Computing differential characteristic sets by change of ordering
François Boulier, François Lemaire, Marc Moreno Maza

To cite this version:
François Boulier, François Lemaire, Marc Moreno Maza. Computing differential characteristic sets by change of ordering. submitted to the Journal of Symbolic Computation. 2007. <hal-00141095v1>

HAL Id: hal-00141095
https://hal.archives-ouvertes.fr/hal-00141095v1
Submitted on 11 Apr 2007 (v1), last revised 21 Sep 2007 (v2)

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

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Computing differential characteristic sets by change of ordering

François Boulier, François Lemaire and Marc Moreno Maza

Université Lille 1, LIFL, 59655 Villeneuve d’Ascq France
University of Western Ontario, ORCCA, London Canada
{boulier, lemaire}@lifl.fr, moreno@scl.csd.uwo.ca

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.

Introduction

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

The proposed algorithm, called† $\text{PARDI}$ applies for systems of partial differential polynomial equations. It specializes to systems of ordinary differential polynomial equations and is then called‡ $\text{PODI}$. It specializes to nondifferential polynomial equations where it is called§ $\text{PALGIE}$.

The three variants were implemented: $\text{PARDI}$ in MAPLE and C, $\text{PODI}$ in C and $\text{PALGIE}$ 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

*Submitted to the Journal of Symbolic Computation, April 2007
†$\text{PARDI}$ is an acronym for Prime pARtial Differential Ideal. In French, “$pardi$” is an old-fashioned swearword such as, say, “egad” in English.
‡$\text{PODI}$ is an acronym for Prime Ordinary Differential Ideal.
§$\text{PALGIE}$ is an acronym for Prime ALGebraic IdEal. However, since “algie” means “suffering” in French, one might also understand $\text{PALGIE}$ as “polynomial suffering” say.
as the application to changes of variables, described below, were implemented in MAPLE.

First example. Our first example is academic. It is completely detailed in section 2.6. 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 \( \partial/\partial y \).

\[
\begin{align*}
&u_x^2 - 4u, \quad u_{xy}v_y - u + 1, \quad v_{xx} - u_x.
\end{align*}
\]

The differential ideal \( p \) generated by these differential polynomials is prime. With respect to the following ordering (ranking) \( \mathcal{R} \) 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 \( p \) admits the following set \( C \) for characteristic set

\[
\begin{align*}
v_{xx} - u_x, \quad 4v_yu + u_xu_y - u_xu_yu, \quad u_x^2 - 4u, \quad u_y^2 - 2u.
\end{align*}
\]

With respect to the following elimination ranking \( \mathcal{R} \),

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

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

\[
\begin{align*}
v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1, \quad v_{xy}v_y - v_{yy}^3 + v_{yy}, \quad v_{xx} - 2v_{yy}, \quad u - v_y^2.
\end{align*}
\]

The PARDI algorithm computes \( \overline{C} \) from \( C \), \( \mathcal{R} \) and \( \overline{\mathcal{R}} \) or \( C \) from \( \overline{C} \), \( \mathcal{R} \) and \( \overline{\mathcal{R}} \).

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

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

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

\[
\begin{align*}
v_t^1 + v^1v_x^1 + v^2v_y^1 + p_x &= 0, \quad v_t^2 + v^1v_x^2 + v^2v_y^2 + p_y = 0, \quad v_x^1 + v_y^1 = 0.
\end{align*}
\]

The differential polynomials which appear on the lefthand sides of the equations generate a prime differential ideal \( p \). 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 \textbf{Rosenfeld–Gröbner} provides with nearly no computation the characteristic set \( C \) of \( p \)

\[
\begin{align*}
p_{xx} + 2v_x^2v_y^1 + 2(v_y^2)^2 + p_{yy}, \quad v_t^1 + v^1v_x^1 + p_x - v_y^2v^1, \quad v_x^1 + v_y^2, \quad v_t^2 + v^1v_x^2 + v^2v_y^2 + p_y.
\end{align*}
\]
For some elimination ranking \((p, v^1) \gg \degrevlex(v^2)\) with \(t > x > y\) an implementation of \texttt{PARDI} was able to compute a characteristic set \(C\) of \(p\). This characteristic set cannot be written in this paper. \texttt{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 1):

\[
\text{rank } C = \{p_x, p_y, v^1, v_{xxxt}, v_{xxxtt}, v_{xyyt}, v_{xxyyt}\}.
\]

\textbf{Figure 1:} Euler’s equations for perfect fluids: the diagram of the differential indeterminate \(v^2\).

\textit{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 \(V_e\)

\[
\begin{align*}
\dot{x}_1 &= -k_{12} x_1 + k_{21} x_2 - \frac{V_e x_1}{k_e + x_1}, \\
\dot{x}_2 &= k_{12} x_1 - k_{21} x_2.
\end{align*}
\]

Assume that \(x_1\) is observed (a file of measures is available) while \(x_2\) is not observed. The \texttt{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 [Bronstein et al., 2002] implementation of the PALGIE algorithm was used by Kogan and 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 [Kraft, 1985, Gurevich, 1964] but, from a computational point of view, the most naive approach to decide equivalence requires very hard computations. In each orbit Kogan and 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 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 \( p \) 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 \( p \) w.r.t. the following block elimination ordering. This problem was efficiently solved by PALGIE.

\[ (p, q) \gg (I_1|_f, I_2|_f, I_3|_f). \]
Historical considerations. 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 in [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. Recently, an approach based on the Gröbner walk idea was studied by Golubitsky [2004], in the particular setting of prime ordinary differential ideals. A preliminary version of our paper was published in [Boulier et al., 2001].

Restrictions and generalizations. 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 the differential case, it happens quite often that the input equations already form characteristic sets w.r.t. some ranking, our third example shows.

The proposed algorithm generalizes to ideals which are not necessarily prime. However, for the reasons explained above and the legibility of the paper, we prefer to restrict ourselves to the prime case.

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 \( \phi : R \rightarrow \overline{R} \), 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 \( p \) is usually not a characteristic set of the ideal \( \overline{p} = \phi p \) and there is usually no ranking w.r.t. which a characteristic set of \( \overline{p} \) could be easily deduced from \( \overline{C} \). The idea is then to apply PARDI over \( \overline{C} \) but to test membership in \( \overline{p} \) by performing the inverse changes of coordinates and testing membership in \( p \) using \( C \).
Nice features of our approach. Our approach offers several 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. Hence we wish that the simplicity of our approach helps in popularizing all triangular decomposition methods.

Summary of the paper. Section 1 presents some necessary general definitions and concepts. It is divided in a computer science, a commutative algebra and a differential algebra subsections. The commutative algebra subsection presents two subalgorithms called by PARDI, which were not considered by Boulier et al. [2001]. Section 2 is devoted to the study of PARDI and its subalgorithms. A new criterion for avoiding critical pairs is given in section 2.3.1. Subsection 2.4 describes our algorithm for computing gcds of univariate polynomials over the field of fractions of some factor ring. The main difficulty consists here to carry out the exact quotient operations required by the efficient pseudoremainders sequences methods. It is overcome in section 2.4.1. Last, a complete example is detailed in section 2.6. In order to simplify and shorten our 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).

