# Computing differential characteristic sets by change of ordering

François Boulier

*Université Lille 1, LIFL, 59655 Villeneuve d'Ascq France*

François Lemaire

*Université Lille 1, LIFL, 59655 Villeneuve d'Ascq France*

Marc Moreno Maza

*University of Western Ontario, ORCCA, London Canada*

**Abstract**

We describe an algorithm for converting a characteristic set of a prime differential ideal from one ranking into another. This algorithm was implemented in many different languages and has been applied within various software and projects. It permitted to solve formerly unsolved problems.

*Key words:* differential algebra; ranking; PARDI; characteristic sets

## Introduction

*Description.* In this paper, we describe an algorithm which solves the following problem: given a characteristic set $C$ of a prime differential ideal $\mathfrak{p}$ w.r.t some ranking $\mathscr{R}$ and another ranking $\overline{\mathscr{R}} \neq \mathscr{R}$, compute a characteristic set $\overline{C}$ of $\mathfrak{p}$ w.r.t. $\overline{\mathscr{R}}$.

The proposed algorithm, called [1] PARDI applies for systems of partial differential polynomial equations. It specializes to systems of ordinary differential polynomial equations

*Email addresses:* Francois.Boulier@lifl.fr (François Boulier), Francois.Lemaire@lifl.fr (François Lemaire), moreno@scl.csd.uwo.ca (Marc Moreno Maza).

*URLs:* http://www.lifl.fr/~boulier (François Boulier), http://www.lifl.fr/~lemaire (François Lemaire), http://www.csd.uwo.ca/~moreno (Marc Moreno Maza).

[1] PARDI is an acronym for Prime pARtial Differential Ideal. In French, "*pardi*" is an oldfashioned swearword such as, say, "*egad*" in English.

and is then called [2] PODI. It specializes to nondifferential polynomial equations where it is called [3] PALGIE.

This article describes an algorithm really designed for applications. Indeed, since its first presentation by Boulier et al. (2001), its different variants were implemented and involved in various applications, that are described in section 4.

*Previously done work on the problem.* As far as we know, Ollivier was the first to solve the problem addressed in this paper. Let's quote (Ollivier, 1990, page 95): "one can [design] a method for constructing a characteristic set of a finitely generated prime differential ideal as soon as one can effectively test membership to this ideal". An algorithm is given in SCRATCHPAD in (Ollivier, 1990, page 97). In most approaches, a known characteristic set provides the membership test algorithm. This functionality was afterwards implemented in the MAPLE *diffalg* package by the first author. The implemented algorithm handles differential ideals given by characteristic sets which do not need to be prime. Such a problem was also considered by Boulier (1999). However, the algorithms presented in (Boulier, 1999) compute differential polynomials which are not necessarily part of the desired characteristic set but only help computing it. They are complementary to PARDI. The problem was also addressed by (Bouziane et al., 2001, section 3.2). Their algorithm does not make use of the primality hypothesis. It computes a representation of the prime differential ideal as an intersection of differential ideals presented by characteristic sets. The desired characteristic set can then easily be picked from these latter by a dimension argument. Their algorithm relies on a test of algebraic invertibility modulo triangular systems (so ours does) but they perform it by means of Gröbner bases computations. The non differential case was addressed by Dahan et al. (2006). In the ordinary differential context, Golubitsky (2004) developed an approach based on the Gröbner walk idea while Golubitsky et al. (2007) make use of a bound for reducing the problem to the non differential case. Last, a former version of this paper was published by Boulier et al. (2001). As far as we know, it is the only published paper addressing our problem in the context of partial differential equations.

*New results.* The version of PARDI given in this paper is different from that given in (Boulier et al., 2001). It is closer to the variants that the authors really implemented. A first difference is that, in the PDE context, the set of critical pairs is more carefully handled. A new criterion for avoiding some of them is given in section 3.3.1. Observe that this criterion does not only apply to PARDI but to all the characteristic sets decomposition algorithms which apply for PDE systems. Avoiding critical pairs is known to be a crucial issue in the context of Gröbner bases, which led to major recent improvements of the Buchberger algorithm. The same must be true for PDE simplifiers also.

Another major difference is the fact that the set of the already processed equations is kept to be a regular chain. All the variants of PARDI are concerned by this improvement. Indeed, in every realistic implementation of any polynomial system simplifier, the equations produced by the computations always need to be cleaned before they can be used for simplifying the following ones. Maintaining the set of the already processed equations as a regular chain makes unfortunately the proofs quite complicated. To illustrate the

[2] PODI is an acronym for Prime Ordinary Differential Ideal.

[3] PALGIE is an acronym for Prime ALGebraic IdEal. However, since "*algie*" means "suffering" in French, one might also understand PALGIE as "polynomial suffering" say.

complication, consider in the PDE context, the case of the critical pair generated by two already processed equations. Our implementation manages to process this critical pair only once. However, the two already processed equations may be dramatically modified at some further computational step because they are part of a regular chain. When this happens, it is actually not necessary to generate a new critical pair between the two modified equations. This fact is not obvious. We prove it in Lemma 18.

Our approach offers several other advantages. It identifies the algebraic subproblems which occur in the differential computations and solves them by a purely algebraic method. This improves the control of the coefficients growth and avoids many useless computations only due to differential considerations. This very important advantage w.r.t. all other approaches permits us to handle some unsolved problems. A last contribution is the conceptual simplicity of our algorithm, which contrasts with the high technicity of its implementation. Everybody knows that the common roots of two univariate polynomials over a field are given by their gcd. Our algorithm applies this very simple idea and replaces any two univariate polynomials by one of their gcd over the fraction field of some quotient ring. This makes much more sense than speaking of full remainders as in the previous approaches. Some methods for computing triangular decompositions of arbitrary ideals (prime or not) are also explicitly formulated in terms of gcd (Kalkbrener, 1993; Lazard, 1991; Moreno Maza, 2000). The use of the gcd made by these methods is however more complicated than that made by PARDI. Indeed in these methods the ideal modulo which the gcd computations are performed has to change during the triangular decomposition, since it depends on the equations already processed. This is not the case in our particular context.

*Remark.* To simplify and shorten this paper, the description of the final purely algebraic treatment is omitted. This version of PARDI thus returns a regular differential system instead of a characteristic set. A description of the missing algorithms can be found in (Boulier et al., 2001). The interested reader may also find them in the source code of the BLAD libraries (function bad_reg_characteristic_quadruple in bad/src).

## 1. Intuitive presentation

This section is dedicated to casual readers. The problem addressed by PARDI is presented at a very intuitive level. Consider the following polynomial equation and denote $\mathfrak{A}$ the ideal that it generates, together with some other non displayed equations:

$$u\,v - w = 0.$$

In the non differential case, polynomial systems simplifiers can be divided in two families, depending on the way they interpret the equation as a rewrite rule: either

$$u\,v \to w \quad \text{or} \quad u \to \frac{w}{v}.$$

In both cases, an ordering (a ranking in the differential context) is required to select the monomial or the variable (more precisely, the rank) which appear on the lefthand sides of the rewrite rules.

The first set of methods transforms polynomials into polynomials and leads to the Gröbner bases theory. Whenever the lefthand sides monomials of two different rules have a nontrivial gcd, a critical pair is generated. When the completion process, which aims

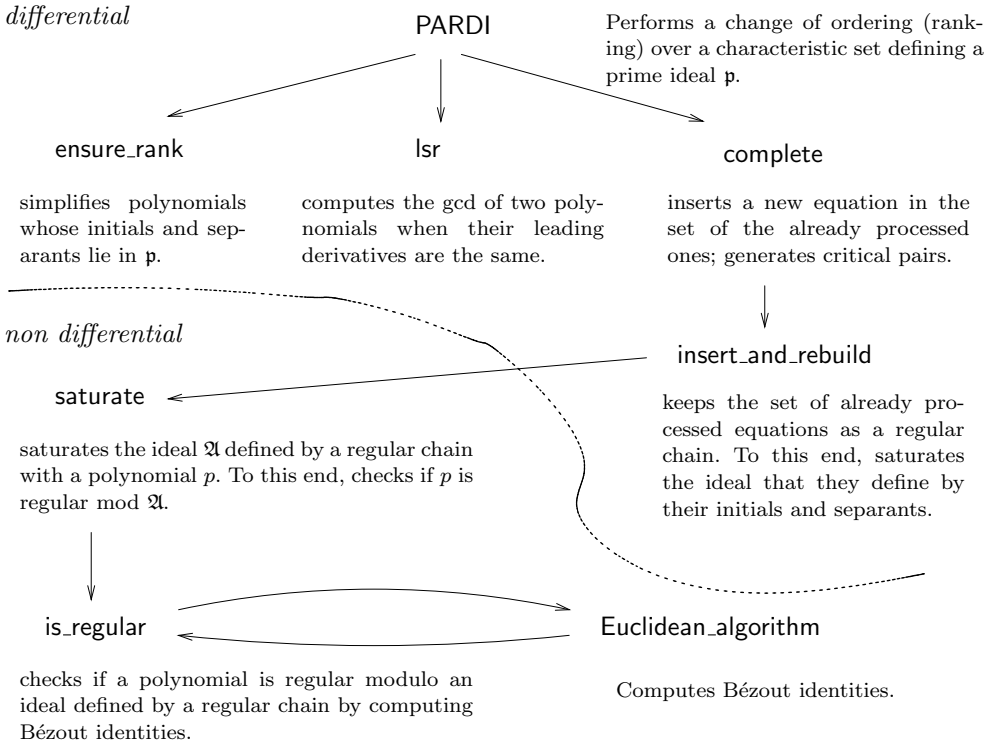


Fig. 1. Dependency graph for PARDI. $A \to B$ means that function $A$ calls function $B$.

at solving the critical pairs, is over, the ideal $\mathfrak{A}$ gets represented by one, possibly large, generating set: a Gröbner basis of $\mathfrak{A}$.

The second set of methods transforms polynomials into rational fractions and causes splittings: the simplifiers handle separately the solutions of the input system which annihilate the denominators of the rewrite rules from the solutions which do not annihilate them. The input system actually gets rewritten as finitely many small systems. Each small system can be associated to some ideal $\mathfrak{B}_k$ and, by means of the ideal-variety correspondence, one gets a representation

$$\mathfrak{A} = \bigcap_k \mathfrak{B}_k.$$

Observe that nontrivial decomposition arise also when $\mathfrak{A}$ is a prime ideal. In this case however, all but one of the $\mathfrak{B}_k$ need to be redundant and should not be generated.

The PARDI algorithm addresses this issue. It assumes that $\mathfrak{A}$ is prime and that membership testing in $\mathfrak{A}$ is algorithmic from the very beginning of the computations. In this case, it is possible to avoid splitting cases: if the denominator of the rewrite rule under consideration does not lie in $\mathfrak{A}$ then the study of the solutions of $\mathfrak{A}$ which annihilate the denominator only leads to redundant ideals $\mathfrak{B}_k$.

In the non differential context, the denominators are the initials of the polynomials. In the differential one, differential equations can be differentiated and the initials of the differentiated equations, which are the separants of the equation, need to be considered

also. In the particular case of partial differential equations, critical pairs arise whenever the leading derivatives, i.e. the lefthand sides of two different rewrite rules, have common derivatives. A completion process pretty similar to that of the Gröbner bases theory must then moreover be implemented.

Observe that the PARDI hypotheses permit to transform a tree exploring algorithm into an iterative algorithm: backtrack implementation (which requires the management of a set of systems to be processed and the duplication of lists of critical pairs at each tree node) is avoided. This feature is very important for tackling real size problems.

*Organization of the paper.* Section 2 is dedicated to the subalgorithms of PARDI which address a typical non differential issue, related to the regular chains theory. It terminates with the presentation of saturate. Section 3 is devoted to the subalgorithms which address differential issues. It culminates with the presentation of PARDI. Applications are given in section 4. The presentation of the algorithms is thus bottom-up. For readers, the advantage is that proofs should be easier to follow. The drawback is that subalgorithms must be understood outside their context but Figure 1 should attenuate it. Each algorithm is presented by a pseudocode in a figure plus two propositions. The first proposition proves the termination. The assumptions on formal parameters and an intuitive description of what each function does are given with the pseudocode. The true specifications of the functions (the ones which are needed for writing proofs) are described and proved separately, in the second proposition.

## 2. The non differential part of PARDI

### *2.1. General definitions and notations*

#### *2.1.1. Computer science*

**Definition 1.** A while loop invariant is a property which holds each time the loop condition is evaluated.

Loop invariants are very important for they permit to prove the correctness of algorithms: they hold in particular when the loop condition evaluates to false i.e. when the loop terminates. Combined to the negation of the loop condition, they give the properties of the datas computed by the loop.

#### *2.1.2. Polynomials*

Let $X$ be an ordered alphabet (possibly infinite).

Let $R = K[X]$ be a polynomial ring where $K$ is a field of characteristic zero. Let $p \in R \setminus K$ be a polynomial. If $x \in X$ is any indeterminate then the *leading coefficient* of $p$ viewed as a univariate polynomial in $x$ (with coefficients in the ring $K[X \setminus \{x\}]$) is denoted $\mathsf{lcoeff}(p, x)$. If $\deg(p, x) = 0$ then $\mathsf{lcoeff}(p, x) = p$. 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 initial of $p$ is the leading coefficient of $p$ w.r.t. its leader). The *rank* of $p$ is the monomial $x^d$. The *reductum* of $p$ is the polynomial $p - i_p\, x^d$. 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 $s_p = \partial p / \partial x$.

