes de U sont 2, 1 et 0, d'où $V^{(3)} = \emptyset$.
- Pour $i = 2$, nous avons:
  - pour la queue x_3^2, l'exposant en x_2 des monômes de U est 0,
  - pour la queue x_3^1, les exposants en x_2 des monômes de U sont 2,1,0,

– pour la queue x_3^0, les exposants en x_2 des monômes de U sont 2,1,0, d'où $V^{(2)} = \emptyset$.

– Pour $i = 1$, nous avons:
  – pour les queues x_3^2, $x_3\, x_2^2$ et x_2^2, l'exposant en x_1 des monômes de U est 0,
  – pour la queue $x_1^2\, x_2\, x_3$, l'exposant en x_1 des monômes de U est 2. Donc, nous avons les deux monômes complémentaires suivants:

$$x_1\, x_2\, x_3,\; x_2\, x_3.$$

  – pour la queue $x_1^4\, x_3$, il y a les quatre monômes complémentaires suivants:

$$x_1^3\, x_3,\; x_1^2\, x_3,\; x_1\, x_3,\; x_3,$$

  – pour la queue $x_1^2\, x_2$, il y a les deux monômes complémentaires suivants:

$$x_1\, x_2,\; x_2,$$

  – enfin, pour la queue x_1^4, il y a les quatre monômes complémentaires suivants:

$$x_1^3,\; x_1^2,\; x_1,\; 1.$$

L'ensemble des monômes complémentaires à U vaut donc:

$$V = \{x_1\, x_2\, x_3,\; x_2\, x_3,\; x_1^3\, x_3,\; x_1^2\, x_3,\; x_1\, x_3,\; x_3,\; x_1\, x_2,\; x_2,\; x_1^3,\; x_1^2,\; x_1,\; 1\}$$

Aucun de ces monômes n'a de variables multiplicatrices, c'est-à-dire nous avons:

$$V = \left\{ \begin{array}{l|c|c|c|}
\cline{2-4}
x_1\, x_2\, x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_2\, x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1^3\, x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1^2\, x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1\, x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1\, x_2, & \bullet & \bullet & \bullet \\ \cline{2-4}
x_2 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1^3 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1^2 & \bullet & \bullet & \bullet \\ \cline{2-4}
x_1 & \bullet & \bullet & \bullet \\ \cline{2-4}
1 & \bullet & \bullet & \bullet \\ \cline{2-4}
\end{array} \right.$$

Ainsi, au voisinage d'un point $x_0 = (x_1^0, x_2^0, x_3^0)$, il existe une unique solution du système (2.0.2) qui est analytique réelle et qui satisfait les conditions intiales suivantes:

$$\left\{ \begin{array}{l}
(\partial_1\, \partial_2\, \partial_3\, y)(x_0) = a, \\
(\partial_2\, \partial_3\, y)(x_0) = b, \\
(\partial_1^3\, \partial_3\, y)(x_0) = c, \\
(\partial_1^2\, \partial_3\, y)(x_0) = d, \\
(\partial_1\, \partial_3\, y)(x_0) = e, \\
(\partial_3\, y)(x_0) = f, \\
(\partial_1\, \partial_2\, y)(x_0) = g, \\
(\partial_2\, y)(x_0) = h, \\
(\partial_1^3\, y)(x_0) = i, \\
(\partial_1^2\, y)(x_0) = j, \\
(\partial_1\, y)(x_0) = k, \\
y(x_0) = l.
\end{array} \right.$$

Nous concluons que le degré de généralité du système (2.0.2) est de 12 constantes.

1.3 Thème du stage

La complétion involutive est une alternative aux bases de Gröbner [5, 12] qui représente une première étape de la réduction de systèmes d'équations polynomiales ou aux dérivées partielles. Pour les systèmes des EDP, un premier algorithme de complétion a été donné par M. Janet (1920) [16, 18]. Il existe plusieurs implémentations de l'algorithme de M. Janet dans divers systèmes de calcul formel [1, 2, 9, 27]. Les nouveaux algorithmes pour le calcul d'une complétion de Janet, utilisant des arbres binaires, ont une complexité plus faible que la complexité des algorithmes initiales (e.g. $O(n + d)$ au lieu de $O(n(n + d))$) [10, 11].

Le but de ce stage est d'utiliser la généricité du language de programmation Aldor [3, 4, 30] pour produire une implémentation efficace et générale de la complétion involutive. L'efficacité sera obtenue en implémentant les divisions involutives de Janet et Pommaret [8, 21, 22] avec les structures de données (arbres binaires) et les algorithmes proposés par V. Gerdt, Y. Blinkov et D. Yanovich [10, 11]. La généricité sera obtenue, comme dans [27], en paramétrisant le programme par la division involutive à utiliser (Janet ou Pommaret). Nous travaillons sous Aldor avec la bibliothèque Algebra ainsi que sous Maple (pour vérifier et comparer les résultats).

Chapitre 2

Bases Involutives

Dans le tout reste du mémoire, nous utiliserons les notations suivantes.

Notation 2.0.1.
- $x = (x_1, \ldots, x_n)$ est un ensemble de n variables indépendentes.
- Nous noterons un monôme par $x^d = x_1^{d_1} \ldots x_n^{d_n}$, où $d = (d_1, \ldots, d_n) \in \mathbb{Z}_+^n$ est un multi-indice.
- Le degré total de x^d est $\deg(x^d) = |d|$, où $|d| = d_1 + \ldots + d_n$ est la longueur du multi-indice d.
- Le degré relatif de x^d en x_i est $\deg_i(x^d) = d_i$.
- Si $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}_+^n$ sont deux multi-indices, alors $\alpha + \beta$ désignera le multi-indice défini par $(\alpha_1 + \beta_1, \ldots, \alpha_n + \beta_n)$.
- Nous noterons par $\mathbb{M}$ l'ensemble de tous les monômes en $x_1, \ldots, \; x_n$, c'est-à-dire $\mathbb{M} = \{x^d \,|\, |d| \geq 0\}$.

Introduisons la définition d'une *division involutive* qui jouera un rôle central dans le reste du mémoire.

Définition 2.0.1. [8] On définit une **division involutive** L sur $\mathbb{M}$ comme une relation $|_L$ sur $U \times \mathbb{M}$ (où U est une partie finie de $\mathbb{M}$) telle que pour tout $u, v \in U$ et $m, p \in \mathbb{M}$, nous avons:

1. $u \,|_L\, m \Rightarrow u \mid m$,
2. $u \,|_L\, u, \quad \forall\, u \in U$,
3. $(u \,|_L\, (u\, m)$ et $u \,|_L\, (u\, p)) \Leftrightarrow (u \,|_L\, (u\, m\, p))$,
4. si $u \,|_L\, m$ et $v \,|_L\, m$, alors $u \,|_L\, v$ ou $v \,|_L\, u$,
5. si $u \,|_L\, v$ et $v \,|_L\, m$, alors $u \,|_L\, m$,
6. si $V \subseteq U$ et $u \in V$, alors “$u \,|_L\, m$ dans U implique $u \,|_L\, m$ dans V”.

Remarque 2.0.1. Il est important de noter que la terminologie “division involutive” a été introduite pour la première fois dans [8]. Elle vient du fait que pour tester l'involution d'un système d'équations aux dérivées partielles [6, 13, 21, 22], il faut utiliser la division de Pommaret (voir Définition 2.0.6) dans un système de coordonnées $x = (x_1, \ldots, x_n)$ suffisamment générique. Cependant, il est important de noter que les autres bases involutives n'ont pas de liens avec le concept de système d'équations aux dérivées partielles involutif. Voir [22] pour plus de détails.

Nous pouvons alors définir le concept de *diviseur involutif* et de *monôme multiplicateur.*

Définition 2.0.2. [8] Soit L une division involutive sur $\mathbb{M}$ et $v = u\, w$, $u, v, w \in \mathbb{M}$.

- Si $u \,|_L\, v$, alors u est appelé un **diviseur involutif** de v et $w = v/u$ est un **monôme multiplicateur** pour u. L'ensemble des monômes multiplicateurs de u est noté par:

$$L(u{,}U) = \{w \in \mathbb{M} \,|\, u \,|_L\, (u\, w)\}.$$

– Si $u \nmid_L v$, alors w est un **monôme non-multiplicateur** pour u.

En restreignant la définition des monômes multiplicateurs aux simples variables x_i, nous obtenons alors les définitions des *variables multiplicatrices* et *non-multiplicatrices.*

Définition 2.0.3. Soient L une division involutive sur $\mathbb{M}$ et U un sous-ensemble fini de $\mathbb{M}$. La division involutive L associe à chaque monôme u de U les deux sous-ensembles de $\{x_1, \dots, x_n\}$ suivants:

– les **variables multiplicatrices** $M_L(u,U) = \{x_i \mid u \mid_L (x_i\, u)\}$,

– les **variables non-multiplicatrices** $NM_L(u,U) = \{x_i \mid u \nmid_L (x_i\, u)\}$.

Nous avons $M_L(u,U) \subset L(u,U)$ et $NM_L(u,U) \not\subseteq L(u,U)$ et:

$$M_L(u,U) \cup NM_L(u,U) = \{x_1, \dots, x_n\}, \quad M_L(u,U) \cap NM_L(u,U) = \emptyset.$$

Il est souvent plus facile de définir une division involutive en spécifiant les variables multiplicatrices et non-multiplicatrices pour un monôme $u \in U$. Pour cela, il suffit de vérifier que les variables multiplicatrices satisfont les points 4, 5 et 6 de Définition 2.0.1. Donnons quelques exemples classiques de divisions involutives.

Définition 2.0.4. *(***division de Thomas***)* Soit U un ensemble de monômes. On définit:

$$h_i(U) \triangleq \max\{\deg_i(u) \mid u \in U\}.$$

Alors, x_i est une *variable multiplicatrice pour u* si:

$$\deg_i(u) = h_i(U).$$

Exemple 2.0.1. Considérons l'ensemble de monômes $U = \{x_1\, x_2,\, x_2^2,\, x_3\}$. Nous avons $h_1(U) = 1$, $h_2(U) = 2$, $h_3(U) = 1$ et donc:

$$M_T(x_1\, x_2,U) = \{x_1\}, \quad M_T(x_2^2,U) = \{x_2\}, \quad M_T(x_3,U) = \{x_3\}.$$

Dans la suite, nous noterons l'ensemble des monômes de U avec leurs variables multiplicatrices comme suit:

$x_1\, x_2$	x_1	$\bullet$	$\bullet$
x_2^2	$\bullet$	x_2	$\bullet$
x_3	$\bullet$	$\bullet$	x_3

Définition 2.0.5. *(***division de Janet***)* Soient $d_1, \dots, d_i \geq 0, U$ un ensemble de monômes et $x_1 > x_2 > \dots > x_n$ un ordre sur les variables x_i. On définit l'ensemble:

$$[d_1, \dots, d_i] \triangleq \{u \in U \mid \deg_j(u) = d_j, 1 \leq j \leq i\}.$$

Alors, x_i est une *variable multiplicatrice pour $u \in U$* si:

– soit $i = 1$ et $\deg_1(u) = \max\{\deg_1(v) \mid v \in U\}$,

– soit $i > 1$, $u \in [d_1, \dots, d_{i-1}]$ et $\deg_i(u) = \max\{\deg_i(v) \mid v \in [d_1, \dots, d_{i-1}]\}$.

