A new algorithm for computing regular representations for radicals of parametric differential ideals

Shahnaz Fakouri¹, Sajjad Rahmany¹* and Abdolali Basiri¹

Abstract: The regular representation of the radical of a differential ideal has various applications such as solving the membership problem, computing Taylor expansion of solutions, finding the Lie symmetries, and solving dynamical systems. Presently, there is no algorithm giving all regular representations for all possible values of the parameters for a polynomial differential ideal with parametric coefficients. In this article, we propose a new algorithm that computes all different regular representations with respect to all possible states of the parameters. Also, we present an efficient criterion to reduce some ineffectual computations. Implementing the algorithm in Maple and several examples reported in this article demonstrate the high efficiency of the algorithm.

Subjects: Algebra; Differential Equations; Pure Mathematics; Computer Mathematics

Keywords: differential algebra; parametric differential system; Rosenfeld-Gröbner algorithm

MR Subject classifications: 68W30; 12H05; 35C11

1. Introduction

Differential algebra was initially founded in the early 20th century to study a system of ordinary or partial differential equations, from a totally algebraic point of view. This mathematical approach was developed by Ritt (1950) and Kolchin (1973) who also prepared a computational algebraic tool for solving partial differential equation (PDE) systems. The well-known Rosenfeld-Gröbner algorithm is one of the most important algorithms in this regard, which indeed can be considered as the beginning point of the computational differential algebra. This algorithm analyses a...
A polynomial differential system in a differential algebraic mode and gives a regular representation of the radical differential ideal generated by this system. The algorithm has various applications for instance in Geometry (Sabzevari, Hashemi, Alizadeh, & Merker, 2016), finding the Lie symmetries of differential equations (Hydon, 2000) and theory of differential equations (Mansfield & Clarkson, 1997). The Rosenfeld-Gröbner algorithm was first described by Boulier (1994) and improved by Boulier, Lazard, Ollivier, and Petitot (1995) thereafter. The input of this algorithm is a polynomial differential system in a differential ring with constant coefficients while its output is a representation of radical of the differential ideal, generated by the input system, as a finite intersection of radical differential ideals. The computed representation allows testing membership in the radical of the ideal generated by given system and also computing Taylor expansions of solutions of this system (Boulier, Lazard, Ollivier, & Petitot, 2009).

As mentioned before, the input of this algorithm is a differential system with constant coefficients. However, sometimes, it is necessary to determine such a representation for a parametric differential system; a differential system that its coefficients depend on some parameters. For instance this system can be used for biological modelling (Boulier, 2007), dynamical systems (Harrington & Van Gorder, 2017; Jauberthie, Trave-Massuyes, & Verdiere, 2016) and so on. In this regard, even if the coefficients field is extended by these parameters, the output of the Rosenfeld-Gröbner algorithm may not include all of the representations for all possible values of the parameters. For example, consider the following differential system whose coefficients depend on some parameters.

\[
P_0 = \begin{cases} 
\alpha \left( \frac{\partial^2}{\partial x^2} v(x, y) \right)^2 + 3bu(x, y) = 0 \\
\left( \frac{\partial^2}{\partial y^2} v(x, y) \right) \left( \frac{\partial}{\partial x} u(x, y) \right) - v(x, y) = 0 \\
c \left( \frac{\partial}{\partial x} v(x, y) \right) + u(x, y) + 3 = 0 
\end{cases}
\]

To simplify, we use indices:

\[
P'_0 = \begin{cases} 
av^2_{xy} + 3bu = 0 \\
v_{xy}u_x - v = 0 \\
cv_x + u + 3 = 0 
\end{cases}
\]

In the extended coefficient field by the parameters, with respect to the orderly ranking with considering \( u < v \) and the graded lexical order \( y < x \), the output of the Rosenfeld-Gröbner algorithm for the system \( P_0 \) is empty. But, is this system really inconsistent for arbitrary values of the parameters? It can be easily shown that the answer is “no”. This system has the following representations for different values of the parameters \( a \) and \( b \). When \( b = 0 \) and \( a \neq 0 \) we have:

\[
\sqrt{[P_0]} = [v, u + 3]
\]

and also when \( b = a = 0 \) and \( c \neq 0 \) we have:

\[
\sqrt{[P'_0]} = [cv + u_xu_y, u_xu_{xy} + u_{xx}u_{yy} - u - 3] \cap [v, u + 3]
\]

Presently, there is no context which introduces an algorithm giving all representations for all possible values of the parameters for a parametric system. Although, Boulier (2018) explains how the Rosenfeld-Gröbner algorithm can manage different types of the parameters. This context was written from our demand and it was never published.

The main goal of this article is to obtain the distinct representations for all possible values of the parameters. According to this viewpoint, a case is a set of polynomial equalities and inequalities over the parameters accepting the same representation. Here, we present an algorithm with branching style that its root vertex corresponds to the inputted system. The inputted system is a differential system whose coefficients depend on some parameters. This algorithm branches some
vertices according to possible values of the parameters and outputs the distinct representation depending on these values of parameters. Moreover, in this algorithm we apply a new criterion which can detect and delete a chain of the redundant computations.

The remainder of this article is organised as follows. We give an introduction to differential algebra in Section 2. In Section 3, we briefly describe the Rosenfeld-Gröbner algorithm. Section 4 describes the new algorithm to decompose a parametric differential system. Finally, some examples are presented in Section 5.

2. Differential algebra
In this section, we review some preliminaries concept of differential algebra. More details can be found in (Hubert, 2000; Kolchin, 1973; Ritt, 1950; Sit, 2002). In the following, we assume that $K$ is a field of characteristic zero.

Definition 2.1. A differential ring is a commutative ring $R$ associated with a finite set of derivations $\delta_1, \ldots, \delta_m$, which commute with each other and satisfy the following equalities:

$$\delta_i(r_1 r_2) = \delta_i(r_1) r_2 + r_1 \delta_i(r_2)$$

where $1 \leq i \leq m$ and $r_1, r_2 \in R$. In a special case, if $m = 1$, then $R$ is called an ordinary differential ring and if $m > 1$, then $R$ is called a partial differential ring.

A differential ideal $I$ of $R$ is an ideal which is also closed under the action of derivations; i.e., $\delta_i(I) \subseteq I$ for each $1 \leq i \leq m$.

We will use $\Theta$ to denote the free multiplicative semi-group generated by the set $\{\delta_1, \ldots, \delta_m\}$. It is easy to see that the elements of $\Theta$, which are called derivative operators of $R$, are in the form of

$$\theta = \delta_{d_1}^{d_1} \delta_{d_2}^{d_2} \cdots \delta_{d_m}^{d_m},$$

where each $d_i$ belongs to $\mathbb{N} \cup \{0\}$ and the sum $\text{ord}(\theta) = \sum_{i=1}^{m} d_i$ is the order of $\theta$.

For each $S \subseteq R$, the smallest subset of $R$ containing $S$, which is also stable under the action of $\Theta$ is denoted by:

$$\Theta S = \{ \theta(s) | \theta \in \Theta, s \in S \}.$$

We denote by $(S)$ (resp. $[S]$) the smallest algebraic (resp. differential) ideal of $R$ containing $S$. If $R_0$ is a subring of $R$, using $R_0[S]$ (resp. $R_0[S]$), we denote the smallest algebraic (resp. differential) subring of $R_0$ containing $S$ and $R_0$ that are generated by $S$ on $R_0$. As a simple fact, we apply an algebraic approach on differential ideals and subrings due to $[S] = (\Theta S)$ and $R_0[S] = R_0[\Theta S]$.

Now, we present the following definition which has an important role in the main concepts.

Definition 2.2. Consider a differential ideal $I$ and a set of differential polynomials $S$ from $R$. The saturation ideal of $I$ by $S$ is defined as follows:

$$I : S^\infty = \{ p \in R | sp \in I \text{ for some } s \in S^\infty \},$$

where $S^\infty$ denotes the smallest multiplicative family of $R$ which contains $S$.

In the sequel, we will deal with a differential polynomial ring $R = K\{u_1, \ldots, u_n\}$ that is a differential ring generated by $U = \{u_1, \ldots, u_n\}$ on $K$, where $K$ is a differential field that for each $\theta$ in $\Theta$ and each constant $k$ in $K$ we have $\theta(k) = 0$. In this ring, each $u_i$ (resp. $\theta u_i$) is called an indeterminate (resp. a derivative).

In the next definition, we recall the concept of ranking that will be used to compare derivatives. In fact, rankings in a differential ring play a similar role of monomial orderings in polynomial rings.
Definition 2.3. Let $\mathcal{R}$ be a differential polynomial ring. Each ranking over $\mathcal{R}$ is an ordering $\prec$ over $\Theta \mathcal{R}$ such that for each derivation $\theta$ and $v, w \in \Theta \mathcal{R}$:

- $v \prec \theta v$, and
- $v \prec w$ implies $\theta v \prec \theta w$.

