Computing Canonical Representatives of Regular Differential Ideals

François Boulier, François Lemaire

▶ To cite this version:

François Boulier, François Lemaire. Computing Canonical Representatives of Regular Differential Ideals. International Symposium on Symbolic and Algebraic Computation, 2000, France. pp.37-46, 10.1145/345542.345571 . hal-00139177

HAL Id: hal-00139177
https://hal.science/hal-00139177

Submitted on 30 Mar 2007

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

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

COMPUTING CANONICAL REPRESENTATIVES OF REGULAR DIFFERENTIAL IDEALS

François Boulier
Université Lille I, LIFL
59655 Villeneuve d'Ascq CEDEX
France
boulier@lifl.fr

François Lemaire
Université Lille I, LIFL
59655 Villeneuve d'Ascq CEDEX
France
lemaire@lifl.fr

ABSTRACT

In this paper, we give three theoretical and practical contributions for solving polynomial ODE or PDE systems. The first one is practical: an algorithm which improves the purely algebraic part of Rosenfeld–Gröbner (the polynomial ODE or PDE systems simplifier which is the core of the Maple 5.5 diffalg package). It is a variant of lextriangular but does not need any Gröbner basis computation. The second one is theoretical: a characterization of the output of Rosenfeld–Gröbner and a clarification of the existing relationship between algebraic and differential characteristic sets. The third one is theoretical as well as practical: an algorithm to compute canonical representatives of differential polynomials modulo regular differential ideals without any use of Gröbner bases. This algorithm simplifies the theory (somehow a "pedagogic" contribution) but permits us also to perform easily linear algebra over the base field in the factor differential ring defined by a regular differential ideal.

Keywords: differential algebra, Rosenfeld–Gröbner, canonical representatives, lextriangular, regular differential ideal, characteristic sets, characteristic presentations.

1. INTRODUCTION

We make precise in next sections some of the terms used in this introduction.

Rosenfeld–Gröbner [5, 7, 8] is a simplifier for systems of polynomial differential equations (ordinary or with partial derivatives). It solves a theoretical problem of differential algebra (deciding membership in the radical of finitely generated differential ideals) and furnishes as a byproduct tools to solve systems of polynomial ODE and PDE. The situation is similar to that of the Buchberger algorithm [10], which solves a theoretical problem of commutative algebra (deciding membership in polynomial ideals) and furnishes as a byproduct tools to solve systems of polynomial equations. It is the heart of the diffalg package which is part of Maple 5.5 standard library.

Technically, given a system Σ of differential polynomials, Rosenfeld–Gröbner outputs finitely many *characteristic presentations* $C_1, \ldots, C_t$ which are particular cases of triangular systems. Each characteristic presentation C_i represents a differential ideal, denoted $[C_i] : H_{C_i}^{\infty}$. The radical $\sqrt{[\Sigma]}$ of the differential ideal generated by Σ is the intersection

$$\sqrt{[\Sigma]} = [C_1] : H_{C_1}^{\infty} \cap \cdots \cap [C_t] : H_{C_t}^{\infty}.$$

Roughly speaking, Rosenfeld–Gröbner consists in two steps.

The *first step* is the differential step which transforms Σ as finitely many systems $A = 0,\ S \neq 0$ of equations and inequations, called *regular differential systems.*

$$\sqrt{[\Sigma]} = [A_1] : S_1^{\infty} \cap \cdots \cap [A_{t'}] : S_{t'}^{\infty}.$$

These systems satisfy the hypotheses of the key Rosenfeld's lemma which "reduces differential problems to purely algebraic ones". Each regular differential system $A = 0,\ S \neq 0$ represents a *regular differential ideal* denoted $[A] : S^{\infty}$. Its set of equations A is triangular but A is not necessary a characteristic presentation of $[A] : S^{\infty}$.

The *second step* is the purely algebraic step which transforms a regular differential system $A = 0,\ S \neq 0$ into finitely many characteristic presentations $C_1, \ldots, C_{t''}$ satisfying

$$[A] : S^{\infty} = [C_1] : H_{C_1}^{\infty} \cap \cdots \cap [C_{t''}] : H_{C_{t''}}^{\infty}.$$

It is applied separately over each regular differential system produced by the first step.

The fact that computing characteristic presentations from regular differential systems is a purely zerodimensional algebraic problem is only claimed in [8, page 35] with no proofs. The first complete proofs of that claim are given in [15].

The first result of this paper is an efficient algorithm, called regCharacteristic *which performs the second step without computing any Gröbner basis.*

The regCharacteristic algorithm first reduces the problem to a dimension zero problem[1]. It puts in the coefficient field all the derivatives occuring in A and S which are not leaders of any element of A, giving sets $\overline{A}$ and $\overline{S}$. It then calls the

[1] As [8, 15] already do.

subalgorithm satTriangular which computes a set of normalized triangular sets $\{T_1, \ldots, T_t\}$ which (nearly immediately) give the desired characteristic presentations $\{C_1, \ldots, C_t\}$.

The satTriangular subalgorithm is a very close variant of the lextriangular algorithm [20, page 129, algorithm D5lextriangular] [24] and [25, page 133] which applies the D5 [11] process. Let's quote [20, page 129]: lextriangular is given a Gröbner basis B of a zero–dimensional ideal of polynomials in $X_1, \ldots, X_n$, sorted by increasing leading monomials for the lexicographical ordering such that $X_1 < \cdots < X_n$ and produces on output a finite family of normalized triangular sets $T_1, \ldots, T_t$ such that

$$V(B) \quad = \quad V(T_1) \cup \cdots \cup V(T_t) \qquad (1)$$

where $V(T_i)$ denotes the set of the common zeroes of the elements of T_i in the algebraic closure of the base field.

Our implementation of satTriangular is directly inspired from [24, 25] (we do not claim any algorithmic improvement w.r.t. these papers): it performs the subresultant algorithm in the factor ring defined by the already built triangular sets instead of performing it generically and specializing its result afterwards.

There are however important theoretical differences between satTriangular and lextriangular: we start from a system of polynomial equations $\overline{A} = 0$, $\overline{S} \neq 0$ which is not a Gröbner basis[2] ; more important, we want much stronger properties for our output than (1) i.e. properties **P1** to **P5** stated in section 4. These properties just do not hold for the output of lextriangular in general and we were led to write complete proofs.

We have implemented two versions of Rosenfeld–Gröbner based on regCharacteristic. One in Maple 5.5 and one in C++. We compare existing methods (which are written in Maple) with our Maple version.

Let us compare our work with existing methods.

Wang and Li solve the *second step* in [21] by using the SimSys [32] algorithm which handles general polynomial systems. They claim [21, p. 59] that a more specialized algorithm than SimSys could be applied. Our regCharacteristic is such an algorithm.

In [8], this second step is solved by computing first a Gröbner basis of the localized ideal $S^{-1}(A)$. This is expensive, does not take into account the fact that A is already triangular and explicits the inverses of the elements of S which are not needed at all.

Algorithm 7.1 in [15] applies exactly the same principles as regCharacteristic. The only difference is that it computes a Gröbner basis of $S^{-1}(A)$ instead of calling satTriangular. It suffers therefore of the drawbacks mentioned above for [8]. Note also that [15] is the first to prove completely that computing characteristic presentations from regular differential systems is a purely algebraic (zerodimensional) problem. This was only claimed in [8, page 35].

Observe that testing the invertibility of differential polynomials modulo triangular sets in order to build characteristic sets was already considered in [23, 9]. The method of these authors is different from ours, at least because it is based on Gröbner bases computations [9, page 7] and [23, page 29].

The second result of this paper consists in two theorems (theorems 2 and 3) which clarify the relationship between characteristic presentations, characteristic sets and regular differential ideals.

Theorem 2 is new in the sense it makes a correspondence with the recent [2, Theorem 6.1]. Its content is essentially proved in [9] and [15, lemma 6.1]. Theorem 3 shows that a characteristic presentation of a regular differential ideal is a canonical representative among all the characteristic sets of that ideal.