Exemple 2.0.2. Considérons l'ensemble de monômes $U = \{x_1\, x_2,\, x_2^2,\, x_3\}$ avec l'ordre $x_1 > x_2 > x_3$ sur les variables x_i. Alors, nous avons:

$$M_J(x_1\, x_2,U) = \{x_1, x_2, x_3\}, \quad M_J(x_2^2,U) = \{x_2, x_3\}, \quad M_J(x_3,U) = \{x_3\}.$$

$x_1\, x_2$	x_1	x_2	x_3
x_2^2	$\bullet$	x_2	x_3
x_3	$\bullet$	$\bullet$	x_3

Définition 2.0.6. *(***division de Pommaret***)* Soit $x_1 > x_2 > \ldots > x_n$ un ordre sur les variables x_i et $u = x_1^{d_1} \ldots x_k^{d_k}$, $d_k > 0$. Alors, les *variables multiplicatrices de* u sont les x_j tels que $j \geq k$.

Exemple 2.0.3. Considérons l'ensemble de monômes $U = \{x_1 x_2,\ x_2^2,\ x_3\}$ avec l'ordre $x_1 > x_2 > x_3$ sur les variables x_i. Alors, nous avons:

$$M_P(x_1 x_2) = \{x_2, x_3\}, \quad M_P(x_2^2) = \{x_2, x_3\}, \quad M_P(x_3) = \{x_3\}.$$

$x_1 x_2$	$\bullet$	x_2	x_3
x_2^2	$\bullet$	x_2	x_3
x_3	$\bullet$	$\bullet$	x_3

Remarque 2.0.2. Tout d'abord, remarquons qu'un même ensemble de monômes admet généralement des variables multiplicatrices différentes suivant la division involutive choisie (voir les Exemples 2.0.1, 2.0.2 et 2.0.3).

Ensuite, notons que, contrairement à la divison de Janet et Pommaret, la division de Thomas ne dépend pas d'un ordre sur les variables x_i.

Finalement, contrairement à la divison de Thomas et de Janet, la division de Pommaret ne dépend que du monôme u et non de l'ensemble U.

Proposition 2.0.1. *[8] Les divisions de Thomas, Janet et Pommaret sont involutives.*

Démonstration. On donne la démonstration pour la division de Janet.

Les conditions 1, 2 et 3 sont évidentes.

4. Soient u et v deux diviseurs involutifs de m. Alors, nous avons soit $\deg_i(u) = \deg_i(v) = d_i$ pour $1 \leq i \leq k$, soit $\deg_k(u) > \deg_k(v)$ (c'est-à-dire, la variable x_k est multiplicatrice pour u et non-multiplicatrice pour v). Parce que u est un diviseur de Janet de m, nous avons

$$\deg_k(m) \geq \deg_k(u) > \deg_k(v)$$

et v n'est pas un diviseur de Janet pour m (contradiction). Alors, si u et v sont des diviseurs de Janet pour m, nous avons $\deg_k(u) = \deg_k(v)$ et $u \mid_J v$.

5. Si $u \mid_J v$, alors $v = u \cdot \mu(u)$, où $\mu(u)$ est un monôme formé avec seulement des variables multiplicatrices pour u. Si $v \mid_J m$, alors nous avons

$$m = v \cdot \mu(v) = u \cdot \mu(u) \cdot \mu(v) = V \cdot \mu(u)$$

et donc $u \mid_J m$.

6. Soit $u = x_1^{d_1} \ldots x_k^{d_k} x_{k+1}^{d_{k+1}} \mu$. Si $u \mid_J m$ dans U, alors $x_1, \ldots, x_{k+1}$ sont des variables multiplicatrices pour $m \in U$. Alors, $\deg_k(u) = \max\{\deg_k(v) \mid v \in [d_1, \ldots, d_{k-1}]\}$, d'où $[d_1, \ldots, d_{k-1}] \in [d_1, \ldots, d_k]$. □

Définition 2.0.7. – Le **cône** du monôme $u \in U$ est défini par $C(u) = u\,\mathbb{M}$.

– Le **cône involutif** de u est défini par $C_L(u) = u\,L(u{,}U)$.

Nous pouvons maintenant définir la notion d'un *ensemble L-involutif* ou *complet.*

Définition 2.0.8. [16, 18] Soient U un sous-ensemble de $\mathbb{M}$ et L une division involutive sur $\mathbb{M}$. Alors, l'ensemble U est **L-involutif** ou **complet** si nous avons:

$$C(U) \triangleq \bigcup_{u \in U} u\,C(u) = C_L(U) \triangleq \bigcup_{u \in U} u\,C_L(u).$$

Exemple 2.0.4. Reconsidérons l'Exemple 2.0.1. L'ensemble $U = \{x_1\,x_2,\ x_2^2,\ x_3\}$ n'est pas complet pour la division de Thomas car la multiplication de $x_1\,x_2$ par rapport à sa variable non-multiplicatrice x_2 est égale à $x_1\,x_2^2 \in C(U)$ mais les monômes x_2^2 et x_3 ne sont pas des diviseurs involutifs de $x_1\,x_2^2$ (x_1 n'est pas une variable multiplicatrice pour x_2^2), c'est-à-dire $x_1\,x_2^2 \notin C_T(U)$.

Exemple 2.0.5. Reconsidérons l'Exemple 2.0.2. L'ensemble $U = \{x_1\,x_2,\ x_2^2,\ x_3\}$ n'est pas complet pour la division de Janet car la multiplication de x_3 par rapport à sa variable non-multiplicatrice x_1 est égale à $x_1\,x_3 \in C(U)$ mais les monômes $x_1\,x_3$ et x_2^2 ne sont pas des diviseurs involutifs de $x_1\,x_3$, c'est-à-dire $x_1\,x_3 \notin C_J(U)$.

Exemple 2.0.6. Reconsidérons l'Exemple 2.0.3. L'ensemble $U = \{x_1\,x_2,\ x_2^2,\ x_3\}$ n'est pas complet pour la division de Pommaret car la multiplication de x_3 par rapport à sa variable non-multiplicatrice x_1 est égale à $x_1\,x_3 \in C(U)$ mais les monômes $x_1\,x_2$ et x_2^2 ne sont pas des diviseurs involutifs de $x_1\,x_3$, c'est-à-dire $x_1\,x_3 \notin C_P(U)$.

Nous avons les définitions suivantes [8].

Définition 2.0.9.
- Un ensemble $\widetilde{U}$ de monômes fini et $L-$involutif est appelé une $L-$**complétion** de $U \subseteq \widetilde{U}$ (ou **complétion involutive** de U) si nous avons:

$$C(U) = C_L(\widetilde{U}) = C(\widetilde{U}).$$

- Un ensemble $U \subseteq \mathbb{M}$ est **de type fini** si U admet une $L-$complétion $\widetilde{U}$.
- Une division involutive est **noethérienne** si tout ensemble fini U de $\mathbb{M}$ est de type fini, c'est-à-dire, U admet une $L-$complétion.

Proposition 2.0.2. *[8] Les divisions de Thomas et de Janet sont noethériennes.*

Proposition 2.0.3. *[8] Si L est une division involutive noethérienne de* $\mathbb{M}$, *alors tout idéal monômial admet une base involutive finie.*

Exemple 2.0.1. La division de Janet est une division involutive et noethérienne. Ainsi, par Proposition 2.0.3, on peut toujours trouver une $L-$complétion $\widetilde{U}$ d'un ensemble fini de monômes U. Une telle complétion $\widetilde{U}$ de U est appelée *base de Janet* de l'idéal monomial engendré par l'ensemble U.

Définition 2.0.10. La multiplication d'un monôme $u \in U$ par une variable x_i est appellée **prolongation** de u. La prolongation est dite **multiplicatrice** si x_i est une variable multiplicatrice pour u sinon, **non-multiplicatrice**.

Définition 2.0.11.
- [8] Un ensemble U est **localement involutif** par rapport à la division involutive L si toute prolongation non-multiplicatrice d'un élément de U admet un diviseur involutif dans U, c'est-à-dire:

$$\forall\, u \in U, \forall\, x_i \in NM_L(u,U), \exists\, v \in U :\ v \,|_L\, (u\, x_i).$$

- [8] Une division involutive L est **continue** si pour tout ensemble fini U et pour toute série finie $\{u_i\}_{1\leq i\leq k}$ d'éléments de U telle que

$$\forall\, i < k, \exists\, x_j \in NM_L(u_i,U) :\ u_{i+1} \,|_L\, (u_i\, x_j),$$

nous avons alors $u_i \neq u_j$ pour $i \neq j$

Théorème 2.0.1. *[8] Si une division involutive L est continue, alors tout ensemble localement involutif par rapport à L est involutif.*

Corollaire 2.0.1. *[8] Les divisions de Thomas, Janet et de Pommaret sont continues.*

Définition 2.0.12. [8] Une division involutive continue L est **constructive** si pour tout $U \subset \mathbb{M}$, $u \in U$, $x_i \in NM_L(u,U)$ tels que $u\,x_i \notin C_L(U)$ et

$$\forall\, v \in U, \forall\, x_j \in NM_L(v,U):\ v\,x_j \mid u\,x_i,\, v\,x_j \neq u\,x_i \Rightarrow v\,x_j \in C_L(U),$$

alors la condition suivante est vérifiée:

$$\forall\, w \in C_L(U),\, u\,x_i \notin w\,L(w, U \cup \{w\}).$$

Proposition 2.0.4. *[8] Les divisions de Thomas, Janet et Pommaret sont constructives.*

Théorème 2.0.2. *[8, 16, 18] Soit U un ensemble (non-involutif) de type fini par rapport à une division constructive L. Alors, il existe une procédure complétant U en un ensemble $L-$involutif $\widetilde{U} \supseteq U$ qui consiste à ajouter à U toutes les prolongations non-multiplicatrices des éléments de U.*

Corollaire 2.0.2. *[8] Si U est un ensemble de type fini par rapport à une division involutive, alors il existe une unique complétion minimale $\bar{U}$ de U telle que pour toute autre complétion $\widetilde{U}$ de U, nous avons $\bar{U} \subseteq \widetilde{U}$.*

