Symbolic methods for studying linear systems of differential and difference equation

Ali El Hajj

UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Sciences et Ingénierie pour l'Information

FACULTÉ DES SCIENCES ET TECHNIQUES

Année : 2021 THÈSE N°X

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline : Mathématiques et ses applications

présentée et soutenue publiquement par

Ali EL HAJJ

le 17 décembre 2021 à 14h

Algorithmes Symboliques pour l'Étude et la Résolution de Systèmes d'Équations Fonctionnelles Linéaires

Thèse dirigée par Moulay A. BARKATOU et Thomas CLUZEAU

JURY :

George LABAHN	Professeur, Université de Waterloo	Rapporteur
Sergei ABRAMOV	Professeur, Université de Moscou	Rapporteur
Guillaume CHÈZE	MCF HDR, Université Toulouse III	Examinateur
Alin BOSTAN	DR, INRIA Saclay Île-de-France	Examinateur
Simoné NALDI	MCF, Université de Limoges	Examinateur
Vladimir SALNIKOV	CR, Université La Rochelle	Examinateur
Moulay A. BARKATOU	Professeur, Université de Limoges	Directeur
Thomas CLUZEAU	MCF, Université de Limoges	Directeur

Remerciements

Tout d'abord, je tiens à exprimer ma plus profonde gratitude à Moulay BARKATOU et Thomas CLUZEAU pour leurs généreux conseils en temps et en connaissances, leur assistance, leur disponibilité, leur patience, et leur soutien continu tout au long de ma thése. J'ai eu beaucoup de chance de les avoir comme directeurs. Travailler avec eux a été extrêmement enrichissant sur les niveaux scientifique et personnel. Leur perfectionnisme m'a poussée à donner le meilleur de moi-même.

Aussi, j'exprime ma gratitude à George LABAHN et Sergei ABRAMOV, de me faire l'honneur de rapporter ma thèse. Je remercie galement Alin BOSTAN, Guillaume CHÈZE, Simoné NALDI, et Vladimir SALNIKOV d'avoir accepté d'être examinateurs pour cette thèse.

J'exprime tous mes remerciements à mes collègues de l'équipe de Calcul Formel, ainsi qu'aux personnels du Département Mathématiques Informatique de l'Université de Limoges pour m'avoir donné la chance de faire mon Master qui était le premier pas vers ma situation actuelle. À mes professeurs: Paul ARMAND, Samir ADLY, Moulay BARKATOU, Noureddine IGBIDA, Olivier RUATTA, Loïc BOURDIN, Jaques-Arthur WEIL et Olivier PROT, merci à chacun de vous. Un grand merci également à vous Annie, Débora et Sophie pour votre gentillesse et aide continue. Je remercie mes amis et collègues de Xlim pour les bons moments que nous avons partagés.

Le plus grand merci va à toute ma famille. Leur soutien inconditionnel et leur amour m'ont aidé réaliser mon rêve de devenir docteur. Enfin, à ma femme Carine, cette thèse repose sur ton soutien et tes encouragements de le début et jusqu'au dernier moment. Difficile en quelques mots de t'exprimer toute ma reconnaissance et tout mon amour.

Notations

$\mathbb{N}$	The set of nonnegative integers
$\mathbb{N}^*$	The set of positive integers
$\mathbb{Z}$	The ring of integers
$\mathbb{Q}$	The field of rational numbers
$\mathbb{C}$	The field of complex numbers
$\overline{C}$	The algebraic closure of a field C
$C[x]$	The ring of polynomials in x over a field C
$C[[x]]$	The ring of formal power series in x over a field C
$C(x)$	The field of rational functions in x over a field C
$C((x))$	The field of formal Laurent series in x over a field C
$\mathbb{M}_{m\times n}(K)$	The K-vector space of $m \times n$ matrices with entries in a field K
$\mathbb{M}_n(K)$	The K-algebra of $n \times n$ matrices with entries in a field K
$\mathrm{GL}_n(K)$	The group of $n \times n$ invertible matrices with entries in a field K
$K^n = \mathbb{M}_{1\times n}(K)$	
id_K	The identity map over a field K
0_n	The n-dimensional zero vector
I_n	The identity matrix of size n
$\mathrm{diag}(x_1, x_2, \dots, x_n)$	The diagonal matrix $\begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & x_n \end{pmatrix}$
A^{-1}	The inverse of an invertible square matrix A
A^T	The transpose of a matrix or vector A
$A(i,j)$	The $(i,j)^{\text{th}}$ entry of a matrix A
$A(i,.)$	The i^{th} row of a matrix A

$\operatorname{rank}(A)$	The rank of a matrix A
$\det(A)$	The determinant of a square matrix A
$\operatorname{den}(A)$	The denominator of a matrix $A \in \mathbb{M}_{m\times n}(C(x))$: the least common multiple (lcm) of the denominators of all entries of A
$f'(x)$	The first derivative of a function $f(x)$ w.r.t. x
$\deg(p)$	The degree of a univariate polynomial p
$p \mid q$	A polynomial p divides a polynomial q
$p \nmid q$	A polynomial p does not divide a polynomial q
$\nu(f)$ with $f \in \overline{C((t))}$	The t-adic valuation of f: $\nu(f) = m$ if $f = t^m(f_0 + f_1\, t + \ldots),\ f_0 \neq 0$ and $\nu(0) = \infty$
$\nu(M)$ with $M \in \mathbb{M}_{m\times n}(\overline{C((t))})$	The t-adic valuation of M: $\nu(M) = \min\{\nu\left(M(i,j)\right)\ ;\ 1 \le i \le m,\ 1 \le j \le n\}$

Contents

General Introduction

Nowadays, the theory of linear functional equations plays a central role in mathematics and widely contributes to the treatment of scientific problems in various fields such as chemistry, physics, mechanics, and control theory. Two well known particular systems of linear functional equations are first oder differential systems having the form

$$\mathbf{y}'(x) = A(x)\,\mathbf{y}(x) + b(x),$$

and first order difference systems having the form

$$\mathbf{y}(x+1) = A(x)\,\mathbf{y}(x) + b(x),$$

where in both equations, the unknown column vector $\mathbf{y}$, the matrix A and the column vector b, have entries which are functions of the variable x. These two familiar systems have been extensively studied over the past years from both a theoretical and algorithmic perspective, and significant progress has been achieved in these topics [8, 16, 25, 42, 59, 61, 64, 88, 90]. In spite of this progress, it has been shown, see e.g. [2, 49, 83], that linear differential and difference systems are strongly connected and share interesting properties. Moreover, some algorithms designed for one type of system can be adapted for the other. For instance, the Moser- and super-reduction algorithms [67, 78] originally developed for differential systems, have been adapted to treat difference systems, see [15, 16]. Also, existing algorithms for computing rational solutions follow the same general strategy for both systems, see for instance [7, 20]. So the theoretical and algorithmic similarities that both differential and difference systems share, suggest the existence of a common mathematical framework behind them. This mathematical framework is given by *pseudo-linear algebra*.

Pseudo-linear algebra is a wide area of mathematics with origins in the 1930's from works by Ore [79] and Jacobson [69]. Around 60 years later, Bronstein and Petkovšek introduced in [48] the basic objects (pseudo-linear systems, pseudo-derivations, skew polynomials) of pseudo-linear algebra in the context of computer algebra. In its most general form, a pseudo-linear system of size n is a system

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

where $\mathbf{y}$ is a column-vector of n unknowns, $M \in \mathbb{M}_n(K)$ where K is a field, ϕ is an automorphism over K and δ is a pseudo-derivation with respect to ϕ, which means $\delta(a+b) = \delta(a) + \delta(b)$ and $\delta(a\,b) = \phi(a)\delta(b) + \delta(a)b$ for all $a, b \in K$.

Pseudo-linear systems form a large class of linear functional systems including the usual differential, difference and also q-difference systems: $\mathbf{y}(q\,x) = A(x)\mathbf{y}(x)$. When specialised to these particular types of systems, efficient methods have been designed for treating local problems (i.e. around a *singularity*) such as:

- Construction of formal solutions [25, 37, 42, 45, 91, 93].
- Formal reduction of systems [3, 15, 16, 41, 43, 67, 78].

There are also several algorithms treating global problems such as:

- Computation of closed-form solutions: rational solutions [5, 7, 20], hypergeometric solutions [11, 54, 84], liouvillian solutions [60, 88]
- Decomposition and factorization of systems [17, 21, 37].

Nevertheless, algorithms handling directly general pseudo-linear systems have been less elaborated. In particular, Barkatou treated in [22] the problem of (global) factorization of a pseudo-linear system of the form (1). Later on, Barkatou, Broughton and Pflügel studied in [23] System (1) and introduced a reduction which generalizes the notions of Moser- and super-irreducible forms known for differential and difference systems. The same authors then derived in [24] the structure of regular solutions for general pseudo-linear systems, and developed an algorithm to compute them.

In this thesis, the research carried out concerns the development of symbolic algorithms for studying and solving pseudo-linear systems. Our work excludes the reduction of systems into scalar equations via a cyclic-vector method [51] or any related method [17, 57].

The thesis is split into three essential parts. In the first part, which consists of Chapter 2, we are interested in the local analysis of systems of the form (1). Moser's reduction [18, 41, 78] and super-reduction algorithms [15, 16, 41, 67] have been proved to be relevant for the local study of differential and difference systems. These algorithms provide useful information on the local invariants at a *singular* point such as the nature of the singularity [78], and the integer slopes of the corresponding *Newton polygon* [16, 19, 26, 73, 86]. They are also used to compute formal solutions [19, 25, 37] as well as closed-form solutions such as rational solutions [7, 20], exponential solutions [84] and hypergeometric solutions [44]. However, solving some of these problems requires a weaker form than super-irreducible forms. This weaker form is called *simple form* and it has been first introduced by Barkatou in [20]. 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 generalised in [23, 24, 49] in the pseudo-linear setting and a generic algorithm for computing super-irreducible forms has been given. The method proposed in [24, 49] to construct a simple form requires to compute first a super-irreducible form. The first direct (that is, without recourse to super-reduction) algorithm for computing simple forms has been developed in [27, 59] for differential systems and then in [28] for difference systems. However, no results seem to be published on q-difference systems and the unifying view for the pseudo-linear case had not been considered. In this spirit, the first main contribution of the present thesis is to prove that the algorithms developed in [27, 28] for computing simple forms can be extended to handle more general pseudo-linear systems. In particular, this shall provide the first method available for q-difference systems.

In the second part of this thesis, which consists of Chapters 3 and 4, we are concerned with the problem of computing closed-form solutions of pseudo-linear systems. In particular, we shall present two new recursive algorithms for computing respectively *rational* and *hypergeometric* solutions of a *partial pseudo-linear system* of the form:

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}) = 0, \\ \quad \vdots \\ L_m(\mathbf{y}) := \delta_m(\mathbf{y}) - M_m\,\phi_m(\mathbf{y}) = 0, \end{cases} \tag{2}$$

where for all $i = 1, \ldots, m$, M_i is a square matrix with rational function entries in m variables $x_1, \ldots, x_m$, ϕ_i is an automorphism acting on x_i and δ_i is a pseudo-derivation with respect to ϕ_i such that for all $j \neq i$, x_j is a constant with respect to ϕ_i and δ_i, i.e., $\phi_i(x_j) = x_j$ and $\delta_i(x_j) = 0$. Here $L_i := I_n\,\delta_i - M_i\,\phi_i$ denotes the matrix pseudo-linear operator associated to the i^{th} system of (2). One underlying motivation for developing such algorithms is that many special (transcendental) functions satisfy such partial pseudo-linear systems. We can think for instance of Hermite, Legendre, Bessel or Tchebychev polynomials which satisfy both a system of differential equations and a system of difference equations. Partial pseudo-linear systems have already been considered and studied in the literature. In particular, an algorithm for computing rational solutions of integrable connections (i.e., the case of System (2) with m differential systems) is developed in [29] and an algorithm for computing hyperexponential solutions of systems over Laurent-Ore algebras is proposed in [74]. Also, in [46, 60, 75, 95], the authors study different issues concerning partial pseudo-linear systems.
Our two recursive algorithms require, in particular, an algorithm for computing solutions of one sole pseudo-linear system of the form (1). Therefore, before considering the case of a partial pseudo-linear system with an arbitrary order m, we first concentrate on the case of a single pseudo-linear system, i.e., $m = 1$ in System (2). For the case $m = 1$, the computation of rational and hypergeometric solutions, and other kind of closed form solutions (such as polynomial, Liouvillian, etc...) of linear functional systems has been widely studied in the particular cases of differential and (q-)difference systems: see, for instance, [5, 7, 12, 13, 20, 54, 84, 88]. However, a unified approach handling pseudo-linear

systems has not been introduced. In this spirit, we shall also present an efficient algorithm for computing rational solutions of a general pseudo-linear system of the form (1).

Besides the theoretical results elaborated in this thesis, a further important contribution arises in the implementation in Maple of all the algorithms presented in the different chapters. All the implementations are gathered and incarnated as internal procedures in our new Maple package PseudoLinearSystems [32]. Whilst some existing packages such as Isolde [39], LinearFunctionalSystems[1], LREtools[2] and QDifferenceEquations[3] are dedicated to the study of individual differential, difference and q-difference systems, the PseudoLinearSystems package is dedicated to the study of more general pseudo-linear systems. The third and last part of this thesis, which consists in Chapter 5, hence concerns the demonstration of the use of the main procedures contained in our package. Moreover, we shall also present throughout this thesis results of some experiments comparing our implementations to those performing same tasks in the Isolde and LinearFunctionalSystems packages. All experiments were carried out using Maple 2017 on a Mac PC with a 2.3 GHz Intel Core i5 processor and a 16 Go 2133 MHz LPDDR3 memory.

The main contributions of this thesis can be summarised as follows:

1. A “direct” and generic algorithm for computing simple forms of general pseudo-linear systems with power series coefficients.

2. A unified and efficient algorithm to compute rational solutions of general pseudo-linear systems with rational functions coefficients.

3. Two new efficient algorithms for computing respectively rational and hypergeometric solutions of partial pseudo-linear systems with rational functions coefficients and arbitrary number of variables.

4. The implementation of these algorithms in the computer algebra software Maple.

The content of the thesis is organised as follows. In Chapter 1 we set our mathematical framework by presenting some basic definitions and results from pseudo-linear algebra [23, 24, 48, 49] that will be used in the following chapters. The remaining four chapters contain our contributions. In the sequel, we briefly describe the material of each of them.

[1] https://www.maplesoft.com/support/help/Maple/view.aspx?path=LinearFunctionalSystems&cid=301

[2] https://www.maplesoft.com/support/help/Maple/view.aspx?path=LREtools&cid=284

[3] https://www.maplesoft.com/support/help/Maple/view.aspx?path=QDifferenceEquations&cid=282

Chapter 2: On Simple Forms of Pseudo-Linear Systems and Their Computations

This chapter constitutes the subject of the first half of the published paper [30] in collaboration with M. A. Barkatou and T. Cluzeau.

Let C be a field of characteristic zero and $K = C((t))$ equipped with the t-adic valuation ν. We consider a pseudo-linear system of the form

$$A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{3}$$

where A, $B \in \mathbb{M}_n(C[[t]])$ such that $\det(A) \neq 0$, ϕ is a C-automorphism over K such that it preserves the valuation, i.e., $\nu(\phi(f)) = \nu(f)$ for all $f \in K$, and δ is a non-zero pseudo-derivation with respect to ϕ. The matrices A and B admit unique t-adic expansions

$$A(t) = \sum_{i \geq 0} A_i t^i, \quad B(t) = \sum_{i \geq 0} B_i t^i.$$

The *leading matrix pencil* of a pseudo-linear system (3) is defined as the matrix polynomial $L_\lambda = A_0\,\lambda + B_0$, where A_0 and B_0 are the constant terms in the t-adic expansions of A and B. System (3) is said to be *simple* if $\det(L_\lambda) \neq 0$ (as a polynomial in λ). Otherwise, it is said to be non-simple.

In this chapter, we present a direct algorithm to compute a simple form (in other words, an *equivalent* simple system) of any pseudo-linear system of the form (3). By direct we mean without computing a super-irreducible form as it is done in [24, 49]. The algorithm is recursive and it follows the ideas of the algorithms developed in [27, 28] for purely differential and difference systems. When System (3) is not simple, then we have $\nu(\det(A)) > 0$ (otherwise, A_0 would be invertible and hence $\det(A_0\,\lambda + B) \neq 0$). The principle of the algorithm then consists in computing iteratively equivalent pseudo-linear systems $A^{(i)}\,\delta(\mathbf{y}) + B^{(i)}\,\phi(\mathbf{y}) = 0$, $i \geq 1$ satisfying $\nu(\det(A^{(i+1)})) < \nu(\det(A^{(i)}))$ with $A^{(0)} = A$. Doing so, we are guaranteed to find a simple system at some stage of the recusrsion.

Moreover, we show how we can use simple forms as an essential tool for computing important local data: *indicial equation, regular solutions* and *k-simple forms*. Finally, we derive a new method for computing a super-irreducible form of a pseudo-linear system based on successive computations of simple forms.

Chapter 3: On Rational Solutions of First Order Pseudo-Linear Systems

This chapter constitutes the subject of the first half of the published paper [33] in collaboration with M. A. Barkatou and T. Cluzeau.

Let C be a field of characteristic zero and $F = C(x)$. We are interested in the problem of computing rational solutions of a pseudo-linear system of the form

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \tag{4}$$

where $M \in \mathbb{M}_n(F)$, ϕ is a C-automorphism over F and δ is a pseudo-derivation with respect to ϕ. We follow the same strategy proposed in [5, 20, 21] for the differential and (q-)difference cases. We first compute a so called *universal denominator* [2], namely, a polynomial that is a multiple of the denominator of any rational solution. Then, by performing a suitable change of dependent variables in (4), we will be reduced to computing polynomial solutions of a system of the same type.

In the sequel, we present a unified algorithm for computing a universal denominator for all rational solutions of a pseudo-linear system. We will see that in the case ϕ is not the identity map, a universal denominator is composed of two parts: what we call the *ϕ-fixed part* and the *non ϕ-fixed part*. On one hand, the ϕ-fixed part can be computed using simple forms. On the other hand, following the ideas of [77] (see also [9, 72]), we propose an efficient algorithm for computing the non ϕ-fixed part. Polynomial solutions are computed by adapting the ideas in [20, 21, 24]. We provide details on how this can be efficiently done for pseudo-linear systems.

Chapter 4: On Rational and Hypergeometric Solutions of Partial Pseudo-Linear Systems

This chapter constitutes the subjects of the second halves of the published papers [30, 33] in collaboration with M. A. Barkatou and T. Cluzeau.

Let C be a field of characteristic zero and $K = C(x_1, \ldots, x_m)$. The object of study in this chapter is a partial pseudo-linear system (2) with $M_i \in \mathbb{M}_n(F)$ for all $i = 1, \ldots, m$. We assume that (2) is integrable, i.e., it satisfies the *integrability conditions*

$$[L_i, L_j] := L_i \circ L_j - L_j \circ L_i = 0, \quad \forall i, j = 1, \ldots, m.$$

The main contribution of this chapter consists in two new efficient algorithms for respectively computing rational and hypergeometric solutions of an integrable system (2). In both algorithms, the strategy proceeds by recursion by considering one by one each system $L_i(\mathbf{y}) = 0$ appearing in (2) as a system in one variable x_i with the other variables viewed as transcendental constants. In particular, we shall prove that in both strategies, we can reduce the number m of variables and maybe the size n of the matrices before applying recursion and considering the second system.

We also provide some remarks concerning the implementation of both algorithms.

Chapter 5: PseudoLinearSystems: A Maple Package for Studying Systems of Pseudo-linear Equations

In this final chapter, we demonstrate the use of several important procedures contained in our Maple package PseudoLinearSystems [32]. This includes, in particular, a procedure to compute a simple form of a pseudo-linear system (3), as well as procedures to compute rational and hypergeometric solutions of a partial pseudo-linear system (2). The package is free available online and a manual for downloading and installing the package, as well as Maple examples covering several types of pseudo-linear systems, are provided on the webpage:

http://www.unilim.fr/pages_perso/ali.el-hajj/PseudoLinearSystems.html

Chapter 1

Basics on Pseudo-linear Systems

In this chapter, we prepare the mathematical framework in which the work of the thesis is based on. This mathematical framework is pseudo-linear algebra which concerns the study of pseudo-linear systems: a large class of linear functional systems including the usual differential, difference and q-difference systems. In the same line as the authors in [23, 24, 48], we define the basic objects that form our framework.

Throughout this chapter, we let K be a commutative field of characteristic zero and ϕ an automorphism of K.

1.1 Pseudo-derivations and $\phi\delta$-fields

Definition 1.1. *Given a commutative field K and an automorphism ϕ over K. A pseudo-derivation with respect to ϕ, or a ϕ-derivation, is any map $\delta : K \to K$ satisfying* Leibniz rule*: for all $a, b \in K$,*

$$\delta(a+b) = \delta(a) + \delta(b),$$

and

$$\delta(a\,b) = \phi(a)\delta(b) + \delta(a)b. \tag{1.1}$$

Example 1.1. *If $\phi = \mathrm{id}_K$, then δ is a standard derivation. Otherwise if $\phi \neq \mathrm{id}_K$, then for any $\gamma \in K$, the map $\delta = \gamma(\mathrm{id}_K - \phi)$ given by $\delta(a) = \gamma(a - \phi(a))$ is a pseudo-derivation with respect to ϕ. Indeed, we have for all $a, b \in K$:*

$$\delta(a+b) = \gamma\,(a+b-\phi(a+b)) = \gamma\,(a-\phi(a)) + \gamma\,(b-\phi(b)) = \delta(a) + \delta(b),$$

and

$$\delta(a\,b) = \gamma(a\,b - \phi(a\,b)) = \phi(a)\gamma(b-\phi(b)) + \gamma(a-\phi(a))b = \phi(a)\delta(b) + \delta(a)b.$$

The above example covers all the possible pseudo-derivations over a commutative field. Indeed, we have the following result:

Lemma 1.1. *([48, Lemma 1]). Let K be a commutative field, ϕ an automorphism over K and δ a pseudo-derivation w.r.t. ϕ. If $\phi \neq \mathrm{id}_K$ then δ has to be of the form*

$$\delta = \gamma\,(\mathrm{id}_K - \phi),$$

for some $\gamma \in K$.

Proof. Since K is commutative, then we have $\delta(a\,b) = \delta(b\,a)$ for all $a, b \in K$. Using Equation (1.1) on both sides yields, after rearranging :

$$(a - \phi(a))\,\delta(b) = (b - \phi(b))\,\delta(a). \tag{1.2}$$

Now if $\phi \neq \mathrm{id}_K$, then there exists $a \in K$ such that $\phi(a) \neq a$. Let $\gamma = \delta(a)/(a - \phi(a))$, then it follows from (1.2) that $\delta(b) = \gamma\,(b - \phi(b))$ for all $b \in K$. $\square$

Remark 1.1. *With the notations of Lemma 1.1, if $\delta \neq 0$ then $\delta = \gamma\,(\mathrm{id}_K - \phi)$ for some $\gamma \in K^*$.*

Definition 1.2. *Let K be a commutative field, ϕ an automorphism over K and δ a pseudo-derivation w.r.t. ϕ. We refer to the triplet (K, ϕ, δ) as a $\phi\delta$*-field.

Example 1.2. *The familiar differential, difference and q-difference fields can be expressed as $\phi\delta$-fields in the following way:*

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

Definition 1.3. *Given a $\phi\delta$-field (K, ϕ, δ), the subfield $\mathcal{C}_K \subset K$ containing all elements c in K that satisfy $\phi(c) = c$ and $\delta(c) = 0$ is called the field of constants of (K, ϕ, δ).*

Remark 1.2. *When $\phi \neq \mathrm{id}_K$, then $\phi(c) = c \iff \delta(c) = 0$ for any $c \in K$.*

We note that the operations on matrices (vectors) commute with ϕ and for two matrices M and N, one has

$$\delta(M + N) = \delta(M) + \delta(N),$$

and

$$\delta(MN) = \delta(M)\phi(N) + M\delta(N) = \phi(M)\delta(N) + \delta(M)N.$$

1.2 Pseudo-linear systems

We present in this section the notion of pseudo-linear systems as a unified class for expressing common types of linear functional systems such as linear differential and (q-)difference systems.

Definition 1.4. *A pseudo-linear system of size n over a $\phi\delta$-field (K, ϕ, δ) is a system of the form*

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \tag{1.3}$$

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

Definition 1.5. *A solution of pseudo-linear system (1.3) over K is a vector $\mathbf{y} \in K^n$ such that $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$.*

Remark 1.3. *The set of solutions of System (1.3) over K is a vector space over $\mathcal{C}_K$ of dimension at most n (see [22]).*

For $\phi \neq \mathrm{id}_K$, previous authors considered pseudo-linear systems of the form

$$\phi(\mathbf{y}) = M\,\mathbf{y}. \tag{1.4}$$

In particular, Harris [62] studied difference systems where $\phi : x \mapsto x+1$, while Adams [14] and Trjitzinsky [91] explored the q-difference case $\phi : x \mapsto q\,x$. A system of the form (1.4) is what we call a *ϕ-system* in this thesis, and such a system will be considered later in Chapter 3. Moreover, the authors in [25, 26, 28] found it convenient to consider linear difference systems of the form

$$\delta(\mathbf{y}) = M\,\mathbf{y}. \tag{1.5}$$

In this thesis, we see System (1.3) the most general form to consider as introduced by Jacobson [69], and the most adaptable for (most of) the algorithms developed later. Anyhow, the conversion between systems (1.4), (1.5) and (1.3) could be easily carried out. Indeed (recall that $\delta = \gamma(\mathrm{id}_K - \phi)$ for some $\gamma \in K^*$):

1. System (1.4) can be rewritten as

$$\mathbf{y} = \phi^{-1}(M)\,\phi^{-1}(\mathbf{y}). \tag{1.6}$$

   Let $\widetilde{\phi} = \phi^{-1}$ and $\widetilde{\delta}$ be a new pseudo-derivation w.r.t. $\widetilde{\phi}$ defined as $\widetilde{\delta} = \mathrm{id}_K - \widetilde{\phi}$. One thus has $\mathbf{y} = \widetilde{\delta}(\mathbf{y}) + \widetilde{\phi}(\mathbf{y})$. Substituting this for $\mathbf{y}$ in the left hand side of (1.6) yields, after rearranging, the system

$$\widetilde{\delta}(\mathbf{y}) = N\,\widetilde{\phi}(\mathbf{y}),$$

   where $N = \widetilde{\phi}(M) - I_n$.

2. System (1.5) can be rewritten as

$$\gamma(\mathbf{y} - \phi(\mathbf{y})) = M\,\mathbf{y}.$$

   Applying ϕ^{-1} on both sides yields the system $\widetilde{\delta}(\mathbf{y}) = N\,\widetilde{\phi}(\mathbf{y})$, where $\widetilde{\phi} = \phi^{-1}$, $\widetilde{\delta} = \widetilde{\phi}(\gamma)(\widetilde{\phi} - \mathrm{id}_K)$ and $N = \widetilde{\phi}(M)$.

3. Given a system of the form $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$. Applying ϕ^{-1} on both sides yields the system

$$\widetilde{\delta}(\mathbf{y}) = N\,\mathbf{y},$$

   where $\widetilde{\delta} = \widetilde{\phi}(\gamma)(\widetilde{\phi} - \mathrm{id}_K)$, $\widetilde{\phi} = \phi^{-1}$ and $N = \widetilde{\phi}(M)$.

4. Given a system of the form $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$. Applying ϕ^{-1} on both sides yields, after rearranging, the system

$$\widetilde{\phi}(\mathbf{y}) = N\,\mathbf{y},$$

   where $\widetilde{\phi} = \phi^{-1}$ and $N = \widetilde{\phi}(\gamma^{-1}\,M + I_n)$. Note that a system of the form (1.3) can be also written as

$$\phi(\mathbf{y}) = N\,\mathbf{y},$$

with $N = (\gamma^{-1} M + I_n)^{-1}$ provided that this matrix exists.

Consequently, after looking at the last sentence of Item 4, when we consider a pseudo-linear system (1.3) with $\phi \neq \mathrm{id}_K$ and $\delta = \gamma\,(\mathrm{id}_K - \phi)$ for some $\gamma \in K^*$, we will always assume that the matrix $M + \gamma\, I_n$ is invertible. In this case, System (1.3) is called *fully integrable*. Note that for a ϕ-system (1.4), being fully integrable means that the matrix of the system is invertible.

Remark 1.4. *Considering solutions over a suitable field extension F of K, every fully integrable system admits a solution space of dimension n over $\mathcal{C}_F = \mathcal{C}_K$. We refer to [75] where a notion of Picard-Vessiot extension is introduced in the pseudo-linear setting. Moreover, from [26, Proposition 2], every ϕ-system $\phi(\mathbf{y}) = B\,\mathbf{y}$ can be effectively reduced to a ϕ-system of smaller size with either B invertible (i.e., we have an equivalent fully integrable system) or $B = 0$. For instance, the system*

$$\delta(\mathbf{y}) = \begin{pmatrix} -1 & 1 & \cdots & 1 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -1 \end{pmatrix} \phi(\mathbf{y}), \quad \phi \neq \mathrm{id}_K, \quad \delta = \mathrm{id}_K - \phi,$$

is not fully integrable. It can be reduced to the scalar pseudo-linear equation $\delta(y) = -\phi(y)$ which is equivalent to $y = 0$.

Given a pseudo-linear system of the form (1.3). For a matrix $T \in \mathrm{GL}_n(K)$, performing the change of variable $\mathbf{y} = T\,\mathbf{z}$ yields a new pseudo-linear system $\delta(\mathbf{z}) = N\,\phi(\mathbf{z})$ where $N \in \mathbb{M}_n(K)$ is given by

$$N = T^{-1}(M\,\phi(T) - \delta(T)).$$

Consequently, the notion of equivalence of two pseudo-linear systems of the form (1.3) is given by the following definition:

Definition 1.6. *Two pseudo-linear systems $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$ and $\delta(\mathbf{y}) = N\,\phi(\mathbf{y})$ over a $\phi\delta$-field (K, ϕ, δ), are said to be equivalent if there exists an invertible matrix $T \in \mathrm{GL}_n(K)$ such that*

$$N = T^{-1}\left(M\,\phi(T) - \delta(T)\right). \tag{1.7}$$

We now define the notion of partial pseudo-linear systems. Such systems are the object of study in Chapter 4. For $i = 1, \ldots, m$, let ϕ_i be an automorphism over K and δ_i be a pseudo-derivation with respect to ϕ_i. We shall also refer to $(K, \{\phi_i, \delta_i\}_{1 \leq i \leq m})$ as a $\phi\delta$-field. The field of constants of $(K, \{\phi_i, \delta_i\}_{1 \leq i \leq m})$ is the subfield $\mathcal{C}_K$ of K containing all elements c in K that satisfy $\phi_i(c) = c$ and $\delta_i(c) = 0$ for all $i \in \{1, \ldots, m\}$. In general, one has

$$\mathcal{C}_K = \bigcap_{i=1}^{m} \mathcal{C}_K^i,$$

where $\mathcal{C}_K^i$ denotes the field of constants of the (particular) $\phi\delta$-field (K, ϕ_i, δ_i).

Definition 1.7. *Given a $\phi\delta$-field $(K, \{\phi_i, \delta_i\}_{1\leq i\leq m})$ such that $\mathcal{C}_K^i \neq \mathcal{C}_K^j$ for all $i \neq j$. A* partial pseudo-linear system *over $(K, \{\phi_i, \delta_i\}_{1\leq i\leq m})$ of size n in m variables, is a system of the form*

$$\begin{cases} \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}) = 0, \\ \quad\vdots \\ \delta_m(\mathbf{y}) - M_m\,\phi_m(\mathbf{y}) = 0, \end{cases}$$

where $\mathbf{y}$ is a vector of n unknowns and $M_i \in \mathbb{M}_n(K)$ for all $i \in \{1, \ldots, m\}$.

1.3 Local $\phi\delta$-fields

In many situations, when looking for solutions (rational, series, ...) of a pseudo-linear system, we may need some information at a "singularity". In this thesis, the notion of a singularity of pseudo-linear systems differs from that in the differential and difference cases. Recall that for a first order differential system $\mathbf{y}'(x) = A(x)\mathbf{y}(x)$, it is well known (see e.g. [20, 84]) that the finite singularities are exactly the singularities of the matrix A, i.e., the zeros of $\mathrm{den}(A)$. On the other hand, for a difference system $\phi(\mathbf{y}(x)) = A(x)\mathbf{y}(x)$ where A is an invertible matrix and $\phi : x \mapsto x + 1$, the finite singularities are among the *shifts* of the zeros of $\mathrm{den}(A)$ and $\mathrm{den}(A^*)$, where $A^* = \phi^{-1}(A^{-1})$ (see [36] for more details).

Definition 1.8. *Given a $\phi\delta$-field (K, ϕ, δ) where ϕ is an automorphism over K acting on a variable x, and δ is a ϕ-derivation. Denote by $\mathcal{C}_K$ the field of constants of (K, ϕ, δ). A point x_0 in $\overline{\mathcal{C}_K}$ is said to be ϕ-*fixed*, or* fixed by *ϕ, if $x - x_0$ and $\phi^j(x - x_0)$ divide each other for some $j \in \mathbb{Z}^*$.*

Definition 1.9. *The singularities of a pseudo-linear system (1.3) defined over a $\phi\delta$-field (K, ϕ, δ) are all ϕ-fixed points, and the point at infinity.*

Example 1.3. *Given a a pseudo-linear system (1.3) over a $\phi\delta$-field (K, ϕ, δ). Denote by $\mathcal{C}_K$ the field of constants of (K, ϕ, δ).*

1. *In the differential case where $\phi = \mathrm{id}_K$, all the points in $\overline{\mathcal{C}_K}$ are ϕ-fixed.*

2. *In the difference case where $\phi : x \mapsto x + r$ with $r \in \mathcal{C}_K^*$, there are no ϕ-fixed points. The only singularity here is ∞.*

3. *In the q-difference case $\phi : x \mapsto q\,x$ where $q \in \mathcal{C}_K^*$ is not a root of unity, the point 0 is the only ϕ-fixed point. The only singularities here are 0 and ∞.*

4. *Let ϕ be the automorphism defined as $\phi : x \mapsto q\,x + r$ where $q \in \mathcal{C}_K^*$ is not a root of unity and $r \in \mathcal{C}_K$. The only ϕ-fixed point is $x_0 = \frac{r}{1-q}$ (see Proposition 3.3 in Chapter 3). The only singularities here are x_0 and ∞.*

Definition 1.10. *A* local $\phi\delta$-field *(K, ϕ, δ) is a $\phi\delta$-field equipped with a discrete valuation $\nu : K \to \mathbb{Z} \cup \{\infty\}$.*

Recall that for a local $\phi\delta$-field (K, ϕ, δ) equipped with a valuation ν, the following properties hold: for $a, b \in K$ one has:

1. $\nu(a) = +\infty \iff a = 0$.
2. $\nu(a\,b) = \nu(a) + \nu(b)$.
3. $\nu(a + b) \geq \min(\nu(a), \nu(b))$, and equality holds if $\nu(a) \neq \nu(b)$.

We introduce now some terminology from [23] that could help to understand the concept of local $\phi\delta$-fields. The *valuation ring* of a local $\phi\delta$-field (K, ϕ, δ) is defined as

$$\mathcal{O} = \{\, a \in K \,;\, \nu(a) \geq 0 \,\}.$$

The set $\mathfrak{M} = \{\, a \in K \,;\, \nu(a) > 0 \,\}$ coincides with the set of non-invertible elements of $\mathcal{O}$, and it is the unique maximal ideal of $\mathcal{O}$. The field $\mathcal{R} = \mathcal{O} \setminus \mathfrak{M}$ is called the *residue field* of (K, ϕ, δ). We denote by π the canonical homomorphism from $\mathcal{O}$ into $\mathcal{R}$.

Example 1.4.

1. *Let $K = \mathbb{C}((x))$, $\phi = \mathrm{id}_K$ and $\delta = \frac{d}{dx}$. The field (K, ϕ, δ) is a local $\phi\delta$-field equipped with the x-adic valuation ν defined by $\nu(f) = m$ if $f = x^m(f_0 + f_1\,x + \ldots)$, $f_0 \neq 0$ and $\nu(0) := \infty$. The valuation ring is $\mathcal{O} = \mathbb{C}[[x]]$. The residue field can be identified with $\mathbb{C}$ and $\pi(a) = a(0)$ for $a \in \mathcal{O}$.*

2. *Let $K = \mathbb{C}((x^{-1}))$, ϕ be the automorphism defined as $\phi : x \mapsto x + r$ where $r \in \mathbb{C}^*$, and $\delta = \mathrm{id}_K - \phi$. Let $t = x^{-1}$. The field (K, ϕ, δ) is a local $\phi\delta$-field equipped with the t-adic valuation defined by $\nu(f) = m$ if $f = t^m(f_0 + f_1\,t + \ldots)$, $f_0 \neq 0$ and $\nu(0) := \infty$. The valuation ring is $\mathcal{O} = \mathbb{C}[[x^{-1}]]$. The residue field can be identified with $\mathbb{C}$ and $\pi(a) = a(\infty)$ for $a \in \mathcal{O}$.*

Definition 1.11. *Let (K, ϕ, δ) be a local $\phi\delta$-field equipped with a valuation ν. Denote by $\mathcal{O}$ its valuation ring. An element $t \in \mathcal{O}$ is said to be a* local parameter *of K if $\nu(t) = 1$.*

Let (K, ϕ, δ) be a local $\phi\delta$-field equipped with a discrete valuation ν, and denote by $\mathcal{R}$ its residue field. We fix a local parameter t of K. Every element $f \in K$ can be uniquely expanded as

$$f(t) = t^{\nu(f)} \sum_{i=0}^{+\infty} f_i\,t^i,$$

where the f_i's are in $\mathcal{R}$ and $f_0 \neq 0$. The definition of the valuation can be extended to a matrix $A \in \mathbb{M}_{m\times n}(K)$ by $\nu(A) = \min\{\nu(A(i,j))\,;\, 1 \le i \le m,\, 1 \le j \le n\}$ where $A(i,j)$ denotes the (i,j)th entry of A. Any non zero matrix $A \in \mathbb{M}_{m\times n}(K)$ can be uniquely expanded as

$$A(t) = t^{\nu(A)} \sum_{i=0}^{+\infty} A_i\, t^i,$$

where the A_i's are matrices with entries in $\mathcal{R}$ and $A_0 \neq 0$.

Definition 1.12. *Given a local $\phi\delta$-field (K, ϕ, δ). We define the* degree *of δ as*

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

Note that δ is continuous if and only if $\omega(\delta) > -\infty$.

Lemma 1.2. *([23, Lemma 3.2]). Given a local $\phi\delta$-field (K, ϕ, δ) and t a local parameter of K. If $\delta \neq 0$ and $\omega(\delta) > -\infty$. Then*

$$\omega(\delta) = \nu(t^{-1}\delta(t)).$$

Throughout this thesis, when talking about local $\phi\delta$-fields, we assume that $\delta \neq 0$ and $\omega(\delta) > -\infty$, and that the automorphism ϕ is an isometry with respect to the valuation, which means that ϕ preserves the valuation:

$$\nu(\phi(a)) = \nu(a), \quad \forall a \in K.$$

Using the latter assumption on ϕ, we can define the following:

Definition 1.13. *Let (K, ϕ, δ) be a local $\phi\delta$-field and $\mathcal{R}$ be its residue field. Let t be a local parameter of K. We denote by c and d the two elements in $\mathcal{R}^*$ satisfying*

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

where ω is the degree of δ, and inductively for $h \in \mathbb{N}_+$:

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

where $[h]_c$ is defined by:

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

Example 1.5.

1. *Let $K = \mathbb{C}((x))$, $\phi = \mathrm{id}_K$, $\delta = \frac{d}{dx}$. The field (K, ϕ, δ) is a local $\phi\delta$-field equipped*

with the t-adic valuation, where $t = x$ is a local parameter of K. We have $\omega(\delta) = -1$ and $c = d = 1$.