Our two results above are generalizations close to results of François Ollivier [27, pages 89–98]. The main difference is that Ollivier only considers differential prime ideals while we consider regular differential ideals do not need to be prime. Ollivier does not reduce the problem to a zero dimensional problem and does not apply the more recent optimizations in [24, 25]. The conditions [27, Théorème 2, page 94 and Définition 10, page 96] Ollivier imposes to characteristic sets are slightly weaker than ours: he does not require that the elements of characteristic sets are primitive (our definition 3, condition **D3**) whence his characteristic sets are not canonical representatives of the prime differential ideals they define. However, when canonical representatives are needed for an application to control theory, Ollivier divides the elements of characteristic sets by their initials and does obtain canonical representatives [27, page 115].

Our third result is a new method to compute canonical forms of differential polynomials modulo a regular differential ideal.

Computing canonical forms of differential polynomials modulo a given differential ideal $\mathfrak{a}$ is a real issue for the set of canonical forms modulo $\mathfrak{a}$ often forms (here it is the case) a vector space over the base field K of the equations. Using canonical forms we thus can look for linear dependencies over K modulo $\mathfrak{a}$ by easily performing linear algebra in the factor ring. This is one of the main ideas carried out by the important FGLM algorithm [14] in the context of Gröbner bases and lifted to regular differential systems in [6].

This problem of computing canonical forms modulo polynomial differential equations was, as far as we know, only addressed in [6] and required the computation of a Gröbner basis. The method we give in section 8 is based on triangular sets and pseudo reduction only. It solves a problem left open in [6].

Our method applies for ordinary (non differential) polynomials modulo regular sets [16, 2] of polynomial equations too and we believe it could be interesting in this context also. Computing canonical forms imposes to compute the inverses of the initials of the elements of the set. These al-

[2] This is quite anecdotic for the algorithmic consequences of the Gianni and Kalkbrener theorem do not apply in our case.

gebraic inverses computations can be performed only once by making the set strongly normalized. This is probably very CPU expensive but is interesting at least for pedagogic reasons.

2. DEFINITIONS AND NOTATIONS

2.1 Commutative algebra

Let X be an ordered alphabet (possibly infinite). A *term* over X is a power product of elements of X.

Let $R = K[X]$ be a polynomial ring where K is a field. Let $p \in R \setminus K$ be a polynomial. The *leader* of p, denoted $\mathrm{ld}\, p$, is the greatest indeterminate x which occurs in p. The polynomial p can be written as

$$p = a_d\, x^d + \cdots + a_1\, x + a_0$$

where $d = \deg(p, x)$ and the polynomials a_i are free of x. The polynomial $i_p = a_d$ is the *initial* of p. The *rank* of p is the monomial x^d. If x^d and y^e are two ranks then $x^d < y^e$ if $x < y$ or $x = y$ and $d < e$. The *separant* of p is the polynomial

$$s_p = \frac{\partial p}{\partial x}.$$

The polynomial p is said to be *monic* if its initial is equal to 1. The set $\mathsf{iter}(p)$ of the *iterated initials* of p is defined as follows: if $p \in K$ then $\mathsf{iter}(p) = \emptyset$ otherwise $\mathsf{iter}(p) = \{p\} \cup \mathsf{iter}(i_p)$.

Let $A \subset R \setminus K$ be a set of polynomials. Then I_A (resp. S_A) denotes the set of the initials (resp. the separants) of its elements. We denote $H_A = I_A \cup S_A$. The set A is said to be *triangular* if its elements have distinct leaders.

Let A be a triangular set. A polynomial p is said to be *normalized* w.r.t. A if the set of leaders of $\mathsf{iter}(p)$ is disjoint from the set of leaders of A. The set A is said to be *normalized* if every $p \in A$ is normalized w.r.t. $A \setminus \{p\}$.

A polynomial p is said to be *strongly normalized* w.r.t. A if no leader of A occurs in the initial of p. The set A is said to be *strongly normalized* if every $p \in A$ is strongly normalized w.r.t. $A \setminus \{p\}$.

Every strongly normalized triangular set is normalized.

We denote $\mathsf{prem}(p, A)$ the pseudo–remainder [17, volume 2, page 407] of p by all the elements of A viewed as univariate polynomials in their leaders.

If R is a unique factorization domain and $p \in R[X]$ then p can be written:

$$p = a_0\, t_0 + \cdots + a_k\, t_k$$

where the t_i are terms over X and the $a_i \in R$. The *content* of p over R is the gcd of its coefficients:

$$\mathrm{cont}(p) = \gcd(a_0, \ldots, a_k)$$

The *primitive part* of p over R is the polynomial

$$\mathrm{pp}(p) = \frac{p}{\mathrm{cont}(p)}.$$

A polynomial is said to be *primitive* if it is equal to its primitive part.

If A is a subset of a ring R then (A) denotes the ideal generated by A. Let $\mathfrak{a}$ be an ideal of R. Then $\sqrt{\mathfrak{a}}$ denotes the radical of $\mathfrak{a}$. If $S = \{s_1, \ldots, s_t\}$ is a finite family of elements of R then the *saturation* $\mathfrak{a} : S^{\infty}$ of $\mathfrak{a}$ by S is the ideal:

$$\mathfrak{a} : S^{\infty} = \{p \in R \mid \exists a_1, \ldots, a_t \in \mathbb{N} \text{ s.t. } s_1^{a_1} \cdots s_t^{a_t}\, p \in \mathfrak{a}\}.$$

2.2 Differential algebra

We only provide a short presentation. The reference books are [28] and [18]. We also refer to the Maple 5.5 diffalg package and thus to the articles [7, 8] which present it.

A *derivation* over a ring R is a map $\delta : R \to R$ which satisfies, for every $a, b \in R$

$$\begin{aligned} \delta(a+b) &= \delta a + \delta b, \\ \delta(a\, b) &= (\delta a) b + a(\delta b). \end{aligned}$$

A *differential ring* is a ring endowed with finitely many derivations which commute pairwise. The commutative monoid generated by the derivations is denoted Θ. Its elements are the *derivation operators* $\theta = \delta_1^{a_1} \cdots \delta_m^{a_m}$ where the a_i are nonnegative integer numbers. The sum of the exponents a_i, called the *order* of the operator θ, is denoted $\mathrm{ord}\, \theta$. The identity operator is the unique operator with order 0. The other ones are called *proper*. If $\phi = \delta_1^{b_1} \cdots \delta_m^{b_m}$ then $\theta\phi = \delta_1^{a_1+b_1} \cdots \delta_m^{a_m+b_m}$. If $a_i > b_i$ for each $1 \le i \le m$ then $\theta/\phi = \delta_1^{a_1-b_1} \cdots \delta_m^{a_m-b_m}$.

A *differential ideal* $\mathfrak{a}$ of R is an ideal of R stable under derivation i.e. such that

$$a \in \mathfrak{a} \Rightarrow \delta a \in \mathfrak{a}.$$

Let A be a nonempty subset of R. We denote $[A]$ the differential ideal generated by A which is the smallest differential ideal which contains A.

2.2.1 Differential polynomials

Let $U = \{u_1, \ldots, u_n\}$ be a set of *differential indeterminates.* Derivation operators act over differential indeterminates giving *derivatives* θu. We denote ΘU the set of all the derivatives. Let K be a differential field. The differential ring of the differential polynomials built over the alphabet ΘU with coefficients in K is denoted $R = K\{U\}$.

A *ranking* is a total ordering over the set of the derivatives [18, page 75] satisfying the following axioms

1. $\delta v > v$ for each derivative v and derivation δ,
2. $v > w \Rightarrow \delta v > \delta w$ for all derivatives v, w and each derivation δ.

Fix a ranking. The infinite alphabet ΘU gets ordered. Consider a polynomial $p \in R \setminus K$. Then the leader, initial, separant ... of p are well defined. Axioms of rankings imply that the separant of p is the initial of every proper derivative of p.