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. One denotes $H_A = I_A \cup S_A$. The set $A$ is said to be *triangular* if its elements have distinct leaders.

Let $q$ be a polynomial. One denotes $\mathsf{pquo}(q, p, x)$ and $\mathsf{prem}(q, p, x)$ the pseudoquotient and the pseudoremainder (von zur Gathen and Gerhard, 1999, Section 6.12) of $q$ by $p$, viewed as univariate polynomials in $x$. If $x$ is omitted, both polynomials are viewed as univariate polynomials in the leader of $p$. One denotes $\mathsf{prem}(q, A)$ "*the*" pseudoremainder $r$ of $q$ by all the elements of $A$ i.e. any polynomial $r$ obtained from $q$ and the elements of $A$ by performing successive pseudoreductions and such that $\mathsf{prem}(r, p) = r$ for every $p \in A$. Without further precisions, $r$ is not uniquely defined. Fix any precise algorithm. By convention, one defines $\mathsf{prem}(q, \emptyset) = q$.

If $A$ is a subset of a ring $R$ then $(A)$ denotes the ideal generated by $A$. By convention, one defines $(A) = (0)$ when $A$ is empty. Let $\mathfrak{A}$ be an ideal of $R$. If $S = \{s_1, \ldots, s_t\}$ then the *saturation* $\mathfrak{A} : S^\infty$ of $\mathfrak{A}$ by $S$ is the ideal $\mathfrak{A} : S^\infty = \{p \in R \mid \exists a_1, \ldots, a_t \in \mathbb{N} \text{ such that } s_1^{a_1} \cdots s_t^{a_t}\, p \in \mathfrak{A}\}$. By convention, one defines $\mathfrak{A} : S^\infty = \mathfrak{A}$ if $S$ is empty.

*2.1.3. Regular chains*

In this section, one considers a triangular set $A = \{p_1, \ldots, p_n\}$ of a polynomial ring $R$. Renaming the indeterminates if needed, one assumes that $R = K[t_1, \ldots, t_m, x_1, \ldots, x_n]$ and that $\mathrm{ld}\, p_i = x_i$ for each $1 \leq i \leq n$. One assumes $x_1 < \cdots < x_n$. Denote $A_i$ the triangular set $\{p_1, \ldots, p_i\}$. Denote $R_i$ the ring $K[t_1, \ldots, t_m, x_1, \ldots, x_i]$. Denote $R_{0,i}$ the ring $K(t_1, \ldots, t_m)[x_1, \ldots, x_i]$. Denote $\mathfrak{A}_i$ the ideal $(A_i) : I_{A_i}^\infty$ of $R_i$ and $\mathfrak{A}_{0,i}$ the ideal $(A_i) : I_{A_i}^\infty$ of $R_{0,i}$. Denote $R_0 = R_{0,n}$ and $\mathfrak{A} = \mathfrak{A}_n$. Let us recall the following key lemma. It permits to conduct proofs and state algorithms in the zerodimensional setting instead of the positive dimension one. Argumenting in the zerodimensional setting is much simpler. All the following lemmas recall "well-known" theorems on triangular sets and regular chains.

**Lemma 2.** *An element $a$ in $R/\mathfrak{A}$ is zero (resp. regular) if and only if, for every nonzero $b \in K[t_1, \ldots, t_m]$, the element $a/b$ in $R_0/\mathfrak{A}_0$ is zero (resp. regular).*

**Proof.** (Boulier et al., 2006, Theorem 1.1). □

Regular chains are defined in (Aubry et al., 1999). See also (Kalkbrener, 1993; Lazard, 1991). We adopt the next definition (Boulier et al., 2006, Definition 3.1).

**Definition 3.** The set $A$ is a *regular chain* if, for each $2 \leq \ell \leq n$, the initial of $p_\ell$ is regular in the ring $R_{\ell-1}/\mathfrak{A}_{\ell-1}$. Assume $A$ is a regular chain. Then $A$ is said to be *squarefree* if, for each $1 \leq \ell \leq n$, the separant of $p_\ell$ is regular in $R_\ell/\mathfrak{A}_\ell$.

**Lemma 4.** *(regular chains decide membership in the ideals that they define)*
*If $A$ is a regular chain then, for each $a \in R$ we have $a \in \mathfrak{A}$ if and only if $\mathsf{prem}(a, A) = 0$.*

**Proof.** See (Aubry et al., 1999, Theorem 6.1), (Aubry, 1999, théorème 4.6.1) or (Boulier et al., 2006, Proposition 3.7). □

**Lemma 5.** *Let $A$ be a regular chain and $1 \leq i \leq n$ be an index. Then $A_i$ is a regular chain and $\mathfrak{A}_i = \mathfrak{A} \cap R_i$. If moreover $A$ is squarefreee then so is $A_i$.*

**Proof.** The fact that $A_i$ is a (squarefree) regular chain if $A$ is so follows from the very definition of regular chains. By Lemma 4 the set of the polynomials of $R_i$ reduced to zero by $A_i$ is $\mathfrak{A}_i$. By Lemma 4, the set of the polynomials of $R_i$ reduced to zero by $A$ is $\mathfrak{A} \cap R_i$. The reduction to zero by $A$ of an element of $R_i$ only involves polynomials of $A_i$. The two sets are thus equal and $\mathfrak{A}_i = \mathfrak{A} \cap R_i$. □

**Lemma 6.** *(corollary to Lazard's lemma)*
*If $A$ is a squarefree regular chain then the ideals $\mathfrak{A}$ and $\mathfrak{A}_0$ are radical.*

**Proof.** (Boulier et al., 1995, Lemma 2) or (Boulier et al., 2006, Corollary 3.3). □

**Lemma 7.** *If $A$ is a squarefree regular chain then $\mathfrak{A} = (A) : H_A^\infty$.*

**Proof.** By Lemma 6 and (Hubert, 2000, Proposition 3.3). □

Observe that these properties still hold if one enlarges the $t$'s with some extra indeterminates which do not occur in $A$. They even hold if the set of the $t$'s is infinite.

### *2.2. Algorithms*

This section is dedicated to the functions is_regular, Euclidean_algorithm and saturate. One keeps the previously introduced notations. In this section, $A$ is assumed to be a regular chain and one denotes $\mathfrak{p}$ a prime ideal containing $\mathfrak{A}$. One assumes moreover that the initials of the elements of $A$ do not lie in $\mathfrak{p}$ and that membership testing in $\mathfrak{p}$ is algorithmic.

Let us explain a bit the relationship between this section and the rest of the paper. The ideal $\mathfrak{p}$ actually is the differential prime ideal passed to PARDI. Membership testing in $\mathfrak{p}$ is performed by means of the known characteristic set $C$ of $\mathfrak{p}$. The set $A$ is actually the set (or a subset) of the already processed differential equations of PARDI. At each loop, PARDI introduces a new differential polynomial in $A$. To keep $A$ as a (squarefree) regular chain, it is necessary to saturate $A$ by the initial and the separant of the new differential equation. This task is devoted to saturate. To achieve it, saturate needs to check the regularity of this initial or separant (called $p$) modulo the ideal $\mathfrak{A}$ defined by $A$. Regularity checking is performed by is_regular and Euclidean_algorithm. Observe that, as long as the separant of the new differential polynomial is not proven regular, the regular chain $A$ cannot be assumed to be squarefree. This complicates a bit the specifications of functions.

The differential polynomial $p$ handled by the three functions may depend on indeterminates (derivatives) different from the leaders of $A$ (indeed, at the beginning of the computations, $A$ is the empty set). One thus defines as $t_1, \ldots, t_m$, the indeterminates different from the leaders of $A$, occuring in $p$ and the elements of $A$. This is implicitly justified by Lemma 2.

Last, recall that at this stage of the paper, one does not need to bother with any differential consideration.

```
function is_regular(p, A = {p_1, ..., p_n}, C)
  Checks that p is regular modulo the ideal 𝔄 defined by A (see section 2.1.3) or exhibits
  a factorization of some p_i
Assumptions
  p is a polynomial of R_0.
  A is a regular chain of R_0 (see section 2.1.3).
  The ideal 𝔄 is included in 𝔭
  The initials of the elements of A and the polynomial p do not lie in 𝔭
  C is a characteristic set of 𝔭
begin
  if p ∈ K(t_1, ..., t_m) then
    return (true, ·)
  else
    let x_ℓ be the leader and i_p be the initial of p
One passes A_{ℓ-1} instead of A to the two functions below,
only to simplify the termination proof
    (bool, g) := is_regular(i_p, A_{ℓ-1}, C)
    if bool then
      (bool, g) := Euclidean_algorithm(p, p_ℓ, x_ℓ, A_{ℓ-1}, C)
    fi
    if not bool then
      return (bool, g)
    elif deg(g, x_ℓ) > 0 then
      return (false, g)
    else
      return (true, ·)
    fi
  fi
end
```

Fig. 2. Function is_regular

*2.2.1. Regularity checking*

This section is dedicated to is_regular and Euclidean_algorithm. Though the proofs and the propositions stated in this section are quite technical, the underlying idea is very simple: it is just a generalization of the well known method to decide whether an integer $a$ is invertible in $\mathbb{Z}/n\mathbb{Z}$ by checking if $\gcd(a, n) = 1$ (von zur Gathen and Gerhard, 1999, Theorem 4.1). If the gcd is different from 1 then a factorization of $n$ is exhibited. The generalization of this idea to triangular sets actually goes back to Moreno Maza and Rioboo (1995).

**Proposition 8.** *(termination)*
*Functions* is_regular *and* Euclidean_algorithm *terminate.*

**Proof.** By induction on the number $n$ of elements of $A$. Basis: $n = 0$. The function is_regular immediately terminates. The function Euclidean_algorithm performs calls to is_regular with $n = 0$. These calls terminate. The loop of the function Euclidean_algorithm

function Euclidean_algorithm($a$, $b$, $x$, $A = \{p_1, \ldots, p_n\}$, $C$)

*If possible, computes a Bézout identity between $a$ and $b$ (indeterminate $x$, coefficients taken modulo the ideal $\mathfrak{A}$ defined by $A$) or exhibits a factorization of some $p_i$*

*Assumptions*

*$A$ is a regular chain.*
*The ideal $\mathfrak{A}$ (defined by $A$) is included in $\mathfrak{p}$*
*The initials of the elements of $A$ do not lie in $\mathfrak{p}$*
*$a$ and $b$ are elements of $(R_0/\mathfrak{A}_0)[x]$*
*$x$ is an indeterminate greater than $x_1, \ldots, x_n$.*
*The leading coefficients of $a$ and $b$ are invertible in $R_0/\mathfrak{A}_0$ and do not lie in $\mathfrak{p}$*
*$C$ is a characteristic set of $\mathfrak{p}$. It is used for memberhip testing in $\mathfrak{p}$*

```
begin
  p := a
  q := b
  while q ≠ 0 do
    r := prem(p, q, x)
    while r ≠ 0 and lcoeff(r, x) ∈ p do
      r := reductum(r, x)
    od
    if r ≠ 0 then
      (bool, h) := is_regular(lcoeff(r, x), A, C)
      if not bool then
        return (bool, h)
      fi
    fi
    p := q
    q := r
  od
  return (true, p)
end
```

Fig. 3. Function Euclidean_algorithm

performs finitely many turns for the degree of $q$ decreases, apart perhaps at the first turn. It thus terminates also.

General case: $n > 0$. One assumes inductively that all calls to functions is_regular and Euclidean_algorithm with $|A| < n$ terminate. The function is_regular performs two calls to these functions with $|A| = \ell - 1 < n$. Thus is_regular terminates for $|A| = n$. The function Euclidean_algorithm performs calls to is_regular with $|A| = n$ which all terminate. Its loop performs finitely many turns for the degree of $q$ decreases. Thus Euclidean_algorithm terminates for $|A| = n$. □

**Proposition 9.** *(specifications of* is_regular *and* Euclidean_algorithm*)*

*The function* is_regular *returns a pair* (bool, $g$) *where* bool *is a boolean and $g$ is a polynomial of $R_0$. If* bool *is* true *then $p$ is invertible in $R_0/\mathfrak{A}_0$. If* bool *is* false *then $g$ is a factor of some $p_\ell \in A$ in the following sense:*

1. *the polynomial* $g$ *has rank* $x_\ell^d$ *for some* $1 \leq \ell \leq n$ *and* $0 < d < deg(p_\ell, x_\ell)$,
2. *there exists a polynomial* $h$ *with leader* $x_\ell$ *s.t.* $g\,h = p_\ell$ *in* $(R_{0,\ell-1}/\mathfrak{A}_{0,\ell-1})[x_\ell]$,
3. *the initial of* $g$ *does not lie in* $\mathfrak{p}$ *and is invertible in* $R_0/\mathfrak{A}_0$.