2. Let $K = \mathbb{C}((x^{-1}))$, ϕ be the $\mathbb{C}$-automorphism over K defined by $\phi : x \mapsto x - 1$, and $\delta = \mathrm{id}_K - \phi$ be a pseudo-derivation w.r.t. ϕ. The field (K, ϕ, δ) is a local $\phi\delta$-field equipped with the t-adic valuation, where $t = x^{-1}$ is a local parameter of K. We have

$$\omega(\delta) = \nu(t^{-1}\delta(t)) = \nu(-(x-1)^{-1}) = 1.$$

Moreover, we have $c = 1$ and $d = -1$.

3. Let ϕ be the $\mathbb{C}$-automorphism defined by $\phi : x \mapsto q\,x + r$ such that $q \in \mathbb{C} \setminus \{0, 1\}$ and $r \in \mathbb{C}$. Let $K = \mathbb{C}((x - x_0))$ where $x_0 = \frac{r}{1-q}$ and $\delta = \mathrm{id}_K - \phi$ be a pseudo-derivation w.r.t. ϕ. The field (K, ϕ, δ) is a local $\phi\delta$-field equipped with the t-adic valuation, where $t = x - x_0$ is a local parameter of K. We have

$$\omega = \omega(\delta) = \nu(t^{-1}\delta(t)) = \nu\left((x - x_0)^{-1}(x - x_0 - \phi(x - x_0))\right) = \nu(1 - q) = 0.$$

Moreover, we have $c = q$ and $d = 1 - q$.

1.4 Local pseudo-linear systems

In the rest of this chapter, we let (K, ϕ, δ) be a local $\phi\delta$-field. We denote by $\mathcal{O}$, respectively $\mathcal{R}$, its valuations ring, respectively residue field, and we fix a local parameter t of K.

Given a pseudo-linear system

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \tag{1.8}$$

defined ove (K, ϕ, δ). The matrix $M \in \mathbb{M}_n(K)$ can be written as $M = t^{\omega - p}(M_0 + M_1\,t + \ldots)$ where the M_i's are matrices with entries in $\mathcal{R}$ with $M_0 \neq 0$, $p \in \mathbb{N}$ is called the *Poincaré rank* of (1.8) and $\omega \in \mathbb{Z}$ is the degree of the ϕ-derivation δ. Multiplying (1.8) by $t^{p-\omega}$ on both sides yields the system

$$t^{p-\omega}\delta(\mathbf{y}) = N\,\phi(\mathbf{y}), \tag{1.9}$$

where $N \in M_n(\mathcal{O})$.

Definition 1.14. *A* local pseudo-linear system *of size n over a local $\phi\delta$-field (K, ϕ, δ) is a system of the form (1.9).*

A local pseudo-linear system of the form (1.9) is obtained when we want to study a pseudo-linear system of the form (1.3) in the neighbourhood of a given singularity x_0. For instance, System (1.3) defined over $\mathbb{C}(x)$ can be written as local pseudo-linear system by means of the map $x \to t + x_0$ if $x_0 \neq \infty$ or $x \to t^{-1}$ otherwise, and by embedding $\mathbb{C}(x)$ into $\mathbb{C}((t))$.

Example 1.6.

1. Let ϕ be the identity map over $\mathbb{C}(x)$ and $\delta = \frac{d}{dx}$. Consider the following differential system defined over $(\mathbb{C}(x), \phi, \delta)$:

$$\delta(\mathbf{y}) = \begin{bmatrix} \dfrac{x^2-1}{x\,(x^2+1)} & 0 \\ \dfrac{1}{x^2+1} & -\dfrac{1}{x} \end{bmatrix} \mathbf{y}.$$

We want to study the system near the ϕ-fixed (singular) point $x_0 = 0$. We introduce the local parameter $t = x$. The idea is to work with $K = \mathbb{C}((t))$ equipped with the t-adic valuation. The degree of δ is thus $\omega = -1$ and the system can be written in the local form (1.9) over (K, ϕ, δ) as

$$t\,\delta(\mathbf{y}) = \begin{bmatrix} \dfrac{t^2-1}{t^2+1} & 0 \\ \dfrac{t}{t^2+1} & -1 \end{bmatrix} \mathbf{y},$$

with Poincaré rank $p = 0$.

2. Let ϕ be an automorphism over $\mathbb{C}(x)$ defined as $\phi : x \mapsto x - 1$, and $\delta = \mathrm{id}_{\mathrm{K}} - \phi$ be a ϕ-derivation. Consider the following system defined over $(\mathbb{C}(x), \phi, \delta)$:

$$\delta(\mathbf{y}) = \begin{bmatrix} x+1 & \dfrac{1}{x} \\ x^2 & 1 \end{bmatrix} \phi(\mathbf{y}),$$

We want to study the system near the only singularity ∞. We introduce the local parameter $t = x^{-1}$ and we work now with $K = \mathbb{C}((t))$ equipped with the t-adic valuation. The degree of δ is thus $\omega = 1$ and the system can be written in the local form (1.9) over (K, ϕ, δ) as

$$t^2\delta(\mathbf{y}) = \begin{bmatrix} t+t^2 & t^3 \\ 1 & t^2 \end{bmatrix} \phi(\mathbf{y}),$$

with Poincaré rank $p = 3$.

A local pseudo-linear system (1.9) can be written in the form

$$A\,\tilde{\delta}(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{1.10}$$

where $\tilde{\delta} = t^{-\omega}\delta$,

$$A = \operatorname{diag}(t^{\alpha_1}, \dots, t^{\alpha_n}), \quad \alpha_i = -\min\{0, \nu(N(i,.)) - p\},$$

and

$$B = -t^{-p} A\, N \in \mathbb{M}_n(\mathcal{O}).$$

Here $N(i,.)$ denotes the i^{th} row of N. A system of the form (1.10) will be the object of study in Chapter 2 and we shall also call it a local pseudo-linear system. In the sequel, we define the notion of equivalence of two pseudo-linear systems of the form (1.10).

Without any loss of generality, we suppose that $\omega = 0$ and thus have $\tilde{\delta} = \delta$. To any pseudo-linear system (1.10), we associate the *pseudo-linear operator*

$$L = A\,\delta + B\,\phi,$$

acting on K^n. The system is then written as $L(\mathbf{y}) = 0$. Note that, in terms of operators, we have

$$\phi.T = \phi(T)\,\phi, \quad \delta.T = T\,\delta + \delta(T)\,\phi,$$

for all $T \in \mathbb{M}_n(K)$.

Definition 1.15. *Two pseudo-linear operators $L = A\,\delta + B\,\phi$ and $L' = A'\,\delta + B'\,\phi$ are said to be* equivalent *if there exist two invertible matrices $S, T \in \mathrm{GL}_n(K)$ such that $L' = S\,L\,T$, that is:*

$$A' = S\,A\,T, \qquad B' = S\,A\,\delta(T) + S\,B\,\phi(T). \tag{1.11}$$

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

For further reading concerning pseudo-linear algebra, we refer to the introductory work by Jacobson [69], the modern exposition in the context of computer algebra by Bronstein and Petkovšek [48], and other notable works [23, 24, 47, 49, 79].

Chapter 2

On Simple Forms of Pseudo-Linear Systems and their Applications

This chapter constitutes the subject of the first half of the published paper [30] in collaboration with M. A. Barkatou and T. Cluzeau.

2.1 Introduction and motivation

In this chapter, we consider pseudo-linear systems over a local $\phi\delta$-field (K, ϕ, δ). To fix ideas, we let C be a field of characteristic zero and $K = C((t))$ be the field of Laurent series in a variable t with coefficients in C, equipped with the t-adic valuation $\nu : K \to \mathbb{Z} \cup \{\infty\}$ defined by $\nu(f) = m$ if $f = x^m(f_0 + f_1\,x + \ldots)$, $f_0 \neq 0$ and $\nu(0) = \infty$. In our algorithms and implementations, we assume that C is a computable field of characteristic zero. In this chapter, we are merely interested in a (local) pseudo-linear system of the form

$$L(\mathbf{y}) = A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{2.1}$$

where A, $B \in \mathbb{M}_n(C[[t]])$ with $\det(A) \neq 0$. We assume that the C-automorphism ϕ over K is an isometry with respect to ν, i.e., it preserves the valuation: $\nu(\phi(a)) = \nu(a)$, for all $a \in K$, and δ is a non-trivial pseudo-derivation w.r.t. ϕ. The matrices A and B admit unique t-adic expansions

$$A(t) = \sum_{i \geq 0} A_i t^i, \quad B(t) = \sum_{i \geq 0} B_i t^i.$$

Definition 2.1. *The leading matrix pencil of a pseudo-linear system (2.1) is defined as the matrix polynomial $L_\lambda = A_0\,\lambda + B_0$, where A_0 and B_0 are the constant terms in the t-adic expansions of A and B.*

We define the notion of simple form for a pseudo-linear system (2.1) as follows:

Definition 2.2. *We say that System (2.1) is* simple *if $\det(L_\lambda) \neq 0$ (or equivalently, L_λ is non-singular). Otherwise, we say that (2.1) is non-simple. This definition also applies to pseudo-linear operators and to non-homogeneous systems $A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = b$, where $b \in C[[t]]^n$.*

Example 2.1. *The pseudo-linear system*

$$\begin{bmatrix} 1 & 0 \\ 0 & t \end{bmatrix} \delta(\mathbf{y}) + \begin{bmatrix} -1 + 2\,t + O\,(t^2) & 0 \\ -\frac{3}{2} + 3\,t + O\,(t^2) & 0 \end{bmatrix} \phi(\mathbf{y}) = 0, \tag{2.2}$$

is non-simple since its associated leading matrix pencil

$$L_\lambda = \begin{bmatrix} \lambda - 1 & 0 \\ -\frac{3}{2} & 0 \end{bmatrix}$$

is singular.

The need of simple forms is essential in studying pseudo-linear systems. To see why, let us consider, for instance, the problem of computing polynomial solutions of a linear

q-difference system

$$\mathbf{y}(q\,x) = N(x)\,\mathbf{y}(x), \tag{2.3}$$

where $q \in C^*$ is not a root of unity and $N(x)$ is an $n \times n$ matrix with rational functions entries in x and coefficients in C. A polynomial solution of degree $s \in \mathbb{N}$ can be viewed as a local formal solution (at $x = \infty$) of the form

$$\mathbf{y}(x) = \sum_{i \geq 0} \mathbf{y}_i\, x^{-i+s}$$

where $\mathbf{y}_i \in C^n$, $\mathbf{y}_0 \neq 0$ and $\mathbf{y}_i = 0$ for $i > s$. The idea now is to work with $K = C((t))$ where $t = x^{-1}$, and we rewrite System (2.3) as a local pseudo-linear system of the form (2.1) (see Sections 1.2 and 1.4) with $\phi(t) = q\,t$ and $\delta = \mathrm{id}_K - \phi$. We now look for formal solutions of the form $\mathbf{y}(t) = \sum_{i\geq 0} \mathbf{y}_i t^{i+s}$ where $s \in \mathbb{N}$ and $\mathbf{y}_i \in C^n$ with $\mathbf{y}_0 \neq 0$. Replacing $\phi(\mathbf{y})$, $\delta(\mathbf{y})$, A and B by their respective t-adic expansions in System (2.1) and comparing coefficients of the same power of t yields, amongst others, the equation

$$(A_0(q^{-s} - 1) + B_0)\mathbf{y}_0 = 0.$$

This implies that, if a q-difference system (2.3) has a non-zero polynomial solution of degree s, then $(q^{-s} - 1)$ must be a root of $\det(A_0\,\lambda + B_0) = \det(L_\lambda)$. Consequently, the degree of any polynomial solution of (2.3) can be bounded by the largest non-negative integer s such that $(q^{-s}-1)$ is a root of $\det(L_\lambda)$. But, it may happen that $\det(L_\lambda)$ vanishes identically in λ, which means that (2.1) is non-simple. In this case, no useful information can be obtained, unless we are able to compute an equivalent system (see Definition 1.15) which is simple, or in other words, a simple form of (2.1). In this spirit, we present in this chapter an efficient algorithm that allows to compute a simple form of any pseudo-linear system of the form (2.1).

Example 2.2. *The equivalence transformations defined by the two matrices*

$$S(t) = \begin{bmatrix} \frac{1}{t} & -\frac{2}{3\,t} \\ 0 & \frac{1}{t} \end{bmatrix}, \quad T(t) = \begin{bmatrix} t & \frac{2}{3\,t} \\ 0 & 1 \end{bmatrix}$$

reduce (2.2) *to the equivalent simple system*

$$\delta(\mathbf{y}) + \begin{bmatrix} -1 & -\frac{2}{3} \\ -3 + 6\,t + O\left(t^2\right) & -2 + 4\,t + O\left(t^2\right) \end{bmatrix} \phi(\mathbf{y}) = 0, \tag{2.4}$$

with a non-singular leading matrix pencil

$$L'_\lambda = \begin{bmatrix} \lambda - 1 & -\frac{2}{3} \\ -3 & \lambda - 2 \end{bmatrix}.$$

Prior to our work, the existing methods [24, 49] for computing a simple form of a pseudo-linear system require to compute first a super-irreducible form [23]. Motivated by the fact that a simple system is not necessarily super-irreducible, the authors in [27, 59] developed the first direct (that is, without recourse to super-reduction) algorithm for computing simple forms for differential systems and later on in [28] for difference systems. The main contribution of this chapter is to provide a unified and direct algorithm for computing simple forms of general pseudo-linear systems. Consequently, this shall give the first method available for q-difference systems.

The rest of the chapter is organized as follows. Section 2.2 contains our main contribution of this chapter, i.e., an algorithm for computing simple forms for general pseudo-linear systems of the form (2.1). We show that for a given pseudo-linear operator L, we can always compute two invertible matrices S and T in $\mathbb{M}_n(K)$ such that the equivalent operator $S\,L\,T$ is simple. We illustrate our approach with a detailed example and we provide a complexity estimate of the algorithm. Section 2.3 is devoted to the applications of simple forms for the local study of pseudo-linear systems. We show how we can use simple forms as an essential tool for computing regular solutions. Moreover, we introduce the notion of *k-simple forms* in the pseudo-linear setting, and we derive a new method for computing a super-irreducible form of a pseudo-linear system based on successive computations of simple forms.

The algorithms and methods developed in this chapter manipulate matrices with power series entries. However, in practice only a finite number of coefficients is necessary and we work with truncated power series. Moreover, the different algorithms developed in this chapter are fully implemented in Maple in our PseudoLinearsystems package [32]. A Maple demonstration for the implementations is given in Chapter 5.

2.2 A direct algorithm to compute simple forms

The goal of this section is to provide a unified direct algorithm for computing simple forms of a general pseudo-linear system of the form (2.1). We shall prove that the method developed in [27, 28, 34, 59] 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 1.15) and the fact that the C-automorphism ϕ preserves the valuation. When System (2.1) is not simple, then we have $\nu(\det(A)) > 0$ (otherwise, A_0 would be invertible and hence $\det(A_0\,\lambda + B) \neq 0$). The principle of the algorithm then consists in computing iteratively equivalent pseudo-linear systems $A^{(i)}\,\delta(\mathbf{y}) + B^{(i)}\,\phi(\mathbf{y}) = 0$, $i \geq 1$ satisfying $\nu(\det(A^{(i+1)})) < \nu(\det(A^{(i)}))$ with $A^{(0)} = A$. Doing so, either we find a simple system or we finally obtain an equivalent system satisfying $\nu(\det(A^{(i)})) = 0$ and such a system is necessarily simple. In order to decrease $\nu(\det(A))$, we shall first apply equivalence transformations in order to put the leading matrix pencil under a particular form from which we can reduce $\nu(\det(A))$.

2.2.1 The distinct steps

Let us consider a non-simple pseudo-linear system of the form (2.1). The constant term A_0 of A is then necessarily singular, so we have $r = \operatorname{rank}(A_0) < n$. Applying, if necessary, two constant equivalence transformations S and T as in Definition 1.15, we can always suppose that A_0 has the form:

$$A_0 = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.5}$$

All non-simple systems considered in the method described below are transformed so that A_0 complies to the form (2.5). If we partition B_0 and L_λ in four blocks as in (2.5), we get:

$$B_0 = \begin{pmatrix} B_0^{11} & B_0^{12} \\ B_0^{21} & B_0^{22} \end{pmatrix}, \quad L_\lambda = \begin{pmatrix} I_r\lambda + B_0^{11} & B_0^{12} \\ B_0^{21} & B_0^{22} \end{pmatrix}. \tag{2.6}$$

Following the terminology of [28], we refer to the last $(n-r)$ rows of L_λ as the *λ-free rows*. We are now ready to state the main results on which our algorithm for computing simple forms relies: Proposition 2.1 shows that it is always possible to transform a pseudo-linear operator into an equivalent one with a leading matrix pencil having its λ-free rows linearly dependent. Proposition 2.2 shows that if the λ-free rows of the leading matrix pencil are linearly dependent, then we can reduce $\nu(\det(A))$.

Proposition 2.1. *Consider a non-simple pseudo-linear system of the form* (2.1) *such that A_0 has the form* (2.5) *and the λ-free rows of L_λ are linearly independent. Then one can compute two matrices $S \in \mathrm{GL}_n(C[[t^{-1}]])$ and $T \in \mathrm{GL}_n(C[[t]])$ such that the equivalent operator $\widehat{L} = S\,L\,T$ has an associated leading matrix pencil of the form*

$$\widehat{L}_\lambda = \begin{pmatrix} I_p\lambda + R^{11} & R^{12} & R^{13} \\ 0 & I_{r-p}\lambda + R^{22} & R^{23} \\ 0 & R^{32} & R^{33} \end{pmatrix}, \quad \begin{cases} 0 \leq p \leq r, \\ \operatorname{rank}(R^{32} \quad R^{33}) < n - r. \end{cases} \tag{2.7}$$

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

Proof. The proof is an adaptation of those of [28, Lemma 1 and Proposition 8] to the general pseudo-linear setting. 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:

$$L'_\lambda = \begin{pmatrix} I_p\lambda + R^{11} & 0 & 0 \\ R^{21} & I_{r-p}\lambda + R^{22} & R^{23} \\ R^{31} & R^{32} & R^{33} \end{pmatrix}, \tag{2.8}$$

where $0 \leq p \leq r$ and

$$\operatorname{rank}\,(R^{32} \quad R^{33}) < n - r. \tag{2.9}$$

To do so, we initialize an integer q to 0, and let

$$S^{(q)} = \begin{pmatrix} I_q & 0 & 0 \\ 0 & 1 & v \\ 0 & 0 & I_{n-q-1} \end{pmatrix}, \quad T^{(q)} = \begin{pmatrix} 1 & -u & 0 \\ 0 & I_{r-q-1} & 0 \\ 0 & 0 & I_{n-r+q} \end{pmatrix},$$

be two $n \times n$ constant matrices where $(1 \quad v)$ is an $(n-q)$-dimensional row vector in the left null space of $L_{\lambda/\lambda=0}$, and u is the subvector of v of index 1 to $r-q-1$. Multiplying L on the left by $S^{(q)}$ and on the right by $T^{(q)}$ yields the (non-simple) operator $\widetilde{L} = S^{(q)} L\, T^{(q)}$, such that its associated leading matrix pencil $\widetilde{L}_\lambda$ is of the form (2.8) with $p = q+1$. Now, if Condition (2.9) is satisfied, then we move to the second step. Otherwise, by incrementing the value of q by 1, we shall repeat this process for the (singular) submatrix of $\widetilde{L}_\lambda$ composed of rows and columns of indices $q+1$ to n. Thus, after at most r iterations, we reach an equivalent operator $L' = S\,L\,T = A'\,\delta + B'\,\phi$ whose leading matrix pencil L'_λ satisfies Condition (2.9), with

$$S = S^{(q)} \cdots S^{(0)}, \quad T = T^{(0)} \cdots T^{(q)}.$$

It is important to note here that since the above transformations are constants, the valuation of the determinant of matrix A does not change (more importantly, it does not increase). Now we partition A' and B' into blocks of the same sizes as those of the matrix given in (2.8), that is

$$A' = \begin{pmatrix} A^{11} & A^{12} & A^{13} \\ A^{21} & A^{22} & A^{23} \\ A^{31} & A^{32} & A^{33} \end{pmatrix}, \quad B' = \begin{pmatrix} B^{11} & B^{12} & B^{13} \\ B^{21} & B^{22} & B^{23} \\ B^{31} & B^{32} & B^{33} \end{pmatrix},$$

with $A^{11} = I_p$, $A^{22} = I_{r-p}$, and the blocks A^{ij}, B^{12} and B^{13} are of positive valuations. The second step consists in multiplying the operator L on the left by the diagonal matrix $S = \operatorname{diag}(t^{-1}\,I_p, I_{r-p}, I_{n-r})$, and on the right by $T = S^{-1}$. This yields an equivalent operator $\widehat{L} = S\, L'\, T = \widehat{A}\, \delta + \widehat{B}\, \phi$, where $\widehat{A}$ and $\widehat{B}$ are given by

$$\widehat{A} = S\, A'\, T = \begin{pmatrix} A^{11} & t^{-1}\, A^{12} & t^{-1}\, A^{13} \\ t\, A^{21} & A^{22} & A^{23} \\ t\, A^{31} & A^{32} & A^{33} \end{pmatrix},$$

and

$$\widehat{B} = S\, A'\, \delta(T) + S\, B\, \phi(T) = \begin{pmatrix} t^{-1}\, \delta(t)\, A^{11} + t^{-1}\, \phi(t)\, B^{11} & t^{-1}\, B^{12} & t^{-1}\, B^{13} \\ \delta(t)\, A^{21} + \phi(t)\, B^{21} & B^{22} & B^{23} \\ \delta(t)\, A^{31} + \phi(t)\, B^{31} & B^{32} & B^{33} \end{pmatrix}.$$

Finally, since ϕ preserves the valuation, the leading matrix pencil $\widehat{L}_\lambda$ of $\widehat{L}$ has the form (2.7) with $\nu(\det(\widehat{A})) = \nu(\det(A')) = \nu(\det(A))$. This ends the proof.

□

Example 2.3. *Let us illustrate the result of Proposition 2.1 on a q-difference system. Let $t = x$ and $q \in C$ be a nonzero element which is not a root of unity. We consider the q-difference system $\phi(\mathbf{y}) = N\, \mathbf{y}$ with ϕ defined by $\phi : x \mapsto q\, x$ and*

$$N = \begin{bmatrix} \dfrac{q^2+1}{q} & -\dfrac{(\beta+x)\, q}{x} \\ \dfrac{x}{q\, x+\beta} & 0 \end{bmatrix}.$$

Here β denotes a nonzero parameter. Note that for $\beta = 100$, we find again the q-difference system considered in [5, Section 4]. Introducing the ϕ-derivation $\delta = \phi - \mathrm{id}_K$, this system can be rewritten as the pseudo-linear system $\delta(\mathbf{y}) = M\, \phi(\mathbf{y})$ where

$$M = I_2 - N^{-1} = \begin{bmatrix} 1 & -\dfrac{q\, x+\beta}{x} \\ \dfrac{x}{(\beta+x)\, q} & -\dfrac{(q^3-q^2+q)\, x+\beta}{(\beta+x)\, q^2} \end{bmatrix}.$$

In order to get the form (2.1), it is enough to multiply the latter system on the left by the diagonal matrix $\mathrm{diag}(x, x+\beta)$. This yields the (local) pseudo-linear system $L(\mathbf{y}) = A\, \delta(\mathbf{y}) + B\, \phi(\mathbf{y}) = 0$, where

$$A = \begin{bmatrix} x & 0 \\ 0 & x+\beta \end{bmatrix}, \quad B = \begin{bmatrix} -x & q\, x+\beta \\ -\dfrac{x}{q} & \dfrac{(q^3-q^2+q)\, x+\beta}{q^2} \end{bmatrix}. \tag{2.10}$$

We are now ready to illustrate the result of Proposition 2.1. The leading matrix of A and *the associated leading matrix pencil are*

$$A_0 = \begin{bmatrix} 0 & 0 \\ 0 & \beta \end{bmatrix}, \quad L_\lambda = \begin{bmatrix} 0 & \beta \\ 0 & \beta\,\lambda + \beta\,q^{-2} \end{bmatrix}.$$

The operator L is clearly not simple as $\det(L_\lambda) = 0$ *and we have* $\nu(\det(A)) = 1$. *First of all, we apply on L the constant transformations*

$$S^{(1)} = \begin{bmatrix} 0 & \beta^{-1} \\ 1 & 0 \end{bmatrix}, \quad T^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

in order to put A_0 in the form (2.5) *with $r = 1$. We then get a new operator $L^{(1)} = S^{(1)}\,L\,T^{(1)}$ whose leading matrix pencil is*

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

We now multiply $L^{(1)}$ on the left by the constant matrix

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

whose first row is a "convenient" element of the left null-space of $L^{(1)}_0$. This yields an equivalent operator $L^{(2)} = S^{(2)}\,L^{(1)}$ whose leading matrix pencil is

$$L^{(2)}_\lambda = \begin{bmatrix} \lambda & 0 \\ \beta & 0 \end{bmatrix},$$

which complies with the form (2.8) *and where Condition* (2.9) *is satisfied. We then proceed by applying on $L^{(2)}$ the two diagonal transformations defined by* $S^{(3)} = \operatorname{diag}(x^{-1}, 1)$ *and* $T^{(3)} = \operatorname{diag}(x, 1)$ *to get a new operator $L^{(3)} = S^{(3)}\,L^{(2)}\,T^{(3)}$ whose leading matrix pencil*

$$L^{(3)}_\lambda = \begin{bmatrix} \lambda + q - 1 & \dfrac{-\lambda - q + 1}{\beta\,q^2} \\ 0 & 0 \end{bmatrix},$$

is of the form (2.7). *Finally, we multiply $L^{(3)}$ on the right by the constant matrix*

$$T^{(4)} = \begin{bmatrix} 1 & \dfrac{1}{\beta\, q^2} \\ 0 & 1 \end{bmatrix}$$

to put A_0 *in the required form* (2.5)*, and get an equivalent operator* $L^{(4)} = A^{(4)}\,\delta + B^{(4)}\,\phi$ *with*

$$A^{(4)} = \begin{bmatrix} \dfrac{\beta + x}{\beta} & \dfrac{x}{\beta^2\, q^2} \\ 0 & x \end{bmatrix}, \quad B^{(4)} = \begin{bmatrix} \dfrac{(q-1)\,(\beta + x\,(q+1))}{\beta} & \dfrac{x\,(q^2-1)}{\beta^2\, q^2} \\ (q\,x + \beta)\, q\, x & \dfrac{((-q+1)\,\beta + q\,x)\,x}{q\,\beta} \end{bmatrix},$$

$$L_\lambda^{(4)} = \begin{bmatrix} \lambda + q - 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

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

Proposition 2.2. *Consider a non-simple pseudo-linear system of the form* (2.1) *such that its associated leading matrix pencil* L_λ *has the form* (2.6) *with* $\operatorname{rank}(B_0^{21} \quad B_0^{22}) < n - r$*. Then one can compute an invertible matrix* $S \in \mathrm{GL}_n\,(C[[t^{-1}]])$ *such that the equivalent operator* $\widehat{L} = S\,L = \widehat{A}\,\delta + \widehat{B}\,\phi$ *satisfies* $\nu(\det(\widehat{A})) < \nu(\det(A))$.

Proof. The proof is an adaptation of those of [28, Proposition 9] and [59, Proposition 4.3.1]. First of all, we can compute a non zero row vector of the form $u = (0 \dots 0\ 1\ w)$ in the left null space of the submatrix $(B_0^{21} \quad B_0^{22})$. Here $w \in K^{n-i}$ where i stands for the position of 1 in u. Multiplying L on the left by the constant matrix

$$S_1 = \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & 1 & w \\ 0 & 0 & I_{n-i} \end{pmatrix},$$

yields an equivalent operator $\overline{L} = S_1\,L = \overline{A}\,\delta + \overline{B}\,\phi$ such that the matrices $\overline{A}_0$ and $\overline{B}_0$ have their $(r+i)$th row equal to zero. We then apply the diagonal transformation $S_2 := \operatorname{diag}(1, \dots, 1, t^{-\mu}, 1, \dots, 1)$ where $\mu = \min\{\nu(\overline{A}(r+i, .)), \nu(\overline{B}(r+i, .))\} > 0$ to get an equivalent operator $\widehat{L} = S_2\,S_1\,L = \widehat{A}\,\delta + \widehat{B}\,\phi$. We thus have the equalities

$$\nu(\det(\widehat{A})) = \nu(\det(S_2)) + \nu(\det(S_1)) + \nu(\det(A)) = -\mu + \nu(\det(A)).$$

This implies $\nu(\det(\widehat{A})) < \nu(\det(A))$ which ends the proof. □

Example 2.4. *In Example 2.3, without modifying the valuation of the determinant of the matrix* A*, we have obtained an equivalent operator* $L^{(4)}$ *whose leading pencil is of the form* (2.7)*. We shall then proceed by reducing the value of the valuation of the determinant of the matrix* $A^{(4)}$*. We note that here* $r = 1$ *and the second row of the leading pencil* $L_\lambda^{(4)}$

is already zero. The second rows of both $A^{(4)}$ and $B^{(4)}$ are of valuation 1, so we apply the transformation defined by $S^{(5)} = \mathrm{diag}(1, x^{-1})$ to obtain the new equivalent operator $L^{(5)} = S^{(5)}\, L^{(4)} = A^{(5)}\,\delta + B^{(5)}\,\phi$ with

$$A^{(5)} = \begin{bmatrix} \frac{x}{\beta} + 1 & \frac{x}{\beta^2\, q^2} \\ 0 & 1 \end{bmatrix}, \quad B^{(5)} = \begin{bmatrix} \frac{(q-1)\,(\beta + x\,(q+1))}{\beta} & \frac{x\,(q^2-1)}{\beta^2\, q^2} \\ (qx+\beta)\,q & \frac{\beta + (x-\beta)\,q}{\beta\, q} \end{bmatrix}. \tag{2.11}$$

As expected, we have $\nu(\det(A^{(5)})) = 0 < 1$. The leading matrix pencil of $L^{(5)}$ is given by

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

and $L^{(5)}$ is a simple q-difference operator. Finally the matrices that transform the non-simple operator L into the simple operator $L^{(5)}$ are given by

$$S = S^{(5)}\, S^{(3)}\, S^{(2)}\, S^{(1)} = \begin{bmatrix} -\frac{1}{\beta\, x\, q^2} & \frac{1}{\beta\, x} \\ \frac{1}{x} & 0 \end{bmatrix}, \quad T = T^{(1)}\, T^{(3)}\, T^{(4)} = \begin{bmatrix} 0 & 1 \\ x & \frac{x}{\beta\, q^2} \end{bmatrix}.$$

2.2.2 An example

Let $\phi : x \mapsto 2\,x - 1$ be a C-automorphism over K and $\delta = \mathrm{id}_K - \phi$ be a ϕ-derivation. Consider the pseudo-linear operator $L = A\,\delta + B\,\phi$, where A and B are given by:

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

Let us compute a simple form at the ϕ-fixed point $x_0 = 1$ (see Definition 1.8). For this we set $t = x - 1$ and we note that $\nu(\det(A)) = 3$. The associated leading matrix pencil is

$$L_\lambda = A_0\,\lambda + B_0 = \begin{bmatrix} \lambda+1 & 1 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

and its determinant vanishes identically in λ, therefore the operator L is non-simple. The λ-free rows of L_λ are linearly dependent and A_0 is already of the form (2.5) with $r = \mathrm{rank}(A_0) = 2$. So we shall first apply Proposition 2.2. The vector $(1\ w_1\ w_2) = (1\ 0\ -1)$ is in the left null space of the matrix constituting the λ-free rows of L_λ. Multiplying L on the left by the constant matrix

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

yields an equivalent operator $L^{(1)} = S^{(1)}\, L = A^{(1)}\, \delta + B^{(1)}\, \phi$, where

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

The third rows of $A^{(1)}$ and $B^{(1)}$ have valuations equal to 1 and 2 respectively. Let $\mu = \min\{1, 2\} = 1$ and multiply $L^{(1)}$ on the left by the diagonal matrix

$$S^{(2)} = \mathrm{diag}(1,\ 1,\ (x-1)^{-\mu},\ 1,\ 1).$$

This yields an equivalent operator $L^{(2)} = S^{(2)}\, L^{(1)} = A^{(2)}\, \delta + B^{(2)}\, \phi$, with

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & x-1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \quad B^{(2)} = \begin{bmatrix} 1 & x & 0 & 0 & x-1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 1 & 0 & x-1 & 0 & 0 \\ 0 & 1 & 0 & 0 & x\,(x-1)^2 \end{bmatrix},$$

$$L_\lambda^{(2)} = A_0^{(2)}\,\lambda + B_0^{(2)} = \begin{bmatrix} \lambda+1 & 1 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & -\lambda \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The operator $L^{(2)}$ is still non-simple but we have decreased the valuation of the determinant of the A-matrix as $\nu\left(\det(A^{(2)})\right) = 2 < \nu\left(\det(A)\right) = 3$. We restart the process on $L^{(2)}$. Multiply first $L^{(2)}$ on the right by the constant matrix

$$T^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

in order to put $A_0^{(2)}$ in the required form (2.5). This yields the equivalent operator $L^{(3)} = L^{(2)}\,T^{(3)} = A^{(3)}\,\delta + B^{(3)}\,\phi$, with

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

$$L_\lambda^{(3)} = A_0^{(3)}\,\lambda + B_0^{(3)} = \begin{bmatrix} \lambda+1 & 1 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The λ-free rows of $L_\lambda^{(3)}$ are linearly independent and $A_0^{(3)}$ is of the form (2.5) with $r = \operatorname{rank}(A_0^{(3)}) = 3$. So we shall now apply Proposition 2.1. The vector $(1\ v_1\ v_2\ v_3\ v_4) = (1\ 0\ 0\ -1\ -1)$ is in the left null space of $L^{(3)}_{\lambda/\lambda=0}$. Multiplying $L^{(3)}$ on the left by the constant matrix

$$S^{(4)} = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and on the right by $T^{(4)} = I_5$, yields an equivalent operator $L^{(4)} = S^{(4)}\, L^{(3)}\, T^{(4)} = A^{(4)}\,\delta + B^{(4)}\,\phi$, with

$$A^{(4)} = \begin{bmatrix} 1 & 0 & 0 & -x+1 & -x+1 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x-1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \quad B^{(4)} = \begin{bmatrix} 0 & x-1 & -x+1 & 0 & -x^3+2\,x^2-x \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 1 & 0 & x-1 & 0 & x-1 \\ 0 & 1 & 0 & 0 & x^3-2\,x^2+x \end{bmatrix},$$

$$L_\lambda^{(4)} = A_0^{(4)}\,\lambda + B_0^{(4)} = \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

$L_\lambda^{(4)}$ complies with the form (2.8) with $p = 1$ and Condition (2.9) is satisfied. We then proceed by applying on $L^{(4)}$ the diagonal transformations defined by

$$S^{(5)} = \operatorname{diag}(\,(x-1)^{-1},\,1,\,1,\,1,\,1), \quad T^{(5)} = \operatorname{diag}(x-1,\,1,\,1,\,1,\,1),$$

to get a new operator $L^{(5)} = S^{(5)}\, L^{(4)}\, T^{(5)} = A^{(5)}\,\delta + B^{(5)}\,\phi$, with

$$A^{(5)} = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x-1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \quad B^{(5)} = \begin{bmatrix} -1 & 1 & -1 & 0 & -(x-1)\,x \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 2\,x-2 & 0 & x-1 & 0 & x-1 \\ 0 & 1 & 0 & 0 & x\,(x-1)^2 \end{bmatrix},$$

$$L_\lambda^{(5)} = A_0^{(5)}\,\lambda + B_0^{(5)} = \begin{bmatrix} \lambda-1 & 1 & -1 & -\lambda & -\lambda \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The matrix $A_0^{(5)}$ is not in the form (2.5) so we can multiply $L^{(5)}$ on the right by the constant matrix

$$T^{(6)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and get an equivalent operator $L^{(6)} = L^{(5)}\,T^{(6)} = A^{(6)}\,\delta + B^{(6)}\,\phi$, where

$$A^{(6)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x-1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \quad B^{(6)} = \begin{bmatrix} -1 & 1 & -1 & -1 & -x^2+x-1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 2\,x-2 & 0 & x-1 & 2\,x-2 & 3\,x-3 \\ 0 & 1 & 0 & 0 & x\,(x-1)^2 \end{bmatrix},$$

and with an associated leading matrix pencil

$$L_\lambda^{(6)} = A_0^{(6)}\,\lambda + B_0^{(6)} = \begin{bmatrix} \lambda-1 & 1 & -1 & -1 & -1 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

having the form (2.7). Note that we have $\nu\left(\det(A^{(6)})\right) = \nu\left(\det(A^{(3)})\right) = 2$ (in fact we have $A^{(6)} = A^{(3)}$ by coincidence). We are now ready to apply Proposition 2.2 in order to reduce $\nu\left(\det(A^{(6)})\right)$. The fourth row of $L_\lambda^{(6)}$ is already zero and the fourth rows of both matrices $A^{(6)}$ and $B^{(6)}$ are of valuations 1, so we multiply $L^{(6)}$ on the left by the matrix $S^{(7)} = \operatorname{diag}(1,\,1,\,1,\,(x-1)^{-1},\,1)$ to obtain the new equivalent operator $L^{(7)} = S^{(7)}\,L^{(6)} = A^{(7)}\,\delta + B^{(7)}\,\phi$, where

$$A^{(7)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \quad B^{(7)} = \begin{bmatrix} -1 & 1 & -1 & -1 & -x^2+x-1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 2 & 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 0 & x^3-2\,x^2+x \end{bmatrix}.$$

As expected, we have $\nu\left(\det(A^{(7)})\right) = 1 < \nu\left(\det(A^{(6)})\right) = 2$. Moreover, $L^{(7)}$ is an equivalent simple pseudo-linear operator with a non-singular leading matrix pencil given by

$$L_\lambda^{(7)} = A_0^{(7)}\,\lambda + B_0^{(7)} = \begin{bmatrix} \lambda-1 & 1 & -1 & -1 & -1 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 2 & 0 & 1 & \lambda+2 & 3 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Finally the matrices that transform the non-simple operator L into the equivalent simple operator $L^{(7)}$ are given by

$$S = S^{(7)}\,S^{(5)}\,S^{(4)}\,S^{(2)}\,S^{(1)} = \begin{bmatrix} \frac{1}{x-1} & 0 & 0 & -\frac{1}{x-1} & -\frac{1}{x-1} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x-1} & 0 & -\frac{1}{x-1} \\ 0 & 0 & 0 & \frac{1}{x-1} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T = T^{(3)}\,T^{(4)}\,T^{(5)}\,T^{(6)} = \begin{bmatrix} x-1 & 0 & 0 & x-1 & x-1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

2.2.3 The complete algorithm

Applying recursively Propositions 2.1 and 2.2 provides an algorithm, called **SimpleForm** below, to compute a simple form of a pseudo-linear system of the form (2.1). The algorithm takes as an input the two matrices A and B in $\mathbb{M}_n(C[[t]])$, the C-automorphism ϕ, the ϕ-derivation δ (those of the pseudo-linear operator $L = A\,\delta + B\,\phi$) and the local parameter t. It returns four matrices $\widehat{A}$, $\widehat{B}$, S and T such that the equivalent operator $\widehat{L} = S\,L\,T = \widehat{A}\,\delta + \widehat{B}\,\phi$ is simple, and the polynomial $\det(\widehat{L}_\lambda)$ which is thus non-zero.

Algorithm **SimpleForm**

Input: Two matrices A and B, the automorphism ϕ, the ϕ-derivation δ (those of $L = A\,\delta + B\,\phi$) and a local parameter t.

Output: Four matrices $\widehat{A}$, $\widehat{B}$, $\widehat{S}$ and $\widehat{T}$ such that the operator $\widehat{L} = \widehat{S}\,L\,\widehat{T} = \widehat{A}\,\delta + \widehat{B}\,\phi$ is simple, and the polynomial $\det(\widehat{L}_\lambda)$.