Let rank $p = v^d$. A differential polynomial q is said to be *partially reduced* w.r.t. p if no proper derivative of v occurs in q. It is said to be *reduced* w.r.t. p if it is partially reduced w.r.t. p and $\deg(q, v) < d$.

A set A of differential polynomials is said to be *differentially triangular* if it is triangular and if its elements are pairwise partially reduced. It is said to be *autoreduced* if its elements are pairwise reduced.

Every autoreduced set is differentially triangular.

If A is a set of differential polynomials and v is a derivative then $A_v = \{\theta p \mid p \in A,\ \mathrm{ld}\,\theta p \leq v\}$. Thus R_v denotes the set of all the differential polynomials having leader less than or equal to v.

2.2.2 Ritt's reduction algorithms

They are generalizations of the Euclidean division algorithm for differential polynomials. One distingues the partial reduction algorithm, denoted $\mathsf{partial\_rem}$ from the full reduction algorithm, denoted $\mathsf{full\_rem}$. We only give specifications of these algorithms. See [18, page 77] for a more precise description. Let q be a differential polynomial and A be a set of differential polynomials. Let $v = \mathrm{ld}\,q$ and $\overline{A} = A \cap R_v$.

If $\overline{q} = \mathsf{partial\_rem}(q, A)$ denotes the *partial remainder* of q by A then $\overline{q}$ is partially reduced w.r.t. all the elements of A and there exists a power product h of elements of $S_{\overline{A}}$ such that $h\,q \equiv \overline{q} \mod (\overline{A}_v)$.

If $\overline{q} = \mathsf{full\_rem}(q, A)$ denotes the *full remainder* of q by A then $\overline{q}$ is reduced w.r.t. all the elements of A, there exists a power product h of elements of $H_{\overline{A}}$ such that $h\,q \equiv \overline{q} \mod (\overline{A}_v)$.

2.2.3 Critical pairs

A pair $\{p_1, p_2\}$ of differential polynomials is said to be a *critical pair*[3] if the leaders of p_1 and p_2 are derivatives of some same differential indeterminate u (say $\mathrm{ld}\,p_1 = \theta_1 u$ and $\mathrm{ld}\,p_2 = \theta_2 u$). Assume A is differentially triangular. Then $\mathrm{critical\_pairs}(A)$ denotes all the critical pairs that can be formed with any two elements of A. Let $\{p_1, p_2\} \in \mathrm{critical\_pairs}(A)$ be a critical pair. Denote θ_{12} the least common multiple between θ_1 and θ_2. The Δ–polynomial $\Delta(p_1, p_2)$ is

$$\Delta(p_1, p_2) = s_2\,\frac{\theta_{12}}{\theta_1}\,p_1 - s_1\,\frac{\theta_{12}}{\theta_2}\,p_2$$

where s_1, s_2 denote the separants of p_1 and p_2. Let $A = 0,\ S \neq 0$ be a system of differential polynomial equations and inequations. The critical pair $\{p_1, p_2\}$ is said to be *solved* by $A = 0,\ S \neq 0$ if there exists a derivative $v < \theta_{12} u$ such that

$$\Delta(p_1, p_2) \in (A_v) : (S \cap R_v)^{\infty}.$$

If $\mathsf{full\_rem}(\Delta(p_1, p_2), A) = 0$ and $H_A \subset S$ then the critical pair $\{p_1, p_2\}$ is solved by $A = 0,\ S \neq 0$.

2.2.4 Regular differential systems

Definition 1. *(regular differential systems)*[4]

A differential system $A = 0,\ S \neq 0$ of a differential polynomial ring R is said to be a regular differential system *(for a ranking $\mathcal{R}$) if*

C1 *A is differentially triangular,*

C2 *S contains the separants of the elements of A and is partially reduced w.r.t. A,*

C3 *all the critical pairs $\{p,\ p'\} \in \mathrm{critical\_pairs}(A)$ are solved by the system $A = 0,\ S \neq 0$ (coherence property*[5]*).*

If $A = 0,\ S \neq 0$ is a regular differential system then the ideal $[A] : S^{\infty}$ (resp. $(A) : S^{\infty}$) is called the *regular differential ideal* (resp. *regular algebraic ideal*) defined by the system.

If $A = 0,\ S \neq 0$ is a regular differential system, we call *derivatives under the stairs of A* the elements of ΘU which are not derivatives of any leader of element of A.

Regular systems enjoy the following properties.

Theorem 1. *Let $A = 0,\ S \neq 0$ be a regular differential system of $R = K\{U\}$. Let L denote the set of leaders of A and N the set of the derivatives under the stairs of A. Then*

1. *the regular algebraic ideal $(A) : S^{\infty}$ is radical (Lazard's lemma) ;*
2. *if $\mathfrak{b}$ denotes a prime ideal minimal over $(A) : S^{\infty}$ then the set N furnishes a transcendence basis of the field of fractions of $R/\mathfrak{b}$ over K (Lazard's lemma) ;*
3. *we have $[A] : S^{\infty} \cap K[L, N] = (A) : S^{\infty}$ (Rosenfeld's lemma) ;*
4. *the regular differential ideal $[A] : S^{\infty}$ is radical (lifting of Lazard's lemma) ;*
5. *if $(A) : S^{\infty}$ has t minimal primes $\mathfrak{b}_i$ then $[A] : S^{\infty}$ has t minimal differential primes $\mathfrak{p}_i$ defined by (lifting of Lazard's lemma)*
$$\mathfrak{p}_i \cap K[L, N] \;=\; \mathfrak{b}_i.$$
Actually $\mathfrak{p}_i = \{p \in R \mid \mathsf{partial\_rem}(p, A) \in \mathfrak{b}_i\}$.

Proof. [8, Lazard's lemma, the lifting of Lazard's lemma and Rosenfeld's lemma]. See also [31, 29, 7, 30, 26, 15, 21]. □

[3] This definition was introduced for the first time in [8, section 4], under the name "pair".

[4] [8, Definition 4.4].

[5] If $I_A \subset S$ and $\mathsf{full\_rem}(\Delta(p, p'), A) = 0$ for every critical pair $\{p,\ p'\} \in \mathrm{critical\_pairs}(A)$ then **C3** is satisfied.

2.2.5 Characteristic sets

Definition 2. *Let $\mathfrak{a}$ be a differential ideal of R. A set $C \subset \mathfrak{a}$ is said to be a* characteristic set *of $\mathfrak{a}$ if C is autoreduced and $\mathfrak{a}$ contains no nonzero polynomial reduced w.r.t. C.*

In the following theorem, the equivalence between the two first items is well–known. This theorem is very close to [27, Théorème 2, page 94] though Ollivier only considers differential prime ideals. The equivalence with the third item may be essentially proved already in [9]. The correspondence with the recent [2, Theorem 6.1] is interesting in itself anyway.

Theorem 2. *Let C be an autoreduced set of differential polynomials and $\mathfrak{a} = [C] : H_C^\infty$. The following conditions are equivalent.*

1. *C is a characteristic set of $\mathfrak{a}$,*
2. *$p \in \mathfrak{a} \Leftrightarrow \mathsf{full\_rem}(p, C) = 0$,*
3. *$C = 0,\ H_C \neq 0$ is a regular differential system such that C is regular*[6] *in the sense of [2] and squarefree*[7] *in the sense of [1, def 4.5.11].*

Proof. $1 \Rightarrow 2$. If $p \in \mathfrak{a}$ then $\mathsf{full\_rem}(p, C) \in \mathfrak{a}$, is reduced w.r.t. C, thus is zero. If $\mathsf{full\_rem}(p, C) = 0$ then, for some power product h of elements of H_C we have $h\,p \in [C] \subset \mathfrak{a}$ whence $p \in \mathfrak{a}$.