*The function* Euclidean_algorithm *returns a pair* (bool, $g$) *where* bool *is a boolean and* $g$ *is a polynomial of* $R_0[x]$. *If* bool *is false then* $g$ *satisfies the properties* 1, 2 *and* 3 *stated just above. If* bool *is true then* $g$ *satisfies the following properties:*

4. $g \in (a,\, b)$ *in* $(R_0/\mathfrak{A}_0)[x]$
5. $g$ *is a common divisor of* $a$ *and* $b$ *in* $(R_0/\mathfrak{A}_0)[x]$
6. *the leading coefficient of* $g$ *w.r.t.* $x$ *does not lie in* $\mathfrak{p}$ *and is invertible in* $R_0/\mathfrak{A}_0$.

**Proof.** By induction on the number $n$ of elements of $A$.

Basis: $n = 0$. For is_regular, this corresponds to the case of $p$ being a nonzero element of the field $K(t_1, \ldots, t_m)$. Then $p$ is invertible in $R_0/\mathfrak{A}_0$ and the pair (*true*, $\cdot$) may be returned in all cases. For Euclidean_algorithm, this corresponds to the case of polynomials $a,\, b \in K(t_1, \ldots, t_m)[x]$. The function and its specifications degenerate to that of the usual Euclidean algorithm between polynomials over a field. The pair (*true*, $p$) may always be returned with $p$, being the gcd of $a$ and $b$. Item 4 follows from (von zur Gathen and Gerhard, 1999, Corollary 3.9). Item 5 is well known (von zur Gathen and Gerhard, 1999, Algorithm 3.5). Item 6 is obvious. Thus Proposition 9 is satisfied.

The general case: $n > 0$. One assumes inductively that the results of the calls to is_regular and Euclidean_algorithm with $|A| < n$ satisfy the proposition.

Function is_regular. If any of the calls to is_regular or Euclidean_algorithm returns a pair (*false*, $g$) then this pair may be returned.

Assume thus that both calls to is_regular and Euclidean_algorithm return pairs of the form (*true*, $g$) and consider the one returned by Euclidean_algorithm. By the induction hypothesis, items 4, 5 and 6 are satisfied. For this function call, index $n$ (respectively polynomials $a$, $b$ and variable $x$) in Euclidean_algorithm corresponds to index $\ell-1$ (respectively polynomials $p$, $p_\ell$ and variable $x_\ell$) in is_regular. Two subcases need to be distinguished.

First subcase: $\deg(g,\, x_\ell) > 0$. Item 5 implies items 1 and 2. Since $\deg(g,\, x_\ell) > 0$, the initial of $g$ is equal to the leading coefficient of $g$ w.r.t. $x_\ell$. Thus item 6, combined to (Boulier et al., 2006, Corollary 1.16), implies item 3.

Second subcase: $\deg(g,\, x_\ell) = 0$. Item 4 implies that there exists $\lambda$ and $\mu$ such that $\lambda\, p + \mu\, p_\ell = g$ in the ring $(R_{0,\ell-1}/\mathfrak{A}_{0,\ell-1})[x_\ell]$. Since $\deg(g,\, x_\ell) = 0$, the polynomial $g$ is equal to its leading coefficient w.r.t. $x_\ell$ and, by item 6, one may choose $\lambda$ and $\mu$ such that $\lambda\, p + \mu\, p_\ell = 1$. Since $p_\ell \in \mathfrak{A}_0$, one concludes that $p$ is invertible in $R_0/\mathfrak{A}_0$. The pair (*true*, $\cdot$) may thus be returned.

Function Euclidean_algorithm. If any call to is_regular returns a pair (*false*, $h$) then this pair may be returned. Otherwise, the function behaves as if $R_0/\mathfrak{A}_0$ were a field. The analysis is then similar to that of the basis of the induction. The fact that the leading coefficient of $p$ does not lie in $\mathfrak{p}$ (item 6) is explicitly checked by the function. □

### *2.2.2. Performing saturations*

This section is dedicated to the study of saturate, given in Figure 4. Instead of returning a regular chain defining the ideal $\mathfrak{A} : p^\infty$, saturate returns a regular chain defining an ideal $\overline{\mathfrak{A}}$ which contains $\mathfrak{A} : p^\infty$. This somewhat surprising property is due to the fact that is_regular needs to check the regularity of many different polynomials. Any of these tests

function saturate($A$, $p$, $C$)
*Saturates the ideal $\mathfrak{A}$ defined by $A$ with $p$. Simplifies $A$ at each failure of* is_regular.
*Assumptions*
*$A$ is a regular chain*
*The ideal $\mathfrak{A}$ (defined by $A$) is included in $\mathfrak{p}$*
*The initials and the separants of $A$ do not lie in $\mathfrak{p}$*
*$p$ is a polynomial which does not lie in $\mathfrak{p}$*
*$C$ is a characteristic set of $\mathfrak{p}$*
begin
$\overline{A} := A$
$newS := \emptyset$
$(bool, g) :=$ is_regular$(p, \overline{A}, C)$
while not *bool* do
let $x_\ell$ be the leader of $g$ and denote $h =$ pquo$(p_\ell, g, x_\ell)$
if $g \in \mathfrak{p}$ then
replace $p_\ell$ by $g$ in $\overline{A}$
store $h$, the initial of $g$ and the separant of $g$ in $newS$
else
replace $p_\ell$ by $h$ in $\overline{A}$
store $g$, the initial of $h$ and the separant of $h$ in $newS$
fi
$(bool, g) :=$ is_regular$(p, \overline{A}, C)$
od
return $(\overline{A}, newS)$
end

Fig. 4. Function saturate

may fail and cause a splitting of $A$. Each time a splitting occurs, the function manages to keep a single branch: the one which is contained in $\mathfrak{p}$.

The function also returns a set $newS$ of polynomials which do not lie in $\mathfrak{p}$. The importance of returning these polynomials is going to appear in the proof of Proposition 31.

One keeps the notations and the hypotheses introduced in the previous sections. One assumes moreover that the separants of the elements of $A$ do not lie in $\mathfrak{p}$. If moreover $A$ is squarefree [4] then the ideal $\mathfrak{A}$, which is defined as $(A) : I_A^\infty$ (section 2.1.3), is also equal to $(A) : H_A^\infty$ by Lemma 7. If $\overline{A}$ is a regular chain, denote $\overline{\mathfrak{A}} = (\overline{A}) : I_{\overline{A}}^\infty$.

**Proposition 10.** *(termination)*
*The* saturate *function terminates.*

**Proof.** The fact that $p \notin \mathfrak{p}$ and $\mathfrak{A} \subset \mathfrak{p}$ implies that, at each loop, $\deg(g, x_\ell)$ and $\deg(h, x_\ell)$ are strictly less than $\deg(p_\ell, x_\ell)$. Thus, at each loop, the degree of some element of $\overline{A}$ decreases strictly. The function thus terminates. □

[4] Observe that one may have $\mathfrak{A} \subset \mathfrak{p}$, the separants of the elements of $A$ outside $\mathfrak{p}$ without having $A$ squarefree.

**Lemma 11.** *Consider the* saturate *function. If the first call* is_regular$(p, \overline{A}, C)$ *(with* $A = \overline{A}$*) returns* (false, $g$) *then the sets* $A_g$ *and* $A_h$ *obtained from* $A$ *by replacing* $p_\ell$ *by* $g$ *and (respectively)* $h$ *have the same set of leaders as* $A$ *and form regular chains which satisfy:*

$$\mathfrak{A} \subset (A_g) : I_{A_g}^\infty \cap (A_h) : I_{A_h}^\infty$$

*If moreover* $A$ *is squarefree then so are* $A_g$ *and* $A_h$ *and the inclusion becomes an equality.*

**Proof.** By Proposition 9, the polynomial $g$ is a nontrivial factor of $p_\ell$ with an initial invertible in $R_0/\mathfrak{A}_0$. The sets $A_g$ and $A_h$ correspond to the sets $B$ and $C$ mentioned in (Boulier et al., 2006, Proposition 3.4). The first part of the lemma is a corollary to that proposition. The second part is a corollary to (Boulier et al., 2006, Proposition 3.5). □

**Lemma 12.** *The* saturate *function returns a set* newS *of polynomials which do not lie in* $\mathfrak{p}$ *and a regular chain* $\overline{A}$ *whose initials and separants do not lie in* $\mathfrak{p}$*, having the same set of leaders as* $A$ *and which satisfies:*

$$\mathfrak{A} \subset \overline{\mathfrak{A}} \subset \mathfrak{p}. \tag{1}$$

*If moreover* $A$ *is squarefree then so is* $\overline{A}$*.*

**Proof.** One claims that the properties of $\overline{A}$ and *newS* stated in the lemma are loop invariants of the function. They are satisfied initially. It is sufficient to prove that they are satisfied after one loop.

The fact that $\overline{A}$ is a regular chain (squarefree if so is $A$) having the same set of leaders as $A$ and which satisfies $\mathfrak{A} \subset \overline{\mathfrak{A}}$ follows from Lemma 11. To prove the second inclusion, one still needs to prove that the polynomial $g$ (or $h$) which replaces $p_\ell$ lies in $\mathfrak{p}$ and that its initial does not lie in $\mathfrak{p}$.

If $g$ is inserted in $\overline{A}$ then it lies in $\mathfrak{p}$ (this is explicitly checked by the function). Otherwise, $h$ lies in $\mathfrak{p}$ for this ideal is prime and the product $g\,h$ belongs to it.

The polynomials $p_\ell$, $g$, $h$ have the same leader $x_\ell$ and we have a relation

$$c\,p_\ell = g\,h \mod \mathfrak{p} \tag{2}$$

where $c$ is a power of the initial of $g$. The initial of $g$ does not lie in $\mathfrak{p}$ by item 3 of Proposition 9 thus $c$ does not either for the ideal prime. The initial $i_\ell$ of $p_\ell$ does not lie in $\mathfrak{p}$ (this is one of the assumptions of the function). The initial of $h$ does not either since the ideal is prime and $h$ multiplied by a suitable power of the initial of $g$ is equal to $c\,i_\ell$ modulo $\mathfrak{p}$.

The second inclusion is thus proven. To conclude the proof of the lemma, one still needs to prove that the elements of *newS* do not lie in $\mathfrak{p}$.

The fact that the initials of $g$ and $h$ do not lie in $\mathfrak{p}$ is already proven. Denote $s_\ell$, $s_g$, $s_h$ the separants of $p_\ell$, $g$, $h$. Differentiating relation (2) w.r.t. $x_\ell$ one gets the relation $c\,s_\ell = s_g\,h + g\,s_h \mod \mathfrak{p}$. We have $s_\ell \notin \mathfrak{p}$ (this is one of the assumptions). Thus if $g \in \mathfrak{p}$ then $s_g \notin \mathfrak{p}$. Conversely, if $h \in \mathfrak{p}$ then $s_h \notin \mathfrak{p}$. This proves that the separants of the elements

of $\overline{A}$ do not lie in $\mathfrak{p}$. A similar argument proves that $g$ and $h$ cannot both belong to $\mathfrak{p}$ hence that the elements of *newS* do not lie in this ideal.

This concludes the proof of the lemma. □

**Proposition 13.** *(specification of* saturate*)*

*The* saturate *function computes a set* newS *of polynomials which do not lie in* $\mathfrak{p}$ *and a regular chain* $\overline{A}$ *whose initials and separants do not lie in* $\mathfrak{p}$*, having the same set of leaders as* $A$ *and which satisfies:*

$$\mathfrak{A} \subset \mathfrak{A} : p^\infty \subset \overline{\mathfrak{A}} \subset \mathfrak{p}. \tag{3}$$

*If moreover* $A$ *is squarefree then so is* $\overline{A}$*.*

**Proof.** Relying on Lemma 12, one only needs to prove that relation (3) holds. Relation (1) implies that $\mathfrak{A} : p^\infty \subset \overline{\mathfrak{A}} : p^\infty$. The inclusion $\mathfrak{A} \subset \mathfrak{A} : p^\infty$ is trivial. At the end of the loop execution, $p$ is regular modulo $\overline{\mathfrak{A}}$ and we have $\overline{\mathfrak{A}} = \overline{\mathfrak{A}} : p^\infty$. The proposition follows. □

The next proposition strengthens Proposition 13. This stronger form in needed to prove Lemma 30.

**Proposition 14.** *(stronger specification of* saturate*)*

*Let* $1 \leq i \leq n$ *be an index. Denote* $\overline{A}_i = A \cap R_i$ *and* $\overline{\mathfrak{A}}_i = (\overline{A}_i) : I_{A_i}^\infty$.

*The* saturate *function computes a set* newS *of polynomials which do not lie in* $\mathfrak{p}$ *and a regular chain* $\overline{A}$ *whose initials and separants do not lie in* $\mathfrak{p}$*, having the same set of leaders as* $A$ *and which satisfies:*

$$\mathfrak{A}_\mathfrak{i} \subset \mathfrak{A}_\mathfrak{i} : p^\infty \subset \overline{\mathfrak{A}}_i \subset \mathfrak{p}.$$

*If moreover* $A$ *is squarefree then so is* $\overline{A}_i$*.*

**Proof.** It is a corollary to Proposition 13 and to Lemma 5. □

## 3. The differential part of PARDI

### *3.1. General definitions and notations*

#### *3.1.1. Differential algebra*

Reference books for differential algebra are those of Ritt (1950) and Kolchin (1973). Let us focus on the theory of differential elimination. A reference book is (Wang, 2003). One also refers to Mansfield (1991); Boulier et al. (1995); Reid et al. (1996); Boulier et al. (1997); Hubert (2000); Bouziane et al. (2001); Sit (2002); Hubert (2003). Some packages dedicated to differential elimination are also available: the *diffgrob* package by Mansfield, the *rif* software by Reid and Wittkopf and the *diffalg* package by Boulier, Hubert and Lemaire.