Initialization: $\widehat{S} \leftarrow I_n$, $\widehat{T} \leftarrow I_n$, $\widehat{A} \leftarrow A$, $\widehat{B} \leftarrow B$ and $\widehat{L}_\lambda \leftarrow A_0\,\lambda + B_0$;

While $\det(\widehat{L}_\lambda)$=0 **do**

1. Compute two matrices S and T as in Proposition 2.1;
2. Update $\widehat{A} \leftarrow S\,\widehat{A}\,T$, $\widehat{B} \leftarrow S\,\widehat{A}\,\delta(T) + S\,\widehat{B}\,\phi(T)$, $\widehat{S} \leftarrow S\,\widehat{S}$ and $\widehat{T} \leftarrow \widehat{T}\,T$;
3. Compute a matrix S as in Proposition 2.2;
4. Update $\widehat{A} \leftarrow S\,\widehat{A}$, $\widehat{B} \leftarrow S\,\widehat{A} + S\,\widehat{B}$ and $\widehat{S} \leftarrow S\,\widehat{S}$;
5. Set $\widehat{L}_\lambda = \widehat{A}_0\,\lambda + \widehat{B}_0$;

end do

Return $\widehat{A}$, $\widehat{B}$, $\widehat{S}$, $\widehat{T}$ and $\det(\widehat{L}_\lambda)$;

Remark 2.1. *Before Steps 1 and 3 of the above algorithm, one needs to apply, if necessary, constant transformations in order to put $\widehat{A}_0$ in the form (2.5).*

Remark 2.2. *As it is explained in the beginning of Section 2.2, in each passage through the **While** loop in Algorithm* **SimpleForm***, the value of $\nu(\det(\widehat{A}))$ decreases at least by 1. This happens in Step 4 of the algorithm (see Proposition 2.2). Thus, after at most $\nu(\det(A))$ iterations where A is the matrix given in input, we find an equivalent simple system.*

The arithmetic complexity of this algorithm is studied in [59] 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 (2.1). Indeed, apart from constant transformations, Formulas (1.11) 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 ϕ and the ϕ-derivation δ will not influence the number of arithmetic operations. We thus state the following complexity estimate of Algorithm **SimpleForm**.

Proposition 2.3. *Let us consider a pseudo-linear system* (2.1)*. Let $h = \nu(\det(A))$ and suppose that the t-adic expansions of A and B are known up to an order $m \geq h$. Then Algorithm* **SimpleForm** *computes a simple form of* (2.1) *using at most $O(n^{\omega+1}\, h + m\, n^3\, h)$ arithmetic operations in the field C, where ω denotes the linear algebra exponent (see [94]).*

Proof. This follows from the previous explanations and [59, Lemmas 4.3.1, 4.3.2, 4.3.3]. □

2.3 Simple forms and local analysis

Simple forms are very useful for the *local analysis* of pseudo-linear systems over the field $C(x)$ of rational functions near a ϕ-fixed singular point x_0 - see Definition 1.8. Recall that (see Section 1.4) in order to locally study a pseudo-linear system, we fix a local parameter t (for instance, $t = x - x_0$ or $t = x^{-1}$ depending on x_0 being finite or not), we imbed $C(x)$ in the local $\phi\delta$-field (K, ϕ, δ) where $K = C((t))$, and we write the system as a local pseudo-linear system (1.9). We remind the reader that this latter system is a system of the form

$$t^{p-\omega}\delta(\mathbf{y}) = M\phi(\mathbf{y}), \tag{2.12}$$

where $M = \sum_{i\geq 0} M_i\, t^i \in M_n(\mathcal{O})$ with $\mathcal{O} = C[[t]]$, $p \in \mathbb{N}$ is the Poincaré rank of (2.12) and $\omega \in \mathbb{Z}$ is the degree of the ϕ-derivation δ. Recall that System (2.12) can be written as

$$A\,\tilde{\delta}(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{2.13}$$

where $\tilde{\delta} = t^{-\omega}\,\delta$, $A = \mathrm{diag}(t^{\alpha_1}, \ldots, t^{\alpha_n}) \in \mathbb{M}_n(\mathcal{O})$ with $\alpha_i = -\min\{0, \nu(M(i,.)) - p\}$, and $B = -t^{-p}\,A\,M \in \mathbb{M}_n(\mathcal{O})$. Here $M(i,.)$ denotes the i^{th} row of M. Then, System (2.12) is said to be simple if System (2.13) is simple.

Definition 2.3. *([23, 24]). With the above notations, a local pseudo-linear system of the form* (2.12) *or* (2.13) *is said to be of the* first kind *if it is equivalent to a system with Poincaré rank equal to 0. In this case, the singularity x_0 is said to be* regular*. Otherwise, x_0 is said to be an* irregular *singularity.*

Note that an equivalent system of (2.12) has to be of the form

$$t^{p-\omega}\delta(\mathbf{y}) = T^{-1}\left(M\,\phi(T) - t^{p-\omega}\,\delta(T)\right)\,\phi(\mathbf{y}),$$

for some $T \in \mathrm{GL}_n(K)$ (see Definition 1.6). We define the *indicial polynomial* of a simple pseudo-linear system as follows:

Definition 2.4. *With the above notations, let us consider a simple system of the* *form* (2.13) *or* (2.12)*. We define its* indicial polynomial *as*

$$\varphi(\lambda) = \det(d\,[\lambda]_c\,A_0 + c^{\lambda}\,B_0), \tag{2.14}$$

where d, c and $[\lambda]_c$ are defined in Definition 1.13.

Remark 2.3. *When System* (2.13) *or* (2.12) *is simple, we have* $\det(L_\lambda) = \det(A_0\,\lambda + B_0) \neq 0$ *so that* $\varphi(\lambda) \neq 0$*. For the differential case ($\phi = \mathrm{id}_K$) and the difference case ($\phi : t \mapsto t+1$), we have $c = d = 1$ which yields $\varphi(\lambda) = \det(L_\lambda)$ a polynomial in λ. In general, if $c = 1$ then $\varphi(\lambda)$ is a polynomial in λ. 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^{λ}. By abuse of language, we shall also call it polynomial and its "roots" are the values of λ such that c^{λ} annihilates $\varphi(\lambda)$. They can be computed using [24, Lemma 3.1].*

In this section, we first recall the notions of Moser- and super-irreducible forms of pseudo-linear systems in the same context as introduced in [23], and we establish the connection between the notions of super-irreducibility and simplicity. Then 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 so called *k-simple forms* which are closely related to super-irreducible forms. Finally we propose a new method to compute a super-irreducible form of a pseudo-linear system based on successive computations of k-simple forms.

2.3.1 Moser- and super-irreducible forms

The notion of *Moser-irreducible* forms introduced by Moser [78] and its generalisation, the *super-irreducible* forms introduced by Hilali and Wazner [67], have been proven to be essential concepts for the symbolic resolution of differential and difference systems, see for instance [25, 86]. The notions of Moser- and super-irreducible forms have been extended for more general pseudo-linear systems in [23]. In the following, we briefly recall these notions. Denote by π the canonical homomorphism from $\mathcal{O} = C[[t]]$ into C.

We associate to the local pseudo-linear system (2.12) with Poincaré rank p the following rational numbers:

$$m_{\phi,\delta}(M) = \begin{cases} p + \dfrac{\mathrm{rank}(M_0)}{n} & \text{if} \quad p > 0, \\ 0 & \text{if} \quad p = 0, \end{cases}$$

and

$$\mu_{\phi,\delta}(M) = \min\left\{ m_{\phi,\delta}\Big(T^{-1}\left(M\,\phi(T) - t^{p-\omega}\,\delta(T)\right)\Big) \quad ; \quad T \in \mathrm{GL}_n(K) \right\}.$$

Definition 2.5. *System* (2.12) *is said to be* Moser-irreducible *if* $m_{\phi,\delta}(M) = \mu_{\phi,\delta}(M)$. *Otherwise it is said to be* Moser-reducible.

Remark 2.4. *The singularity is regular in the sense of Definition 2.3 if and only if* $\mu_{\phi,\delta}(M) = 0$.

The following result gives a criterion for Moser-irreducibility:

Theorem 2.1. *([23, Theorem 3.1]). Suppose that* $p > 0$ *and let* $n_0 = \mathrm{rank}(M_0)$. *Then System* (2.12) *is Moser-irreducible if and only if the polynomial*

$$\theta_1(\lambda) = \pi\left(t^{n_0} \det\left(I_n\,\lambda - t^{-1}\,M\right)\right) \in \mathcal{O}[\lambda],$$

does not vanish identically in λ.

Remark 2.5. *One can easily verify that the polynomial* $\theta_1(\lambda)$ *depends only on* M_0 *and* M_1*:*

$$\theta_1(\lambda) = \pi\left(t^{n_0}\,\det\left(I_n\,\lambda - t^{-1}\,M_0 - M_1\right)\right).$$

Now suppose that the Poincaré rank p of System (2.12) satisfies $p > 0$. Denote by n_i the number of rows of M with valuation i. Define, for $1 \leq k \leq p$, the rational numbers

$$m^{(k)}_{\phi,\delta}(M) = p + \frac{n_0}{n} + \frac{n_1}{n^2} + \cdots + \frac{n_{k-1}}{n^k},$$

and

$$\mu^{(k)}_{\phi,\delta}(M) = \min\left\{m^{(k)}_{\phi,\delta}\left(T^{-1}\left(M\,\phi(T) - t^{p-\omega}\,\delta(T)\right)\right) \quad ; \quad T \in \mathrm{GL}_n(K)\right\}.$$

Definition 2.6. *System* (2.12) *is said to be* k-irreducible *if* $m^{(k)}_{\phi,\delta}(M) = \mu^{(k)}_{\phi,\delta}(M)$. *Otherwise it is said to be* k-reducible. *System* (2.12) *is said to be* super-irreducible *if it is k-irreducible for every k, or equivalently if* $m^{(p)}_{\phi,\delta}(M) = \mu^{(p)}_{\phi,\delta}(M)$.

A criterion for k-reducibility is obtained in the following way (as in the differential case [67]). Define the non-negative integer s_k as

$$s_k = k\,n_0 + (k-1)\,n_1 + \cdots + n_{k-1}, \tag{2.15}$$

and the polynomial

$$\theta_k(\lambda) = \pi\left(t^{s_k}\det(I_n\,\lambda - t^{-k}\,M)\right) \in \mathcal{O}[\lambda]. \tag{2.16}$$

Theorem 2.2. *([23, Theorem 3.2]). System* (2.12) *is k-irreducible if and only if, for* $j = 1, \ldots, k$, *the polynomials* $\theta_j(\lambda)$ *do not vanish identically in* λ.

Remark 2.6. *The notion of 1-irreducible system coincides with that of Moser-irreducible systems.*

We establish now the connection between the notions of super-irreducibility and simplicity.

Proposition 2.4. *If System (2.12) is super-irreducible then it is simple.*

Proof. We have to prove that System (2.13) is simple. Denote by $L_\lambda = A_0\,\lambda + B_0$ its leading matrix pencil. If $p = 0$ then the matrix A is just the identity matrix and this easily results into a simple system. Suppose now that $p > 0$. It is easy to verify that $\det(A) = t^{s_p}$, and one has

$$\det(L_\lambda) = \pi\left(\det(A\,\lambda + B)\right) = \pi\left(\det(A\,\lambda - t^{-p}\,A\,M)\right) = \pi\left(t^{s_p}\det(I_n\,\lambda - t^{-p}\,M)\right) = \theta_p(\lambda).$$

Now if System (2.12) is super-irreducible, then it follows from Theorem 2.2 that the polynomial $\theta_p(\lambda)$ is not zero and (2.13) is simple. □

Note that if a system is simple, then it is not necessarily super-irreducible. It is not even k-irreducible for some positive integer k. Indeed, super-irreducibility requires that $\theta_k(\lambda) \neq 0$ for all $1 \leq k \leq p$, while simplicity requires that only $\theta_p(\lambda) \neq 0$.

Example 2.5. *Let ϕ be the C-automorphism over K defined by $\phi : t \mapsto q\,t$ such that $q \in C^*$ is not a root of unity, and δ be a ϕ-derivation defined as $\delta = \mathrm{id}_K - \phi$ having degree $\omega = 0$. The pseudo-linear operator*

$$L = A\,\delta + B\,\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & t \\ 0 & 0 & t^2 \end{bmatrix}\delta + \begin{bmatrix} -t^2+2 & t^2-3 & 0 \\ 2 & -3 & -t^2 \\ 0 & 0 & -t^3-1 \end{bmatrix}\phi,$$

is simple since $\det(L_\lambda) = \det(A_0\,\lambda + B_0) = -\lambda^2 + \lambda \neq 0$. *Write the system $L(\mathbf{y}) = 0$ in the form $\delta(\mathbf{y}) = N\,\phi(\mathbf{y})$ with*

$$N = -A^{-1}\,B = \begin{bmatrix} t^2-2 & -t^2+3 & 0 \\ -2 & 3 & t^2 + \dfrac{-t^3-1}{t} \\ 0 & 0 & \dfrac{t^3+1}{t^2} \end{bmatrix}.$$

The latter system has a Poincaré rank $p = 2$ and it can be written in the local form (2.12) as

$$t^2\,\delta(\mathbf{y}) = \begin{bmatrix} t^2\,(t^2-2) & -t^2\,(t^2-3) & 0 \\ -2\,t^2 & 3\,t^2 & -t \\ 0 & 0 & t^3+1 \end{bmatrix}\phi(\mathbf{y}).$$

The polynomial $\theta_2(\lambda)$ defined in (2.16) is equal to

$$\theta_2(\lambda) = \pi\left(t\left(\lambda\,t^2 - t^3 - 1\right)\left(-\lambda\,t^2 + \lambda^2 + t^2 - \lambda\right)\right) = 0,$$

and it follows form Theorem 2.2 that the system is not super-irreducible.

2.3.2 Regular solutions and regular singularities

We are interested here in the task of computing so called *regular solutions* (see Definition 2.7 below) of pseudo-linear systems. Appropriate methods have been designed in the past for computing such solutions for the individual types of systems: see for instance [27, 38] for differential, [8, 28] for difference, and [4, 14] for q-difference systems. More recently, a generic algorithm has been developed in [24] for general pseudo-linear systems.

Consider again System (2.12) and recall that it can be written as System (2.13). We denote by e_λ and u the two scalar functions of the variable t in some extension of K satisfying:

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

where c and d are defined in Definition 1.13. It has been shown in [24, Appendix A] that, when looking for regular solutions for specific classes of linear functional systems, e_λ and u can be expressed using the exponential and logarithm function. For instance, in the the pure differential and difference cases, one has $e_\lambda(t) = t^\lambda$ and $u(t) = \ln(t)$, while in the q-difference case ($\phi : x \mapsto q\,x$, $q \neq 0, 1$), one has $e_\lambda(t) = t^\lambda$ and $u(t) = \log_q(t)$.

According to [24, Theorem 4.3], a simple pseudo-linear system (2.13) admits $\deg(\varphi(\lambda))$ linearly independent local solutions $\mathbf{y}_i$'s of the form:

$$\mathbf{y}_i(t) = e_{\lambda_i}(t) \sum_{j=0}^{m_i-1} \mathbf{z}_{i,j}(t)\,\frac{u(t)^{m_i-j}}{(m_i-j)!}, \quad i = 1, \ldots, \deg(\varphi(\lambda)), \tag{2.17}$$

where $\lambda_i \in \overline{C}$ is a root of multiplicity m_i of the indicial polynomial $\varphi(\lambda)$ and $\mathbf{z}_{i,j}(t) \in C[[t]]^n$.

Definition 2.7. *Local solutions of the form* (2.17) *are called* regular solutions *of Systems* (2.12) *and* (2.13).

In order to compute a basis of regular solutions of a given pseudo-linear system, one then needs to effectively compute the polynomial $\varphi(\lambda)$. This polynomial can be read off from System (2.13) only if the system is in simple form. Recall that (see Proposition 2.4) if the system is super-irreducible then it is simple with $\det(L_\lambda) = \theta_p(\lambda) \neq 0$, where $\theta_p(\lambda)$ is defined in (2.16). In [24, 49] and in previous works restricted to the differential or difference case, it is proposed to compute a super-irreducible form of the system to effectively compute $\theta_p(\lambda)$, which allows to obtain $\det(L_\lambda)$ and $\varphi(\lambda)$. The algorithm **SimpleForm** developed in the present dissertation can then be used to compute $\det(L_\lambda)$ and $\varphi(\lambda)$ for any pseudo-linear system directly without recourse to reduction algorithms. Note that such an approach has already been proposed for the differential [27, 59] and difference [28] cases.

Once $\varphi(\lambda)$ is computed, we proceed as follows: let λ_i be root of multiplicity m_i of $\varphi(\lambda)$. The change of variable $\mathbf{y} = e_{\lambda_i}\mathbf{z}$ reduces the problem to computing m_i series solutions $\mathbf{z}_{i,j}(t) \in C[[t]]^n$ with $j = 0, \ldots, m_i - 1$. Computing these series solutions can be done using the monomial-by-monomial method described in [24, Section 5]. Doing so, we obtain one regular solution of the form (2.17). Repeating the same process for each root of $\varphi(\lambda)$, we finally obtain $\deg(\varphi(\lambda))$ regular solutions of (2.13). We refer to [24, 49] for more details on regular solutions of pseudo-linear systems.

Example 2.6. *Let us go back to the q-difference system considered in Examples 2.3 and 2.4. For a local q-difference system at $x = 0$, we take as local parameter $t = x$ and consider the local pseudo-linear system $A\delta(\mathbf{y}) + B\phi(\mathbf{y}) = 0$, where A and B are given by (2.10). 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** *provides an equivalent simple system $A^{(5)}\delta(\mathbf{y}) + B^{(5)}\phi(\mathbf{y})$, where $A^{(5)}$ and $B^{(5)}$ are given by (2.11) and 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 $\lambda = \pm 1$ (see Remark 2.3) and from (2.17), it follows that the system admits two linearly independent regular solutions at $x = 0$ given by $\mathbf{y}_1(x) = x\,\mathbf{z}_{1,0}(x)$, $\mathbf{y}_2(x) = x^{-1}\,\mathbf{z}_{2,0}(x)$ with

$$\mathbf{z}_{1,0} = \begin{bmatrix} \beta^{-1} \\ \beta^{-2}q^{-2}x + O\left(x^2\right) \end{bmatrix}, \quad \mathbf{z}_{2,0} = \begin{bmatrix} 1 \\ \beta^{-1}x + O\left(x^2\right) \end{bmatrix}.$$

Algorithm **SimpleForm** can also be used to determine the nature of a singularity in the regular / irregular classification. Indeed, the singularity of System (2.13) is regular if and only if the system admits a basis of n regular solutions of the form (2.17). We then get the following lemma:

Lemma 2.1. *([24, Theorem 3.2]). The singularity of a simple pseudo-linear system (2.13) is regular if and only if $\deg(\varphi(\lambda)) = n$, where $\varphi(\lambda)$ is the indicial polynomial of (2.13).*

Note that $\deg(\varphi(\lambda)) = n$ is equivalent to $\det(A_0) \neq 0$. Algorithm **SimpleForm** then provides an alternative to Moser's reduction [16, 18, 78] for recognising the nature of a singularity.

2.3.3 k-simple forms

The notion of k-simple forms has been first introduced by Pflügel [86] for differential systems as a generalisation of the notion of simple form [20]. These forms turned out to be very useful for computing important *local data* of a differential system such as formal solutions and the integer slopes of the *Newton polygon* [19, 90]. The method proposed in [86] to compute k-simple forms consists in applying super-reduction algorithms [67] first. Later on, a direct, i.e. without applying super-reduction, algorithm was developed in [34, 59] for computing k-simple forms of differential systems. We introduce here the notion of k-simple forms in the general pseudo-linear setting.

Consider again a local pseudo-linear system of the form (2.12). For any nonnegative integer k, we define

$$A^{(k)} = \operatorname{diag}(t^{\alpha_1}, \ldots, t^{\alpha_n}), \quad \alpha_i = \max\left(0, p - k - \nu(M(i,.))\right), \tag{2.18}$$

and

$$B^{(k)} = -t^{k-p}\, A^{(k)}\, M. \tag{2.19}$$

Note that $B^{(k)} \in \mathbb{M}_n(C[[t]])$ since we have $\nu(A^{(k)}\, M) \geq k - p$.

Definition 2.8. *Let k be a nonnegative integer and $\delta_k = t^{k-\omega}\,\delta$. Then System (2.12) is said to be* k-simple *if the pseudo-linear system*

$$A^{(k)}\,\delta_k(\mathbf{y}) + B^{(k)}\,\phi(\mathbf{y}) = 0 \tag{2.20}$$

is simple, i.e., $\det(A_0^{(k)}\,\lambda + B_0^{(k)}) \neq 0$. *When this is the case, we call*

$$\Psi_k(\lambda) = \det(A_0^{(k)}\,\lambda + B_0^{(k)}),$$

the characteristic polynomial of (2.12) associated to k.

Remark 2.7. *The notion of 0-simple systems coincides with the notion of simple systems.*

If System (2.12) is not k-simple, then applying Algorithm **SimpleForm** to the non-simple system (2.20), produces a new simple system

$$\widetilde{A^{(k)}}\,\delta_k(\mathbf{y}) + \widetilde{B^{(k)}}\,\phi(\mathbf{y}) = 0.$$

The system $t^{p-\omega}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y})$ where $\widetilde{M} = -t^{p-k}\,\widetilde{A^{(k)}}^{-1}\,\widetilde{B^{(k)}}$ is then an equivalent

k-simple system of (2.12) (or a k-simple form of (2.12)). Note that System (2.12) is necessarily k-simple for $k \geq p$ since in this case the α_i's in (2.18) are all equal to zero so that $A^{(k)} = I_n$.

Example 2.7. *Let $\phi : x \mapsto x - 1$ be a C-automorphism over K and $\delta = (x-1)(\mathrm{id}_K - \phi)$ be a ϕ-derivation. As ∞ is the only singularity in this case, then we introduce the local parameter $t = x^{-1}$. The degree of δ is thus $\omega = 0$. Let us compute a 2-simple form of the pseudo-linear system*

$$x^{-3}\delta(\mathbf{y}) = \begin{bmatrix} 0 & \frac{1}{x^4} - \frac{1}{x^5} & 0 & \frac{1}{x^3} \\ \frac{1}{x^4} & 0 & \frac{4}{x^6} & 0 \\ \frac{1}{x^2} & \frac{1}{x^3} & 0 & \frac{1}{x} + \frac{1}{x^3} \\ 0 & \frac{4}{x^3} + \frac{2}{x^2} + 1 & 0 & \frac{1}{x^3} - \frac{1}{x} \end{bmatrix} \phi(\mathbf{y}) \tag{2.21}$$

having Poincaré rank $p = 3$. Using the above notations, System (2.21) can be transformed into the pseudo linear system $A^{(2)}\,\delta_2(\mathbf{y}) + B^{(2)}\,\phi(\mathbf{y}) = 0$ where $\delta_2 = x^{-2}\,\delta$ and

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & x^{-1} \end{bmatrix}, \quad B^{(2)} = \begin{bmatrix} 0 & -\frac{x-1}{x^4} & 0 & -\frac{1}{x^2} \\ -\frac{1}{x^3} & 0 & -\frac{4}{x^5} & 0 \\ -\frac{1}{x} & -\frac{1}{x^2} & 0 & -\frac{x^2+1}{x^2} \\ 0 & -\frac{x^3+2\,x+4}{x^3} & 0 & \frac{x^2-1}{x^3} \end{bmatrix}.$$

The latter system is not simple and, using Algorithm **SimpleForm**, *we compute an equivalent simple system $\widetilde{A^{(2)}}\,\delta_2(\mathbf{y}) + \widetilde{B^{(2)}}\,\phi(\mathbf{y}) = 0$ where $\widetilde{A^{(2)}} = I_4$ and*

$$\widetilde{B^{(2)}} = \begin{bmatrix} -\frac{1}{x^2} & -\frac{1}{x^3} & 0 & -\frac{1}{x} \\ -\frac{1}{(x-1)\,x^2} & -\frac{1}{x^2} & -\frac{4}{x^4} & 0 \\ -\frac{1}{x\,(x-1)} & -\frac{1}{(x-1)\,x^2} & 0 & -\frac{x^2+1}{x^2} \\ 0 & -\frac{x^3+2\,x+4}{(x-1)\,x^2} & 0 & \frac{x^2-1}{x^2} \end{bmatrix}.$$

Finally, the latter system can be written as

$$x^{-3}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y}), \tag{2.22}$$

where $\widetilde{M} = -x^{-1}\, \widetilde{A^{(2)}}^{-1}\, \widetilde{B^{(2)}}$*, and we say that System* (2.22) *is a 2-simple form of System* (2.21)*.*

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 2.2. *Any super-irreducible system of the form* (2.12) *is* k*-simple for* $k = 0, \ldots, p-1$*.*

Proof. Consider a system of the form (2.12) and let $A^{(k)}$ and $B^{(k)}$ be defined as in (2.18) and (2.19). One can easily verify that the matrix $A^{(k)}$ satisfies $\det(A^{(k)}) = s_{p-k}$ where s_{p-k} is defined in (2.15). Moreover one has for $k = 0, \ldots, p-1$,

$$\begin{aligned} \det\left(A_0^{(k)}\,\lambda + B_0^{(k)}\right) &= \pi\left(\det\left(A^{(k)}\,\lambda + B^{(k)}\right)\right) \\ &= \pi\left(\det\left(A\,\lambda - t^{k-p}\,A^{(k)}\,M\right)\right) \\ &= \pi\left(t^{s_{p-k}}\det\left(I_n\,\lambda - t^{k-p}\,M\right)\right) \\ &= \theta_{p-k}(\lambda), \end{aligned}$$

where $\theta_{p-k}(\lambda)$ is defined in (2.16). If System (2.12) is super-irreducible, then it is in particular p-irreducible. It follows from Theorem 2.2 that $\theta_{p-k}(\lambda) \neq 0$ for $k = 0, \ldots, p-1$, and hence (2.12) is k-simple. □

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, see for instace Example 2.5. So if one is interested in computing a k-simple form for just one value of k then direct (i.e. avoiding super-reduction) methods have to be preferred. Our Algorithm **SimpleForm** provides such a direct method. It is just sufficient to apply **SimpleForm** with the adequate derivation δ_k (see Section 2.2.3).

Remark 2.8. *Consider a pseudo-linear system* (2.12) *with Poincaré rank* $p > 0$ *and suppose that the* t*-adic expansion of the matrix* M *is known up to an order* $m \geq n\,p$*. Rewriting the system as* (2.20) *and setting* $h = \nu(\det(A_k))$*, we find that* $h \leq n\,p$*. Computing a* k*-simple form of System* (2.12) *via Algorithm* **SimpleForm** *can then be done in* $O(m\,p\,n^4)$ *operations in* C*, while using the super-reduction algorithm [23], it can be done in* $O(m\,p\,n^4\min(n-1, p))$ *operations in* C *(see [41, Proposition 4.3]). Therefore, using Algorithm* **SimpleForm***, we gain the factor* $\min(n-1, p)$ *w.r.t. the super-reduction algorithm. This can be explained by the fact that the super-reduction algorithm provides* k*-simple forms for all* $k = 0, \ldots, p-1$*, while Algorithm* **SimpleForm** *provides a* k*-simple form for a single integer* k *between 0 and* $p-1$*.*

2.3.4 Super-reduction using simple forms

The following two lemmas give rise to a new method, based on successive computations of k-simple forms, to compute a super-irreducible form of a pseudo-linear system of the form (2.12).

Lemma 2.3. *For any $k \in \{1, \dots, p\}$, if $t^{p-\omega}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y})$ is a k-simple form of $t^{p-\omega}\,\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$, and $t^{p-\omega}\,\delta(\mathbf{y}) = \widehat{M}\,\phi(\mathbf{y})$ is a $(k-1)$-simple form of $t^{p-\omega}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y})$, then $t^{p-\omega}\,\delta(\mathbf{y}) = \widehat{M}\,\phi(\mathbf{y})$ is k-simple and $(k-1)$-simple.*

Proof. The proof is based on the results on the preservation of the simplicity developed in [34, Section 3] and [59, Section 4.5] for differential systems, which can be translated to pseudo-linear systems. □

Lemma 2.4. *If System (2.12) is k-simple for all values of $k \in \{0, \dots, p-1\}$, then it is super-irreducible.*

Proof. Let $A^{(k)}$ and $B^{(k)}$ be defined as in (2.18) and (2.19). One has

$$\det(A_0^{(k)}\,\lambda + B_0^{(k)}) = \theta_{p-k}(\lambda),$$

where $\theta_{p-k}(\lambda)$ is defined in (2.16). System (2.12) being k-simple for all values of $k \in \{0, \dots, p-1\}$, means that $\det(A_0^{(k)}\,\lambda + B_0^{(k)}) \neq 0$, and then it follows from Theorem 2.2 that System (2.12) is super-irreducible. □

Lemma 2.4 suggests the following algorithm for computing a super-irreducible form of a pseudo-linear system (2.12): compute first a $(p-1)$-simple form $t^{p-\omega}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y})$ of (2.12). Then compute a $(p-2)$-simple form $t^{p-\omega}\,\delta(\mathbf{y}) = \widehat{M}\,\phi(\mathbf{y})$ of $t^{p-\omega}\,\delta(\mathbf{y}) = \widetilde{M}\,\phi(\mathbf{y})$. Iterate this process until we get a 0-simple form. According to Lemma 2.3, the last system obtained is guaranteed to be k-simple for all values of $k \in \{0, \dots, p-1\}$, and thus super-irreducible.

Example 2.8. *Let us compute a super-irreducible form near the singularity 0 of a q-difference system $\mathbf{y}(q\,x) = W(x)\,\mathbf{y}(x)$ where $q \in C^*$ is not a root of unity, and*

$$W = \begin{bmatrix} 0 & \dfrac{1}{x^3} & q\,x^3 \\ x^2 + \dfrac{1}{x} & \dfrac{q}{x} & -q\,x^2 - 1 \\ \dfrac{1}{q\,x^2} - 1 & 0 & x \end{bmatrix}.$$

Define the local parameter $t = x$. The system can be written in the local form (2.12) as

$$x^3\,\delta(\mathbf{y}) = M\,\phi(\mathbf{y}) \tag{2.23}$$

where $\phi : x \mapsto x/q$, $\delta = \mathrm{id}_K - \phi$ *with degree* $\omega = 0$ *and*

$$M = x^3\left(\phi(W) - I_3\right) = \begin{bmatrix} -x^3 & q^3 & \frac{x^6}{q^2} \\ \frac{x^2\left(q^3+x^3\right)}{q^2} & x^2\left(q^2-x\right) & -\frac{x^3\left(x^2+q\right)}{q} \\ x\left(-x^2+q\right) & 0 & -\frac{x^3\left(-x+q\right)}{q} \end{bmatrix}.$$

Here the Poincaré rank is $p = 3$. *First we compute a 2-simple form. For this, we write* (2.23) *under the form* $A^{(2)}\,\delta_2(\mathbf{y}) + B^{(2)}\,\phi(\mathbf{y}) = 0$ *where*

$$\delta_2 = x^2\,\delta, \quad A^{(2)} = \mathrm{diag}(x,1,1), \quad B^{(2)} = -x^{-1}\,A^{(2)}\,M.$$

Using Algorithm **SimpleForm**, *we compute an equivalent simple system*

$$\widetilde{A^{(2)}}\,\delta_2(\mathbf{y}) + \widetilde{B^{(2)}}\,\phi(\mathbf{y}) = 0, \tag{2.24}$$

where

$$\widetilde{A^{(2)}} = I_3, \quad \widetilde{B^{(2)}} = \begin{bmatrix} -x\left(-x+q\right) & \frac{x\left(x^2+q\right)}{q} & \frac{-q^3-x^3}{q^2} \\ 0 & \frac{x^2\left(-x+q\right)}{q} & x^2-q \\ -q^2 & -\frac{x^5}{q^2} & x^2 \end{bmatrix}.$$

The system

$$x^3\delta(\mathbf{y}) = \widetilde{M}\phi(\mathbf{y}), \tag{2.25}$$

where $\widetilde{M} = -x\widetilde{A^{(2)}}^{-1}\widetilde{B^{(2)}}$, *is thus a 2-simple form of* (2.23). *Now we compute a 1-simple form of* (2.25). *For this, we write* (2.25) *under the form*

$$D^{(1)}\,\delta_1(\mathbf{y}) + N^{(1)}\,\phi(\mathbf{y}) = 0, \tag{2.26}$$

where $\delta_1 = x\,\delta$, $D^{(1)} = \mathrm{diag}(x,x,x)$ *and* $N^{(1)} = -x^{-2}\,D^{(1)}\,\widetilde{M}$. *Note that* (2.26) *can be directly obtained from* (2.24) *by setting* $D^{(1)} = x\,\widetilde{A^{(2)}}$ *and* $N^{(1)} = \widetilde{B^{(2)}}$. *Applying Algorithm* **SimpleForm** *to* (2.26) *produces an equivalent simple system*

$$\widetilde{D^{(1)}}\,\delta_1(\mathbf{y}) + \widetilde{N^{(1)}}\,\widetilde{\phi}(\mathbf{y}) = 0, \tag{2.27}$$

where

$$\widetilde{D^{(1)}} = \operatorname{diag}(1, x, x), \quad \widetilde{N^{(1)}} = \begin{bmatrix} x - q & \dfrac{-q^2 + 2\,x^2 + q}{q} & -\dfrac{x\,(q^2 + x)}{q^2} \\ 0 & \dfrac{x^2\,(-x + q)}{q} & x^2 - q \\ -q^2 & \dfrac{-x^5 - q^4}{q^2} & x^2 \end{bmatrix}.$$

The system

$$x^3\,\delta(\mathbf{y}) = \widehat{M}\,\phi(\mathbf{y}), \tag{2.28}$$

where

$$\widehat{M} = -x^2\,\widetilde{D^{(1)}}^{-1}\,\widetilde{N^{(1)}} = \begin{bmatrix} x^2\,(-x + q) & \dfrac{x^2\,(q^2 - 2\,x^2 - q)}{q} & \dfrac{x^3\,(q^2 + x)}{q^2} \\ 0 & -\dfrac{x^3\,(-x + q)}{q} & x\,(-x^2 + q) \\ q^2 x & \dfrac{x\,(x^5 + q^4)}{q^2} & -x^3 \end{bmatrix},$$

is thus a 1-simple form of (2.25). Note that System (2.28) is also 2-simple. Then, one can easily check that (2.28) is already 0-simple. Therefore, System (2.28) is an equivalent k-simple system of (2.23) for all $k = 0, \ldots, p-1$, and hence it is a super-irreducible form of (2.23). Finally, we can say that the system $\mathbf{y}(q\,x) = \widehat{W}(x)\,\mathbf{y}(x)$ where

$$\widehat{W}(x) = I_3 + q^{-3}x^{-3}\widehat{M}(qx) = \begin{bmatrix} \dfrac{1}{x} & \dfrac{-2\,q\,x^2 + q - 1}{x\,q} & \dfrac{q + x}{q} \\ 0 & x & \dfrac{-q\,x^2 + 1}{q\,x^2} \\ \dfrac{1}{x^2} & \dfrac{q\,x^5 + 1}{x^2} & 0 \end{bmatrix},$$

is a super-irreducible form of $\mathbf{y}(qx) = W(x)\,\mathbf{y}(x)$ near the singularity 0.

Our new method developed in the present section requires "p times" the computation of a k-simple form. Hence, computing a super-irreducible form of System (2.12) using our method can be done in at most $O(m\,p^2\,n^4)$ operations in C where m denotes the order of the t-adic expansion of the matrix M in (2.12) (see Remark 2.8). From a complexity perspective, the method developed in [23], which is implemented in the current version of the ISOLDE [39] package, seems to be better as it requires at most $O(m\,p\,n^4\,\min(n-1, p))$ operations in C (see [41, Proposition 4.3]). However, after performing some practical experiments, we have noticed an advantage of our method compared to the one in [23]. But we do believe that if (some of) the internal procedures of the ISOLDE package are updated, then the method in [23] will claim the practical advantage. We end this chapter by presenting, without commenting, the timings obtained from one of our experiments held for series of differential systems of sizes n and Poincaré ranks p. The first, respectively second, entry of each cell denotes the time (in seconds) obtained using our method, respectively the method from [23].

n \ p	5	10	30	50
3	0.16 | 0.09	0.26 | 0.25	0.65 | 0.62	1.16 | 1.91
5	0.26 | 0.21	0.54 | 0.47	1.29 | 2.15	1.98 | 4.76
10	1.31 | 1.21	2.32 | 2.93	5.60 | 11.81	9.19 | 28.43
15	2.70 | 2.33	8.71 | 10.85	20.58 | 36.62	34.87 | 87.53

Table 2.1: Computing a super-irreducible form of a differential system.

Chapter 3

On Rational Solutions of First Order Pseudo-Linear Systems

This chapter constitutes the subject of the first half of the published paper [33] in collaboration with M. A. Barkatou and T. Cluzeau.

3.1 Introduction

Let $F = C(x)$ be the field of rational functions in a variable x with coefficients in a field C of characteristic zero, ϕ be a C-automorphism over F and δ be a pseudo-derivation with respect to ϕ. The main objective of this chapter is to develop a unified algorithm to compute all rational solutions of a first order pseudo-linear system defined over the $\phi\delta$-field (F, ϕ, δ), having the form

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \tag{3.1}$$

where $M \in \mathbb{M}_n(F)$. The problem of computing rational solutions has been studied for the particular types of differential, difference, and q-difference systems respectively in [20], [21], and [5] (see also [2, 7, 95]). Our approach for computing rational solutions follows the same strategy as the one used in the latter works. This strategy mainly consists in two steps:

1. We first compute a so called *universal denominator* $u \in C[x]$ which satisfies that every rational solution $\mathbf{y} \in F^n$ of (3.1) can be written as

$$\mathbf{y} = \mathbf{z}/u, \quad \mathbf{z} \in C[x]^n.$$

2. We perform the change of variable $\mathbf{y} = \mathbf{z}/u$ and we compute polynomial solutions $\mathbf{z} \in C[x]^n$ of the new pseudo-linear system

$$\delta(\mathbf{z}) = \phi\left(u^{-1}\right)\left(u\,M + \delta(u)\right)\,\phi(\mathbf{z}), \tag{3.2}$$

which is of the same type as System (3.1).

To begin, in the next section we present a unified method for computing polynomial solutions of a general pseudo-linear system of the form (3.1). We show that the degree of a polynomial solution can be obtained by "inspecting" the integer roots of the indicial polynomial at infinity. However, this indicial polynomial is not immediately apparent, and so we propose to use simple forms to effectively compute it. The other parts (monomials) of the polynomial solution can afterwards be computed using existing methods in the literature. In Section 3.3, we develop a unified algorithm for computing a universal denominator for all rational solutions of System (3.1). We will see that in the case where ϕ is not the identity map, a universal denominator is composed of two parts: what we call the *ϕ-fixed part* and the *non ϕ-fixed part*. On one hand, the ϕ-fixed part can be computed using simple forms. On the other hand, following the ideas of [77] (see also [9, 72]), we propose an efficient algorithm for computing the non ϕ-fixed part. Examples of computations are also given to clarify our approaches.

The different algorithms proposed in this chapter are fully implemented in Maple in our PseudoLinearsystems package [32]. We shall also present throughout this chapter some experimental results obtained while comparing our implementations with the existing ones from the Isolde [39] and LinearFunctionalSystems[1] packages. A demonstration for our implementations is provided in Chapter 5.

3.2 A unified algorithm to compute polynomial solutions

We consider in this section the problem of computing all polynomial solutions of a first order pseudo-linear system of the form (3.1). In order to compute a polynomial solution $\mathbf{y} \in C[x]^n$, we carry out the following two tasks:

1. Compute a degree bound of the polynomial.
2. Once a degree bound is obtained, compute the different monomials $\alpha_i\, x^{\mu_i}$ ($\alpha_i \in C^n$, $\mu_i \in \mathbb{N}$) of the polynomial.