$2 \Rightarrow 3$. The differential system $C = 0,\ H_C \neq 0$ satisfies conditions **C1** and **C2**. The set C reduces to zero all $\Delta(p, p')$ such that $\{p, p'\} \in \mathrm{critical\_pairs}(C)$ (for C reduces $\mathfrak{a}$ to zero) thus $C = 0,\ H_C \neq 0$ satisfies **C3** and is a regular differential system. C reduces to zero $(C) : I_C^\infty \subset \mathfrak{a}$ whence is regular in the sense of [2] by [2, Theorem 6.1]. C reduces to zero $(C) : H_C^\infty \subset \mathfrak{a}$ without differentiating any element of C whence $(C) : H_C^\infty \subset (C) : I_C^\infty$. Since the converse inclusion obviously holds too we have $(C) : I_C^\infty = (C) : H_C^\infty$ thus C is squarefree by theorem 1 (1).

$3 \Rightarrow 1$. Let f be a differential polynomial reduced w.r.t. C. We must prove $f = 0$. We have $f \in K[L, N]$. Since $C = 0,\ H_C \neq 0$ is a regular differential system, theorem 1 (3) applies and $f \in (C) : H_C^\infty$. The set C is regular in the sense of [2] thus C is a characteristic set of $(C) : I_C^\infty$ by [2, Theorem 6.1]. Since C is squarefree, $(C) : I_C^\infty$ is radical and $(C) : I_C^\infty = (C) : H_C^\infty$ by [15, Proposition 3.3]. Therefore C is a characteristic set of $(C) : H_C^\infty$ and $f = 0$. □

2.2.6 Characteristic presentations

The following definition is different from that of [8, Definition 6.1] but we do believe both definitions are equivalent (we do not prove this claim). Hubert weakens the definition of characteristic presentations in [15] (she only imposes **D1** and **D2**) thus looses canonicity properties.

Definition 3. *A set $C \subset K\{U\}$ is said to be a* characteristic presentation *of the differential ideal $[C] : H_C^\infty$ if*

D1 *the differential system $C = 0,\ H_C \neq 0$ is regular,*

D2 *if $p \in R$ then $p \in [C] : H_C^\infty \Leftrightarrow \mathsf{full\_rem}(p, C) = 0$,*

D3 *C is a strongly normalized autoreduced set of $K[L, N]$ such that the elements of C are primitive over $K[N]$ where L denotes the set of leaders of the elements of C and N denotes the other derivatives occuring in C.*

Corollary 1. *A set C of differential polynomials is a characteristic presentation of $[C] : H_C^\infty$ if and only if C is a characteristic set of $[C] : H_C^\infty$ which satisfies* **D3**.

Theorem 3. *A characteristic presentation C is a canonical representative of the regular differential $[C] : H_C^\infty$ (it only depends on the ideal and on the ranking).*

Proof. Let C and C' be two characteristic presentations of $[C] : H_C^\infty$. Both sets have the same rank for they are characteristic sets of the same ideal. Consider any $f \in C$ and $f' \in C'$ having the same rank. Denote i and i' their initials. The polynomial $i'\,f - i\,f' \in (C) : H_C^\infty$. Since C and C' are autoreduced and strongly normalized, this polynomial is reduced w.r.t. both C and C' whence is zero. Since f and f' are primitive over $K[N]$ we have $f = f'$ thus $C = C'$. □

In practice, in order to have canonicity properties, we impose also the coefficients in K of the elements of C to be normalized. In the case $K = \mathbb{Q}(Y)$ is a pure transcendental field extension of the field of the rational numbers (Y is an alphabet of indeterminates) then the elements of C are polynomials in $\mathbb{Z}[Y \cup N \cup L]$ primitive over the ring $\mathbb{Z}[Y \cup N]$.

3. THE PROBLEM

We are given a regular differential system $A = 0,\ S \neq 0$ of R. We assume moreover that $H_A \subset S$. We want to compute sets $C_1, \ldots, C_t$ of differential polynomials such that the following conditions hold

A1 each C_i is a characteristic presentation of the differential ideal $[C_i] : H_{C_i}^\infty$,

A2 $[A] : S^\infty = [C_1] : H_{C_1}^\infty \cap \cdots \cap [C_t] : H_{C_t}^\infty$,

A3 the intersection is not redundant: if $\mathfrak{p}$ is a differential prime component of $[A] : S^\infty$ then $\mathfrak{p}$ is a minimal differential prime of exactly one differential ideal $[C_i] : H_{C_i}^\infty$.

4. REGCHARACTERISTIC

Denote X the set of the derivatives occuring in $A \cup S$ and $L \subset X$ the set of the leaders of the elements of A and $N = X \setminus L$. Denote $G = K(N)$ the ring obtained by putting the elements of N in the base field of the differential polynomial ring. The algorithm involves three steps.

[6] A triangular set $C = f_1 < \cdots < f_n$ is *regular* in the sense of [2] if the initial of f_k does not divide zero modulo $(f_1, \ldots, f_{k-1}) : (i_1 \cdots i_{k-1})^\infty$ for every $1 \leq k \leq n$.

[7] A regular (in the sense of [2]) triangular set C is said to be *squarefree* if $(C) : I_C^\infty$ is radical (in [1, def 4.5.11], the qualifier *separable* is used in place of *squarefree*).

1. Transform the system $A = 0,\ S \neq 0$ into a system $\overline{A} = 0,\ \overline{S} \neq 0$ of $G[L]$. That step is purely formal.

2. Apply the algorithm satTriangular (described later) over $\overline{A} = 0,\ \overline{S} \neq 0$. This algorithm returns a possibly empty set of squarefree normalized[8] autoreduced sets $\{T_1, \ldots, T_t\}$ of $G[L]$ satisfying the following properties:

   P1 if the set is empty then $(\overline{A}) : \overline{S}^{\infty} = G[L]$,

   P2 $(\overline{A}) : \overline{S}^{\infty} = (T_1) \cap \cdots \cap (T_t)$,

   P3 if $i \neq j$ then $(T_i) + (T_j) = (1)$,

   P4 for every $1 \leq i \leq t$ we have $(T_i) = (T_i) : H_{T_i}^{\infty}$,

   P5 $\mathrm{ld}\, T_i = \mathrm{ld}\, A$.

3. Transform each triangular set T_i of $G[L]$ as a triangular set C_i of $K[X]$ by replacing every polynomial p/s occuring in the T_i systems by the primitive part of p over $K[N]$. The obtained systems $C_1, \ldots, C_t$ are the characteristic presentations we are looking for.

4.1 Proof of the algorithm

The part of regCharacteristic that we prove in this section is shared together with [15, Algorithm 7.1]. Proofs can thus be found in [15, Theorem 3.10 and 6.2]. We give them anew to make this article selfcontained.

Let ϕ be the canonical ring homomorphism $K[X] \to G[L]$.

Lemma 1. *$(A) : S^{\infty} = (C_1) : H_{C_1}^{\infty} \cap \cdots \cap (C_t) : H_{C_t}^{\infty}$.*

Proof. The relation above is obtained by applying ϕ^{-1} over **P2** componentwise. Indeed, ϕ^{-1} preserves intersections by [13, Proposition 2.2, (a)] and maps $(\overline{A}) : \overline{S}^{\infty}$ (resp. (T_i)) to $(A) : S^{\infty}$ (resp. $(C_i) : H_{C_i}^{\infty}$) by **P4**, [13, Proposition 2.2] and the fact that the nonzero elements of $K[N]$ belong to none of the minimal primes of $(A) : S^{\infty}$ (resp. $(C_i) : H_{C_i}^{\infty}$) by theorem 1 (2) (resp. **P5** and theorem 1 (2)). □

Lemma 2. *A prime ideal $\mathfrak{b}$ is minimal over $(A) : S^{\infty}$ iff $\mathfrak{b}$ is minimal over some $(C_i) : H_{C_i}^{\infty}$.*

Proof. By **P5** and theorem 1 (2), the prime ideals which are minimal over $(A) : S^{\infty}$ and the ones which are minimal over the $(C_i) : H_{C_i}^{\infty}$ all have the same dimension. If $\mathfrak{b} \subset \mathfrak{b}'$ are two prime ideals having the same dimension then $\mathfrak{b} = \mathfrak{b}'$. The lemma follows now from lemma 1. □