Orderings are divided into two different types: Rankings for which $\text{ord}(\theta) < \text{ord}(\phi)$ implies that $\theta(v) < \phi(w)$, which are called orderly, and those for which $v \prec w$ concludes $\theta v \prec \phi w$, which are called elimination rankings.

Now, we introduce some terms of differential polynomials. Let $p \in \mathcal{R} = K\{u_1, \ldots, u_n\}$ be a differential polynomial and $\prec$ be a ranking over $\Theta \mathcal{R}$. The leader $\text{ld}(p)$ of $p$ is the highest derivative appearing in $p$ with respect to $\mathcal{R}$. If $\text{ld}(p) = u$ and $d$ is the degree of $u$ in $p$, then the initial $I_p \in \mathcal{R}$ is defined to be the coefficient of $u^d$ in $p$. The separant $S_p$ of $p$ is the polynomial $\partial p / \partial u$. The rank of $p$ is the monomial $u^d$, which is denoted by $\text{rank } p$. The rank of a set of polynomials is similarly the set of ranks of the elements of the set. Let $A = \{p_1, \ldots, p_m\}$ and $A' = \{q_1, \ldots, q_m'\}$ be two non-empty subsets of $\mathcal{R}\setminus K$. Assume that, $\text{rank } p_i \leq \text{rank } p_{i+1}$ and $\text{rank } q_j \leq \text{rank } q_{j+1}$ for all $1 < m, j < m'$. The set $A$ is said to be of lower rank than $A'$, when either there exist some $i \leq \min(m, m')$ such that $\text{rank } p_i < \text{rank } q_j$, and $\text{rank } p_j = \text{rank } q_{j+1}$ for $1 \leq j < m$ and $m'$ and $\text{rank } p_1 = \text{rank } q_1$ for $1 \leq j \leq m'$. Two sets of polynomials that none of them is of lower rank than the other one are called of the same rank.

For any subset $A$ of $\mathcal{R}\setminus K$, we denote the set of the initials (resp. separants) of the elements of $A$ by $I_A$ (resp. $S_A$). Let now $H_A = I_A \cup S_A$. Then, $H^\#_A$ denotes the set of all the power products of the initials and the separants of the elements of $A$.

Definition 2.4. Consider two differential polynomials $p$ and $q$ in the differential ring $\mathcal{R} = K\{u_1, \ldots, u_n\}$ and let $u^d$ be the rank of $p$. Polynomial $q$ is called partially reduced with respect to $p$ if no proper derivative of $u$ appears in $q$. Also, if the degree of $q$ in $u$ is less than $d$ we say that $q$ is algebraically reduced with respect to $p$. A polynomial $q$ is called reduced with respect to the polynomial $p$ if $q$ is partially and algebraically reduced with respect to $p$.

A set $A \subseteq \mathcal{R}\setminus K$ is said to be autoreduced if any element of $A$ is reduced with respect to any other element of the set. Also if the elements of $A$ are pairwise partially reduced and the leaders of them are pairwise different, we say that $A$ is differentially triangular.

One of the important procedures in the context of differential algebra is the way of reducing a differential polynomial with respect to an autoreduced set. Let $p$ be a differential polynomial and $\Delta = \{p_1, \ldots, p_r\}$ be an autoreduced set. By Ritt’s algorithm of reduction (see Kolchin 1973), there exists a differential polynomial $r$ and non-negative integers $b_1, \ldots, b_r$ and $a_1, \ldots, a_s$ such that

$$r = i_{p_1}^{b_1} \cdots i_{p_s}^{b_s} s_{p_1}^{a_1} \cdots s_{p_s}^{a_s} p \mod[\Delta],$$

where $i_{p_r}$ and $s_{p_r}$ denote the initial and the separant of $p_r$, respectively, and $r$ is reduced with respect to $A$. If $r$ is just partially reduced with respect to $A$, then we can write

$$r = s_{p_1}^{b_1} \cdots s_{p_s}^{b_s} p \mod[\Delta].$$

For the simplicity in notations, we denote $r$ by $\text{pfullrem}A$ (resp. $\text{pprem}A$), if $r$ is reduced (resp. partially reduced) with respect to $A$.

Now, we explain the concept of $\Delta$ – polynomial of two differential polynomials, which plays a similar role to $S$ – polynomial in algebraic polynomials ring. Before that, we need to know the
Definition of critical pairs (Boulier et al., 2009). Also, we should state that for each derivative \( \partial u \) and \( \phi u \) of the same differential indeterminate \( u \), the least common derivative is denoted by \( \text{lcm}(\partial u, \phi u) \), which equals to \( \text{lcm}(\theta, \phi)u \).

**Definition 2.5.** A set \( \{p_1, p_2\} \) of differential polynomials is said to be a critical pair if \( p_1 \) and \( p_2 \) are distinct and the leaders of them have common derivatives. If \( A \) is a set of differential polynomials then critical pairs \( \text{cp}(A) \) denotes the set of all the pairs which can be formed between any two elements of \( A \).

**Definition 2.6.** Let \( \{p_1, p_2\} \) be a critical pair and denote \( \theta_1 u = \text{ld}(p_1) \), \( \theta_2 u = \text{ld}(p_2) \) and \( \theta_{12} u = \text{lcd}(\theta_1 u_1, \theta_2 u_2) \). The \( \Delta \) – polynomial \( \Delta(p_1, p_2) \) of \( p_1 \) and \( p_2 \) is defined as follows:

\[
\Delta(p_1, p_2) = sp_1 \frac{\theta_{12}}{\theta_1} p_1 - sp_2 \frac{\theta_{12}}{\theta_2} p_2
\]

If \( A \) is a subset of \( \mathcal{R} \), then the set of all \( \Delta \) – pols \( \text{A} \) denotes the set of all possible \( \Delta \) – polynomials which can be formed between any two elements of \( A \).

Here, we give the definition of solved pairs, which was defined earlier by Boulier et al. (2009) for the first time.

**Definition 2.7.** A critical pair \( \{p_1, p_2\} \) is said to be solved by a differential system of equations and inequations \( A = 0 \) and \( S \neq 0 \) if there exists a derivative \( v < \text{lcm}(p_1, p_2) \) such that:

\[
\Delta(p_1, p_2) \in \mathcal{A}_{v} : (S \cap \mathcal{R}_v)^\infty
\]

where \( \mathcal{A}_{v} = \{ \theta p | p \in A, \theta \in \Theta, \text{ld}(\theta p) \leq v \} \) and \( S \cap \mathcal{R}_v = \{ p \in A | \text{ld}(p) \leq v \} \).

One can apply the following criterion (lemma 6 in Boulier et al., 2009) to recognise the solved pairs:

**Lemma 2.8.** Let \( \{p_1, p_2\} \) be a critical pair such that \( \text{ld}p_1 \neq \text{ld}p_2 \). If \( \Delta(p_1, p_2) \) is reduced to zero by a set \( A \) and \( S \) is a set of inequations containing \( H_A \); then, the critical pair \( \{p_1, p_2\} \) is solved by the system \( A = 0 \) and \( S \neq 0 \).

**Definition 2.9.** Let \( \mathcal{R} = K\{U\} \) be a differential polynomial ring and \( < \) be a ranking over \( \Theta U \). A differential system of equations \( A = 0 \) and inequations \( S \neq 0 \) of \( \mathcal{R} \) is said to be a regular differential system with respect to \( < \) if:

- \( A \) is differentially triangular
- \( S \) contains the separants of the elements of \( A \) and it is partially reduced with respect to \( A \)
- All critical pairs of \( A \) are solved by the system \( A = 0 \) and \( S \neq 0 \) (coherent property)

The differential ideal \( |A| : S^\infty \) is called the regular differential ideal defined by the system. Moreover, the system is considered inconsistent if \( |A| : S^\infty \) equals to \( \mathcal{R} \), and consistent otherwise.

At the end of this section, let us mention some properties and a brief explanation of Gröbner bases for an algebraic ideal (see Becker & Weispfenning, 1991; Cox, Little, & O’Shea, 1992), which is applied in the next sections. Consider \( R = K[x_1, \ldots, x_n] \) as a polynomial ring and \( <_x \) as a term ordering over \( R \). Note that the reduction in the polynomial ring is algebraic and the remainder of the reduction is denoted by algrem.

If \( G \) is a Gröbner basis with respect to \( <_x \) for ideal \( I = \langle A \rangle \) then we have:

- \( I = \langle G \rangle \), which means that \( G \) is also a generator for the ideal \( I \).
• If \( G \) is also reduced, then it is unique with respect to the ordering \( \prec \); in other words, for an ideal, there is just one reduced Gröbner basis with respect to certain ordering.

• The ideal \( I \) is equal to \( R \) (the system \( A = 0 \) has no answer) if and only if \( 1 \in G \) or the reduced Gröbner basis is equal with the set \( \{1\} \).

