

# Simple Forms and Rational Solutions of Pseudo-Linear Systems

Moulay Barkatou, Thomas Cluzeau, Ali El Hajj

University of Limoges ; CNRS ; XLIM UMR 7252 ; MATHIS

{moulay.barkatou,thomas.cluzeau,ali.el-hajj}@unilim.fr

## ABSTRACT

In this paper, we first provide a unified algorithm for computing simple forms for systems of pseudo-linear equations. We prove that the existing methods for linear differential and difference systems can be extended to handle more general pseudo-linear systems. We explain how the reduction to a simple form can be used to compute efficiently local data for a system of pseudo-linear equations. We then propose an alternative, again based on simple forms, to previous algorithms for computing rational solutions of pseudo-linear systems. Moreover we develop a new algorithm for computing rational solutions of systems in two variables composed of linear differential and difference equations. Finally, we show that this algorithm can be generalized to the case of a system of partial pseudo-linear equations. All the algorithms described in this paper have been implemented in Maple and some examples of computations are provided.

## KEYWORDS

Computer algebra, Algorithms for functional equations, Local systems of pseudo-linear equations, Rational solutions of pseudo-linear systems,  $q$ -Difference systems, Implementation.

### ACM Reference Format:

Moulay Barkatou, Thomas Cluzeau, Ali El Hajj. 2019. Simple Forms and Rational Solutions of Pseudo-Linear Systems. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '19)*, July 15–18, 2019, Beijing, China. ACM, Beijing, China, 8 pages. <https://doi.org/10.1145/3326229.3326246>

## 1 INTRODUCTION

In the same line as the authors of [10], we use the mathematical framework of pseudo-linear algebra ([20, 25]) to develop algorithms for studying and solving *pseudo-linear systems* (see Definition 2.1): a large class of linear functional systems including the usual differential, difference and  $q$ -difference systems.

The first half of this paper concerns the local study of a pseudo-linear system. Moser's reduction [5, 17, 27] and super-reduction algorithms [4, 10, 17, 21, 24] have been proved to be relevant for studying locally differential and difference systems. Another useful form for the local study is the so-called *simple form* that has been introduced in [6]. Simple forms are easier to compute than super-irreducible ones and they are often sufficient to get the most

important local data. The notions of super-irreducible forms and simple forms have been introduced in [9, 10, 21] in the pseudo-linear setting and a generic algorithm for computing super-irreducible forms has been given. The method proposed in [10, 21] to construct a simple form requires to compute first a super-irreducible form which might be costly. The first direct (i.e., without recourse to super-reduction) algorithms for computing simple forms have been given in [11, 15, 22] for differential systems and in [13] for difference systems. The first main contribution of the present paper is to prove that these algorithms can be adapted to handle more general pseudo-linear systems. We provide an implementation [14] which is generic and flexible enough to treat a large class of pseudo-linear systems. We show that our algorithm can be used for computing local data (indicial equation, regular solutions,  $k$ -simple forms).

The second half of this paper deals with the computation of rational solutions of a set of fully integrable pseudo-linear systems in one or several variables. The problem of computing rational solutions of ordinary linear functional systems has been well studied mainly in the differential [6], difference [3] and  $q$ -difference [2] cases. It is well known that a crucial step is the computation of degree bounds both for the denominator and the numerator of rational solutions. The key point to get such bounds is to compute the *indicial polynomial* of the system at the so-called *fixed singularities*. We use here simple forms to compute the indicial polynomial as an alternative to super-reduction [4, 24] or EG-eliminations [1] algorithms. This provides another method for differential, difference and  $q$ -difference systems and this also permits to compute rational solutions of any pseudo-linear system. The second main contribution of the present paper is an algorithm for computing rational solutions of a fully integrable system of several pseudo-linear systems. We provide the details of the algorithm for the case of a difference/differential system and explain how this can be generalized to an arbitrary number of any type of pseudo-linear systems.

The rest of the paper is organized as follows. Section 2 recalls useful notions about pseudo-linear systems. Section 3 contains our first contribution, i.e., an algorithm for computing simple forms of pseudo-linear systems. Section 4 is devoted to the applications of simple forms for the local study of pseudo-linear systems. Section 5 deals with the computation of rational solutions of one single pseudo-linear system. Finally, Section 6 presents our second main contribution, i.e., an algorithm for computing rational solutions of a set of fully integrable pseudo-linear systems.

Our MAPLE implementation of the different algorithms developed in the present paper are freely available and the interested reader can find many examples on the webpage: see [14].

## 2 PSEUDO-LINEAR SYSTEMS

The present paper is concerned with general algorithms that can be applied to linear differential, difference or  $q$ -difference systems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ISSAC '19, July 15–18, 2019, Beijing, China

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6084-5/19/07...\$15.00

<https://doi.org/10.1145/3326229.3326246>

To cover all these cases (and more) in a unified framework, we use the general notion of *pseudo-linear systems*.

Let  $K$  be a commutative field of characteristic zero,  $\phi$  an automorphism of  $K$  and  $\delta$  a  $\phi$ -derivation, that is an additive map from  $K$  to itself satisfying the *Leibniz rule*  $\delta(ab) = \phi(a)\delta(b) + \delta(a)b$ , for all  $a, b \in K$ . We call the triplet  $(K, \phi, \delta)$  a  $\phi\delta$ -field. We recall that the constants of  $(K, \phi, \delta)$  are those elements  $c$  of  $K$  satisfying  $\phi(c) = c$  and  $\delta(c) = 0$ . They form a subfield of  $K$  which we denote by  $C_K$ . We will talk about a *local*  $\phi\delta$ -field when the field  $K$  is equipped with a discrete valuation  $v : K \rightarrow \mathbb{Z} \cup \{\infty\}$  and in that case we will require that the automorphism  $\phi$  is an *isometry*, that is it preserves the valuation:  $v(\phi(a)) = v(a)$ , for all  $a \in K$ .

Here are some familiar examples of  $\phi\delta$ -fields:

- Differential case:  $K = \mathbb{C}(x)$  or  $\mathbb{C}((x))$ ,  $\phi = \text{id}_K$ , and  $\delta = \frac{d}{dx}$ .
- Difference case:  $K = \mathbb{C}(x)$  or  $\mathbb{C}((x^{-1}))$ ,  $\phi$  the  $\mathbb{C}$ -automorphism defined by  $\phi(x) = x + 1$ , and  $\delta = \text{id}_K - \phi$ .
- $q$ -Difference case:  $K = \mathbb{C}(x)$  or  $\mathbb{C}((x))$ ,  $\phi$  the  $\mathbb{C}$ -automorphism defined by  $\phi(x) = qx$ ,  $q \in \mathbb{C}^*$ , and  $\delta = \text{id}_K - \phi$ .

Recall (see, e.g., [20, Lemma 1]) that for a general  $\phi\delta$ -field  $K$ , the  $\phi$ -derivation  $\delta$  is necessarily of the form  $\gamma(\text{id}_K - \phi)$ , for some  $\gamma \in K$ , when  $\phi \neq \text{id}_K$ , otherwise  $\delta$  is a usual derivation.

If  $M$  is a matrix (vector) with entries in  $K$ , and  $f$  is a map from  $K$  to itself, we define  $f(M)$  to be the matrix (vector) obtained by applying  $f$  to all entries of  $M$ . We note that the operations on matrices (vectors) commute with  $\phi$  and for matrices  $M, N$  one has  $\delta(M + N) = \delta(M) + \delta(N)$  and  $\delta(MN) = \delta(M)\phi(N) + M\delta(N) = \phi(M)\delta(N) + \delta(M)N$ .

When  $K$  is a local field, its valuation map  $v$  is extended to matrices and vectors with coefficients in  $K$  as follows: the valuation  $v(M)$  of a matrix (or vector)  $M$  is defined as the minimum of the valuations of its entries. When a *local parameter*  $t$  is fixed, any nonzero element  $a$  of  $K$  can be expanded as a formal Laurent series in  $t$  with coefficients belonging to the *residue field*  $C$  of  $K$ . Thus, every nonzero matrix  $M \in \mathbb{M}_n(K)$  possesses a  $t$ -adic expansion  $M = t^{v(M)} \sum_{i=0}^{+\infty} M_i t^i$ , where  $M_i \in \mathbb{M}_n(C)$  and  $M_0 \neq 0$  is called the *leading matrix* of  $M$ . Throughout this paper we will assume that the  $\phi$ -derivation  $\delta$  is not the zero map.