A *derivation* over a ring $R$ is a map $\delta : R \to R$ such that $\delta(a+b) = \delta a + \delta b$ and $\delta(a\,b) = (\delta a)b + a(\delta b)$ for every $a, b \in R$. A *differential ring* is a ring endowed with finitely many derivations which commute pairwise. The commutative monoid generated by the derivations is denoted by $\Theta$. 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 $\theta$, is denoted by $\operatorname{ord}\theta$. 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 $a_i \geq b_i$ for each $1 \leq i \leq m$ 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 $R$ closed under derivation i.e. such that $a \in \mathfrak{A} \Rightarrow \delta a \in \mathfrak{A}$. Let $A$ be a nonempty subset of $R$. One denotes $[A]$ the differential ideal generated by $A$ which is the smallest differential ideal which contains $A$.

*3.1.2. Differential polynomials*

Let $U = \{u_1, \ldots, u_n\}$ be a set of *differential indeterminates*. Derivation operators apply over differential indeterminates giving *derivatives* $\theta u$. One denotes $\Theta U$ the set of all the derivatives. Let $K$ be a differential field. The differential ring of the differential polynomials built over the alphabet $\Theta U$ with coefficients in $K$ is denoted $R = K\{U\}$.

A *ranking* is a total ordering over the set of the derivatives (Kolchin, 1973, chapter I, §8) satisfying the following axioms

(1) $\delta v > v$ for each derivative $v$ and derivation $\delta$,

(2) $v > w \Rightarrow \delta v > \delta w$ for all derivatives $v, w$ and each derivation $\delta$.

Let us fix a ranking. The infinite alphabet $\Theta U$ gets ordered. Consider a polynomial $p \in R \setminus K$. Then the leader, initial, ... 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. It is said to be *partially autoreduced* if its elements are pairwise partially reduced. Autoreduced implies differentially triangular.

**Definition 15.** If $A$ is a set of differential polynomials and $v$ is a derivative then $A_v = \{p \in \Theta A \mid \operatorname{ld} p \leq v\}$.

Thus $R_v$ denotes the set of the differential polynomials with leader less than or equal to $v$.

*3.1.3. Ritt's reduction algorithms*

One distinguishes the partial reduction algorithm, which is denoted partial_rem from the full reduction algorithm, denoted full_rem. Let $q$ and $p$ be two differential polynomials. The *partial remainder* $\mathsf{partial\_rem}(q, p)$ is the pseudoremainder of $q$ by the (infinite) set of all the *proper* derivatives of $p$. The *full remainder* $\mathsf{full\_rem}(q, p)$ is the pseudoremainder of $q$ by the set of all the derivatives of $p$ (including $p$). A precise algorithm is given in (Kolchin, 1973, chapter I, §9). Let $A$ be a set of differential polynomials. One denotes $\mathsf{partial\_rem}(q, A)$ and $\mathsf{full\_rem}(q, A)$ respectively the partial remainder and the full remainder of $q$ by all the elements of $A$.

Let $v = \operatorname{ld} q$ and $\overline{A} = A \cap R_v$.

The partial remainder $\overline{q}$ of $q$ by $A$ 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 $h\,q \equiv \overline{q} \mod (\overline{A}_v)$.

The full remainder $\overline{q}$ of $q$ by $A$ is reduced w.r.t. all the elements of $A$ and there exists a power product $h$ of elements of $H_{\overline{A}}$ such that $h\,q \equiv \overline{q} \mod (\overline{A}_v)$.

*3.1.4. Critical pairs*

A pair $\{p_1, p_2\}$ of differential polynomials is said to be a *critical pair* 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 $\mathrm{ld}\,p_2 = \theta_2 u$). Denote $\theta_{12} u = \mathrm{lcd}(\mathrm{ld}\,p_1, \mathrm{ld}\,p_2)$ the least common derivative of $\mathrm{ld}\,p_1$ and $\mathrm{ld}\,p_2$ defined by $\theta_{12} = \mathrm{lcm}(\theta_1, \theta_2)$.

One distinguishes the *triangular situation* which arises when $\theta_{12} \neq \theta_1$ and $\theta_{12} \neq \theta_2$ from the *nontriangular* one which arises when $\theta_{12} = \theta_2$ (say). In the first case, the critical pair is said to be a *triangular critical pair*. In the last one, it is said to be a *reduction critical pair*. In this article, one does not need to consider the case $\theta_1 = \theta_2$. In the triangular situation, the $\Delta$–polynomial $\Delta(p_1, p_2)$ is

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

In the nontriangular one,

$$\Delta(p_1, p_2) = \mathsf{prem}(p_2, \frac{\theta_2}{\theta_1} p_1).$$

**Definition 16.** If $\{p, p'\}$ is a reduction critical pair with $\mathrm{ld}\,p > \mathrm{ld}\,p'$ then

$$\mathrm{hi}\,(\{p, p'\}) \underset{\mathrm{def}}{=} p, \quad \mathrm{lo}\,(\{p, p'\}) \underset{\mathrm{def}}{=} p'.$$

If $D$ is a list of critical pairs then

$$\mathrm{hi}(D) \underset{\mathrm{def}}{=} \{\mathrm{hi}\,(\{p, p'\}) \mid \{p, p'\} \text{ is a reduction critical pair of } D\}\,.$$

**Definition 17.** A critical pair $\{p, p'\}$ is said to be *solved* by a system $F = 0,\ S \neq 0$ if there exists a derivative $v < \mathrm{lcd}(\mathrm{ld}\,p, \mathrm{ld}\,p')$ such that $\Delta(p, p') \in (F_v) : (S \cap R_v)^\infty$.

In the context of PARDI, the set $F$ to be considered contains some regular chain $A$ and, after a call to saturate, it may happen that some element (say) $p_\ell$ of $A$ gets replaced by one of its factor (say) $g$. Now, the polynomial $p_\ell$ may be involved in some critical pair $\{p_\ell, p_r\}$, considered at some previous stage by PARDI hence solved by $F = 0,\ S \neq 0$. Since $p_\ell$ is replaced by $g$ in $F$, one may wonder if one should not generate and consider the pair $\{g, p_r\}$. In fact, this is not necessary. The following lemma provides the key argument of the proof. For legibility, one only states a simplified version. For a general version, one should simply replace the sentence $p_\ell = g\,h$ in the lemma by the statement given in the item 2 of Proposition 9. Only the triangular case needs to be considered. This lemma is used in the proof of Proposition 31.

**Lemma 18.** *Let $\{p_\ell, p_r\}$ be a triangular critical pair, solved by a differential system $F = 0,\ S \neq 0$. Assume that $p_\ell = g\,h$ with $\mathrm{ld}\,p_\ell = \mathrm{ld}\,g = \mathrm{ld}\,h$. Denote $F' = F \cup \{g\}$ and $S' = S \cup \{h\}$.*

*The critical pair $\{g, p_r\}$ is solved by the differential system $F' = 0,\ S' \neq 0$.*

**Proof.** Denote $\mathrm{ld}\, p_\ell = \theta_\ell\, u$, $\mathrm{ld}\, p_r = \theta_r\, u$ and $\theta_{\ell r} = \mathrm{lcm}(\theta_\ell,\, \theta_r)$. One assumes that $\{p_\ell,\, p_r\}$ is solved by $F = 0,\ S \neq 0$ i.e, denoting $s_\ell$ and $s_r$ the separants of $p_\ell$ and $p_r$, that there exists some $v < \theta_{\ell r}\, u$ such that:

$$s_r\, \frac{\theta_{\ell r}}{\theta_\ell}\, p_\ell - s_\ell\, \frac{\theta_{\ell r}}{\theta_r}\, p_r \in (F_v) : (S \cap R_v)^\infty. \tag{4}$$

Denote $s_g$ and $s_h$ the separants of $g$ and $h$. Since $p_\ell$, $g$ and $h$ have the same leader, one has $s_\ell = s_g\, h + s_h\, g$. In formula (4), replace $s_\ell$ by this expression, $p_\ell$ by $g\, h$ and expand $(\theta_{\ell r}/\theta_\ell)\, (g\, h)$. Using the fact that $F \subset F'$ and $S \subset S'$, replace $F$ by $F'$ and $S$ by $S'$ in the right-hand side of the formula. Using the fact that $g \in F'$, remove from (4), every product of the form $(\varphi\, g)\, (\psi\, h)$ such that $\mathrm{ld}\, \varphi\, g < v$. Remove the term $s_h\, g\, (\theta_{\ell r}/\theta_r)\, p_r$ for it involves $g$ as a factor. One obtains:

$$s_r \left( \frac{\theta_{\ell r}}{\theta_\ell}\, g \right) h - s_g\, h\, \frac{\theta_{\ell r}}{\theta_r}\, p_r \in (F'_v) : (S' \cap R_v)^\infty. \tag{5}$$

The left-hand side of (5) is equal to $h\, \Delta(g,\, p_r)$. Since $h \in S'$, one concludes that $\Delta(g,\, p_r) \in (F'_v) : (S' \cap R_v)^\infty$ i.e. that the critical pair $\{g,\, p_r\}$ is solved by the differential system $F' = 0,\ S' \neq 0$. □

*3.1.5. Characteristic sets*

The traditional definition is due to Ritt: a subset $C$ of a differential ideal $\mathfrak{A}$ is said to be a *characteristic set* of $\mathfrak{A}$ if $C$ is autoreduced and $\mathfrak{A}$ contains no nonzero element reduced w.r.t. $C$.

One adopts in this paper a slightly more general definition, which relinquishes Ritt's autoreduction requirement and was given by Aubry et al. (1999). Their definition, given in the purely algebraic setting readily lifts to the differential one.

**Definition 19.** A subset $C$ of a differential ideal $\mathfrak{A}$ is said to be a *characteristic set* of $\mathfrak{A}$ if $C$ is differentially triangular, the initials of the elements of $C$ are not reduced to zero by $C$ (by Ritt's full reduction algorithm) and $\mathfrak{A}$ contains no nonzero element reduced w.r.t. $C$.

Every characteristic set in the sense of Ritt is a characteristic set in the sense of Aubry et al. (1999). Conversely, if $C$ is a characteristic set in the sense of Aubry et al. (1999), it can be made autoreduced by pseudoreducing each of its elements by the other ones. This autoreduction process does not change the rank of $C$ since it is required that the initials of the elements of $C$ are not reduced to zero by $C$. Every theorem about Ritt's characteristic sets which only relies on rank considerations therefore applies to the more general definition. The following proposition provides a useful example. It slightly generalizes well known results on characteristic sets since $C$ is not assumed to be autoreduced.

**Proposition 20.** *If $C$ is a characteristic set of $\mathfrak{A}$ and $H_C$ contains no zero divisor in the factor ring $R/\mathfrak{A}$ then $\mathfrak{A} = [C] : H_C^\infty$ and $p \in \mathfrak{A}$ if and only if* $\mathsf{full\_rem}(p,\, C) = 0$. *This is the case when $\mathfrak{A}$ is prime.*

Let $G = \langle A,\ D,\ P,\ S\rangle$ be a quadruple.
**I1** $\mathfrak{p} = I(G)$ ;
**I2** the set $A$ is a partially autoreduced squarefree regular chain ;
**I3** every critical pair made of elements of $A$ is nearly solved by $G$ ;
**I4** the initials and separants of the elements of $A$ and of the critical pairs of $D$ belong to $S$ ;
**I5** if $\{p,\ p'\} \in D$ is a reduction pair such that $p' = \mathrm{lo}\,(\{p, p'\})$ then $p' \in I^{\mathrm{ld}\,p'}(G)$.

Fig. 5. Invariant properties kept by PARDI

**Proof.** Let $p$ be a differential polynomial and denote $r = \mathsf{full\_rem}(p,\ C)$. Assume $p \in \mathfrak{A}$. Since $C \subset \mathfrak{A}$ one has $r \in \mathfrak{A}$. The remainder $r$ is reduced w.r.t. $C$. It is thus zero. This proves $\mathfrak{A} \subset [C] : H_C^\infty$.

Assume $p \in [C] : H_C^\infty$. Then $h\,p \in [C] \subset \mathfrak{A}$ where $h$ denotes some power product of initials and separants of $C$. Since $h$ does not divide zero modulo $\mathfrak{A}$, one has $p \in \mathfrak{A}$ hence $[C] : H_C^\infty \subset \mathfrak{A}$. Combining the two inclusions, $\mathfrak{A} = [C] : H_C^\infty$ is proven.