Note that having the following lemma (Boulier et al., 2009), even if the set of variables is infinite, Gröbner bases can be computed of finitely generated ideals of \( K[x_1, \ldots, x_n] \). So, we can compute Gröbner bases of (non-differential) ideals in differential polynomial rings.

**Lemma 2.10.** Let \( I \) be an ideal of a ring \( R \) and \( x \) be transcendental over \( R \). If \( \varphi \) denotes the canonical ring homomorphism \( \varphi : R \to R[x] \), then \( \varphi^{-1}(\varphi(r)) = r \).

Consider the system of equations \( A = \{p_1, \ldots, p_n\} \) and inequations \( S = \{h_1, \ldots, h_m\} \) of \( R \). To get a Gröbner basis \( G \) of \( (A) : S^n \), we just need to run the following steps:

1. Introduce a new set \( \{z_1, \ldots, z_m\} \) of unknowns and rewrite each inequation \( h_i \neq 0 \) as \( h_iz_i = 1 \).
2. Compute a Gröbner basis \( G_0 \) of the ideal generated with the set \( \{p_1, \ldots, p_n, h_1z_1 - 1, \ldots, h_mz_m - 1\} \) with respect to any ordering that eliminates the \( z \)'s.
3. Construct \( G \) by selecting the members of \( G_0 \) that do not involve \( z_i \) for any \( 1 \leq i \leq m \) \( (G = G_0 \cap R) \).

### 3. Rosenfeld-Gröbner algorithm

In this section, we describe the Rosenfeld-Gröbner algorithm (see Boulier et al., 2009 for more details). Let \( P_0 \subseteq R \), the Rosenfeld-Gröbner algorithm decomposes the differential radical ideal \( \sqrt{[P_0]} \) as a finite intersection of regular differential ideals \( P_i \)’s:

\[
\sqrt{[P_0]} = P_1 \cap \cdots \cap P_t
\]

Each regular differential ideal \( P_i \) is presented by two sets of differential polynomial equations \( A_i \) and differential polynomial inequations \( S_i \) such that \( P_i = [A_i] : S_i^n \). If \( G_i \) is a Gröbner basis for the saturation ideal \( P_i \), so \( G_i \) reduces to zero a differential polynomial \( p \) if and only if \( p \) belongs to \( P_i \). Although, the ideal membership problem is algorithmically undecidable (Umirbaev, 2016), this algorithm permits to test the membership in \( \sqrt{[P_0]} \).

This decomposition is not uniquely determined in general, not only because there might be redundant components (some of them may be contained in another) but also it depends on many choices made during its construction, such as the choice of ranking. The following theorem (Boulier et al., 2009) shows that there is such a decomposition for each system of differential polynomials.

**Theorem 3.1.** If \( P_0 = 0, S_0 \neq 0 \) is a differential system in a differential polynomial ring \( R \), then it is possible to compute finitely many consistent regular differential systems \( A_i = 0, S_i \neq 0 \) (1 \( \leq i \leq t \)) such that:

\[
\Gamma = \sqrt{[P_0]} : S_0^n = [A_1] : S_1^n \cap \cdots \cap [A_t] : S_t^n
\]

This decomposition may contain components redundant w.r.t. \( \Gamma \). Operations needed in this regard are addition, multiplication, differentiation and equality test with zero in the base field of \( R \).

A lemma of D. Lazard (Boulier et al., 1995) shows particularly that the regular ideals \( [A_i] : S_i^n \) are radical. The following theorem is a differential analogue to Hilbert’s theorem of zeros (Kolchin, 1973).
Theorem 3.2. (Theorem of zeros) Let $\mathcal{R} = K\{U\}$ be a differential polynomial ring over a differential field of characteristic zero and $P$ be a differential ideal of $\mathcal{R}$. A differential polynomial $p$ vanishes on every solution of $P$ in any differential field extension of $K$, if and only if $p \in \sqrt{P}$.

Corollary 3.3. A differential polynomial $p$ vanishes on every solution of a system of polynomial differential equations and inequalities $A = 0$, $S \neq 0$ if and only if $p \in \sqrt{|A| : S^\infty}$.

Here, we give a brief exposition of the current version of the Rosenfeld-Gröbner algorithm, which is presented in (Boulier et al., 2009). The input of this algorithm is a differential system of equations and inequalities with a ranking and the output is a set of differential regular ideals. Since a differential regular ideal is differentially triangular and coherent, the algorithm reduces each polynomial equation with respect to others and constructs possible $\Delta$-polynomials, which can be formed between each pair of polynomials modulo processed polynomials. Then, the initials and the separatants of the new polynomials are added to the set of inequations. Finally, to observe any solution, the current system is split by the following lemma which is a corollary of the theorem of zeros:

Lemma 3.4. If $A = 0$, $S \neq 0$ is a differential system and $p$ is a differential polynomial then,

$$\sqrt{|A| : S^\infty} = \sqrt{|A| : p} : S^\infty \cap \sqrt{|A| : (S \cup \{p\})^\infty}.$$  

This process continues until all systems are regular. Eventually, by the following Rosenfeld’s lemma (Rosenfeld, 1959), the consistent regular systems are detected and generated.

Theorem 3.5. (Rosenfeld’s lemma) If $A = 0$, $S \neq 0$ is a regular differential system of a differential polynomial ring $\mathcal{R}$ for a ranking $<$ then, every differential polynomial in $|A| : S^\infty$ that is partially reduced with respect to $A$, belongs to $(A) : S^\infty$.

By Rosenfeld’s lemma, $1 \in |A| : S^\infty$ if and only if $1 \in (A) : S^\infty$. It concludes that we can verify whether the regular differential system of equations $A = 0$ and inequations $S \neq 0$ is inconsistent, by verifying whether the reduced Gröbner basis of algebraic ideal $(A) : S^\infty$ equals to $\{1\}$.

The complete sub-algorithm of this algorithm is based on two criteria which detect unnecessary pairs. These criteria are the analogues of the first and second Buchberger criteria (Buchberger, 1979) established in computing Gröbner bases for ideals.

Proposition 3.6. Let $p$ and $q$ be two differential polynomials in the one differential indeterminate $u$ such that $ld(p) = \partial u$ and $ld(q) = \phi u$. If they are linear, homogeneous, with constant coefficients, and $\text{lcd}(\partial u, \phi u) = \psi u$, then $\Delta(p, q)$ will be reduced to zero with respect to $\{p, q\}$.

Proposition 3.7. Let $A = 0$ and $S \neq 0$ be a differential system and $[p_1, p_2, p_3]$ be the triple of differential polynomials such that the separatants of $p_i$’s belong to $S$ and they satisfy in conditions C: (i) the leaders $ld(p_i) = \partial u_i$ $(1 \leq i \leq 3)$ have common derivatives and are pairwise different, (ii) $\text{lcd}(\partial u_i, \partial u_j)$ is a derivative of $\partial u_i$ and (iii) one of the following conditions holds:

1. $\partial u$ is not a derivative of $\partial u$ (i) $\neq j$
2. $p_1 < p_2 < p_3$ or $p_3 < p_2 < p_1$
3. $p_2 < p_1 < p_3$ and $\text{degree}(p_1, \partial u) = 1$
4. $p_1 < p_2 < p_3$ and $\text{degree}(p_2, \partial u) = 1$
If the critical pairs \( \{p_1, p_2\} \) and \( \{p_2, p_3\} \) are solved by \( A = 0 \) and \( S \neq 0 \), then the critical pair \( \{p_1, p_3\} \) is also solved by this system.

Now, we present the general form of the algorithm. In each quadruple \( (A, D, P, S) \) the sets \( A, P, S \) and \( D \) contain the equations already processed, the non processed equations, the inequations and the possible critical pairs of the processed equations that have to be solved.