**Definition 2.1.** A first-order *pseudo-linear system* of size  $n$  over  $(K, \phi, \delta)$  is a system of the form

$$\delta(\mathbf{y}) = M \phi(\mathbf{y}), \quad (1)$$

where  $\mathbf{y}$  is a column-vector of  $n$  unknowns and  $M \in \mathbb{M}_n(K)$ .

Familiar examples of pseudo-linear systems are differential, difference or  $q$ -difference systems. In the *non-differential case* ( $\phi \neq \text{id}_K$ ), pseudo-linear systems are often given under the alternative form  $\phi(\mathbf{y}) = N \mathbf{y}$ , where  $N$  is an invertible matrix in  $\mathbb{M}_n(K)$ . However, one can easily transform such a system into the form (1) (see [10, Appendix A] or [21]). Conversely, if  $\phi \neq \text{id}_K$  and  $\delta = \gamma(\text{id}_K - \phi)$  with  $\gamma \in K^*$ , then a system of the form (1) can be rewritten as  $\phi(\mathbf{y}) = N \mathbf{y}$ , with  $N = (\gamma^{-1}M + I_n)^{-1}$  provided this inverse matrix exists. Consequently, when dealing with non-differential systems of the form (1), we will assume that the matrix  $(M + \gamma I_n)$  is invertible and call the system (1) *fully integrable*.

A *solution over  $K$*  of system (1) is a column-vector  $\mathbf{y} \in K^n$  such that  $\delta(\mathbf{y}) = M \phi(\mathbf{y})$ . The set of solutions over  $K$  of a system (1) of

size  $n$  is a vector space over the field of constants  $C_K$  of dimension at most  $n$  [8]. In order to *completely* solve a system of size  $n$  (that is forming a space of solutions of exact dimension  $n$ ), one needs to consider solutions over a suitable  $\phi\delta$ -field extension  $F$  of  $K$ . Such extension always exists (see, e.g., [18, 23, 29]).

### 3 SIMPLE FORMS

In this section we consider pseudo-linear systems over a local  $\phi\delta$ -field  $(K, \phi, \delta)$ . To fix ideas, we let  $K = C((t))$  be the field of Laurent series in a variable  $t$  equipped with the  $t$ -adic valuation  $v$  and  $C$  is a constant field. In our algorithms and implementations we assume that  $C$  is a computable field of characteristic zero. Recall that the automorphism  $\phi$  is an isometry with respect to  $v$  and  $\delta$  is not trivial.

#### 3.1 Preliminaries

Consider a pseudo-linear system of the form (1). Multiplying on the left System (1) by an appropriate invertible diagonal matrix whose nonzero entries are powers of  $t$ , one can always rewrite (1) as:

$$A \delta(\mathbf{y}) + B \phi(\mathbf{y}) = 0, \quad (2)$$

where  $A$  and  $B$  belong to  $\mathbb{M}_n(C[[t]])$ , with  $C[[t]]$  the ring of power series in  $t$ , and  $\det(A) \neq 0$ . Conversely, any system of the form (2) with  $A, B \in \mathbb{M}_n(C[[t]])$  such that  $\det(A) \neq 0$  can be rewritten in the form (1) with  $M = -A^{-1}B \in \mathbb{M}_n(K)$ . The fully integrability condition  $\det(M + \gamma I_n) \neq 0$  reads as  $\det(-B + \gamma A) \neq 0$ .

To any pseudo-linear system (2), we associate the *pseudo-linear operator*  $L = A \delta + B \phi$  acting on  $C[[t]]^n$  and  $K^n$ . The system is then written as  $L(\mathbf{y}) = 0$ . Note that, in terms of operators, we have  $\phi.T = \phi(T)\phi$  and  $\delta.T = T\delta + \delta(T)\phi$ , for all  $T \in \mathbb{M}_n(K)$ . Given a pseudo-linear system (2), one is often interested in finding an *equivalent* system satisfying some interesting properties. The notion of equivalence is defined as follows:

**Definition 3.1.** Two pseudo-linear operators  $L = A \delta + B \phi$  and  $L' = A' \delta + B' \phi$  are said to be *equivalent* if there exist two matrices  $S, T \in \text{GL}_n(K)$  such that  $L' = S L T$ , that is:

$$A' = S A T, \quad B' = S A \delta(T) + S B \phi(T). \quad (3)$$

Two pseudo-linear systems  $L(\mathbf{y}) = 0$  and  $L'(\mathbf{y}) = 0$  are equivalent if the operators  $L$  and  $L'$  are equivalent.

Note that even if we allow the transformations  $S$  and  $T$  to have their coefficients in the field  $K = C((t))$ , we can always ensure that the coefficient matrices of the resulting operator have their entries in the ring  $C[[t]]$ .

We define the notion of simple form for pseudo-linear systems (2).

**Definition 3.2.** The *leading matrix pencil* of a pseudo-linear system (2) is defined as the matrix polynomial  $L_\lambda = A_0 \lambda + B_0$ , where  $A_0, B_0$  denote the constant terms in the  $t$ -adic expansion of  $A, B$ . We say that System (2) is *simple* if  $\det(L_\lambda) \neq 0$ .

This definition also applies to pseudo-linear operators. Note that if  $A_0$  is invertible, then System (2) is necessarily simple but the converse does not always hold.

The notion of simple form was introduced for the first time in [6] and [3] where it was used as a main tool in the first direct algorithms for finding rational solutions of differential or difference systems with rational function coefficients. It has been also used in [16] in an

algorithm for computing the *regular solutions* of a system of linear differential equations. In the general pseudo-linear setting, simple forms have been used in [10, Section 4.2] and [21, Section 4.2.1]. In all these works, the methods proposed for computing a simple form of a given system of the form (2) require to compute first a *super-irreducible* form of the associated system (1), which turned out to be not very efficient. Later on, direct algorithms have been developed to construct a simple form of a (differential or difference) system without computing first a super-irreducible form (see [11, 15, 22] for the differential case and [13] for the difference case). The goal of this section is to provide a unified direct algorithm for computing simple forms of general pseudo-linear systems.

### 3.2 Computing simple forms

In this part, we shall prove that the method developed in [11, 13, 15, 22] for the purely differential (or difference) case can be adapted to the general pseudo-linear setting. The key points are the use of the general notion of equivalence (see Definition 3.1) and the fact that the  $K$ -automorphism  $\phi$  preserves the valuation.

When System (2) is not simple, then we have  $v(\det(A)) > 0$  (otherwise,  $A_0$  would be invertible and hence  $\det(\lambda A_0 + B) \neq 0$ ).

The principle of the algorithm then consists in computing iteratively equivalent pseudo-linear systems  $A^{(i)} \delta(y) + B^{(i)} \phi(y) = 0$ ,  $i \geq 1$  satisfying  $v(\det(A^{(i+1)})) < v(\det(A^{(i)}))$  with  $A^{(0)} = A$ . Doing so, either we find a simple system or we finally obtain an equivalent system satisfying  $v(\det(A^{(i)})) = 0$  and such a system is necessarily simple. To decrease  $v(\det(A))$ , we first apply equivalence transformations to put the leading matrix pencil under the particular form (5).