The first paragraph proves also that $\mathfrak{A}$ is reduced to zero by $C$. Consider now a differential polynomial $p$ reduced to zero by $C$. It belongs to $[C] : H_C^\infty$ and, by the second paragraph above, it belongs to $\mathfrak{A}$. This proves that $p \in \mathfrak{A}$ if and only if it is reduced to zero by $C$.

Assume $\mathfrak{A}$ is prime and $C$ is a characteristic set of $\mathfrak{A}$. Since the initials of $C$ are not reduced to zero by $C$, they do not belong to $\mathfrak{A}$. Since $\mathfrak{A}$ is prime, they are not zero divisors mod $\mathfrak{A}$. One still needs to prove that the separants of $C$ do not belong to $\mathfrak{A}$. If $p$ is an element of $C$, has rank $v^d$ for some derivative $v$ and some $d > 1$ then its separant $s_p$ has rank $v^{d-1}$. Since the initial of $p$ does not lie in $\mathfrak{A}$, the rank of the separant is equal to that of $\mathsf{full\_rem}(s_p,\ C)$. Thus $s_p$ is not reduced to zero by $C$ hence does not lie in $\mathfrak{A}$. If the degree $d = 1$ then the separant is equal to the initial of $p$. It does not lie in $\mathfrak{A}$ either.

This proves that $H_C$ contains no zero divisor in $R/\mathfrak{A}$ when $\mathfrak{A}$ is prime and concludes the proof of the proposition. □

*3.1.6. Quadruples*

The main data structure handled by PARDI is a quadruple $G = \langle A, D, P, S\rangle$. Throughout its execution, PARDI keeps true the properties stated in Figure 5. Roughly speaking, $A$ is the set of the differential polynomial equations already processed, $D$ is the set of the critical pairs to be processed, $P$ is the set of the differential polynomial equations to be processed, $S$ is the set of the differential polynomial inequations ($\neq 0$) already processed. The notation $\mathrm{hi}(D)$ and the expression "solved pair" used in the next definitions are defined in section 3.1.4.

**Definition 21.** Let $G = \langle A, D, P, S\rangle$ be a quadruple and $F = A \cup \mathrm{hi}(D) \cup P$. The system $F = 0,\ S \neq 0$ is called the system *associated* to $G$ and $I(G) = [F] : S^\infty$ is called the differential ideal *associated* to $G$.

**Definition 22.** If $v$ is any derivative and $F = 0,\ S \neq 0$ is a system then $I^v(F, S)$ denotes the algebraic ideal $(F_v) : (S \cap R_v)^\infty$. If $G = \langle A, D, P, S\rangle$ is a quadruple then $I^v(G) \underset{\mathrm{def}}{=} I^v(F, S)$ where $F = 0,\ S \neq 0$ is the system associated to $G$.

**Definition 23.** A critical pair is said to be *solved* by a quadruple $G$ if it is solved by the system associated to $G$.

**Definition 24.** A critical pair $\{p, p'\}$ is said to be *nearly solved* by a quadruple $G$ if it is solved by $G$ or if it lies in $D$.

*3.2. Algorithms applying the "master-student relationship"*

This section is dedicated to the study of functions ensure_rank and lsr. These two algorithms are not really concerned by differential considerations but they apply the so called "master-student relationship" which is formulated in terms of quadruples. We give them here for this reason.

*Master-student relationship.* Recall that quadruples are denoted $\langle A, D, P, S\rangle$. To decide whether a quantity is zero or not modulo $\mathfrak{p}$ one just needs to decide whether this quantity is reduced to zero or not by the "master" $C$ (the known characteristic set of $\mathfrak{p}$). Assume it is. Then one checks if it is also reduced to zero by the "student" $A$ (the characteristic set to be). If it is reduced to zero by $A$ then it is discarded else it is stored in $P$ (the set of equations to be processed, i.e. to be "learned" by the student).

*3.2.1. Ensuring the rank of a differential polynomial*

The function ensure_rank is called by PARDI to ensure that the initial and the separant of the new differential equation to be processed does not lie in $\mathfrak{p}$. The ranking implicitly used is the target ranking $\overline{\mathscr{R}}$.

**Proposition 25.** *(termination)*
*The* ensure_rank *function terminates.*

**Proposition 26.** *(specifications of* ensure_rank*)*
*The* ensure_rank *function returns a pair* (r, newP) *such that* $p \equiv r \mod \mathfrak{p}$*, the initial and the separant of* $r$ *do not lie in* $\mathfrak{p}$ *and* $\text{newP} \subset \mathfrak{p}$.

*3.2.2. The gcd (lsr sorry) of two polynomials over a factor ring*

In this section one studies the function lsr described in Figure 7 which provides an algorithm for computing the gcd (more precisely the last nonzero subresultant) of two polynomials $a$, $b$, in one indeterminate $x$ and coefficients in the field of fractions of a factor ring. lsr is actually called by PARDI when some new differential polynomial $p$ to be inserted in the set $A$ of the already processed equations has the same leader as some element $q$ of $A$. Then $p$ and $q$ are replaced by their gcd, computed with coefficients taken modulo $\mathfrak{p}$. The lsr algorithm is non differential in the sense that it does not manipulate the separants of the polynomials $p$ and $q$ and that it does not generate any critical pair. This is a true major improvement w.r.t. the Rosenfeld–Gröbner algorithm of the MAPLE *diffalg* package.

The ranking implicitly used is the target ranking $\overline{\mathscr{R}}$. One introduces the following notations:

(1) $R^- = K[w \in \Theta U \mid w < x]$
(2) $\mathfrak{p}^- = \mathfrak{p} \cap R^-$
(3) $I^-(G) = (F \cap R^-) : (S \cap R^-)^\infty$ where $F = 0,\ S \neq 0$ denotes the system associated to the current quadruple $G$.

function ensure_rank($p$, $G = \langle A, D, P, S\rangle$, $C$)
*Simplifies $p$ while its initial or its separant lies in $\mathfrak{p}$*
*Assumptions*
  *$p$ is a nonzero differential polynomial which lies in $\mathfrak{p}$*
  *$G$ is a quadruple*
  *$C$ is a characteristic set of the differential prime ideal $\mathfrak{p}$*
begin
  $r := p$
  $newP := P$
*Denote $i_r$ and $s_r$ the initial and the separant of $r$*
  while $r \notin K$ and ($i_r \in \mathfrak{p}$ or $s_r \in \mathfrak{p}$) do
    if $i_r \in \mathfrak{p}$ then
      if prem($i_r$, $A$) $\neq 0$ then
        $newP := newP \cup \{i_r\}$
      fi
      $r :=$ reductum($r$)  *i.e. $r - i_r\, x^d$ where $x^d = \operatorname{rank} r$*
    else
      if prem($s_r$, $A$) $\neq 0$ then
        $newP := newP \cup \{s_r\}$
      fi
      $r := d\, r - s_r\, x$  *where $x^d = \operatorname{rank} r$*
    fi
  od
  return ($r$, $newP$)
end

Fig. 6. The ensure_rank function

Observe that $\mathfrak{p}^-$ is prime, $R^-/\mathfrak{p}^-$ is a domain and $\mathrm{Fr}(R^-/\mathfrak{p}^-)$ is a field.

**Proposition 27.** *(termination)*
*The function* lsr *terminates.*

**Proof.** It is a variant of the Euclidean algorithm. Apart perhaps at the first turn, the degree of $q$ in $x$ strictly decreases at each turn. □

**Proposition 28.** *(specifications of* lsr*)*
*The* lsr *function returns a triple ($g$, $newP$, $newS$) satisfying the properties:*
*(1) $g$ is a gcd of $a$ and $b$ in the ring $\mathrm{Fr}(R^-/\mathfrak{p}^-)[x]$*
*(2) $\deg(g, x) > 0$ and its initial and separant do not lie in $\mathfrak{p}$*
*(3) $(a, b) \subset (g) : h^\infty$ in the ring $(R^-/I^-(G'))[x]$ where $h$ is an element of the multiplicative family generated by $newS$ and $G' = \langle A, D, newP, newS\rangle$.*

The sets $newP$ and $newS$ are updated version of $P$ and $S$ obtained by applying the "master-student relationship" idea.

**Proof.** Observe that the pseudocode of lsr is nothing but the Euclidean algorithm applied on $a$ and $b$ in $\mathrm{Fr}(R^-/\mathfrak{p}^-)[x]$ together with instructions which store in $newP$ every leading

```
function lsr(a, b, x, G = ⟨A, D, P, S⟩, C)
Computes a gcd of a and b viewed as polynomials in x and coefficients modulo 𝔭.
Assumptions
  a, b are polynomials with leader x, partially reduced w.r.t. A
  a, b lie in 𝔭 but their initials and separants do not
  G is a quadruple satisfying properties I1 to I5 given in Figure 5
  C is a characteristic set of the differential prime ideal 𝔭
begin
  p := a
  q := b
  newP := P
  newS := S
  while q ≠ 0 do
    r := prem(p, q, x)
    while r ≠ 0 and lcoeff(r, x) ∈ 𝔭 do
      if prem(lcoeff(r, x), A) ≠ 0 then
        newP := newP ∪ {lcoeff(r, x)}
      fi
      r := reductum(r, x)
    od
    if r ≠ 0 then
      newS := newS ∪ {lcoeff(r, x)}
      p := q
      q := r
    fi
  od
  g := p
  return (g, newP, newS)
end
```

Fig. 7. The lsr function

coefficient which is zero in $R^-/\mathfrak{p}^-$ but not reduced to zero by $A$ and stores in *newS* the "true" leading coefficients of the computed pseudoremainders (among the coefficients in $R^-$, the first one which is nonzero in $R^-/\mathfrak{p}^-$).

Item 1. Therefore, the returned polynomial $g$ is a gcd of $a$ and $b$ in $\mathrm{Fr}(R^-/\mathfrak{p}^-)[x]$ hence item 1 holds.

Item 2. All the computed pseudoremainders belong to the ideal $(a, b)$ of the ring $\mathrm{Fr}(R^-/\mathfrak{p}^-)[x]$. Since $a, b \in \mathfrak{p}$, all the computed pseudoremainders lie in $\mathfrak{p}$ thus the first pseudoremainder which does not depend on $x$, lies in $\mathfrak{p}^-$, hence is zero in $\mathrm{Fr}(R^-/\mathfrak{p}^-)[x]$. This proves that the last nonzero pseudoremainder $g$ satisfies $\deg(g, x) > 0$

For this reason, the leading coefficients w.r.t. $x$ are equal to the initials of the computed pseudoremainders. The function explicitly tests that they do not lie in $\mathfrak{p}$. Thus the initial of $g$ does not lie in $\mathfrak{p}$.

The fact that the separant of $g$ does not lie in $\mathfrak{p}$ is a mere application of the fact that two squarefree univariate polynomials over a field have a squarefree gcd. Let us precise this. Denote $\eta$ a generic zero (Zariski and Samuel, 1958, chapter VI, §5) of $\mathfrak{p}$. It is a zero

of $a$ and $b$ but not a zero of their separants $s_a$ and $s_b$ since these polynomials do not lie in $\mathfrak{p}$. Therefore $\eta$ is a simple zero of $a$ and $b$ hence a simple zero of their gcd $g$. Thus $\eta$ is not a zero of the separant $s_g$ of $g$ and, using the fact that $\eta$ is generic, $s_g \notin \mathfrak{p}$. This concludes the proof of item 2.

Item 3. The computed pseudoremainders sequence is indeed a variant of pseudoremainders sequence computed in $R^-[x]$, where, at each step, some coefficients of the current pseudoremainder are considered as zero. Since the coefficients which are considered as zero are stored in *newP* and the leading of the coefficients which are considered as nonzero are stored in *newS*, the pseudoremainders sequence is computed with coefficients taken modulo $I^-(G')$ i.e. in $(R^-/I^-(G'))[x]$.

Now, if all the leading coefficients of the pseudoremainders sequence were invertible, one would have $a, b \in (g)$ by a well-known property of the (extended) Euclidean algorithm (von zur Gathen and Gerhard, 1999, Algorithm 3.6). Denoting $M$ the multiplicative family generated by the product $h$ of the leading coefficients of the pseudoremainders and $\varphi$ the ring homomorphism (localization at $h$) which maps $R^-/I^-(G')$ to $(M/I^-(G'))^{-1}(R^-/I^-(G'))$, one thus has $\varphi(a), \varphi(b) \in (\varphi(g))$. By (Zariski and Samuel, 1958, Chapter IV, Theorem 15(a)), the ideal $(g) : h^\infty$ is the contraction w.r.t. $\varphi$ of the ideal $(\varphi(g))$. Thus $(a, b) \in (g) : h^\infty$ in $(R^-/I^-(G'))[x]$ and item 3 is proven. □

*Performing exact quotient operations.* In practical implementations, the returned gcd is actually the last nonzero subresultant of $a$ and $b$ and the computation is performed using a variant of a (good) pseudoremainder sequence algorithm. We chose the algorithm of Ducos (2000). Such an algorithm actually computes a sequence of subresultants $p_1, \ldots, p_n$ of $a$ and $b$ in $(R^-/\mathfrak{p}^-)[x]$. The only issue with such efficient algorithms consists in performing the exact quotient operations of the algorithm in $R^-/\mathfrak{p}^-$. Let's describe how we proceed. At each step $i$ one verifies that the leading coefficient of the current subresultant $p_i$ is nonzero in $R^-/\mathfrak{p}^-$. Assume this is the case. Then one continues the Ducos (2000) algorithm without normalizing $p_i$ in any sense w.r.t. $\mathfrak{p}$. Assume the leading coefficients of all the encountered subresultants are nonzero in $R^-/\mathfrak{p}^-$. Then the algorithm behaves exactly as Ducos (2000) in $R^-[v]$ whence exact quotient operations just have to be done in $R^-$. Assume now that the leading coefficient of $p_i$ is zero in $R^-/\mathfrak{p}^-$. Then one replaces $p_i$ by its reductum (i.e. one removes this coefficient from $p_i$), possibly many times, giving a polynomial $\overline{p}_i$. Then one restarts lsr over $p_{i-1}$ and $\overline{p}_i$.