**Algorithm 3.1. The Rosenfeld-Gröbner algorithm**

| Input: | The system \( P_0 = 0 \) and \( S \neq 0 \) and a ranking \( < \). |
| Output: | The regular systems \( A_i = 0 \) and \( S_i \neq 0 \) s.t. \( \sqrt[\infty]{P_0 : S_0} = \sqrt[\infty]{A_1 : S_1} \cap \ldots \cap \sqrt[\infty]{A_n : S_n} \). |

**Begin**

Set \( \text{ToDo} := \{ (\emptyset, \emptyset, P_0, S_0) \} \)

Done := \( \emptyset \)

while \( \text{ToDo} \neq \emptyset \) do

Select and remove a quadruple \( (A, D, P, S) \) from \( \text{ToDo} \)

if \( D = P = \emptyset \) then

Partially reduce the elements of \( A \) pairwise

if the rank of \( A \) was not modified then

Partially reduce the elements of \( S \) w.r.t. \( A \)

if the reduced Gröbner basis of \( (A) : S_0 \) is not \( \{1\} \) then

Done := Done \( \cup \{ A = 0, S \neq 0 \} \)

end if

end if

else

if \( P \neq \emptyset \) then

Select and remove \( q_0 \) from \( P \)

else

Select and remove a critical pair \( \{p_0, p'_0\} \) from \( D \)

\( q_0 := \Delta(p_0, p'_0) \)

end if

\( q := q_0 \), fullrem \( A \)

if \( q = 0 \) then

\( \text{ToDo} := \text{ToDo} \cup \{ (A, D, P, S) \} \)

else

\( q_i := q - iv^2 (v^2 \text{ and } i \text{ are the initial and the rank of } q, \text{ respectively}) \)

\( q_s := dq - vs \) (\( s \) is the separant of \( q \))

\( \text{ToDo} := \text{ToDo} \cup \{ (A, D, P \cup \{ i, q_i \}, S) \} \)

\( \text{ToDo} := \text{ToDo} \cup \{ (A, D, P \cup \{ s, q_s \}, S \cup \{ i \}) \} \)

\( \text{ToDo} := \text{ToDo} \cup \text{Complete} ( (A, D, P, S), q) \)

end if

end if

end do

Return Done

**End**
Algorithm 3.2. The Complete sub-algorithm

**Input:** A quadruple \((A, D^\prime, P^\prime, S)\) and a reduced polynomial \(q\) w.r.t \(A\)

**Output:** The refinement of \((A, D^\prime, P^\prime, S)\) with regard to \(q\) based on two propositions 3.6 and 3.7.

**Begin**

Set \(A^\prime := \{q\} \cup \{p \in A \mid Idp \text{ is not a derivative of } ldq\}\)

Obtain

\[D_1 \subseteq D_0\] as follows:

- a critical pair \((p, q)\) in \(D_0\) is not kept in \(D_1\) only if:
  - \(p\) and \(q\) satisfy in the hypotheses of the Proposition 3.6 or
  - there exists a critical pair \((q, p')\) in \(D_1\) such that the triple \([q, p', p]\) satisfies in the conditions \(C\) of the Proposition 3.7.

\[D_2 \subseteq D^\prime\] as follows:

- a critical pair \((p, q)\) in \(D_0\) is not kept in \(D_1\) only if:
  - there exist a critical pair \((p, p')\) in \(D_1\) such that the triple \([p, q, p']\) satisfies in the conditions \(C\) of the Proposition 3.7 and \(\text{lcd}(Idp, Idp')\) is different from both \(\text{lcd}(ldp, ldq)\) and \(\text{lcd}(ldp', ldq)\).

\[D^\prime := D_1 \cup D_2\]

\[P^\prime := p^\prime\]

\[S^\prime := S \cup \{i_q, s_q\}\]

**Return** \((A^\prime, D^\prime, P^\prime, S^\prime)\)

**End**

4. The statement of the main results

In this section, we deal with a differential system in which the coefficients depend on some parameters. We define the parametric differential ring as \(R = K(\alpha_1, \ldots, \alpha_t|u_1, \ldots, u_n)\). In this ring, the coefficients of the differential polynomials are in \(K(\alpha_1, \ldots, \alpha_t)\). Here, we present a parametric-Rosenfeld-Gröbner algorithm that its input is a system of equations and inequations in the parametric differential ring \(R\). This algorithm outputs all distinct representations for all possible values of the parameters. These representations are recognised by the pair \((N, W)\) which contains equations and inequations in \(K(\alpha_1, \ldots, \alpha_t)\). To be comparable, this pair is presented as Quasi-canonical which is defined as follows (Montes, 2002):

**Definition 4.1.** The pair \((N, W)\) is said to be \(k\)-quasi-canonical if it satisfies:

- \(N\) is the reduced Gröbner basis in \(\text{Quot}(K)(\alpha_1, \ldots, \alpha_t)\) where \(\text{Quot}(K)\) is the quotient field of \(K\), with respect to the ordering \(\prec\).  
- The inequations in \(W\) are algebraically reduced modulo \(N\) and are irreducible over \(k(\alpha_1, \ldots, \alpha_t)\), where \(k\) is some intermediate field between \(\text{Quot}(K)\) and the algebraic closure \(K\). They are normalised in a canonical form in order to be recognised when compared.
- \(\prod_{w \in W} w \notin \sqrt{N}\).
- The polynomials in \(N\) are square-free.
- If some polynomial in \(N\) factors, no factor of it belongs to \(W\).

If \(\prod_{w \in W} w \in \sqrt{N}\), then we say that this pair is inconsistent.
It is necessary to mention the terms zero and non-zero differential conditions of a polynomial \( p \) are the initial, separat or the factors of \( p \) that can be considered zero and non-zero, respectively. Each of these conditions is in \( K[\alpha_1, ..., \alpha_i] \), it is called a parametric condition. Also, the terms new zero and new non-zero conditions are those which have not been considered yet. The Parametric-Rosenfeld-Gröbner algorithm has a tree structure that its root vertex corresponds to the sextuplet \( \langle \emptyset, \emptyset, P_0, \emptyset, \emptyset, \emptyset \rangle \), where \( P_0 = 0 \) and \( S_0 \neq 0 \) is the inputted system. This vertex is branched by the sub-algorithm Make-Tree I regarding zero and non-zero parametric and differential conditions of the polynomials of \( P_0 \), which are obtained by the sub-algorithms Find-New-DiffCondition and Find-New-ParCondition, and some new vertices are created. Each vertex \( v \) corresponds to a sextuplet \( \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \), where \( A_v \) is the set of all processed polynomials, \( S_v \) is the union of the set of inequations \( S_0 \) with the non-zero differential conditions of the polynomials of \( A_v \), \( D_v \) is set of all critical pairs of \( A_v \) that should be solved by the system \( A_v = 0 \) and \( S_v \neq 0 \), \( P_v \) contains all polynomials that are not processed already, and \( N_v \) and \( W_v \) are respectively the zero and non-zero parametric conditions and are presented as \( k \)-quasi-canonical. The branching process continues until at a vertex \( v \) either the sub-algorithms Parametric-Consistency and Differential-Consistency recognise that the system \( A_v = 0 \) and \( S_v \neq 0 \) is inconsistent (in this case, the vertex \( v \) is called an inconsistent vertex), or both sets \( D_v \) and \( P_v \) are empty (in this case, the system \( A_v = 0 \) and \( S_v \neq 0 \) is regular and the vertex \( v \) is called a terminal vertex). Finally, the consistent regular systems are detected using Rosenfeld’s lemma that was stated in the previous section and outputted. Each set of outputted regular systems that contains a given parametric conditions makes a regular representation corresponding to these parametric conditions. The general structure of the algorithm is as follows.

**Algorithm 4.1. The Parametric-Rosenfeld-Gröbner algorithm**

\[\begin{align*}
\textbf{Input:} & \quad \text{A differential system } P_0 = 0, \ S_0 \neq 0 \text{ in } K[\alpha_1, ..., \alpha_i]\{u_1, ..., u_n\}, \text{ a ranking } \prec \text{ and a total ordering } \preceq \\
\textbf{Output:} & \quad \text{All distinct representations of } \sqrt{P_0} : S_0^\infty \text{ for all possible value of parameters } \alpha_1, ..., \alpha_i \\
\textbf{Begin} & \\
\text{Set root vertex: } & \langle \emptyset, \emptyset, P_0, \emptyset, \emptyset, \emptyset \rangle \\
\text{Make-Tree I(} & \langle \emptyset, \emptyset, P_0, \emptyset, \emptyset, \emptyset \rangle \text{) (It starts to the branching recursively.)} \\
\text{At each vertex } & v, \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle, \text{ is stored in the variables table.} \\
\quad & \text{if } \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \text{ is a terminal vertex then} \\
\quad & \quad \text{if the reduced } \text{Gröbner basis of } \langle A_v \rangle : S_v^\infty \text{ does not equal to } \{1\} \text{ then} \\
\quad & \quad \quad \text{return } \langle [A_v, S_v], [N_v, W_v] \rangle \\
\quad & \quad \text{end if} \\
\quad & \text{end if} \\
\text{End} & \\
\end{align*}\]

We now present the general structure of the recursive sub-algorithm Make-Tree I. The input of this sub-algorithm is a vertex \( v \) corresponding to the sextuplet \( \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \). This algorithm creates some new branches when the sub-algorithm Find-New-ParCondition finds any new zero or non-zero parametric conditions. If there is not any parametric condition, then the sub-algorithm Make-Tree II is called for branching the vertex. Make-Tree II branches a vertex according to differential conditions of given polynomial. When any new vertex is created by these algorithms, then the algorithm Make-Tree I is called again and some other new branches are created. This process continues until the created vertex \( v \) is a terminal or inconsistent vertex.
Algorithm 4.2. The Make-Tree I sub-algorithm

Input: A vertex $v$ corresponding to the sextuplet $(A_v, D_v, P_v, S_v, N_v, W_v)$.

Output: Some new vertices.