The second task is completely accomplished using the monomial-by-monomial method developed in [24], and we shall not explain more on this. We only focus on the task of computing a degree bound of a possible polynomial solution.

3.2.1 Computing a degree bound

Given a pseudo-linear system (3.1) defined over the $\phi\delta$-field (F, ϕ, δ). The idea is to imbed (F, ϕ, δ) in the local $\phi\delta$-field (K, ϕ, δ) where $K = C((x^{-1}))$, equipped with the t-adic valuation ν (here $t = x^{-1}$). Denote by ω the degree of δ (see Definition 1.12). Our system (3.1) can then be written as a local pseudo-linear system of the form

$$L(\mathbf{y}) = A\,\widetilde{\delta}(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{3.3}$$

where $\widetilde{\delta} = t^{-\omega}\,\delta$,

$$A = \operatorname{diag}(t^{\alpha_1}, \ldots, t^{\alpha_n}) \in \mathbb{M}_n(C[[t]]), \quad \alpha_i = -\min\{0, \nu\,(M(i,.)) - \omega\},$$

and

$$B = -t^{-\omega}\,A\,M \in \mathbb{M}_n(C[[t]]).$$

[1] https://www.maplesoft.com/support/help/Maple/view.aspx?path=LinearFunctionalSystems&cid=301

Here $M(i,.)$ denotes the i^{th} row of M. A polynomial solution of degree $s \in \mathbb{N}$ can be viewed as a local formal solution (at $x = \infty$) of the form

$$\mathbf{y}(t) = \sum_{i \geq 0} t^{i-s}\, \mathbf{y}_i, \tag{3.4}$$

where $\mathbf{y}_i \in C^n$ with $\mathbf{y}_0 \neq 0$. Recall that (see Definition 1.13) there exist $c, d \in C^*$ such that

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

and for $h \in \mathbb{Z}$:

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

where $[h]_c$ is defined by:

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

Replacing $\phi(\mathbf{y})$, $\widetilde{\delta}(\mathbf{y})$, A and B by their respective t-adic expansions in System (3.3) and comparing coefficients of the same power of t (after simplifying the factor t^{-s}) yields, amongst others, the equation

$$\left(d\, [-s]_c\, A_0 + c^{-s}\, B_0\right)\, \mathbf{y}_0 = 0,$$

where A_0 and B_0 are the constant terms in the t-adic expansions of A and B. Thus, in order that System (3.3) admits a formal solution (3.4), $-s$ must be a root (see Remark 2.3) of the indicial polynomial

$$\varphi(\lambda) = \det(d\, [\lambda]_c\, A_0 + c^{\lambda}\, B_0).$$

If System (3.3) is simple, which means $\det(A_0\, \lambda + B_0) \neq 0$, then $\varphi(\lambda) \neq 0$ and the degree of any polynomial solution can be bounded by the largest nonnegative integer s such that $-s$ is a root of $\varphi(\lambda)$. Otherwise if (3.3) is non-simple, then $\varphi(\lambda)$ vanishes identically in λ, and in this case one needs to compute a simple form of (3.3) in order to effectively compute $\varphi(\lambda)$. This can be done by applying Algorithm **SimpleForm** developed in Chapter 2 to System (3.3). Note that in previous works related to differential [20] and difference [7] systems, the method proposed to compute $\varphi(\lambda)$ consists in applying super-reduction algorithms [15, 41], while in the q-difference case [5], due to the absence of a super-reduction algorithm for q-difference systems in that period, Abramov proposed to use the technique of EG-eliminations [3] in order to compute $\varphi(\lambda)$. We present in Section 3.2.3 some experimental results obtained when comparing the three approaches.

Remark 3.1. *It is important to note here that computing a simple form of System* (3.3) *does not affect the degree bound of any polynomial solution of System* (3.1)*. Indeed, applying Algorithm* **SimpleForm** *to* (3.3) *yields an equivalent simple operator* $L' = S\,L\,T$ *where in particular* T *is an invertible matrix which is polynomial in* $t = x^{-1}$*. Moreover, due to the special forms of the entries of* T *(see the proof of Proposition 2.1), the matrix* T^{-1} *is polynomial in* x*. So if* $\mathbf{y}$ *is a polynomial solution of System* (3.1) *then* $\mathbf{z} = T^{-1}\,\mathbf{y}$ *is a polynomial solution of the equivalent system* $\delta(\mathbf{z}) = \widetilde{M}\,\phi(\mathbf{z})$ *where* $\widetilde{M} = T^{-1}\,(M\,\phi(T) - \delta(T))$*.*

3.2.2 Examples

We clarify our approach for computing polynomial solutions of pseudo-linear systems with the following two examples:

Example 3.1. *Let* $\phi : x \mapsto q\,x$ *be a* C*-automorphism over* F *such that* $q \in C^*$ *is not a root of unity, and let us compute the polynomial solutions of the* q*-difference system* $\phi(\mathbf{z}) = N\,\mathbf{z}$ *where*

$$N(x) = \begin{bmatrix} \dfrac{(q\,x+\beta)\,(q^2+1)}{\beta+x} & -\dfrac{(q\,x+\beta)\,q^2}{x} \\ \dfrac{q\,x}{\beta+x} & 0 \end{bmatrix}.$$

Here β *is a non-zero parameter. Note that for* $\beta = 100$*, we find again the system considered in the second part of [5, Section 4]. The given system can be written as a first order pseudo-linear system* $\delta(\mathbf{z}) = M\,\phi(\mathbf{z})$ *where* $\delta = \mathrm{id}_F - \phi$ *and*

$$M(x) = N(x)^{-1} - I_2 = \begin{bmatrix} -1 & \dfrac{\beta+x}{q\,x} \\ -\dfrac{x}{q^2\,(q\,x+\beta)} & \dfrac{-q^3+q^2+1}{q^3} \end{bmatrix}.$$

As explained above, we now work with $K = C((x^{-1}))$*: the completion of* F *w.r.t. the* t*-adic valuation* ν *(here* $t = x^{-1}$*). The degree of* δ *is thus* $\omega = 0$ *and one has* $c = q^{-1}$*,* $d = 1-q^{-1}$ *(see Definition 1.13). The system can be written in a local form* $A\,\delta(\mathbf{z}) + B\,\phi(\mathbf{z}) = 0$ *where* $A = I_n$ *and* $B = -M$*. The leading matrix pencil of the latter system*

$$L_\lambda = \begin{bmatrix} \lambda+1 & -q^{-1} \\ q^{-3} & \lambda - \dfrac{-q^3+q^2+1}{q^3} \end{bmatrix}$$

satisfies $\det(L_\lambda) \neq 0$*. Hence the system is already simple and its associated indicial* *polynomial is given by*

$$\varphi(\lambda) = 1 - q^{-\lambda-1} - q^{-\lambda-3} + q^{-2\lambda-4}.$$

The integer roots of $\varphi(\lambda)$ are -1 and -3 (see Remark 2.3) so that we get the degree bound $s = 3$. We then proceed by using the method in [24] to compute the monomials of the solutions, and we finally obtain a basis of polynomial solutions given by

$$\mathbf{z}_1(x) = \begin{bmatrix} \beta + x \\ x \end{bmatrix}, \quad \mathbf{z}_2(x) = \begin{bmatrix} x^2 + \dfrac{x^3}{\beta} \\ \dfrac{x^3}{q^2\beta} \end{bmatrix}. \tag{3.5}$$

The results are coherent with those obtained in [5, Section 4] for the case $\beta = 100$. Note that in [5, 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 is already simple at infinity.

Example 3.2. *Let $\phi : x \mapsto 3\,x + 2$ be a C-automorphism over F and $\delta = \mathrm{id}_F - \phi$ be a ϕ-derivation. Consider the pseudo-linear system*

$$\delta(\mathbf{y}) = \begin{bmatrix} -\dfrac{242\,x^3 + 410\,x^2 + 208\,x + 40}{9\,(3\,x+2)^2\,(3\,x+1)} & 0 \\ -\dfrac{2\,(x+1)^3\,(x-2)\,9\,(13\,x+2)}{(3\,x+2)^2\,(3\,x+1)\,x} & -\dfrac{26\,x+2}{27x} \end{bmatrix} \phi(\mathbf{y}). \tag{3.6}$$

We now work with $K = C((t))$ equipped with the t-adic valuation ν where $t = x^{-1}$. One thus has $\omega = 0$ with $c = \frac{1}{3}$ and $d = \frac{2}{3}$. System (3.6) can be written as a local pseudo-linear system $A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0$ where $A(x) = \mathrm{diag}(1, x^{-1})$ and

$$B(x) = \begin{bmatrix} \dfrac{242\,x^3 + 410\,x^2 + 208\,x + 40}{9\,(3\,x+2)^2\,(3\,x+1)} & 0 \\ \dfrac{2\,(x+1)^3\,(x-2)\,(13\,x+2)}{9x^2\,(3\,x+2)^2\,(3\,x+1)} & \dfrac{26\,x+2}{27x^2} \end{bmatrix}.$$

The latter system is non-simple since its leading matrix pencil

$$L_\lambda = \begin{bmatrix} \lambda + \dfrac{242}{243} & 0 \\ \dfrac{26}{243} & 0 \end{bmatrix},$$

is singular. Using Algorithm **SimpleForm***, we compute an equivalent simple system $I_n\delta(\mathbf{y}) + \widetilde{B}\phi(\mathbf{y}) = 0$ where*

$$\widetilde{B} = \begin{bmatrix} \frac{6318\,x^5+18548\,x^4+27804\,x^3+24254\,x^2+9648\,x+968}{117\,x(3\,x+2)^3(3\,x+1)} & \frac{-1019304\,x^5-1251624\,x^4+2872782\,x^3+5840670\,x^2+2986280\,x+326216}{4563\,x(3\,x+2)^3(3\,x+1)} \\ \frac{2(x+1)^3(x-2)(13\,x+2)}{9x(3\,x+2)^3(3\,x+1)} & \frac{36816\,x^5+76878\,x^4+32808\,x^3-27054\,x^2-21872\,x-2696}{351\,x(3\,x+2)^3(3\,x+1)} \end{bmatrix},$$

and with an associated indicial polynomial

$$\varphi(\lambda) = 1 - \frac{28}{729}\,3^{-\lambda} + \frac{3^{-2\,\lambda}}{19683}.$$

The integer roots of $\varphi(\lambda)$ are -6 and -3 (see Remark 2.3) so that we get the degree bound $s = 6$. Now using the monomial-by-monomial method [24] we can finally get the basis of polynomial solutions

$$\mathbf{y}_1(x) = \begin{bmatrix} x^5 - 5x^4 + 8x^3 - 4x^2 \\ x^6 - 3\,x^5 - x^4 + 17\,x + 14 \end{bmatrix}, \quad \mathbf{y}_2(x) = \begin{bmatrix} 0 \\ x^3 - 3\,x - 2 \end{bmatrix}. \qquad (3.7)$$

3.2.3 Comparison with existing implementations

The above explanations give rise to an algorithm, called **PolySols_1PLS**, which takes as an input a pseudo-linear system of the form (3.1) and return a matrix whose columns form a basis of all polynomial solutions, or 0_n (the zero vector of dimension n) if there are no non-trivial polynomial solutions. The algorithm is implemented in our Maple package PseudoLinearSystems [32].

Besides being generic enough to cover all types of pseudo-linear systems, Algorithm **PolySols_1PLS** appears to be, most of the times, faster than existing algorithms for computing polynomial solutions of familiar linear functional systems. To illustrate this fact, we present some timings of two of our experiments. All the input systems considered were constructed in such a way they had a full fundamental matrix of polynomial solutions whose all entries are generated using the Maple command:

```
randpoly(x, coeffs = rand(-20 .. 20), degree = 10, terms = 10);
```

The first experiment was carried out on several differential systems $\mathbf{y}'(x) = A(x)\,\mathbf{y}(x)$: we compare our implementation with the procedures **Mpolsolde** from the Isolde [39] package and **PolynomialSolution** from the LinearFunctionalSystems package. The second experiment was carried out on several difference systems $\mathbf{y}(x+1) = A(x)\,\mathbf{y}(x)$: we compare our implementation with the procedures **deltaPS** from the Isolde package and also **PolynomialSolution** from the LinearFunctionalSystems package. The two tables below show the respective CPU times (in seconds) needed by each implementation to compute all polynomial solutions of differential and difference systems with sizes $n = 5,\ 10,\ 15,\ 20$.

n	**PolySols_1PLS**	**Mpolsolde**	**PolynomialSolution**
5	0.459	0.953	0.451
10	3.950	9.832	4.722
15	23.846	59.158	24.139
20	78.481	227.034	86.412

Table 3.1: Results of the first experiment (differential systems).

n	**PolySols_1PLS**	**deltaPS**	**PolynomialSolution**
5	0.379	0.334	2.612
10	5.650	4.295	55.188
15	23.239	30.095	1385.85
20	90.323	99.686	$*$

Table 3.2: Results of the second experiment (difference systems).

Looking at the results of the experiment held for differential systems, we can spot a clear advantage of **PolySols_1PLS** compared to **Mpolsolde**. The only difference between the two implementations is that we use simple forms to compute a degree bound of polynomial solutions while **Mpolsolde** uses super-reduction [41]. We noticed that this is where most of the time is spent after going deeper into the analysis of each step of both implementations. On the other hand, **PolySols_1PLS** and **PolynomialSolution** appear to have a somehow similar performance (we couldn't analyse each step of the procedure **PolynomialSolution** since it is embedded in the Maple system).

The results of the second experiment show that **PolySols_1PLS** and **deltaPS** behave likewise (with a slight advantage of **PolySols_1PLS**), while **PolynomialSolution** behaves badly for difference systems (the symbol $*$ indicates that a computation did not terminate after 4 hours).

3.3 Universal denominator

In the general setting of pseudo-linear algebra with $F = C(x)$, two cases can be distinguished:

1. The case $\phi = \mathrm{id}_F$ corresponds to differential systems.

2. The case $\phi \neq \mathrm{id}_F$ corresponds to all systems of the form

$$\phi(\mathbf{y}) = N\,\mathbf{y}, \tag{3.8}$$

   where $N \in \mathrm{GL}_n(F)$ and $\phi : x \mapsto q\,x + r$. Here $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. We will refer to a system of the form (3.8) as a *ϕ-system*. Recall that such a system can be written (in various ways) as a pseudo-linear system $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$ (see Section 1.2). Systems of the form (3.8) include pure difference ($q = 1$ and $r \neq 0$) and pure q-difference ($r = 0$) systems.

In the differential case $\phi = \mathrm{id}_F$, assuming that $\delta = \frac{d}{dx}$ is the usual derivation of F, we have a linear differential system of the form $\mathbf{y}' = A\,\mathbf{y}$, with $' := \frac{d}{dx}$ and $A \in \mathbb{M}_n(F)$. We briefly review the method used in [20] for computing a universal denominator. Here, the poles of any rational solution are among the poles of the matrix A. Consequently, the denominator of any rational solution has the form

$$\prod_{i=1}^{s} p_i^{\alpha_i},$$

where $p_1, \ldots, p_s$ are the irreducible factors of the denominator $\mathrm{den}(A)$ of the matrix A and, for $i = 1, \ldots, s$, α_i is a *local exponent* at p_i. A universal denominator can thus be deduced from the knowledge of the local exponents at each p_i. In order to compute a local exponent α_i, one needs to compute the integer roots of the indicial polynomial $\varphi(\lambda)$ at p_i and take $-\alpha_i$ as the smallest integer root of $\varphi(\lambda)$. However, $\varphi(\lambda)$ can not be read off unless the system is in simple form. In [20], the method proposed to compute a simple form is to apply first super-reduction [67]. The **SimpleForm** algorithm developed in Chapter 2 thus provides an alternative as it allows to compute directly a simple form without recourse to super-reduction algorithms. For further reading on the computation of a universal denominator and rational solutions of differential systems, we recommend the reader to consult [20] and the references therein.

Concerning the case $\phi \neq \mathrm{id}_F$, algorithms for computing a universal denominator have been developed only for the pure difference [7, 21] and q-difference [5] cases. In Section 3.3.2, we shall develop a unified and efficient method for computing a universal denominator of a ϕ-system (3.8) in the case where the automorphism ϕ of $F = C(x)$ is given by $\phi : x \mapsto q\,x + r$, with $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. Note that this restriction on the automorphism ϕ of F is natural as, for the purposes of the present chapter, one needs ϕ to send polynomials to polynomials. Following the same lines as [21], we define the following two polynomials in the variable x from the denominators of the matrix $N \in \mathrm{GL}_n(F)$ of System (3.8) and its inverse:

$$a := \phi^{-1}(\mathrm{den}(N)), \quad b := \mathrm{den}(N^{-1}). \tag{3.9}$$

The *dispersion set* $E_\phi(a, b)$ of the polynomials a and b is defined as:

$$E_\phi(a, b) := \{s \in \mathbb{N}\,;\, \deg\left(\gcd(a, \phi^s(b))\right) > 0\}\,, \tag{3.10}$$

and plays an important role in the following. Note that the notion of dispersion set was firstly introduced by Abramov in [1]. Except in the pure difference case ($r \neq 0$ and $q = 1$) which is considered in Section 3.3.1 below, a universal denominator for rational solutions of a ϕ-system (3.8) is decomposed into two distinct parts, i.e., two polynomial factors. One part is called the *ϕ-fixed part* as it corresponds to the ϕ-fixed singularity $x_\phi := \frac{r}{1-q}$ (see Proposition 3.3 below) and the other part is called the *non ϕ-fixed part*. On one hand, the computation of the ϕ-fixed part can be tackled by computing a simple form at x_ϕ to get the local exponents at x_ϕ (it is similar to the computation of the part of a universal denominator corresponding to a given p_i in the differential case considered above). On the other hand, the non ϕ-fixed part can be computed from the dispersion set $E_\phi(a, b)$. Before developing our unified and efficient approach (see Section 3.3.2) to compute a universal denominator, we briefly recall how one proceeds in the known cases of pure difference and q-difference systems.

3.3.1 Existing methods for pure difference and q-difference systems

Let us consider a pure difference system of the form

$$\phi(\mathbf{y}) = N\,\mathbf{y}, \qquad \phi : x \mapsto x + 1, \tag{3.11}$$

where $N \in \mathrm{GL}_n(F)$. This means that we have $q = r = 1$. It has been shown in [21] that the irreducible factors of a universal denominator for System (3.11) are among the irreducible factors of a and b defined in (3.9) or their shifts[2]. Indeed, we have the following:

Proposition 3.1 ([21], Proposition 1). *Let $\mathbf{y} \in F^n$ be a rational solution of (3.11) and p be an irreducible polynomial in $C[x]$ such that p^s divides $\mathrm{den}(\mathbf{y})$ for some $s \in \mathbb{N}^*$. Let a and b be defined as in (3.9). Then:*

1. *If $\phi(p)$ does not divide $\mathrm{den}(\mathbf{y})$, then p^s divides a.*
2. *If $\phi^{-1}(p)$ does not divide $\mathrm{den}(\mathbf{y})$, then p^s divides b.*
3. *If both $\phi(p)$ and $\phi^{-1}(p)$ do not divide $\mathrm{den}(\mathbf{y})$, then p^s divides $\gcd(a, b)$.*

Consequently, it is natural to think that the different parts of a universal denominator for System (3.11) can be computed from the polynomials a and b. In fact we have the following result:

Proposition 3.2 ([21], Theorem 1). *Given a System (3.11). If its associated dispersion set $E_\phi(a, b)$ is empty, then $G(x) = 1$ is a universal denominator, i.e., all rational solutions*

[2] $f(x)$ is a shift of $g(x)$ if $f(x) = g(x + r)$ for some $r \in \mathbb{Z}^*$

are polynomials. Otherwise, a universal denominator for (3.11) *is given by:*

$$G(x) = \gcd\left(\prod_{i=0}^{h} \phi^{-i}(a(x)), \prod_{j=0}^{h} \phi^{j}(b(x))\right), \quad h := \max(E_\phi(a,b)). \tag{3.12}$$

Remark 3.2. *The method proposed in [7, 21] to compute $E_\phi(a,b)$ consists in computing first the resultant $\mathrm{Res}_x(a, \phi^m(b))$ which is a polynomial in m. Then the elements of $E_\phi(a,b)$ are exactly the roots of $\mathrm{Res}_x(a, \phi^m(b))$.*

From a computational point of view, the algorithm developed in [21, Proposition 3] (see also [7, Section 3.1]) allows to compute a universal denominator without expanding the products in Formula (3.12). For the sake of completeness, we shall present it here (we call it Algorithm **UD** as in [7]):

Algorithm **UD**

Input: A system of the form (3.11).
Output: A universal denominator for all rational solutions of (3.11).
1. Compute the polynomials a and b as in (3.9), and the dispersion set $E_\phi(a,b)$.
2. Initialize $v = a$, $w = b$, $G(x) = 1$ and $E = E_\phi(a,b)$.
While E is not empty **do**
- Set $i = \max(E)$.
- Compute $d = \gcd(v, \phi^i(w))$, $v = v/d$ and $w = w/\phi^{-i}(d)$.
- Put $G(x) = G(x)\prod_{j=0}^{i} \phi^{-j}(d)$.
- Update $E = E \setminus \{i\}$.

End While
Return $G(x)$.

Let us now consider a pure q-difference system of the form

$$\phi(\mathbf{y}) = N\,\mathbf{y}, \qquad \phi : x \mapsto q\,x, \tag{3.13}$$

where $q \in C^*$ is not a root of unity, and $N \in \mathrm{GL}_n(F)$. The computation of rational solutions of q-difference systems is studied in [5]. One might think at first sight, due to the similar forms of (3.13) and (3.11), that a universal denominator of a q-difference system can be computed in the same way as the pure difference case. However, this is not true. The polynomial $p = x$ is the only monic irreducible polynomial that is fixed by ϕ in the sense that p and $\phi(p)$ divide each other. More generally, p and $\phi^j(p)$ divide each other for all $j \in \mathbb{Z}$. Consequently, if $\mathbf{y} \in F^n$ is a rational solution of (3.13) and $p^s = x^s$ divides $\mathrm{den}(\mathbf{y})$ for some $s \in \mathbb{N}^*$, then for sure both $\phi(p)$ and $\phi^{-1}(p)$ divide $\mathrm{den}(\mathbf{y})$. Therefore, the hypotheses of Proposition 3.1 are not valid for the polynomial $p = x$. This is why a universal denominator for a q-difference system 3.13 is written under the form

$$x^{\alpha}\, G(x),$$

where $\alpha \in \mathbb{N}$ and $G(x) \in C[x]$ is not divisible by x. The factor x^{α} is thus what we call the ϕ-fixed part of a universal denominator for a q-difference system. A bound for α can be obtained from the local exponents at the ϕ-fixed singularity $x_{\phi} = 0$. In [5], the technique proposed to compute a bound for α is EG-eliminations [3]. In this thesis, we propose to use simple forms. This can be done first by imbedding F into the local field $K = C((t))$ where $t = x$ and rewriting (3.13) as a local pseudo-linear system of the form (2.1). Applying next Algorithm **SimpleForm** to this system yields an equivalent simple system with an associated indicial polynomial $\varphi(\lambda)$ which is non-zero. Finally, α is the largest non-negative integer such that $-\alpha$ is a root of $\varphi(\lambda)$ (see Remark 2.3).

The other factor $G(x)$ is what we call the non ϕ-fixed part. It can be computed as in the pure difference case using the formula (3.12) in Proposition 3.2 above (with the appropriate ϕ). Equivalently, $G(x)$ can be seen as the output of Algorithm **UD** applied to (3.13). Moreover, following the ideas of [77] (see also [9, 72]), we propose in Section 3.3.4 an alternative efficient algorithm for computing the non ϕ-fixed part. The reader can consult [3, 5] for additional details concerning universal denominators and rational solutions of q-difference systems.

Example 3.3. *Let us go back to the q-difference system considered in Examples 2.3 and 2.4. We remind the reader that this is the system*

$$\phi(\mathbf{y}) = \begin{bmatrix} \dfrac{q^2+1}{q} & -\dfrac{(\beta+x)\,q}{x} \\ \dfrac{x}{q\,x+\beta} & 0 \end{bmatrix} \mathbf{y}, \qquad \phi : x \mapsto q\,x,$$

where $q \in C^*$ is not a root of unity and β is a non-zero parameter. As mentioned above, a universal denominator for all rational solutions can be written as $u(x) = x^{\alpha}\, G(x)$, where $\alpha \in \mathbb{N}$ and $G(x) \in C[x]$ is not divisible by x. Let us first compute α. Instead of using EG-eliminations as it is done in [5, Section 4], we will use the **SimpleForm** *algorithm. Let $K = C((t))$ where $t = x$ is the local parameter, and we introduce the ϕ-derivation $\delta = \phi - \mathrm{id}_K$. We then have $\omega = 0$ (the degree of δ), $c = q$ so that $c \neq 1$ and $d = q - 1$ (see Definition 1.13). The system can be written as the local pseudo-linear system $A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0$ defined over the local $\phi\delta$-field (K, ϕ, δ), where A and B happen to be given by (2.10). From Examples 2.3 and 2.4, the output of Algorithm* **SimpleForm** *provided an equivalent simple system $A^{(5)}\delta(\mathbf{y}) + B^{(5)}\phi(\mathbf{y})$ where $A^{(5)}$ and $B^{(5)}$ are given* *by* (2.11) *and with*

$$\det(L_\lambda) = (\lambda\, q - q + 1)\,(\lambda + q - 1)\, q^{-1},$$

which yields the indicial polynomial

$$\varphi(\lambda) = q^{2\,\lambda} - q^{\lambda-1} - q^{\lambda+1} + 1.$$

The roots of $\varphi(\lambda)$ *are thus* $\lambda = \pm 1$ *(see Remark 2.3), which implies that* $\alpha = 1$*. On the other hand, applying Algorithm* **UD** *on the given system, we get that* $G(x) = x + \beta$*. It follows that a universal denominator is given by*

$$u(x) = x\,(x + \beta).$$

One then can proceed to compute a basis of rational solutions. The change of variable $\mathbf{y} = \mathbf{z}/u$ *yields the* q*-difference system considered in Example 3.1. We have seen that this system admits a basis of polynomial solutions given by* (3.5)*. Finally, we obtain the rational solutions of the original system:*

$$\mathbf{y}_1(x) = \begin{bmatrix} \dfrac{1}{x} \\ \\ \dfrac{1}{\beta + x} \end{bmatrix}, \quad \mathbf{y}_2(x) = \begin{bmatrix} \dfrac{x}{\beta} \\ \\ \dfrac{x^2}{(\beta + x)\,\beta\, q^2} \end{bmatrix}.$$

The results are coherent with those obtained in [5, Section 4] for the case $\beta = 100$*.*

3.3.2 A unified and efficient approach for pseudo-linear systems

We consider a ϕ-system (3.8), where the automorphism ϕ of $F = C(x)$ is given by $\phi : x \mapsto q\,x + r$, with $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. Let us first remark that in the case $r \neq 0$ and $q \neq 1$ is not a root of unity, performing the change of independent variable $x = z - \frac{r}{1-q}$, we are reduced to a pure q-difference system. In other words, after performing (if necessary) a change of independent variables, the computation of a universal denominator for the class of ϕ-systems considered here can always be done using one of the algorithms recalled in Section 3.3.1 for the pure difference and q-difference cases. However, in the following, we prefer to develop a unified approach treating directly all ϕ-systems.

As for pure q-difference systems, we shall decompose a universal denominator as a product of two factors: the ϕ-fixed part and the non ϕ-fixed part. To achieve this, we first need to determine the polynomials that are fixed by ϕ. We say that two polynomials p_1 and p_2 in $C[x]$ are *associated*, and we write $p_1 \sim p_2$, if they divide each other. We introduce the set

$$F_\phi := \{p \in C[x] \setminus \{0\} \; ; \; \deg(p) \geq 1, \; \exists s \in \mathbb{N}^*, \, p \sim \phi^s(p)\} \, ,$$

where $\phi^s(p(x)) = p(\phi^s(x))$. We remark that

$$\forall s \in \mathbb{N}, \quad \phi^s(x) = q^s \, x + r \, [s]_q, \quad [s]_q := \begin{cases} \dfrac{q^s - 1}{q - 1} & ; \quad q \neq 1, \\ s & ; \quad q = 1. \end{cases} \tag{3.14}$$

Proposition 3.3. *With the previous notation, we have the following:*

1. *If $q = 1$, then $F_\phi = \emptyset$.*
2. *Otherwise, $F_\phi = \left\{ c \left(x - \dfrac{r}{1-q} \right)^s \; ; \; c \in C^*, \; s \in \mathbb{N}^* \right\}$.*

Proof. If $q = 1$, then $p \sim \phi^j(p)$ for some $j \neq 0$ if and only if p is a constant and we are done. Now let $q \neq 1$. From (3.14), we have that, for all $j \in \mathbb{N}^*$,

$$\phi^j(x) = q^j \, x + r \, \frac{q^j - 1}{q - 1} = \tilde{q} \, x + \tilde{r}$$

has the same form as $\phi(x) = q \, x + r$ so that it suffices to look for non constant polynomials p such that $p \sim \phi(p)$. Let us write $p(x) = \sum_{i=0}^{s} p_i \, x^{s-i}$ with $p_0 = 1$ and $s \geq 1$. Then $p \sim \phi(p)$ means that there exists $\alpha \in C^*$ such that $\phi(p) = \alpha \, p$ which yields

$$(q \, x + r)^s + p_1 \, (q \, x + r)^{s-1} + \cdots + p_s = \alpha \, (x^s + p_1 \, x^{s-1} + \cdots + p_s).$$

By expanding the lefthand side of the latter equality and equating the coefficients of each x^i, $i = 0, \ldots, s$, we get

$$q^s = \alpha, \quad \forall i = 1, \ldots, s, \; (q^i - 1) \, p_i = \sum_{j=0}^{i-1} \binom{s-j}{s-i} r^{i-j} \, p_j.$$

Solving the latter linear system successively for $p_1, p_2, \ldots, p_s$, we obtain

$$\forall i = 1, \ldots, s, \quad p_i = \binom{s}{i} \left(\frac{r}{q-1} \right)^i,$$

which yields

$$p(x) = \left(x - \frac{r}{1-q} \right)^s . \tag{3.15}$$

Finally, F_ϕ contains only polynomials of the form (3.15) up to a multiplicative constant $c \in C^*$. $\square$

From Proposition 3.3, for $q = 1$ and $r \neq 0$, the set F_ϕ is empty which justifies why in the pure difference case, one does not have to consider a ϕ-fixed part in a universal denominator. Moreover, for the pure q-difference case $q \neq 1$ and $r = 0$, Proposition 3.3 implies that the only monic irreducible element in F_ϕ is $p = x$ meaning that the only ϕ-fixed singularity is $x_\phi = 0$. In the general case with $q \neq 1$, the only monic irreducible element in F_ϕ is $x - \frac{r}{1-q}$ and we thus write a universal denominator as a product

$$\left(x - \frac{r}{1-q}\right)^\alpha G(x), \tag{3.16}$$

where $\alpha \in \mathbb{N}$ and the polynomial $G(x)$ is not divisible by $x - \frac{r}{1-q}$. Here, $\left(x - \frac{r}{1-q}\right)^\alpha$ is the ϕ-fixed part and $G(x)$ is the non ϕ-fixed part. In order to construct a universal denominator, one needs to compute both α and $G(x)$ in (3.16). On one hand, α can be obtained from a simple form at the ϕ-fixed singularity $x_\phi = \frac{r}{1-q}$. This will be the subject of the next section.

On the other hand, the non ϕ-fixed part $G(x)$ can be obtained using Proposition 3.2 (with the appropriate ϕ). Equivalently, applying Algorithm **UD** from Section 3.3.1 to System (3.8) allows to compute $G(x)$. To achieve this, one first needs to compute the dispersion set $E_\phi(a, b)$ defined by (3.10), which is usually done by a resultant computation (see Remark 3.2 and [5]), and then, for each element in $E_\phi(a, b)$, several gcd's are computed in order to get $G(x)$. However, in [77] (see also [9, 72]), the authors remark that if we first compute a factorisation of the polynomials a and b, then $E_\phi(a, b)$ can be computed without computing resultants, which is often more efficient in practice. This fact is assured in [9, 77] by complexity analysis and experimental evidences. In Section 3.3.4, we give a unified version of the latter efficient approach for all ϕ-systems.

3.3.3 Computing the ϕ-fixed part

As we have seen above, a universal denominator of a ϕ-system (3.8) can be written as (3.16): a product of ϕ-fixed part and non ϕ-fixed part. We focus here on the computation of the ϕ-fixed part, mainly the computation of the non-negative ineteger α in (3.16). As in the q-difference case, we use Algorithm **Simpleform**. We introduce the ϕ-derivation $\delta = \mathrm{id}_F - \phi$. System (3.8) can thus be written as the pseudo-linear system $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$ where $M = N^{-1} - I_n \in \mathbb{M}_n(F)$.

Remark 3.3. *In our implementations, we prefer to write System (3.8) as the pseudo-linear system $\delta(\mathbf{y}) = M\,\widetilde{\phi}(\mathbf{y})$ where $\widetilde{\phi} = \phi^{-1}$, $\delta = \mathrm{id}_F - \widetilde{\phi}$ and $M = \widetilde{\phi}(N) - I_n$. From a computational point of view, this is better since we avoid matrix inversion when we compute M.*

Let $K = C((t))$ with $t = x - x_\phi$, be the completion of F w.r.t. to the t-adic valuation ν. The degree of δ is thus $\omega = 0$ and the system $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$ can be written as the local pseudo-linear system

$$A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0, \tag{3.17}$$

where

$$A = \operatorname{diag}(t^{\alpha_1}, \ldots, t^{\alpha_n}), \quad \alpha_i = -\min\{0, \nu(M(i,.))\}, \quad B = -A\,M \in \mathbb{M}_n(C[[t]]).$$

Applying Algorithm **SimpleForm** developed in Chapter 2 to System (3.17) yields an equivalent simple system with an associated indicial polynomial $\varphi(\lambda)$. Finally, α is the largest non-negative integer root such that $-\alpha$ is a root of $\varphi(\lambda)$ (see Remark 2.3). The algorithm to compute the ϕ-fixed part of a universal denominator for System (3.8) can be summarised in the following scheme. The input is either a ϕ-system (3.8) or a pseudo-linear system (3.1) where $\phi : x \mapsto q\,x + r$ and $\delta = \gamma(\mathrm{id}_F - \phi)$ with $\gamma \in F^*$. Here $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be 1.

Algorithm **FixedPart**

Input: A ϕ-system (3.8) or System (3.1) with the above conditions.
Output: The ϕ-fixed part of a universal denominator.
1. **Localisation:** Rewrite the input system as the local pseudo-linear system (3.17).
2. Apply Algorithm **SimpleForm** to (3.17) to obtain the indicial polynomial $\varphi(\lambda)$.
3. Compute $\alpha = \max\{\, \lambda \in \mathbb{N} \quad ; \quad \varphi(-\lambda) = 0 \,\}$.

Return $\left(x - \dfrac{r}{1-q}\right)^{\alpha}$.

Example 3.4. *Consider the ϕ-system (3.8) with $q = 3$, $r = 2$ and*

$$N = \begin{bmatrix} \dfrac{3\,x+2}{9\,x} & 0 \\ \dfrac{2\,(x+1)^3\,(13\,x+2)}{3\,(3\,x+2)\,(3\,x+1)\,x} & \dfrac{(x-1)\,(x-2)}{3\,(3\,x+2)\,(3\,x+1)} \end{bmatrix}.$$

The only ϕ-fixed singularity is $x_\phi = -1$ and thus we write a universal denominator under the form

$$u(x) = (x+1)^{\alpha}\,G(x),$$

where $G(x)$ is not divisible by $x+1$. Let us compute α. The idea is to work with $K = C((t))$ with $t = x+1$, equipped with the t-adic valuation ν. We introduce the ϕ-derivation $\delta = \mathrm{id}_K - \phi$, and then we have $\omega = 0$ (the degree of δ), $c = q$ and $d = 1 - q$ (see Definition 1.13). The system can be rewritten as the local pseudo-linear system

$A\,\delta(\mathbf{y}) + B\,\phi(\mathbf{y}) = 0$ *where* $A = I_2$ *and*

$$B = \begin{bmatrix} -\dfrac{6\,x-2}{3\,x+2} & 0 \\ \dfrac{18\,(x+1)^3\,(13\,x+2)}{3\,x^3-7\,x^2+4} & -\dfrac{26\,x^2+30\,x+4}{(x-1)\,(x-2)} \end{bmatrix}.$$

The latter system is already simple, and the indicial polynomial is

$$\varphi(\lambda) = \left(-1+9\,3^{\lambda}\right)\left(-1+3^{\lambda}\right).$$

The only non-negative integer root of $\varphi(\lambda)$ *is -2 which implies that* $\alpha = 2$.

3.3.4 Computing the dispersion set and the non ϕ-fixed part

Let us consider a ϕ-system (3.8). The dispersion set $E_\phi(a,b)$ defined by (3.10) is usually computed as follows. One first compute the resultant $\mathrm{Res}_x(a, \phi^m(b))$. This resultant is a polynomial in $[m]_q$ defined in (3.14) and the elements of $E_\phi(a,b)$ are computed from the roots in C of this polynomial. Note that if $d_1 = \deg(a)$ and $d_2 = \deg(b)$, then the resultant $\mathrm{Res}_x(a, \phi^m(b))$ is of degree $d_1\,d_2$ (see for instance [77, Proposition 1]). Moreover, its coefficients are expected to be significantly larger than those of a and b. Consequently, this resultant-based algorithm to compute the dispersion set appears to be inefficient in practice.

In this section, we extend the ideas of [77] to compute the dispersion set $E_\phi(a,b)$ for any automorphism ϕ of F defined by $\phi : x \mapsto q\,x + r$ with $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. The approach relies on a factorization into irreducible factors of the polynomials a and b given in (3.9). Moreover, we will see that this approach allows to compute directly all the factors of the non ϕ-fixed part of a universal denominator (this means while computing $E_\phi(a,b)$), unlike the existing resultant-based algorithms who require the computation of several gcd's afterwards (see Algorithm **UD** above).