Tab. 2.1 – ***Algorithme:*** *Complétion involutive*

Entrée: un ensemble fini de monômes U **Sortie:** une complétion involutive $\widetilde{U}$ de U

begin
  $\widetilde{U} := U$
  tant que $\exists\, u \in \tilde{U}$ et $x \in NM_L(u,\tilde{U})$ tels que
    $u\,x$ n'admet pas des diviseurs involutifs dans $\widetilde{U}$
    choisir des tels u et x tels que $u\,x$ soit le minimum par rapport à l'ordre $\prec$
    $\widetilde{U} := \widetilde{U} \cup \{u\,x\}$
end

Exemple 2.0.7. Soient l'ensemble $U = \{x_1\,x_2,\ x_2^2,\ x_3\}$ avec la relation d'ordre $x_1 > x_2 > x_3$ et la division de Janet comme division involutive. Alors, nous avons la liste des variables multiplicatrices suivante:

$x_1\,x_2$	x_1	x_2	x_3
x_2^2	$\bullet$	x_2	x_3
x_3	$\bullet$	$\bullet$	x_3

Les prolongations non-multiplicatrices des monômes sont $x_1\,x_2^2$ et $x_1\,x_3$. Nous choisissons le plus petit des deux monômes, c'est-à-dire $x_1\,x_3$, et nous l'ajoutons à U pour obtenir $V = \{x_1\,x_3, x_1\,x_2,\ x_2^2,\ x_3\}$. Les variables multiplicatrices de V sont les suivantes:

$x_1\,x_2$	x_1	x_2	x_3
$x_1\,x_3$	x_1	$\bullet$	x_3
x_2^2	$\bullet$	x_2	x_3
x_3	$\bullet$	$\bullet$	x_3

On calcule de nouveau les prolongations non-multiplicatrices, et on trouve $x_1\,x_3$, que l'on ajoute à U pour obtenir V:

$x_1\,x_2$	x_1	x_2	x_3
$x_1\,x_3$	x_1	$\bullet$	x_3
x_2^2	$\bullet$	x_2	x_3
$x_2\,x_3$	$\bullet$	$\bullet$	x_3
x_3	$\bullet$	$\bullet$	x_3

De nouveau, nous calculons les prolongations non-multiplicatrices des monômes de V. Il n'y a pas de telles prolongations sans diviseurs involutifs dans U. Alors, la complétion involutive de l'ensemble $U = \{x_1\,x_2,\, x_2^2,\, x_3\}$ est $\widetilde{U} = \{x_1\,x_2,\, x_1\,x_3,\, x_2^2,\, x_2\,x_3,\, x_3\}$.

Chapitre 3

Arbres de Janet

3.1 Définitions

Dans ce chapitre, nous considérons la division involutive de Janet et l'ordre lexicographique. Rappelons que pour la division de Janet, l'ensemble de monômes $U \subset \mathbb{M}$ est divisé en sous-ensembles de la forme:

$$[d_1, \ldots, d_i] \triangleq \{u \in U \mid d_j = \deg_j(u), 1 \leq j \leq i\}.$$

La variable x_{i+1} est *multiplicatrice* pour $u \in U$ si

$$\deg_{i+1}(u) = \max\{\deg_{i+1}(v) \mid v \in [d_1, \ldots, d_i]\},$$

sinon *non-multiplicatrice*.

On représente un ensemble fini U de monômes par un arbre binaire, appelé **arbre de Janet**, dont la structure reflète la partition des éléments de U en sous-ensembles de la forme $[d_1, \ldots, d_i]$.

La racine de cet arbre est le monôme $1 = x_1^0 \ldots x_n^0$. Chaque élément ν de l'arbre est défini par un triplet $\{u, ndg, nvr\}$, où:

- $mon(\nu) = u$ est le monôme de ν,
- ndg est un pointeur vers le noeud suivant en degré (ou le pointeur vers la "gauche"),
- nvr est un pointeur vers le noeud suivant en variable (ou le pointeur vers la "droite").

Si l'on ne peut aller ni à gauche ni à droite, on assigne *nil* aux pointeurs correspondants.

Définition 3.1.1. – Les **noeuds internes** sont caractérisés par:

$$(nvr \neq nil) \vee ((ndg \neq nil) \wedge (\deg(mon(ndg)) = \deg(u) + 1)).$$

– Les **feuilles** sont définies par:

$$(nvr = nil) \wedge (ndg = nil).$$

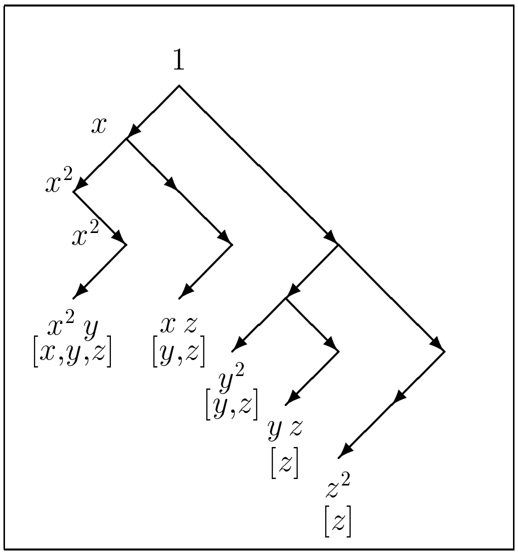
Les feuilles de l'arbre sont les monômes de l'ensemble U.

Exemple 3.1.1. Considérons l'ensemble de monômes $U = \{x^2 y,\ x z,\ y^2,\ y z,\ z^2\}$ et l'ordre $x > y > z$. Les variables multiplicatrices et non-multiplicatrices de U sont données par le tableau suivant:

Monôme u	Variables	
	$M_J(u,U)$	$NM_J(u,U)$
$x^2\,y$	$x,\,y,\,z$	—
$x\,z$	$y,\,z$	x
y^2	$y,\,z$	x
$y\,z$	z	$x,\,y$
z^2	z	$x,\,y$

Pour construire l'arbre de Janet associé à U, nous partons de la racine 1 et nous considérons la première variable indépendante (qui, dans notre exemple, est $x_1 = x$). Tant qu'il existe le *ndg* (c'est-à-dire $1 \cdot x \mid x^2\,y$), on calcule le *ndg*. On arrive au monôme x^2, pour lequel $nvr = nil$ (car $x^2 \cdot x \nmid x^2\,y$), mais, comme $x^2 \neq x^2\,y$, il existe *nvr*, qui est aussi x^2. Quand on passe à *nvr*, on considère la variable suivante. Alors, on a x^2 comme noeud et l'on considère maintenant la deuxième variable (ici $x_2 = y$). Le *ndg* de ce noeud est maintenant $x^2\,y$. Comme $x^2\,y$ n'a plus ni *ndg* ni *nvr*, ce monôme constitue une feuille de l'arbre de Janet (on peut aussi voir cela en remarquant $x^2\,y \in U$).

Au commencement, la racine a toutes les variables multiplicatrices assignées. Le calcul de *ndg* n'affecte pas leur nombre. Par contre, si l'on peut calculer *nvr* et $ndg \neq nil$, alors on supprime la variable courante de la liste des variables multiplicatrices et on considère la variable suivante.



On cherche la base de Janet pour l'idéal engendré par $U = \{x^2\,y,\, x\,z,\, y^2,\, y\,z,\, z^2\}$. La liste des monômes et leurs variables multiplicatrices est donnée par le tableau suivant:

$x^2\,y$	x	y	z
$x\,z$	$\bullet$	y	z
y^2	$\bullet$	y	z
$y\,z$	$\bullet$	$\bullet$	z
z^2	$\bullet$	$\bullet$	z

Alors, la base de Janet de l'ensemble U est définie par:

$$\widetilde{U} = \{x^2\,y,\, x^2\,z,\, x\,y^2,\, x\,y\,z,\, x\,z,\, y^2,\, y\,z,\, z^2\}.$$

L'intérêt majeur pour représenter un ensemble de monômes U par un arbre de Janet est que la recherche d'un diviseur involutif d'un monôme dans U a une borne de complexité bien plus faible que les bornes des algorithmes classiques.

Théorème 3.1.1. *[8] Soit d le degré total maximal des monômes d'un ensemble fini U. Alors, la borne de complexité de l'algorithme* jdivisor *(donné dans la section suivante) est*

$$O(d+n),$$

où n est le nombre de variables indépendantes x_i.

3.2 Descriptions des algorithmes

L'algorithme jtr construit l'arbre de Janet associé à un ensemble fini U de monômes. Cet algorithme utilise les fonctions suivantes:

- *ndg*(*n*,*i*): réalise la multiplication du monôme u avec la i-ème variable x_i,
- *leaves*(jtr): donne les feuilles d'un arbre binaire,
- *sort*: classe un ensemble suivant l'ordre lexicographique,
- *tree*: crée un arbre.

On utilise aussi, dans les algorithmes suivants, la fonction *nodes*(jtr) qui liste tous les noeuds d'un arbre donné. De plus, pour calculer le pointeur *ndg* d'un élément d'un arbre de Janet, nous utilisons la fonction $leftallowed$. De manière similaire, pour calculer le pointeur *nvr* d'un élément d'un arbre de Janet, nous utilisons la fonction *rightallowed*. Nous désignerons aussi par $left$ le pointeur *ndg* et par *right* le pointeur *nvr*.

L'algorithme jtr est donné dans le Table 4.

L'idée pour calculer la base monomiale de Janet d'un ensemble donné U est:

1. on part de l'arbre trivial, [1,*nil*,*nil*], correspondant au monôme unité, et de l'ensemble $V := U$;
2. si le monôme unité est un élément de l'ensemble U, alors la base est $\{1\}$, et sinon, on suit l'algorithme;
3. on initialise la base par $\emptyset$
4. tant que l'ensemble V est non vide, on choisit un élément x de V sans diviseurs dans $V \setminus \{x\}$, et on redéfinit V comme $V \setminus \{x\}$. Le plus simple est de choisir le plus petit élément de V. Si x n'a pas de diviseurs involutifs parmi les feuilles de l'arbre (le réponse est donnée par la fonction *jdivisor*), l'on ajoute l'élément x à la base, on insère x dans l'arbre, et l'on ajoute les prolongations non-multiplicatrices des feuilles à V (les deux dernières opérations étant réalisées par la fonction *jinsert*). Le monôme 1 ne peut pas être consideré comme diviseur involutif de x, même si l'on a $1 \mid_J x$, parce que 1 ne fait pas partie de l'ensemble initial (il est ajouté seulement pour initialiser l'arbre).
5. on finit l'algorithme quand V est vide, et on retourne la base involutive.

L'insertion se fait selon l'idée suivante:

1. Nous commençons avec l'arbre ν et nous prenons 1 pour indice de la variable actuelle.
2. Nous calculons le monôme à insérer dans l'arbre à l'aide de la fonction *olt* et nous construisons l'arbre ayant pour feuilles l'ensemble des feuilles de l'arbre de départ et le monôme à insérer.