Begin

if $v$ is a inconsistent vertex or a terminal vertex then
    STOP
else
    if $D_v \neq \emptyset$ then
        Select and remove a critical pair $\{p_1, p_2\}$ from $D_v$ and $q := \Delta(p_1, p_2) \text{algrem} N_v$
    else
        Select and remove a polynomial $p$ from $P_v$ and $q := p \text{algrem} N_v$
    end if
    $q := q \text{fullrem} A_v$
    if $q = 0$ then
        Make-Tree I $(A_v, D_v, P_v, S_v, N_v, W_v)$
    else
        Find-New-ParCondition $(q, N_v, W_v) = (q', N'_v, W'_v, \text{newcond})$ (It contains the refinement of $q, N_v$ and $W_v$ with the new zero and non-zero parametric conditions)
        if there is no new parametric condition then
            Make-Tree II $(v, q')$
        else
            for each zero parametric condition $zc$ do
                Set $A_u := \{p \in A_v \mid p \text{ is algebraically reduced w.r.t} zc\}$
                $P_u := P_v \cup (A_v \setminus A_u) \cup \{q' \text{algrem} zc\}$
                the new vertex $u := (A_u, D_v, P_u, S_v, N'_v \cup \{zc\}, W'_v)$
                Make-Tree I $(A_u, D_u, P_u, N_u, W_u)$
            end do
            if the set of non-zero conditions NZC is not empty then
                Set the new vertex $u := (A_v, D_v, P_v, S_v, N'_v \cup \text{NZC})$
                Make-Tree II $(A_u, D_u, P_u, N_u, W_u), q')$
            end if
        end if
    end if

End

This algorithm creates some new vertices according to zero and non-zero conditions. The first step is processing the parametric conditions of the given polynomial. The zero conditions are added one by one to the set of zero parametric conditions $N_v$. Then, all processed polynomials $A_v$ that are not algebraically reduced with respect to the new zero condition are moved to the set of unprocessed polynomials $P_v$. In this way, the new vertices are created and the algorithm Make-Tree I is called again for branching them. If the non-zero conditions set is not empty, then another new vertex is created by adding these conditions to the set $W_v$ and the sub-algorithm Make-Tree II is called to process the differential conditions of the polynomial and branch the vertex.

The inputs of Make-Tree II are a vertex and a polynomial $q$. The general form of this algorithm is as follows:
Algorithm 4.3. The Make-Tree II sub-algorithm

Input: A vertex v and a polynomial q.
Output: Some new vertices.
Begin

Find-New-DiffCondition(q, S_v) (It contains two set of zero and non-zero conditions.)
if there is no new differential condition then
    Make-Tree I(Update(v, q, ∅))
else if the set of non-zero conditions is empty then
    for each zero condition z do
        q' := subs(z = 0, q)
        Set the new vertex u : (A_v, D_v, P_v, {z, q'}, S_v, N_v, W_v)
        Make-Tree I((A_u, D_u, P_u, S_u, N_u, W_u))
    end do
else
    Make-Tree I(Update(v, q, non zero conditions))
Let {z_1, ..., z_t} be the set of zero conditions
for i = t down to 1 do
    q' := subs(z = 0, q)
    Set the new vertex u : (A_v, D_v, P_v, {z_i, q'}, S_v, {z_1, ..., z_{i-1}}, N_v, W_v)
    Make-Tree I((A_u, D_u, P_u, S_u, N_u, W_u))
end do
end if
End

Generally, in Make-Tree II there are two possibilities to create a new vertex:

- When the polynomial q has some zero conditions, these conditions are imposed on the polynomial q. The result is a new polynomial with the zero conditions that are added to the set of non-processed polynomials P_v and non-zero conditions that are added to the set of inequations S_v.
- When the polynomial q does not have any zero condition, the non-zero conditions (if there is any) are added to the set of inequations S_v, the polynomial q is added to the set of processed polynomial A_v and former members of the set A_v, and the set D_v is updated. This process is performed in the sub-algorithm update, which will be described later.

The parametric and differential conditions of a polynomial are obtained by the sub-algorithms Find-New-ParCondition and Find-New-DiffCondition, respectively. In the following, we explain these algorithms. The inputs of Find-New-ParCondition are a polynomial p with two sets of the zero parametric conditions N and the non-zero parametric conditions W. This algorithm checks the initial of the polynomial p according to old zero and non-zero conditions and returns new conditions, which can be considered zero and non-zero. Before presenting this algorithm, it should be noted that the set of irreducible polynomials that are factors of the polynomials of W is denoted by PriFact(W), which is normalised in a canonical. Also, for each polynomial in K[α_1, ..., α_t]{u_1, ..., u_n}, the notation pcoef(p) denotes the product of the factors of p, which are in K[α_1, ..., α_t].
Algorithm 4.4. The Find-New-ParCondition sub-algorithm

| Input: A polynomial p and two sets of zero and non-zero conditions N and W. |
| Output: The refinement of p, N and W and also, zero and non-zero conditions of p. |
| Begin |
| set \( p' := p; \) sw := true; \( N' := N; \) w' := W; |
| while sw = true do |
| if \( pcoef(p') \in \sqrt{N} \) then |
| \( p' := p' - v'L'; \) \( v' \) and \( v' \) are the initial and the rank of \( p' \), respectively \( N' := \) Gröbner basis for \( N' \cup \{l'\} \) w.r.t \( \prec \) \( W' := \{w\text{algrem}_{\mathbb{N}}\} w \in W' \} \) |
| else sw := false |
| end if |
| end do |
| \( p' := p'\text{algrem}_{\mathbb{N}}N' \) |
| cond := PriFact(pcoef(l)) \ minus W' |
| if \( p' \notin K[\alpha] \) then |
| return cond (as the zero and non-zero conditions) |
| else |
| return cond (just as the zero condition) |
| end if |
| End |

This algorithm checks the parametric coefficient of the initial of the polynomial \( p \), \( pcoef(l) \). When \( pcoef(l) \) is zero, it begins omitting iteratively the leading term of \( p \). In the following, the parametric coefficient \( pcoef(l) \), which is in \( \sqrt{N} \), is added to \( N \) in order to improve \( N \) and the new Gröbner basis is recomputed. Then, \( W \) is also improved. When \( pcoef(l) \) is not decidable regarding the set of zero and non-zero conditions \( N \) and \( W \), then \( p \) is algebraically reduced with respect to \( N \). The algorithm now computes the set \( cond \) containing the irreducible factors of \( pcoef(l) \) that are not in \( W \) and returns it. If this is empty, it means that \( pcoef(l) \) is not zero and \( p \) does not have any new parametric condition. Otherwise, if \( p \) is not in \( K[\alpha] \), so the set \( cond \) is as zero and non-zero parametric conditions of \( p \). Otherwise, the set \( cond \) is as just zero parametric conditions of \( p \), because the inputted polynomial of this algorithm comes from the set of equations and cannot be non-zero.

Now, we present the sub-algorithm Find-New-DiffCondition that computes the zero and non-zero differential conditions of a polynomial. The inputs of this algorithm are a differential polynomial \( p \) and the set of inequations \( Sv \). This sub-algorithm verifies the polynomial \( p \) and its initial and separant and finds the new zero and non-zero conditions regarding the set of inequations \( S_v \). The following proposition which works as an efficient criterion in this algorithm helps to remove some ineffectual conditions.

**Proposition 4.2.** Let \( p \in \mathcal{R} = K(U) \) be a differential polynomial with the non-zero initial \( l \) and the separant \( s \). Let also \( S \) be a set of differential inequations. If there is a member \( h \in S \) such that it is reduced to zero with respect to \( p \), then \( p \) cannot be zero.

**Proof.** Suppose not. Since for some \( h \in S \) we have \( h \) fullrem \( p = 0 \) (the initial and the separant of \( p \) are non-zero and the reduction by \( p \) is possible), then for some integers \( a \), we have \( p^{a_1} s^{a_2} = 0 \mod(|p|) \), suggesting that for a non-zero polynomial \( b \in \mathcal{R} \) and a derivative \( \theta \in \Theta \), \( \Theta^{a_1} s^{a_2} h = b \). Because \( p \) is assumed to be zero, \( h \) should be zero as well, which contradicts the assumption.
Algorithm 4.5. The Find-New-DiffCondition sub-algorithm

Input: A polynomial \( p \) and the set of inequations \( S \).
Output: The zero and non-zero conditions regarding \( p \).