Let us consider a non-simple pseudo-linear system of the form (2). The leading matrix  $A_0$  of  $A$  is then necessarily singular, so we have that  $r := \text{rank}(A_0) < n$ . By using constant transformation matrices  $S$  and  $T$  in Definition 3.1, we can suppose that  $A_0$  has the form:

$$A_0 = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}. \quad (4)$$

All non-simple systems considered in the method described below are transformed so that  $A_0$  complies to the form (4). If we partition  $B_0$  and  $L_\lambda$  in four blocks as in (4), we get:

$$B_0 = \begin{pmatrix} B_0^{11} & B_0^{12} \\ B_0^{21} & B_0^{22} \end{pmatrix}, \quad L_\lambda = \begin{pmatrix} I_r \lambda + B_0^{11} & B_0^{12} \\ B_0^{21} & B_0^{22} \end{pmatrix}.$$

Following the terminology of [13], we will refer to the last  $(n-r)$  rows of  $L_\lambda$  as the  $\lambda$ -free rows. We are now ready to state the main results on which our algorithm for computing simple forms relies: Proposition 3.3 shows that it is always possible to transform a pseudo-linear operator into an equivalent one with a leading matrix pencil having its  $\lambda$ -free rows linearly dependent. Proposition 3.5 shows that if the  $\lambda$ -free rows of the leading matrix pencil are linearly dependent, then we can reduce  $v(\det(A))$ .

**PROPOSITION 3.3.** *Consider a non-simple pseudo-linear system of the form (2) such that  $A_0$  has the form (4) and the  $\lambda$ -free rows of  $L_\lambda$  are linearly independent. Then one can compute two matrices  $S, T \in GL_n(K)$  such that the equivalent operator  $\widehat{L} = S L T$  has an*

*associated leading matrix pencil of the form*

$$\widehat{L}_\lambda = \begin{pmatrix} I_p \lambda + R^{11} & R^{12} & R^{13} \\ 0 & I_{r-p} \lambda + R^{22} & R^{23} \\ 0 & R^{32} & R^{33} \end{pmatrix}, \quad \begin{cases} 0 \leq p \leq r, \\ \text{rank}(R^{32} \ R^{33}) < n-r. \end{cases} \quad (5)$$

Moreover if we note  $\widehat{L} = \widehat{A} \delta + \widehat{B} \phi$ , then  $v(\det(\widehat{A})) = v(\det(A))$ .

**PROOF.** The proof is an adaptation of those of [13, Lemma 1 and Proposition 8] to the general pseudo-linear setting. We recall the main points. The first step consists in applying recursively constant equivalence transformations in order to obtain an operator  $L' = S L T$  having a leading matrix pencil of the particular form:

$$L'_\lambda = \begin{pmatrix} I_p \lambda + R^{11} & 0 & 0 \\ R^{21} & I_{r-p} \lambda + R^{22} & R^{23} \\ R^{31} & R^{32} & R^{33} \end{pmatrix}, \quad \begin{cases} 0 \leq p \leq r, \\ \text{rank}(R^{32} \ R^{33}) < n-r. \end{cases} \quad (6)$$

As  $S$  and  $T$  are constant matrices, the formulas (3) simply reduce to  $A' = S A T$  and  $B' = S B T$ , so the same process, as in [13, Lemma 1 and Proposition 8], applies to any pseudo-linear system whatever  $\phi$  and  $\delta$  are. In order to get the particular form (5), we then apply the equivalence transformation defined by  $S = \text{diag}(t^{-1} I_p, I_{r-p}, I_{n-r})$  and  $T = S^{-1}$ . This can be directly checked from Formulas (3) as in the proof of [13, Proposition 8] since  $\phi$  preserves the valuation which is crucial here. Now Formulas (3) yield  $\widehat{A} = S A' S^{-1}$  so that  $v(\det(\widehat{A})) = v(\det(A')) = v(\det(A))$ .  $\square$

**Example 3.4.** Let us illustrate the result of Proposition 3.3 on a system of  $q$ -difference equations. Let  $t = x$  and  $q \in C$  a nonzero element which is not a root of unity. We consider the  $q$ -difference system  $D \phi(y) = N y$  with  $\phi$  defined by  $\phi(x) = q x$  and

$$D = \begin{bmatrix} q^3 x - q x & 0 \\ 0 & q x + \alpha \end{bmatrix}, \quad N = \begin{bmatrix} q^4(x-1) & q^2 x - q^4 x - \alpha q^4 + \alpha q^2 \\ x & 0 \end{bmatrix},$$

where  $\alpha$  denotes a nonzero parameter. Note that for  $\alpha = 100$ , we find again the  $q$ -difference system considered in [2, Section 4]. Introducing the  $\phi$ -derivation  $\delta = \phi - \text{id}_K$ , this system can be rewritten as the pseudo-linear system  $L(y) = A \delta(y) + B \phi(y) = 0$ , where

$$A = \begin{bmatrix} x & 0 \\ 0 & x + \alpha \end{bmatrix}, \quad B = \begin{bmatrix} -x & q x + \alpha \\ -\frac{x}{q} & \frac{(q^3 - q^2 + q)x + \alpha}{q^2} \end{bmatrix}.$$

The operator  $L$  is clearly not simple and we have  $v(\det(A)) = 1$ . By replacing  $L$  by  $P L P$  where  $P$  is a permutation matrix we can achieve that  $A_0$  is of the form (4) with  $r = 1$ . The leading matrix pencil of the operator is then

$$L_\lambda^{(1)} = \begin{bmatrix} \lambda + q^{-2} & 0 \\ \alpha & 0 \end{bmatrix}.$$

We now multiply the operator on the left by the constant matrix

$$S^{(2)} = \begin{bmatrix} 1 & -\frac{1}{\alpha q^2} \\ 0 & 1 \end{bmatrix},$$

whose first row is a convenient element of the left null-space of  $L_0^{(1)}$ . This yields an equivalent operator whose leading matrix pencil is

$$L_\lambda^{(2)} = S^{(2)} L_\lambda^{(1)} = \begin{bmatrix} \lambda & 0 \\ \alpha & 0 \end{bmatrix},$$

which complies with the form (6). We then proceed by applying the diagonal transformations defined by  $S^{(3)} = \text{diag}(x^{-1}, 1)$  and  $T^{(3)} = \text{diag}(x, 1)$  to get a new operator whose leading matrix pencil

$$L_\lambda^{(3)} = \begin{bmatrix} \lambda + q - 1 & \frac{-\lambda - q + 1}{\alpha q^2} \\ 0 & 0 \end{bmatrix},$$

is of the form (5). We then apply a constant transformation to put  $A_0$  in the required form (4), and finally get an equivalent operator with

$$A^{(4)} = \begin{bmatrix} \frac{\alpha+x}{\alpha} & \frac{x}{\alpha^2 q^2} \\ 0 & x \end{bmatrix}, \quad L_\lambda^{(4)} = \begin{bmatrix} \lambda + q - 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that we have  $v(\det(A^{(4)})) = v(\det(A)) = 1$ .

**PROPOSITION 3.5.** *Consider a non-simple pseudo-linear system of the form (2) such that its associated leading matrix pencil  $L_\lambda$  complies to the form (5). Then one can compute an invertible matrix  $S \in \text{GL}_n(K)$  such that the equivalent operator  $\widehat{L} = S L = \widehat{A} \delta + \widehat{B} \phi$  satisfies  $v(\det(\widehat{A})) < v(\det(A))$ .*

**PROOF.** We just need to check that the proof of [13, Proposition 9] can be adapted to the general pseudo-linear setting. We first use the hypothesis  $\text{rank}(R^{32} \ R^{33}) < n - r$  to compute a constant transformation  $S_1$  which reduces  $L$  to a new operator  $\bar{L} = S_1 L = \bar{A} \delta + \bar{B} \phi$  such that the matrices  $\bar{A}_0$  and  $\bar{B}_0$  have their  $(r + i)$ th row equal to zero, for some position  $r + i$ . We then apply the diagonal transformation  $S_2 := \text{diag}(1, \dots, 1, t^{-\mu}, 1, \dots, 1)$  with  $\mu = \min\{v(A(r + i, \cdot)), v(B(r + i, \cdot))\} > 0$ , where  $A(i, \cdot)$  denotes the  $i$ th row of a matrix  $A$ , to get an equivalent operator  $\widehat{L} = S_2 S_1 L = \widehat{A} \delta + \widehat{B} \phi$ . We thus have the equalities  $v(\det(\widehat{A})) = v(\det(S_2)) + v(\det(S_1)) + v(\det(A)) = -\mu + v(\det(A))$ . This implies  $v(\det(\widehat{A})) < v(\det(A))$  which ends the proof. For more details, see [13, Proposition 9] or [22, Proposition 4.3.1].  $\square$