TAB. 3.1 – *fonction leftallowed*(u,l,i)

Entrées: u un monôme
l un ensemble ordonné
i l'indice d'une variable
Sorties: $true$ si $u \cdot x_i$ divise le plus grand
élément de l et $false$ sinon

```
begin
  ans := true
  if u ∈ l or l = ∅ then ans := false
  else
    leaf := l_1
    if deg_i(leaf) < deg_i(u) or
      (deg_i(leaf) = deg_i(u) and deg_{i+1}(leaf) ≤ deg_{i+1}(u)) then ans := false
end
```

TAB. 3.2 – *fonction rightallowed*(u,l,i)

Entrées: u un monôme
l un ensemble ordonné
i l'indice d'une variable
Sorties: $true$ si l'on peut calculer nvr et $false$ sinon

```
begin
  ans := true
  if u ∈ l or l = ∅ then ans := false
  else
    if ¬((¬leftallowed(u,l,i) ∧ u | l_1) ∨ (leftallowed(u,l,i) ∧ u | l_2)) then
      ans := false
  return ans
end
```

TAB. 3.3 – ***Algorithme:*** *l'arbre de Janet* $jtr(lst,node,i,sst)$

Entrées: lst un ensemble de monômes
$node$ la racine de l'arbre
i l'indice de la "plus grande" variable (par défaut $i = 1$)
sst l'ensemble des variables multiplicatrices
Sorties: un arbre ayant les monômes de lst pour feuilles
et $node$ comme racine

```
begin
  n := node
  st := sst
  l := sort(lst)
  if n ∈ l then
    tr := tree(n)
    l := l \ {n}
  else
    if leftallowed(n,l,i) then left := jtr(l,ndg(n,i),i,st)
    else left := nil
  l := sort(l \ leaves(left))
  if l ≠ ∅ and rightallowed(n,l,i) then
    if left ≠ ∅ then st := st \ {i}
    i := i + 1
    right := jtr(l,n,i,st)
  else right := nil
  tr := tree(n, left, right)
  return tr
end
```

3. Nous pouvons élargir l'ensemble V dans deux situations:
   (a) Soit le degré en x_i du noeud actuel ν est plus petit que le degré en x_i de u et le noeud actuel n'admet pas de noeud à sa droite. Dans ce cas, la variable x_i devient non-multiplicatrice pour toutes les feuilles du sous-arbre ayant la racine en ν et nous calculons les prolongations non-multiplicatives avec la fonction *jprolong*.
   (b) Soit ν et u ont le même degré en x_i et ν n'admet pas de noeud à sa droite. Dans ce cas, x_i est une variable non-multiplicatrice pour u, et nous ajoutons $u \cdot x_i$ à V.

L'algorithme *jdivisor* cherche le diviseur involutif du monôme u parmi les éléments de l'ensemble $leaves(JT)$. Si u n'a pas de diviseurs involutifs dans $leaves(JT)$, alors *jdivisor* retourne *nil*. Puisque nous ne pouvons pas considérer l'arbre [1,*nil*,*nil*] (pour des raisons qui tiennent à *MJBAS*), cette situation particulière est exclue dans les lignes 4–5.

On construit un arbre ayant comme racine le monôme w et le monôme

$$w \prod_{j=1}^{n} x_i^{\deg_j(u)-\deg_j(w)}$$

comme unique feuille. La fonction *lolt*, qui prend les mêmes arguments que *olt*, calcule cette feuille.

L'algorithme *jprolong* prend comme arguments un sous-arbre *jt* tel que x_i est une variable non-multiplicatrice pour l'ensemble de ses feuilles et un ensemble V. Il retourne l'ensemble V élargi des prolongations non-multiplicatrices en x_i des monômes de $leaves(jt)$.

L'algorithme *jinsert* insère dans l'arbre JT le monôme u vérifiant la condition

$$jdivisor(JT,u) = nil$$

et ajoute à l'ensemble V les prolongations non-multiplicatrices des monômes de $leaves(jt)$. *mult_set*(*nd*) donne les variables multiplicatrices de la racine.

Enfin, l'algorithme *MJBAS* calcule la base de Janet minimale de l'idéal monomial $Id(U)$, où U est un ensemble fini de monômes.

Pour choisir un $x \in V$ sans diviseurs dans $V \setminus \{x\}$, il suffit de prendre le plus petit élément de V pour l'ordre $\prec$ (dans ce cas, l'ordre lexicographique).

On travaille avec les multi-indices: chaque multi-indice représentant un monôme. Par exemple, [1,2,1] représente le monôme $x\,y^2\,z$ et $[0,0,\ldots,0]$ le monôme 1.

Comme cela, la comparaison des degrés relatifs en la variable i revient à la comparaison des enièmes éléments des deux multi-indices.

3.3 Implantations sous Maple et Aldor

Nous avons implémenté les algorithmes en Maple et en Aldor. Autant que possible, nous avons essayé de garder les mêmes noms pour les fonctions utilisées. Parce que nous avons travaillé avec des listes en Aldor, les programmes Maple ont aussi pris comme arguments des listes.

TAB. 3.4 – ***Algorithme:*** *jdivisor* $jdivisor(JT,w)$

Entrées: JT un arbre de Janet
w un monôme
Sorties: le monôme-feuille $u \mid_J w$ et nil sinon

```
begin
  ν := JT
  i := 1
  if leaves(JT) = nodes(JT) then
    ans := nil
  else
    while node(ν) | w and i < n do
      while left(ν) and node(left(ν)) | w do
        ν := left(ν)
      if right(ν) then
        ν := right(ν)
      else if left(ν) then
        ans := nil
      else
        ans := node(ν)
      i := i + 1
  return ans
end
```

TAB. 3.5 – ***Algorithme:*** *One Leaf Tree olt(u,i,w)*

Entrées: u un monôme
i l'index de la variable x_i
w un diviseur de u
Sorties: un arbre ayant pour racine w et pour unique feuille
$$w \prod_{j=1}^{n} x_i^{\deg_j(u)-\deg_j(w)}$$

```
begin
  p := lolt(u,i,w)
  tr := tree(p)
  j := n
  while j ≥ i do
    while deg_j(up(p,j)) ≥ deg_j(w) do
      tr := tree(up(p,j),tr,empty)
      p := up(p,j)
    if deg_j(up(p,j)) < deg_j(w) then
      tr := tree(p,empty,tr)
    j := j − 1
  return tr
end
```

TAB. 3.6 – *fonction lolt*(u,i,w)

Entrées: u un monôme
i l'index de la variable x_i
w un diviseur de u
Sorties: $w \prod_{j=1}^{n} x_i^{\deg_j(u)-\deg_j(w)}$

```
begin
  p := 1
  for k from 1 to n do
    if k < i then
      p := p · x_k^{deg_k(w)}
    else
      p := p · x_k^{deg_k(u)}
  return p end
```

TAB. 3.7 – ***Algorithme:*** *jprolong jprolong*(jt,i,V)

Entrées: jt un (sous-)arbre de Janet
i l'index de la variable non-multiplicatrice x_i
pour l'ensemble des éléments de $leaves(jt)$
V un ensemble de monômes
Sorties: l'ensemble V élargi des prolongations non-multiplicatrices
en x_i des monômes de $leaves(jt)$

```
begin
  ans := V
  if nodes(jt) ≠ leaves(jt) then
    ν := jt
    while left(ν) do
      if right(ν) then
        ans := jprolong(right(ν),i,ans)
        ν := left(ν)
    if right(ν) then
      ans := jprolong(right(ν),i,ans)
    else
      ans := ans ∪ {node(ν) · x_i}
  return ans
end
```

TAB. 3.8 – ***Algorithme:*** *jinsert* $jinsert(JT,u,V)$

Entrées: JT un arbre de Janet
u un monôme à insérer tel que $jdivisor(JT,u) = nil$
V un ensemble de monômes
Sorties: JTR l'arbre de Janet obtenu par l'insertion de u dans JT
l'ensemble V élargi des prolongations non-multiplicatrices
des monômes de $leaves(jt)$

```
begin
  tree := JT
  i := 1
  τ := sort(leaves(tree) \ {1})
  nd := node(Jt)
  ν := Jt
  while i ≤ n do
    while deg_i(node(ν)) < deg_i(u) and left(ν) do
      ν := left(ν)
    if deg_i(node(ν)) < deg_i(u) then
      V := jprolong(ν,i,V)
      α := lolt(u,i,node(left(ν)))
      τ := sort(τ ∪ {α})
      tree := jtr(τ,nd,1,mult_set(nd))
      i := n + 1
    else
      if left(ν) then
        V := V ∪ {u · x_i}
      if right(ν) then
        i := i + 1
        ν := right(ν)
      else if i < n then
        α := lolt(u,i,node(left(ν)))
        if α ∤ x, ∀x ∈ τ then
          τ := sort(τ ∪ {α})
        tree := jtr(τ,nd,1,mult_set(nd))
        i := n + 1
        else i := i + 1
  return (tree,V)
end
```

3.3.1 Maple

Maple permet d'avoir une "liste de liste de ...", et nous avons eu aucun problème avec une structure de type:

$$arbre = [[noeud],[arbre_{gauche}],[arbre_{droite}]].$$

Nous avons dû faire une fonction booléenne $div(\alpha,\beta)$ qui retourne *true* si le monôme représenté par la liste α divise le monôme représenté par la liste β. Aussi, pour ordonner une liste de listes, nous avons dû faire une fonction supplémentaire, *lsort*, peu efficace mais qui nous a permis de terminer le programme. Une autre problème a été l'absence d'une fonction qui élimine l'élément a d'une liste (pas l'élément d'indice a), ce qui nous a obligé de convertir les *liste* en *set* et les *set* en *liste*.

3.3.2 Aldor

Aldor est un language compilé dont toutes les valeurs (fonctions, constantes, variables, etc) ont des types. Le compilateur ne force jamais un type dans un autre. Pour utiliser les fonctions d'un certain domaine, il faut toujours les importer. Aussi, il faut spécifier si une variable est d'un type ou un autre (par exemple, si l'on a importé de $MachineInteger$ et de $Integer$, il faut dire auquel de ces deux domaines appartient la variable 1). Mais, on peut avoir le même nom pour deux fonctions ayant le même résultat (par exemple, la fonction *tree* peut prendre un seul argument et rétourne l'arbre trivial, ou trois arguments, et rétourne un arbre non-trivial) - ce qui n'est pas possible en Maple.

Premièrement, nous avons construit le type "Tree". On peut associer ce type avec différents autres types, et l'on obtient: "Tree Integer" - pour un arbre des entiers, "Tree List Integer" - pour un arbre qui a comme noeuds des listes d'entiers, etc. les fonctions prédéfinies de ce type sont: "empty" (l'arbre vide), "tree(noeud)" et "tree(noeud, partie$_{\mathrm{gauche}}$, partie$_{\mathrm{droite}}$)" qui construisent un arbre, "empty?" qui décide si l'arbre est vide ou pas, "node(arbre)", "left(arbre)" et "right(arbre)" qui retournent respectivement le noeud, la partie gauche et la partie droite de l'arbre, "nodes(arbre)" et "leaves(arbre)" qui retourne respectivement les noeuds et les feuilles de l'arbre.