Conventions. Each important algorithm is presented by a pseudocode in a figure plus two propositions. The first proposition proves the termination. The conditions that the formal parameters must satisfy are given together with the pseudocode. The properties of the returned values are described and proved in the second proposition.
1. General definitions and notations

1.1. Computer science

Definition: 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.

1.2. Commutative algebra

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 $\text{lcoeff}(p, x)$. If $\deg(p, x) = 0$ then $\text{lcoeff}(p, x) = p$.

The leader of $p$, denoted $\text{ld}p$, is the greatest indeterminate $x$ which occurs in $p$. 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 $\text{pquo}(q, p, x)$ and $\text{prem}(q, p, x)$ the pseudo-quotient 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 $\text{prem}(q, A)$ “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 $\text{prem}(r, p) = r$ for every $p \in A$. Without further precisions, $r$ is not uniquely defined. Fix any precise algorithm. By convention, one defines $\text{prem}(q, \emptyset) = q$.

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.
1.2.1. Regular chains

A regular element of a ring $R_0$ is by definition a non zerodivisor of $R_0$. An element $a \in R_0$ is said to be invertible if there exists some $\overline{a} \in R_0$ such that $a \overline{a} = 1$. Invertible implies regular.

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 may assume that $R = K[t_1, \ldots, t_m, x_1, \ldots, x_n]$ and that $\text{ld} p_i = x_i$ 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 $A_{i0}$ the triangular set $\{p_1, \ldots, p_i\}$. Denote $R_{i0}$ the ring $K(t_1, \ldots, t_m)[x_1, \ldots, x_i]$. Denote $A_i$ the ideal $(A_i): I_{\infty}^A$ of $R_i$ and $A_{i0}$ the ideal $(A_i): I_{\infty}^{A_{i0}}$ of $R_{i0}$. Denote $R_0 = R_{0,n}$ and $A = A_n$.

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.

**Lemma 1.1:** 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] and Lazard [1991]. We adopt the next definition [Boulier et al., 2006, Definition 3.1].

**Definition:** 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 $R_{\ell}/\mathfrak{a}_{\ell}$.

All the following lemmas recall “well-known” theorems on regular chains.

**Lemma 1.2:** (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 $\text{prem}(a, A) = 0$.

**Proof:** See [Aubry et al., 1999, Theorem 6.1], [Aubry, 1999, théorème 4.6.1] or [Boulier et al., 2006, Proposition 3.7].

**Lemma 1.3:** 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 squarefree 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 1.2 the set of the polynomials of $R_i$ reduced to zero by $A_i$ is $\mathfrak{a}_i$. By Lemma 1.2, 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 1.4: *(corollary to Lazard’s lemma)*

If $A$ is a squarefree regular chain then the ideals $A$ and $A_0$ are radical.

Proof: [Boulier et al., 1995, Lemma 2] or [Boulier et al., 2006, Corollary 3.3]. □

Lemma 1.5: If $A$ is a squarefree regular chain then $A = (A) : H^\infty_A$.

Proof: By Lemma 1.4 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.

1.2.2. Checking regularity

This section is dedicated to the study of two functions called: is regular and Euclidean algorithm. One keeps the notations of the above section. In this section, $A$ is assumed to be a regular chain and one denotes $p$ a prime ideal containing $A$. One assumes moreover that the initials of the elements of $A$ do not lie in $p$ and that membership testing in $p$ is algorithmic.

Let us explain a bit the relationship between this section and the rest of the paper. The function is regular is actually called by complete which is itself a subfunction of PARDI. The ideal $p$ actually is the differential prime ideal passed to PARDI but, at this stage of the paper, one does not need to bother with differential considerations. Membership testing in $p$ is performed by means of the known characteristic set $C$ of $p$. Observe that the (differential) polynomial $p$ passed to is regular may depend on indeterminates different from the leaders of $A$ (indeed, at the beginning of the computations, $A$ is the empty set). Then before performing the function call, one defines as $t_1, \ldots, t_m$, the indeterminates different from the leaders of $A$, occurring in $p$ and the elements of $A$. This is implicitly justified by Lemma 1.1.

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 1.1: *(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 performs finitely many turns for the degree of $q$ decreases, apart perhaps at the first turn. It thus terminates also.
function is_regular($p, A = \{p_1, \ldots, p_n\}, C)$

Assumptions

- $p$ is a polynomial of $R_0$.
- $A$ is a regular chain of $R_0$ (see section 1.2.1).
- The ideal $\mathfrak{a}$ (defined by $A$, see section 1.2.1) 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 $p \in K(t_1, \ldots, t_m)$ then
    return (true, \cdot)
  else
    let $x_\ell$ be the leader and $i_p$ be the initial of $p$
  One passes $A_{\ell-1}$ instead of $A$ to the two functions below, only to simplify the termination proof
  (bool, g) := is_regular($i_p$, $A_{\ell-1}$, C)
  if bool then
    (bool, g) := Euclidean_algorithm($p$, $p_\ell$, $x_\ell$, $A_{\ell-1}$, C)
  fi
  if not bool then
    return (bool, g)
  elseif $deg(g, x_\ell) > 0$ then
    return (false, g)
  else
    return (true, \cdot)
  fi
end

Figure 2: Function is_regular

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 $|A| = n$. \hfill \Box

Proposition 1.2: (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 $0 < d < deg(p_\ell, x_\ell)$,
function Euclidean_algorithm(a, b, x, A = \{p_1, \ldots, p_n\}, C)

Assumptions

A is a regular chain.
The ideal \( A \) (defined by \( A \)) is included in \( p \)
The initials of the elements of \( A \) do not lie in \( p \)
\( a \) 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 \((R_0/\mathfrak{A}_0)[x]\) and do not lie in \( p \)
\( C \) is a characteristic set of \( p \). It is used for membership testing in \( p \)

begin
\( p := a \)
\( q := b \)
while \( q \neq 0 \) do
\( r := \text{prem}(p, q, x) \)
while \( r \neq 0 \) and \( \text{lcoeff}(r, x) \in p \) do
\( r := \text{reductum}(r, x) \)
\( \text{od} \)
if \( r \neq 0 \) then
\( (\text{bool}, h) := \text{is_regular}(\text{lcoeff}(r, x), A, C) \)
if not \( \text{bool} \) then
\( \text{return} (\text{bool}, h) \)
fi
fi
\( p := q \)
\( q := r \)
\( \text{od} \)
\( \text{return} (\text{true}, p) \)
end

Figure 3: Function Euclidean_algorithm

1. there exists a polynomial \( g \) with leader \( x_\ell \) s.t. \( g h = p_\ell \) in \((R_0, \ell-1)/\mathfrak{A}_0, \ell-1)[x_\ell] \),
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 \( p \) and is invertible in \( R_0/\mathfrak{A}_0 \).