**Example 3.6.** In Example 3.4, without modifying the valuation of the determinant of the  $A$ -matrix, we have obtained an equivalent operator  $L^{(4)}$  whose leading pencil is of the form (5). We shall then proceed by reducing the value of the valuation of the determinant of the  $A$ -matrix. We note that here  $r = 1$  and the second row of the leading pencil  $L_\lambda^{(4)}$  is already zero. So, we apply the transformations  $S^{(5)} = \text{diag}(1, x^{-1})$  and  $T^{(5)} = I_2$  to obtain the new equivalent operator  $L^{(5)} = S^{(5)} L^{(4)} T^{(5)} = A^{(5)} \delta + B^{(5)} \phi$  with

$$A^{(5)} = \begin{bmatrix} \frac{x}{\alpha} + 1 & \frac{x}{\alpha^2 q^2} \\ 0 & 1 \end{bmatrix}, \quad B^{(5)} = \begin{bmatrix} \frac{(q-1)(\alpha+x(q+1))}{\alpha} & \frac{x(q^2-1)}{\alpha^2 q^2} \\ (qx+\alpha)q & \frac{\alpha+(x-\alpha)q}{\alpha q} \end{bmatrix}.$$

As expected, we have  $v(\det(A^{(5)})) = 0 < 1$  and  $L^{(5)}$  is then a simple  $q$ -difference operator with leading matrix pencil given by:

$$L_\lambda^{(5)} = \begin{bmatrix} \lambda + q - 1 & 0 \\ \alpha q & \frac{1+(\lambda-1)q}{q} \end{bmatrix}.$$

### 3.3 Complexity estimate and implementation

Applying recursively Propositions 3.3 and 3.5 provides an algorithm, called SimpleForm below, to compute a simple form of a pseudo-linear system. The arithmetic complexity of this algorithm is studied in [22] in the case of differential systems. However, the algorithm merely performs linear algebra calculations on the first coefficients in the  $t$ -adic expansions of the matrices  $A$  and  $B$  defining System (2). Indeed, apart from constant transformations, Formulas (3) are only used with diagonal matrices  $S$  and  $T$  of a very particular form, e.g., only powers of  $t$  can appear in the diagonal, so that the automorphism  $\phi$  and the  $\phi$ -derivation  $\delta$  will not influence the number of asymptotic arithmetic operations. We can thus state the following complexity estimate of our algorithm:

**PROPOSITION 3.7.** *Let us consider a pseudo-linear system (2). Let  $d = v(\det(A))$  and suppose that the  $t$ -adic expansions of  $A$  and  $B$  are known up to an order  $k \geq d$ . Then Algorithm SimpleForm computes a simple form of (2) using at most  $O(n^{\omega+1} d + k n^3 d)$  arithmetic operations in the field  $C$ , where  $\omega$  denotes the linear algebra exponent.*

**PROOF.** This follows from the previous explanations and [22, Lemmas 4.3.1, 4.3.2, 4.3.3].  $\square$

The SIMPLEFORM algorithm has been implemented in MAPLE and the implementation is freely available with a list of examples covering several different types of pseudo-linear systems (see [14]). The automorphism  $\phi$  and the  $\phi$ -derivation  $\delta$  are given by the user as MAPLE procedures. For instance to treat the  $q$ -difference system considered in Examples 3.4 and 3.6, the user must first define:

```
> PhiAction:= proc(M,x) return subs(x=q*x,M) end;
> DeltaAction:= proc(M,x) return PhiAction(M,x)-M end;
The user then enters the matrices  $A$  and  $B$  with rational function entries, specifies the local parameter  $t$  (for instance,  $t = x$  for Examples 3.4 and 3.6) and runs:
> SimpleForm(A,B,x,t,DeltaAction,PhiAction);
```

$$\begin{bmatrix} \frac{\alpha+x}{\alpha} & \frac{x}{\alpha^2 q^2} \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \frac{xq^2}{\alpha} + q - 1 - \frac{x}{\alpha} & \frac{x}{\alpha^2} - \frac{x}{\alpha^2 q^2} \\ xq^2 + \alpha q & -1 + \frac{x}{\alpha} + q^{-1} \end{bmatrix},$$

$$\begin{bmatrix} -\frac{1}{\alpha x q^2} & \frac{1}{\alpha x} \\ x^{-1} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ x & \frac{x}{\alpha q^2} \end{bmatrix}, \frac{(\lambda + q - 1)(\lambda q - q + 1)}{q}, 0.040.$$

The output is a list containing respectively the matrices  $\widehat{A}$  and  $\widehat{B}$  of the equivalent simple operator  $\widehat{L} = \widehat{A} \delta + \widehat{B} \phi$ , the two invertible matrices  $S$  and  $T$  such that  $\widehat{L} = S L T$ , the determinant of the leading matrix pencil of  $\widehat{L}$  and finally the execution time.

### 4 SIMPLE FORMS AND LOCAL ANALYSIS

Simple forms are very useful for the *local analysis* of pseudo-linear systems of the form (1) over the field  $C(x)$  of rational functions near “fixed” singular points, i.e., poles of the coefficient matrix  $M$  which are fixed by the automorphism  $\phi$ . For instance, in the purely differential case all poles of  $M$  (including the potential pole at  $\infty$ ) are concerned, while only the point at  $\infty$  is concerned in the difference case ( $\phi(x) = x + 1$ ). In the  $q$ -difference case ( $\phi(x) = qx$ ,  $q \neq 0, 1$ ) one has two fixed points: 0 and  $\infty$ .

In this section, we show how Algorithm SimpleForm can be used to determine the nature of a singularity in the regular / irregular classification, to compute a basis of regular solutions, and to compute the so called *k-simple forms* which are closely related to super-irreducible forms. The results proposed here will be further developed in an extended version of this paper.

In what follows, we consider pseudo-linear systems of the form (1) with rational function coefficients and a fixed singular point  $x_0$ . We define a local parameter  $t$  (for instance,  $t = x - x_0$  or  $t = x^{-1}$  depending on  $x_0$  being finite or not) and we imbed  $C(x)$  in the local field  $K = C((t))$ , with  $C$  a computable subfield of  $\overline{\mathbb{Q}}$ .

#### 4.1 Regular solutions and regular singularities

We first recall useful notions concerning *local pseudo-linear systems* using the notations of [10, 21].

The degree of a derivation  $\delta$  is defined by:

$$\omega(\delta) = \inf_{a \in K, a \neq 0} v(a^{-1} \delta a).$$

If  $\delta \neq 0$ , then one has  $\omega(\delta) = v(t^{-1} \delta t)$  and for a pseudo-linear system with  $\phi \neq \text{id}_K$  and  $\delta = \gamma(\text{id}_K - \phi)$  for some  $\gamma \in K$ , we have  $\omega(\delta) = v(\gamma)$ .

Given an automorphism  $\phi$  preserving the valuation of  $K$  and a  $\phi$ -derivation  $\delta$  of degree  $\omega$ , there exist  $c, d \in C^*$  such that:

$$\phi(t) = ct + O(t^2), \quad t^{-\omega} \delta(t) = dt + O(t^2),$$

and for  $h \in \mathbb{N}_+$ , we have:

$$\phi(t^h) = c^h t^h + O(t^{h+1}), \quad t^{-\omega} \delta(t^h) = d[h]_c t^h + O(t^{h+1}),$$

where  $[h]_c$  is defined by:

$$[h]_c = \begin{cases} \frac{1 - c^h}{1 - c} & ; \quad c \neq 1, \\ h & ; \quad c = 1. \end{cases}$$

The notation  $[h]_c$  is extended to  $h \in C^*$  using  $c^h = \exp(h \log(c))$ .