Les fonctions supplémentaires à faire ont été:

- *div*, l'équivalent de la fonction *div* que nous avons construit en Maple;
- $size(T : OutputType, m : ListT)$ qui est l'équivalent de $nops(m)$ de Maple;
- $firstind$ - comme la fonction *leaves* donne les feuilles et les variables multiplicatrices associées, $firstind$ *leaves* donne seulement les feuilles de l'arbre;
- $zero(n)$ qui donne la liste nulle avec n éléments;
- $multset(n)$ qui compte l'ensemble des variables multiplicatrices associées à la racine de l'arbre.

En Aldor, on peut facilement sorter une liste des listes , dans l'ordre qui convient, par la fonction $sort!(liste, ordre)$, et c'est le motif principal pour lequel je n'ai pas insisté sur l'efficacité de son équivalent fait en Maple. On peut aussi, par la fonction $remove(e, l)$ éliminer l'élément e de la liste l, et par la fonction $insert(e, l)$ insérer l'élément e de la liste l.

TAB. 3.9 – ***Algorithme:*** *MJBAS* $MJBAS(U)$

Entrées: U un ensemble de monômes
Sorties: la base de Janet minimale de l'idéal $Id(U)$

begin
  $jt := \{1\}$
  $\tau := jtr(jt,1,1,mult\_set(n))$
  $V := U$
  if $1 \in V$ then $ans := \{1\}$
  else $ans := \emptyset$
    while $V \neq \emptyset$ do
      choose $x \in V$ without divisor in $V \setminus \{x\}$
      $V := V \setminus \{x\}$
      if $jdivisor(\tau,x) = nil$ then
        $ans := ans \cup \{x\}$
        $(\tau,V) := jinsert(\tau,x,V)$
  return ans
end

Chapitre 4

Une application des bases de Janet

Le but du *calcul inverse de la dérivation* est de calculer les solutions formelles (analytiques) y au voisinage d'un point $x^0 = (x_1^0, \ldots, x_n^0)$ d'un système d'équations aux dérivées partielles inhomogène de la forme suivante

$$\begin{cases} \partial_\alpha y(x) = f_\alpha(x), \\ \partial_\beta y(x) = f_\beta(x), \\ \vdots \\ \partial_\zeta y(x) = f_\zeta(x), \end{cases} \tag{4.1}$$

où $x = (x_1, \ldots, x_n)$ sont n variables indépendantes, $f_\alpha, \ldots, f_\zeta$ sont des fonctions analytiques réelles (au voisinage d'un point $x^0 = (x_1^0, \ldots, x_n^0)$) fixées,

$$\alpha = (\alpha_1, \ldots, \alpha_n),\, \beta = (\beta_1, \ldots, \beta_n), \ldots,\, \zeta = (\zeta_1, \ldots, \zeta_n) \in \mathbb{Z}_+^n$$

sont des multi-indices et:

$$\partial_\alpha\, y(x) \triangleq \frac{\partial^{\alpha_1+\alpha_2+\ldots+\alpha_n} y(x)}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_n^{\alpha_n}}.$$

Nous associons aux premiers membres de (4.1) l'ensemble fini de monômes:

$$U = \{\alpha, \beta, \ldots, \zeta\}.$$

Considérons l'ordre sur les variables indépendantes $x_n > x_{n-1} > \ldots > x_1$ (dans la suite, il nous arrivera aussi d'utiliser l'ordre $x_1 > x_2 > \ldots > x_n$: cela revient à utiliser l'ordre précédent en permutant les variables x_i). Pour chacun des monômes de U, nous définisons ses variables multiplicatrices pour le classement $x_n > \ldots > x_2 > x_1$, c'est-à-dire x_i est une variable multiplicatrice pour $x_1^{\theta_1} x_2^{\theta_2} \ldots x_n^{\theta_n} \in U$ s'il n'existe aucun monôme de U ayant les degrés θ_n en x_n, θ_{n-1} en x_{n_1}, …, θ_{i+1} en x_{i+1} et un degré strictement supérieur à θ_i en x_i.

Soit $U = \{\alpha, \beta, \ldots, \zeta\}$ un ensemble fini de monômes. Rappelons que l'ensemble des monômes complémentaires $V = \cup_{i=1}^n V^{(i)}$ de U est défini par:

$$\begin{cases} V^{(n)} = \{x_n^k \mid k < \max\{\alpha_n, \beta_n, \ldots, \zeta_n\}, k \notin \{\alpha_n, \beta_n, \ldots, \zeta_n\}\}, \\ \\ V^{(i)} = \{x_n^{\theta_n} \ldots x_{i+1}^{\theta_{i+1}}\, x_i^k \mid \theta = (\theta_1, \ldots, \theta_n) \in U, k < \max\{\alpha_i, \beta_i, \ldots, \zeta_i\}, k \notin \{\alpha_i, \beta_i, \ldots, \zeta_i\}\}. \end{cases}$$

Nous associons à tout monôme de V les variables multiplicatrices suivantes:

- $x_1, \ldots, x_{n-1}$ à tout monôme de $V^{(n)}$,

– $x_1, \ldots, x_{i-1}$ à tout monôme u de $V^{(i)}$ ainsi que celles des variables $x_{i+1}, \ldots, x_n$ qui sont multiplicatrices pour u par rapport à U.

Exemple 4.0.1. Considérons l'ensemble fini de monômes $U = \{x_1^2, x_2^2, x_1\, x_2^2\}$ et l'ordre $x_1 > x_2$. Nous avons les variables multiplicatrices et non-multiplicatrices suivantes:

$$\left\{\begin{array}{l} x_1^2 \\ x_1\, x_2^2 \\ x_2^2 \end{array}\right. \quad \begin{array}{|c|c|} \hline 1 & 2 \\ \hline \bullet & 2 \\ \hline \bullet & 2 \\ \hline \end{array} \tag{4.2}$$

Notons que l'ensemble U est complet car les prolongements de $x_1\, x_2^2$ et de x_2^2 par rapport à leurs variables non-multiplicatrices x_1 (i.e. $x_1^2\, x_2^2$ et $x_1\, x_2^2$) admettent un diviseur involutif dans U (i.e. $x_1^2\, x_2^2 \in C_J(x_1^2)$ et $x_1\, x_2^2 \in C_J(x_1\, x_2^2)$).

L'ensemble complémentaire V de U est défini par $V = V^{(1)} \cup V^{(2)}$, où

$$\begin{aligned} V^{(2)} &= \{x_1^k \,|\, k < \max\{\deg_1(x_1^2) = 2,\ \deg_1(x_1\, x_2^2) = 1,\ \deg_1(x_2^2) = 0\}\} = \emptyset, \\ V^{(1)} &= \{x_1^2\, x_2^k \,|\, k < 0\} \cup \{x_1\, x_2^k \,|\, k < 2\} \cup \{x_2^k \,|\, k < 2\} \\ &= \{x_1\, x_2, x_1\} \cup \{x_2, 1\}. \end{aligned}$$

Nous avons les variables multiplicatrices et non-multiplicatrices de V suivantes:

$$\left\{\begin{array}{c} x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{array}\right. \quad \begin{array}{|cc|} \hline \bullet & \bullet \\ \hline \bullet & \bullet \\ \hline \bullet & \bullet \\ \hline \bullet & \bullet \\ \hline \end{array} \tag{4.3}$$

Lemme 4.0.1. *Soient U un ensemble complet de monômes et V son ensemble complémentaire pour la division de Janet et l'ordre $x_n > x_{n-1} > \ldots > x_1$.*

1. *Nous avons la série formelle suivante:*

$$y(x) = \sum_{\aleph=(\aleph_1,\ldots,\aleph_n)\in\mathbb{M}} a_\aleph\, x^\aleph = \sum_{\theta\in U\cup V} \sum_{\sigma\in C_J(\theta)} a_{\theta+\sigma}\, x^{\theta+\sigma} \tag{4.4}$$

$$= \sum_{\theta\in U\cup V} \left(\sum_{\sigma\in C_J(\theta)} a_{\theta+\sigma}\, x^\sigma \right) x^\theta. \tag{4.5}$$

2. *Pour $\theta \in U \cup V$, nous avons les relations suivantes:*

$$(\partial_\theta y(x))_{\{x_i=0\,|\,x_i\in NM_J(\theta)\}} = \partial_\theta \left(\sum_{\sigma\in C_J(\theta)} a_{\theta+\sigma} x^{\theta+\sigma} \right), \tag{4.6}$$

Démonstration. 1. Puisque U est un ensemble complet de monômes, tout monôme de $\mathbb{M}$ appartient à un unique cône involutif d'un élément de U ou de V. Ainsi, nous obtenons (4.4) et (4.5).

2. Nous avons:

$$\begin{aligned} \partial_\theta\, y(x) &= \partial_\theta \left(\textstyle\sum_{\aleph\in\mathbb{M}} a_\aleph\, x^\aleph\right) = \partial_\theta \left(\textstyle\sum_{\sigma\in C_J(\theta)} a_{\theta+\sigma}\, x^{\theta+\sigma}\right) \\ &= \textstyle\sum_{\sigma\in\mathbb{M}} \left(\prod_{j=1}^n \frac{(\theta_j+\sigma_j)!}{\sigma_j!}\right) a_{\theta+\sigma}\, x^\sigma, \end{aligned}$$

$$\Rightarrow (\partial_\theta\, y(x))_{\{x_i=0\,|\,x_i\in NM_J(\theta)\}} \;= \sum_{\sigma\in C_J(\theta)} \left(\prod_{j=1}^{n} \frac{(\theta_j+\sigma_j)\,!}{\sigma_j\,!}\right) a_{\theta+\sigma}\, x^{\sigma},$$

$$= \partial_\theta\big(\textstyle\sum_{\sigma\in C_J(\theta)} a_{\theta+\sigma}\, x^{\theta+\sigma}\big),$$

d'où, nous obtenons (4.6). □

Théorème 4.0.1. *[18, 23] Soient* $U = \{\alpha, \beta, \ldots, \zeta\}$ *un ensemble complet de monômes et son ensemble complémentaire* $V = \{A, B \ldots, Z\}$ *pour la division de Janet et l'ordre:*

$$x_n > x_{n-1} > \ldots > x_1.$$

Pour tout $\theta \in U$, *considérons une fonction analytique réelle* $f_\theta(x)$ *définie dans un voisinage de* $x^0 = (0, \ldots, 0)$. *Alors, le système linéaire d'EDP*

$$\begin{cases} \partial_\alpha\, y(x) = f_\alpha(x), \\ \partial_\beta\, y(x) = f_\beta(x), \\ \vdots \\ \partial_\zeta\, y(x) = f_\zeta(x), \end{cases} \tag{4.7}$$