The function Euclidean_algorithm returns a pair \((\text{bool}, g)\) where \( \text{bool} \) is a boolean and \( g \) is a polynomial of \( R_0[x] \). If \( \text{bool} \) is false then \( g \) satisfies the properties 1, 2 and 3 stated just above. If \( \text{bool} \) 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 \( p \) and is invertible in \( R_0/\mathfrak{A}_0 \).
Proof: By induction on the number \( n \) of elements of \( A \).

Basis: \( n = 0 \). For \texttt{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 \texttt{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 1.2 is satisfied.

The general case: \( n > 0 \). One assumes inductively that the results of the calls to \texttt{is\_regular} and \texttt{Euclidean\_algorithm} with \( |A| < n \) satisfy the proposition.

Function \texttt{is\_regular}. If any of the calls to \texttt{is\_regular} or \texttt{Euclidean\_algorithm} returns a pair \((false, g)\) then this pair may be returned.

Assume thus that both calls to \texttt{is\_regular} and \texttt{Euclidean\_algorithm} return pairs of the form \((true, g)\) and consider the one returned by \texttt{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 \texttt{Euclidean\_algorithm} corresponds to index \( \ell - 1 \) (respectively polynomials \( p, p_\ell \) and variable \( x_\ell \)) in \texttt{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 \texttt{Euclidean\_algorithm}. If any call to \texttt{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.

\[ \square \]

1.2.3. Saturating ideals

This section is dedicated to the study of the \texttt{saturate} function, given in Figure 4, which is called by the function \texttt{insert\_and\_rebuild} described in Figure 7.

This function aims at saturating the ideal \( \mathfrak{A} \) defined by a regular chain \( A \) by a polynomial \( p \). However, instead of returning a regular chain defining the ideal \( \mathfrak{A} : p^{\infty} \), it 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 \texttt{is\_regular} needs to check the regularity of many different polynomials. Any of these tests 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 \( \mathfrak{p} \).
function saturate(A, p, C)
Assumptions
A is a regular chain
The ideal A (defined by A) is included in p
The initials and the separants of A do not lie in p
p is a polynomial which does not lie in p
C is a characteristic set of p
begin
A := A
newS := ∅
(bool, g) := is_regular(p, A, C)
while not bool do
let x_ℓ be the leader of g and denote h = pquo(p_ℓ, g, x_ℓ)
if g ∈ p then
replace p_ℓ by g in A
store h, the initial of g and the separant of g in newS
else
replace p_ℓ by h in A
store g, the initial of h and the separant of h in newS
fi
(bool, g) := is_regular(p, A, C)
od
return (A, newS)
end

Figure 4: Function saturate

The function also returns a set newS of polynomials which do not lie in p. The importance of returning these polynomials is going to appear in the proof of Proposition 2.2.

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 p. If moreover A is squarefree\(^*\) then the ideal A, which is defined as (A) : I^∞_A (section 1.2.1), is also equal to (A) : H^∞_A by Lemma 1.5. If A is a regular chain, denote \(\overline{A} = (A) : I^∞_A\).

Proposition 1.3: (termination)
The saturate function terminates.

Proof: The fact that \(p \notin p\) and \(A \subset p\) implies that, at each loop, \(\deg(g, x_\ell)\) and

\(^*\)Observe that one may have \(A \subset p\), the separants of the elements of A outside p without having A squarefree.
deg(h, x_\ell) are strictly less than deg(p_\ell, x_\ell). Thus, at each loop, the degree of some element of A decreases strictly. The function thus terminates. 

**Lemma 1.6:** Consider the saturate function. If the first call is\_regular(p, 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:

\[
A \subset (A_g) : I^\infty_{A_g} \cap (A_h) : I^\infty_{A_h}
\]

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

**Proof:** By Proposition 1.2, the polynomial g is a nontrivial factor of p_\ell with an initial invertible in R_0/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]. \(\square\)

**Lemma 1.7:** The saturate function returns a set newS of polynomials which do not lie in p and a regular chain \(\overline{A}\) whose initials and separants do not lie in p, having the same set of leaders as A and which satisfies:

\[
A \subset \overline{A} \subset p.
\]

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 \(A \subset \overline{A}\) follows from Lemma 1.6. To prove the second inclusion, one still needs to prove that the polynomial g (or h) which replaces p_\ell lies in p and that its initial does not lie in p.

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

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

\[
c p_\ell = g h \mod p
\]

where c is a power of the initial of g. The initial of g does not lie in p by item 3 of Proposition 1.2 thus c does not either for the ideal prime. The initial i_\ell of p_\ell does not lie in 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 ci_\ell modulo p.
The second inclusion is thus proven. To conclude the proof of the lemma, one still needs to prove that the elements of \( \text{newS} \) 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 separators 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 separators 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 \( \text{newS} \) do not lie in this ideal.

This concludes the proof of the lemma. \( \square \)

**Proposition 1.4:** (specification of saturate)

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

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

If moreover \( A \) is squarefree then so is \( \overline{A} \).

**Proof:** Relying on Lemma 1.7, one only needs to prove that relation (3) holds. Relation (1) implies that \( \overline{A} : p^\infty \subset \overline{A} : p^\infty \). The inclusion \( \overline{A} \subset \overline{A} : p^\infty \) is trivial. At the end of the loop execution, \( p \) is regular modulo \( \overline{A} \) and we have \( \overline{A} = \overline{A} : p^\infty \). The proposition follows. \( \square \)

The next proposition strengthens Proposition 1.4. This stronger form is needed to prove Lemma 2.1.

**Proposition 1.5:** (stronger specification of saturate)

Let \( 1 \leq i \leq n \) be an index. Denote \( \overline{A}_i = A \cap R_i \) and \( \overline{A}_i = (\overline{A}_i) : I^\infty_i \).

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

\[
\overline{A}_i \subset \overline{A}_i : p^\infty \subset \overline{A}_i \subset \mathfrak{p}.
\]

If moreover \( A \) is squarefree then so is \( \overline{A}_i \).

**Proof:** It is a corollary to Proposition 1.4 and to Lemma 1.3. \( \square \)

### 1.3. Differential algebra

A derivation over a ring $R$ is a map $\delta : R \rightarrow R$ such that $\delta(a + b) = \delta a + \delta b$ and $\delta(ab) = (\delta a)b + a(\delta b)$ 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 $\theta = \delta^{a_1} \cdots \delta^{a_m}$ 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 $\text{ord} \, \theta$. The identity operator is the unique operator with order 0. The other ones are called proper. If $\phi = \delta^{b_1} \cdots \delta^{b_m}$ then $\theta \phi = \delta^{a_1+b_1} \cdots \delta^{a_m+b_m}$. If $a_i \geq b_i$ for each $1 \leq i \leq m$ then $\theta/\phi = \delta^{a_1-b_1} \cdots \delta^{a_m-b_m}$.