Lemma 3. *Each system $C_i = 0,\ H_{C_i} \neq 0$ is a regular differential system.*

Proof. It suffices to prove if $\{p, p'\} \in \text{critical\_pairs}(C_i)$ then $\mathsf{full\_rem}(\Delta(p, p'), C_i) \in \mathfrak{b}$ where $\mathfrak{b}$ is any prime ideal minimal over $(C_i) : H_{C_i}^{\infty}$. By lemma 2 and theorem 1 (5) $\mathfrak{b}$ is the intersection with $K[X]$ of some differential prime ideal $\mathfrak{p}$ minimal over $[A] : S^{\infty}$. Since p, p', C_i belong to $\mathfrak{p}$ we have $\mathsf{full\_rem}(\Delta(p, p'), C_i) \in \mathfrak{b}$. □

Lemma 4. *Let $1 \leq i \leq t$ be an index. The set C_i is a characteristic set, in the sense of Ritt, of the ideal $(C_i) : H_{C_i}^{\infty}$*

Proof. Let us prove that $p \in (C_i) : H_{C_i}^{\infty}$ iff $\mathsf{prem}(p, C_i) = 0$ for any $p \in K[X]$. The implication from right to left is clear. The converse one comes from the following: first, $p \in (C_i) : H_{C_i}^{\infty}$ iff $\phi p \in (T_i)$ by **P4** ; second, $\mathsf{prem}(p, C_i) = 0$ iff $\mathsf{prem}(\phi p, T_i) = 0$; third $\mathsf{prem}(\phi p, T_i) = 0$ iff $\phi p \xrightarrow[T_i]{*} 0$ for[9] the elements of T_i are monic ; last, T_i is a Gröbner basis of (T_i) [3, Lemma 5.66 (Buchberger's first criterion)] thus reduces to zero all the elements of (T_i) [3, Proposition 5.38]. □

Proposition 1. *(condition* **A1***)*

Let $1 \leq i \leq t$ be an index. The set C_i is a characteristic presentation of the ideal $[C_i] : H_{C_i}^{\infty}$.

Proof. The set C_i is autoreduced. All its elements are strongly normalized for they are obtained by multiplying monic polynomials by elements of $K[N]$. They are primitive over $K[N]$ by construction. Thus C_i satisfies **D3**. Condition **D1** holds by lemma 3. Condition **D2** thus holds by theorem 1 (3) and lemma 4.

Lemma 5. *A differential prime $\mathfrak{p}$ is minimal over $[A] : S^{\infty}$ iff it is minimal over some $[C_i] : H_{C_i}^{\infty}$.*

Proof. By lemmas 2, 3 and theorem 1 (5). □

Proposition 2. *(condition* **A2***)*

$$[A] : S^{\infty} = [C_1] : H_{C_1}^{\infty} \cap \cdots \cap [C_t] : H_{C_t}^{\infty}.$$

Proof. The proposition is a corollary of lemma 5. □

Proposition 3. *(condition* **A3***)*

If $\mathfrak{p}$ is a differential prime component of $[A] : S^{\infty}$ then $\mathfrak{p}$ is a differential prime component of exactly one differential ideal $[C_i] : H_{C_i}^{\infty}$.

Proof. Let $\mathfrak{p}$ be a minimal differential prime of $[C_i] : H_{C_i}^{\infty}$ and of $[C_j] : H_{C_j}^{\infty}$. Then $\mathfrak{b} = \mathfrak{p} \cap K[X]$ is a minimal prime of $(A) : S^{\infty}$ by lemma 5. Then $(\phi \mathfrak{b})$ is a minimal prime of (T_i) and of (T_j). Property **P3** implies $i = j$. □

5. THE INVERT SUBALGORITHM

Computing the normalized sets T_i amounts to normalizing the polynomials of $\overline{A}$, which consists in inverting the initials of the polynomials of $\overline{A}$.

Denote $L = \{X_1, \ldots, X_n\}$.

[8] Observe in this case, every normalized set is strongly normalized for its elements are monic.

[9] This denotes the reduction in the sense of the Gröbner basis theory [3, page 199].

5.1 Specification of invert

We describe precisely the inputs and outputs of the algorithm invert:

Inputs of invert:

- $p \in G[L]$, a non zero polynomial
- $T = \{p_1, \ldots, p_k\}$ a normalized triangular set of the ring $G[X_1, \ldots, X_k]$ with $k \leq n$.

Outputs of invert:

- either the inverse $q \in G[L]$ such that $pq = 1 \pmod{(T)}$. We say invert has found the inverse of p modulo the ideal (T).
- or a triple (j, g, h) (if invert could not compute an inverse) with $j \leq k$ such that
  - $\mathrm{ld}(g) = \mathrm{ld}(h) = X_j$
  - g and h are monic
  - $p_j = g\,h \pmod{(p_1, \ldots, p_{j-1})}$

5.2 Algorithmic scheme of invert

The algorithms invert and ExtEuclid (see below) respectively are simplified versions of the functions QuasiRecipElseSplit and extendedSubResGcdElseSplit taken from [25] and based on a splitting process à la D5 [11]. See [20, 1] too. The implementation is far more complicated and optimizations are detailed in section 7.

Denote p and q two nonconstant polynomials of the ring $G[X_1, \ldots, X_n]$ such that q is monic and has leader X_k.

The polynomial $\mathsf{quo}(p, q, X_k)$ (resp. $\mathsf{rem}(p, q, X_k)$) denotes the quotient (resp remainder) of the Euclidean division of p by q. The division does not raise any problem since q is monic.

If $T = \{p_1, \ldots, p_k\}$ is a normalized triangular set of the ring $G[X_1, \ldots, X_k]$, then, in the algorithms, $p \mod T$ denotes $\mathsf{rem}(\ldots \mathsf{rem}(p, p_k, X_k) \ldots, p_1, X_1)$, (i.e. the remainder of the Euclidian division of p by all polynomials of T). Remark: T can be considered as a Gröbner basis and $p \mod T$ is equal to the normal form [3, page 199] of p by T.

```
invert($p, \{p_1, \ldots, p_k\}$)
    if $p \in G$
        return $1/p$
    else
        let $1 \leq j \leq k$ such that $\mathrm{ld}\,p = \mathrm{ld}\,p_j$
        $(g, u, v)$ := ExtEuclid($p, p_j, \{p_1, \ldots, p_{j-1}\}$)
        if $g = 1$ then
            return $u$
        else
            return to top level the triple
                $(j, g, \mathsf{quo}(p_j, g, X_k) \mod \{p_1, \ldots, p_{j-1}\})$
        fi
    fi
```

The algorithm invert is based on the algorithm ExtEuclid we describe below.

Specification of ExtEuclid

$T = \{p_1, \ldots, p_{k-1}\}$ is a normalized triangular set of the ring $G[X_1, \ldots, X_{k-1}]$ and p and q are two polynomials of $G[X_1, \ldots, X_k]$ with $\mathrm{ld}(p) = \mathrm{ld}(q) = X_k$ and q is monic. ExtEuclid tries to computes a triple (g, u, v) of three polynomials of $G[X_1, \ldots, X_k]$ such that:

- $up + vq = g \pmod{(T)}$ (Bézout [4] identity)
- $\mathrm{lcoeff}(g, X_k) = 1$, i.e. g is either the constant polynomial 1 or a monic polynomial with leader X_k.
- g divides both p and q modulo the ideal (T)

Algorithmic scheme of ExtEuclid

```
ExtEuclid($p, q, T$)
    if $q = 0$ then
        $\overline{\imath_p}$ :=invert($i_p, T$)
        return ($\overline{\imath_p}\, p \mod T, \overline{\imath_p}, 0$)
    else
        $a$ := quo($p, q, X_k, T$)
        $b$ := rem($p, q, X_k, T$)
        $\overline{\imath_b}$ :=invert($i_b, T$)
        $(g, u, v)$ := ExtEuclid($q$, $\overline{\imath_b}\, b \mod T$, $T$)
        return ($g, \overline{\imath_b}\, v \mod T, u - \overline{\imath_b}\, v\, a \mod T$)
    fi
```

Remark: in case of splitting, the algorithm ExtEuclid does not compute the expected triple because it is interrupted by the algorithm invert at the line “return to top level”.