First note that there exists $s \in \mathbb{N}$ such that $\deg\left(\gcd(a, \phi^s(b))\right) > 0$, this means $s \in E_\phi(a,b)$, if and only if there exist an irreducible factor f of a and an irreducible factor g of b such that $f \sim \phi^s(g)$. Then, we have the following result:

Proposition 3.4. *Let us consider two monic irreducible polynomials f and g of the same degree d and write $f(x) = \sum_{i=0}^{d} f_i\,x^{d-i}$, $f_0 = 1$, $g(x) = \sum_{i=0}^{d} g_i\,x^{d-i}$, $g_0 = 1$. If $f \sim \phi^s(g)$, then we have the following explicit formulas for s:*

1. *If* $q = 1$, *then* $s = \dfrac{f_1 - g_1}{d\, r}$ *(see [77]).*

2. *Otherwise, if* f *and* g *are both different from* $x - \frac{r}{1-q}$, *then if* k *denotes the smallest positive integer such that* $(q-1)^k\, f_k - \binom{d}{k}\, r^k \neq 0$, *we have*

$$s = \frac{\log(A_k)}{k\, \log(q)}, \qquad A_k := 1 + \frac{(q-1)^k\,(g_k - f_k)}{(q-1)^k\, f_k - \binom{d}{k}\, r^k}. \tag{3.18}$$

Proof. If $f \sim \phi^s(g)$, then necessarily $\phi^s(g) = q^{d\,s}\, f$. Now, a direct calculation shows that

$$\phi^s(g) = \sum_{k=0}^{d} \sum_{i=d-k}^{d} q^{s\,(d-k)} \binom{i}{d-k}\, g_{d-i}\, r^{i-d+k}\, [s]_q^{i-d+k}\, x^{d-k}.$$

Therefore, equating the coefficients of x^{d-k} in the equality $\phi^s(g) = q^{d\,s}\, f$, for $k \in \{1, \dots, d\}$, yields an equation of degree k in $[s]_q$ which can be written as:

$$f_k - g_k + \sum_{i=1}^{k} \left(f_k \binom{k}{i} (q-1)^i - \binom{d-k+i}{i} r^i\, g_{k-i} \right) [s]_q^i = 0. \tag{3.19}$$

For $k = 1$, Equation (3.19) implies

$$\Big((q-1) f_1 - d\, r\Big)\, [s]_q + f_1 - g_1 = 0. \tag{3.20}$$

If $q = 1$, then (3.20) yields $[s]_q = s = \dfrac{f_1 - g_1}{d\, r}$ which was also the result obtained in [77]. Otherwise, when $q \neq 1$, it may happen (namely, when $f_1\,(q-1) - d\, r = 0$) that the coefficient of $[s]_q$ in (3.20) vanishes which implies $g_1 = f_1$ and in this case Equation (3.19) for $k = 1$ will not provide any formula for $[s]_q$.
Let k be the smallest positive integer such that

$$f_k\,(q-1)^k - \binom{d}{k}\, r^k \neq 0.$$

Such a k always exists as, by hypothesis, $f(x) \neq x - \frac{r}{q-1} \notin F_\phi$. From Equation (3.19), we then have that, for all $i = 1, \dots, k$,

$$g_{k-i} = f_{k-i} = \frac{\binom{d}{k-i} r^{k-i}}{(q-1)^{k-i}}.$$

Moreover, Equation (3.19) has then exactly degree k in $[s]_q$ and can be simplified to get:

$$f_k - g_k + \left(f_k - \frac{\binom{d}{k}\, r^k}{(q-1)^k} \right) \Big((1 + (q-1)\,[s]_q)^k - 1 \Big) = 0.$$

Finally, using the definition (3.14) of $[s]_q$, we obtain $q^{s\,k} = A_k$ where A_k is defined in the statement of the proposition. This ends the proof. $\square$

Proposition 3.4 leads to an efficient unified algorithm for computing the dispersion set. Note also that, for our purpose, an important advantage of this approach, compared to resultant based algorithms that still need gcd's calculations, is that it also provides directly the factors of the non ϕ-fixed part of a universal denominator of the rational solutions of a ϕ-system. This is summarised in the following scheme. We call the algorithm **NonFixedPart**. Note that a (somehow) similar algorithm is developed in [9] for the pure-difference case.

Algorithm **NonFixedPart**

Input: A ϕ-system of the form (3.8).
Output: The dispersion set $E_\phi(a, b)$ of a and b defined by (3.9) and the non ϕ-fixed part of a universal denominator for rational solutions of (3.8).

1. Set $E_\phi(a, b) = \emptyset$ and $G(x) = 1$.

2. Factor a and b defined by (3.9) as products of powers of distinct monic irreducible polynomials called respectively u_j's and v_l's.

3. **For** each pair (u_j, v_l) such that $\deg(v_l) = \deg(u_j) = d$
   (we write $u_j(x) = \sum_{i=0}^{d} f_i\, x^{d-i}$, $v_l(x) = \sum_{i=0}^{d} g_i\, x^{d-i}$ - see Proposition 3.4)

   - **If** $q = 1$, then $s = \dfrac{f_1 - g_1}{d\, r}$.
     Else let k be the smallest positive integer such that
     $$f_k\,(q-1)^k - \binom{d}{k}\, r^k \neq 0$$
     and s be as in (3.18).
     End If

   - **If** $s \in \mathbb{N}$ and $u_j \sim \phi^s(v_l)$, then set $E_\phi(a, b) = E_\phi(a, b) \cup \{s\}$ and
     $$G(x) = G(x) \prod_{i=0}^{s} \phi^{-i}(u_j).$$
     End If

   End For

4. **Return** $E_\phi(a, b)$ and $G(x)$.

Example 3.5. *Let us come back to the ϕ-system considered in Example 3.4. We have already computed the ϕ-fixed part so that a universal denominator is of the form*

$$u(x) = (x+1)^2\, G(x),$$

where $G(x)$ is not divisible by $x+1$. Now we shall compute the non ϕ-fixed part $G(x)$. The factorisations of the polynomials a and b defined in (3.9) are given by:

$$a(x) = x\,(x-1)\,(x-2)\,, \quad b(x) = (x-1)\,(x-2)\left(x+\frac{2}{3}\right).$$

Here, inspecting directly pairs of irreducible factors of a and b, we easily check that:

$$x \sim \phi^1(x-2), \quad x-1 \sim \phi^0(x-1), \quad x-2 \sim \phi^0(x-2),$$

are the only possible associations. The dispersion set is thus $E_\phi(a,b) = \{0,1\}$ and the multiple of $G(x)$ obtained is $\phi^0(x)\,\phi^{-1}(x)\,\phi^0(x-1)\,\phi^0(x-2) = x\,(x-1)\,(x-2)^2$ because $\phi^{-1}(x) = \frac{1}{3}(x-2)$. A universal denominator is thus given by

$$u(x) = (x+1)^2\, x\,(x-1)\,(x-2)^2.$$

One then can compute a basis of rational solutions. We perform the change of variable $\mathbf{y} = \mathbf{z}/u$. This a yields a new ϕ-system of the form

$$\phi(\mathbf{y}(x)) = \begin{bmatrix} \dfrac{(27\,x+9)\,(3\,x+2)^2}{(x-1)\,(x-2)^2} & 0 \\ \dfrac{(702\,x+108)\,(x+1)^3}{(x-1)\,(x-2)^2} & \dfrac{27\,x}{x-2} \end{bmatrix} \mathbf{y}(x),$$

for which we are reduced to compute its polynomial solutions. The latter system can be written as the pseudo-linear system (3.6), and we have seen that it admits a basis of polynomial solutions given by (3.7). Finally, a basis of the rational solutions of the original ϕ-system is given by:

$$\mathbf{y}_1(x) = \begin{bmatrix} \dfrac{x}{(x+1)^2} \\ 1 \end{bmatrix}, \quad \mathbf{y}_2(x) = \begin{bmatrix} 0 \\ \dfrac{1}{x\,(x-1)\,(x-2)} \end{bmatrix}.$$

3.4 The complete algorithm

The results developed in the previous sections of this chapter naturally yield a generic algorithm to compute all rational solutions of a pseudo-linear system of the form (3.1), where either $\phi = \mathrm{id}_F$ and $\delta = \frac{d}{dx}$, or $\phi : x \mapsto q\,x + r$ and $\delta = \gamma(\mathrm{id}_F - \phi)$ with $\gamma \in F^*$. Here $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. We call the algorithm **RatSols_1PLS** and it proceeds as follows:

Algorithm **RatSols_1PLS**

Input: A pseudo-linear system (3.1): $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$.
Output: A matrix whose columns form a basis of all rational solutions, or 0_n (the zero vector of dimension n) if there are no non-trivial rational solutions.
1. Compute a universal denominator:

If $\phi = \mathrm{id}_F$ (differential systems) then for each irreducible factor p_i of $\mathrm{den}(M)$, use Algorithm **SimpleForm** to compute the corresponding local exponent α_i, and set

$$u(x) = \prod p_i^{\alpha_i}.$$

Else

- Rewrite (3.1) as a ϕ-system (3.8).
- Let $G(x)$ be the polynomial output of Algorithm **NonFixedPart** applied to (3.8).
- **If** $q = 1$ then set $u(x) = G(x)$.
  Else
  * Let $\left(x - \frac{r}{1-q}\right)^{\alpha}$ be the output of Algorithm **FixedPart** applied to (3.1).
  * Set $u(x) = \left(x - \frac{r}{1-q}\right)^{\alpha} G(x)$.

  End If

End If

2. Compute polynomial solutions: Let Z be the output of applying Algorithm **PolySols_1PLS** (see Section 3.2) to System (3.2). Here Z is either a matrix whose columns form a basis of all polynomial solutions or the vector 0_n.

Return $u^{-1}\,Z$.

3.5 Some comparison tests

We present in this section some of the experimental results that we have obtained when comparing the performance of our implementation to that of existing algorithms. For our knowledge, the (implemented) existing algorithms only deals with the three particular systems: differential, difference, and q-difference. For this reason, the experiments were held on these three kind of systems. More precisely:

- Experiment 1 held for differential systems $\mathbf{y}'(x) = A(x)\,\mathbf{y}(x)$.
- Experiment 2 held for difference systems $\mathbf{y}(x+1) = A(x)\,\mathbf{y}(x)$.
- Experiment 3 held for q-difference systems $\mathbf{y}(q\,x) = A(x)\,\mathbf{y}(x)$.

All input systems in each experiment have been constructed from a full fundamental matrix $Y(x)$ of rational solutions such that the numerator of each entry of $Y(x)$ is generated by the Maple command `randpoly(x, degree = 5)` and such that the denominator $\mathrm{den}(Y)$ is fixed for each experiment. Moreover, all the implementations used in the experiments perform two main computational tasks (universal denominator + polynomial solutions), so it is also interesting to compare the times needed in each implementation to accomplish each task. We denote by

- t_1: the time needed to compute a universal denominator.
- t_2: the time needed to compute the polynomial solutions (after the suitable change of variable).
- T: the overall time needed to compute all rational solutions (normally T is almost equal to $t_1 + t_2$).

Experiment 1: differential systems

Here we chose $\mathrm{den}(Y) = x$, and we have tested matrix dimensions $n = 5, 7, 9, 11, 12$. The matrix $A(x)$ of each input system thus satisfies

$$\mathrm{den}(A) = x\,p_n(x),$$

where $p_n(x)$ is an irreducible (large) polynomial that changes as n changes. For instance, for $n = 7$ we have

$$\begin{aligned} p_7(x) = {} & 21\,x^{34} - 50\,x^{33} - 97\,x^{32} - 141\,x^{31} - 50\,x^{30} + 479\,x^{29} + 340\,x^{28} + 61\,x^{27} + 359\,x^{26} \\ & - 62\,x^{25} - 788\,x^{24} + 2125\,x^{23} + 1293\,x^{22} - 741\,x^{21} - 448\,x^{20} + 2602\,x^{19} \\ & - 629\,x^{18} - 6338\,x^{17} + 5831\,x^{16} + 2298\,x^{15} - 8617\,x^{14} + 2683\,x^{13} + 2333\,x^{12} \\ & - 4118\,x^{11} + 912\,x^{10} + 2188\,x^{9} - 916\,x^{8} - 1001\,x^{7} + 363\,x^{6} + 375\,x^{5} - 60\,x^{4} \\ & - 73\,x^{3} + 18\,x^{2} + 14\,x - 1. \end{aligned}$$

We compare our implementation with the procedures **Mratsolde** from the ISOLDE [39] package and **RationalSolution** from the LINEARFUNCTIONALSYSTEMS package. The table below shows the results of Experiment 1 in terms of the CPU time (in seconds).

	RatSols_1PLS			**Mratsolde**			**RationalSolution**		
	t_1	t_2	T	t_1	t_2	T	t_1	t_2	T
$n=5$	0.27	0.09	0.40	0.21	0.31	0.53	1.14	0.09	1.30
$n=7$	1.65	0.20	1.94	1.88	0.50	2.25	7.00	0.09	7.15
$n=9$	40.07	0.21	40.49	45.04	1.20	47.11	35.14	0.20	37.18
$n=11$	376.55	0.23	376.99	470.79	1.62	472.44	150.23	0.24	152.12
$n=12$	1013.9	0.35	1015.5	2015.2	2.92	2019.0	294.10	0.37	298.15

Table 3.3: Results of Experiment 1 (differential systems).

From these results, we can notice an evident advantage of our implementation compared to **Mratsolde**, but **RationalSolution** appears to be significantly the best[3] (just for differential systems, see the next experiments).

Remark 3.4. *Going deeper through the analysis of each value of t_1, we have noticed that in all cases, almost all the time is spent while computing a degree bound (which is zero) of $p_n(x)$. This is expected as $p_n(x)$ is very "complicated". Note that to compute this bound, we use simple forms,* **Mratsolde** *uses super-reduction [41] and* **RationalSolution** *uses EG-eliminations [3].*

Experiment 2: difference systems

In this experiment we chose $\mathrm{den}(Y) = x\,(x+1)\,(4\,x^5 + 2\,x^4 - 5\,x^3 - 9\,x^2 + 1)$. We have tested input systems of the form $\mathbf{y}(x+1) = A(x)\,\mathbf{y}(x)$ with matrix dimensions $n = 5, 10, 15, 20$. We compare our implementation with the procedures **deltaRS** from Isolde and **RationalSolution** from the LinearFunctionalSystems package. The table below shows the results, in terms of the time (in seconds), between the three different implementations.

	RatSols_1PLS			**deltaRS**			**RationalSolution**		
	t_1	t_2	T	t_1	t_2	T	t_1	t_2	T
$n=5$	0.13	0.38	0.68	0.20	0.52	0.73	0.62	3.28	4.46
$n=10$	2.06	3.86	6.00	2.71	6.28	9.01	9.99	81.41	92.32
$n=15$	20.34	19.35	39.90	31.91	31.97	63.93	59.59	$*$	$*$
$n=20$	122.18	74.22	196.88	187.13	113.63	300.84	252.33	$*$	$*$

Table 3.4: Results of Experiment 2 (difference systems).

[3]Be careful, we have encountered several examples where the procedure `LinearFunctionalSystems[RationalSolution]` fails to deliver the correct solution for a differential system

Looking at the results of the above table, we can see that in our implementation, the values of t_1, t_2 and T are better than those of **deltaRS**, and much better than those of **RationalSolution** which behaves badly (the symbol $*$ indicates that a computation did not terminate after 4 hours).

Experiment 3: q-difference systems

Here we chose $\mathrm{den}(Y) = x\, f(x)\, f(q\, x)$ where $f(x) = 4\,x^5 + 2\,x^4 - 5\,x^3 - 9\,x^2 + 1$. The test input systems are of the form $\mathbf{y}(q\,x) = A(x)\,\mathbf{y}(x)$ with matrix dimensions $n = 3,\ 5,\ 7,\ 10$. The results of the experiment are presented in the next table. The symbol $*$ indicates that a computation did not terminate after 4 hours.

	RatSols_1PLS			**RationalSolution**		
	t_1	t_2	T	t_1	t_2	T
$n = 3$	0.36	0.10	0.50	1.536	0.08	147.55
$n = 5$	0.90	0.20	1.58	63.20	0.20	346.28
$n = 7$	3.14	0.53	4.20	928.62	0.58	$*$
$n = 10$	59.10	2.582	63.16	$*$	$*$	$*$

Table 3.5: Results of Experiment 3 (q-difference systems).

The obtained results explain themselves. **RationalSolution** behaves surprisingly in a bad and weird manner. We don't believe that this is just due to the different methods used to compute a universal denominator or polynomial solutions. We believe that there is a defect in the internal procedures of the LinearFunctionalSystems package (for q-difference systems). This defect is also illustrated by the fact that, for all input systems, there is a huge difference between $t_1 + t_2$ and T.

Chapter 4

On Rational and Hypergeometric Solutions of Partial Pseudo-Linear Systems

This chapter constitutes the subjects of the second halves of the published papers [30, 33] in collaboration with M. A. Barkatou and T. Cluzeau.

4.1 Introduction

Let C be a field of characteristic zero and $K = C(x_1, \dots, x_m)$ be the field of rational functions in m independent variables $x_1, \dots, x_m$ with coefficients in C. For $i = 1, \dots, m$, let ϕ_i be a C-automorphism of K, and δ_i be a pseudo-derivation with respect to ϕ_i such that for all $j \neq i$, x_j is a constant with respect to ϕ_i and δ_i, i.e., $\phi_i(x_j) = x_j$ and $\delta_i(x_j) = 0$. In the present chapter, the object of study is a *partial pseudo-linear system* defined over the $\phi\delta$-field $(K, \{\phi_i, \delta_i\}_{1 \leq i \leq m})$ (see Section 1.2), having the form:

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\, \phi_1(\mathbf{y}) = 0, \\ \quad \vdots \\ L_m(\mathbf{y}) := \delta_m(\mathbf{y}) - M_m\, \phi_m(\mathbf{y}) = 0, \end{cases} \tag{4.1}$$

where $\mathbf{y}$ is a vector of n unknown functions of $x_1, \dots, x_m$ and the matrix $M_i \in \mathbb{M}_n(K)$ for all $i = 1, \dots, m$. One underlying motivation for studying partial pseudo-linear systems is that many special and transcendental functions are solutions of such systems. For instance, one can check that the vector $\mathbf{y}(x,k) = (H(x,k), H(x,k+1))^T$, where $H(x,k)$ are the Hermite polynomials

$$H(x,k) = k! \sum_{j=0}^{[k/2]} \frac{(-1)^j (2\,x)^{k-2j}}{j!(k-2\,j)!}, \tag{4.2}$$

satisfies the system

$$\mathbf{y}(k+1,x) = \begin{pmatrix} 0 & 1 \\ -2\,k & 2\,x \end{pmatrix} \mathbf{y}(k,x), \qquad \frac{\partial \mathbf{y}}{\partial x}(k,x) = \begin{pmatrix} 2\,x & -1 \\ 2\,k & 0 \end{pmatrix} \mathbf{y}(k,x). \tag{4.3}$$

We assume that System (4.1) is integrable, this means it satisfies the *integrability conditions*:

$$[L_i, L_j] := L_i \circ L_j - L_j \circ L_i = 0, \quad \forall i,j = 1, \dots, m, \tag{4.4}$$

where $L_i := I_n\, \delta_i - M_i\, \phi_i$ denotes the matrix pseudo-linear operator associated to the ith system of (4.1). Following the terminology of [46, Definition 2] and [75, 95], we further suppose that (4.1) is *fully integrable*, i.e., for all $i = 1, \dots, m$ with $\phi_i \neq \mathrm{id}_K$ and $\delta_i = \gamma_i\,(\mathrm{id}_K - \phi_i)$ where $\gamma_i \in K^*$, the matrix $M_i + \gamma_i\, I_n$ is invertible (see Section 1.2). The integrability conditions assure that the space of rational solutions of System (4.1) is of finite dimension over C (at most n). This implies, in particular, that there exists a (not necessarily unique) polynomial (called *universal denominator*) $U \in C[x_1, \dots, x_m]$ such that for any rational solution $\mathbf{y}$ of (4.1), $U\mathbf{y}$ is a vector of polynomials. Note that, the existence of a universal denominator is not always guaranteed if one considers other kinds of linear partial differential (or difference) systems or equations. For instance, it was shown in [80, 81] that there is no algorithm for testing the existence of a universal denominator for rational solutions of linear partial differential or difference equations with rational function coefficients (i.e., even an algorithm that only answers YES or NO to the question of existence, does not exist). One can also consult [70, 71] where it was shown that for some scalar linear partial difference equations (such as $(\mathbf{y}(x_1+1, x_2) - \mathbf{y}(x_1, x_2+1) = 0)$, there is no universal denominator for all rational solutions.

In this chapter, we are interested in computing rational solutions, and more generally hypergeometric solutions, of fully integrable systems of the form (4.1). In both cases, we shall use a recursive approach that has been already adapted in different contexts. In particular, this recursive approach is adapted in [29] for computing rational solutions of integrable connections (i.e., the case of System (4.1) with m differential systems), and in [74] for computing hypergeometric solutions[1] of more general partial systems over Laurent-Ore algebras. Contrary to [74], this chapter contains a specific algorithm for computing rational solutions which is useful in itself for computing *eigenrings* (see Section 4.2.2.5).

The first main contribution of the present chapter consists in a new efficient algorithm for computing all rational solutions of a partial pseudo-linear system (4.1). The method proceeds by recursion and, in particular, it requires, for $i = 1, \dots, m$, an algorithm for computing all rational solutions of a sole pseudo-linear system of the form $\delta_i(\mathbf{y}) - N\,\phi_i(\mathbf{y}) = 0$, where $N \in \mathbb{M}_s(C(p_1, \dots, p_r)(x_i))$, $1 \leq s \leq n$ and $p_1, \dots, p_r$ are parameters which are constants with respect to ϕ_i and δ_i. This can be done using Algorithm **RatSol_1PLS** developed in Chapter 3. In order to speed up the computation of rational solutions of System (4.1), our implementation [32] takes into account two aspects. First, some necessary conditions for an irreducible polynomial to appear in the denominator of a rational solution are obtained by inspecting the irreducible factors of the denominators of all the matrices M_i (see Section 4.2.2.3). Moreover, in the recursive process, as the m pseudo-linear systems in (4.1) can be considered in an arbitrary order, we tried to see, through experiments, if there are some orders better than others from the computational point of view. The timings obtained from most of our experiments indicate that the best strategy seems to be to consider first the non-differential systems (i.e., $\phi_i \neq \mathrm{id}_K$) and then the differential systems (see Section 4.2.2.4).

The second main contribution of this chapter is a new efficient algorithm for computing hypergeometric solutions of System (4.1). The method also proceeds by recursion and uses the same strategy as in [29, Sections 5] for integrable connections. Our approach relies on an algorithm for computing hypergeometric solutions of a general first

[1] In [29, 74], hypergeometric solutions are called hyperexponential solutions

order pseudo-linear system $\delta_i(\mathbf{y}) - N\,\phi_i(\mathbf{y}) = 0$, where $N \in \mathbb{M}_s(C(p_1,\ldots,p_r)(x_i))$, $1 \le s \le n$ and $p_1,\ldots,p_r$ are parameters which are constants with respect to ϕ_i and δ_i. To our knowledge, such an algorithm exists only for differential [84], difference [44], and q-difference [13] systems. Note that the algorithm in [13] reduces the search of solutions of the given system to the search of solutions of several scalar equations, while the algorithms in [44, 84] are direct, i.e., they do not reduce to the scalar case. Our current implementation for computing hypergeometric solutions of partial pseudo-linear systems only deals with a system (4.1) composed of differential and/or difference systems.

The rest of the chapter is organised as follows. In Section 4.2, we present our first contribution, that is a new recursive algorithm for computing rational solutions of partial pseudo-linear systems. For the sake of clarity, we first give in Section 4.2.1 the basic ideas of our algorithm for a System (4.1) composed of one pure differential system and one pure difference system. In Section 4.2.2, we extend these ideas to handle System (4.1) with arbitrary order m and develop the general algorithm. We also provide some explanations concerning our implementation. This includes necessary conditions for an irreducible polynomial to appear in the denominator of a rational solution and also includes timings comparing different strategies. Some applications of the algorithm are also given. Section 4.3 is devoted to the computation of hypergeometric solutions of partial pseudo-linear systems. We first review the algorithms developed for pure differential and difference systems and then develop our algorithm. Examples of computations are also given to clarify our approaches, and a demonstration of the different implementations is provided in Chapter 5.

4.2 Rational solutions

In this section, we present a new algorithm for computing rational solutions of a partial pseudo-linear system (4.1) which is fully integrable and satisfies the integrability conditions (4.4). We extend the ideas developed in [29] for integrable connections (i.e., the case where all the systems are differential systems) to handle a more general system (4.1). For $i, j = 1,\ldots,m$, the variable x_j's ($j \neq i$) are constants with respect to ϕ_i and δ_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 considered as constant parameters.

Definition 4.1. *Let $K = C(x_1,\ldots,x_m)$. A* rational solution *of a partial pseudo-linear system (4.1) is a vector $\mathbf{y} \in K^n$ that satisfies $L_i(\mathbf{y}) = 0$, for all $i = 1,\ldots,m$.*

Example 4.1. *Let $K = C(x_1,x_2)$ and consider the partial pseudo-linear system*

$$\begin{cases} \delta_1(\mathbf{y}) = (x_2/x_1)\,\phi_1(\mathbf{y}), \\ \delta_2(\mathbf{y}) = (x_1-1)\,\phi_2(\mathbf{y}), \end{cases}$$

where

$$\phi_1 = \mathrm{id}_K, \quad \delta_1 = \frac{\partial}{\partial x_1}, \quad \phi_2 : (x_1, x_2) \mapsto (x_1, x_2 - 1), \quad \delta_2 = \mathrm{id}_K - \phi_2.$$

One can check that the function $y(x_1, x_2) = x_1^{x_2}$ *is a solution of the system but it is not a rational solution in the sense of Definition 4.1. Also, the vector*

$$\mathbf{y}(x, k) = (H(x, k), H(x, k+1))^T,$$

where the $H(x, k)$*'s are the Hermite polynomials* (4.2)*, is not a rational solution of* (4.3) *in the sense of Definition 4.1.*

The objective of this section is to develop an algorithm for computing all rational solutions of (4.1). For the sake of clarity, we explain first the details of the algorithm in the case $m = 2$ with one difference system and one differential system. We then extend these ideas for a general system (4.1) with arbitrary order m, composed of any type of pseudo-linear systems.

4.2.1 The case of one difference and one differential system

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

$$\{\mathbf{y}(k+1, x) = A(k, x)\,\mathbf{y}(k, x), \quad \mathbf{y}'(k, x) = B(k, x)\,\mathbf{y}(k, x)\}, \tag{4.5}$$

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

$$A'(k, x) = B(k+1, x)\,A(k, x) - A(k, x)\,B(k, x). \tag{4.6}$$

4.2.1.1 Description of the approach

Let us describe our method for computing rational solutions of System (4.5). 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 $\mathbf{w}_1, \ldots, \mathbf{w}_s \in K^n$ $(0 \le s \le n)$ of its rational solutions (see Chapter 3 or [7]). If we do not find any nonzero rational solution, then we are done as (4.5) does not admit any nonzero rational solution. Let $W \in \mathbb{M}_{n \times s}(K)$ be a matrix having for columns the $\mathbf{w}_i$'s.

Lemma 4.1. *With the above notations, the matrix* $W' - B\,W \in \mathbb{M}_{n \times s}(K)$ *is a solution of* $\mathbf{y}(k+1, x) = A(k, x)\,\mathbf{y}(k, x)$*.*

Proof. Let $Z = W' - B\,W$. Using the fact that $W(k+1, x) = A(k, x)\,W(k, x)$, we get

$$Z(k+1, x) = A'(k, x)\,W(k, x) + A(k, x)\,W'(k, x) - B(k+1, x)\,A(k, x)\,W(k, x).$$

It then follows from integrability condition (4.6) that

$$Z(k+1,x) = A(k,x)\,(W'(k,x) - B(k,x)\,W(k,x)) = A(k,x)\,Z(k,x).$$

□

Let us complete $\mathbf{w}_1, \ldots, \mathbf{w}_s$ into a basis $\mathbf{w}_1, \ldots, \mathbf{w}_n$ of K^n and define

$$P = (W \quad V) \in \mathrm{GL}_n(K),$$

where $V \in \mathbb{M}_{n\times(n-s)}(K)$ has $\mathbf{w}_{s+1}, \ldots, \mathbf{w}_n$ as columns. Performing the change of variables $\mathbf{y} = P\,\mathbf{z}$ in System (4.5), the differential system becomes

$$\mathbf{z}' = P^{-1}\,(B\,P - P')\,\mathbf{z},$$

and has the following properties:

Lemma 4.2. *With the above notations, let us write*

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

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

Proof. The equation $P\,N = B\,P - P'$ yields

$$W\,N^{11} + V\,N^{21} = -(W' - B\,W).$$

From Lemma 4.1, the matrix $W' - B\,W$ 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 $W' - B\,W = W\,C$. We then obtain

$$W\,(N^{11} + C) + V\,N^{21} = 0,$$

which ends the proof as the columns of $P = (W \quad V)$ form a basis of K^n. □

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

Theorem 4.1. *Let $W \in \mathbb{M}_{n\times s}(K)$ 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 $N^{11} \in \mathbb{M}_s(C(x))$ be the unique solution of the matrix linear system $W\,N^{11} = -(W' - B\,W)$. If $\mathbf{z}_1, \ldots, \mathbf{z}_r \in C(x)^s$ is a basis of* *rational solutions of* $\mathbf{z}'(x) = N^{11}(x)\,\mathbf{z}(x)$*, then* $W\,\mathbf{z}_1, \ldots, W\,\mathbf{z}_r \in K^n$ *is a basis of rational solutions of* (4.5)*. Moreover, every rational solution of* (4.5) *can be obtained in such a way.*

Proof. Let $\mathbf{z} \in C(x)^s$ be a rational solution of $\mathbf{z}'(x) = N^{11}(x)\,\mathbf{z}(x)$ and let us consider $\mathbf{y}(k,x) = W(k,x)\,\mathbf{z}(x)$. We have

$$\mathbf{y}(k+1,x) = W(k+1,x)\,\mathbf{z}(x) = A(k,x)\,W(k,x)\,\mathbf{z}(x) = A(k,x)\,\mathbf{y}(k,x).$$

Moreover $\mathbf{y}' = W'\,\mathbf{z} + W\,\mathbf{z}' = W'\,\mathbf{z} + W\,N^{11}\,\mathbf{z} = B\,\mathbf{y}$, by definition of N^{11}. This ends the first part of the proof. Now let $\mathbf{y}$ be a solution of (4.5). 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

$$\mathbf{y} = W\,z = (W \quad V)\,(\mathbf{z}^T \quad 0^T)^T.$$

Thus, $\mathbf{y}$ is a solution of $\mathbf{y}'(k,x) = B(k,x)\,\mathbf{y}(k,x)$ if and only if $(\mathbf{z}^T \quad 0^T)^T$ is a solution of $\mathbf{y}' = N\,\mathbf{y}$ where $N = P^{-1}\,(B\,P - P')$. This is equivalent to $\mathbf{z}'(x) = N^{11}(x)\,\mathbf{z}(x)$ and yields the desired result. □

4.2.1.2 Algorithm and example

Theorem 4.1 naturally provides an algorithm for computing a basis of rational solutions of System (4.5). It proceeds as follows:

Algorithm **RationalSolutions_DifferenceDifferential**

Input: A system of the form (4.5).
Output: A matrix whose columns form a basis of rational solutions of (4.5).

1. Compute a basis of rational solutions of $\mathbf{y}(k+1,x) = A(k,x)\,\mathbf{y}(k,x)$.
2. Let $W \in \mathbb{M}_{n\times s}(K)$ be a matrix whose columns form a basis of the rational solutions of $\mathbf{y}(k+1,x) = A(k,x)\,\mathbf{y}(k,x)$.
3. Compute the unique solution $N^{11} \in \mathbb{M}_s(C(x))$ of the matrix linear system $W\,N^{11} = -(W' - B\,W)$.
4. Compute a basis $\mathbf{z}_1, \ldots, \mathbf{z}_r$ of the rational solutions of the differential system $\mathbf{z}'(x) = N^{11}(x)\,\mathbf{z}(x)$,
5. **Return** $W\,\mathbf{z}_1, \ldots, W\,\mathbf{z}_r$.

In Steps 1 and 3 if we do not find any nonzero rational solution, then we return 0_n (the zero vector of dimension n) since this implies that (4.5) does not admit nonzero rational solutions.

Example 4.2. *Consider the system ([4.5](#)) with matrices $A(x,k)$ and $B(x,k)$ given by:*

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

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

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

$$W = \begin{bmatrix} 2 & \frac{1}{k} \\ 2 & \frac{1}{k} \\ x+k & 0 \\ -2 & -\frac{1}{k} \end{bmatrix}.$$

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

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

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

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

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

$$Y = W\,Z = \begin{bmatrix} \frac{x}{k} & 2 \\ \frac{x}{k} & 2 \\ 0 & x+k \\ -\frac{x}{k} & -2 \end{bmatrix}.$$

4.2.2 The general case: partial pseudo-linear systems

Let $K = C(x_1, \ldots, x_m)$. The process explained in Section 4.2.1 can be generalised to an arbitrary number of pseudo-linear systems of any type, in other words, to a general system (4.1). Our method proceeds by recursion and relies on an algorithm for computing rational solutions of a first order pseudo-linear system $\delta(\mathbf{y}) = M\,\phi(\mathbf{y})$. Such an algorithm has been described in Chapter 3 (in particular Section 3.4) both for differential systems ($\phi = \mathrm{id}_{C(x)}$) and for ϕ-systems where $\phi(f(x)) = f(q\,x+r)$ for all $f \in C(x)$, with $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. Consequently, for all $i = 1, \ldots, m$ such that $\phi_i \neq \mathrm{id}_K$, we assume that ϕ_i satisfies the above conditions.

4.2.2.1 A recursive approach

Let us now give the details of our recursive approach. We first consider the pseudo-linear system $L_1(\mathbf{y}) = 0$ (see also Section 4.2.2.4) over $C(x_2, \ldots, x_m)(x_1)$. We compute a basis $\mathbf{w}_1, \ldots, \mathbf{w}_s \in K^n$ $(0 \leq s \leq n)$ of rational solutions of $L_1(\mathbf{y}) = 0$ (see Section 3.4). If we do not find any nonzero rational solution, then we stop as (4.1) does not admit any nonzero rational solution. Otherwise, denote by $W \in \mathbb{M}_{n\times s}(K)$ the matrix whose columns are the $\mathbf{w}_i$'s. We complete $\mathbf{w}_1, \ldots, \mathbf{w}_s$ into a basis $\mathbf{w}_1, \ldots, \mathbf{w}_n$ of K^n and we define the matrix $P = (W \quad V) \in \mathrm{GL}_n(K)$, where $V \in \mathbb{M}_{n\times(n-s)}(K)$ has $\mathbf{w}_{s+1}, \ldots, \mathbf{w}_n$ as columns. Performing the change of dependent variables $\mathbf{y} = P\,\mathbf{z}$ in System (4.1), we obtain the equivalent system

$$\left\{ \begin{array}{l} \widetilde{L_1}(\mathbf{z}) := \delta_1(\mathbf{z}) - N_1\,\phi_1(\mathbf{z}) = 0, \\ \quad\vdots \\ \widetilde{L_m}(\mathbf{z}) := \delta_m(\mathbf{z}) - N_m\,\phi_m(\mathbf{z}) = 0, \end{array} \right. \tag{4.7}$$

where $N_i = P^{-1}\,[\,M_i\,\phi_i(P) - \delta_i(P)\,]$ for all $i = 1, \ldots, m$. We have the following result as an analogue of Lemma 4.2.

Lemma 4.3. *With the above notations, let us decompose the matrices N_i's of System (4.7) by blocks as*

$$N_i = \left[\begin{array}{c|c} N_i^{11} & N_i^{12} \\ \hline N_i^{21} & N_i^{22} \end{array}\right],$$

where $N_i^{11} \in \mathbb{M}_s(K)$. *Then, for all* $i = 1, \ldots, m$, *the matrix* $N_i^{11} \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ *does not depend on* x_1. *Moreover it can be computed as the unique solution of the matrix linear system* $W\, N_i^{11} = -L_i(W)$, *and in particular* $N_1^{11} = 0$. *Finally,* $N_i^{21} = 0$ *for all* $i = 1, \ldots, m$.

Proof. The equation $P\, N_i = M_i\, \phi_i(P) - \delta_i(P)$ yields

$$W\, N_i^{11} + V\, N_i^{21} = -L_i(W).$$

From the integrability conditions (4.4), we get that, for all $i = 1, \ldots, m$, $L_i(W)$ is a rational solution of the system $L_1(\mathbf{y}) = 0$ so that there exists a unique constant matrix $C \in \mathbb{M}_s(C(x_2, \ldots, x_m))$, i.e., not depending on x_1, such that $L_i(W) = W\, C$. We then obtain, for all $i = 1, \ldots, m$,

$$W\, (N_i^{11} + C) + V\, N_i^{21} = 0,$$

which ends the proof as the columns of $P = (W \quad V)$ form a basis of K^n. □

From Lemma 4.3, we deduce the following result justifying the correctness of our iterative algorithm for computing rational solutions of System (4.1).

Theorem 4.2. *Given a partial pseudo-linear system* (4.1). *Let* $W \in \mathbb{M}_{n\times s}(K)$ *be a matrix whose columns form a basis of the rational solutions of* $L_1(\mathbf{y}) = 0$. *For* $i = 2, \ldots, m$, *let* $N_i^{11} \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ *be the unique solution of the matrix linear system* $W\, N_i^{11} = -L_i(W)$. *Suppose that* $Z \in \mathbb{M}_{s\times r}(C(x_2, \ldots, x_m))$ *is a matrix whose columns form a basis of the rational solutions of the partial pseudo-linear system of size* s *over* $C(x_2, \ldots, x_m)$

$$\begin{cases} \delta_2(\mathbf{y}) - N_2^{11}\, \phi_2(\mathbf{y}) = 0, \\ \quad \vdots \\ \delta_m(\mathbf{y}) - N_m^{11}\, \phi_m(\mathbf{y}) = 0, \end{cases} \tag{4.8}$$

then the columns of the matrix $W Z \in \mathbb{M}_{n\times r}(K)$ *form a basis of all rational solutions of* (4.1). *Moreover, every rational solution of* (4.1) *can be obtained in such a way.*

Proof. Let $Z \in \mathbb{M}_{s\times r}(C(x_2, \ldots, x_m))$ be a matrix whose columns form a basis of all rational solutions of (4.8) and let us consider $Y = W Z$. We have

$$L_1(Y) = \delta_1(W)\, \phi_1(Z) + W\, \delta_1(Z) - M_1\, \phi_1(W)\, \phi_1(Z) = \delta_1(W)\, Z - M_1\, \phi_1(W)\, Z = 0,$$

since W is a solution of $L_1(\mathbf{y}) = 0$. Now for $i = 2, \ldots, m$, by definition of N_i^{11}, we have:

$$\begin{aligned} L_i(Y) &= \delta_i(W)\, \phi_i(Z) + W\, \delta_i(Z) - M_i\, \phi_i(W)\, \phi_i(Z) \\ &= \Big[\delta_i(W) + W\, N_i^{11} - M_i\, \phi_i(W)\Big]\, \phi_i(Z) = 0. \end{aligned}$$

This ends the first part of the proof. Now let Y be a solution of (4.1). In particular, Y is a rational solution of $L_1(\mathbf{y}) = 0$ so that there exists $Z \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ such that $Y = WZ = (W \quad V)\,(Z^T \quad 0^T)^T$. Thus, for $i = 2, \ldots, m$, Y is a solution of $L_i(\mathbf{y}) = 0$ if and only if $(Z^T \quad 0^T)^T$ is a solution of the system (4.7). This is equivalent to Z being a solution to system (4.8) and yields the desired result. □

4.2.2.2 Algorithm and examples

Theorem 4.2 shows that rational solutions of (4.1) can be computed recursively. Indeed, we have reduced the problem of computing rational solutions of System (4.1) of size n in m variables to that of computing rational solutions of System (4.8) of size $s \leq n$ in $m-1$ variables. This gives rise to the following iterative algorithm for computing a basis of rational solutions of System (4.1).

Algorithm **RationalSolutions_PPLS**

Input: An integrable system of the form (4.1).
Output: A matrix whose columns form a basis of rational solutions of (4.1) or 0_n (the zero vector of dimension n) if no non-trivial rational solution exists.

1. Compute a basis of rational solutions of $L_1(\mathbf{y}) = 0$ (see Section 3.4).
2. If there are no non-trivial rational solutions of $L_1(\mathbf{y}) = 0$, then **Return** 0_n and **Stop**.
3. Let $W \in \mathbb{M}_{n\times s}(K)$ be a matrix whose columns form a basis of the rational solutions of $L_1(\mathbf{y}) = 0$.
4. If $m = 1$, then **Return** W and **Stop**.
5. For $i = 2, \ldots, m$, compute the unique solution $N_i^{11} \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ of the matrix linear system $W\,N_i^{11} = -L_i(W)$.
6. **Return** W multiplied by the result of applying the current algorithm to System (4.8).