A differential ideal $a$ of $R$ is an ideal of $R$ closed under derivation i.e. such that $a \in A \Rightarrow \delta a \in A$. 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 $A$.

### 1.3.1. 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, page 75] 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:** If $A$ is a set of differential polynomials and $v$ is a derivative then $A_v = \{ p \in \Theta A \mid \text{ld} \, p \leq v \}$.
Thus $R_v$ denotes the set of the differential polynomials with leader less than or equal to $v$.

1.3.2. Ritt’s reduction algorithms

One distinguishes the partial reduction algorithm, which is denoted $\text{partial\_rem}$ from the full reduction algorithm, denoted $\text{full\_rem}$ [Kolchin, 1973, page 77]. Let $q$ and $p$ be two differential polynomials. The partial remainder $\text{partial\_rem}(q, p)$ is the pseudoremainder of $q$ by the (infinite) set of all the proper derivatives of $p$. The full remainder $\text{full\_rem}(q, p)$ 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, section 9]. Let $A$ be a set of differential polynomials. One denotes $\text{partial\_rem}(q, A)$ and $\text{full\_rem}(q, A)$ respectively the partial remainder and the full remainder of $q$ by all the elements of $A$.

Let $v = \text{ld} q$ 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 $hq \equiv \overline{q}$ 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 $hq \equiv \overline{q}$ mod $(\overline{A}_v)$.

1.3.3. 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 $\text{ld} p_1 = \theta_1 u$ and $\text{ld} p_2 = \theta_2 u$). Denote $\theta_{12} u = \text{lcd}(\text{ld} p_1, \text{ld} p_2)$ the least common derivative of $\text{ld} p_1$ and $\text{ld} p_2$ defined by $\theta_{12} = \text{lcm}(\theta_1, \theta_2)$.

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,

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

Definition: If $\{p, p'\}$ is a reduction critical pair with (say) $\text{ld} p > \text{ld} p'$ then $\text{hi}(\{p, p'\}) \equiv p$ and $\text{lo}(\{p, p'\}) \equiv p'$. If $D$ is a list of critical pairs then

$$\text{hi}(D) = \{\text{hi}(\{p, p'\}) : \{p, p'\} \text{ is a reduction critical pair of } D\}.$$
**Definition:** A critical pair \( \{ p, p' \} \) is said to be solved by a system \( F = 0, \ S \neq 0 \) if there exists a derivative \( v < \text{lcm}(\text{lcm}(p, p')) \) 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) \( p_t \) of \( A \) gets replaced by one of its factor (say) \( g \). Now, the polynomial \( p_t \) may be involved in some critical pair \( \{ p_t, p_r \} \), considered at some previous stage by PARDI hence solved by \( F = 0, \ S \neq 0 \). Since \( p_t \) 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 \( p_t = gh \) in the lemma by the statement given in the item 2 of Proposition 1.2. Only the triangular case needs to be considered. This lemma is used in the proof of Proposition 2.2.

**Lemma 1.8:** Let \( \{ p_t, p_r \} \) be a triangular critical pair, solved by a differential system \( F = 0, \ S \neq 0 \). Assume that \( p_t = gh \) with \( \text{lcm}(p_t) = \text{lcm}(g) = \text{lcm}(h) \). 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 \( \text{lcm}(p_t) = \theta_t u, \ \text{lcm}(p_r) = \theta_r v \) and \( \theta_t = \text{lcm}(\theta_t, \theta_r) \). One assumes that \( \{ p_t, p_r \} \) is solved by \( F = 0, \ S \neq 0 \) i.e., denoting \( s_t \) and \( s_r \) the separants of \( p_t \) and \( p_r \), that there exists some \( v < \theta_t \) such that:

\[
s_t \frac{\theta_r}{\theta_t} p_t - s_r \frac{\theta_t}{\theta_r} p_r \in (F_v) : (S \cap R_v)\infty.
\]

Denote \( s_g \) and \( s_h \) the separants of \( g \) and \( h \). Since \( p_t, g \) and \( h \) have the same leader, one has \( s_t = s_g h + s_h g \). In formula (4), replace \( s_t \) by this expression, \( p_t \) by \( g h \) and expand \( (\theta_t / \theta_t) (g h) \). Using the fact that \( F \subseteq F' \) and \( S \subseteq S' \), 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 \( \text{lcm}(\varphi) < v \). Remove the term \( s_h g (\theta_t / \theta_r) p_r \) for it involves \( g \) as a factor. One obtains:

\[
s_r \left( \frac{\theta_r}{\theta_t} g \right) h - s_g h \frac{\theta_r}{\theta_t} p_r \in (F'_v) : (S' \cap R_v)\infty.
\]

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 \( F' = 0, \ S' \neq 0 \).

**1.3.4. 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:** 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 1.6:** If $C$ is a characteristic set of $\mathfrak{A}$ and $H_C$ contains no zero divisor in the factor ring $\mathbb{R}/\mathfrak{A}$ then $\mathfrak{A} = [C]: H_C^\infty$ and $p \in \mathfrak{A}$ if and only if $\text{full\_rem}(p, C) = 0$. This is the case when $\mathfrak{A}$ is prime.

**Proof:** Let $p$ be a differential polynomial and denote $r = \text{full\_rem}(p, C)$. 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 of $\mathfrak{A}$. 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 $\text{full\_rem}(s_p, C)$. 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 $\mathbb{R}/\mathfrak{A}$ when $\mathfrak{A}$ is prime and concludes the proof of the proposition. □
2. The differential part of PARDI

Given a known characteristic set \( C \) w.r.t. a ranking \( R \) of a prime differential ideal \( p \) and a new ranking \( \overline{R} \), one wants to compute a characteristic set \( \overline{C} \) of \( p \) w.r.t. \( \overline{R} \). The main data structure is a quadruple \( G = \langle A, D, P, S \rangle \), defined in section 2.1. At the end of the computations, the desired characteristic set is found in \( A \).

Throughout this section, the ranking implicitly used is the target ranking \( \overline{R} \).

2.1. Quadruples

The main data structure handled by the PARDI algorithm is a quadruple \( G = \langle A, D, P, S \rangle \). Roughly speaking, \( A \) is the set of the differential polynomial equations already processed (it will contain \( \overline{C} \) at the end of the computations), \( D \) 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 \( \text{hi}(D) \) and the expression “solved pair” used in the next definitions are defined in section 1.3.3.

Definition: Let \( G = \langle A, D, P, S \rangle \) be a quadruple and \( F = A \cup \text{hi}(D) \cup P \). 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: 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 \( I^v(G) \stackrel{\text{def}}{=} I^v(F, S) \) where \( F = 0, S \neq 0 \) is the system associated to \( G \).