Begin

set zerocond, non – zerocond := ∅

if \( p = u_1 \cdot \cdots \cdot u_t \) is a product of distinct factors \( \{u_1, \ldots, u_t\} \) then

for \( i = 1 \) to \( t \) do

if \( i_u_i \in S \) or for each \( h \in S \), \( h \ fullrem \{u_i\} \neq 0 \) then

Add \( c \) to zerocond

end if

end do

else

for \( c \in \{l_p, s_p\} \backslash K \) do \( (l_p \) and \( s_p \) are the initial and the separant of \( p \))

if \( l_c, s_c \in S \) or for each \( h \in S \), \( h \ fullrem \{c\} \neq 0 \) then

Add \( c \) to zerocond and non – zerocond

end if

end do

end if

Return \( (\text{zerocond, non} – \text{zerocond}) \)
End

In the Find-New-DiffCondition algorithm if the inputted polynomial \( p \) is a product of distinct non-constant factors, then each of these factors may be a zero condition. However, some of these factors that satisfy the hypothesis of Proposition 4.2 cannot be zero and they are removed from zero conditions. It should be noted that in this algorithm we use two obvious ways to find the factors of the inputted polynomial \( p \). When \( p \) is a product of some polynomials, we consider these polynomials as its factors. For example, let the inputted polynomial be \( p = ayux, y, \) such that \( a, u_y \) and \( u_x \) are three factors of \( p \). Also, when the greatest common divisor (gcd) of all terms of \( p \) does not belong to \( K \), we consider it as a factor of \( p \). For example, if the inputted polynomial is \( p = u^4 + u \), so \( u \) is the gcd of the two terms of \( p \), suggesting that \( u \) and \( u^3 + 1 \) are two factors of \( p \).

Otherwise, the initial and the separant of \( p \), provided not being in \( K \), should be considered zero and non-zero. However, each of them that satisfies in the hypothesis of Proposition 4.2 will be removed from zero conditions. When a polynomial \( q \) satisfies in Proposition 4.2, it means that there is a polynomial \( h \) in the set of inequations \( S_v \) such that it forces the polynomial \( q \) not be zero. In fact, this proposition plays an important role in the Parametric-Rosenfeld-Gröbner algorithm. As mentioned, this proposition decreases the number of zero conditions leads to a decrease in the number of ineffectual branches that will be made in the sub-algorithm Make-Tree II.

Now, we explain the general structure of the sub-algorithm Update. The input of this algorithm is a vertex \( u \) corresponding to the sextuplet \( \langle A_u, D_u, P_u, S_u, N_u, W_u \rangle \), a polynomial \( q \) and non-zero conditions. This algorithm adds the new non-zero conditions to the set of inequations \( S_u \) and the new polynomial \( q \) to the set of processed polynomials \( A_u \). Then based on two analogues of Buchberger’s criteria, it starts to make possible critical pairs between \( q \) and the other members of \( A_u \) and adds them to the set of critical pairs \( D_u \). In this way, a new vertex that corresponds to the sextuplet \( \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \) is created.
Algorithm 4.6. The Update sub-algorithm

Input: A vertex $u$, a polynomial $q$ and its non-zero conditions.
Output: A new vertex $v$.

Begin
Set $A_v := \{q\} \cup \{p \in A_u \mid p \text{ is reduced w.r.t. } q\}$
\[ P_v := P_u \cup \{p \mid p \in A_u \text{ s.th } \text{lcd}(ldq, ldp) \neq ldq \text{ and } p \text{ is not reduced w.r.t. } q\} \]
\[ D_0 := \{\text{possible critical pairs } (p, q) \mid p \in A_u \text{ s.th } p \text{ is reduced w.r.t. } q \text{ or } \text{lcd}(ldq, ldp) = ldq\} \]

Obtain
\[ D_1 \subseteq D_0 \text{ as follows:} \]
\[ \text{a critical pair } (p, q) \in D_0 \text{ is not kept in } D_1 \text{ only if}: \]
\[ \bullet \text{ } p \text{ and } q \text{ satisfy in the hypotheses of the Proposition 3.6 or} \]
\[ \bullet \text{ there exist a critical pair } (q, p') \text{ in } D_1 \text{ such that} \]
\[ \text{the triple } (q, p', p) \text{ satisfies the conditions C of the Proposition 3.7.} \]
\[ D_2 \subseteq D_0 \text{ as follows:} \]
\[ \text{a critical pair } (p, q) \in D_0 \text{ is not kept in } D_1 \text{ only if}: \]
\[ \bullet \text{ there exist a critical pair } (p, p') \text{ in } D_u \text{ such that the triple} \]
\[ (p, q, p') \text{ satisfies in the conditions C of the Proposition 3.7 and } \text{lcd}(ldp, ldp') \]
\[ \text{is different from both } \text{lcd}(ldp, ldq) \text{ and } \text{lcd}(ldp', ldq). \]
\[ D_v := D_1 \cup D_2 \]
\[ S_v := S_u \cup \{lq, s_q\} \]
Return $(A_v, D_v, P_v, S_v, N_v, W_v)$
End

If a polynomial $p$ in the set $A_v$ is not reduced with respect to the new polynomial $q$, then it exits from the set $A_v$ for reprocessing. Now, if $ldp$ is not reduced with respect to $ldq$, then the critical pair $(p, q)$ is added to the set $D_0$ to recognise more solved critical pairs by the analogues of the Buchberger criteria; otherwise, it is added to the set $P_u$. The rest of the algorithm is based on two propositions 3.6 and 3.7.

Remark 1. This sub-algorithm is similar to the Complete sub-algorithm. For further information about this sub-algorithm, see the part 5.2 of the article (Boulier et al., 2009).

Finally, we present the sub-algorithms Parametric-Consistency and Differential-Consistency, which check the termination conditions in the Make-Tree I algorithm. Parametric-Consistency checks the sets of zero and non-zero parametric conditions $N_v$ and $W_v$ of a vertex $v$ and returns FALSE if the pair $(N_v, W_v)$ is inconsistent. Otherwise, it returns a $k$-quasi-canonical representation of the pair $(N_v, W_v)$ according the definition 4.1.

Algorithm 4.7. The Parametric-Consistency sub-algorithm

Input: The sets of zero and non-zero conditions $N, W$ and an ordering $<_q$.
Output: FALSE, if the pair $(N, W)$ is inconsistent and the $k$-quasi-canonical representation of $(N, W)$, otherwise.

Begin
Set $W' := \text{PriFact}(\{q \text{ algrem }<_q N \mid q \in W\})$
\[ sw := \text{true}; \quad N' := N; \]
if $\prod_{q \in W} q \in \sqrt{(N)}$ then
\[ \text{return FALSE} \]
End
else
    While \( sw = \text{true} \) do
        \( sw := \text{false}; \)
        \( N'' := \text{Drop any factor of a } p \in N' \text{ that belongs to } W', \text{ as well as multiple factors.} \)
        If \( N'' \neq N' \) then
            \( sw := \text{true} \)
            \( N' := \text{Gröbner basis for } N'' \text{ w.r.t } \prec_\alpha \)
            \( W' := \text{PriFact}(\{q \text{ algrem } A | q \in W') \}
        end if
    end do
    \( \text{return } N' \text{ and } W' \)
end if
End

For more details and the proof of correctness of this algorithm, the reader can refer to (Montes, 2002).

Now, we explain another sub-algorithm for checking the consistency of the vertices, which is called Differential-Consistency sub-algorithm. For each vertex \( v \) corresponding to the sextuplet \( \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \), if the output of the Parametric-Consistency is not \( \text{FALSE} \), then this vertex will be refined by the \( k \)-quasi-canonical representation of \( N_v \) and \( W_v \). After that, Differential-Consistency sub-algorithm checks the consistency of this vertex regarding the differential conditions. According to this algorithm, this vertex is inconsistent if either a non-zero element of \( K \) appears among the equations \( A_v \), or 0 appears among the inequations \( S_v \), or the system \( A_v = 0 \) and \( S_v \neq 0 \) satisfies in the following proposition. As mentioned in subsection 5.5.2 of (Boulier et al., 2009), using this proposition is not very CPU expensive and can point out inconsistencies.

Otherwise, it refines the vertex by reducing the set of inequations \( S_v \) with respect to the set of processed equations \( A_v \).

**Proposition 4.3.** Let \( A = 0 \) and \( S \neq 0 \) be a system of equations and inequations such that the initials and the separants of the members of \( A \) belong to \( S'\). If there is a polynomial \( h \in S \) such that \( h \) is reduced to 0 with respect to \( A \), then the system \( A = 0 \) and \( S \neq 0 \) is inconsistent.

**Proof.** Let \( A = \{p_1, ..., p_t\} \) and for some \( h \in S \) we have \( h \text{fullrem} A = 0 \). Then, for some integers \( a_i \) and \( b_i \) we have \( p_1^{a_1} \cdots p_t^{a_t} s_1^{b_1} \cdots s_t^{b_t} h = 0 \text{ (mod } A) \). Thus, for some non-zero polynomials \( q_i \) and derivatives \( \theta_i \in \Theta \) we can write:

\[
p_1^{a_1} \cdots p_t^{a_t} s_1^{b_1} \cdots s_t^{b_t} h = q_1 \theta_1 p_1 + \cdots + q_t \theta_t p_t.
\]

Since for each \( i \), \( \theta_i p_i = 0 \) and the coefficient of \( h \) is non-zero, \( h \) should be zero and so the system \( A = 0 \) and \( S \neq 0 \) is inconsistent.