muni des conditions initiales suivantes

$$\begin{cases} (\partial_A\, y)(x)_{\{x_i=0\,|\,x_i\in NM_J(A)\}} = g_A(x_j, x_j \in M_J(A)), \\ (\partial_B\, y)(x)_{\{x_i=0\,|\,x_i\in NM_J(B)\}} = g_B(x_j, x_j \in M_J(B)), \\ \vdots \\ (\partial_Z\, y)(x)_{\{x_i=0\,|\,x_i\in NM_J(Z)\}} = g_Z(x_j, x_j \in M_J(Z)), \end{cases} \tag{4.8}$$

où $g_A, g_B, \ldots, g_Z$ *sont des fonctions analytiques réelles au voisinage de* $x^0 = (0, \ldots, 0)$, *admet une solution analytique réelle* y *dans un voisinage de* $x^0 = (0, \ldots, 0)$ *si et seulement si les conditions de compatibilités suivantes sont verifiées*

$$\partial_i\, f_\theta = \partial_\aleph f_\rho, \quad \forall\, \theta \in U, \forall\, x_i \in NM_J(\theta), \tag{4.9}$$

traduisant le fait que $x_i\, \theta \in C_J(\rho)$ *avec* $\rho \in U$, *c'est-à-dire la relation:*

$$x_i\, \theta = \aleph\, \rho, \quad \aleph \in L(\rho,U).$$

La solution y *de (4.7) et (4.8) est définie localement dans un voisinage de* $x^0 = (0, \ldots ,0)$ *par:*

$$\begin{aligned} y(x) \;&= \sum_{\theta\in U} \left(\left(\prod_{\{j=1,\ldots,n\,|\,\theta_j\neq 0\}} \int_0^{x_j} \frac{(x_j-\xi_j)^{\theta_j-1}}{(\theta_j-1)\,!}\, d\xi_j\right) \big(f_\theta(x_j, x_j \in M_J(\theta))\big)\right) \\ &+ \sum_{T\in V} \left(\left(\prod_{\{j=1,\ldots,n\,|\,T_j\neq 0\}} \int_0^{x_j} \frac{(x_j-\xi_j)^{T_j-1}}{(T_j-1)\,!}\, d\xi_j\right) \big(g_T(x_j, x_j \in M_J(T))\big)\right). \end{aligned} \tag{4.10}$$

Exemple 4.0.1. Considérons le système d'EDP suivant:

$$\begin{cases} \partial_1^2\, y(x) = f_1(x) & x_1^2 \\ \partial_2^2\, y(x) = f_2(x) & x_2^2 \end{cases} \quad \begin{array}{|cc|} \hline x_1 & x_2 \\ \hline \bullet & x_2 \\ \hline \end{array} \tag{4.11}$$

Déterminons le développement en série au voisinage de $x_0 = (0, 0)$ d'une solution générale de (4.11). Pour cela, nous devons déterminer le degré de généralité de (4.11).

L'ensemble des monômes $U = \{x_1^2, x_2^2\}$ correspondant aux derivées de tête ne forment pas un ensemble complet (voir l'Exemple 1.2.1 de l'introduction). Nous complétons le système en lui ajoutant la nouvelle équation

$$\partial_1\, \partial_2^2\, y(x) = \partial_1 f_2(x)$$

correspondant à la prolongation non-multiplicatrice $x_1\, x_2^2$ du monôme x_2^2. Nous obtenons le nouveau système suivant:

$$\left\{\begin{array}{lll} \partial_1^2\, y(x) = f_1(x) & x_1^2 & \boxed{x_1 \quad x_2} \\ \partial_2^2\, y(x) = f_2(x) & x_2^2 & \boxed{\bullet \quad x_2} \\ \partial_1\, \partial_2^2\, y(x) = \partial_1\, f_2(x) & x_1 x_2^2 & \boxed{\bullet \quad x_2} \end{array}\right. \tag{4.12}$$

Dans l'Exemple 4.0.1, nous avons déjà calculer l'ensemble (4.13) des monômes complémentaires de $\{x_1^2, x_2^2, x_1, x_2^2\}$:

$$\left\{\begin{array}{cc} x_1\, x_2 & \boxed{\bullet \quad \bullet} \\ x_1 & \boxed{\bullet \quad \bullet} \\ x_2 & \boxed{\bullet \quad \bullet} \\ 1 & \boxed{\bullet \quad \bullet} \end{array}\right. \tag{4.13}$$

Alors, nous avons le développement de y au voisinage de $x^0 = (0{,}0)$ suivant

$$\begin{aligned} y(x) = & \sum_{k,l\geq 0} a_{k+2,l}\, x_1^{k+2}\, x_2^l + \sum_{l\geq 0} a_{1,l+2}\, x_1\, x_2^{l+2} + \sum_{l\geq 0} a_{0,l+2}\, x_2^{l+2} + \\ & + a_{12}\, x_1\, x_2 + a_{10}\, x_1 + a_{01}\, x_2 + a_{00} \end{aligned} \tag{4.14}$$

ou, si l'on note

$$\left\{\begin{array}{ll} b_1(x_1{,}x_2) & = \displaystyle\sum_{k,l\geq 0} a_{k+2,l} x_1^k\, x_2^l, \\ b_2(x_2) & = \displaystyle\sum_{l\geq 0} a_{1,l+2}\, x_2^l, \\ b_3(x_2) & = \displaystyle\sum_{l\geq 0} a_{0,l+2}\, x_2^l, \end{array}\right.$$

on obtient:

$$y(x) = b_1(x_1{,}x_2)\, x_1^2 + b_2(x_2)\, x_1\, x_2^2 + b_3(x_2)\, x_2^2 + a_{12}\, x_1\, x_2 + a_{10}\, x_1 + a_{01}\, x_2 + a_{00}. \tag{4.15}$$

En dérivant dans le système (4.12) par rapport aux variables non-multiplicatrices, nous obtenons $x_1^2\ x_2^2 = x_2^2\,(x_1^2)$, ce qui correspond à la condition de compatibilité suivante:

$$\partial_1^2 f_2(x) = \partial_2^2 f_1(x). \tag{4.16}$$

Donc, si la condition (4.16) est satisfaite, le système (4.11) admet une unique solution au voisinage de $x^0 = (0{,}0)$ qui satisfait les conditions initiales déduites de (4.13), c'est-à-dire:

$$\left\{\begin{array}{ll} y(x)_{x_1=x_2=0} & = \alpha, \\ \partial_1\, y(x)_{x_1=x_2=0} & = \beta, \\ \partial_2\, y(x)_{x_1=x_2=0} & = \gamma, \\ \partial_1\, \partial_2\, y(x)_{x_1=x_2=0} & = \delta. \end{array}\right. \tag{4.17}$$

Donc, le système (4.11) dévient:

$$\begin{cases} \partial_1^2\, y(x) & = f_1(x), \\ \partial_2^2\, y(x) & = f_2(x), \\ y(0{,}0) & = \alpha, \\ (\partial_1\, y)(0{,}0) & = \beta, \\ (\partial_2\, y)(0{,}0) & = \gamma, \\ (\partial_1\, \partial_2\, y)(0{,}0) & = \delta. \end{cases} \tag{4.18}$$

La solution du système (4.18) est définie au voisinage de (0,0) par:

$$y = \int_0^{x_1} \frac{(x_1-\xi_1)^1}{1!}\, f_1(\xi_1,\xi_2)\, d\xi_1 + \int_0^{x_2} \frac{(x_2-\xi_2)^1}{1!}\, f_2(0{,}\xi_2)\, d\xi_2 + \alpha + \int_0^{x_1} \frac{(x_1-\xi_1)^0}{0!}\, \beta\, d\xi_1 +$$

$$\int_0^{x_2} \frac{(x_2-\xi_2)^0}{0!}\, \gamma\, d\xi_2 + \int_0^{x_1}\int_0^{x_2} \frac{(x_1-\xi_1)^0}{0!}\, \frac{(x_2-\xi_2)^0}{0!}\, \delta\, d\xi_1\, d\xi_2.$$

En effectuant les calculs, nous obtenons finalement:

$$y(x) = \int_0^{x_1} (x_1-\xi_1)\, f_1(\xi_1,\xi_2)\, d\xi_1 + \int_0^{x_2} (x_2-\xi_2)\, f_2(0{,}\xi_2)\, d\xi_2 + \alpha + \beta\, x_1 + \gamma\, x_2 + \delta\, x_1\, x_2 \tag{4.19}$$

Chapitre 5

Programmes en Maple et Aldor

5.1 Programme Maple

Nous donnons le programme en Maple qui réalise une complétion involutive pour la division de Janet. Nous avons travaillé avec des listes.

Les fonctions **leftallowed** et **rightallowed** testent s'il peut y avoir un *ndg* ou un *nvr*.

```
leftallowed := proc(n, lst, i)
   local ans, leaf;
   ans:=true;
   if member(n,lst) or lst=[] or i>nops(n) then ans:=false
   else leaf:=lst[1];
      if n[i]>leaf[i] or ( i<nops(n) and n[i]=leaf[i] and n[i+1]<=leaf[i+1])
       then ans:=ans and false fi; fi;
   ans; end:

rightallowed := proc(n, lst, i)
   local ans;
   ans:=true;
   if member(n,lst) or lst=[] or i>nops(n) then ans:=false
   else
      if not((not leftallowed(n,lst,i) and div(n, lst[1])) or
       (nops(lst)>=2 and leftallowed(n,lst,i) and div(n,lst[2])))
       then ans:=ans and false; fi; fi;
   ans; end:
```

La fonction **nodes** détermine tous les noeux d'un arbre de Janet.

```
nodes := proc(tr, l)
   local n,p,left,right;
   p:=l;
   if tr[1][1]=[] then p:=[]
   else
      n:=tr[1]; left:=tr[2]; right:=tr[3]; p:=[op(p),n[1]];
      if left<>[] then p:=nodes(left,p) fi;
      if right<>[] then p:=nodes(right,p) fi; fi;
   p; end:
```

La fonction **leaves** (resp. **multleaves**) détermine les feuilles d'un arbre (resp. les feuilles et les variables multiplicatrices associées).

```
leaves := proc(tr, l)
   local n,p,left,right;
   p:=l;
   if tr<>[] then n:=tr[1]; left:=tr[2]; right:=tr[3];
   if (tr[2]=[] and tr[3]=[]) then p:=[op(p),n[1]]; fi;
   if left<>[] then p:=leaves(left,p) fi;
   if right<>[] then p:=leaves(right,p) fi; fi;
   p; end:

multleaves:=proc(tr, l)
   local n,p,left,right;
   p:=l;
   if tr<>[] then n:=tr[1]; left:=tr[2]; right:=tr[3];
   if (tr[2]=[] and tr[3]=[]) then p:=[op(p),n]; fi;
   if left<>[] then p:=multleaves(left,p) fi;
   if right<>[] then p:=multleaves(right,p) fi; fi;
   p; end:
```