6. SATTRIANGULAR

Recall $H_{\overline{A}} \subset \overline{S}$ and $L = \mathrm{ld}(\overline{A})$. The algorithm satTriangular builds a finite sequence $(\mathcal{F}_i)_{1 \leq i \leq r}$ of r sets of systems of equations and inequations.

Initially, take $\mathcal{F}_0 = \{(\overline{A} = 0,\ \overline{S} \neq 0)\}$. We suppose we have built the set $\mathcal{F}_i$. Two cases may arise:

- Denote $\mathcal{F}_i = \{(T_1 = 0, S_1 \neq 0), \ldots, (T_t = 0, S_t \neq 0)\}$. If each T_i is a normalized triangular set and each S_i is empty, the algorithm then stops and outputs the set $\{T_1, \ldots, T_t\}$.
- $\mathcal{F}_i$ contains a system $A = 0, S \neq 0$ such that A contains a non monic polynomial or S is not empty. Then transform $A = 0, S \neq 0$ with one of the two rules R1 or R2 (defined below). Both rules compute a set $\mathcal{F}$ of zero, one or two systems of equations and inequations.

  Take $\mathcal{F}_{i+1} = \mathcal{F}_i \setminus \{A = 0, S \neq 0\} \cup \mathcal{F}$.

R1 : try to make a polynomial monic. If there exists some non monic $p_k \in A$ s.t. $p_1, \ldots, p_{k-1}$ are monic, three cases are possible:

R1.1 The initial of p_k is zero modulo $(p_1, \ldots, p_{k-1})$. Take $\mathcal{F} = \emptyset$.

R1.2 invert finds the inverse q of i_{p_k} modulo the ideal $(p_1, \ldots, p_{k-1})$. Take $\mathcal{F} = \{A' = 0,\ S \neq 0\}$ where $A' = A \setminus \{p_k\} \cup \{\overline{p}_k\}$ where $\overline{p}_k = q\,p_k \bmod \{p_1, \ldots, p_{k-1}\}$.

R1.3 invert does not find the inverse of i_{p_k} modulo $(p_1, \ldots, p_{k-1})$, but a triple (j, g, h) such that $1 \leq j < k$ and $p_j = g\,h \pmod{(p_1, \ldots, p_{j-1})}$. Take $\mathcal{F} = \{(A_g = 0,\ S \neq 0), (A_h = 0,\ S \neq 0)\}$ where $A_g = A \setminus \{p_j\} \cup \{g\}$ and $A_h = A \setminus \{p_j\} \cup \{h\}$.

R2 : try to get rid of an inequation. If there is some $s \in S$ such that $\operatorname{ld} s = \operatorname{ld} p_k$ and $p_1, \ldots, p_k$ are monic then three cases are possible:

R2.1 s is zero modulo $(p_1, \ldots, p_k)$. Take $\mathcal{F} = \emptyset$.

R2.2 invert finds the inverse of s modulo $(p_1, \ldots, p_k)$. Take $\mathcal{F} = \{A = 0, S' \neq 0\}$ where $S' = S \setminus \{s\}$.

R2.3 invert does not find the inverse of s modulo the ideal $(p_1, \ldots, p_k)$ but a triple (j, g, h) such that $1 \leq j \leq k$ and $p_j = g\,h \pmod{(p_1, \ldots, p_{j-1})}$. Take $\mathcal{F} = \{(A_g = 0,\ S \neq 0), (A_h = 0,\ S \neq 0)\}$ where $A_g = A \setminus \{p_j\} \cup \{g\}$ and $A_h = A \setminus \{p_j\} \cup \{h\}$.

6.1 Proof

If $I \subset G[L]$ is an ideal, $V(I)$ denotes the set of zeros of I in the algebraic closure of G.

Lemma 6. *Let $I \subset G[L]$ be a zerodimensional ideal. Let $S \subset G[L]$ a finite set of polynomials. Then $I = I : S^{\infty}$ iff for each $s \in S$ and $z \in V(I)$, we have $s(z) \neq 0$.*

Proof. (sketched) The proof follows from the two following points: first, every zero of I is an irreducible component of the algebraic variety of I (since I is zerodimensional); second, the algebraic variety of $I : S^{\infty}$ is the union of all the irreducible components of I which do not annihilate any element of S. □

Proposition 4. *For $1 \leq i \leq r$, $\mathcal{F}_i$ satisfies invariants **I1** to **I3** where $\mathcal{F}_i = \{(A_1 = 0, S_1 \neq 0), \ldots, (A_t = 0, S_t \neq 0)\}$.*

I1 $\bigcap_{1 \leq l \leq t} (A_l) : S_l^{\infty} = (\overline{A}) : \overline{S}^{\infty}$.

I2 $(A_l) : S_l^{\infty} = (A_l) : (S_l \cup H_{A_l})^{\infty}$*, for each* $1 \leq l \leq t$.

I3 $(A_l) : S_l^{\infty} + (A_m) : S_m^{\infty} = G[L]$*, for each* $1 \leq l < m \leq t$.

Corollary 2. *For each $1 \leq l \leq t$, the ideal $(A_l) : S_l^{\infty}$ has dimension zero and is radical.*

Proof. This is a corollary to invariant **I2** and Lazard's lemma (theorem 1 (1,2)). □

Proof. **of proposition 4**

The proof is an induction on i. The basis of the induction is clear. Suppose $\mathcal{F}_i$ verifies the invariants. We prove that $\mathcal{F}_{i+1}$ does too. We distinguish several cases, corresponding to the way $\mathcal{F}_{i+1}$ is built.

• Case R1.1
It suffices to prove $(A) : S^{\infty} = G[L]$. The initial of p_k belongs to both (A) and H_A. Thus $(A) : (S \cup H_A)^{\infty} = G[L]$. By invariant **I2**, $(A) : S^{\infty} = G[L]$.

• Case R1.2
We have $(A) = (A')$ since $q\,p_k \in (A)$ and $p = i_{p_k}\,q\,p_k \pmod{(p_1, \ldots, p_{k-1})}$.

Therefore, by saturating, $(A) : S^{\infty} = (A') : S^{\infty}$. Thus, $\mathcal{F}_{i+1}$ satisfies **I1** and **I3**.

It remains to prove **I2**. We have $(A) : S^{\infty} = (A) : (S \cup H_A)^{\infty}$ (invariant **I2** over $\mathcal{F}_i$). Lemma 6 implies that the polynomials of H_A do not vanish on the zeros of $(A) : S^{\infty}$. Since $S_{A'} = S_A \setminus \{s_{p_k}\} \cup \{s_{\overline{p}_k}\}$ and $s_{\overline{p}_k} = q\,s_{p_k} \pmod{(p_1, \ldots, p_{k-1})}$, the polynomials of $H_{A'}$ do not vanish either on the zeros of $(A) : S^{\infty}$. Thus, by lemma 6, $(A) : S^{\infty} = (A) : (S \cup H_{A'})^{\infty} = (A') : S^{\infty} = (A') : (S \cup H_{A'})^{\infty}$, so $\mathcal{F}_{i+1}$ verifies **I2**.

• Case R1.3
We have $p_j = g\,h \pmod{(p_1, \ldots, p_{j-1})}$. Since p_j, g and h have the same leader, $s_{p_j} = s_g h + g s_h \pmod{(p_1, \ldots, p_{j-1})}$.

Denote $I = (A) : S^{\infty}$ and $I_g = (A_g) : S^{\infty}$ and $I_h = (A_h) : S^{\infty}$.