**Algorithm 4.8 The Differential-Consistency sub-algorithm**

**Input:** A vertex \( v \) corresponding to a sextuplet \( \langle A_v, D_v, P_v, S_v, N_v, W_v \rangle \).

**Output:** \( \text{FALSE} \), if the system \( A_v = 0 \) and \( S_v \neq 0 \) is inconsistent and the refinement of \( v \), otherwise.

**Begin**
if \( A_v \cap K \neq \emptyset \) then
Return FALSE
else
  if $0 \in \{ r | r = q \text{rem} A_v, q \in S_v \}$ then
    Return FALSE
  else
    Return $\langle A_v, D_v, P_v, \{ r | r = q \text{fullrem} A_v, q \in S_v \} \setminus K, N_v, W_v \rangle$ (Refinement of data)
  end if
end if
End

Now, we prove the correctness of the Parametric-Rosenfeld-Gröbner algorithm, but before that we need the following lemma.

Lemma 4.4. If $A = 0$, $S\neq0$ is a differential system and $p_1p_2$ is a product of two differential polynomials then:

$$\sqrt{[A, p_1p_2]} : S^\infty = \sqrt{[A, p_1]} : S^\infty \cap \sqrt{[A, p_2]} : S^\infty$$

Proof. We begin by showing that $\sqrt{[A, p_1p_2]} : S^\infty \subseteq \sqrt{[A, p_1]} : S^\infty \cap \sqrt{[A, p_2]} : S^\infty$. Let $p$ be a differential polynomial in $\sqrt{[A, p_1p_2]} : S^\infty$. According to the theorem of zeros, it suffices to show that $p$ vanishes on every solution of the systems of $A = 0, p_1 = 0, S\neq0$ and $A = 0, p_2 = 0, S\neq0$. Now, let $\alpha$ be a solution of these systems, then it also will be a solution of the system $A = 0, p_1p_2 = 0, S\neq0$. Therefore, $p$ vanishes on $\alpha$ and it will belong to $\sqrt{[A, p_1]} : S^\infty \cap \sqrt{[A, p_2]} : S^\infty$. Conversely, let $p$ be in $\sqrt{[A, p_1]} : S^\infty \cap \sqrt{[A, p_2]} : S^\infty$ and $\alpha$ be a solution of the system $A = 0, p_1p_2 = 0, S\neq0$. We only need to show that $p$ vanishes on $\alpha$. Since $p_1p_2$ vanishes on $\alpha$, so $p_1$ or $p_2$ vanishes on $\alpha$. If $p_1$ is so, then $\alpha$ will be a solution of the system $A = 0, p_1 = 0, S\neq0$ and $p$ vanishes on $\alpha$ (since $p \in \sqrt{[A, p_1]} : S^\infty$). Similarly, if $p_2$ vanishes on $\alpha$, then $p$ is so. Therefore $p$ belongs to $\sqrt{[A, p_1p_2]} : S^\infty$ and the proof is completed.

Theorem 4.5. Consider the parametric system $A = 0$ and $S\neq0$ in the parametric differential ring $K[v_1, \ldots, v_n, u_1, \ldots, u_m]$, the term ordering $\prec_\alpha$ and the ranking $\prec$. The Parametric-Rosenfeld-Gröbner algorithm decomposes the $A = 0, S\neq0$ with a tree structure that at each terminal vertex $v$ the pair $(N_v, W_v)$ of parametric conditions is $k$-quasi-canonical for some field $k$, the system $A_v = 0$ and $S_v\neq0$ is regular, and the systems corresponding to terminal vertices $v_1, \ldots, v_t$ with the same parametric conditions satisfy in:

$$\sqrt{A} : S^\infty = [A_{v_1}] : S_{v_1}^\infty \cap \cdots \cap [A_{v_t}] : S_{v_t}^\infty$$

for these conditions.

Proof. According to Parametric-Consistency, the pair $(N_v, W_v)$ at each terminal vertex $v$ is $k$-quasi-canonical. Now, we prove that the systems corresponding to terminal vertices are regular. At each terminal vertex $v$, two sets $D_v$ and $P_v$ are empty, which means that all possible critical pairs between the members of $A_v$ are solved with the system $A_v = 0$ and $S_v\neq0$. Also, as $D_v$ is the set of critical pairs that should be solved and each member of $A_v$ is reduced with respect to others. So, the set $A_v$ is autoreduced and coherent. Furthermore, according to the Update sub-algorithm, when a new polynomial is added to $A_v$, the initial and separant of this is added to $S_v$ such $S_v$ contains all initials and separatants of the members of $A_v$, and the system $A_v = 0$ and $S_v\neq0$ is regular. Now, it is the time to show the regular ideals corresponding with terminal vertices with the same conditions of the pair $(N_v, W_v)$ satisfy in intended intersection according to this condition. As already noted, the sub-algorithm Make-Tree I firstly branches the vertex $v$ with respect to zero and non-zero parametric conditions of a polynomial $p$, which are computed by the Find-New-ParCondition.
sub-algorithm, and makes new vertices with different parametric conditions. If this polynomial does not have any parametric condition, the Make-Tree II begins to branch the vertex by differential conditions of the polynomial and the parametric conditions of these new vertices are equal. So, it is sufficient that we show when a vertex \( v \) is branched by differential conditions of a polynomial, the radical ideal \( \sqrt{A_v} : S_v \) equals to the intersection of radicals of saturation ideals corresponding to the created vertices. If the polynomial \( p \) is as a product of some polynomials, then Find-New-DiffCondition returns some zero conditions and Make-Tree II makes some new vertices by adding these conditions to the set of equations. According to the previous lemma, the radical of the saturation ideal of the system corresponding to the vertex \( v \) is equal to the intersections of radicals of saturation ideals corresponding to new vertices. Otherwise, Find-New-DiffCondition returns some conditions that should be considered zero and non-zero. In addition, according to Lemma 4.4, the radical of the saturation ideal of the system corresponding to the vertex \( v \) is also equal to the intersections of radicals of saturation ideals corresponding to the new vertices. So, the radical of the corresponding saturation ideal with the root vertex for each parametric condition is as the intersections of radicals of saturation ideals corresponding to terminal vertices with the same parametric conditions. Finally, since these ideals are regular and regular ideals are radical (Lazard’s lemma, see Boulier et al., 1995), the claim of the theorem is proved.

5. Example

In this section, we present the output of the Parametric-Rosenfeld-Gröbner algorithm for several examples in the differential ring \( \mathcal{R} = \mathbb{Q}[a, b, c, d(x, y, z)] \{ m, u, v, w \} \) considering the orderly ranking as \( w < v < u < m \) and elimination ranking as \( m < u < v < w \) for the graded lexical order \( z < y < x \). The implementation of this algorithm in Maple is available at the address http://faculty.du.ac.ir/rahmani/software. The output of this algorithm is in the form of a discussion about the distinct representations of given system as an intersection of regular ideals for different cases of values of parameters. As stated in section 5 of (Boulier, 2018), DifferentialAlgebra package of Maple permits to define parameters and makes such decompositions.

First consider the parametric differential system \( P_1 \), also known as Lorenz system (Harrington & Van Gorder, 2017), with the orderly ranking.

\[
P_1 = \begin{cases} 
  u_x - a(v - u) = 0 \\
  v_x - u(b - w) + v = 0 \\
  w_x - uv + cw = 0
\end{cases}
\]

Table 1 describes the discussion of decomposing of system \( P_1 \) for different values of the parameters \( a, b \) and \( c \). As can be seen, when the parameter \( a \) is considered non-zero, we have the following representation for the radical of ideal \( \sqrt{P_1} \) for each arbitrary values of \( b \) and \( c \):

\[
\sqrt{[P_1]} = [au - av + u_x, -abu + auw + au + au_x + u_x + u_{xx}, -abcu^2 + au^2 + acu^2 + auu_x + u^3u_x \\
+ auu_x - au^2 + cuu_x + cuu_{xx} + uu_x + uu_{xx} + uu_{x,xx} - u_x^2 - u_xu_{xx}] : \{u\}^\infty \cap \{u, v, cw + w_x\}
\]

It is notable that if we substitute \( a = 0 \) in this representation, then the first ideal will be inconsistent, suggesting that it is essential that this parameter is not zero. According to this table for the case \( a = 0 \) we have another representation as follows:

\[
\sqrt{P_1} = [ux, -bu + uw + v +vx, bcu - u^2v - cv - cv_x - vx - v_{xx}] : \{u\}^\infty \cap \{u, v + v_x, cw + w_x\},
\]

This representation is also independent of the values of \( b \) and \( c \).