Let us illustrate our algorithm on the following examples:

Example 4.3. *We consider a partial pseudo-linear system composed of one pure difference system, one pure q-difference system and one pure differential system, defined as follows:*

$$\begin{cases} \mathbf{y}(x_1+1, x_2, x_3) = A_1(x_1, x_2, x_3)\,\mathbf{y}(x_1, x_2, x_3), \\ \mathbf{y}(x_1, qx_2, x_3) = A_2(x_1, x_2, x_3)\,\mathbf{y}(x_1, x_2, x_3), \\ \frac{\partial}{\partial x_3}\mathbf{y}(x_1, x_2, x_3) = A_3(x_1, x_2, x_3)\,\mathbf{y}(x_1, x_2, x_3), \end{cases} \tag{4.9}$$

where $q \in \mathbb{Q}^*$ is not a root of unity. Let $K = \mathbb{Q}(q)(x_1, x_2, x_3)$. The matrices $A_1, A_2 \in \mathrm{GL}_2(K)$ and $A_3 \in \mathbb{M}_2(K)$ are given by:

$$A_1 = \begin{bmatrix} \dfrac{x_1+1}{x_1} & \dfrac{-q\,x_3\,(x_3+x_1)}{x_2^2\,x_1} \\ 0 & \dfrac{x_3+x_1}{x_3+x_1+1} \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & \dfrac{-x_3\,(x_3+x_1)\,(q-1)}{x_2^2} \\ 0 & q \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & \dfrac{q\,(x_3+x_1)}{x_2^2} \\ 0 & \dfrac{-1}{x_1+x_3} \end{bmatrix}.$$

Rewriting the three systems as pseudo-linear systems (see Section 1.2 for more details), System (4.9) can be transformed into the form (4.1) with

$$\left\{ \begin{array}{ll} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}), & M_1 = \phi_1(A_1) - I_2, \\ L_2(\mathbf{y}) := \delta_2(\mathbf{y}) - M_2\,\phi_2(\mathbf{y}), & M_2 = \phi_2(A_2) - I_2, \\ L_3(\mathbf{y}) := \delta_3(\mathbf{y}) - M_3\,\phi_3(\mathbf{y}), & M_3 = A_3, \end{array} \right. \tag{4.10}$$

where the ϕ_i's are the automorphisms defined by:

$$\phi_1 : (x_1, x_2, x_3) \mapsto (x_1 - 1, x_2, x_3), \quad \phi_2 : (x_1, x_2, x_3) \mapsto (x_1, x_2/q, x_3), \quad \phi_3 = \mathrm{id}_K,$$

and the δ_i's are the ϕ_i-derivations defined by:

$$\delta_1 = \mathrm{id}_K - \phi_1, \quad \delta_2 = \mathrm{id}_K - \phi_2, \quad \delta_3 = \partial/\partial x_3.$$

Let us describe our iterative process for computing rational solutions of System (4.10). Computing rational solutions of the system $L_1(\mathbf{y}) = 0$, we get two linearly independent rational solutions given by the columns of

$$W_1 = \begin{bmatrix} \dfrac{x_1 - x_3}{x_2^4} & \dfrac{1}{x_2^4} \\ \dfrac{-1}{q\,x_2^2\,(x_3+x_1)} & \dfrac{1}{q\,x_2^2\,x_3\,(x_3+x_1)} \end{bmatrix}.$$

Solving the linear systems $W_1\,N_2^{11} = -L_2(W_1)$ and $W_1\,N_3^{11} = -L_3(W_1)$ we get:

$$N_2^{11} = \begin{bmatrix} -q^4+1 & 0 \\ -q^3\,(q-1)\,x_3 & -q^3+1 \end{bmatrix}, \quad N_3^{11} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix},$$

with N_2^{11} and N_3^{11} both independent from x_1. We are then reduced to solving the partial *pseudo-linear system*

$$\begin{cases} \widetilde{L_2}(\mathbf{y}) := \delta_2(\mathbf{y}) - N_2^{11}\,\phi_2(\mathbf{y}) = 0, \\ \widetilde{L_3}(\mathbf{y}) := \delta_3(\mathbf{y}) - N_3^{11}\,\phi_3(\mathbf{y}) = 0. \end{cases}$$

The rational solutions of the system $\widetilde{L_2}(\mathbf{y}) = 0$ *are given by the columns of the matrix*

$$W_2 = \begin{bmatrix} x_2^4 & 0 \\ x_2^4\,x_3 & x_2^3 \end{bmatrix}.$$

Now, solving the linear system $W_2\,\widehat{N_3}^{11} = -\widetilde{L_3}(W_2)$*, we get*

$$\widehat{N_3}^{11} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix},$$

which is independent from x_1 *and* x_2*. We are next reduced to computing rational solutions of the system* $\delta_3(\mathbf{y}) - \widehat{N_3}^{11}\,\phi_3(\mathbf{y}) = 0$*. We find that they are given by the columns of the matrix*

$$W_3 = \begin{bmatrix} 0 & 1 \\ x_3 & 0 \end{bmatrix}.$$

Finally, a basis of rational solutions of (4.9) *is spanned by the columns of*

$$W_1\,W_2\,W_3 = \begin{bmatrix} \dfrac{x_3}{x_2} & x_1 \\ \dfrac{x_2}{(x_3 + x_1)\,q} & 0 \end{bmatrix}.$$

Example 4.4. *Let* $K = \mathbb{Q}(x_1, x_2, x_3)$ *and consider the following partial pseudo-linear system*

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}), \\ L_2(\mathbf{y}) := \delta_2(\mathbf{y}) - M_2\,\phi_2(\mathbf{y}), \\ L_3(\mathbf{y}) := \delta_3(\mathbf{y}) - M_3\,\phi_3(\mathbf{y}), \end{cases} \tag{4.11}$$

where the ϕ_i*'s are the automorphisms over* K *defined by:*

$$\phi_1 : (x_1, x_2, x_3) \mapsto (x_1 - 5, x_2, x_3), \quad \phi_2 : (x_1, x_2, x_3) \mapsto (x_1, -3\,x_2 - 5, x_3),$$

$$\phi_3 : (x_1, x_2, x_3) \mapsto (x_1, x_2, -3\,x_3),$$

and the δ_i*'s are the* ϕ_i*-derivations defined by:*

$$\delta_1 = \mathrm{id}_K - \phi_1, \quad \delta_2 = \mathrm{id}_K - \phi_2, \quad \delta_3 = \mathrm{id}_K - \phi_3.$$

The matrices $M_i \in \mathbb{M}_2(K)$ *are given by*

$$M_1 = \begin{bmatrix} \frac{5\,x_3+5\,x_1-25}{{x_1}^2+(x_3-10)\,x_1-6\,x_3+25} & \frac{-5\,x_3\,(x_3+x_1-5)}{({x_1}^2+(x_3-10)\,x_1-6\,x_3+25)\,x_2} \\ \frac{-5\,x_2}{({x_1}^2+(x_3-10)\,x_1-6\,x_3+25)\,(x_3+x_1)} & \frac{5\,x_3}{({x_1}^2+(x_3-10)\,x_1-6\,x_3+25)\,(x_3+x_1)} \end{bmatrix},$$

$$M_2 = \begin{bmatrix} \frac{x_3\,(4\,x_2+5)}{((x_1-1)\,x_3+{x_1}^2)\,x_2} & \frac{x_3\,(4\,x_2+5)\,x_1\,(x_3+x_1)}{x_2\,(3\,x_2+5)\,({x_1}^2+x_1x_3-x_3)} \\ \frac{4\,x_2+5}{(x_1-1)\,x_3+{x_1}^2} & \frac{x_3\,(4\,x_2+5)}{(3\,x_2+5)\,({x_1}^2+x_1x_3-x_3)} \end{bmatrix},$$

$$M_3 = \begin{bmatrix} -\frac{4\,x_3}{{x_1}^2-3\,x_1x_3+3\,x_3} & \frac{4\,x_3x_1\,(-3\,x_3+x_1)}{((-3\,x_1+3)\,x_3+{x_1}^2)\,x_2} \\ -\frac{4\,x_2x_3}{({x_1}^2-3\,x_1x_3+3\,x_3)\,(x_3+x_1)} & -\frac{12\,{x_3}^2}{({x_1}^2-3\,x_1x_3+3\,x_3)\,(x_3+x_1)} \end{bmatrix}.$$

Computing rational solutions of the system $L_1(\mathbf{y}) = 0$*, we obtain two linearly independent rational solutions given by the columns of*

$$W_1 = \begin{bmatrix} x_1 - x_3 & 1 \\ -\frac{x_2\,(x_1+x_3-1)}{x_3+x_1} & \frac{x_2}{x_3} \end{bmatrix}.$$

The unique solutions of the linear systems $W_1\,N_2^{11} = -L_2(W_1)$ *and* $W_1\,N_3^{11} = -L_3(W_1)$ *are given by:*

$$N_2^{11} = \begin{bmatrix} 0 & 0 \\ \frac{x_3\,(4\,x_2+5)}{x_2} & -\frac{4\,x_2+5}{x_2} \end{bmatrix}, \quad N_3^{11} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{4}{3} \end{bmatrix},$$

with N_2^{11} *and* N_3^{11} *both independent from* x_1*. We are then reduced to solving the partial pseudo-linear system*

$$\begin{cases} \widetilde{L_2}(\mathbf{y}) := \delta_2(\mathbf{y}) - N_2^{11}\,\phi_2(\mathbf{y}) = 0, \\ \widetilde{L_3}(\mathbf{y}) := \delta_3(\mathbf{y}) - N_3^{11}\,\phi_3(\mathbf{y}) = 0. \end{cases}$$

The rational solutions of the system $\widetilde{L_2}(\mathbf{y}) = 0$ *are given by the columns of the matrix*

$$W_2 = \begin{bmatrix} 1 & 0 \\ x_3 & \frac{1}{x_2} \end{bmatrix}.$$

The unique solution of the matrix linear system $W_2\,\widehat{N_3}^{11} = -\widetilde{L_3}(W_2)$ *is*

$$\widehat{N_3}^{11} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{4}{3} \end{bmatrix},$$

which is independent from the variables x_1 *and* x_2*. The rational solutions of the system* $\delta_3(\mathbf{y}) - \widehat{N_3}^{11}\,\phi_3(\mathbf{y}) = 0$ *are given by the columns of*

$$W_3 = \begin{bmatrix} 0 & 1 \\ x_3 & 0 \end{bmatrix}.$$

Finally, a basis of rational solutions of System (4.11) *is spanned by the columns of*

$$W_1\,W_2\,W_3 = \begin{bmatrix} \frac{x_3}{x_2} & x_1 \\ 1 & \frac{x_2}{x_3+x_1} \end{bmatrix}.$$

Example 4.5. *Let* $K = \mathbb{Q}(x_1, x_2, x_3)$ *and consider the partial pseudo-linear system*

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}), \\ L_2(\mathbf{y}) := \delta_2(\mathbf{y}) - M_2\,\phi_2(\mathbf{y}), \\ L_3(\mathbf{y}) := \delta_3(\mathbf{y}) - M_3\,\phi_3(\mathbf{y}), \end{cases}$$

where the ϕ_i*'s are the automorphisms over* K *defined by:*

$$\phi_1 : (x_1, x_2, x_3) \mapsto (x_1 - 1, x_2, x_3), \quad \phi_2 : (x_1, x_2, x_3) \mapsto (x_1, x_2 + 1, x_3), \quad \phi_3 = \mathrm{id}_K,$$

and the δ_i*'s are the* ϕ_i*-derivations defined by:*

$$\delta_1 = \mathrm{id}_K - \phi_1, \quad \delta_2 = \mathrm{id}_K - \phi_2, \quad \delta_3 = \partial/\partial x_3.$$

The matrices M_1*,* M_2 *and* M_3 *are matrices in* $\mathbb{M}_3(K)$ *given by:*

$$M_1 = \begin{bmatrix} \frac{(x_3-1)\,x_1 - 1 + (x_3-1)\,x_2}{2\,x_1 + 2\,x_2} & 0 & \frac{(x_3-1)\,x_1 + 1 + (x_3-1)\,x_2}{2\,x_1 + 2\,x_2} \\ 0 & x_3 - 1 & 0 \\ \frac{(x_3-1)\,x_1 + 1 + (x_3-1)\,x_2}{2\,x_1 + 2\,x_2} & 0 & \frac{(x_3-1)\,x_1 - 1 + (x_3-1)\,x_2}{2\,x_1 + 2\,x_2} \end{bmatrix},$$

$$M_2 =$$

$$\begin{bmatrix} \frac{(-x_3+1)x_2{}^2-(x_3-1)(x_3+x_1+1)x_2+(-x_1+1)x_3{}^2-x_3x_1+x_1}{2\,(x_1+x_2)(x_3+x_2+1)x_3} & 0 & \frac{(-x_3+1)x_2{}^2-(x_3-1)(x_3+x_1+1)x_2+(-x_1-1)x_3{}^2+x_3x_1+x_1}{2\,(x_1+x_2)(x_3+x_2+1)x_3} \\ 0 & \frac{-x_3+1}{x_3} & 0 \\ \frac{(-x_3+1)x_2{}^2-(x_3-1)(x_3+x_1+1)x_2+(-x_1-1)x_3{}^2+x_3x_1+x_1}{2\,(x_1+x_2)(x_3+x_2+1)x_3} & 0 & \frac{(-x_3+1)x_2{}^2-(x_3-1)(x_3+x_1+1)x_2+(-x_1+1)x_3{}^2-x_3x_1+x_1}{2\,(x_1+x_2)(x_3+x_2+1)x_3} \end{bmatrix},$$

$$M_3 = \begin{bmatrix} \frac{x_3\,(x_1+1+x_2)+x_2\,(x_1+x_2)}{2\,(x_3+x_2)\,x_3} & 0 & \frac{(x_1+x_2-1)\,x_3+x_2\,(x_1+x_2)}{2\,(x_3+x_2)\,x_3} \\ 0 & \frac{x_1+x_2}{x_3} & 0 \\ \frac{(x_1+x_2-1)\,x_3+x_2\,(x_1+x_2)}{2\,(x_3+x_2)\,x_3} & 0 & \frac{x_3\,(x_1+1+x_2)+x_2\,(x_1+x_2)}{2\,(x_3+x_2)\,x_3} \end{bmatrix}.$$

The system $L_1(\mathbf{y}) = 0$ admits only one ($s = 1$) rational solution given by

$$W_1 = \begin{bmatrix} -\frac{1}{x_1+x_2} \\ 0 \\ \frac{1}{x_1+x_2} \end{bmatrix},$$

so we expect to be reduced to solving a partial system of size $s = 1$, i.e., a system of partial scalar equations. Indeed, the unique solutions of the linear systems $W_1\,N_2^{11} = -L_2(W_1)$ and $W_1\,N_3^{11} = -L_3(W_1)$ are given by:

$$N_2^{11} = -\frac{1}{x_2+x_3+1}, \quad N_3^{11} = \frac{1}{x_2+x_3},$$

and then we consider the partial system

$$\begin{cases} \delta_2(\mathbf{y}) + \dfrac{1}{x_2+x_3+1}\,\phi_2(\mathbf{y}) = 0, \\[2ex] \delta_3(\mathbf{y}) - \dfrac{1}{x_2+x_3}\,\phi_3(\mathbf{y}) = 0. \end{cases}$$

One can check that the latter partial system has a solution $\mathbf{y}(x_2, x_3) = x_2 + x_3$. Finally the original system admits only one rational solution given by:

$$W_1\,\mathbf{y} = \begin{bmatrix} \frac{-x_2-x_3}{x_1+x_2} \\ 0 \\ \frac{x_3+x_2}{x_1+x_2} \end{bmatrix}.$$

4.2.2.3 Necessary conditions for denominators

A rational solution of the partial pseudo-linear system (4.1) is, in particular, a rational solution of each pseudo-linear system $\delta_i(\mathbf{y}) = M_i\,\phi(\mathbf{y})$, $i = 1, \ldots, m$. This necessarily imposes some necessary conditions on the irreducible factors of the denominator of a rational solution of System (4.1) (see [29, Proposition 8] in the integrable connection case). In some cases, taking into account these necessary conditions can significantly speed up the timings of Algorithm **RationalSolutions_PPLS** as it allows to not consider some irreducible factors when computing universal denominators.

For a pure differential system ($\phi_i = \mathrm{id}_K$ and $\delta_i = \partial/\partial x_i$), an irreducible factor of the denominator of a rational solution must divide the denominator of the matrix M_i (see e.g. [20]). For the case of a ϕ-system we have the following consequence of Proposition 3.2. This result can be found in [21] for the pure difference case and can be adapted directly for any ϕ-system considered here.

Proposition 4.1 ([21], Proposition 2)**.** *Consider a ϕ-system*

$$\phi(\mathbf{y}) = N\,\mathbf{y}, \tag{4.12}$$

where $N \in \mathbb{M}_n(C(x))$ and $\phi : x \mapsto q\,x + r$ with $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. With the notations of Section 3.3, assume that $E_\phi(a,b) \neq \emptyset$ and let $h = \max(E_\phi(a,b))$. Let $p \neq x - \frac{r}{1-q} \in C[x]$ be an irreducible polynomial. If p divides the denominator of a non-zero rational solution of System (4.12), then there exist $1 \leq i \leq h+1$ and $0 \leq j \leq h$ such that $i + j \in E_\phi(a,b)$ and p divides both $\phi^{-i}(\mathrm{den}(N))$ and $\phi^{j}(\mathrm{den}(N^{-1}))$.

For the sake of clarity, before giving a result in the general case, we first consider the case of a partial pseudo-linear system with only $m = 2$ pseudo-linear systems being written either as a pure differential system or a ϕ-system. We obtain the following result as a consequence of the discussion above and Proposition 4.1.

Necessary Condition 1. *Let $K = C(x_1, x_2)$ and consider a partial pseudo-linear system*

$$L_1(\mathbf{y}) = 0, \quad L_2(\mathbf{y}) = 0. \tag{4.13}$$

Let A_1 denote the matrix of the system $L_1(\mathbf{y}) = 0$ and $p \in C[x_1, x_2]$ be an irreducible factor of $\mathrm{den}(A_1)$ which involves the variable x_2. Then we have the following result depending on the type of each pseudo-linear system:

1. *If for $i = 1, 2$, $L_i = I_n \frac{\partial}{\partial x_i} - A_i$ then if p appears in the denominator of a rational solution of (4.13), then $p \mid \mathrm{den}(A_2)$ (see [29, Proposition 8]).*
2. *If $L_1 = I_n \frac{\partial}{\partial x_1} - A_1$, $L_2 = I_n\,\phi_2 - A_2$, then if p appears in the denominator of a rational solution of (4.13), there exists $i \in \mathbb{N}^*$ such that $p \mid \phi_2^{-i}(\mathrm{den}(A_2))$.*
3. *If $L_1 = I_n\,\phi_1 - A_1$, $L_2 = I_n \frac{\partial}{\partial x_2} - A_2$, then if p appears in the denominator of a rational solution of (4.13), there exists $i \in \mathbb{N}^*$ such that $p \mid \phi_1^{i}(\mathrm{den}(A_2))$.*
4. *If for $i = 1, 2$, $L_i = I_n\,\phi_i - A_i$, then if p appears in the denominator of a rational solution of (4.13), there exists $i, j \in \mathbb{N}^*$ such that $p \mid \phi_1^{i}(\phi_2^{-j}(\mathrm{den}(A_2)))$.*

Let us illustrate the latter necessary condition on an example.

Example 4.6. *Consider a partial pseudo linear system of the form*

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

where the matrices A and B are given by:

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

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

The factorizations of the denominators of the matrices A and B are given respectively by:

$$\mathrm{den}(A)(x,k) = (x+k)\,x^3\left(k^2-k\,x+x\right)(k-x)\,,$$

$$\mathrm{den}(B)(x,k) = (x+k+1)\,x^2\left(k^2-k\,x+x\right)(k+1-x)\,.$$

The irreducible factor $p(x,k) = k^2 - k\,x + x$ of den(A) *clearly satisfies that, for all $i \in \mathbb{N}^*$, $p \nmid \mathrm{den}(B)(x,k-i)$. Therefore, from Case 2 of Necessary Condition 1, p can not appear in the denominator of any rational solution of the system. However, the latter necessary condition does not allow to draw any conclusion concerning the factors $x+k$ and $k-x$ of* den(A) *(the factor x does not involve the variable k so that it can not be considered in our result). We can indeed check the previous observations as the rational solutions of the original system are given by:*

$$\mathbf{y}_1(x,k) = \begin{bmatrix} \frac{1}{(x+k)} \\ 0 \end{bmatrix}, \quad \mathbf{y}_2(x,k) = \begin{bmatrix} \frac{k}{x^2} \\ k + \frac{x}{k-x} \end{bmatrix}.$$

The gain for our algorithm is that when computing a universal denominator for the differential system $\frac{\partial \mathbf{y}}{\partial x}(x,k) = A(x,k)\,\mathbf{y}(x,k)$, there is no need to compute a simple form (or a super-irreducible form) at $p(x) = k^2 - k\,x + x$ (see Section 3.3).

We now give a generalisation of the latter necessary condition in the case of a partial pseudo-linear system (4.1) composed of m pseudo-linear systems. We distinguish the case when the first system is a differential system (Necessary Condition 2) from that where it is a ϕ-system (Necessary Condition 3). Note that for $\phi_i \neq \mathrm{id}_K$, the systems are written here under the form of a pseudo-linear system $\delta_i(\mathbf{y}) = M_i\,\phi_i(\mathbf{y})$ and not of a ϕ-system $\phi_i(\mathbf{y}) = N_i\,\mathbf{y}$. This is the reason why matrices $\gamma_i^{-1}\,M_i + I_n$ appear in the following results (see Section 1.2).

Necessary Condition 2. *Let $K = C(x_1, \dots, x_m)$. Consider a system of the form (4.1) and suppose that $L_1(\mathbf{y}) = 0$ is a pure differential system, i.e., $\phi_1 = \mathrm{id_K}$ and $\delta_1 = \frac{\partial}{\partial x_1}$. Let $p \in C[x_1, \dots, x_m]$ be an irreducible factor of $\mathrm{den}(M_1)$ such that p involves the variable x_i for some $i \in \{2, \dots, m\}$. Moreover, suppose that one of the following two conditions holds:*

1. $(\phi_i, \delta_i) = \left(\mathrm{id_K}, \frac{\partial}{\partial \mathrm{x_i}}\right)$ *and* $p \nmid \mathrm{den}(M_i)$.
2. $\phi_i \neq \mathrm{id_K}$ *(i.e., $\delta_i = \gamma_i\,(\mathrm{id_K} - \phi_\mathrm{i})$ for some $\gamma_i \in K^*$) and*

$$\forall j \in \mathbb{N}^*, \quad \phi_i^j(p) \nmid \mathrm{den}((\gamma_i^{-1}\,M_i + I_n)^{-1}).$$

Then p cannot appear in the denominator of any rational solution of (4.1).

Necessary Condition 3. *Let $K = C(x_1, \dots, x_m)$. Consider a system of the form (4.1) and suppose that $\phi_1 \neq \mathrm{id_K}$ (i.e., $\delta_1 = \gamma_1\,(\mathrm{id_K} - \phi_1)$ for some $\gamma_1 \in K^*$). Let $p \in C[x_1, \dots, x_m]$ be an irreducible factor of $\mathrm{den}((\gamma_1^{-1}M_1 + I_n)^{-1})$ such that p involves the variable x_i for some $i \in \{2, \dots, m\}$. Moreover, suppose that one of the following two conditions holds:*

1. $(\phi_i, \delta_i) = \left(\mathrm{id_K}, \frac{\partial}{\partial \mathrm{x_i}}\right)$ *and, for all $j \in \mathbb{N}^*$, $p \nmid \phi_i^j(\mathrm{den}(M_i))$.*
2. $\phi_i \neq \mathrm{id_K}$ *(i.e., $\delta_i = \gamma_i\,(\mathrm{id_K} - \phi_\mathrm{i})$ for some $\gamma_i \in K^*$) and*

$$\forall j,\, k \in \mathbb{N}^*, \quad p \nmid \phi_1^j\left(\phi_i^{-k}\left(\mathrm{den}(\left(\gamma_i^{-1}\,M_i + I_n\right)^{-1})\right)\right).$$

Then p cannot appear in the denominator of any rational solution of (4.1).

4.2.2.4 Implementation and comparison of different strategies

Algorithm **RationalSolutions_PPLS** has been implemented in Maple in our package PseudoLinearSystems [32]. It includes part of the necessary conditions given in Section 4.2.2.3. Note that we have also implemented a similar version of Algorithm **RationalSolutions_PPLS** which takes directly as input a partial pseudo-linear system composed of pure differential, difference or q-difference systems (such as System (4.9)), i.e., there is no need to transform every system into a pseudo-linear system as we did in Example 4.3.

In the recursive process of Algorithm **RationalSolutions_PPLS**, the pseudo-linear systems in (4.1) can be considered in an arbitrary order. We have thus tried to see (through examples) if there are some orders better than others from the computational point of view. Let us give some timings of one of our experiments in the case of $m = 2$ pseudo-linear systems where one system is a pure differential system (with independent variable x and usual derivation $\frac{\partial}{\partial x}$) and the other is a pure difference system (with independent variable k, $\phi : (x, k) \mapsto (x, k-1)$ and $\delta = \mathrm{id}_K - \phi$). In this experiment the matrices of the systems are generated from a randomly chosen fundamental matrix of rational solutions but whose denominator denoted by U is fixed as a product of some of the following three polynomials:

$$U_1(x,k) = (x+k)\,(x-k)^2\left(-k^2+x\right)\left(-k^3+x^2+3\right),$$

$$U_2(x,k) =$$
$$-77\,k^8x^6 + 51\,k^2x^{12} - 31\,k^5x^8 + 10\,k^4x^9 - 68\,x^{13} - 91\,x^{12} + 81\,k^{10} - 40\,k^4x^6 + 47\,k^2x^5 + 49\,kx,$$

$$U_3(x,k) =$$
$$k\,(6\,k^{10}x + 5\,kx^9 + 6\,k^2x^7 + 3\,k^7 + 2\,k^6x - 4\,x^7 + 4\,k^4x^2 + k^4x - 3\,x^4 - 5\,k).$$

We compare two strategies:

1. Strategy 1: we start with the differential system.
2. Strategy 2: we start with the difference system.

The following table gives the timings (in seconds) obtained for computing the fundamental matrix of rational solutions with each strategy, for different dimensions n of the systems, and for different fixed denominators U of the rational solutions.

	$U = U_1$			$U = U_1U_2$			$U = U_1U_2U_3$		
	$n = 3$	$n = 6$	$n = 9$	$n = 3$	$n = 6$	$n = 9$	$n = 3$	$n = 6$	$n = 9$
Strategy 1	0.48	2.29	9.01	22.92	187.55	574.83	249.92	912.90	1703.7
Strategy 2	0.39	2.83	16.46	0.35	2.16	12.22	0.94	3.39	15.17

The table seems to indicate that Strategy 2, i.e., starting with the difference system, gives, in general, better timings. In particular, the difference between the distinct timings seems to be particularly significant when the denominator includes *large* irreducible factors as U_2 and U_3. In the case $U = U_1$, we do not have large singularities in the denominator and Strategy 1 behaves well. Going deeper into the analysis of these timings for each step of the algorithm, we can see that, in Strategy 1, most of the time is spent in computing simple forms which can be quite involved for singularities as the ones given by U_2 and U_3. In Strategy 2, we have no simple form computations to get a universal denominator of the first system (as it is a difference system, see Section 3.3.1) and then, the large factors U_2 and U_3 disappear as the differential system to be considered next only involves the variable x. For instance, in Example 4.3, if we start with the differential system with matrix $M_3 = A_3$, we must compute a simple form at the singularity given by the irreducible factor $x_1 + x_3$ of $\mathrm{den}(A_3)$. But if we treat first the difference and the q-difference systems as it is done in Example 4.3, we can see that at the end of the process, the differential system to be considered is $\delta_3(\mathbf{y}) - \widehat{N_3}^{11}\,\phi_3(\mathbf{y}) = 0$, where $\widehat{N_3}^{11}$ has no finite singularities, and therefore no simple form computations are needed to get a universal denominator of the differential system.

From these observations (and other comparisons that we have performed), we make the choice to treat the ϕ-systems ($\phi_i \neq \mathrm{id}_K$) first and to consider the differential systems at the end of the iterative process, where the systems involve fewer independent variables and may also be of smaller size.

4.2.2.5 Applications: eigenring and decomposition of systems

Computing rational solutions of partial pseudo-linear systems is useful for computing hypergeometric solutions as we will see in Section 4.3. It can also be useful for computing the so called *eigenring* of System (4.1).

Let $K = C(x)$. For a differential system $\mathbf{y}'(x) = M(x)\,\mathbf{y}(x)$ over K, the eigenring is defined as the set of matrices $P \in \mathbb{M}_n(K)$ such that $P' = M\,P - P\,M$ (see [43, 89]), while for a ϕ-system $\phi(\mathbf{y}) = M\,\mathbf{y}$ over K, the eigenring is defined as the set of matrices $P \in \mathbb{M}_n(K)$ such that $M\,P = \phi(P)\,M$ (see [21]). These notions can be generalised for pseudo-linear systems as follows. Let (K, ϕ, δ) be $\phi\delta$-field and denote by $\mathcal{C}_K$ its field of constants (see Definition 1.3). Consider a fully integrable first order pseudo-linear system over (K, ϕ, δ) of the form

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \quad M \in \mathbb{M}_n(K). \tag{4.14}$$

Definition 4.2. *([22]). The eigenring of System (4.14) is defined as the set*

$$\mathcal{E}(M) = \{\, P \in \mathbb{M}_n(K) \quad ; \quad \delta(P) = M\,\phi(P) - P\,M \,\}\,.$$

Let $P \in \mathbb{M}_n(K)$ be an element of $\mathcal{E}(M)$. The equation $\delta(P) = M\,\phi(P) - P\,M$ can be viewed as a first order pseudo-linear system of size n^2 over (K, ϕ, δ). This can be realised through the linear map

$$\begin{aligned} \mathrm{Vec}: \quad \mathbb{M}_n(K) &\longrightarrow K^{n^2} \\ A = \begin{pmatrix} A(1,.) \\ \vdots \\ A(n,.) \end{pmatrix} &\longrightarrow \begin{pmatrix} A(1,.)^T \\ \vdots \\ A(n,.)^T \end{pmatrix}, \end{aligned}$$

where $A(i,.)$ is the i^{th} row of A. Using the fact that $\mathrm{Vec}(A\,B\,C) = \left(A \otimes C^T\right) \mathrm{Vec}(B)$ where $\otimes$ is the Kronecker product of the matrices, we get that if $\phi = \mathrm{id}_K$ then $\mathbf{y} = \mathrm{Vec}(P)$ is a solution of

$$\delta(\mathbf{y}) = \left(M \otimes I_n - I_n \otimes M^T\right)\, \phi(\mathbf{y}).$$

Otherwise if $\phi \neq \mathrm{id}_K$, i.e. $\delta = \gamma(\mathrm{id}_K - \phi)$ for some $\gamma \in K^*$, then $\mathbf{y} = \mathrm{Vec}(P)$ is a solution of

$$\delta(\mathbf{y}) = \left((M + \gamma\, I_n) \otimes (\gamma^{-1} M^T + I_n)^{-1} - \gamma\, I_{n^2}\right)\, \phi(\mathbf{y}).$$

The eigenring $\mathcal{E}(M)$ is thus a $\mathcal{C}_K$-vector space of dimension at most n^2. Moreover, $\mathcal{E}(M)$ contains the identity matrix I_n and the product of two elements in $\mathcal{E}(M)$ belongs to $\mathcal{E}(M)$. Consequently, $\mathcal{E}(M)$ is a $\mathcal{C}_K$-algebra.

Remark 4.1. *For a ϕ-system $\phi(\mathbf{y}) = M\,\mathbf{y}$ with $M \in \mathrm{GL}_n(K)$, its eigenring can be viewed as the set of matrices $P \in \mathbb{M}_n(K)$ such that $\mathrm{Vec}(P)$ is a solutions of the system $\phi(\mathbf{y}) = \left(M \otimes (M^T)^{-1}\right) \mathbf{y}$, see [21].*

Definition 4.3. *([22]). A pseudo-linear system (4.14) is said to be* decomposable *if it is equivalent to a system $\delta(\mathbf{y}) = N\,\phi(\mathbf{y})$ such that N is a block diagonal matrix of the form*

$$N = \begin{pmatrix} N^{1,1} & 0 & \cdots & 0 \\ 0 & N^{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & N^{k,k} \end{pmatrix}, \tag{4.15}$$

where for all $i = 1, \ldots, k$, $N^{i,i}$ is a square matrix of size $n_i < n$.

The computation of the eigenring is useful for decomposing systems. Indeed, we have the following result:

Theorem 4.3. *([22, Theorem 3.2]). Let $\mathcal{E}(M)$ be the eigenring of System (4.14) such that $\mathcal{E}(M)$ contains an element P having $k \geq 2$ distinct eigenvalues $\lambda_1, \ldots, \lambda_k \in \overline{\mathcal{C}_K}$. Let $T \in \mathrm{GL}_n(K)$ such that the matrix J defined by $J = T^{-1}\,P\,T$ is in Jordan form. Consider the system $\delta(\mathbf{y}) = N\,\phi(\mathbf{y})$ where $N = T^{-1}(M\,\phi(T) - \delta(T))$. Then N has the form (4.15).*

The interested reader can consult [21, 22, 37, 43, 55, 87, 89, 95] for more details on eigenrings of linear functional systems and their applications.

Example 4.7. *Let us go back to the partial pseudo-linear system* (4.11) *defined over* $K = \mathbb{Q}(x_1, x_2, x_3)$*. Its eigenring is defined as the set of matrices* $P \in \mathbb{M}_2(K)$ *verifying*

$$\forall i = 1, 2, 3, \quad \delta_i(P) = M_i\, \phi_i(P) - P\, M_i.$$

Applying Algorithm **RationalSolutions_PPLS** *to the partial system of size 4*

$$\begin{cases} \delta_1(\mathbf{y}) = \big((M_1 + I_2) \otimes (M_1^T + I_2)^{-1} - I_4\big)\ \phi_1(\mathbf{y}), \\ \delta_2(\mathbf{y}) = \big((M_2 + I_2) \otimes (M_2^T + I_2)^{-1} - I_4\big)\ \phi_2(\mathbf{y}), \\ \delta_3(\mathbf{y}) = \big((M_3 + I_2) \otimes (M_3^T + I_2)^{-1} - I_4\big)\ \phi_3(\mathbf{y}), \end{cases}$$

yields a basis of rational solutions given by the columns of

$$\begin{bmatrix} -\dfrac{(x_3+x_1)\,x_3}{x_2\,({x_1}^2+x_1x_3-x_3)} & -\dfrac{x_2x_1}{{x_1}^2+x_1x_3-x_3} & \dfrac{2\,x_3}{{x_1}^2+x_1x_3-x_3} & 1 \\ \dfrac{(x_3+x_1)\,{x_3}^2}{{x_2}^2\,({x_1}^2+x_1x_3-x_3)} & \dfrac{(x_3+x_1)\,{x_1}^2}{{x_1}^2+x_1x_3-x_3} & -\dfrac{(2\,x_3+2\,x_1)\,x_3x_1}{((x_1-1)\,x_3+{x_1}^2)\,x_2} & 0 \\ \dfrac{-x_3-x_1}{(x_1-1)\,x_3+{x_1}^2} & -\dfrac{{x_2}^2}{((x_1-1)\,x_3+{x_1}^2)\,(x_3+x_1)} & \dfrac{2\,x_2}{(x_1-1)\,x_3+{x_1}^2} & 0 \\ \dfrac{(x_3+x_1)\,x_3}{x_2\,({x_1}^2+x_1x_3-x_3)} & \dfrac{x_2x_1}{{x_1}^2+x_1x_3-x_3} & -\dfrac{(2\,x_3+2\,x_1)\,x_1}{{x_1}^2+x_1x_3-x_3} & 1 \end{bmatrix}.$$

Hence, a basis of the eigenring of (4.11) *is given by*

$$\left\{ \begin{bmatrix} -\dfrac{(x_3+x_1)\,x_3}{x_2\,({x_1}^2+x_1x_3-x_3)} & \dfrac{(x_3+x_1)\,{x_3}^2}{{x_2}^2\,({x_1}^2+x_1x_3-x_3)} \\ \dfrac{-x_3-x_1}{(x_1-1)\,x_3+{x_1}^2} & \dfrac{(x_3+x_1)\,x_3}{x_2\,({x_1}^2+x_1x_3-x_3)} \end{bmatrix}, \right.$$

$$\begin{bmatrix} -\dfrac{x_2x_1}{{x_1}^2+x_1x_3-x_3} & \dfrac{(x_3+x_1)\,{x_1}^2}{{x_1}^2+x_1x_3-x_3} \\ -\dfrac{{x_2}^2}{((x_1-1)\,x_3+{x_1}^2)\,(x_3+x_1)} & \dfrac{x_2x_1}{{x_1}^2+x_1x_3-x_3} \end{bmatrix},$$

$$\begin{bmatrix} \dfrac{2\,x_3}{{x_1}^2+x_1x_3-x_3} & -\dfrac{2\,x_3\,(x_3+x_1)\,x_1}{((x_1-1)\,x_3+{x_1}^2)\,x_2} \\ \dfrac{2\,x_2}{(x_1-1)\,x_3+{x_1}^2} & -\dfrac{2\ (x_3+x_1)\,x_1}{{x_1}^2+x_1x_3-x_3} \end{bmatrix},$$

$$\left. \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}.$$

Let $P \in \mathbb{M}_2(K)$ be the third element of the latter basis. Then P admits two distinct eigenvalues 0 and -2. Consider the matrix

$$T = \begin{bmatrix} \frac{(x_3+x_1)\,x_1}{x_1{}^2+x_1x_3-x_3} & -\frac{x_3}{x_1{}^2+x_1x_3-x_3} \\ \frac{x_2}{x_1{}^2+x_1x_3-x_3} & -\frac{x_2}{x_1{}^2+x_1x_3-x_3} \end{bmatrix}.$$

One has $J = T^{-1}\,P\,T = \operatorname{diag}(0,-2)$ is in Jordan form. Therefore, System (4.11) can be decomposed into the equivalent system

$$\begin{cases} \delta_1(\mathbf{y}) = N_1\,\phi_1(\mathbf{y}), \\ \delta_2(\mathbf{y}) = N_2\,\phi_2(\mathbf{y}), \\ \delta_3(\mathbf{y}) = N_3\,\phi_3(\mathbf{y}), \end{cases}$$

where for all $i = 1,2,3$, $N_i = T^{-1}(M_i\,\phi_i(T) - \delta_i(T))$ is a (block) diagonal matrix:

$$N_1 = \begin{bmatrix} \frac{5\,x_1{}^2+(10\,x_3-25)\,x_1+5\,x_3{}^2-20\,x_3}{(x_3+x_1)\,(x_1{}^2+(x_3-10)\,x_1-6\,x_3+25)} & 0 \\ 0 & \frac{10\,x_1+5\,x_3-25}{x_1{}^2+(x_3-10)\,x_1-6\,x_3+25} \end{bmatrix},$$

$$N_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-4\,x_2-5}{x_2} \end{bmatrix},$$

$$N_3 = \begin{bmatrix} -\frac{4\,x_1x_3}{(x_3+x_1)\,((-3\,x_1+3)\,x_3+x_1{}^2)} & 0 \\ 0 & \frac{4\,(x_1-1)\,x_3}{(-3\,x_1+3)\,x_3+x_1{}^2} \end{bmatrix}.$$

4.3 Hypergeometric solutions

In this section, we present our second contribution of this chapter: a recursive algorithm for computing *hypergeometric solutions* of a partial pseudo-linear system (4.1) which is fully integrable and satisfies the integrability conditions (4.4). The recursive approach that we follow is the one used in [29] for integrable connections and in [74] in the more general context of Laurent-Ore algebras. Here we provide details on how this can be efficiently done for partial pseudo-linear systems.