Definition: A critical pair is said to be solved by a quadruple \( G \) if it is solved by the system associated to \( G \).

Definition: 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 \( D \).

Throughout its execution, the PARDI algorithm keeps true the properties stated in Figure 5.

2.2. The “master–student” relationship

One of the main ideas of PARDI consists in applying a “master–student” relationship between \( C \) and \( A \). To decide whether a quantity is zero or not modulo \( p \) one just needs to decide whether this quantity is reduced to zero or not by the “master” \( C \). If it is, one checks if it is also reduced to zero by the “student” \( A \) and we store in \( P \) (the equations to be processed) every quantity reduced to zero by \( C \) but not by \( A \).

The function \textbf{ensure_rank} in Figure 6 provides an easy illustration of the “master–student” relationship. It takes as input a differential polynomial \( p \), a quadruple \( G \) and the known characteristic set \( C \). It simplifies \( p \) while its initial or its separant lies in \( p \). It returns the simplified value of \( p \) together with an updated value of \( P \).
Let \( G = \langle A, D, P, S \rangle \) be a quadruple.

I1 \( p = I(G) \);
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 \( p' = \text{lo}(\{ p, p' \}) \) then \( p' \in I^{\text{id}}_{p'}(G) \).

**Figure 5:** Invariant properties kept by PARDI

---

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

**Figure 6:** The ensure_rank function
2.3. 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 7.

**Proposition 2.1:** (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 1.3. \( \square \)

Before proving Proposition 2.2, 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

\[
I^{v_1}(G) \subset I^{v_2}(G) \subset I^{v_3}(G) \subset \cdots
\]

\[
\cap \quad \cap \quad \cap
\]

\[
I^{v_1}(G') \subset I^{v_2}(G') \subset I^{v_3}(G') \subset \cdots
\]

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 2.1:** 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 \( \text{hi}(D') \) which is part of the associated system of \( G' \).

Proposition 1.5 (one needs this stronger form of Proposition 1.4 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 \( v \).

\( \square \)

**Proposition 2.2:** (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 \( v \).
function complete((A, D, P, S), C, p)
Assumptions
G = (A, D, P, S) is a quadruple satisfying properties I1 to I5
p is a differential polynomial which lies in 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 p
C is a characteristic set of the differential prime ideal p
begin
(A', S') := insert_and_rebuild(p, A, C)
D' := D ∪ \{{p_ℓ, p}\} | p_ℓ ∈ A, \{p_ℓ, p\} is a critical pair
P' := P
S' := S ∪ \{i_p, s_p\}
return (A', D', P', S')
end

function insert_and_rebuild(p, A, C)
begin
\overline{A} := \{f ∈ A | \text{ld } f < \text{ld } p\}
Observe that the leaders of \overline{A} are not derivatives of \text{ld } p
B := \{p\} ∪ \{f ∈ A | \text{ld } f > \text{ld } p \text{ and } \text{ld } f \text{ is not a derivative of } \text{ld } p\}
Recall that \text{ld } p \text{ is distinct from the leaders of } \overline{A}
Denote B = \{p_1, ..., p_t\} (s.t. \text{ld } p_i < \text{ld } p_{i+1})
\overline{S} := ∅
for k := 1 to t do
\overline{p}_k := partial_rem(p_k, \overline{A})
Denote $i_{p_k}$ and $s_{p_k}$ the initial and the separant of $\overline{p}_k$
(\overline{A}, newS) := saturate(\overline{A}, i_{p_k}, C)
\overline{S} := \overline{S} ∪ newS
\overline{A} := \overline{A} ∪ \{\overline{p}_k\}
(\overline{A}, newS) := saturate(\overline{A}, s_{p_k}, C)
\overline{S} := \overline{S} ∪ newS
od
return (\overline{A}, \overline{S})
end

Figure 7: The complete function and its insert_and_rebuild subfunction
Proof: Lemma 2.1 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 $p \in p$, 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 $p$ which is prime, after this operation, one has $I(G') \subset p : (i_p, s_p)^\infty = p$. Some algebraic operations are performed by saturate on $A'$. Proposition 1.4, which describes them, shows that $(A') : h^\infty \subset p$. 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 $A$ is squarefree comes from the fact that $A$ is squarefree, combined to Lemma 1.3. After the first call to saturate, $A$ is still squarefree by Proposition 1.4. Just before the second call to saturate, $A$ 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 2.1 and Lemma 1.8. The fact that saturate stores in newS (see Figure 4) the factor $g$ or $h$ which does not lie in $p$ permits to apply Lemma 1.8.

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 $p$.

Property I5. It is satisfied by $G$ hence, using Lemma 2.1, 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 $\{p, p\}$ with $p = \lo(\{p, p\})$. Since $p \in A'$ we have $p \in I^p(G')$. Thus property I5 is satisfied by $G'$.

2.3.1. 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 in [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 2.3:** 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 in [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 Rosenfeld’s lemma [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.

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

In this section one studies the function \( \text{lsr} \) described in Figure 8 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. \( \text{lsr} \) is actually called by \( \text{PARDI} \) 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 \( p \). One introduces the following notations:

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

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

**Proposition 2.4:** (termination)

The function \( \text{lsr} \) 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. \( \square \)

**Proposition 2.5:** (specifications of \( \text{lsr} \))

The \( \text{lsr} \) function returns a triple \( (g, \text{newP}, \text{newS}) \) satisfying the properties:

1. \( g \) is a gcd of \( a \) and \( b \) in the ring \( \text{Fr}(R^- / p^-)[x] \)
2. \( \text{deg}(g, x) > 0 \) and its initial and separant do not lie in \( p \)
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 \( \text{newS} \) and \( G' = \langle A, D, \text{newP}, \text{newS} \rangle \).

The sets \( \text{newP} \) and \( \text{newS} \) are updated version of \( P \) and \( S \) obtained by applying the “master–student” relationship idea described in the beginning of section 2.

**Proof:** Observe that the pseudocode of \( \text{lsr} \) is nothing but the Euclidean algorithm applied on \( a \) and \( b \) in \( \text{Fr}(R^- / p^-)[x] \) together with instructions which store in \( \text{newP} \) every leading coefficient which is zero in \( R^- / p^- \) but not reduced to
function $\text{lsr}(a, b, x, G = (A, D, P, S), C)$

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 $I_1$ to $I_5$ 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 $q \neq 0$ do

$r := \text{prem}(p, q, x)$

while $r \neq 0$ and $lcoeff(r, x) \in p$ do

if $\text{prem}(lcoeff(r, x), A) \neq 0$ then

$newP := newP \cup \{lcoeff(r, x)\}$
fi

$r := \text{reductum}(r, x)$

od

if $r \neq 0$ then

$newS := newS \cup \{lcoeff(r, x)\}$
$p := q$
$q := r$

fi

od

$g := p$

return $(g, newP, newS)$

depth end

Figure 8: The $\text{lsr}$ function

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 $R^-/p^-$).