Now, consider the Genesio-Tesi system (Harrington & Van Gorder, 2017) that contains three ODE equations and the orderly ranking.
Table 1. Discussion of the system $P_1$ (Lorenz system)

| Case            | Regular ideals                                                                 |
|-----------------|-------------------------------------------------------------------------------|
| $a \neq 0$      | $[au - av + u, -abu + auw + au + au + u, + u_{xx} - abc, + abu^2 + acu + u_{xy} + au_{xx} - au^2 + cu_{x} + cu_{ux} + u_{xx} + w_{xx} - u_{x}^2 - u_{xx}] : [u]^\infty,$ |
|                 | $[u, v, w + w_{x}]$                                                         |
| $a = 0$         | $[u_{y}, -bu + auw + v + v_{x}, bcu - a^2 + cu - cv_{x} - v_{x} - v_{xx}] :$  |
|                 | $[u]^\infty, [u, v + v_{x}, cw + w_{x}]$                                     |

Table 2. Discussion of system $P_3$

| Case                        | Regular ideals                                                                 |
|-----------------------------|-------------------------------------------------------------------------------|
| $a, b, c, a + 1 \neq 0$     | $[4acu_{x} - b^{2}v + 4cu_{y} - 2c, v_{y}, au_{xy} - 1], [au_{x}^{2} +$ $bu_{x} + v_{x}^{2} + cu_{y}, u_{xy}, v_{y}, au_{xy} - 1] : [2au_{x} + bv + 2bu_{x}]^\infty,$ $[u, au_{xy} - 1]$ |
| $a, c, a + 1 \neq 0, b = 0$ | $[u_{x}, au_{xy} - 1], [au_{x}^{2} + bu_{x} + v_{x}^{2} + cu_{y}, u_{xy}, v_{y}, au_{xy} - 1] : [2au_{x} + bv + 2bu_{x}]^\infty$ |
| $a, a + 1 \neq 0, b = c = 0$| $[u_{x}, u_{xy}, au_{xy} - 1], [v, u_{xy}, au_{xy} - 1]$                        |
| $a, a + 1, b \neq 0, c = 0$ | $[u_{x}, au_{x}, b + bu_{x}, au_{xy} - 1], [u_{x}, u_{xy}, au_{xy} - 1],$ $[v, u_{xy}, au_{xy} - 1]$ |
| $a = -1, b, c \neq 0$      | $[v, u_{y}, -u_{xy} - 1], [v, u_{y}, -u_{xy} - 1] : [v]^{\infty},$ $[bu_{x} + 2cu_{y}, u_{x}, v_{y}, -u_{xy} - 1] : [v]^{\infty}$ |
| $a = -1, c = 0, b \neq 0$  | $[v, u_{y}, -u_{xy} - 1], [u_{x}, u_{x}, -u_{xy} - 1]$                         |
| $a = -1, b = 0, c \neq 0$  | $[u_{x}, -u_{xy} - 1]$                                                       |
| $a = -1, b, c = 0$         | $[u_{x}, -u_{xy} - 1]$                                                       |

$P_2 = \begin{cases} 
 w - v_{x} = 0 \\
 v - u_{x} = 0 \\
 au + bv + cw + u^{3} - w_{x} = 0 
\end{cases}$

The output of the Parametric-Rosenfeld-Gröbner algorithm contains only one case meaning that for each value of the parameters $a$, $b$ and $c$ we have the same representation as follows:

$\sqrt{P_2} = [v - u_{x}, w - u_{x}, u^{2} + au + bu_{x} + cu_{x} - u_{x,x} - u_{x,x}]$

For arbitrary values of $a, b$ and $c$ the ideal of this representation is regular and consistent, so this representation can be used for all cases of the values of the parameters.

Finally, we discuss the following PDE system with the elimination ranking. Although this system has no physical significance, it has more distinct representations.

$P_3 = \begin{cases} 
 (a + 1)u_{x,x}^{2} + bu_{x} + cu_{x} = 0 \\
 u_{y} = 0 \\
 au_{y} - 1 = 0 
\end{cases}$

Table 2 presents the decomposing of this system. The table shows all distinct representations for all possible values of the parameters. It means that for not mentioned cases there is no regular representation for the radical of ideal $[P_3]$. For instance, in the case $a = 0$, the system $P_3$ is inconsistent so the output of Parametric-Rosenfeld-Gröbner algorithm does not contain this case.

6. Conclusion

We presented a new algorithm with branching style that examines the values of the parameters of the parametric differential system and returns all distinct regular representations for all possible
values of these parameters. The branches of this algorithm are made by differential or parametric conditions of equations. When a vertex is branched by parametric conditions only the information of this vertex is needed to follow the branches below it, and no other information about the upper or lateral vertices is needed. Accordingly, this algorithm can be implemented in parallel such that all distinct representations are made individually and simultaneously in a time-saving manner. We also applied a new criterion for reducing the ineffectual branches made by differential conditions that should be processed in this algorithm.

Funding
The authors received no direct funding for this research.

Author details
Shahnaz Fakouri1
E-mail: sh.fakouri@std.du.ac.ir
Sajjad Rahmany1
E-mail: rahman@du.ac.ir
Abdolali Basiri2
E-mail: basiri@du.ac.ir
1 Department of mathematics, Damghan University, Damghan, Iran.

Citation information
Cite this article as: A new algorithm for computing regular representations for radicals of parametric differential ideals, Shahnaz Fakouri, Sajjad Rahmany & Abdolali Basiri, Cogent Mathematics & Statistics (2018), 5: 1507131.

References
Becker, T., & Weispfenning, V. (1991). Gröbner bases: A computational approach to commutative algebra, Graduate Texts in Mathematics (Vol. 141). New York, NY: Springer.
[Boulier, F. (1994). Étude et Implantation de Quelques Algorithmes en Algèbre Différentielle (PhD thesis), Université Lille 1, 59655, Villeneuve d’Ascq, France.
Boulier, F. (2007). Differential elimination and biological modelling, radon series on computational and applied mathematics. Gröbner Bases in Symbolic Analysis, 111–139.
Boulier, F. (2016). The management of parameters in the MAPLE differentialalgebra package. working paper or preprint (never published). Retrieved from https://hal.archives-ouvertes.fr/hal-01825191.
Boulier, F., Lazard, D., Ollivier, F., & Petitot, M. (1995). Representation for the radical of a finitely generated differential ideal. In ISSAC’95, International Symposium on Symbolic and Algebraic Computation (pp. 158–166). Retrieved from http://hal.archives-ouvertes.fr/hal-00138020
Boulier, F., Lazard, D., Ollivier, F., & Petitot, M. (2009). Computing representations for radicals of finitely generated differential ideals. Applications Algebra Engrg Commission Computation, 20, 273–321.
Buchberger, B. (1979). A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In symbolic and algebraic computation, EUROCAM’79. Internat. Symposium Marseille Lecture Notes in Computer Science, Springer, Berlin, 72, 3–21.
Cox, D., Little, J., & O’Shea, D. (1992). Ideals, varieties and algorithms. an introduction to computational algebraic geometry and commutative algebra, undergraduate texts in mathematics. New York, NY: Springer.
Harrington, H. A., & Van Gorder, R. A. (2017). Reduction of dimension for nonlinear dynamical systems. Nonlinear Dynam, 88, 715–734. doi:10.1007/s11071-016-3272-5
Hubert, E. (2000). Factorization free decomposition algorithms in differential algebra. Journal Symbolic Computation, 29(4–5), 641–662.
Hydon, P. E. (2000). Symmetry methods for differential equations. Cambridge: Cambridge University Press.
Jauberthie, C., Trave-Massuyes, L., & Verdiere, N. (2016). set-membership identifiaility of nonlinear models and related parameter estimation properties. International Journal Applications Mathematical Computation Sciences, 26, 2083–8492. doi:10.1515/amcs-2016-0057
Kolchin, E. R. (1973). Differential algebra and algebraic groups. New York, NY: Academic Press.
Mansfield, E. L., & Clarkson, P. A. (1997). Applications of the differential algebra package difficulto2 to classical symmetries of differential equations. Journal Symbolic Computation, 23, 517–533. doi:10.1006/jsco.1996.0105
Montes, A. (2002). A new algorithm for discussing Gröbner bases with parameters. Journal Symbolic Computation, 33, 183–208. doi:10.1006/jsco.2001.0504
Ritt, J. F. (1950). Differential algebra. New York, NY: Dover.
Rosenfeld, A. (1959). Specializations in differential algebra. Transactions Amer Mathematical Soc, 90, 394–407. doi:10.1090/S0002-9947-1959-0107642-2
Sabzevari, M., Hashemi, A., Alizadeh, B. M., & Merker, J. (2016). Lie algebras of infinitesimal CR automorphisms of weighted homogeneous and homogeneous CR-Generic submanifolds of $\mathbb{C}^n$. Filomat, 30(6), 1387–1411.
Sit, W. Y. (2002). The Ritt-Kolchin theory for differential polynomials. In L. Guo, W. F. Keigher, P. J. Cassidy, & W. Y. Sit (Eds.), Differential algebra and related topics (pp. 1–70). World Scientific Publishing Co. Inc. Retrieved from https://doi.org/10.1142/9789812778437_0001
Umirbaev, U. (2016). Algorithmic problems for differential polynomial algebras. Journal Algebra, 455, 77–92. doi:10.1016/j.jalgebra.2016.02.010