This idea is very simple but very important. Elements of $R^-/\mathfrak{p}^-$ are residue classes. They can be computationally represented by any of their elements. For pseudoremainder sequences algorithms, the most convenient choice is to represent residue classes by representatives which make easy the exact quotient operations. This can be done by not normalizing coefficients at all. One just needs to make sure that leading coefficients are nonzero in the factor ring.

### *3.3. Algorithms handling critical pairs*

This section is dedicated to the study of complete, its subfunction insert_and_rebuild and PARDI which are really concerned by differential considerations. In particular, they need to handle lists of critical pairs.

function complete($\langle A,\ D,\ P,\ S\rangle,\ C,\ p$)
*Inserts $p$ in the set of the already processed equations $A$. Generates critical pairs.*
*Assumptions*
$G = \langle A,\ D,\ P,\ S\rangle$ *is a quadruple satisfying properties* **I1** *to* **I5**
$p$ *is a differential polynomial which lies in* $\mathfrak{p}$ *and is partially reduced w.r.t.* $A$
*The leader of* $p$ *is distinct from that of the elements of* $A$
*The initial* $i_p$ *and the separant* $s_p$ *of* $p$ *do not lie in* $\mathfrak{p}$
$C$ *is a characteristic set of the differential prime ideal* $\mathfrak{p}$
begin
$(A',\ \overline{S}) :=$ insert_and_rebuild$(p, A, C)$
$D' := D \cup \{\{p_\ell,\ p\} \mid p_\ell \in A,\ \{p_\ell,\ p\}$ is a critical pair$\}$
$P' := P$
$S' := S \cup \overline{S} \cup \{i_p,\ s_p\}$
return $\langle A',\ D',\ P',\ S'\rangle$
end

function insert_and_rebuild$(p,\ A,\ C)$
*Keeps $A$ as a squarefree regular chain*
begin
$\overline{A} := \{f \in A \mid \operatorname{ld} f < \operatorname{ld} p\}$
*Observe that the leaders of* $\overline{A}$ *are not derivatives of* $\operatorname{ld} p$
$B := \{p\} \cup \{f \in A \mid \operatorname{ld} f > \operatorname{ld} p$ and $\operatorname{ld} f$ is not a derivative of $\operatorname{ld} p\}$
*Recall that* $\operatorname{ld} p$ *is distinct from the leaders of* $\overline{A}$
Denote $B = \{p_1, \ldots, p_t\}$ (s.t. $\operatorname{ld} p_i < \operatorname{ld} p_{i+1}$)
$\overline{S} := \emptyset$
for $k := 1$ to $t$ do
$\overline{p}_k :=$ partial_rem$(p_k,\ \overline{A})$
*Denote* $i_{\overline{p}_k}$ *and* $s_{\overline{p}_k}$ *the initial and the separant of* $\overline{p}_k$
$(\overline{A},\ newS) :=$ saturate$(\overline{A},\ i_{\overline{p}_k},\ C)$
$\overline{S} := \overline{S} \cup newS$
$\overline{A} := \overline{A} \cup \{\overline{p}_k\}$
$(\overline{A},\ newS) :=$ saturate$(\overline{A},\ s_{\overline{p}_k},\ C)$
$\overline{S} := \overline{S} \cup newS$
od
return $(\overline{A},\ \overline{S})$
end

Fig. 8. The complete function and its insert_and_rebuild subfunction

*3.3.1. Completion of a quadruple*

One of the key steps of the PARDI algorithm consists in inserting a new differential polynomial $p$ (picked or computed from one of the lists $D$ and $P$) in the component $A$ of a quadruple $G$. This operation is performed by the complete function given in Figure 8. The ranking implicitly used is the target ranking $\overline{\mathscr{R}}$.

**Proposition 29.** *(termination)*
*The* complete *function terminates.*

**Proof.** One only needs to prove the termination of insert_and_rebuild. This function calls finitely many times saturate, which terminates by Proposition 10. □

Before proving Proposition 31, one establishes a lemma which proves that the ideals $I^v(G)$ grow i.e. that if

$$v_1 < v_2 < v_3 < \cdots$$

is an increasing sequence of derivatives and $G'$ denotes the next value of the quadruple $G$ then

$$\begin{array}{ccccccc} I^{v_1}(G) & \subset & I^{v_2}(G) & \subset & I^{v_3}(G) & \subset & \cdots \\ \cap & & \cap & & \cap & & \\ I^{v_1}(G') & \subset & I^{v_2}(G') & \subset & I^{v_3}(G') & \subset & \cdots \end{array}$$

This lemma is very important for it proves that if a critical pair is solved before the call to complete then it keeps being solved afterwards.

**Lemma 30.** *The* complete *function returns a quadruple* $G' = \langle A', D', P', S' \rangle$ *such that* $I^v(G) \subset I^v(G')$ *for every derivative* $v$.

**Proof.** The ideal $I^v(G)$ is modified by different operations. Some of these operations make the ideal clearly grow (insertion of $p$ in $A$, insertion of its initial and separant in $S$). The other operations are: the withdrawal of some differential polynomials from $A$ and the algebraic operations performed by saturate.

The withdrawn polynomials are the ones whose leader is a derivative of the leader of $p$. They are recovered in $D'$ because they are stored in reduction critical pairs by complete and thus belong to $\mathrm{hi}(D')$ which is part of the associated system of $G'$.

Proposition 14 (one needs this stronger form of Proposition 13 here) proves that the algebraic operations performed by saturate make the ideal $I^v(G)$ grow.

Thus, all the operations performed by complete imply that $I^v(G) \subset I^v(G')$ for each derivative $v$. □

**Proposition 31.** *(specifications of* complete*)*
*The* complete *function returns a quadruple* $G' = \langle A', D', P', S' \rangle$ *which satisfies properties* **I1** *to* **I5** *and such that* $I^v(G) \subset I^v(G')$ *for every derivative* $v$.

**Proof.** Lemma 30 implies that $I^v(G) \subset I^v(G')$ for every derivative $v$.

Property **I1**. The inclusion $I(G) \subset I(G')$ is thus proven. One only needs to prove $I(G') \subset I(G)$. $G'$ is obtained from $G$ by the following operations. The polynomial $p$ is stored in $A'$. Since $p \in \mathfrak{p}$, after this operation, one still has $I(G') \subset I(G)$. The initial and separant of $p$ are stored in $S'$. Since these polynomials do not lie in $\mathfrak{p}$ which is prime, after this operation, one has $I(G') \subset \mathfrak{p} : (i_p\, s_p)^\infty = \mathfrak{p}$. Some algebraic operations are performed by saturate on $A'$. Proposition 13, which describes them, shows that $(A') : H_{A'}^\infty \subset \mathfrak{p}$. Hence $I(G') \subset I(G)$ and $G'$ satisfies **I1**.

Property **I2**. One only needs to focus on insert_and_rebuild. The fact that the initial value of $\overline{A}$ is squarefree comes from the fact that $A$ is squarefree, combined to Lemma 5. After the first call to saturate, $\overline{A}$ is still squarefree by Proposition 13. Just before the

second call to saturate, $\overline{A}$ is no more squarefree. It gets squarefree after this call since its separant is made regular.

Property **I3** (sketched). The critical pairs defined by $A'$ which are not in $D'$ are solved by $G'$. The key arguments are given in Lemma 30 and Lemma 18. The fact that saturate stores in *newS* (see Figure 4) the factor $g$ or $h$ which does not lie in $\mathfrak{p}$ permits to apply Lemma 18.

Property **I4**. It holds for it is satisfied by $G$, the initial and separant of the new polynomials $p$ inserted in $A'$ are stored in $S'$ and saturate stores in *newS* (see Figure 4) the initial and the separant of the factor $g$ or $h$ which lies in $\mathfrak{p}$.

Property **I5**. It is satisfied by $G$ hence, using Lemma 30, it holds for reduction critical pairs of $D'$ which are already in $D$. Reduction critical pairs which lie in $D'$ but not in $D$ are of the form $\{p, p_\ell\}$ with $p = \mathrm{lo}(\{p, p_\ell\})$. Since $p \in A'$ we have $p \in I^{\mathrm{ld}\, p}(G')$. Thus property **I5** is satisfied by $G'$. □

*Avoiding critical pairs: a new criterion.* Not all new critical pairs between $p$ and the elements of $A$ need to be generated. Moreover, some of the critical pairs present in $D$ can be simply removed (i.e. not kept in $D'$).

One can implement an analogue of Buchberger's second criterion as described by Boulier et al. (1997) but the resulting algorithm is quite technical. The following new criterion is much easier to implement and turns out to be very efficient. It only tells us how to remove critical pairs in $D$ but it removes more critical pairs that the analogue of Buchberger's second criterion (in the differential setting).

**Proposition 32.** *Let $\{p, p'\} \in D$ be a critical pair. If $\{p, p'\}$ is not a reduction critical pair and $\{p, p'\} \not\subset A'$ then the critical pair does not need to be kept in $D'$.*

This criterion is proven in the (less interesting) context of Gröbner bases by Boulier (2001). We are not going to prove it in this paper but the idea is very simple: properties on critical pairs are only useful for proving that the hypotheses of the so called lemma of Rosenfeld (1959). hold for the set $A$ at the end of computations (the main loop of PARDI). Therefore critical pairs which contain at least one polynomial withdrawn from $A$ are irrelevant. Now, one must take care not to remove reduction critical pairs for these ones contain generators of the ideal (elements of the set of equations of the associated system of the quadruple). It is surprising that this criterion was not discovered earlier (at least in the context of Gröbner bases, see (Becker and Weispfenning, 1991)). We believe that this is due to the fact that reduction critical pairs were not distinguished from the other ones while they play a very special role.

### *3.3.2.* PARDI

In this section, one studies the main function PARDI, described in Figure 9. Given a known characteristic set $C$ w.r.t. a ranking $\mathscr{R}$ of a prime differential ideal $\mathfrak{p}$ and a target ranking $\overline{\mathscr{R}}$, one wants to compute a characteristic set $\overline{C}$ of $\mathfrak{p}$ w.r.t. $\overline{\mathscr{R}}$. The ranking implicitly used is the target ranking $\overline{\mathscr{R}}$. The main data structure is a quadruple $G = \langle A, D, P, S\rangle$. At the end of the computations, the desired characteristic set is "almost" found in $A$. Indeed, in this paper, PARDI is presented as returning a regular differential system (definition 36) $A = 0,\ S \neq 0$. Some work must still be performed in order to convert this regular differential system as a characteristic set. There are different ways to perform this last step. One of them is described by Boulier et al. (2001). Another one is given in (Boulier, 2006, regalise algorithm, sketched in section 6.2.2).

function PARDI($C, \mathscr{R}, \overline{\mathscr{R}}$)
*Performs a change of ranking* $\mathscr{R} \to \overline{\mathscr{R}}$ *over the characteristic set* $C$.
*Assumptions*
  $C$ *is a characteristic set of* $\mathfrak{p}$ *w.r.t ranking* $\mathscr{R}$
  $\mathscr{R}$ *is a ranking*
  $\overline{\mathscr{R}}$ *is another (target) ranking*
begin
  $\langle A, D, P, S\rangle := \langle \emptyset, \emptyset, C, H_C$ taken w.r.t. $\mathscr{R}\rangle$
  while $D \neq \emptyset$ or $P \neq \emptyset$ do
    Take and remove some $p \in P$ or some critical pair $\{p_1, p_2\} \in D$.
      In the latter case let $p = \Delta(p_1, p_2)$.
    $\overline{p} :=$ partial_rem$(p, A)$
    $(\overline{p},\ P) :=$ ensure_rank$(\overline{p}, G = \langle A, D, P, S\rangle, C)$
    if $\overline{p} \neq 0$ then
      if there exists some $q \in A$ such that $\mathrm{ld}\,\overline{p} = \mathrm{ld}\,q$ then
        $(g,\ P,\ S) :=$ lsr$(\overline{p}, q, \mathrm{ld}\,q,\ \langle A, D, P, S\rangle, C)$
        if $g \neq q$ then
*Instead of calling* complete*, one could actually just replace* $q$ *by* $g$ *in* $A$ *and all the critical pairs of* $D$*. The key argument is in Lemma 18. We choose not to do it in this paper to shorten proofs.*
          $\langle A, D, P, S\rangle :=$ complete$(\langle A \setminus \{q\}, D, P, S\rangle, C, g)$
          enlarge $S$ with pquo$(q,\ g)$
        fi
      else
        $\langle A, D, P, S\rangle :=$ complete$(\langle A, D, P, S\rangle, C, \overline{p})$
      fi
    fi
  od
  $S :=$ partial_rem$(S,\ A)$
  return $(A,\ S)$
end

Fig. 9. The main function PARDI

*About the inequations.* Observe that the inequations (the set $S$) are not used anywhere in the algorithms described in this paper. They are however useful for stating the properties of Figure 5 hence in the proofs. They may be needed for converting the regular differential system as a characteristic set. It depends on the algorithm applied for this step. Observe that in the case of PALGIE and PODI the best known algorithm, which seems to be regalise, does not use the inequations either. Using regalise in this setting permits to completely avoid inequations and thereby simplifies the pseudocodes given in this paper.