Definition 4.4. *Let $K = C(x)$ and consider a first order pseudo-linear system*

$$\delta(\mathbf{y}) = M\,\phi(\mathbf{y}), \tag{4.16}$$

defined over the $\phi\delta$-field (K,ϕ,δ). Let H be an extension field of K having the same *field of constants. A non zero element $h \in H$ is said to be a hypergeometric term over (K, ϕ, δ) if $\delta(h)/\phi(h) \in K$. A hypergeometric solution of System (4.16) is a product $h\mathbf{g}$ of a hypergeometric term h over (K, ϕ, δ) by a vector $\mathbf{g} \in K^n$ such that $\delta(h\mathbf{g}) = M\phi(h\mathbf{g})$.*

Remark 4.2. *In [29, 84], a hypergeometric solution of a differential system is referred to as an* exponential *solution, and a hypergeometric term is referred to as an* exponential part *(see also Section 4.3.1.1 below).*
*For a ϕ-system $\phi(\mathbf{y}) = M\mathbf{y}$, the authors in [11, 12] give the following definition for hypergeometric terms: "**A non zero element $h \in K = C(x)$ is called a hypergeometric term over K if $\phi(h)/h \in K$.**" In this case, $\phi(h)/h \in K$ is equivalent to $\delta(h)/\phi(h) \in K$ since $\delta(h)/\phi(h) = \gamma(h/\phi(h) - 1)$ for some $\gamma \in K^*$. Therefore, Definition 4.4 for the general pseudo-linear setting covers the previously reported differential and ϕ-system cases.*

Definition 4.5. *Let $K = C(x_1, \ldots, x_m)$ and consider a partial pseudo-linear system of the form (4.1) defined over the $\phi\delta$-field $(K, \{\phi_i, \delta_i\}_{1 \le i \le m})$. Let H be an extension field of K having the same field of constants. A non zero element $h \in H$ is said to be a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{1 \le i \le m})$ if $\delta_i(h)/\phi_i(h) \in K$ for all $i \in \{1, \ldots, m\}$. A hypergeometric solution of (4.1) is a product $h\mathbf{g}$ of a hypergeometric term h over $(K, \{\phi_i, \delta_i\}_{1 \le i \le m})$ by a vector $\mathbf{g} \in K^n$ such that $\delta_i(h\mathbf{g}) = M_i\, \phi_i(h\mathbf{g})$ for all $i \in \{1, \ldots, m\}$.*

Example 4.8. *Let $K = C(x_1, x_2)$ and let us consider the same system in Example 4.1:*

$$\begin{cases} \delta_1(\mathbf{y}) = (x_2/x_1)\, \phi_1(\mathbf{y}), \\ \delta_2(\mathbf{y}) = (x_1 - 1)\, \phi_2(\mathbf{y}), \end{cases}$$

where

$$\phi_1 = \mathrm{id}_K, \quad \delta_1 = \frac{\partial}{\partial x_1}, \quad \phi_2 : (x_1, x_2) \mapsto (x_1, x_2 - 1), \quad \delta_2 = \mathrm{id}_K - \phi_2.$$

The function $\mathbf{y}(x_1, x_2) = x_1^{x_2}$ is a solution of the system. Moreover, $h = x_1^{x_2}$ is a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{1 \le i \le 2})$ since

$$\delta_1(h)/\phi_1(h) = x_2\, x_1^{-1} \in K, \quad \delta_2(h)/\phi_2(h) = x_1^{-1} - 1 \in K.$$

Therefore, $\mathbf{y}$ is a hypergeometric solution in the sense of Definition 4.5. It can be written as $\mathbf{y} = h\mathbf{g}$ where $\mathbf{g} = 1 \in K$.

Before entering the details of our recursive algorithm for (4.1), let us recall how one proceeds in the case $m = 1$ for differential and difference systems.

4.3.1 The differential and difference cases

Let $K = C(x)$. In this section, we review the principle of the algorithms developed in [44, 84] for computing hypergeometric solutions of differential and difference systems.

4.3.1.1 Pflügel's algorithm for differential systems

Consider a first order differential system

$$\mathbf{y}'(x) = M(x)\mathbf{y}(x), \tag{4.17}$$

where $M \in \mathbb{M}_n(K)$, and let $x_0 \in \overline{C}$ be a singularity of the system. Define $t = x - x_0$ if x_0 is finite and $t = x^{-1}$ otherwise. The matrix M can be written as

$$M = t^{-1-p} \sum_{i=0}^{+\infty} M_i \, t^i,$$

where $p \in \mathbb{N}$ is the Poincaré rank of the system, and the matrices $M_i \in \mathbb{M}_n(C)$ with $M_0 \neq 0$.

Definition 4.6. *With the above notations, a non-ramified local exponential part of System (4.17) at x_0, is a polynomial in t^{-1} of the form*

$$f = \frac{\omega_k}{t^{k+1}} + \frac{\omega_{k-1}}{t^k} + \cdots + \frac{\omega_0}{t}, \tag{4.18}$$

where $0 \leq k \leq p$ and the ω_i's are in C, such that there exists a formal local solution of the system of the form

$$\widehat{\mathbf{y}}(t) = \mathrm{e}^{\int f(t)\, dt}\, \widehat{\mathbf{z}}(t), \tag{4.19}$$

where $\widehat{\mathbf{z}}(t)$ is a vector of formal power series in t.

For a differential system (4.17), we seek a solution of the form

$$\mathbf{y}(x) = \mathrm{e}^{\int u(x)\, dx}\, P(x), \tag{4.20}$$

where $u(x) \in K$ and $P(x) \in C[x]^n$. Such a solution is called an exponential solution in [84] (we shall also call it a hypergeometric solution) and $e^{\int u(x) dx}$ is called the exponential part.

Remark 4.3. *We can refer to $\mathrm{e}^{\int u(x)\, dx}$ as the hypergeometric term over K. Indeed, System (4.17) can be written as $\delta(\mathbf{y}) = M\phi(\mathbf{y})$ where $\phi = \mathrm{id}_K$ and $\delta = \frac{d}{dx}$. If we denote by $h = \mathrm{e}^{\int u(x)\, dx}$ then one has $\delta(h)/\phi(h) = u(x) \in K$.*

The algorithm given in [84] to compute an exponential solution of the form (4.20) can be summarised as follows.

1. Compute the possible values of $u(x)$: this can be done by computing the non-ramified local exponential parts at each singularity, including infinity. We shall explain below how to compute these parts. Each combination composed of one non-ramified exponential part at each singularity yields a possible candidate for $u(x)$.

2. Perform the change of variable $\mathbf{y} = \mathrm{e}^{\int u(x)\,dx}\,\mathbf{z}$ in (4.17).

3. Search for a basis of polynomial solutions of the resulting system

$$\mathbf{z}'(x) = (M(x) - u(x)I_n)\,\mathbf{z}(x).$$

   Each polynomial solution $P(x)$ gives an exponential solution (4.20).

A classical approach to compute the non-ramified local exponential parts at a singularity x_0 is the *Newton algorithm* [16, 42, 90]. This algorithm consists first in reducing the system, via a cyclic vector method [17, 51, 57], to an nth scalar differential equation, and then constructing the *Newton Polygon* [19, 90]. The roots of the *Newton polynomials*[2] give the values of the ω_i's occurring in (4.18). However, one would like to avoid such approach since computing an equivalent scalar equation can be very costly in general: see for instance [52, 76]. An alternative method has been proposed by Pflügel in [86] to compute the non-ramified local exponential parts without reduction to scalar equations. In particular, Pflügel proved that if System (4.17) is in k-simple form, then its associated characteristic polynomial $\Psi_k(\lambda)$ (see Definition 2.8) plays the same role as the Newton polynomial.

The algorithm in [84] then proceeds as follows: let $k = p$. Compute a k-simple form of (4.17) (note that the system is already is simple at p) and let $\Psi_k(\lambda)$ be its associated characteristic polynomial. The possible values of ω_k appearing in (4.18) are then roots of $\Psi_k(\lambda)$. For each root ω_k of $\Psi_k(\lambda)$, perform the change of variable $\mathbf{y} = \mathrm{e}^{\int \frac{\omega_k}{t^{k+1}}\,dt}\,\mathbf{z}$. This leads a new system

$$\mathbf{y}'(x) = N(x)\,\mathbf{y}(x), \quad N = M - \frac{\omega_k}{t^{k+1}}\,I_n. \tag{4.21}$$

One then applies recursion by computing a $(k-1)$-simple form of System (4.21). At the end of the recursive process, one finds several combinations of the form $(\omega_k, \omega_{k-1}, \ldots, \omega_0)$. Each combination yields a non-ramified local exponential part of the form (4.18).

The method proposed in [84] to compute a k-simple form at each stage of the recursion consists in applying first super-reduction [41, 67]. However, a system which is k-simple for a fixed k is not necessarily super-irreducible (see for instance Example 2.5 or [59, Remark 4.3.2]). So if one is interested in computing a k-simple form for just one

[2] Newton polynomials are polynomials associated with the slopes of the Newton polygon

value of k then direct methods have to be preferred. Our Algorithm **SimpleForm** from Chapter 2 (and the algorithm in [34, 59] for differential systems) provides such a direct method. Note that Pflügel's algorithm [84] is implemented in the current version of the ISOLDE package [39]. In our PSEUDOLINEARSYSTEMS package [32] we have updated this implementation in such a way that it uses simple forms instead of super-reduction.

Example 4.9. *Consider System (4.17) with*

$$M = \begin{bmatrix} \dfrac{x^2 - x - 1}{x\,(x-1)^2} & \dfrac{x^2 - 3\,x + 3}{x\,(x-1)^2} \\ \dfrac{-x^2 + 2\,x - 2}{(x-1)^2} & \dfrac{x^3 - 2\,x + 2}{x\,(x-1)^2} \end{bmatrix}.$$

The system has three singularities, 0, 1 and ∞. Let us compute first the non-ramified local exponential parts at 0. In this case, the Poincaré rank of the system is $p = 0$. So It is enough to compute a 0-simple form of the system, but as expected it is already 0-simple with an associated characteristic (indicial) polynomial $\Psi_0(\lambda) = (\lambda+1)\,(\lambda-2)$. The roots of $\Psi_0(\lambda)$ are -1 and 2, therefore the non-ramified local exponential parts at 0 are $\frac{-1}{x}$ and $\frac{2}{x}$. Concerning the singularity 1, the system has a Poincaré rank $p = 1$. The system is already 1-simple with an associated characteristic polynomial $\Psi_1(\lambda) = \lambda^2$. The only root of $\Psi_1(\lambda)$ is 0. Now we have to consider again the system $\mathbf{y}'(x) = M(x)\,\mathbf{y}(x)$ since $M - \frac{0}{(x-1)^2}\,I_2 = M$. Using our Algorithm **SimpleForm**, *we compute an equivalent 0-simple system*

$$\mathbf{y}'(x) = \begin{bmatrix} \dfrac{-2\,x^3 + 5\,x^2 - 5\,x + 1}{(x-1)\,x} & \dfrac{4\,x^3 - 10\,x^2 + 14\,x - 6}{(x-1)\,x} \\ \dfrac{-x^2 + 2\,x - 2}{x-1} & \dfrac{2\,x^3 - 4\,x^2 + 6\,x - 2}{(x-1)\,x} \end{bmatrix} y(x),$$

with an associated characteristic (indicial) polynomial $\Psi_0(\lambda) = \lambda^2 - \lambda$ having 0 and 1 as roots. The non-ramified local exponential parts at 1 are thus 0 and $\frac{1}{x-1}$. Using the same concept for the singularity ∞, we find two non-ramified local exponential parts given by 0 and 1. If the original system has an exponential solution (4.20), then the possible candidates of $u(x)$ are

$$\left\{-\frac{1}{x},\, -\frac{1}{x} + \frac{1}{x-1},\, -\frac{1}{x} + 1,\, -\frac{1}{x} + 1 + \frac{1}{x-1},\, \frac{2}{x},\, \frac{2}{x} + \frac{1}{x-1},\, \frac{2}{x} + 1,\, \frac{2}{x} + 1 + \frac{1}{x-1}\right\}.$$

Note that the number of these candidates can be furthermore reduced using the method in [53]. Computing polynomial solutions of the system $\mathbf{z}'(x) = \left(M + \frac{1}{x}\,I_2\right)\mathbf{z}(x)$ and those of $\mathbf{z}'(x) = \left(M - \left(-\frac{1}{x} + 1\right)I_2\right)\mathbf{z}(x)$ we get respectively:

$$P_1(x) = \begin{bmatrix} x^3 \\ x^3 \end{bmatrix}, \quad P_2(x) = \begin{bmatrix} x-1 \\ x^2 - x \end{bmatrix}.$$

A basis of exponential solutions of the original system is hence given by:

$$\mathbf{y}_1(x) = \begin{bmatrix} x^2 \\ x^2 \end{bmatrix}, \quad \mathbf{y}_2(x) = \begin{bmatrix} \frac{\mathrm{e}^x\,(x-1)}{x} \\ \mathrm{e}^x\,(x-1) \end{bmatrix}.$$

4.3.1.2 Barkatou and Van Hoeij algorithm for difference systems

Let $\phi : x \mapsto x+1$ be the forward shift automorphism over K and consider a pure difference system of the form

$$\phi(\mathbf{y}) = N\,\mathbf{y}, \tag{4.22}$$

where $N \in \mathrm{GL}_n(K)$. We are interested in computing hypergeometric solutions of the form

$$\mathbf{y}(x) = c^x\,\mathrm{Sol}\left(\phi - \frac{a(x)}{b(x)}\right)\,P(x), \tag{4.23}$$

where $c \in C^*$, $P(x) \in C[x]^n$ and $a(x), b(x) \in C[x]^*$. The notation $\mathrm{Sol}\left(\phi - \frac{a(x)}{b(x)}\right)$ stands for the solution of the scalar recurrence equation $\phi(\mathbf{z}(x)) = \frac{a(x)}{b(x)}\,\mathbf{z}(x)$. Here the hypergeometric term over K is $h = c^x\,\mathrm{Sol}\left(\phi - \frac{a(x)}{b(x)}\right)$. Algorithms for computing hypergeometric solutions of scalar difference equations have been proposed in [54, 82]. For a difference system (4.22), an algorithm was developed by Barkatou and Van Hoeij in [44]. It can be summarised as follows:

1. Compute possible candidates for $a(x)$, $b(x)$ and c: this is explained below.

2. For each combination $(a(x), b(x), c)$, perform the change of dependent variables $\mathbf{y} = c^x\,\mathrm{Sol}\left(\phi - \frac{a(x)}{b(x)}\right)\mathbf{z}$ in (4.22).

3. Search for a basis of polynomial solutions of the resulting system

$$\phi(\mathbf{z}) = \left(\frac{b(x)}{a(x)\,c}\,N\right)\mathbf{z}.$$

   Each polynomial solution $P(x)$ gives a hypergeometric solution (4.23).

According to [44], the possible candidates for the polynomial $a(x)$, respectively $b(x)$, are amongst the factors of the denominator of N^{-1}, respectively N. Let us now explain how to compute a possible candidate for c. The idea is to *localize* the system at infinity. Let $K_1 = C((x^{-1}))$: the completion of the field K with respect to the t-adic valuation ν (here

$t = x^{-1}$). Our System (4.22) can be written as the local pseudo-linear system

$$\delta(\mathbf{y}) = M\,\tilde{\phi}(\mathbf{y}), \tag{4.24}$$

where $\tilde{\phi} = \phi^{-1}$, $\delta = \mathrm{id}_{K_1} - \tilde{\phi}$ and $M = \tilde{\phi}(N) - I_n \in \mathbb{M}_n(K_1)$. Here the degree of δ is $\omega = 1$ (see Definition 1.12). The matrix M is uniquely written as

$$M = t^{1-p} \sum_{i=0}^{+\infty} M_i\, t^i,$$

where $p \in \mathbb{N}$ is the Poincaré rank of the system, and the matrices $M_i \in \mathbb{M}_n(C)$ with $M_0 \neq 0$.

Remark 4.4. *Multiplying System (4.24) by t^{p-1} on both sides yields a local pseudo-linear system in the sense of our Definition 1.14. Hence, without loss of generality, we can also call System (4.24) a local pseudo-linear system.*

A hypergeometric solution (4.23) can be viewed as a non-zero local formal solution (at ∞) of the form

$$\mathbf{y}(x) = \Gamma(x)^{k-1}\, c^x\, x^d\, \mathbf{z}(x), \tag{4.25}$$

where $\Gamma(x)$ is the Gamma function, $k \in \mathbb{N}$, $d \in C$, and $\mathbf{z}(x)$ is an n-dimensional vector of formal power series in $x^{-1} = t$. A formal solution (4.25) is called a local hypergeometric solution. Let k be a non-negative integer such that $k \leq p$ and let $\delta_k = t^{k-1}\delta$. System (4.24) can then be written as

$$L(\mathbf{y}) = A^{(k)}\delta_k(\mathbf{y}) + B^{(k)}\,\tilde{\phi}(\mathbf{y}) = 0, \tag{4.26}$$

where

$$A^{(k)} = \mathrm{diag}(t^{\alpha_1}, \ldots, t^{\alpha_n}), \quad \alpha_i = -\max\{0, 1-k-\nu(M(i,.))\},$$

and

$$B^{(k)} = -t^{k-1}A^{(k)}M \in \mathbb{M}_n(C[[t]]).$$

The action of the pseudo-linear operator L on a local solution (4.25) yields

$$L(\mathbf{y}) = \Gamma(x-1)^{k-1}c^{x-1}x^d \left(\left((c-\varepsilon_k)A_0^{(k)} + B_0^{(k)}\right) z_0 + O(x^{-1})\right),$$

where $\varepsilon_k = 1$ if $k = 1$ and $\varepsilon_k = 0$ otherwise. It follows that if a local hypergeometric solution (4.25) exists, then one must have

$$\left((c-\varepsilon_k)A_0^{(k)} + B_0^{(k)}\right) z_0 = 0,$$

which means that $c - \varepsilon_k$ must be a root of the polynomial

$$\Psi_k(\lambda) = \det\left(A_0^{(k)}\lambda + B_0^{(k)}\right).$$

But it may happen that $\Psi_k(\lambda)$ vanishes identically in λ. The method proposed in [44] to effectively compute the polynomials $\Psi_k(\lambda)$ for $k = 0, \ldots, p$, is to compute (once) a super-irreducible form (see Section 2.3.1) of (4.24) using the algorithm in [15]. The possible values of c can then be obtained from the roots of the polynomials $\Psi_0(\lambda), \ldots, \Psi_p(\lambda)$.

The algorithm of [44] is implemented in the current version of the ISOLDE package [39]. This implementation takes advantage of the results developed in [54, 82] to reduce the number of combinations $(a(x), b(x), c)$ to be checked. In our PSEUDOLINEARSYSTEMS package [32] we have modified this implementation in such a way that it uses the algorithm developed in Section 2.3.4 instead of the one in [15], for computing a super-irreducible form of (4.24). We have noticed that in several examples, our implementation is more often better from a computational point of view. See for instance "HS_DifferenceExample" on the webpage of [32].

Example 4.10. *Consider System* (4.22) *with*

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

One has $\mathrm{den}(N) = x\,(x+1)$ *and* $\mathrm{den}(N^{-1}) = x^2\,(x-1)\,(x^2-x-1)$. *The system can be written in the local form* (4.24) *as*

$$\delta(\mathbf{y}) = \begin{bmatrix} -\frac{1}{x-1} & 0 & -\frac{x-2}{x} & 0 \\ 1 & -1 & \frac{2}{x} & -x+1 \\ -1 & 1 & x-3 & 1 \\ -\frac{x+1}{x-1} & \frac{x}{x-1} & \frac{x^2-3\,x+1}{x\,(x-1)} & \frac{x^2-2\,x+2}{x-1} \end{bmatrix} \phi(\mathbf{y}),$$

with Poincaré rank $p = 2$. *The latter system can be transformed into an equivalent super-irreducible system*

$$\delta(\mathbf{y}) = \begin{bmatrix} -\frac{1}{x-1} & 0 & \frac{-x+2}{x} & 0 \\ -\frac{2}{x-1} & \frac{1}{x-1} & \frac{x^2-x-1}{x\,(x-1)} & 0 \\ -1 & 1 & x-3 & 0 \\ \frac{-x-1}{x-1} & \frac{x}{x-1} & \frac{x^2-3\,x+1}{x\,(x-1)} & x-2 \end{bmatrix} \phi(\mathbf{y}),$$

with associated characteristic polynomials

$$\psi_0(\lambda) = (\lambda+1)^2, \quad \psi_1(\lambda) = \lambda^2, \quad \psi_2(\lambda) = \lambda^2(\lambda-1)^2.$$

Thus, if the original system admits a hypergeometric solution (4.23), then the possible values of c are -1, 0 and 1. For $c = 1$, $a(x) = x(x-1)$ and $b(x) = x+1$, the system $\phi(\mathbf{z}) = \left(\frac{b(x)}{a(x)\,c}\,N\right)\mathbf{z}$ admits only one polynomial solution

$$P(x) = \begin{bmatrix} 0 \\ x^2 - x \\ 0 \\ -x^2 + x \end{bmatrix}.$$

A hypergeometric solution is thus given by

$$\mathbf{y}(x) = c^x \operatorname{Sol}\left(\phi - \frac{a(x)}{b(x)}\right) P(x) = \begin{bmatrix} 0 \\ -\Gamma\,(x) \\ 0 \\ \Gamma\,(x) \end{bmatrix},$$

and it is in fact the only hypergeometric solution of the original system.

4.3.2 Description of the recursive approach

Let $K = C(x_1, \ldots, x_m)$. Let us sketch the general iterative process for computing a basis of hypergeometric solutions of an integrable partial pseudo-linear system of the form (4.1). We extend the ideas developed in [29] for integrable connections (i.e., the case where all the systems are differential systems) to handle a more general system (4.1). The recursive process described below resembles the one proposed in Subsection 4.2.2.1 for computing rational solutions. For $i = 1, \ldots, m$, the variable x_j's ($j \neq i$) are constants with respect to ϕ_i and δ_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 considered as constant parameters.

Lemma 4.4. *Let h be a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{1\leq i\leq m})$ and $\mathbf{g} \in K^n$. If $h\mathbf{g}$ is a hypergeometric solution of System (4.1) then $\mathbf{g}$ is a rational solution of the system*

$$\begin{cases} \widetilde{L_1}(\mathbf{z}) := \delta_1(\mathbf{z}) - h^{-1}\,\phi_1(h)\,[\,M_1 - f_1\,I_n]\,\phi_1(\mathbf{z}) = 0, \\ \quad\vdots \\ \widetilde{L_m}(\mathbf{z}) := \delta_m(\mathbf{z}) - h^{-1}\,\phi_m(h)\,[\,M_m - f_m\,I_n]\,\phi_m(\mathbf{z}) = 0, \end{cases}$$

where $f_i = \delta_i(h)/\phi_i(h)$ for all $i = 1, \ldots, m$.

Proof. For all $i = 1, \ldots, m$, we have $\widetilde{L_i}(\mathbf{g}) = \delta_i(\mathbf{z}) - h^{-1}[\,M_i\,\phi_i(h\mathbf{g}) + \delta(h)\,\phi(\mathbf{g})\,]$. Using the equality $\delta_i(h\mathbf{g}) = \delta_i(h)\,\phi_i(\mathbf{g}) + h\,\delta_i(\mathbf{g})$, we get $\widetilde{L_i}(\mathbf{g}) = h^{-1}[\,\delta_i(h\mathbf{g}) - M_i\,\phi_i(h\mathbf{g})\,] = 0$, since $h\mathbf{g}$ is a solution of (4.1). □

We consider first the pseudo-linear system $L_1(\mathbf{y}) = 0$ defined over $C(x_2, \ldots, x_m)(x_1)$. We compute a basis of hypergeometric solutions of $L_1(\mathbf{y}) = 0$. If we do not find any nonzero hypergeometric solution, then we stop as (4.1) does not admit any nonzero hypergeometric solution. Otherwise, let $h\mathbf{g}$ be a hypergeometric solution of $L_1(\mathbf{y}) = 0$, where h is a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{1\leq i\leq m})$ and $\mathbf{g} \in K^n$. Performing the change of dependent variables $\mathbf{y} = h\,\mathbf{z}$ in System (4.1) yields the system

$$\begin{cases} \widetilde{L_1}(\mathbf{z}) := \delta_1(\mathbf{z}) - h^{-1}\,\phi_1(h)\,[\,M_1 - f_1\,I_n]\,\phi_1(\mathbf{z}) = 0, \\ \quad\vdots \\ \widetilde{L_m}(\mathbf{z}) := \delta_m(\mathbf{z}) - h^{-1}\,\phi_m(h)\,[\,M_m - f_m\,I_n]\,\phi_m(\mathbf{z}) = 0, \end{cases} \tag{4.27}$$

where $f_i = \delta_i(h)/\phi_i(h)$ for all $i = 1, \ldots, m$. From the integrability conditions (4.4) and Lemma 4.4, it follows that System (4.27) is integrable.

We compute now a basis $\mathbf{w}_1, \ldots, \mathbf{w}_s \in K^n$ $(0 \leq s \leq n)$ of rational solutions of $\widetilde{L}_1(\mathbf{z}) = 0$ (see Section 3.4) and denote by $W \in \mathbb{M}_{n\times s}(K)$ the matrix whose columns are the $\mathbf{w}_i$'s. We complete $\mathbf{w}_1, \ldots, \mathbf{w}_s$ into a basis $\mathbf{w}_1, \ldots, \mathbf{w}_n$ of K^n and define $P = (W \quad V) \in \mathrm{GL}_n(K)$, where $V \in \mathbb{M}_{n\times(n-s)}(K)$ has $\mathbf{w}_{s+1}, \ldots, \mathbf{w}_n$ as columns. Performing the change of dependent variables $\mathbf{z} = P\,\mathbf{y}$ in System (4.27) yields the system

$$\begin{cases} \delta_1(\mathbf{y}) - N_1\,\phi_1(\mathbf{y}) = 0, \\ \quad\vdots \\ \delta_m(\mathbf{y}) - N_m\,\phi_m(\mathbf{y}) = 0, \end{cases} \tag{4.28}$$

where for all $i = 1, \ldots, m$,

$$N_i = P^{-1}\left[\,h^{-1}\,\phi_i(h)\,(M_i - f_i I_n)\,\phi_i(P) - \delta_i(P)\,\right]. \tag{4.29}$$

We have the following result as an analogue of Lemma 4.3.

Lemma 4.5. *With the above notations, let us decompose the matrices N_i's of System (4.28) by blocks as*

$$N_i = \left[\begin{array}{c|c} N_i^{11} & N_i^{12} \\ \hline N_i^{21} & N_i^{22} \end{array}\right],$$

where $N_i^{11} \in \mathbb{M}_s(K)$. Then, for all $i = 1, \ldots, m$, the matrix $N_i^{11} \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ does not depend on x_1. Moreover it can be computed as the unique solution of the matrix linear system $W\, N_i^{11} = -\widetilde{L}_i(W)$, and in particular $N_1^{11} = 0$. Finally, $N_i^{21} = 0$ for all $i = 1, \ldots, m$.

Proof. Equation (4.29) yields in particular

$$W\, N_i^{11} + V\, N_i^{21} = -\widetilde{L}_i(W).$$

From the integrability conditions $[\widetilde{L}_i, \widetilde{L}_j] := \widetilde{L}_i \circ \widetilde{L}_j - \widetilde{L}_j \circ \widetilde{L}_i = 0$, for all $1 \leq i, j \leq m$, we get that, for all $i = 1, \ldots, m$, $\widetilde{L}_i(W)$ is a rational solution of the system $\widetilde{L_1}(\mathbf{y}) = 0$ so that there exists a unique constant matrix $C \in \mathbb{M}_s(C(x_2, \ldots, x_m))$, i.e., not depending on x_1, such that $\widetilde{L}_i(W) = W\, C$. We then obtain, for all $i = 1, \ldots, m$,

$$W\,(N_i^{11} + C) + V\, N_i^{21} = 0,$$

which ends the proof as the columns of $P = (W \quad V)$ form a basis of K^n. □

From Lemma 4.5, we deduce the following analog of Theorem 4.2 which proves that all hypergeometric solutions of (4.1) can be computed recursively.

Theorem 4.4. *Given a partial pseudo-linear system (4.1). Let $h\mathbf{g}$, where h is a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{1 \leq i \leq m})$ and $g \in K^n$, be a hypergeometric solution of $L_1(\mathbf{y}) = 0$. For all $i = 1, \ldots, m$, denote by $\widetilde{L}_i$ the pseudo linear operator of the form*

$$\widetilde{L}_i = \delta_i - h^{-1}\, \phi_i(h)\, [\, M_i - f_i I_n]\, \phi_i,$$

where $f_i = \delta_i(h)/\phi_i(h) \in K$. Let $W \in \mathbb{M}_{n \times s}(K)$ be a matrix whose columns form a basis of the rational solutions of $\widetilde{L_1}(\mathbf{y}) = 0$. For $i = 2, \ldots, m$, let $N_i^{11} \in \mathbb{M}_s(C(x_2, \ldots, x_m))$ be the unique solution of the matrix linear system $W\, N_i^{11} = -\widetilde{L}_i(W)$. Suppose that $Z \in \mathbb{M}_{s \times r}(C(x_2, \ldots, x_m))$ is a matrix whose columns form a basis of all hypergeometric solutions of the partial pseudo-lineas system of size s over $C(x_2, \ldots, x_m)$

$$\left\{\begin{array}{l} \delta_2(\mathbf{y}) - N_2^{11}\, \phi_2(\mathbf{y}) = 0, \\ \quad\vdots \\ \delta_m(\mathbf{y}) - N_m^{11}\, \phi_m(\mathbf{y}) = 0, \end{array}\right. \tag{4.30}$$

then the columns of the matrix $hWZ \in \mathbb{M}_{n\times r}(K)$ form a basis of all hypergeometric solutions of (4.1). Moreover, every hypergeometric solution of (4.1) can be obtained in such a way.

Proof. Let $Z \in \mathbb{M}_{s\times r}(C(x_2,\dots,x_m))$ be a matrix whose columns form a basis of all hypergeometric solutions of (4.30) and let us consider $Y = h\,W\,Z$. We have

$$\begin{aligned} L_1(Y) &= \delta_1(h)\,\phi_1(W)\,\phi_1(Z) + h\,\delta_1(W)\,\phi_1(Z) + h\,W\,\delta_1(Z) - M_1\,\phi_1(h)\,\phi_1(W)\,\phi_1(Z) \\ &= \phi_1(h)\left[\delta_1(W) - h^{-1}\,\phi_1(h)\,(M_1 - f_1 I_n)\,\phi_1(W)\right] Z = 0. \end{aligned}$$

Now for $i = 2,\dots,m$, we have

$$\begin{aligned} L_i(Y) &= \delta_i(h)\,\phi_i(W)\,\phi_i(Z) + h\,\delta_i(W)\,\phi_i(Z) + h\,W\,\delta_i(Z) - M_i\,\phi_i(h)\,\phi_i(W)\,\phi_i(Z) \\ &= \phi_i(h)\left[\left[(\,f_i I_n - M_i\,)\,\phi_i(W) + \frac{h}{\phi_i(h)}\,\delta_i(W)\right]\phi_i(Z) + \frac{h}{\phi_i(h)}\,W\,\delta_i(W)\right] \\ &= h\,W\left[-N_i^{11}\,\phi_i(Z) + \delta_i(Z)\right] = 0. \end{aligned}$$

This ends the first part of the proof. Now let $Y = h\mathbf{g}$ be a hypergeometric solution of (4.1), where h is a hypergeometric term over $(K, \{\phi_i,\delta_i\}_{1\le i\le m})$ and $\mathbf{g} \in K^n$. From Lemma 4.4, $\mathbf{g}$ is a rational solution of $\widetilde{L_i}(\mathbf{y}) = 0$ for all $i = 1,\dots,m$. In particular $\mathbf{g}$ is a rational solution of $\widetilde{L_1}(\mathbf{y}) = 0$ so that there exists $Z \in \mathbb{M}_s(C(x_2,\dots,x_m))$ such that $\mathbf{g} = W\,Z$. This implies that $Y = h\,(W \quad V)\,(Z^T \quad 0^T)^T$. Thus, for $i = 2,\dots,m$, Y is a solution of $L_i(\mathbf{y}) = 0$ if and only if $(Z^T \quad 0^T)^T$ is a solution of the system (4.28). This is equivalent to Z being a solution to system (4.30) and yields the desired result. □

4.3.3 Algorithm and example

Theorem 4.4 shows that hypergeometric solutions of (4.1) can be computed recursively. Again we have reduced the problem of computing hypergeometric solutions of System (4.1) of size n in m variables to that of computing hypergeometric solutions of System (4.30) of size $s \le n$ in $m-1$ variables. The algorithm proceeds as follows:

Algorithm **HypergeometricSolutions**

Input: An integrable system of the form (4.1).
Output: A matrix whose columns form a basis of hypergeometric solutions of (4.1) or 0_n (the zero vector of dimension n) if no non-trivial hypergeometric solution exists.

1. If there are no non-trivial hypergeometric solutions of $L_1(\mathbf{y}) = 0$, then **Return** 0_n and **Stop**.
2. Compute an $n \times d$ matrix $H = (h_1 \mathbf{g}_1 \dots h_d \mathbf{g}_d)$ whose columns form a basis of all hypergeometric solutions of $L_1(\mathbf{y}) = 0$, where $\mathbf{g}_i \in K^n$ and h_i are hypergeometric terms over $(K, \{\phi_i, \delta_i\}_{1 \leq i \leq m})$.
3. If $m = 1$, then **Return** H and **Stop**.
4. Set $S = \begin{bmatrix} \ \end{bmatrix}$: an empty matrix.
5. **For** every h in $\{h_1, \dots, h_d\}$ **do**
   - Let $\widetilde{L_i} = \delta_i - h^{-1} \phi_i(h) \, (M_i - f_i I_n) \, \phi_i$, where $f_i = \delta_i(h)/\phi_i(h)$.
   - Compute a matrix $W \in \mathbb{M}_{n \times s}(K)$ whose columns form a basis of the rational solutions of $\widetilde{L_1}(\mathbf{y}) = 0$.
   - For $i = 2, \dots, m$, compute the unique solution $N_i^{11} \in \mathbb{M}_s(\overline{C}(x_2, \dots, x_m))$ of the matrix linear system $W \, N_i^{11} = -\widetilde{L_i}(W)$.
   - Compute Z: the result of applying the current algorithm to System (4.30).
   - If $Z = 0_n$ then **Return** 0_n and **Stop**.
   - $S :=$ the matrix having the columns of S and those of the matrix $h \, W \, Z$.

   End For
6. **Return** S.

Example 4.11. *Let $K = C(x, k)$ and consider the partial pseudo-linear system composed of one difference and one differential system:*

$$\begin{cases} \mathbf{y}(x, k+1) = A(x, k) \, \mathbf{y}(x, k) \\ \dfrac{\partial}{\partial x} \mathbf{y}(x, k) = B(x, k) \, \mathbf{y}(x, k) \end{cases} \tag{4.31}$$

where $A \in \mathrm{GL}_n(K)$ and $B \in \mathbb{M}_n(K)$ are given by:

$$A = \begin{bmatrix} \dfrac{(k+1)(-x+k)^2}{k(-x+k+1)^2} & 0 & 0 \\ -\dfrac{(k+1)(k^2x - 2kx^2 + x^3 - k^2 + 4kx - 3x^2 + x)}{k(-x+k+1)^2} & \dfrac{x(k+1)}{k} & 0 \\ \dfrac{(k^2-k-1)x^2}{k} & -\dfrac{(k^2-k-1)x^2}{k} & kx \end{bmatrix},$$

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

Rewriting the two systems as pseudo-linear systems (see Section 1.2 for more details), System (4.31) can be transformed into the form (4.1) with

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}), & M_1 = \phi_1(A) - I_2, \\ L_2(\mathbf{y}) := \delta_2(\mathbf{y}) - M_2\,\phi_2(\mathbf{y}), & M_2 = B, \end{cases} \tag{4.32}$$

where

$$\phi_1 : (x,k) \mapsto (x, k-1), \quad \phi_2 = id_K, \quad \delta_1 = id_K - \phi_1, \quad \delta_2 = \frac{\partial}{\partial x}.$$

Computing hypergeometric solutions of the difference system $L_1(\mathbf{y}) = 0$ using the algorithm of [44] (see also Section 4.3.1.2), we get a basis composed of three hypergeometric solutions given by:

$$\mathrm{Sol}\left(\sigma - \frac{a(k)}{b(k)}\right)\begin{bmatrix} k \\ k \\ 0 \end{bmatrix}, \quad x^k\,\mathrm{Sol}\left(\sigma - \frac{a(k)}{b(k)}\right)\begin{bmatrix} 0 \\ \dfrac{(-x+k)^2 k}{x} \\ (-x+k)^2 k \end{bmatrix},$$

$$x^k\,\mathrm{Sol}\left(\sigma - \frac{k\,a(k)}{b(k)}\right)\begin{bmatrix} 0 \\ 0 \\ (-x+k)^2 x \end{bmatrix},$$

where $\sigma = \phi_1^{-1} : (x,k) \mapsto (x, k+1)$, $a(k) = (-x+k)^2$ and $b(k) = (-x+k+1)^2$. The notation $\mathrm{Sol}\left(\sigma - \frac{a(k)}{b(k)}\right)$ stands for the solution of the scalar recurrence equation $\sigma(\mathbf{z}) = \frac{a(k)}{b(k)}\,\mathbf{z}$. The hypergeometric terms of latter basis are the elements of the set

$$\begin{aligned}\{h_1,h_2,h_3\} &= \left\{1^k\mathrm{Sol}\left(\sigma-\frac{a(k)}{b(k)}\right),\quad x^k\mathrm{Sol}\left(\sigma-\frac{a(k)}{b(k)}\right),\quad x^k\mathrm{Sol}\left(\sigma-\frac{k\,a(k)}{b(k)}\right)\right\}\\ &= \left\{\frac{1}{(-x+k)^2},\quad \frac{x^k}{(-x+k)^2},\quad \frac{x^k\,\Gamma(k)}{(-x+k)^2}\right\}.\end{aligned}$$

Now take $h=h_1$ and perform the change of variable $\mathbf{y}=h\,\mathbf{z}$ in (4.32). This yields the system over $C(x,k)$:

$$\begin{cases}\widetilde{L_1}(\mathbf{z}) := \delta_1(\mathbf{z})-h^{-1}\,\phi_1(h)\,[\,M_1-f_1I_3\,]\,\phi_1(\mathbf{z})=0,\\ \widetilde{L_2}(\mathbf{z}) := \delta_2(\mathbf{z})-h^{-1}\,\phi_2(h)\,[\,M_2-f_2I_3\,]\,\phi_2(\mathbf{z})=0,\end{cases}$$

where $f_i=\delta_i(h)/\phi_i(h)$ for $i=1,2$. The only rational solution of the difference system $\widetilde{L_1}(\mathbf{z})=0$ is

$$W=\begin{bmatrix}k\\k\\0\end{bmatrix}.$$

Now, solving the matrix linear system $W\,N^{11}=-\widetilde{L_2}(W)$ we get $N^{11}=-1\in C(x)$. We are then reduced to computing the hypergeometric (exponential) solution of the scalar differential equation $\delta_2(\mathbf{y})-N^{11}\phi_2(\mathbf{y})=0$. We find that it is given by $Z=\mathrm{e}^{-x}\in C(x)$. Therefore, we get a hypergeometric solution of (4.31) given by

$$S_1=h\,W\,Z=\begin{bmatrix}\dfrac{k\,\mathrm{e}^{-x}}{(-x+k)^2}\\ \dfrac{k\,\mathrm{e}^{-x}}{(-x+k)^2}\\ 0\end{bmatrix}.$$

Finally, repeating the same process for $h=h_2$ and $h=h_3$, we get two other hypergeometric solutions

$$S_2=\begin{bmatrix}0\\ x^k\,k\,\mathrm{e}^{x}\\ x^{k+1}\,k\,\mathrm{e}^{x}\end{bmatrix},\quad S_3=\begin{bmatrix}0\\0\\x^k\,\Gamma(k)\end{bmatrix}.$$

4.3.4 Remarks on the implementation

The recursive algorithm **HypergeometricSolutions** is implemented in our Maple package PseudoLinearSystems [32]. We repeat that our current implementation only deals with a partial pseudo-linear system (4.1) composed of differential and/or difference systems. We use the algorithm form [84], respectively [44], to compute hypergeometric solutions of a differential, respectively difference, system, but with simple forms instead of super-reduction as explained in Sections 4.3.1.1 and 4.3.1.2.