Let us now introduce  $\varphi(\lambda) := \det(d[\lambda]_c A_0 + c^\lambda B_0)$ . The *indicial equation* of a simple pseudo-linear system of the form  $A t^{-\omega} \delta(y) + B \phi(y) = 0$  is then defined by  $\varphi(\lambda) = 0$ . When the system is simple,  $\det(L_\lambda) = \det(A_0 \lambda + B_0) \neq 0$  so that  $\varphi(\lambda) \neq 0$ . For the differential and the difference cases, we have  $c = d = 1$  which yields  $\varphi(\lambda) = \det(L_\lambda)$ , a polynomial in  $\lambda$ . Otherwise, if  $c \in C^*$  is not a root of unity and  $d$  is not an eigenvalue of the matrix  $(A_0 \lambda + (c - 1) B_0)$ , then  $\varphi(\lambda)$  is a polynomial in  $c^\lambda$ . By abuse of language, we shall also call it polynomial and its roots are the values of  $\lambda$  such that  $\varphi(\lambda) = 0$ . They can be computed using [10, Lemma 3.1]. The functions  $t^\lambda$  for  $\lambda \in \overline{C}$  and  $\log(t)$  are involved in the local analysis of differential systems. In the pseudo-linear setting, we introduce their analog as follows: we define  $e_\lambda(t)$  and  $u(t)$  to be two elements in some extension of  $K$  such that:

$$\frac{\phi(e_\lambda(t))}{e_\lambda(t)} = c^\lambda + O(t), \quad t^{-\omega} \frac{\delta(e_\lambda(t))}{e_\lambda(t)} = d[\lambda]_c + O(t),$$

and

$$t^{-\omega} \delta(u(t)) = 1 + O(t).$$

A simple pseudo-linear system (2) then admits  $\deg(L_\lambda)$  linearly independent local solutions  $y_i$ 's of the form:

$$y_i(t) = e_{\lambda_i}(t) \sum_{j=0}^{m_i-1} z_{i,j}(t) \frac{u(t)^{m_i-j}}{(m_i-j)!}, \quad i = 1, \dots, \deg(L_\lambda), \quad (7)$$

where  $\lambda_i \in \overline{C}$  is a root of multiplicity  $m_i$  of the indicial polynomial  $\varphi(\lambda)$  and  $z_{i,j}(t) \in C[[t]]^n$ . Local solutions of the form (7) are called *regular solutions*.

In order to compute a basis of regular solutions of a given pseudo-linear system, one then needs to compute the indicial polynomial  $\varphi(\lambda)$ . This polynomial can be read off from the system (2) only if the system is in simple form. In [10, 21] and in previous works restricted to the differential or difference case, it is proposed to compute a super-irreducible form of the system to obtain  $\varphi(\lambda)$ . The algorithm SimpleForm developed in Section 3 can then be used to compute the indicial polynomial of any pseudo-linear system directly without computing first a super-irreducible form. Note that such an approach has already been proposed for the differential [11, 22] and difference [13] cases. This improves the algorithms that have been proposed for computing the regular solutions in the differential/difference cases and our new algorithm allows to extend them to more general classes of pseudo-linear systems.

*Example 4.1.* Let us go back to the  $q$ -difference system considered in Examples 3.4 and 3.6. For a local  $q$ -difference system at  $x = 0$ , we take as local parameter  $t = x$ . Then we have  $\omega = 0$ ,  $c = q$  so that  $c \neq 1$ ,  $d = q - 1$ ,  $e_\lambda(x) = x^\lambda$ , and  $u(x) = \log_q(x)$ . The output of Algorithm SimpleForm given in Section 3.3 provides a simple form with  $\det(L_\lambda) = (\lambda q - q + 1)(\lambda + q - 1)q^{-1}$  which yields  $\varphi(\lambda) = q^2 \lambda - q^{\lambda-1} - q^{\lambda+1} + 1$ . The roots of  $\varphi(\lambda)$  are thus  $\lambda = \pm 1$  and from (7), it follows that the system admits two linearly independent regular solutions at  $x = 0$  given by  $y_1(x) = x z_{1,0}(x)$ ,  $y_2(x) = x^{-1} z_{2,0}(x)$  without logarithmic terms as  $m_1 = m_2 = 1$ .

Algorithm SimpleForm can also be used to determine the nature of a singularity in the regular / irregular classification. Indeed, the singularity of System (2) is regular if the system admits a basis of  $n$  regular solutions of the form (7). We then get the following lemma:

**LEMMA 4.2.** *The singularity of a pseudo-linear system (2) is regular if and only if  $\deg(\det(\widehat{L}_\lambda)) = n$ , where  $\widehat{L}_\lambda$  is the leading matrix pencil of the simple system returned by Algorithm SimpleForm when applied to (2).*

Note that  $\deg(\det(\widehat{L}_\lambda)) = n$  is equivalent to  $\det(\widehat{A}_0) \neq 0$ . Algorithm SimpleForm then provides an alternative to Moser's reduction ([4, 5, 27]) for recognizing the nature of a singularity.

#### 4.2 k-simple forms and super-irreducible forms

We keep the notation of the previous section and consider a pseudo-linear system of the form (1). The matrix  $M \in \mathbb{M}_n(K)$  can be uniquely written as  $M(t) = t^{\omega-p} \sum_{i=0}^{+\infty} M_i t^i$ , with  $M_0 \neq 0$  and  $p \in \mathbb{N}$  is called the *Poincaré rank* of the system. For any non-negative integer  $k$ , we define  $A^{(k)}(t) = \text{diag}(t^{\alpha_1}, \dots, t^{\alpha_n})$  with  $\alpha_i = \max(0, \omega - k - v(M(i, \cdot)))$ , and  $B^{(k)}(t) = -t^{k-\omega} A(t)^{(k)} M(t)$  so that  $B^{(k)}(t) \in \mathbb{M}_n(C[[t]])$  since we have  $v(A^{(k)}(t) M(t)) \geq \omega - k$ .

*Definition 4.3.* Let  $k$  be a nonnegative integer and  $\delta_k = t^{k+\omega} \delta$ . Then System (1) is said to be  $k$ -simple if the pseudo-linear system  $A^{(k)}(t) \delta_k(y) + B^{(k)}(t) \phi(y) = 0$  is simple, i.e.,  $\det(A_0^{(k)} \lambda + B_0^{(k)}) \neq 0$ .

Note that System (1) is necessarily  $k$ -simple for  $k \geq p$  since in this case the  $\alpha_i$ 's are all equal to zero so that  $A^{(k)} = I_n$ . In the differential case,  $k$ -simple forms have been shown to be very useful for computing *local data* at an irregular singularity (see [15, 22]).

Another notion that is used to study locally pseudo-linear systems is the super-reduction notion. It has been first introduced by Hilali-Wazner in [24] for the purely differential case as a generalization of *Moser-reduction* and then generalized to the difference case by the first author in [4]. More recently, it is introduced in the pseudo-linear setting in [9, 10, 21]. For space restriction reasons we shall not recall the exact definition of super-irreducible forms here; we just note that a system of Poincaré rank  $p > \omega$  is super-irreducible (with respect to the rows) if and only if it is  $k$ -simple for all  $k = 0, \dots, p-1$ . The reader is invited to consult [9, 10, 15, 21] for more details.

As it has already been pointed out, super-reduction algorithms can be used to compute simple forms of pseudo-linear systems. More generally, we have the following result:

**LEMMA 4.4.** *If a pseudo-linear system is super-irreducible (with respect to the columns), then it can be rewritten as a  $k$ -simple system for any  $k \in \{0, \dots, p-1\}$ .*

We refer to [15, Lemma 1.1] for a proof in the differential case that can be easily adapted to the pseudo-linear setting. Computing super-irreducible forms can then be a way to compute  $k$ -simple forms. However, a pseudo-linear system which is  $k$ -simple for a fixed  $k$  is not necessarily super-irreducible. So if one is interested in computing a  $k$ -simple form for just a value of  $k$  then direct methods have to be preferred. Our algorithm `SimpleForm` provides such a direct method. Indeed, for computing a  $k$ -simple form of a pseudo-linear system of Poincaré rank  $p$  for a given value of  $k < p$ , it is sufficient to run `SimpleForm` with the adequate derivation  $\delta_k$ .