La fonction **multset** donne la liste des variables multiplicatrices associées à la racine d'un arbre de Janet.

```
multset:=proc(n)
   local i,p;
   p:=[];
   for i from 1 to n do p:=[op(p), i] od;
   p; end:
```

On construit l'arbre de Janet par la fonction **jtr** qui prend comme arguments une liste de monômes, la racine de l'arbre, l'indice de la plus grande variable (dans tous ces cas, on considère $ii = 1$), l'arbre vide et la liste des variables multiplicatrices associées à la racine.

```
jtr := proc (lst, nod, ii, t, sst)
   local n, l, left, right, tr, i,st;
   st:=sst; tr := t; i := ii; n := [nod,st];
   l := lsort(lst);
   if member(n[1],l) then
      tr := [n, [], []]; l := subsop(1 = NULL,l)
   else
      if leftallowed(n[1],l,i) then left := jtr(l,ndg(n[1],i),i,tr,st)
      else left := [] fi;
      l := lsort(convert((convert(l,set) minus
       convert(leaves(left,[]),set)),list));
      if l <> [] and rightallowed(n[1],l,i) then
         if left<>[] then  st:=subsop(1=NULL, st); fi;
         i := i+1;
         right := jtr(l,n[1],i,tr,st);
      else right := [] fi;
      tr := [n, left, right]; fi;
   tr end:
```

la fonction **jdivisor** implémente l'algorithme *jdivisor*.

```
jdivisor := proc(jt, w)
   local nu, ans, i;
   nu := jt; i := 1;
   if leaves(jt,[])=nodes(jt,[]) then ans:=[]
   else
      while div(nu[1][1], w) and i<=nops(w) do
         while nops(nu[2])>0 and div(nu[2][1][1], w) do  nu := nu[2]; od;
         if nops(nu[3])>0 then nu := nu[3];
         elif nops(nu[2])>0 then ans:=[]
         else ans := nu[1] fi;
         i:=i+1; od; fi;
   ans; end:
```

La fonction **olt** implémente l'algorithme $OneLeafTree$, et la fonction **lolt** calcule la feuille de cet arbre (plus rapide que de calculer $t := olt(u,i,w)$ et puis de revenir avec $leaves(t,[])$ ou $multleaves(t,[])$)

```
olt := proc(u, i, w)
   local p,k,tr,j;
   p:=[];
   for k from 1 to nops(u) do
      if k<i then p:=[op(p), w[k]];
      else p:=[op(p), u[k]]; fi od;
   tr:=[p,[],[]]; j:=nops(u);
   while j>=i do
      while up(p,j)[j]>=w[j] do
         tr:=[up(p,j),tr,[]];
         p:=up(p,j) od;
      if up(p,j)[j]<w[j] then tr:=[p,[],tr] fi;
      j:=j-1; od;
   tr; end:

lolt := proc(u, i, w)
   local p,k,tr,j; p:=[];
   for k from 1 to nops(u) do
      if k<i then p:=[op(p), w[k]];
      else p:=[op(p), u[k]]; fi od;
   p; end:
```

La fonction **jprolong** réalise l'algorithme *jprolong*.

```
jprolong := proc(jt, ii, VV)
   local nu,i,V;
   V:=VV; i:=ii; nu:=jt;
   if leaves(nu,[])<>nodes(nu,[]) then
      while nops(nu[2])>0 do
         if nops(nu[3])>0 then V:=jprolong(nu[3], i, V) fi;
         nu:=nu[2]; od;
      if nops(nu[3])>0 then V:=jprolong(nu[3], i, V)
      else V:=[op(V), ndg(nu[1][1], i)]; fi; fi;
   V; end:
```

La fonction **jinsert** réalise l'algorithme *jinsert*.

```
jinsert := proc(jt, u, VV)
   local ans, rep, V,i, nu, JTree, nod, alpha, tree;
   V:=VV; tree:=jt; i:=1;
   JTree:=lsort(leaves(jt,[]));
   JTree:=convert((convert(JTree,set) minus {zero(nops(u))}),list);
   nod:=jt[1]; nu:=jt;
   while i<=nops(u) do
      while nu[1][1][i]<u[i] and nops(nu[2])>0 do nu:=nu[2]; od;
      if nu[1][1][i]<u[i] then
         V:=jprolong(nu,i,V);
         alpha:=lolt(u,i,ndg(nu[1][1],i));
         JTree:=lsort([op(JTree),alpha]);
         tree:=jtr(JTree, nod[1], 1, [], nod[2]);
         i:=nops(u)+1
      else
         if nu[2]<>[] then V:=[op(V), ndg(u,i)] fi;
         if nops(nu[3])<>0 then
            i:=i+1;
            nu:=nu[3];
            else if i<nops(u) then
               alpha:=lolt(u,i+1,nu[1][1]);
               rep:=true;
               for i from 1 to nops(JTree) do
                  if member(alpha, nodes(arbre(JTree,[]),[])) then
                     rep:= rep and false fi; od;
               if rep then
                  JTree:=lsort([op(JTree),alpha]) fi;
                  tree:=jtr(JTree, nod[1], 1, [], nod[2]);
                  i:=nops(u)+1
               else i:=i+1; fi; fi; fi; od;
   ans:=(tree, V);
   ans; end:
```

On a besoin de choisir un élément x de l'ensemble ordonné vv qui n'a pas de diviseurs dans l'ensemble $vv \setminus \{x\}$. Le choix le plus rapide est réalisé dans la fonction **choisir** qui est le plus petit élément de vv.

```
choisir := proc(vv)
   local v, n, ans;
   v := lsort(convert(vv,set)); n := nops(v); ans := v[n];
   ans; end:
```

La completion involutive est réalisé dans la fonction **mjbas**.

```
mjbas := proc (u)
   local n,jt,JTR,v,ans,x,ll;
   n:=nops(u[1]); jt:=[zero(n)];
   JTR:=jtr(jt, zero(n), 1, [], multset(n));
   v:=u;
   if member(zero(n),v) then ans:={zero(n)}
   else
```

```
      ans:={};
      while v<>{} do
         x:=choisir(v);
         v:=convert(v,set) minus {x};
         if jdivisor(JTR,x)=[] then
            ans:=ans union {x};
            (JTR,v):=jinsert(JTR, x, convert(v,list));
            v:=convert(v,set); fi; od; fi;
   ans; end:
```

5.2 Programme Aldor

D'abord, nous avons construit un type "Tree"

```
Tree(T:OutputType):OutputType with {
   export from T;
   empty: %;
   tree: T -> %;
   tree: (T, %, %) -> %;
   empty?: % -> Boolean;
   node: % -> T;
   left: % -> %;
   right: % -> %;
   nodes: % -> List T;
   leaves: % -> List T;
}
==
add { }
```

Puis, nous avons construit l'arbre de Janet, en travaillant avec des listes des multi-indices. Nous avons eu besoin d'une fonction supplémentaire **size** qui donne la dimension d'une liste, d'une fonction **divide** et qui retourne *true* si le monôme répresenté par la liste u divise le monôme répresenté par la liste v, d'une fonction **lroot** qui calcule le ndg de mm en k, et de les deux fonctions-test **leftallowed** et **rightallowed**. La fonction **firstind** retourne la liste des monômes (on l'utilise avec les fonctions *leaves* et *nodes* qui retournent la liste des monômes et des variables multiplicatrices associées).

```
size(T:OutputType, m:List T):MachineInteger == {
   local r:List T; local i:MachineInteger;
   r:=m; i:=0;
   while not empty? r repeat {
      r:=rest r; i:=i+1;}
   i;
}

divide(u:List Integer, v:List Integer):Boolean == {
   import from Integer;
   l1:=u; l2:=v; ans:=true;
   if empty? l1 or empty? l2 then ans := false;
   while not empty? l1 and not empty? l2 repeat {
      if first l1 > first l2 then p:=false else p:=true;
```

```
      ans:=ans and p;
      l1:=rest l1; l2:=rest l2;
      }
   ans;
}

lroot(mm:List Integer, k:MachineInteger):List Integer == {
   r:=mm; lr:List Integer:= empty; i:MachineInteger:=1;
   while not empty? r repeat {
      p:Integer:=first r;
      if i=k then p:=p+1;
      lr:=append!(lr,p);
      r:=rest r; i:=i+1;}
   lr;
}

leftallowed(n:List Integer, ll:List List Integer, i:MachineInteger)
 :Boolean == {
   import from MachineInteger, Integer, List Integer, Boolean;
   ans := true;
   if member?(n, ll) or empty? ll then ans := false else {
      leaf := first ll;
      if (n.i>leaf.i) or (n.i=leaf.i and n.(i+1)<=leaf.(i+1))
       then ans := ans and false;
}
   ans;
}

rightallowed(n:List Integer, ll:List List Integer, i:MachineInteger)
 :Boolean == {
   import from MachineInteger, Boolean;
   ans := true;
   if member?(n, ll) or empty? ll then ans := false
   else
      if not((not leftallowed(n, ll, i) and divide(n, first ll)) or
       (not empty? rest ll and leftallowed(n, ll, i) and
       divide(n, first rest ll))) then ans := ans and false;
   ans;
}

firstind(x:List List List Integer):List List Integer == {
   import from List List Integer, List List List Integer;
   local l:List List Integer;
   local ll:List List List Integer;
   ll:=x; l:=empty;
   while not empty? ll repeat {
      l:=append!(l, first first(ll));
      ll:=rest ll;
   }
```

```
   l;
}

zero(n:MachineInteger):List Integer == {
   import from MachineInteger, Integer, List Integer;
   local i:MachineInteger:=1;
   local p:List Integer := empty;
   while i<=n repeat { p:=append!(p,0); i:=i+1}
   p;
}

multset(n:MachineInteger):List Integer == {
   import from MachineInteger, Integer, List Integer;
   local i:Integer:=1; local j:MachineInteger:=1;
   local p:List Integer := empty;
   while j<=n repeat { p:=append!(p,i); i:=i+1; j:=j+1;}
   p;
}

jtr(ll:List List Integer, nn:List Integer, ii:MachineInteger,
 t:Tree List List Integer,st:List Integer):Tree List List Integer == {
   import from MachineInteger, Tree List Integer, List List Integer;
   local i:MachineInteger;
   local s: List Integer; local n:List List Integer;
   local lft:Tree List List Integer; local rgt:Tree List List Integer;
   l := sort!(ll, >=);
   tr:=t; s:=st; i:=ii; n:=[nn, s];
   if member?(n.1,l) then {
      tr:=tree(n);
      l:=rest(l);
   }
   else {
      if leftallowed(n.1,l,i) then
         lft := jtr(l, lroot(n.1,i), i, tr,s)
      else lft := empty;
      if not empty? l and rightallowed(n.1,l,i) then {
         if leftallowed(n.1,l,i) then s:=remove(i,s);
         while not empty? l and not empty? lft and
          member?(first l, firstind leaves(lft)) repeat l := rest l;
         i := i+1;
         rgt := jtr(l,n.1,i,tr,s);
      }
      else rgt := empty;
      tr := tree(n, lft, rgt);
   }
   tr;
}
```