*Proof of **I3***: we claim

$$\begin{aligned} V(I_g) \cup V(I_h) &= V(I) && (2) \\ V(I_g) \cap V(I_h) &= \emptyset && (3) \end{aligned}$$

We have $V(I) \supset V(I_g) \cup V(I_h)$ for $I \subset I_g$ and $I \subset I_h$.

The converse inclusion. Let $z \in V(I)$.

$$\begin{aligned} g(z)h(z) &= 0 && (4) \\ s_g(z)h(z) + g(z)s_h(z) &\neq 0 && (5) \end{aligned}$$

Thus, if $g(z) = 0$ then $s_g(z) \neq 0$ and $h(z) \neq 0$; if $g(z) \neq 0$, then $h(z) = 0$ and $s_h(z) \neq 0$. This implies z is either a zero of I_g or a zero of I_h, and it can't be zero of both ideals. Thus 2 and 3 are proved and $\mathcal{F}_{i+1}$ satisfies **I3**.

*Proof of **I2***:

By invariant **I2** on $\mathcal{F}_i$ and lemma 6, the polynomials of H_A do not vanish on $V(I)$. By relation 2, they do not vanish on $V(I_g)$. The polynomial s_g does not vanish on $V(I_g)$ (consequence of relation 5). We have $H_{A_g} = H_A \setminus \{s_{p_j}\} \cup \{s_g\}$. Therefore, by lemma 6, $(A_g) : S^{\infty} = (A_g) : (S \cup H_{A_g})^{\infty}$. The same proof holds for I_h. This ends the proof of **I2**.

By the theorem of zeros, $\sqrt{I} = \sqrt{I_g} \cap \sqrt{I_h}$. By invariant **I2** and theorem 1(1), all these ideals are radical. Thus $\mathcal{F}_{i+1}$ satisfies **I1**.

• Case R2.1

Same as **R1.1**.

• Case R2.2
The inequation s admits an inverse we call q. The property $s\,q = 1 \pmod{(p_1, \ldots, p_k)}$ clearly implies that s does not vanish on the zeros of $(A) : S^\infty$. Thus, by lemma 6, we have $(A) : S^\infty = (A) : S'^\infty$ and $\mathcal{F}_{i+1}$ satisfies the three invariants.

• Case R2.3
Same as **R1.3**

This ends the proof of proposition 4 □

PROPOSITION 5. *The algorithms stops.*

PROOF. To each system $\Sigma = (A = 0, S \neq 0)$, associate the sum of the degrees of the elements of A in their leader, the number of the non monic elements of A and the number of elements of S. Denote $v(\Sigma)$ this positive integer.

When the algorithm rewrites a system Σ into a set $\mathcal{F}$, then $v(\Sigma') < v(\Sigma)$ for each $\Sigma' \in \mathcal{F}$.

By [19, Satz 6.6] (i.e. every infinite locally finite tree contains a branch of infinite length), the algorithm stops. □

PROPOSITION 6. *The output* $\{T_1, \ldots, T_t\}$ *of the algorithm satisfies* ***P1****, ...,* ***P5***

PROOF. This is an immediate consequence of the invariants **I1**, **I2** and **I3** satisfied by $\mathcal{F}_r$. □

7. IMPLEMENTATION

The codes of the algorithms invert and ExtEuclid, in section 5, just are algorithmic schemes. Most of the optimizations below are implemented in our private version of the Maple package diffalg. They were suggested to us by Marc Moreno Maza, who is very familiar with triangularization algorithms.

- One can compute pairs instead of triples in ExtEuclid following [17, Volume 2, page 325]: using the final pair and the Bézout identity, one can recover the missing coefficient by a mere Euclidean division modulo a triangular set.
- For efficiency reasons, one can implement different versions of ExtEuclid, one which computes a full Bézout identity, one which only computes one of the Bézout coefficients (for computing an inverse), and one which only computes the gcd (for testing invertibility). For example, Marc Moreno Maza observed that the last division of the former item can be very expensive.
- The classical scheme of the extended Euclidean algorithm can be performed with the subresultants algorithm which controls the growth of the coefficients of the intermediate remainders. The recent Ducos [12] optimization of the subresultants algorithm is worth being implemented. It replaces some pseudoremainders of the traditional subresultant algorithm by a few more computations which involve smaller datas. The alternative [22] algorithm could be used as well.
- The rule R2.3 can be specialized when $j = k$. In this case, the system $A_g = 0,\ S \neq 0$ is inconsistent. The same optimization applies for R1.3 when $j = k - 1$.
- It is interesting to invert separants of the equations as soon as possible. Once the separants of all considered equations are inverted, the ideal generated by these equations is radical (Lazard's lemma). This gives another optimization of the rule R2.3 when $j = k$. In this case, s is necessarily invertible modulo (A_h) and can be removed from S.
- A good strategy consists in applying **R2** as soon as possible in the case both **R1** and **R2** apply. Indeed, this may split the systems into smaller systems which are easier to handle.

7.1 Experiments

Some other algorithms which have the same specifications as regCharacteristic are implemented in Maple [8] and [15] but are superseded by ours: the first step of the other algorithms consists in computing a Gröbner basis of $(A) : S^\infty$. This is certainly not the best way since the Buchberger algorithm does not take into account the fact that A is already triangular and explicits the inverses of the inequations, which are not needed.

The advantage of our method only appears for differential systems which make Rosenfeld–Gröbner spend a lot of time on the purely algebraic treatment of regular differential systems.

Let's try the following system which has no physical significance but makes Rosenfeld–Gröbner spend most of the time in the purely algebraic part (second step):

$$v\,u_{xx} + u_{xx}^2 + u_x, \quad u_{yy} + u_y$$

for the orderly ranking

$$\cdots > u_{xx} > u_{xy} > u_{yy} > v_{xx} > v_{xy}$$
$$> v_{yy} > u_x > u_y > v_x > v_y > u > v.$$

We perform our comparisons in Maple 5.5 over a Sun Ultra 5 at 333Mhz with 128Mb memory.

Our private version of Rosenfeld–Gröbner produces six regular differential ideals in 75 seconds, including 62 seconds for regCharacteristic. One of the regular differential systems $A = 0,\ S \neq 0$ to deal with is quite large. We do not give its characteristic presentation.

The other implementations of Rosenfeld–Gröbner (the one in Maple 5.5 and its variant by Hubert) cannot carry this example out: Maple 5.5 does not succeed in computing the Gröbner basis of the localized ideal $S^{-1}(A)$.

8. CANONICAL FORMS

Let C be the characteristic presentation of the differential ideal $\mathfrak{a} = [C] : H_C^\infty$ in the differential ring $R = K\{U\}$. Denote L the set of leaders of C and N the set of the other derivatives occuring in C.

To any differential polynomial q we may associate a fraction

$$\mathrm{NF}(q, C) = \frac{a}{b}$$

satisfying

1. $b\,q = a \mod \mathfrak{a}$,
2. $\mathrm{NF}(q, C)$ is a canonical form of the equivalence class of q in $R/\mathfrak{a}$ (the fraction only depends on the ranking, the differential ideal $\mathfrak{a}$ and the equivalence class of q). In particular
$$q = q' \mod \mathfrak{a} \quad \Rightarrow \quad \mathrm{NF}(q, C) = \mathrm{NF}(q', C).$$
3. $a \in K[L, N]$ is reduced w.r.t. C and $b \in K[N]$ (in particular b does not divide zero in $R/\mathfrak{a}$).

The following proposition is straightforward but very important for it permits us to perform easily linear algebra over K in $R/\mathfrak{a}$.

Proposition 7. *The set* $\{\mathrm{NF}(q, C) \mid q \in R\}$ *forms a vector space over* K.

We now show how to compute normal forms. First consider the case $q \in K[L, N]$ for which things are easy since C is strongly normalized.

Let $\overline{q} = \mathsf{prem}(q, C)$. For some power product h of initials of C we have $h\,q = \overline{q} \mod (C)$. We define $\mathrm{NF}(q, C)$ as the fraction a/b obtained by making $\overline{q}/h$ irreducible. The differential polynomial $\overline{q}$ is reduced w.r.t. C. Since C is strongly normalized, $h \in K[N]$.