Finally, we point out that Algorithm `SimpleForm` can also be used to compute super-irreducible forms. Indeed, we have:

**LEMMA 4.5.** *If a pseudo-linear system is  $k$ -simple for all values of  $k \in \{0, \dots, p-1\}$ , then it is super-irreducible.*

This lemma suggests the following algorithm for computing super-irreducible forms. Apply Algorithm `SimpleForm` to compute a  $(p-1)$ -simple form. Then apply Algorithm `SimpleForm` to the system obtained to compute a  $(p-2)$ -simple form and iterate this process until we get a 0-simple form. The correctness of this method is based on the results on the preservation of the simplicity developed in [15, Section 3] and [22, Section 4.5] for differential systems.

## 5 RATIONAL SOLUTIONS OF A SINGLE PSEUDO-LINEAR SYSTEM

Let  $K = C(x)$  be the field of rational functions in  $x$  with coefficients in  $C$ . In this section we consider the problem of computing rational solutions  $y \in K^n$  of a pseudo-linear system  $A \delta(y) + B \phi(y) = 0$ , where  $A, B \in \mathbb{M}_n(C[x])$ . Algorithms performing this task have been developed for the familiar families of pseudo-linear systems: we refer respectively to [6], [3], and [2] for differential, difference and

$q$ -difference systems. These algorithms use the following strategy. They first compute a *universal denominator*  $u \in C[x]$  which satisfies that every rational solution  $y \in K^n$  can be written  $y = z/u$  with  $z \in C[x]^n$ . Then, performing the change of variables  $y = z/u$ , we are reduced to computing polynomial solutions  $z \in C[x]^n$  of the new pseudo-linear system  $\phi(u) A \delta(z) + (-\delta(u) A + u B) \phi(z) = 0$ .

The universal denominator  $u$  is a finite product of terms of the form  $u_i^{m_i}$  with  $m_i \in \mathbb{N}$  and the irreducible polynomials  $u_i$ 's whose roots are the finite singularities of the system can be read off from the system. The problem then consists in finding the  $m_i$ 's. Here two cases have to be distinguished. If a finite singularity is not fixed by the automorphism  $\phi$ , then the corresponding  $m_i$  can be obtained by resultant and gcd's calculations. This is the case for all finite singularities of a difference system (see [3, 7]) and for all finite singularities of a  $q$ -difference system except zero (see [2]). Otherwise, if the singularity is fixed by  $\phi$  (for instance  $x = 0$  in the  $q$ -difference case or all the singularities in the differential case), then the corresponding  $m_i$  can not be obtained as before and one needs to compute the indicial polynomial  $\varphi(\lambda)$  of the system at this singularity and take  $-m_i$  as the smallest integer root of  $\varphi(\lambda)$ .

In [6], [3], and [2], the method proposed to compute the indicial polynomial  $\varphi(\lambda)$  consists either in applying the super-reduction (see Section 4) or EG-eliminations (see [1]). The `SimpleForm` algorithm developed in the present paper allows to compute directly the indicial polynomial (see Section 4) and provides thus an alternative to previous methods. In particular, our generic implementation allows to obtain new efficient implementations for computing rational solutions of pseudo-linear systems. Note also that EG-eliminations could provide extra solutions that need to be removed afterwards which is not the case when using simple forms.

Once the universal denominator  $u$  is computed, we are then reduced to computing polynomial solutions. Here, we proceed as follows. We compute a bound  $m$  for the degree of the polynomial solutions and then we use the method described in [3, 6, 10] to compute the coefficients of the polynomials. The degree bound  $-m$  is here again obtained from the indicial polynomial at infinity.

*Example 5.1.* Let us come back to the  $q$ -difference system considered in Examples 3.4, 3.6, 4.1. The universal denominator has the form  $u(x) = x^{m_0} (x + \alpha)^{m-\alpha}$ . The singularity  $x = -\alpha$  is not fixed by  $\phi$  so that using gcd's computations we find  $m-\alpha = 1$ . The singularity  $x = 0$  is fixed by  $\phi$ . Consequently, instead of using EG-eliminations as it is done in [2, Section 4], we use the `SimpleForm` algorithm. From the results obtained in Example 4.1, we get  $m_0 = 1$  so that  $u(x) = x(x + \alpha)$ . In this particular example, the change of variables  $y = z/u$  yields a system that is already simple at infinity and the corresponding indicial polynomial is  $\varphi(\lambda) = 1 - q^{-\lambda-1} - q^{-\lambda-3} + q^{-2\lambda-4}$ . The integer roots of  $\varphi(\lambda)$  are  $-1$  and  $-3$  so that we get the degree bound  $m = 3$ . We then proceed to get the polynomial solutions

$$z_1(x) = \begin{bmatrix} \alpha + x & x \end{bmatrix}^T, \quad z_2(x) = \begin{bmatrix} x^2 + \frac{x^3}{\alpha} & \frac{x^3}{q^2 \alpha} \end{bmatrix}^T.$$

Finally, we obtain the rational solutions of the original system:

$$y_1(x) = \begin{bmatrix} \frac{1}{x} & \frac{1}{\alpha+x} \end{bmatrix}^T, \quad y_2(x) = \begin{bmatrix} \frac{x}{\alpha} & \frac{x^2}{(\alpha+x)\alpha q^2} \end{bmatrix}^T.$$

The results are coherent with those obtained in [2, Section 4] for the case  $\alpha = 100$ . Note that in [2, Section 4], the author needs to apply

EG-eliminations to get the indicial polynomial at infinity whereas for this particular example, we can get it directly since the system obtained for polynomial solutions is already simple at infinity.

## 6 RATIONAL SOLUTIONS OF SEVERAL PSEUDO-LINEAR SYSTEMS

In this section, we present our second main contribution, i.e., an algorithm for computing rational solutions of a system of partial pseudo-linear equations. The underlying motivation for developing such an algorithm is that many special (transcendental) functions satisfy such partial pseudo-linear systems. We can think for instance of Legendre, Bessel or Tchebychev polynomials which satisfy both a system of differential equation and a system of difference equations. Systems of partial pseudo-linear equations have already been considered in [12] in the purely differential case and in [19, 26, 28] for more general systems. In particular, an algorithm for computing rational solutions of integrable connections is developed in [12] and an algorithm for computing hyperexponential solutions of systems over Laurent-Ore algebras is proposed in [26]. Note that a specific algorithm for rational solutions is useful in itself for decomposing / factoring the systems (see, for instance, [12, Example 3]).

Let  $K = C(x_1, \dots, x_m)$ . We consider a partial pseudo-linear system of the form:

$$\begin{cases} L_1(\mathbf{y}) := A_1 \delta_1(\mathbf{y}) + B_1 \phi_1(\mathbf{y}) = 0, \\ \vdots \\ L_m(\mathbf{y}) := A_m \delta_m(\mathbf{y}) + B_m \phi_m(\mathbf{y}) = 0, \end{cases} \quad (8)$$

with  $A_i \in \text{GL}_n(K)$ ,  $B_i \in \mathbb{M}_n(K)$ . For  $i = 1, \dots, m$ , the variable  $x_j$ 's ( $j \neq i$ ) are constants with respect to  $\phi_i$  and  $\delta_i$ . This allows to view  $L_i(\mathbf{y}) = 0$  as a pseudo-linear system with respect to  $x_i$  and where the other variables  $x_j$ 's are constant parameters. We always assume that System (8) is *integrable*, i.e.,  $[L_i, L_j] := L_i.L_j - L_j.L_i = 0$  and following the terminology of [19, Definition 2] and [28, 29], we further suppose that (8) is *fully integrable*, i.e., if  $\phi_i \neq \text{id}_K$  and  $\delta_i = \gamma_i (\text{id}_K - \phi_i)$ , then we suppose that  $-\gamma_i A_i + B_i$  is invertible.

**Definition 6.1.** A rational solution of (8) is a vector  $\mathbf{y} \in K^n$  that satisfies  $L_i(\mathbf{y}) = 0$ , for  $i = 1, \dots, m$ .

The goal of this part is to develop an algorithm for computing rational solutions of (8). To achieve this we shall follow the lines of the algorithm given in [12] for the purely differential case. We improve it by proposing a more efficient method to compute the new coefficient matrices when applying recursion. For the sake of clarity and for space restrictions reasons, we explain the details of the algorithm in the case of one difference system and one differential system. We then give the main ideas for partial pseudo-linear systems.