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

**Item 2.** All the computed pseudoremainders belong to the ideal $(a, b)$ of the ring $\text{Fr}(R^-/p^-)[x]$. Since $a, b \in p$, all the computed pseudoremainders lie in $p$ thus the first pseudoremainder which does not depend on $x$, lies in $p^-$, hence is zero in $\text{Fr}(R^-/p^-)[x]$. 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 \( p \). Thus the initial of \( g \) does not lie in \( p \).

The fact that the separant of \( g \) does not lie in \( p \) 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, paragraph 5] of \( p \). 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 \( p \). 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, \( s_g \notin p \). 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 \( \text{newP} \) and the leading of the coefficients which are considered as nonzero are stored in \( \text{newS} \), 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.

2.4.1. 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] but the Lombardi et al. [2000] algorithm would fit as well).

Such an algorithm actually computes a sequence of subresultants \( p_1, \ldots, p_n \) of \( a \) and \( b \) in \((R^{-}/p^-)[x]\).

The only issue with such efficient algorithms consists in performing the exact quotient operations of the algorithm in \( R^{-}/p^- \). 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 \( R^{-}/p^- \). Assume this is the case. Then one continues the Ducos [2000] algorithm without normalizing \( p_i \) in any sense w.r.t. \( p \). Assume the leading coefficients of all the encountered subresultants are nonzero in \( R^{-}/p^- \). 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 \( R^{-}/p^- \). 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 $R^-/p^-$ 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.

2.4.2. Identifying algebraic subproblems

The lsr algorithm is purely algebraic in the following sense:

1. it does not manipulate the separants of the polynomials $p$ and $q$ ;
2. it does not generate any critical pair.

It is going to be used by the PARDI algorithm when two differential polynomials having the same leader are encountered. This is a true major improvement w.r.t. the Rosenfeld–Gröbner algorithm of the MAPLE diffalg package as explained in [Boulier et al., 2001].

2.5. The main algorithm

In this section, one studies the main function PARDI, described in Figure 9.

The missing final step. In this paper, PARDI is presented as returning a regular differential system (defined later) $A = 0$, $S \neq 0$. 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 in [Boulier et al., 2001]. Another one is given in [Boulier, 2006, regalise algorithm, sketched in section 6.2.2]. One does not give any here, in order to shorten the paper.

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 permits to completely avoid inequations and thereby simplifies the pseudocodes given in this paper.

**Proposition 2.6:** (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, Section I.10]. This rank cannot strictly decrease
function PARDI(\(C, \mathcal{R}, \mathcal{R}'\))

Assumptions

- \(C\) is a characteristic set of \(p\) w.r.t ranking \(\mathcal{R}\)
- \(\mathcal{R}\) is a ranking
- \(\mathcal{R}'\) is another (target) ranking

\[
\langle A, D, P, S \rangle := \langle \emptyset, \emptyset, C, H_C \rangle \text{ taken w.r.t. } \mathcal{R}
\]

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 \(p = \Delta(p_1, p_2)\).

\[\bar{p} := \text{partial\_rem}(p, A)\]

\[(\bar{p}, P) := \text{ensure\_rank}(\bar{p}, G = \langle A, D, P, S \rangle, C)\]

if \(p \neq 0\) then

if there exists some \(q \in A\) such that \(\text{ld } \bar{p} = \text{ld } q\) then

\[\langle g, P, S \rangle := \text{lsr}(\bar{p}, q, \text{ld } q, \langle A, D, P, S \rangle, C)\]

if \(g \neq q\) then

Instead of calling \text{complete}, one could actually just replace \(q\) by \(g\) in \(A\) and all the critical pairs of \(D\). The key argument is in Lemma 1.8. We choose not to do it in this paper to shorten proofs.

\[\langle A, D, P, S \rangle := \text{complete}(\langle A \setminus \{q\}, D, P, S \rangle, C, g)\]

enlarge \(S\) with \(\text{pquo}(q, g)\)

fi

else

\[\langle A, D, P, S \rangle := \text{complete}(\langle A, D, P, S \rangle, C, \bar{p})\]

fi

fi

od

\[S := \text{partial\_rem}(S, A)\]

return \((A, S)\)

end

\[\text{Figure 9: The main function PARDI}\]

at each turn by [Kolchin, 1973, Proposition 3, page 81]. 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 \text{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 \(p\).

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 \text{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, page 75]. By a classical argument of graph theory [König, 1950, Satz 6.6] (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 $I_1$ to $I_5$ are loop invariants of \textsc{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 preceding Lemma 2.1.

**Lemma 2.2:** 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 $I_1$ to $I_5$ then $I^v(G) \subseteq I^v(G')$ 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 $hp = \overline{p} \mod I^v(G)$. The call to \textsc{ensure_rank} may modify $p$ but stores in $P$ the initials and separators needed to keep this relation true.

Observe that, strictly speaking, $G$ does not satisfy $I_1$ to $I_5$ 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 \textsc{complete}, the withdrawal of $q$ from $A$ is also delayed. Therefore one assumes in the following text that $G$ does satisfy $I_1$ to $I_5$ before any call to \textsc{complete} or \textsc{lsr}. Three subcases need to be considered.

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

Second subcase: $p \neq 0$ and there does not exist any $q \in A$ having the same leader as $\overline{p}$. Then \textsc{complete} is called and, using Proposition 2.2 plus the fact that $G$ satisfies properties $I_1$ to $I_5$, the lemma is proven.

Third subcase: $p \neq 0$ and there exists some $q \in A$ having the same leader as $\overline{p}$. Then, by Proposition 2.5, the call to \textsc{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 \textsc{lsr}, $R^-$ denotes the ring of the differential polynomials depending on derivatives strictly less than $v$ and $I^-(G)$ is defined as in section 2.4. This gcd is inserted in $G$ by \textsc{complete} hence, using Proposition 2.2 plus the fact that $G$ satisfies properties $I_1$ to $I_5$, 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 $I^v(G) \subseteq I^v(G')$ 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 $p = \text{hi}(\{p, p'\})$ and denote $v = \text{ld} \ p$. Since the critical pair is a reduction one, $\Delta(p, p') = \text{prem}(p, \phi p')$ for some differential operator $\phi$ such that $\text{ld} \ \phi p' = v$. Using the fact that $p' = \text{lo}(\{p, p'\} )$ 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 2.3: 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 $p \subset I(G)$ comes from Lemma 2.2. The converse inclusion is clear.

Property I2 comes from Proposition 2.2.

Property I3 (sketched). The critical pairs solved by $G$ are solved by $G'$. The key arguments are given in Lemma 2.2 and Lemma 1.8. Storing $\text{pquo}(q, g)$ in $S$ after the first call to complete permits to apply Lemma 1.8.

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 $\text{hi}(\{p, p'\})$.

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 2.2.