**Proposition 33.** *(termination)*
*The* PARDI *function terminates.*

**Proof.** The rank of $A$ decreases at each turn w.r.t. the classical ordering on autoreduced sets (Kolchin, 1973, chapter I, §10). This rank cannot strictly decrease at each turn by

(Kolchin, 1973, chapter I, §10, Proposition 3). It is sufficient to establish that it cannot indefinitely keep the same value.

The rank of $A$ does not change only if (1) $g = q$ after a call to lsr or all the coefficients of the differential polynomial (2) picked and removed from $P$ or (3) computed from a critical pair of $D$, belong to $\mathfrak{p}$.

In the three cases, the algorithm does not generate any critical pair (provided that the case $g = q$ is handled separately after a call to lsr). Therefore it is impossible to extract infinitely many critical pairs from $D$ and it is sufficient to consider the two first cases: in these two cases, one differential polynomial is picked from $P$ and is replaced by finitely many differential polynomials with a lower leader. Rankings are well orderings (Kolchin, 1973, chapter I, §8). By a classical argument of graph theory (i.e. every infinite, locally finite tree involves a branch of infinite length) this cannot happen infinitely many times. Thus the algorithm terminates. □

Before proving that the properties **I1** to **I5** are loop invariants of PARDI, one establishes a lemma which proves that if a critical pair is solved at some loop iteration then it keeps being solved afterwards. See the more detailed comments preceeding Lemma 30.

**Lemma 34.** *Denote $G = \langle A, D, P, S\rangle$ the value of the quadruple at the beginning of the loop body and $G' = \langle A', D', P', S'\rangle$ its value after execution of the loop body.*

*If $G$ satisfies properties* **I1** *to* **I5** *then $I^v(G) \subset I^v(G')$ for every derivative $v$.*

**Proof.** Denote $F = 0,\ S \neq 0$ the system associated to $G$ and $F' = 0,\ S' \neq 0$ the system associated to $G'$. Two cases need to be considered.

First case: $p$ is picked from the set $P$. Denote $v$ the leader of $p$ and $\overline{p}$ the partial remainder of $p$ by $A$. Then, for some $h \in S \cap R_v$ we have $h\,p = \overline{p} \mod I^v(G)$. The call to ensure_rank may modify $\overline{p}$ but stores in $P$ the initials and separants needed to keep this relation true.

Observe that, strictly speaking, $G$ does not satisfy **I1** to **I5** just after the withdrawal of $p$ from $P$. However, for the needs of the proof, one may assume that one has delayed the withdrawal of $p$ from $P$ until the end of the loop body. Similarly, one may also assume that, before the first call to complete, the withdrawal of $q$ from $A$ is also delayed. Therefore one assumes in the following text that $G$ does satisfy **I1** to **I5** before any call to complete or lsr. Three subcases need to be considered.

First subcase: $\overline{p} = 0$. Then $p \in I^v(G)$, one has $I^{v'}(G) = I^{v'}(G')$ for each $v'$ and the lemma is proven.

Second subcase: $\overline{p} \neq 0$ and there does not exist any $q \in A$ having the same leader as $\overline{p}$. Then complete is called and, using Proposition 31 plus the fact that $G$ satisfies properties **I1** to **I5**, the lemma is proven.

Third subcase: $\overline{p} \neq 0$ and there exists some $q \in A$ having the same leader as $\overline{p}$. Then, by Proposition 28, the call to lsr provides a gcd $g$ of $\overline{p}$ and $q$ which has leader $v$ and satisfies: $\overline{p},\ q \in (g) : h^\infty$ in $(R^-/I^-(G))[v]$ where $h \in S \cap R_v$, the values of $P$ and $S$ are the ones updated by lsr, $R^-$ denotes the ring of the differential polynomials depending on derivatives strictly less than $v$ and $I^-(G)$ is defined as in section 3.2.2. This gcd is inserted in $G$ by complete hence, using Proposition 31 plus the fact that $G$ satisfies properties **I1** to **I5**, the lemma is proven. Observe that after the insertion of $g$, the polynomial $q$ is redundant and may be removed from $A$.

Second case: a critical pair is picked from $D$. First observe that one only needs to focus on the case of a reduction critical pair since the other ones do not enter the definition of the associated systems of the quadruples.

To shorten the proof, one also assumes that $\Delta$–polynomials are temporarily stored in $P$ before being handled by the remaining instructions of the loop body. That way, relying on the analysis of the first case, one only needs to prove that $I^v(G) \subset I^v(G')$ for each derivative $v$, if a reduction critical pair is picked and removed from $D$ and the corresponding $\Delta$–polynomial is stored in $P$.

Denote $\{p, p'\}$ the reduction critical pair, assume $p = \mathrm{hi}(\{p, p'\})$ and denote $v = \mathrm{ld}\, p$. Since the critical pair is a reduction one, $\Delta(p, p') = \mathsf{prem}(p, \phi p')$ for some differential operator $\phi$ such that $\mathrm{ld}\, \phi p' = v$. Using the fact that $p' = \mathrm{lo}(\{p, p'\})$ and properties **I4** and **I5** satisfied by $G$, one sees that $p$ can be reconstructed from $p'$ and the $\Delta$–polynomial i.e. $p \in (F'_v) : (S' \cap R_v)^\infty$. □

**Lemma 35.** *Properties* **I1** *to* **I5** *are loop invariants of* PARDI.

**Proof.** These properties are all satisfied initially by $G = \langle \emptyset, \emptyset, C, H_C \rangle$.

Property **I1**. The inclusion $\mathfrak{p} \subset I(G)$ comes from Lemma 34. The converse inclusion is clear.

Property **I2** comes from Proposition 31.

Property **I3** (sketched). The critical pairs solved by $G$ are solved by $G'$. The key arguments are given in Lemma 34 and Lemma 18. Storing $\mathsf{pquo}(q, g)$ in $S$ after the first call to complete permits to apply Lemma 18.

Critical pairs still present in $D'$ are nearly solved by $G'$.

Consider a critical pair $\{p, p'\}$ removed from $D$. It is solved by $G'$ for the $\Delta$–polynomial is stored in $A'$ by complete and has a leader strictly less than the leader of $\mathrm{hi}(\{p, p'\})$.

Property **I4**. The only function which inserts some polynomial in $A$ or some critical pair in $D$ is the complete function. The proof thus follows from Proposition 31.

Property **I5**. The case of the reduction critical pairs generated by complete is considered in Proposition 31. That of the other ones is solved by Lemma 34. □

The following definition is borrowed from (Boulier et al., 1997). Regular differential systems are systems over which Rosenfeld's lemma (Rosenfeld, 1959) applies. See more precisely (Boulier et al., 1997, Definition 4.3 and Theorem 4.1).

**Definition 36.** A differential system $A = 0,\ S \neq 0$ is a *regular differential system* if

**C1** $A$ is differentially triangular (partially autoreduced and triangular) ;

**C2** the separants of $A$ belong to $S$ and $S$ is partially reduced w.r.t. $A$ ;

**C3** all the critical pairs that can be formed with the elements of $A$ are solved by the system $A = 0,\ S \neq 0$.

**Proposition 37.** *(specification of* PARDI*)*

*The differential system $A = 0,\ S \neq 0$ returned by* PARDI *is a regular differential system w.r.t. $\overline{\mathscr{R}}$ such that $[A] : S^\infty = \mathfrak{p}$.*

**Proof.** The returned quadruple $G$ satisfies properties **I1** to **I5** by Lemma 35. It also satisfies $D = P = \emptyset$. Property **I2** implies property **C1**. Property **I4** and the fact that PARDI partially reduces the elements of $S$ by $A$ before returning implies that **C2** holds. Property **I3** combined with the fact that $D$ is empty implies that **C3** holds. Therefore $A = 0,\ S \neq 0$ is a regular differential system. Property **I1** combined to the fact that $D = P = \emptyset$ implies that $[A] : S^\infty = \mathfrak{p}$. □

## 4. Applications

The three variants of PARDI were implemented: PARDI in MAPLE and C, PODI in C and PALGIE in MAPLE, C and ALDOR. The C implementation is available within the BLAD libraries (Boulier, 2004). It is involved within the LÉPISME project (Lemaire, 2004) which addresses the parameters estimation problem in the nonlinear control theory (see the third example below). Some generalizations such as the application to changes of variables, described in the introduction, were implemented in MAPLE. Our examples show that the restriction to prime ideals is realistic. Indeed most differential systems coming from real problems generate differential prime ideals. Quite often, nondifferential polynomial systems in positive dimension either generate prime ideals or can be decomposed into prime ideals. Assuming that prime ideals are given by characteristic sets is realistic too, in particular in the ordinary differential case, our third example shows.

*First example.* Our first example is academic. One considers the following three partial differential polynomials. There are two differential indeterminates $u$ and $v$ (which can be viewed as two unknown functions of two independent variables $x$ and $y$) and two derivations $\partial/\partial x$ and $\partial/\partial y$.

$$u_x^2 - 4u, \quad u_{xy}v_y - u + 1, \quad v_{xx} - u_x.$$

The differential ideal $\mathfrak{p}$ generated by these differential polynomials is prime. With respect to the following ordering (ranking) $\mathscr{R}$ on the derivatives of $u$ and $v$

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

the differential ideal $\mathfrak{p}$ admits the following set $C$ for characteristic set

$$v_{xx} - u_x, \quad 4v_y u + u_x u_y - u_x u_y u, \quad u_x^2 - 4u, \quad u_y^2 - 2u.$$

With respect to the following elimination ranking $\overline{\mathscr{R}}$,

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

it admits the following set $\overline{C}$ for characteristic set

$$v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1, \quad v_{xy}v_y - v_{yy}^3 + v_{yy}, \quad v_{xx} - 2v_{yy}, \quad u - v_{yy}^2.$$

The PARDI algorithm computes $\overline{C}$ from $C$, $\mathscr{R}$ and $\overline{\mathscr{R}}$ or $C$ from $\overline{C}$, $\overline{\mathscr{R}}$ and $\mathscr{R}$.

*Second example.* Our second example is related to fluid dynamics. Euler's equations for perfect fluids write

$$\vec{v} + (\vec{v} \cdot \vec{\nabla})\,\vec{v} + \vec{\nabla} p = \vec{0}, \quad \vec{\nabla}\vec{v} = 0.$$

In two dimensions, denoting $\vec{v} = (v^1, v^2)$ and $\vec{\nabla} = (\partial/\partial x, \partial/\partial y)$, one gets three differential polynomial equations

$$v_t^1 + v^1 v_x^1 + v^2 v_y^1 + p_x = 0, \quad v_t^2 + v^1 v_x^2 + v^2 v_y^2 + p_y = 0, \quad v_x^1 + v_y^2 = 0.$$

The differential polynomials which appear on the lefthand sides of the equations generate a prime differential ideal $\mathfrak{p}$. There are three differential indeterminates $v^1$, $v^2$ (components of the speed) and the pressure $p$. They depend on three independent variables $x$, $y$ (space variables) and the time $t$. For some orderly ranking, the general simplifier Rosenfeld–Gröbner provides with nearly no computation the characteristic set $C$ of $\mathfrak{p}$

$$p_{xx} + 2\, v_x^2\, v_y^1 + 2\,(v_y^2)^2 + p_{yy}, \quad v_t^1 + v^2\, v_y^1 + p_x - v_y^2\, v^1, \quad v_x^1 + v_y^2, \quad v_t^2 + v^1\, v_x^2 + v^2\, v_y^2 + p_y.$$

For some elimination ranking $(p,\, v^1) \gg \mathrm{degrevlex}(v^2)$ with $t > x > y$ an implementation of PARDI was able to compute a characteristic set $\overline{C}$ of $\mathfrak{p}$. This characteristic set cannot be written in this paper. PARDI is the very first algorithm to solve this elimination problem, given by Pommaret and only partially carried out by Pommaret (1992) and Boulier (1994). It is the first time that the computation of this characteristic set succeeds. There are 7 equations involving more than 50 different derivatives. We have (see Figure 10):

$$\mathrm{rank}\, \overline{C} = \{p_x,\, p_y,\, v^1,\, v^2_{xxxxt},\, v^2_{xxxtt},\, v^2_{xxytt},\, v^2_{xxxyyt}\}.$$

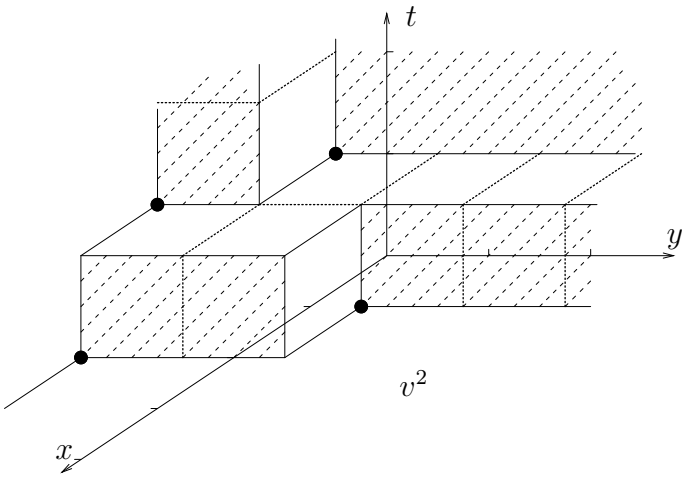


Fig. 10. Euler's equations for perfect fluids: the diagram of the differential indeterminate $v^2$.