### 6.1 Difference/Differential systems

Let  $K = C(k, x)$  and consider the fully integrable system

$$\{\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x), \quad \mathbf{y}'(k, x) = B(k, x) \mathbf{y}(k, x)\}, \quad (9)$$

where  $\mathbf{y}' = \partial \mathbf{y} / \partial x$ ,  $A \in \text{GL}_n(K)$ ,  $B \in \mathbb{M}_n(K)$  and the integrability condition reads:

$$A'(k, x) = B(k+1, x) A(k, x) - A(k, x) B(k, x). \quad (10)$$

**REMARK.** The function  $\mathbf{y} = x^k$  is clearly a solution of the system  $\{\mathbf{y}(k+1, x) = x \mathbf{y}(k, x), \mathbf{y}'(k, x) = (k/x) \mathbf{y}(k, x)\}$  but it is not a rational solution in the sense of Definition 6.1 since  $x^k \notin K = C(k, x)$ .

Let us describe our method for computing rational solutions of System (9). We first consider the system  $\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x)$  as a difference system over  $C(x)(k)$  viewing  $x$  as a constant parameter independent from  $k$ . We compute a basis  $\mathbf{u}_1, \dots, \mathbf{u}_s \in K^n$  ( $0 \leq s \leq n$ ) of its rational solutions (see Section 5 or [3]). If we do not find any nonzero rational solution, then we are done as (9) does not admit any nonzero rational solution.

Let us complete  $\mathbf{u}_1, \dots, \mathbf{u}_s$  into a basis  $\mathbf{u}_1, \dots, \mathbf{u}_n$  of  $K^n$  and define  $P = \begin{pmatrix} U & V \end{pmatrix} \in \text{GL}_n(K)$ , where  $V \in \mathbb{M}_{n \times (n-s)}(K)$  has the vectors  $\mathbf{u}_{s+1}, \dots, \mathbf{u}_n$  as columns. Performing the change of variables  $\mathbf{y} = P \mathbf{z}$  in System (9), the differential system is transformed to  $\mathbf{z}' = P^{-1} (B P - P') \mathbf{z}$  which has the following properties:

**LEMMA 6.2.** With the above notation, let us write

$$\tilde{B} := P^{-1} (B P - P') = \begin{pmatrix} \tilde{B}^{11} & \tilde{B}^{12} \\ \tilde{B}^{21} & \tilde{B}^{22} \end{pmatrix},$$

where  $\tilde{B}^{11} \in \mathbb{M}_s(K)$ . Then the matrix  $\tilde{B}^{11} \in \mathbb{M}_s(C(x))$  does not depend on  $k$  and it is the unique solution of the matrix linear system  $U \tilde{B}^{11} = -(U' - B U)$ . Furthermore  $\tilde{B}^{21} = 0$ .

**PROOF.** With the previous notation, the equation  $P \tilde{B} = B P - P'$  yields  $U \tilde{B}^{11} + V \tilde{B}^{21} = -(U' - B U)$ . From the integrability condition (10),  $U' - B U$  is a solution of  $\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x)$  so that there exists a unique matrix  $C \in \mathbb{M}_s(C(x))$  (i.e., constant with respect to  $k$ ) such that  $U' - B U = U C$ . We then obtain  $U (\tilde{B}^{11} + C) + V \tilde{B}^{21} = 0$  which ends the proof as the columns of  $P = \begin{pmatrix} U & V \end{pmatrix}$  form a basis of  $K^n$ .  $\square$

The next theorem shows that we are now reduced to computing the rational solutions of the differential system  $\mathbf{z}'(x) = \tilde{B}^{11}(x) \mathbf{z}(x)$ .

**THEOREM 6.3.** Let  $U \in \mathbb{M}_{n \times s}(K)$  be a matrix whose columns form a basis of the rational solutions of  $\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x)$ . Let  $\tilde{B}^{11} \in \mathbb{M}_s(C(x))$  be the unique solution of the matrix linear system  $U \tilde{B}^{11} = -(U' - B U)$ . If  $\mathbf{z}_1, \dots, \mathbf{z}_r \in C(x)^s$  is a basis of rational solutions of  $\mathbf{z}'(x) = \tilde{B}^{11}(x) \mathbf{z}(x)$ , then  $U \mathbf{z}_1, \dots, U \mathbf{z}_r \in K^n$  is a basis of rational solutions of (9). Moreover, every rational solution of (9) can be obtained in such a way.

**PROOF.** Let  $\mathbf{z} \in C(x)^s$  be a rational solution of  $\mathbf{z}'(x) = \tilde{B}^{11}(x) \mathbf{z}(x)$  and let us consider  $\mathbf{y}(k, x) = U(k, x) \mathbf{z}(x)$ . We have  $\mathbf{y}(k+1, x) = U(k+1, x) \mathbf{z}(x) = A(k, x) U(k, x) \mathbf{z}(x) = A(k, x) \mathbf{y}(k, x)$ . Moreover  $\mathbf{y}' = U' \mathbf{z} + U \mathbf{z}' = U' \mathbf{z} + U \tilde{B}^{11} \mathbf{z} = B \mathbf{y}$ , by definition of  $\tilde{B}^{11}$ . This ends the first part of the proof. Now let  $\mathbf{y}$  be a solution of (9). In particular,  $\mathbf{y}$  is a solution of  $\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x)$  so that there exists  $\mathbf{z} \in C(x)^s$  such that  $\mathbf{y} = U \mathbf{z} = \begin{pmatrix} U & V \end{pmatrix} \begin{pmatrix} \mathbf{z}^T & 0^T \end{pmatrix}^T$ . Thus,  $\mathbf{y}$  is a solution of  $\mathbf{y}'(k, x) = B(k, x) \mathbf{y}(k, x)$  if and only if  $\begin{pmatrix} \mathbf{z}^T & 0^T \end{pmatrix}^T$  is a solution of  $\mathbf{y}' = \tilde{B} \mathbf{y}$ . This is equivalent to  $\mathbf{z}'(x) = \tilde{B}^{11}(x) \mathbf{z}(x)$  and yields the desired result.  $\square$

Theorem 6.3 naturally provides an algorithm for computing a basis of rational solutions of System (9). It proceed as follows:

- (1) Compute a matrix  $U \in \mathbb{M}_{n \times s}(K)$  whose columns form a basis of the rational solutions of  $\mathbf{y}(k+1, x) = A(k, x) \mathbf{y}(k, x)$ ,

- (2) Compute the unique solution  $\tilde{B}^{11} \in \mathbb{M}_s(C(x))$  of the matrix linear system  $U \tilde{B}^{11} = -(U' - BU)$ ,
- (3) Compute a basis  $z_1, \dots, z_r$  of the rational solutions of the differential system  $z'(x) = \tilde{B}^{11}(x) z(x)$ ,
- (4) Return  $U z_1, \dots, U z_r$ .

In Steps (1) and (3) if we do not find any nonzero rational solution, then we should return 0 since this implies that (9) does not admit nonzero rational solutions. Note that, contrary to [12] where a specific method is proposed, we propose here to solve the linear system for  $\tilde{B}^{11}$  by any fast algorithm.

This algorithm has been implemented in MAPLE. The implementation is available at [14] with some examples.