Les fonctions **jdivisor** et **lolt** sont utilisées dans la fonction **jprolong**.

```
jdivisor(jt:Tree List List Integer, w:List Integer):List List Integer == {
```

```
   import from Tree List List Integer,List Integer,MachineInteger,
     List List List Integer;
   local nu: Tree List List Integer;
   local ans: List List Integer;
   local i: MachineInteger;
   nu :=jt; i:=1;
   if leaves jt= nodes jt then ans:=empty
   else {
      while divide(first node nu, w) and i<= size(Integer, w) repeat {
         while not empty? left nu and divide(first node left nu, w)
          repeat nu:=left nu;
         if not empty? right nu then nu := right nu
         else {
            if not empty? left nu then ans := empty
            else ans:= node nu;
            i := i+1;
         }
      }
   }
   ans;
}

lolt(u:List Integer, i:MachineInteger, w:List Integer):List Integer =={
   import from List Integer, MachineInteger;
   local p: List Integer;
   local k:MachineInteger;
   p := empty; k := 1;
   while k<=size(Integer, u) repeat {
      if k<i then p:=append!(p,w.k) else p:=append!(p,u.k);
      k := k+1;
   }
   p;
}

jprolong(jt:Tree List List Integer, ii:MachineInteger,
 VV:List List Integer):List List Integer =={
   import from Tree List List Integer, MachineInteger,
    List List List Integer;
   local V: List List Integer;
   local i:  MachineInteger;
   local nu: Tree List List Integer;
   V := VV; i := ii; nu := jt;
   if leaves nu ~= nodes nu then {
      while not empty? left nu repeat {
         if not empty? right nu then V := jprolong(right nu, i, V);
         nu := left nu;
      }
      if not empty? right nu then V := jprolong(right nu, i, V)
      else V := insert(lroot(first node nu,i),V);
```

```
   }
   V;
}
```

La fonction **choisir** calcule non seulement le plus petit élément d'une liste mais retourne aussi la liste privé de cet élément.

```
jinsert(jt:Tree List List Integer, u: List Integer, VV:List List Integer)
 :(Tree List List Integer, List List Integer) =={
   import from List List Integer, Tree List List Integer, MachineInteger,
    List  List List Integer, Integer;
   local V: List List Integer;  local Jtree: List List Integer;
   local tree:Tree List List Integer;
   local i:MachineInteger;
   V := VV; tree := jt; i := 1;
   Jtree := sort!(firstind leaves jt,>=);
   Jtree := remove(zero(size(Integer, u)),Jtree);
   nod := node jt; nu := jt; i:=1;
   while i<=size(Integer, u) repeat {
      while first(node nu).i < u.i and not empty? left nu repeat
         nu := left nu;
      if first(node nu).i < u.i then {
         V := jprolong(nu,i,V);
         alpha := lolt(u,i,lroot(first node nu,i));
         Jtree := sort!(insert(alpha,Jtree),>=);
         tree:= jtr(Jtree, first nod, 1, empty, nod.2);
         i := size(Integer, u) + 1;
      }
      else {
         if not empty? left nu then V := append!(V, lroot(u,i));
         if not empty? right nu then {
            i:=i+1; nu:=right nu;
         }
         else {
            if i<size(Integer, u) then {
               alpha := lolt(u,i+1,first node nu);
               if not member?(alpha, firstind nodes
                (jtr(Jtree, nod.1, 1, empty, nod.2 )))
                then Jtree := sort!(insert(alpha,Jtree));
               tree := jtr(Jtree,first nod, 1, empty, nod.2);
               i := size(Integer, u) + 1;
            }
            else i := i+1;
         }
      }
   }
   (tree,V);
}

choisir(vv:List List Integer):(List Integer, List List Integer) == {
   import from List List Integer, List Integer;
```

```
   local v: List List Integer;
   v := vv; v := sort!(v, <=);
   (first v, rest v);
}

mjbas(u:List List Integer):List List Integer == {
   import from List List Integer, MachineInteger, List Integer;
   local ans: List List Integer; local v: List List Integer;
   local JTR: Tree List List Integer;
   local n: MachineInteger; local x: List Integer;
   n := size(Integer, first u); jt := [zero n];
   JTR := jtr(jt, zero(n), 1, empty, multset(n));
   v := u;
   if member?(zero n, v) then ans := [zero n]
   else {
      ans := empty;
      while not empty? v repeat {
         (x,v) := choisir v;
         if jdivisor(JTR,x)=empty then {
            ans := insert(x,ans);
            (JTR,v) := jinsert(JTR,x,v);
         }
      }
   }
   ans;
}
```

Chapitre 6

Conclusion

Nous avons implémenté les algorithmes de calcul d'une base monomiale de Janet dans Maple et Aldor Généralement, le temps d'éxécution de même algorithme *mjbas* est 10 fois plus petit en Aldor qu'en Maple. Les algorithmes utilisés ont été proposés par V. Gerdt, Y. Blinkov et D. Yanovich [10, 11]. Pour des motifs d'implémentation, nous avons modifié quelques parties. Par example, tant que V. Gerdt, Y. Blinkov et D. Yanovich utilisent dans *jinsert* une réunion des arbres, nous avons reconstruit l'arbre à partir des feuilles des arbres, ce qui rend inutile l'algorithme *OneLeafTree* et n'utilise que la fonction *lolt*. La base monomiale que nous avons calculé ne donne pas les variables multiplicatrices des monômes, mais le résultat peut être obtenu en appliquant successivement les fonctions *jtr* et *leaves* (respectivement *multleaves* en Maple) au résultat de la fonction *mjbas*.

Dans le temps qui me reste, nous allons implémenter en Aldor les algorithmes pour calculer une base polynomiale et paramétriser le programme pour spécifier la division utilisée (Janet ou Pommaret).

Bibliographie

[1] Blinkov, Y. A, Cid, C. F., Gerdt, V. P., Plesken, W., Robertz, D. "The MAPLE package "Janet": I. Polynomial Systems", soumis à CASC 2003.

[2] Blinkov, Y. A, Cid, C. F., Gerdt, V. P., Plesken, W., Robertz, D. "The MAPLE package "Janet": II. Linear partial differential equations", soumis à CASC 2003.

[3] Broadbery, P., Bronstein, M. *A first course in Aldor with libaldor.*

[4] Bronstein, M., Moreno Maza, M. (2002). *Algebra User Guide and Reference Manual Version 1.0.1* - Juin 2001.

[5] Buchberger, B. (1985). "An algorithmic method in polynomial ideal theory", dans N. Bose ed. *Multidimensional Systems Theory*, D. Reidel, 1985, pp. 184-232.

[6] Cartan, E. (1945). *Les systèmes différentiels extérieurs et leurs applications géométriques*, Hermann.

[7] Einstein, A. (1953). *The Meaning of Relativity*, Princeton, 1988.

[8] Gerdt, V. P., Blinkov, Y. A. (1998). "Involutive bases of polynomial ideals", *Mathematics and Computers in Simulation*, vol. 45, pp. 519-541.

[9] Gerdt, V.P., Beth, M., Czichowski, G. "Involutive divisions in *Mathematica*: implementation and some applications", http://invo.jinr.ru/papers.phtml.

[10] Gerdt, V.P., Blinkov, Yu. A., Yanovich, D.A. (2001). "Construction of Janet Bases. I. Monomial Bases" dans *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, pp.233-247.

[11] Gerdt, V.P., Blinkov, Yu.A., Yanovich, D.A. (2001). "Construction of Janet Bases. II. Polynomial Bases" dans *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, pp.249-263.

[12] Greul, G.-M., Pfister G. (2002). *A Singular Introduction to Commutative Algebra*, Springer.

[13] Goldschmidt, H. (1967). "Existence theorems for analytic linear partial differential equations", *Annals of Math.*, Vol. 86 , pp. 246-270

[14] Goldschmidt, H. (1968). "Prolongation of linear partial differential equations: I a conjecture of Elie Cartan, II inhomogeneous equations", *Ann. Sci. Ecole. Norm. Sup.*, Vol. 1 , pp. 417-444, pp. 617-625.

[15] Goldschmidt, H. (1969). "Integrability criterion for systems of nonlinear partial differential equations", *J. Differential Geom.*, Vol. 1, pp. 269-307.

[16] Janet, M. (1920). "Sur les systèmes aux dérivées partielles", *Journal de Math.*, 8$^{\text{ème}}$ série, III, pp. 65-151.

[17] Janet, M. (1924). "Les modules de formes algébriques et la théorie générale des systèmes différentiels", Ann. Ec. Normale, 3$^{\text{ème}}$ série, 41, pp. 27-65.

[18] Janet, M. (1929). *Leçons sur les systèmes d'équations aux dérivées partielles*, Cahiers Scientifiques IV, Gauthier-villars.

[19] Kolchin, E. R. (1973). *Differential Algebra and Algebraic Groups*, Pure Appl. Math., No. 54, Academic Press.

[20] Char B. et al. (1991). *MapleV Language Reference Manual*, Springer Verlag.

[21] Pommaret, J. F. (1978). *Systems of Partial Differential Equations and Lie Pseudo-groups*, Gordon and Breach.

[22] Pommaret, J. F. (1994). *Partial Differential Equations and Group Theory: New Perspectives for Applications*, Kluwer.

[23] Quadrat, A. (2002). "Systems of partial differential equations: an introduction to the works of C. Riquier and M. Janet", notes de cours du *groupe de travail Involution*, INRIA Sophia Antipolis.

[24] Riquier, C. (1910). *Les systèmes d'équations aux dérivées partielles*, Gauthier-Villars.

[25] Riquier, C. (1928). *La méthode des fonctions majorantes et les systèmes d'équations aux dérivées partielles*, Mémorial Sci. Math. XXXII, Gauthier-Villars.

[26] Ritt, J. F. (1950). *Differential Algebra*, AMS Colloq. Publ., Vol. 33.

[27] Hausdorf, M., Seiler, M. (2002). "Involutive Bases in MuPAD - Part I: Involutive Divisions", mathPad 11/1, pp.51-56.

[28] Schwarz, F. (1998). "Janet Bases for Symmetry Groups", dans *Groebner Bases and Applications*, London Mathematical Society Lecture Notes 251, pp.221-234.

[29] Spencer, D. C. (1965). "Overdetermined Systems of Partial Differential Equations", *Bull. Amer. Math. Soc.*, Vol. 75, pp. 1-114.

[30] Watt, S. M., Broadbery, P. A, Iglio, P. Morrison, S. C., Steinbach, J. M., Sutor, R. S. *Aldor Compiler User Guide*, www.aldor.org.