Property I5. The case of the reduction critical pairs generated by complete is considered in Proposition 2.2. That of the other ones is solved by Lemma 2.2. \qed

The following definition is borrowed from [Boulier et al., 1997].

Definition: 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 2.7: (specification of PARDI)

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

Proof: The returned quadruple $G$ satisfies properties $I_1$ to $I_5$ by Lemma 2.3. It also satisfies $D = P = \emptyset$. Property $I_2$ implies property $C_1$. Property $I_4$ and the fact that PARDI partially reduces the elements of $S$ by $A$ before returning implies that $C_2$ holds. Property $I_3$ combined with the fact that $D$ is empty implies that $C_3$ holds. Therefore $A = 0$, $S \neq 0$ is a regular differential system. Property $I_1$ combined to the fact that $D = P = \emptyset$ implies that $[A] : S^\infty = p$. □

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]. This lemma reduces the problem of computing a differential characteristic set to the problem of computing a non differential one. See the comments at the beginning of section 2.5. As explained in the introduction, one does not develop this part in this paper.

2.6. A detailed example

This last section does not bring any new result.

One just illustrates the behaviour of PARDI over the first example given in introduction and verifies that some of the properties stated in Figure 5 hold at each step. The criterion stated in Proposition 2.3 is applied. The reader may check that the avoided corresponding $\Delta$–polynomial are reduced to zero by the set $A$ of the quadruple.

This implementation of PARDI was done using the BLAD libraries. These libraries can be freely downloaded from [Boulier, 2004]. This example is available in the file bad.pardi located in the examples directory of the libraries.

In this example, a slight variant of $I_2$ is maintained and full reductions are performed instead of partial ones. The strategy consists in picking differential polynomials in $P$ first and critical pairs in $D$ only if $P$ is empty. The lists $P$ and $D$ are sorted according to some heuristic criterion which does not need to be stated. Each time a differential polynomial is written, its leader w.r.t. the ranking $\overline{R}$ occurs at the leftmost place: it is the first derivative written.

When PARDI first enters the loop, $A$ and $D$ are empty, all elements of $C$ belong to $P$ and the initials and separatants of $C$ taken w.r.t. the ranking $\overline{R}$ lie in $S$, i.e.

$A = \emptyset$
$D = \emptyset$
$P = [u_y^2 - 2u, -u_x + v_{xx}, u_x^2 - 4u, -u_xu_yu + u_xu_y + 4uv_y]$
$S = [u, u_x, u_y]$

Since $F = P = C$ and $S = H_C$ invariant $I_1$ is clearly satisfied. The other invariants are initially trivial.

Remark. Since inequations are not involved in the computations, one does not provide the value of $S$ in the next steps. See the comments at the beginning of section 2.5.
At the first turn, the differential polynomial $u_y^2 - 2u$ is picked and removed from $P$. The complete subfunction stores it in $A'$.

\[
\begin{align*}
A &= [u_y^2 - 2u] \\
D &= [] \\
P &= [-u_x + v_{xx}, u_x^2 - 4u, -u_x u_y u + u_x u_y + 4uv_y]
\end{align*}
\]

At the second turn the differential polynomial $-u_x + v_{xx}$ is picked and removed from $P$. The complete subfunction inserts it in $A'$ (after normalizing its sign) and stores in $D'$ a critical pair generated by this differential polynomial and the one already present in $A$.

\[
\begin{align*}
A &= [u_y^2 - 2u, u_x - v_{xx}] \\
D &= \{(u_x - v_{xx}, u_y^2 - 2u)\} \\
P &= [u_x^2 - 4u, -u_x u_y u + u_x u_y + 4uv_y]
\end{align*}
\]

At the third turn the differential polynomial $u_y^2 - 4u$ is picked and removed from $P$. After reduction by $A$ it becomes $4u - v_{xx}^2$. The complete subfunction stores it in $A'$ and stores in $D'$ the two (reduction) critical pairs generated by this differential polynomial and the elements of $A$. The elements $u_y^2 - 2u$ and $u_x - v_{xx}$ of $A$ are not kept in $A'$. These two differential polynomials belong however to $\text{hi}(D')$ so that (see the proof of Lemma 2.1):

\[
F = A \cup \text{hi}(D) \cup P = F' = A' \cup \text{hi}(D') \cup P'.
\]

Thus $I(G) = I(G')$ and invariant $\mathbf{I}_1$ still holds. The old critical pair in $D$ is not kept in $D'$ by Proposition 2.3.

\[
\begin{align*}
A &= [4u - v_{xx}^2] \\
D &= \{(4u - v_{xx}^2, u_y^2 - 2u), (4u - v_{xx}^2, u_x - v_{xx})\} \\
P &= [-u_x u_y u + u_x u_y + 4uv_y]
\end{align*}
\]

At the fourth turn the differential polynomial $-u_x u_y u + u_x u_y + 4uv_y$ is picked and removed from $P$. The differential polynomial obtained after the reduction by $A$ has a nontrivial content. The implementation of PARDI verifies that this content does not lie in $p$ and removes it. The obtained differential polynomial is $v_{xxx}v_{xyy}v_{xx}^2 - 4v_{xxx}v_{xyy} - 16v_y$. It is stored in $A'$.

\[
\begin{align*}
A &= [v_{xxx}v_{xyy}v_{xx}^2 - 4v_{xxx}v_{xyy} - 16v_y, 4u - v_{xx}^2] \\
D &= \{(4u - v_{xx}^2, u_y^2 - 2u), (4u - v_{xx}^2, u_x - v_{xx})\} \\
P &= []
\end{align*}
\]

At the fifth turn, the critical pair $\{4u - v_{xx}^2, u_y^2 - 2u\}$ is picked and removed from $D$. The $\Delta$–polynomial is computed, reduced by $A$. The content of the
result is removed. This provides the new differential polynomial \(v_{xyy}^2 - 2\). It is stored in \(A'\) and one critical pair is stored in \(D'\). The implementation of `insert_and_rebuild`, which maintains a variant of invariant \(I_2\), multiplies the differential polynomial with leader \(v_{xxx}\) by the algebraic inverse \(v_{xyy}/2\) of \(v_{xyy}\) modulo \((v_{xyy}^2 - 2)\) and simplifies using the new differential polynomial.

\[
A = [v_{xyy}^2 - 2, v_{xxx}v_{xy}^2 - 4v_{xxx} - 8v_{xyy}v_y, 4u - v_{xx}^2] \\
D = [(4u - v_{xx}^2, u_x - v_{xx}), \{v_{xyy}^2 - 2, v_{xxx}v_{xy}^2 - 4v_{xxx} - 8v_{xyy}v_y\}] \\
P = []
\]