*Example 6.4.* Consider the system (9) with:

$$A(k, x) = \begin{bmatrix} \frac{k}{k+1} & 0 & \frac{2}{(k+1)(x+k)} & 0 \\ \frac{-k^2+(-x-1)k-x-1}{k+1} & \frac{x^2-1+x(k+1)}{k+1} & \frac{2}{(k+1)(x+k)} & \frac{-k^2+x^2-2k-2}{k+1} \\ 0 & 0 & \frac{x+k+1}{x+k} & 0 \\ \frac{(x-1)k+x}{k+1} & \frac{-x^2+1}{k+1} & \frac{-2}{(k+1)(x+k)} & \frac{x(k+1)-x^2+1}{k+1} \end{bmatrix},$$

$$B(k, x) = \begin{bmatrix} x^{-1} & 0 & \frac{-2}{x(x+k)} & 0 \\ \frac{x^2-1+(-k^2-k)x}{x^3-x} & \frac{x^2-2x-1}{x^2-1} & \frac{-2}{x(x+k)} & \frac{-k^2+x^2-k-2x-1}{x^2-1} \\ 0 & 0 & (x+k)^{-1} & 0 \\ -x^{-1} & -1 & \frac{2}{x(x+k)} & -1 \end{bmatrix}.$$

Computing rational solutions of  $y(k+1, x) = A(k, x) y(k, x)$  we get  $s = 2$  linearly independent solutions given by the columns of

$$U = \begin{bmatrix} -k^{-1} & -k^{-1} & 0 & k^{-1} \\ -1 & -1 & -\frac{x}{2} - \frac{k}{2} & 1 \end{bmatrix}^T.$$

Now, solving the linear system  $U \tilde{B}^{11} = -(U' - BU)$  we get:

$$\tilde{B}^{11} = \begin{bmatrix} x^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{M}_2(C(x)).$$

The differential system  $z'(x) = \tilde{B}^{11}(x) z(x)$  admits  $r = 2$  linearly independent rational solutions given by the columns of

$$Z = \begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}.$$

Finally, a basis of rational solutions of the original system is spanned by the columns of

$$Y = U Z = \begin{bmatrix} -\frac{x}{k} & -\frac{x}{k} & 0 & \frac{x}{k} \\ -1 & -1 & -\frac{x}{2} - \frac{k}{2} & 1 \end{bmatrix}^T.$$

## 6.2 The general case

The process explained in the previous section 6.1 can be generalized to partial pseudo-linear systems. Let us sketch the general iterative process for computing a basis of rational solutions of a system of the form (8). As the matrices  $A_i$  are invertible, we suppose that  $A_i = I_n$ . Let  $U \in \mathbb{M}_{n \times s}(K)$  be a matrix whose columns form a basis of the rational solutions of  $L_1(y)$ . Then to apply recursion one needs to compute the matrices  $\tilde{B}_i \in \mathbb{M}_s(C(x_2, \dots, x_n))$  solutions of the  $m-1$  matrix linear systems  $U \tilde{B}_i = L_i(U)$  for  $i = 2, \dots, m$ . These linear systems share the same matrix  $U$  and only the right hand sides can change. Consequently, to solve efficiently the  $m-1$

systems one can compute an LU decomposition of  $U$  at once and use it for solving each linear system. We then apply recursion by considering the  $m-1$  pseudo-linear systems  $\tilde{L}_i = I_s \delta_i + \tilde{B}_i \phi_i$  of size  $s \leq n$  with coefficients in  $C(x_2, \dots, x_m)$ . More details will appear in an extended version of this paper.

## REFERENCES

- [1] Abramov S. (1999) EG-eliminations, J. of Difference Equations and Applications, 5(4-5), p. 393-433.
- [2] Abramov S. (2002) A direct algorithm to compute rational solutions of first order linear  $q$ -difference systems. Discrete Mathematics 246, p. 3-12.
- [3] Abramov S., Barkatou M. (1998). Rational solutions of first order linear difference systems. In Proc. ISSAC'98, p. 124-131.
- [4] Barkatou M. (1989) Contribution à l'étude des équations différentielles et aux différences dans le champ complexe, PhD Thesis, Institut Nat. Polytech. Grenoble.
- [5] Barkatou M. (1995) A rational version of Moser's algorithm. In Proc. ISSAC'95, p. 297-302.
- [6] Barkatou M. (1999) On rational solutions of systems of linear differential equations. J. of Symb. Comp., 28, 547-567.
- [7] Barkatou M. (1999) Rational solutions of matrix difference equations: the problem of equivalence and factorization. In Proc. ISSAC'99, p. 277-282.
- [8] Barkatou M. (2007) Factoring Systems of Linear Functional Systems Using Eigenings. Computer algebra 2006, 22-42, World Sci. Publ., Hackensack, NJ.
- [9] Barkatou M., Broughton G., Pflügel E. (2008) Regular systems of linear functional equations and applications. In Proc. ISSAC'08, p. 15-22.
- [10] Barkatou M., Broughton G., Pflügel E. (2010) A monomial-by-monomial method for computing regular solutions of systems of pseudo-linear equations. Math. in Comp. Sci., 4(2-3), p. 267-288.
- [11] Barkatou M., Cluzeau T., El Bacha C. (2011) Simple forms of higher-order linear differential systems and their applications in computing regular solutions. J. of Symb. Comp., 46(6), p. 633-658.
- [12] Barkatou M., Cluzeau T., El Bacha C., Weil J.-A. (2012) Computing closed form solutions of integrable connections, In Proc. ISSAC'12, p. 43-50. IntegrableConnections package [http://www.unilim.fr/pages\\_perso/thomas.cluzeau/Packages/IntegrableConnections/PDS.html](http://www.unilim.fr/pages_perso/thomas.cluzeau/Packages/IntegrableConnections/PDS.html)
- [13] Barkatou M., Cluzeau T., El Bacha C. (2018) On the computation of simple forms and regular solutions of linear difference systems. Advances in Computer Algebra. WWC 2016. Springer Proc. in Mathematics & Statistics, vol 226.
- [14] Barkatou, M., Cluzeau, T., El Hajj, A. (2019) Pseudo-linear systems: simple forms and rational Solutions, [http://www.unilim.fr/pages\\_perso/ali.el-hajj/Implementations.html](http://www.unilim.fr/pages_perso/ali.el-hajj/Implementations.html)
- [15] Barkatou M., El Bacha C. (2013) On  $k$ -simple forms of first-order linear differential systems and their computation. J. of Symb. Comp., 54, p. 36-58.
- [16] Barkatou M., Pflügel E. (1999) An algorithm computing the regular formal solutions of a system of linear differential equations. J. of Symb. Comp., 28(4-5), 1999, p. 569-587.
- [17] Barkatou M., Pflügel E. (2009) On the Moser- and super-reduction algorithms of systems of linear differential equations and their complexity. J. of Symb. Comp., 44(8), p. 1017-1036.
- [18] Bialynicki-Birula, A. (1962) On Galois theory of fields with operators. Amer. J. Math. 84, p. 89-109.
- [19] Bronstein M., Li, Z., Wu, M. (2005) Picard-Vessiot extensions for linear functional systems. In Proc. ISSAC'05, p. 68-75.
- [20] Bronstein M., Petkovsek, M. (1996) An introduction to pseudo-linear algebra. Theor. Comput. Sci., 157, p. 3-33.
- [21] Broughton G. (2013) Symbolic algorithms for the local analysis of systems of pseudo-linear equations, PhD Thesis, Kingston University.
- [22] El Bacha C. (2011) Méthodes Algébriques pour la Résolution d'équations Différentielles Matricielles d'Ordre Arbitraire. PhD Thesis. Université de Limoges.
- [23] Franke, C. H. (1963) Picard-Vessiot theory of linear homogeneous difference equations. Trans. Amer. Math. Soc. 108, p. 491-515.
- [24] Hilali A. and Wazner A. (1987) Formes super-irréductibles des systèmes différentiels linéaires. Numer. Math., 50, p. 429-449.
- [25] Jacobson N. (1937) Pseudo-linear transformations. Annals of Mathematics, 38(2), second series, p. 484-507.
- [26] Li Z., Singer M. S., Wu M., Zheng D. (2006) A recursive method for determining the one-dimensional submodules of Laurent-Ore modules. In Proc. ISSAC'06, p. 200-208.
- [27] Moser J. (1960) The order of a singularity in Fuchs theory. Math. Z., p. 379-398.
- [28] Wu, M. (2005) On solutions of linear functional systems and factorization of modules over Laurent-Ore algebras. PhD Thesis, Univ. of Nice-Sophia Antipolis.
- [29] Wu M., Li Z. (2007) On solutions of linear systems and factorization of Laurent-Ore modules. Computer algebra 2006, 109-136, World Sci. Publ., Hackensack, NJ.