Lemma 7. $b\,q = a \mod \mathfrak{a}$.

Proof. We have $h\,q = \overline{q} \mod \mathfrak{a}$. Factors of h are not zero divisors in $R/\mathfrak{a}$ thus we may factor them out from the above relation. □

Lemma 8. $\mathrm{NF}(q, C) = \mathrm{NF}(q', C)$ *for any* $q' \in K[L, N]$ *such that* $q = q' \mod \mathfrak{a}$.

Proof. Denote $\mathrm{NF}(q', C) = a'/b'$. The differential polynomial $a\,b' - a'\,b \in \mathfrak{a}$, is reduced w.r.t. C for b, $b' \in K[N]$ and a, a' are reduced w.r.t. C. It is zero for C is a characteristic set of $\mathfrak{a}$. Since the fractions are reduced $a = a'$ and $b = b'$. □

We now consider the general case. The problem is due to the separants which do not belong to $K[N]$. It is overruled by inverting them using the extended Euclidean algorithm.

Let q be a differential polynomial. Let $\overline{q} = \mathsf{partial\_rem}(q, C)$. For some power product h of separants of elements of C we have $h\,q = \overline{q} \mod \mathfrak{a}$.

Using the invert algorithm, we may compute[10] the inverse of h modulo $(C) : H_C^\infty$ in $K(N)[L]$. Multiplying by some element of $K[N]$ to clear denominators we find a differential polynomial $\overline{h}$ such that $h\,\overline{h} = g \mod \mathfrak{a}$ and $g \in K[N]$. Therefore $g\,q = \overline{h}\,\overline{q} \mod \mathfrak{a}$. Let $\mathrm{NF}(\overline{h}\,\overline{q}, C) = \overline{a}/\overline{b}$. We define $\mathrm{NF}(q, C)$ to be the fraction a/b obtained by making the fraction $\overline{a}/g\,\overline{b}$ irreducible.

Proofs of the general case are easy variants of the former ones.

Canonicity properties are consequences of theorem 3 and lemma 8.

CONCLUSION

The algorithm presented in this paper is a first step in merging the efficient solvers of polynomial equations implemented by [25, 1] and the Rosenfeld–Gröbner algorithm. Merging these solvers, we expect to obtain an efficient solver for differential polynomial equations in which there would not be any distinction between differential and algebraic parts. In particular, such a new solver would be able to handle the purely algebraic subproblems which also arise in the differential part of the current Rosenfeld–Gröbner. It would be very interesting for systems of DAE and PDAE.

We would like to thank Marc Moreno Maza for many explanations about his algorithms and triangular sets. We would like to thank both Marc Moreno Maza and François Ollivier for their comments on a preliminary version of this paper.

9. REFERENCES

[1] P. Aubry. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom.* PhD thesis, Université Paris VI, 1999.

[2] P. Aubry, D. Lazard, and M. M. Maza. On the theories of triangular sets. *Journal of Symbolic Computation*, 28:105–124, 1999.

[3] T. Becker and V. Weispfenning. *Gröbner Bases: a computational approach to commutative algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer Verlag, 1991.

[4] E. Bézout. *Cours de mathématiques à l'usage des Gardes du Pavillon et de la Marine*, volume III. Musier, Paris, 1766.

[5] F. Boulier. *Étude et implantation de quelques algorithmes en algèbre différentielle.* PhD thesis, Université Lille I, 59655, Villeneuve d'Ascq, France, 1994.

[6] F. Boulier. Efficient computation of regular differential systems by change of rankings using Kähler differentials. Technical report, Université Lille I, 59655, Villeneuve d'Ascq, France, November 1999. (ref. LIFL99-14, submitted to MEGA2000).

[7] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Representation for the radical of a finitely generated differential ideal. In *proceedings of ISSAC'95*, pages 158–166, Montréal, Canada, 1995.

[10] Observe that, though h is invertible mod $(C) : H_C^\infty$ the invert algorithm may fail to compute the inverse. In that case however a splitting of the ideal is exhibited and computations can be restarted over each of its branches. One avoids this problem by precomputing the inverses of the separants of the elements of C

[8] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Computing representations for radicals of finitely generated differential ideals. Technical report, Université Lille I, LIFL, 59655, Villeneuve d'Ascq, France, 1997. (technical report IT306 of the LIFL, available at `http://www.lifl.fr/~boulier`).

[9] D. Bouziane, A. Kandri Rody, and H. Maârouf. Unmixed–Dimensional Decomposition of a Finitely Generated Perfect Differential Ideal. *Journal of Symbolic Computation*, 1996. (submitted).

[10] B. Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero–Dimensional Polynomial Ideal (German)*. PhD thesis, Math. Inst. Univ. of Innsbruck, Austria, 1965.

[11] J. D. Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *Proceedings of EUROCAL85, vol. 2*, volume 204 of *Lecture Notes in Computer Science*, pages 289–290. Springer Verlag, 1985.

[12] L. Ducos. Optimizations of the subresultant algorithm. *Journal of Pure and Applied Algebra*, 1998. (to appear).

[13] D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry*, volume 150 of *Graduate Texts in Mathematics*. Springer Verlag, 1995.

[14] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of Gröbner bases by change of orderings. *Journal of Symbolic Computation*, 16:329–344, 1993.

[15] É. Hubert. Factorization free decomposition algorithms in differential algebra. *Journal of Symbolic Computation*, 1999. (to appear).

[16] M. Kalkbrener. A Generalized Euclidean Algorithm for Computing Triangular Representations of Algebraic Varieties. *Journal of Symbolic Computation*, 15:143–167, 1993.

[17] D. E. Knuth. *The art of computer programming.* Addison–Wesley, 1966.

[18] E. R. Kolchin. *Differential Algebra and Algebraic Groups*. Academic Press, New York, 1973.

[19] D. König. *Theorie der endlichen und unendlichen Graphen*. Chelsea publ. Co., New York, 1950.

[20] D. Lazard. Solving Zero–dimensional Algebraic Systems. *Journal of Symbolic Computation*, 13:117–131, 1992.

[21] Z. Li and D. Wang. Coherent, regular and simple systems in zero decompositions of partial differential systems. *Systems Science and Mathematical Sciences*, 12:43–60, 1999.

[22] H. Lombardi, M.-F. Roy, and M. Safey El Din. New structure theorem for subresultants. (preprint), 1999.

[23] H. Maârouf. *Étude de Quelques Problèmes Effectifs en Algèbre Différentielle*. PhD thesis, Université Cadi Ayyad, Morocco, 1996.

[24] M. M. Maza and R. Rioboo. Polynomial gcd computations over towers of algebraic extensions. In *Proceedings of AAECC11*, pages 365–382. Springer Verlag, 1995.

[25] M. Moreno Maza. *Calculs de Pgcd au–dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Équations Algébriques*. PhD thesis, Université Paris VI, France, 1997.

[26] S. Morrison. The Differential Ideal $[P] : M^{\infty}$. *Journal of Symbolic Computation*, 28:631–656, 1999.

[27] F. Ollivier. *Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité*. PhD thesis, École Polytechnique, 91128, Palaiseau, France, 1990.

[28] J. F. Ritt. *Differential Algebra*. Dover Publications Inc., New York, 1950.

[29] A. Rosenfeld. Specializations in differential algebra. *Trans. Amer. Math. Soc.*, 90:394–407, 1959.

[30] J. Schicho and Z. Li. A construction of radical ideals in polynomial algebra. Technical report, RISC, Johannes Kepler University, Linz, Austria, august 1995.

[31] A. Seidenberg. An elimination theory for differential algebra. *Univ. California Publ. Math. (New Series)*, 3:31–65, 1956.

[32] D. Wang. Decomposing polynomial systems into simple systems. *Journal of Symbolic Computation*, 25:295–314, 1998.