At the sixth turn, the critical pair \(\{4u - v_{xx}^2, u_x - v_{xx}\}\) is picked and removed from \(D\). The \(\Delta\)-polynomial is computed and reduced by \(A\). The content of the result is removed. This provides the new differential polynomial \(p = 4v_{xyy}v_y - v_{xx}^2 + 4\). Since \(p\) has the same leader as the element \(q = v_{xyy}^2 - 2 \in A\) the \(lsr\) function is called in order to compute a gcd \(g \in R^- = K[w \in \Theta U \mid w < v_{xyy}]\). Here \(R^- = K[w \in \Theta U \mid w < v_{xyy}]\). The pseudoremainder \(\text{prem}(q, p, v_{xyy})\) is the polynomial \(r = v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16\). It does not involve \(v_{xyy}\) thus it is equal to its leading coefficient w.r.t. this derivative. The characteristic set \(C\) of \(p\) pseudoreduces \(r\) to zero (notice that Proposition 2.5 proves this fact without having to perform the pseudoreductions). Thus the gcd \(g = p\). Since \(r\) is not reduced to zero by \(A\), it is stored in \(P\). The old critical pair in \(D\) is removed using Proposition 2.3.

\[
A = [4v_{xyy}v_y - v_{xx}^2 + 4, v_{xxx} - 2, 4u - v_{xx}^2] \\
D = [(4v_{xyy}v_y - v_{xx}^2 + 4, v_{xxx} - 2)] \\
P = [v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16]
\]

At the seventh turn the differential polynomial \(v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16\) is picked and removed from \(P\). It is stored in \(A'\) and throws away the differential polynomials with leaders \(v_{xyy}\) and \(v_{xxx}\). Two reduction critical pairs are stored in \(D'\) but the old critical pair can be removed using Proposition 2.3.

\[
A = [v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16, 4u - v_{xx}^2] \\
D = [(v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16, v_{xxx} - 2), \{v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16, 4v_{xyy}v_y - v_{xx}^2 + 4\}] \\
P = []
\]

At the eighth turn, the first critical pair is picked and removed from \(D\). After reduction by \(A\), it provides a differential polynomial \(p = v_{xx}^3 - 4v_{xx} - 8v_{xyy}v_y\). The \(lsr\) function is called on \(p\) and the polynomial \(q \in A\) with rank \(v_{xx}^4\). It returns a gcd \(g = v_{xxx}v_{xy}^2 - 2v_{xx} - 4v_{xyy}v_y\) (after removal of its content). It stores in \(P'\) a differential polynomial representing the resultant of \(p\) and \(q\) (after removal of its content) but which is not reduced to zero by \(A\). The gcd is stored in \(A'\) and
replaces the differential polynomial with rank $v_{xx}^4$. The function `insert_and_rebuild` pseudoreduces the differential polynomial with leader $u$ using it. No critical pair is generated.

\[
A = [v_{xx}v_{xy}^2 - 2v_{xx} - 4v_{xy}v_y, uv_{xy}^4 - 4uv_{xy}^2 + 4u - 4v_{xy}^2v_y^2]
\]

\[
D = [(v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16, 4v_{xxx}v_y - v_{xx}^2 + 4)]
\]

\[
P = [v_{xy}^4 - 4v_{xy}^2 - 8v_y^2 + 4]
\]

At the ninth turn, the differential polynomial $v_{xy}^4 - 4v_{xy}^2 - 8v_y^2 + 4$ is picked and removed from $P$. It is stored in $A'$. One critical pair is stored in $D'$.

\[
A = [v_{xy}^4 - 4v_{xy}^2 - 8v_y^2 + 4, 2v_{xx}v_y - v_{xy}^2 + 2v_{xy} + 2u - v_{xy}^2]
\]

\[
D = [(v_{xx}^4 - 8v_{xx}^2 - 32v_y^2 + 16, 4v_{xxx}v_y - v_{xx}^2 + 4),
      \{v_{xy}^4 - 4v_{xy}^2 - 8v_y^2 + 4, 2v_{xx}v_y - v_{xy}^2 + 2v_{xy}\}]
\]

\[
P = []
\]

At the tenth turn, the first critical pair is picked and removed from $D$. The $\Delta$-polynomial is computed and reduced by $A$. The content of the result is removed. One gets a differential polynomial $p = v_{xy}^4 - 2v_{xy} - 4v_{yy}v_y$. The situation is very similar to that of the eighth turn. The lsr function is called on $p$ and the polynomial $q \in A$ with rank $v_{xy}^4$. It returns a gcd $g = v_{xy}v_{yy}^2 - v_{xy} - 2v_{yy}v_y$ (after removal of its content). It stores in $P'$ a differential polynomial representing the resultant of $p$ and $q$ (after removal of its content) but which is not reduced to zero by $A$. The implementation of `insert_and_rebuild` pseudoreduces the differential polynomial with leader $v_{xx}$ using it. The gcd is stored in $A'$ and replaces the differential polynomial with rank $v_{xy}^4$. The function `insert_and_rebuild` pseudoreduces the differential polynomial with leader $v_{xx}$ using it. A critical pair is stored in $D'$. An old critical pair is removed from $D$ using Proposition 2.3.

\[
A = [v_{xy}v_{yy}^2 - v_{xy} - 2v_{yy}v_y,
     v_{xx}v_{yy}^6 - 3v_{xx}v_{yy}^4 + 3v_{xx}v_{yy}^2 - v_{xx} + 2v_{yy}^5 - 4v_{yy}v_y^2 - 4v_{yy}^3 + 2v_{yy},
     uv_{yy}^4 - 2uv_{yy}^2 + u - 2v_{yy}^2v_y^2]
\]

\[
D = [(v_{xy}v_{yy}^2 - v_{xy} - 2v_{yy}v_y,
     v_{xx}v_{yy}^6 - 3v_{xx}v_{yy}^4 + 3v_{xx}v_{yy}^2 - v_{xx} + 2v_{yy}^5 - 4v_{yy}v_y^2 - 4v_{yy}^3 + 2v_{yy}]}
\]

\[
P = [v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1]
\]

At the eleventh turn, the differential polynomial $v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1$ is picked and removed from $P$. It is inserted in $A'$. The function `insert_and_rebuild` pseudoreduces the other differential polynomials of $A'$ using it and removes the contents. This simplifies $A'$. An analogue of Buchberger's second criterion [Boulier et al., 1997, Proposition 4.2] not stated in this paper permits us to generate only
one critical pair instead of two.

\[ A = [v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1, v_{xy}v_y - v_{yy}^3 + v_{yy}, v_{xx} - 2v_{yy}, u - v_{yy}^2] \]

\[ D = \{\{v_{xy}v_y - v_{yy}^3 + v_{yy}, v_{xx} - 2v_{yy}\}, \{v_{yy}^4 - 2v_{yy}^2 - 2v_y^2 + 1, v_{xy}v_y - v_{yy}^3 + v_{yy}\}\} \]

\[ P = [] \]

There are two critical pairs left. At the next steps, A pseudoreduces the first one to zero, it pseudoreduces the second one to zero and PARDI returns A and the set of inequations S.

References