*Third example.* Our third example comes from the parameters estimation problem in nonlinear control theory. We only sketch it in this introduction. A more detailed presentation was developed by Boulier et al. (2004) and Boulier (2007). The problem is this one: given a system of parametric ordinary differential equations and some measures, estimate the values of the unknown parameters. As an example, consider the following system, depending on the four parameters $k_{12}$, $k_{21}$, $k_e$ and $V_e$

$$\dot{x}_1 = -k_{12}\, x_1 + k_{21}\, x_2 - \frac{V_e\, x_1}{k_e + x_1}, \quad \dot{x}_2 = k_{12}\, x_1 - k_{21}\, x_2.$$

Assume that $x_1$ is observed (a file of measures is available) while $x_2$ is not observed. The PODI algorithm can be applied over this system in order to eliminate the non observed variable $x_2$. The computed characteristic set involves the following differential equation, involving the observed variable $x_1$ and the unknown parameters:

$$\ddot{x}_1\,(x_1+k_e)^2+[k_{12}+k_{21}]\,\dot{x}_1\,(x_1+k_e)^2+[V_e]\,\dot{x}_1\,k_e+[k_{21}\,V_e]\,x_1\,(x_1+k_e)=0.$$

This equation provides, by means of mixed numerical and symbolic computations, a first estimation of the values of the unknown parameters. This first estimation can then be used as a starting value for the Newton methods, widely used by practitioners, in order to obtain a more accurate estimation.

The PODI algorithm is here involved complementarily to the traditional numeric methods. It avoids guessing the starting point of the Newton methods. Algebraically, the input system already is a characteristic set of the ideal that it defines w.r.t. some (orderly) ranking. The rational fraction is equivalent to a polynomial since its denominator cannot vanish: parameters and differential indeterminates are assumed to take positive values. The target ranking is the block elimination ranking:

$$x_2 \gg (x_1,\, k_e,\, V_e,\, k_{12},\, k_{21}).$$

*Fourth example.* Our fourth example is related to the classical invariant theory. An ALDOR implementation of the PALGIE algorithm was used by Kogan and Moreno Maza (2002) as the core of a method for efficiently solving a problem of the classical invariant theory: deciding the equivalence of any two ternary cubics, that is, two homogeneous polynomials in three variables of degree three, under the action of a linear change of variables. The classification of ternary cubics is well known but, from a computational point of view, the most naive approach to decide equivalence requires very hard computations. In each orbit Kogan and Moreno Maza (2002) identify a "simple" canonical form and provide an algorithm that matches an arbitrary cubic with its canonical form. A corresponding linear change of variables is computed explicitly. The algorithm of Kogan and Moreno Maza (2002) is based on the differential geometry approach first introduced by Olver (1999).

Let us consider some ternary cubic $F(x,\, y,\, z)$ and let us sketch the method. First one removes one of the variables by replacing $F$ by its inhomogeneous projective version $f(p,\, q)$. Then one specializes at $f$ a set of fundamental differential invariants (Olver, 1999) of the considered action group. As the result, one gets a description of the signature manifold (Olver, 1999) of $f$ w.r.t. the two parameters $p$ and $q$. However, since two different parameterizations can define the same manifold, in order to compare the signatures of two different cubics $f$ and $\bar{f}$, one needs to eliminate $p$ and $q$ and compare the corresponding implicit equations.

From a computational point of view, the signature manifold of $f$ can be defined by some set of three polynomial equations in some polynomial ring $\mathbb{C}[I_1|_f,\, I_2|_f,\, I_3|_f,\, p,\, q]$ where each unknown $I_k|_f$ stands for some invariant specialized at $f$. It turns out that this set forms a characteristic set of the prime ideal $\mathfrak{p}$ that it defines w.r.t. the ordering $I_1|_f > I_2|_f > I_3|_f > p > q$. The implicitization of the signature manifold of $f$ amounts to compute a characteristic set of $\mathfrak{p}$ w.r.t. the following block elimination ordering. This problem was efficiently solved by PALGIE.

$$(p,\, q) \gg (I_1|_f,\, I_2|_f,\, I_3|_f).$$

*Changes of coordinates.* Our algorithm easily extends to perform invertible changes of coordinates on the dependent and independent variables. Such maps realize ring isomorphisms between two differential polynomial rings $\phi : R \to \overline{R}$, and one–to–one correspondences between the differential ideals of $R$ and the ones of $\overline{R}$. However the image $\overline{C}$ of a characteristic set $C$ of $\mathfrak{p}$ is usually not a characteristic set of the ideal $\overline{\mathfrak{p}} = \phi\mathfrak{p}$ and there is usually no ranking w.r.t. which a characteristic set of $\overline{\mathfrak{p}}$ could be easily deduced from $\overline{C}$. The idea is then to apply PARDI over $\overline{C}$ but to test membership in $\overline{\mathfrak{p}}$ by performing the inverse changes of coordinates and testing membership in $\mathfrak{p}$ using $C$.

## References


Aubry, P., 1999. Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom. Ph.D. thesis, Université Paris VI.

Aubry, P., Lazard, D., Moreno Maza, M., 1999. On the theories of triangular sets. Journal of Symbolic Computation 28, 105–124.

Becker, T., Weispfenning, V., 1991. Gröbner Bases: a computational approach to commutative algebra. Vol. 141 of Graduate Texts in Mathematics. Springer Verlag.

Boulier, F., 1994. Étude et implantation de quelques algorithmes en algèbre différentielle. Ph.D. thesis, Université Lille I, 59655, Villeneuve d'Ascq, France, http://tel.archives-ouvertes.fr/tel-00137866.

Boulier, F., November 1999. Efficient computation of regular differential systems by change of rankings using Kähler differentials. Tech. rep., Université Lille I, 59655, Villeneuve d'Ascq, France, ref. LIFL 1999–14, presented at the MEGA 2000 conference. http://hal.archives-ouvertes.fr/hal-00139738.

Boulier, F., October 2001. A new criterion to avoid useless critical pairs in Buchberger's algorithm. Tech. rep., Université Lille I, 59655, Villeneuve d'Ascq, France, ref. LIFL 2001–07.

Boulier, F., 2004. The BLAD libraries. http://www.lifl.fr/~boulier/BLAD.

Boulier, F., May 2006. Réécriture algébrique dans les systèmes d'équations différentielles polynomiales en vue d'applications dans les Sciences du Vivant. Mémoire d'habilitation à diriger des recherches. Université Lille I, LIFL, 59655 Villeneuve d'Ascq, France. http://tel.archives-ouvertes.fr/tel-00137153.

Boulier, F., 2007. Differential Elimination and Biological Modelling. Radon Series Comp. Appl. Math. 2, 111–139, http://hal.archives-ouvertes.fr/hal-00139364.

Boulier, F., Denis-Vidal, L., Henin, T., Lemaire, F., 2004. LÉPISME. In: proceedings of the ICPSS conference. Submitted to the Journal of Symbolic Computation, http://hal.archives-ouvertes.fr/hal-00140368.

Boulier, F., Lazard, D., Ollivier, F., Petitot, M., 1995. Representation for the radical of a finitely generated differential ideal. In: ISSAC'95: Proceedings of the 1995 international symposium on Symbolic and algebraic computation. ACM Press, New York, NY, USA, pp. 158–166, http://hal.archives-ouvertes.fr/hal-00138020.

Boulier, F., Lazard, D., Ollivier, F., Petitot, M., 1997. Computing representations for radicals of finitely generated differential ideals. Tech. rep., Université Lille I, LIFL, 59655, Villeneuve d'Ascq, France, ref. IT306. December 1998 version published in the HDR memoir of Michel Petitot. http://hal.archives-ouvertes.fr/hal-00139061.

Boulier, F., Lemaire, F., Maza, M. M., 2001. PARDI! In: ISSAC'01: Proceedings of the 2001 international symposium on Symbolic and algebraic computation. ACM Press, New York, NY, USA, pp. 38–47, http://hal.archives-ouvertes.fr/hal-00139354.

Boulier, F., Lemaire, F., Moreno Maza, M., 2006. Well known theorems on triangular systems and the $D^5$ principle. In: Proceedings of Transgressive Computing 2006. Granada, Spain, pp. 79–91, http://hal.archives-ouvertes.fr/hal-00137158.

Bouziane, D., Kandri Rody, A., Maârouf, H., 2001. Unmixed–Dimensional Decomposition of a Finitely Generated Perfect Differential Ideal. Journal of Symbolic Computation 31, 631–649.

Dahan, X., Jin, X., Moreno Maza, M., Schost, É., 2006. Change of Ordering for Regular Chains in Positive Dimension. Theoretical Computer ScienceTo appear.

Ducos, L., 2000. Optimizations of the subresultant algorithm. Journal of Pure and Applied Algebra 145, 149–163.

Golubitsky, O., 2004. Gröbner walk for characteristic sets of prime differential ideals. In: V. Ganzha and E. Mayr and E. Vorozhtsov (Ed.), Proceedings of the 7th Workshop on Computer Algebra in Scientific Computing. TU München, Germany, pp. 207–221.

Golubitsky, O., Kondratieva, M., Moreno Maza, M., Ovchinnikov, A., 2007. Bounds and algebraic algorithms in differential algebra: the ordinary case. In: Decker, W., Dewar, M., Kaltofen, E., Watt, S. M. (Eds.), Challenges in Symbolic Computation Software. No. 06271 in Dagstuhl Seminar Proceedings. Internationales Begegnungs und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

Hubert, É., 2000. Factorization free decomposition algorithms in differential algebra. Journal of Symbolic Computation 29 (4,5), 641–662.

Hubert, É., 2003. Notes on triangular sets and triangulation–decomposition algorithm II: Differential Systems. Symbolic and Numerical Scientific Computing 2001, 40–87.

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

Kogan, I., Moreno Maza, M., 2002. Computation of canonical forms for ternary cubics. In: proceedings of ISSAC 2002. Lille, pp. 151–160.

Kolchin, E. R., 1973. Differential Algebra and Algebraic Groups. Academic Press, New York.

Lazard, D., 1991. A new method for solving algebraic systems of positive dimension. Discrete Applied Mathematics 33, 147–160.

Lemaire, F., 2004. LÉPISME. `http://www2.lifl.fr/~lemaire/lepisme`.

Mansfield, E. L., 1991. Differential Gröbner Bases. Ph.D. thesis, University of Sydney, Australia.

Moreno Maza, M., 2000. On Triangular Decompositions of Algebraic Varieties. Tech. rep., NAG, presented at the MEGA2000 conference. Submitted to the Journal of Symbolic Computation.

Moreno Maza, M., Rioboo, R., 1995. Polynomial gcd computations over towers of algebraic extensions. In: Proceedings of AAECC11. Springer Verlag, pp. 365–382.

Ollivier, F., 1990. Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité. Ph.D. thesis, École Polytechnique, Palaiseau, France.

Olver, P. J., 1999. Classical Invariant Theory. Cambridge University Press.

Pommaret, J.-F., 1992. New Perspectives in Control Theory for Partial Differential Equations. IMA Journal of Mathematics Control and Information 9, 305–330.

Reid, G. J., Wittkopf, A. D., Boulton, A., 1996. Reduction of systems of nonlinear partial differential equations to simplified involutive forms. European Journal of Applied Math., 604–635.

Ritt, J. F., 1950. Differential Algebra. Dover Publications Inc., New York, `http://www.ams.org/online_bks/coll33`.

Rosenfeld, A., 1959. Specializations in differential algebra. Trans. Amer. Math. Soc. 90, 394–407.

Sit, W., 2002. The Ritt–Kolchin theory for differential polynomials. In: proceedings of the international workshop: Differential Algebra and Related Topics.

von zur Gathen, J., Gerhard, J., 1999. Modern Computer Algebra. Cambridge University Press, United Kingdom.

Wang, D., 2003. Elimination Practice: Software Tools and Applications. Imperial College Press, London.

Zariski, O., Samuel, P., 1958. Commutative Algebra. Van Nostrand, New York, Also volumes 28 and 29 of the *Graduate Texts in Mathematics*, Springer Verlag.