During the calculation of hypergeometric solutions of $L_1(y) = 0$, non-hypergeometric terms (see Definition 4.5) appearing as functions of $x_2, \ldots, x_m$ may be involved. When this is the case, we discard these terms since they can not lead to hypergeometric solutions in the sense of Definition 4.5). We explain more on when and where this case might happen. Denote by $F = C(x_2, \ldots, x_m)$. Suppose that $L_1(\mathbf{y}) = 0$, defined over $F(x_1)$, is a differential system $\left(\phi_1 = \mathrm{id}_K,\, \delta_1 = \frac{\partial}{\partial x_1}\right)$ admitting a hypergeometric (exponential) solution

$$\mathbf{y}(x_1, \ldots, x_m) = e^{\int u\, dx_1}\, P,$$

where $u = u(x_1, \ldots, x_m) \in F(x_1)$ and $P = P(x_1, \ldots, x_m) \in F[x_1]^n$ (see Section 4.3.1.1). The hypergeometric term over (K, ϕ_1, δ_1) is $h = e^{\int u\, dx_1}$. The base field here is F, so u might depend on a variable x_i for some $i = 2, \ldots, m$. When this is the case, the term h might not be hypergeometric over (K, ϕ_i, δ_i). Thus, the solution $\mathbf{y}$ should not be considered in further computations as it will not lead to hypergeometric solutions of (4.1) in the sense of Definition 4.5. To settle this dilemma, we perform a series of verification tests, for each u obtained during the computation of hypergeometric solutions of the differential system $L_1(\mathbf{y}) = 0$, to check whether h is indeed a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{2 \le i \le m})$ or not. If not, we discard u.

Example 4.12. *Let $m = 2$ and suppose that we have obtained $u = x_1^{-1} x_2$. If $L_2(\mathbf{y}) = 0$ is a differential system with $\phi_2 = \mathrm{id}_K$ and $\delta_2 = \frac{\partial}{\partial x_2}$, then one has $\delta_2(h)/\phi_2(h) = \log(x_1) \notin K$. This means that the term h is not hypergeometric over (K, ϕ_2, δ_2) and thus we discard u. Now if $L_2(\mathbf{y}) = 0$ is a difference system with $\phi_2 : (x_1, x_2) \mapsto (x_1, x_2 + 1)$ and $\delta_2 = \mathrm{id}_K - \phi_2$, then one has $\delta_2(h)/\phi_2(h) = x_1^{-1} - 1 \in K$. This means that the term h is hypergeometric over (K, ϕ_2, δ_2) and thus we keep u.*

The same concept is applied if $L_1(\mathbf{y}) = 0$ is a difference system with

$$\phi_1 : (x_1, x_2, \ldots, x_m) \mapsto (x_1 + 1, x_2, \ldots, x_m), \quad \delta_1 = \mathrm{id}_K - \phi_1.$$

Suppose that $L_1(\mathbf{y}) = 0$, defined over $F(x_1)$, admits a hypergeometric solution

$$\mathbf{y}(x_1, \ldots, x_m) = c^{x_1}\, \mathrm{Sol}\left(\phi_1 - \frac{a}{b}\right) P,$$

where $c \in F^*$, $P = P(x_1, \ldots, x_m) \in F[x_1]^n$ and $a, b \in F[x_1]$ (see Section 4.3.1.2). The hypergeometric term over (K, ϕ_1, δ_1) is $h = c^{x_1}\, \mathrm{Sol}\left(\phi_1 - \frac{a}{b}\right)$. The base field here is F, so the values of c, a and b might involve a variable x_i for some $i = 2, \ldots, m$. When this is the case, the term h might not be hypergeometric over (K, ϕ_i, δ_i), so it will not lead to hypergeometric solutions of (4.1) in the sense of Definition 4.5. Therefore, while computing hypergeometric solutions of the difference system $L_1(\mathbf{y}) = 0$ and once we have obtained a triplet (c, a, b), we perform a verification test to check whether h is a hypergeometric term over $(K, \{\phi_i, \delta_i\}_{2 \leq i \leq m})$ or not. If not, we discard (c, a, b).

Chapter 5

PseudoLinearSystems: A Maple Package for Studying Systems of Pseudo-linear Equations

This chapter constitutes the subject of a part of the published paper [31] in collaboration with M. A. Barkatou and T. Cluzeau.

5.1 Introduction

One major contribution of the present thesis arises in the implementation in Maple of all the algorithms developed in the previous chapters. All implementations are gathered and incarnated as internal procedures in our package PseudoLinearSystems [32]. Whilst some existing packages (such as LinearFunctionalSystems and Isolde [39]) are dedicated to the study of the individual differential, difference and q-difference systems, the PseudoLinearSystems package is dedicated to the study of pseudo-linear systems: a very large class of linear functional systems including differential and (q-)difference systems.

Another novelty of our package is that it uses simple forms as a basic tool to compute necessary local data. This is not the case in the existing packages where super-reduction algorithms [15, 41, 67] or EG-eliminations [3] are used. So in particular, the package PseudoLinearSystems contains a generic procedure for computing a simple form of a pseudo-linear system, as well as local data useful for the local analysis: k-simple forms, super-irreducible forms, indicial polynomials, etc. The package also contains generic procedures devoted to the computation of closed form (polynomial, rational, hypergeometric) solutions for first order pseudo-linear systems and for partial pseudo-linear systems.

In this final chapter, we shall demonstrate the use of several important procedures contained in the package. Note that the package is freely available online. A manual for downloading and installing the package, as well as Maple examples covering several types of pseudo-linear systems, are provided on the webpage:

http://www.unilim.fr/pages_perso/ali.el-hajj/PseudoLinearSystems.html

5.2 Simple and super-irreducible forms

Let $K = C((t))$ be the field of Laurent series in a variable t over a constant field $C \subset \overline{\mathbb{Q}}$, and equipped with the t-adic valuation ν. Moreover, let ϕ be C-automorphism over K preserving the valuation, i.e. $\nu(\phi(a)) = \nu(a)$ for all $a \in K$, and δ be a pseudo-derivation with respect to ϕ. The triplet (K, ϕ, δ) is thus a local $\phi\delta$-field (see Definition 1.10).

5.2.1 The SimpleForm procedure

In Chapter 2, we have developed a generic algorithm to compute a simple form of any (local) pseudo-linear system of the form

$$A\,\delta(y) + B\,\phi(y) = 0,$$

where A and B are square matrices in $\mathbb{M}_n(C[[t]])$, with $C[[t]]$ the ring of formal power series in t, such that $\det(A) \neq 0$. Here L denotes the pseudo-linear operator $L = A\,\delta + B\,\phi$. The algorithm is called **SimpleForm** and it proceeds as presented in the scheme of Section 2.2.3. It is implemented in the PSEUDOLINEARSYSTEMS package as a procedure holding the same name. The inputs of the procedure are:

- The matrices A and B with rational function entries in x admitting power series expansions.
- The variable x and the local parameter t (for instance $t = x - x_0$ for the singularity x_0, and $t = 1/x$ for the singularity ∞).
- The automorphism ϕ and the derivation δ provided as Maple procedures.
- λ: a name.

The output is a list containing respectively the matrices $\widehat{A}$ and $\widehat{B}$ of the equivalent simple system $\widehat{L}(y) = \widehat{A}\,\delta(y) + \widehat{B}\,\phi(y) = 0$, the leading matrix pencil $\widehat{L}_\lambda = \widehat{A}_0\,\lambda + \widehat{B}_0$, the determinant of $\widehat{L}_\lambda$ which is thus not zero, and finally the two invertible matrices S and T such that $\widehat{L} = S\,L\,T$. For instance, to treat the pseudo-linear system considered in Example 2.2.2, the user must first define:

```
> PhiAction:= proc(M,x) return subs(x=2*x-1,M) end:
> DeltaAction:= proc(M,x) return M-PhiAction(M,x) end:
```

The user then enters the matrices A and B, specifies the local parameter $t = x - 1$ and runs:

```
> SimpleForm(A,B,x,t,PhiAction,DeltaAction,lambda);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x-1 \end{bmatrix}, \begin{bmatrix} -1 & 1 & -1 & -1 & -x^2+x-1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x-1 & -(x-1)\,x \\ 2 & 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 0 & x^3-2\,x^2+x \end{bmatrix},$$

$$\begin{bmatrix} \lambda-1 & 1 & -1 & -1 & -1 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 0 & \lambda & 0 & 0 \\ 2 & 0 & 1 & \lambda+2 & 3 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, -\lambda^3-\lambda^2,$$

$$\left[\begin{array}{ccccc} \frac{1}{x-1} & 0 & 0 & -\frac{1}{x-1} & \frac{1}{x-1} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x-1} & 0 & -\frac{1}{x-1} \\ 0 & 0 & 0 & \frac{1}{x-1} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right], \left[\begin{array}{ccccc} x-1 & 0 & 0 & x-1 & x-1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

5.2.2 The SuperReduced procedure

The PSEUDOLINEARSYSTEMS package contains procedures based on the **SimpleForm** procedure for computing important local data of pseudo-linear systems. In particular, it contains the procedure **SuperReduced** which computes a super-irreducible form (see Section 2.3.1) of any local pseudo-linear system

$$t^{-\omega}\,\delta(\mathbf{y}) = M\phi(\mathbf{y}), \tag{5.1}$$

where $\omega \in \mathbb{Z}$ is the degree of the ϕ-derivation δ (see Definition 1.12), and $M \in \mathbb{M}_n(K)$ is uniquely written as $M = t^{-p}\sum_{i=0}^{+\infty} M_i\, t^i$, with $M_0 \neq 0$ and $p \in \mathbb{N}$ is the Poincaré rank of the system.

Computing a super-irreducible form can be done, as explained in Section 2.3.4, by iteratively computing a k-simple form (see Definition 2.8) for $k = p-1, \ldots, 0$, using the **SimpleForm** procedure just by altering at each iteration the derivation δ in its input. The **SuperReduced** procedure takes as input:

- The matrix M with rational function entries in x.
- The variable x and the local parameter t.
- The automorphism ϕ and the derivation δ provided as Maple procedures, as well as the degree ω of δ.
- λ: a name.

It returns the matrix $\widehat{M}$ of an equivalent super-irreducible system $t^{-\omega}\,\delta(y) = \widehat{M}\,\phi(y)$, and the list of pairs $[k, \Psi_k(\lambda)]$ for $k = m, \ldots, 0$, where m is the minimal Poincaré rank, and where the $\Psi_k(\lambda)$'s are the characteristic polynomials of each of the k-simple systems. The last output is the matrix $T \in \mathrm{GL}_n(K)$ such that $\widehat{M} = T^{-1}(M\phi(T) - \delta(T))$. Let us treat for instance System (2.23) written in the form (5.1) where $\phi : x \mapsto x/q$ with

$q \notin \{0,1\}$, $\delta = \mathrm{id}_K - \phi$ with degree $\omega = 0$ and

$$M = \begin{bmatrix} -1 & \frac{q^3}{x^3} & \frac{x^3}{q^2} \\ \frac{q^3+x^3}{q^2\,x} & \frac{q^2-x}{x} & \frac{-x^2-q}{q} \\ \frac{-x^2+q}{x^2} & 0 & \frac{x-q}{q} \end{bmatrix}.$$

Here the Poincaré rank is $p = 3$. The user should first enter the matrix M and then run the following commands:

```
> t:=x:
> PhiAction:= proc(M,x) return subs(x=x/q,M) end:
> DeltaAction:= proc(M,x) return PhiAction(M,x)-M end:
> omega:=0:
> SuperReduced(M,x,t,PhiAction,DeltaAction,omega,lambda);
```

$$\begin{bmatrix} \frac{-x+q}{x} & \frac{q^2-2\,x^2-q}{xq} & \frac{q^2+x}{q^2} \\ 0 & \frac{x-q}{q} & \frac{-x^2+q}{x^2} \\ \frac{q^2}{x^2} & \frac{x^5+q^4}{x^2q^2} & -1 \end{bmatrix}, \left[[2, \lambda\left(-q^3+\lambda^2\right)], [1, -q^3\left(\lambda-1\right)], [0, q^3]\right], \begin{bmatrix} 0 & 0 & 1 \\ x & x & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The minimal Poincaré rank obtained is $m = 2$. If one is interested in computing just m (without additional information), then it is sufficient to use the procedure **MinimalPoincareRank**. The package also contains the procedure **K_SimpleForm** which computes a k-simple form, for a given $k \in \{0, \ldots, p\}$, of any local pseudo-linear system (5.1).

5.3 Closed-form solutions of first order pseudo-linear systems

Let $K = C(x)$ be the field of rational functions in a variable x with coefficients in a constant field $C \subset \overline{\mathbb{Q}}$. In this section we demonstrate the use of some procedures devoted for the computation of: rational solutions of a first order pseudo-linear system, exponential solutions of differential systems, and hypergeometric solutions of difference systems.

5.3.1 The RatSols_1PLS procedure

Consider a first order pseudo-linear system of the form

$$\delta(\mathbf{y}) = M\,\phi(y), \tag{5.2}$$

where $M \in \mathbb{M}_n(K)$ and, either $\phi = \mathrm{id}_K$ and $\delta = \frac{d}{dx}$, or $\phi : x \mapsto q\,x + r$ and $\delta = \gamma(\mathrm{id}_K - \phi)$ with $\gamma \in K^*$. Here $r \in C$ and $q \in C^*$ is not a root of unity, but if $r \neq 0$ then q is allowed to be equal to 1. In Chapter 3 we have developed a unified algorithm to compute all rational solutions of any system of the form (5.2). It is called **RatSols_1PLS** and it is sketched precisely in Section 3.4. We have implemented the algorithm in our PseudoLinearSystems package as a procedure holding the same name. The algorithm consists mainly in two steps: the computation of a universal denominator and then the computation of polynomial solutions.

If $\phi \neq \mathrm{id}_K$, then we have seen that a universal denominator is composed of a ϕ-fixed part and a non ϕ-fixed part (if $q = 1$ then there is no ϕ-fixed part). We use the implementation of Algorithm **NonFixedPart** from Section 3.3.4 to compute the non ϕ-fixed part, and then (if $q \neq 1$) we use the implementation of Algorithm **FixedPart** from Section 3.3.3 to compute the ϕ-fixed part. Otherwise if $\phi = \mathrm{id}_K$, then (5.2) is simply a pure differential system and a universal denominator is obtained by calling the procedure **SimpleForm** at each finite singularity of the system. For the computation of polynomial solutions, we use the implementation of Algorithm **PolySols_1PLS** described in Section 3.2.

The **RatSols_1PLS** procedure takes as input:

- The matrix M with rational function entries in x.
- The variable x.
- The list $[q, r, \gamma]$ indicating that $\phi : x \mapsto q\,x + r$ and $\delta = \gamma(\mathrm{id}_K - \phi)$ for some $\gamma \in K^*$, or the list $[1, 0]$ indicating that $\phi = \mathrm{id}_K$ and $\delta = \frac{d}{dx}$.

The output is a matrix whose columns form a basis of all rational solutions of (5.2), or 0_n (the zero vector of dimension n) if there are no non-trivial rational solutions. For instance, in order to compute the rational solutions of System (5.2) with $\phi : x \mapsto 3\,x + 2$, $\delta = x\,(\mathrm{id}_K - \phi)$ and

$$M = \begin{bmatrix} \dfrac{6\,x^2 - 2\,x}{3\,x + 2} & 0 \\ -\dfrac{(234\,x + 36)\,(x+1)^3\,x}{3\,x^3 - 7\,x^2 + 4} & \dfrac{26\,x^3 + 30\,x^2 + 4\,x}{(x-1)\,(x-2)} \end{bmatrix},$$

the user should run the command

```
> RatSols_1PLS(M,x,[3,2,x]);
```

$$\begin{bmatrix} \frac{x}{(x+1)^2} & 0 \\ \frac{x^3-3x^2+2x-7}{x(x-1)(x-2)} & \frac{1}{x(x-1)(x-2)} \end{bmatrix}$$

The PseudoLinearSystems package contains another version of the **RatSols_1PLS** procedure, called the **RationalSolutions_1System** procedure, which is devoted to computing all rational solutions of first order differential, difference, or q-difference systems:

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

where $A(x) \in \mathbb{M}_n(K)$. This procedure takes as an input the matrix $A(x)$ defining the system, the variable x, and the type of the system: `'differential'`, `'difference'`, or `'qdifference'`. For instance let $A(x)$ be the matrix given by:

$$A = \begin{bmatrix} \frac{x^2-3x-3}{x(x-3)(x-1)} & \frac{x^2-5x+9}{3x(x-3)(x-1)} \\ \frac{-3x^2+12x-18}{(x-3)(x-1)} & \frac{x^3-2x^2-2x+6}{x(x-3)(x-1)} \end{bmatrix}. \tag{5.3}$$

```
> RationalSolutions_1System(A,x,'differential');
```

$$\begin{bmatrix} x^2 \\ 3x^2 \end{bmatrix}$$

```
> RationalSolutions_1System(A,x,'difference');    # No rational solutions
```

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Below are some additional useful procedures:

- **UnivDenom_PLS**: computes a universal denominator of a pseudo-linear system of the form (5.2). It takes the same inputs as procedure **RatSols_1PLS**.
- **PolySols_1PLS**: computes all polynomial solutions of a pseudo-linear system of the form (5.2). It also takes the same inputs as procedure **RatSols_1PLS**.
- **UnivDenom_PhiSystem**: computes a universal denominator of a ϕ-system $\phi(\mathbf{y}) = M\,\mathbf{y}$ where $\phi : x \mapsto q\,x + r$. It takes as input the matrix defining the

system, the variable, and the two rational numbers q and r such that $q \in C^*$ is not a root of unity but if $r \neq 0$ then q is allowed to be equal to 1.

- **RatSols_PhiSystem**, resp. **PolySols_PhiSystem**: computes all rational, resp. polynomial, solutions of a ϕ-system $\phi(\mathbf{y}) = M\,\mathbf{y}$ where $\phi : x \mapsto q\,x + r$. It takes the same inputs as procedure **UnivDenom_PhiSystem**.

- **Disp_PLS**: Given two polynomials $a(x)$ and $b(x)$ in $C[x]$. This procedure computes the set of the nonnegative integers s such that $\deg(\gcd(a, \phi^s(b))) > 0$ where ϕ is the automorphism defined by $\phi : x \mapsto q\,x + r$. In particular, this procedure uses the idea of the full factorisation explained in Section 3.3.4. It takes as input $a(x)$, $b(x)$, the variable x and the two rational numbers q and r.

5.3.2 The ExponentialSolutions procedure for differential systems

The package also contains the procedure **ExponentialSolutions** which computes all exponential solutions of a differential system $\mathbf{y}'(x) = A(x)\,\mathbf{y}(x)$ where $A(x) \in \mathbb{M}_n(K)$ (see Section 4.3.1.1). This procedure implements the algorithm from [84] but it uses the **SimpleForm** procedure to compute non-ramified local exponential parts (see Definition 4.6). This is the main difference compared to the current procedure in the Isolde package. **ExponentialSolutions** takes as input the matrix $A(x)$ and the variable x. It returns a matrix whose columns form a basis of all exponential solutions. Our package also includes the following procedures:

- **LocalExpParts**: computes a list of admissible non-ramified local exponential parts of the system at a given singularity. Again, this procedure uses the **SimpleForm** procedure to compute the different ω's in (4.18).

- **Exp_Sols**: another version of **ExponentialSolutions**. It returns a sequence

$$[u_1, P_1], \ldots, [u_m, P_m],$$

where $u_i \in K$ and $P_i \in C[x]^n$. Each list $[u_i, P_i]$ represents an exponential solution

$$\mathbf{y}(x) = e^{\int u_i(x)dx}\, P_i(x).$$

For instance let $A(x)$ be the matrix defined in (5.3). The poles of $A(x)$ are x, $x-1$ and $x-3$.

```
> LocalExpParts(A,x,x);
```

$$\left[-\frac{1}{x}, \frac{2}{x}\right]$$

```
> LocalExpParts(A,x,x-1);
```

$$\left[0, \frac{1}{x-1}\right]$$

```
> LocalExpParts(A,x,x-3);
```

$$\left[0, \frac{1}{x-3}\right]$$

```
> LocalExpParts(A,x,infinity);
```

$$[0, 1]$$

```
> Exp_Sols(A,x);
```

$$\left[\frac{x-1}{x}, \begin{bmatrix} -x+3 \\ -3\,x^2+9\,x \end{bmatrix}\right], \left[-\frac{1}{x}, \begin{bmatrix} x^3 \\ 3\,x^3 \end{bmatrix}\right]$$

```
> ExponentialSolutions(A,x);
```

$$\begin{bmatrix} -\frac{\mathrm{e}^x\,(x-3)}{x} & x^2 \\ -3\,\mathrm{e}^x\,(x-3) & 3\,x^2 \end{bmatrix}$$

5.3.3 The HypergeoemtericSolutions_Difference procedure for difference systems

PseudoLinearSystems also contains the **HypergeoemtericSolutions_Difference** procedure. This procedure computes all hypergeometric solutions of a first order difference system $\phi(\mathbf{y}(x)) = A(x)\,\mathbf{y}(x)$ where $\phi : x \mapsto x+1$ and $A(x) \in \mathrm{GL}_n(K)$, see Section 4.3.1.2. It implements the algorithm from [44] but it uses the **SimpleForm** procedure to compute the values of c in a hypergeometric solution (4.23). This is the main difference compared to the current procedure in Isolde. **HypergeoemtericSolutions_Difference** takes as input the matrix $A(x)$ and the variable x. It returns a matrix whose columns form a basis of all hypergeometric solutions. The **HS_Difference** procedure is another version of **HypergeoemtericSolutions_Difference**. It returns a sequence

$$[a_1, b_1, c_1, P_1], \ldots, [a_m, b_m, c_m, P_m],$$

where $a_i, b_i \in C[x]$, $c_i \in C^*$ and $P_i \in C[x]^n$. Each list $[a_i, b_i, c_i, P_i]$ represents a hypergeometric solution

$$\mathbf{y}(x) = c_i^x \operatorname{Sol}\left(\phi - \frac{a_i(x)}{b_i(x)}\right) P_i(x).$$

For instance let

$$A(x) = \begin{bmatrix} \frac{x^3+4x^2+4x-2}{(x+4)(x+2)(x+1)} & \frac{x^2+3x+1}{(x+2)(x+1)} & \frac{x+1}{x+4} & \frac{2x+4}{x+4} \\ -\frac{x^3+4x^2+4x-2}{(x+4)(x+2)(x+1)} & \frac{1}{(x+2)(x+1)} & -\frac{x+1}{x+4} & -\frac{x}{x+4} \\ -\frac{x(2x^2+8x+9)}{(x+4)(x+2)(x+1)} & -\frac{x^2+3x+1}{(x+2)(x+1)} & -\frac{2x+2}{x+4} & -\frac{2x}{x+4} \\ \frac{x+1}{x+4} & 0 & \frac{x+1}{x+4} & \frac{x}{x+4} \end{bmatrix}.$$

```
> HS_Difference(A,x);
```

$$\left[\left[x+1, x+4, -1, \begin{bmatrix} x^2+\frac{7}{2}x+\frac{5}{2} \\ -x^2-\frac{7}{2}x-\frac{5}{2} \\ -3x^2-\frac{23}{2}x-\frac{19}{2} \\ x^2+\frac{7}{2}x+\frac{5}{2} \end{bmatrix}\right], \left[x+1, x+4, 1, \begin{bmatrix} x+1 \\ -x-1 \\ -x+1 \\ x+1 \end{bmatrix}\right],\right.$$

$$\left[x+1, (x+4)(x+2), -1, \begin{bmatrix} -(x+3)(x+2)^2(x+1) \\ x^3+6x^2+11x+6 \\ (x+3)(x+2)^2(x+1) \\ 0 \end{bmatrix}\right],$$

$$\left.\left[x+1, (x+4)(x+2), 1, \begin{bmatrix} -x^4-6x^3-11x^2-6x \\ -x^3-6x^2-11x-6 \\ x^4+6x^3+11x^2+6x \\ 0 \end{bmatrix}\right]\right]$$

```
> HypergeometricSolutions_Difference(A,x);
```

$$\left[\begin{array}{cccc} \frac{(60\,x+150)\,(-1)^x}{(x+3)\,(x+2)} & \frac{60}{(x+3)\,(x+2)} & \frac{360\;(-1)^{x+1}\,(x+2)}{\Gamma\,(x+2)} & \frac{-360\,x}{\Gamma\,(x+2)} \\ \frac{(-60\,x-150)\,(-1)^x}{(x+3)\,(x+2)} & \frac{-60}{(x+3)\,(x+2)} & \frac{360\;(-1)^x}{\Gamma\,(x+2)} & \frac{-360}{\Gamma(x+2)} \\ \frac{(-180\,x^2-690\,x-570)\,(-1)^x}{(x+1)\,(x+3)\,(x+2)} & \frac{-60\,x+60}{(x+1)\,(x+3)\,(x+2)} & \frac{360\;(x+2)\,(-1)^x}{\Gamma\,(x+2)} & \frac{360\,x}{\Gamma\,(x+2)} \\ \frac{(60\,x+150)\,(-1)^x}{(x+3)\,(x+2)} & \frac{60}{(x+3)\,(x+2)} & 0 & 0 \end{array}\right]$$

5.4 Closed-form solutions of partial pseudo-linear systems

Let $K = C(x_1, \ldots, x_m)$ be the field of rational functions in the variables $x_1, \ldots, x_m$ with coefficients in a constant field $C \subset \overline{\mathbb{Q}}$. For $i = 1, \ldots, m$, let ϕ_i be a C-automorphism over K, and δ_i be a ϕ_i-derivation such that for all $j \neq i$, x_j is a constant with respect to ϕ_i and δ_i, i.e., $\phi_i(x_j) = x_j$ and $\delta_i(x_j) = 0$. We consider a partial pseudo-linear system of the form

$$\begin{cases} L_1(\mathbf{y}) := \delta_1(\mathbf{y}) - M_1\,\phi_1(\mathbf{y}) = 0, \\ \quad\vdots \\ L_m(\mathbf{y}) := \delta_m(\mathbf{y}) - M_m\,\phi_m(\mathbf{y}) = 0, \end{cases} \tag{5.4}$$

where $M_i \in \mathbb{M}_n(K)$ for all $i = 1, \ldots, m$. Each automorphism ϕ_i with its corresponding ϕ_i-derivation δ_i should be as in one of the following two cases:

Case 1:

$$\phi_i : (x_1, \ldots, x_i, \ldots, x_m) \mapsto (x_1, \ldots, q_i\,x_i{+}r_i, \ldots, x_m) \quad \text{and} \quad \delta_i = \gamma_i(\mathrm{id}_K{-}\phi_i) \text{ with } \gamma_i \in K^*.$$

Here $r_i \in C$ and $q_i \in C^*$ is not a root of unity, but if $r_i \neq 0$ then q_i is allowed to be 1.

Case 2:

$$\phi_i = \mathrm{id}_K \quad \text{and} \quad \delta_i = \frac{\partial}{\partial x_i}.$$

We assume that System (4.1) is integrable, i.e., it satisfies the integrability conditions:

$$[L_i, L_j] := L_i \circ L_j - L_j \circ L_i = 0, \quad \forall i, j = 1, \ldots, m, \tag{5.5}$$

where $L_i := I_n\,\delta_i - M_i\,\phi_i$ denotes the matrix pseudo-linear operator associated to the ith system of (5.4).

5.4.1 The RationalSolutions_PPLS procedure

In Chapter 4, we have first developed a recursive algorithm which computes all rational solutions of a system of the form (5.4). It is called **RationalSolutions_PPLS** and it is sketched in details in Section 4.2.2.2. We have implemented this algorithm as a procedure holding the same name. This procedure calls the **RatSols_1PLS** procedure to compute, at each iteration, rational solutions of a first order order pseudo-linear system. It takes as input:

- A list $[M_1, \ldots, M_m]$ of the matrices defining System (5.4).
- A list of the respective variables $[x_1, \ldots, x_m]$.
- A list $[E_1, \ldots, E_m]$ where each E_i is either a list $[q_i, r_i, \gamma_i]$ indicating **Case 1**, or a list $[0, 1]$ indicating **Case 2**.

If the integrability conditions (5.5) are satisfied, **RationalSolutions_PPLS** then returns a matrix whose columns form a basis of all rational solutions of (5.4). For example, to compute rational solutions of System 4.10, the user should run the following commands:

```
> Mat:= [ M[1], M[2], M[3] ]:
> X:= [ x[1], x[2], x[3] ]:
> E1:= [ 1, -1, 1 ]:
> E2:= [ 1/q, 0, 1 ]:
> E3:= [ 0, 1 ]:
> RationalSolutions_PPLS( Mat, X, [ E1, E2, E3 ] );
```

$$\begin{bmatrix} \dfrac{x_3}{x_2} & x_1 \\ \dfrac{x_2}{(x_3 + x_1)\, q} & 0 \end{bmatrix}$$

While for System (4.11), the user should run:

```
> Mat:= [ M[1], M[2], M[3] ]:
> X:= [ x[1], x[2], x[3] ]:
> E1:= [ 1, -5, 1 ]:
> E2:= [ -3, -5, 1 ]:
> E3:= [ -3, 0, 1]:
> RationalSolutions_PPLS( Mat, X, [ E1, E2, E3 ] );
```

$$\begin{bmatrix} \dfrac{x_3}{x_2} & x_1 \\ 1 & \dfrac{x_2}{x_3 + x_1} \end{bmatrix}$$

In the sequel we give the following useful procedures:

- **PolynomialSolutions_PPLS**: computes all polynomial solutions of a partial pseudo-linear system (5.4). It takes the same inputs as the procedure **RationalSolutions_PPLS**.

- **RationalSolutions**, resp. **PolynomialSolutions**: another version of **RationalSolutions_PPLS**, resp. **PolynomialSolutions_PPLS**. It computes all rational, resp. polynomial solutions of a partial pseudo-linear system composed of first order differential, difference and q-difference systems (such as System 4.9). The two procedures take as input a list of the matrices defining the system, a list of the respective variables, and a list of respective types: `'differential'`, `'difference'`, or `'qdifference'`.

- **EigenRing**: computes the eigenring of a partial pseudo-linear system (5.4) (see Section 4.2.2.5). It takes the same inputs as the **RationalSolutions_PPLS** procedure.

5.4.2 The HypergeometricSolutions procedure

In the second part of Chapter 4, we have developed the recursive algorithm **HypergeometricSolutions** which computes all hypergeometric solutions of a partial pseudo-linear system (5.4). The steps of the algorithm are provided in Section 4.3.3. Our current implementation only deals with a partial pseudo-linear system composed of first order differential and /or difference systems. The procedure **HypergeometricSolutions** uses procedure **ExponentialSolutions**, resp. **HypergeometricSolutions_Difference**, to compute hypergeometric solutions of a first order differential, resp. difference, system. It takes as input a list of the matrices defining the system, a list of the respective variables, and a list of respective types: `'differential'` or `'difference'`. For instance, to treat System (4.31), the user should run:

```
> HypergeometricSolutions([A,B],[k,x],['difference','differential']);
```

$$\begin{bmatrix} \dfrac{k\,\mathrm{e}^{-x}}{(-x+k)^2} & 0 & 0 \\ \dfrac{k\,\mathrm{e}^{-x}}{(-x+k)^2} & x^k\,k\,\mathrm{e}^{x} & 0 \\ 0 & x^{k+1}\,k\,\mathrm{e}^{x} & x^k\,\Gamma(k) \end{bmatrix}$$

While to treat the system

$$\left\{\begin{array}{l} \dfrac{\partial}{\partial x_1}\mathbf{y}(x_1,x_2,x_3) = A_1(x_1,x_2,x_3)\,\mathbf{y}(x_1,x_2,x_3) \\ \dfrac{\partial}{\partial x_2}\mathbf{y}(x_1,x_2,x_3) = A_2(x_1,x_2,x_3)\,\mathbf{y}(x_1,x_2,x_3) \\ \dfrac{\partial}{\partial x_3}\mathbf{y}(x_1,x_2,x_3) = A_3(x_1,x_2,x_3)\,\mathbf{y}(x_1,x_2,x_3) \end{array}\right.$$

with

$$A_1 = \begin{bmatrix} \dfrac{4\,x_3+x_1}{4\,x_1x_3+x_1} & \dfrac{2\,x_1x_3-2\,x_3}{4\,x_1x_3+x_1} \\ \dfrac{2\,x_1-2}{4\,x_1x_3+x_1} & \dfrac{4\,x_1x_3+1}{4\,x_1x_3+x_1} \end{bmatrix}, \quad A_2 = \begin{bmatrix} \dfrac{4\,x_3}{4\,x_3x_2{}^2+x_2{}^2} & -\dfrac{2\,x_3}{4\,x_3x_2{}^2+x_2{}^2} \\ -\dfrac{2}{4\,x_3\,x_2^2+x_2^2} & \dfrac{1}{(4\,x_3x_2{}^2+x_2{}^2)} \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 1 \\ \dfrac{3+4\,x_3}{4\,x_3{}^2+x_3} & \dfrac{16\,x_3{}^2-3}{8\,x_3{}^2+2\,x_3} \end{bmatrix},$$

the user should run:

```
> Mat:= [ A[1], A[2], A[3] ]:
> X:= [ x[1], x[2], x[3] ]:
> type:= [ 'differential', 'differential', 'differential' ]:
> HypergeometricSolutions( Mat, X, type );
```

$$\begin{bmatrix} -\dfrac{2\,x_1\,\mathrm{e}^{-\frac{1}{x_2}}}{\sqrt{x_3}} & \dfrac{\mathrm{e}^{x_1+2\,x_3}}{2} \\ \dfrac{x_1\,\mathrm{e}^{-\frac{1}{x_2}}}{x_3^{3/2}} & \mathrm{e}^{x_1+2\,x_3} \end{bmatrix}$$

Bibliography

[1] S. Abramov. On the summation of rational functions. *USSR Computational Mathematics and Mathematical Physics,* 11(4), 324-330, 1971.

[2] S. Abramov. Rational solutions of linear differential and difference equations with polynomial coefficients. *USSR Computational Mathematics and Mathematical Physic,* 29(6), 7-12, 1991.

[3] S. Abramov. EG-eliminations. *Journal of Difference Equations and Applications*, 5(4-5), 393-433, 1999.

[4] S. Abramov. Rational solutions of first order linear q-difference systems. *In Proceedings of FPSAC'99, Barcelona, Spain,* 1-9, 1999.

[5] S. Abramov. A direct algorithm to compute rational solutions of first order linear q-difference systems. *Discrete Mathematics,* 246(1-3), 3-12, 2002.

[6] S. Abramov. EG-eliminations as a tool for computing rational solutions of linear q-difference systems of arbitrary order with polynomial coefficients. *Computer Algebra International Conference,* 2017.

[7] S. Abramov, M. A. Barkatou. Rational solutions of first order linear difference systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 124-131, 1998.

[8] S. Abramov, M. Bronstein. On solutions of linear functional systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 1-6, 2001.

[9] S. Abramov, A. Gheffar, D. Khmelnov. Factorization of polynomials and gcd computations for finding universal denominators. *In Proceedings of the 12th International Conference on Computer Algebra in Scientific Computing,* 4-18, 2010.

[10] S. Abramov, A. Gheffar, D. Khmelnov. Rational solutions of linear difference equations: universal denominators and denominator bounds. *Programming and Computer Software,* 37(2), 78-86, 2011.

[11] S. Abramov, P. Paule, M. Petkovšek. q-Hypergeometric solutions of q-difference equations. *Discrete Mathematics,* 180, 3-22, 1998.

[12] S. Abramov, M.Petkovšek, A. A. Ryabenko. Hypergeometric solutions of first-order linear difference systems with rational-function coefficients. *International Workshop on Computer Algebra in Scientific Computing,* 1-14, 2015.

[13] S. Abramov, A. A. Ryabenko, D. Khmelnov. Laurent, rational, and hypergeometric solutions of linear q-difference systems of arbitrary order with polynomial coefficients. *Programming and Computer Software,* 44(2), 120-130, 2017.

[14] C. Admas. On the linear ordinary q-difference equation. *Annals of Mathematics,* 30(2), 195-205, 1929.

[15] M. A. Barkatou. On the reduction of linear systems of difference equations. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 1-6, 1989.

[16] M. A. Barkatou. Contribution à l'étude des équations différentielles et aux différences dans le champ complexe. *PhD Thesis, Institut National Polytechnique de Grenoble,* 1989.

[17] M. A. Barkatou. An algorithm for computing a companion block diagonal form for a system of linear differential equations. *Applicable Algebra in Engineering, Communication and Computing,* 4(3), 185-195, 1993.

[18] M. A. Barkatou. A rational version of Moser's algorithm. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 297-302, 1995.

[19] M. A. Barkatou. An algorithm to compute the exponential part of a formal fundamental matrix solution of a linear differential system. *Applicable Algebra in Engineering, Communication and Computing,* 8(1), 1-23, 1997.

[20] M. A. Barkatou. On rational solutions of systems of linear differential equations. *Journal of Symbolic Computation,* 28, 547-567, 1999.

[21] M. A. Barkatou. Rational solutions of matrix difference equations: the problem of equivalence and factorization. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 277-282, 1999.

[22] M. A. Barkatou. Factoring Systems of Linear Functional Systems Using Eigenrings. *Computer Algebra 2006, Latest Advances in Symbolic Algorithms, World Scientific,* 22-42, 2007.

[23] M. A. Barkatou, G. Broughton, E. Pflügel. Regular systems of linear functional equations and applications. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 15-22, 2008.

[24] M. A. Barkatou, G. Broughton, E. Pflügel. A monomial-by-monomial method for computing regular solutions of systems of pseudo-linear equations. *Mathematics in Computer Science,* 4(2-3), 267-288, 2010.

[25] M. A. Barkatou, G. Chen. Computing the exponential part of a formal fundamental matrix solution of a linear difference system. *Journal of Difference Equations and Applications,* 5, 117-142, 2007.

[26] M. A. Barkatou, G. Chen. Some formal invariants of linear difference systems and their Computations. *Crelle's Journal,* 2001, 1-23, 2001.

[27] M. A. Barkatou, T. Cluzeau, C. El Bacha. Simple forms of higher-order linear differential systems and their applications in computing regular solutions. *Journal of Symbolic Computation,* 46(6), 633-658, 2011.

[28] M. A. Barkatou, T. Cluzeau, C. El Bacha. On the computation of simple forms and regular solutions of linear difference systems. *Advances in Computer Algebra, WWCA 2016, Springer Proc. in Mathematics & Statistics,* 226, 19-49, 2018.

[29] M. A. Barkatou, T. Cluzeau, C. El Bacha, J. A. Weil. Computing closed form solutions of integrable connections. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 43-50, 2012.

[30] M. A. Barkatou, T. Cluzeau, A. El Hajj. Simple forms and rational solutions of pseudo-linear systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 26-33, 2019.

[31] M. A. Barkatou, T. Cluzeau, A. El Hajj. PseudoLinearSystems – a maple package for studying systems of pseudo-linear equations. *Maple in Mathematics Education and Research,* 327-329, 2019.

[32] M. A. Barkatou, T. Cluzeau, A. El Hajj. The Pseudo-linearSystems package. http://www.unilim.fr/pages_perso/ali.el-hajj/PseudoLinearSystems.html

[33] M. A. Barkatou, T. Cluzeau, A. El Hajj. On rational solutions of pseudo-linear systems. *Computer Algebra in Scientific Computing,* 42-61, 2021.

[34] M. A. Barkatou, C. El Bacha. On k-simple forms of first-order linear differential systems and their computation. *Journal of Symbolic Computation,* 54, 36-58, 2013.

[35] M. A. Barkatou, C. El Bacha, G. Labahn, E. Pflügel. On simultaneous row and column reduction of higher-order linear differential systems. *Journal of Symbolic Computation,* 49, 45-64, 2013.

[36] M. A. Barkatou, M. Jaroschek. Removing apparent singularities of linear difference systems. *Journal of Symbolic Computation,* 102, 86-107, 2021.

[37] M. A. Barkatou, E. Pflügel. On the Equivalence Problem of Linear Differential Systems and its Application for Factoring Completely Reducible Systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 268-275, 1998.

[38] M. A. Barkatou, E. Pflügel. An algorithm computing the regular formal solutions of a system of linear differential equations. *Journal of Symbolic Computation,* 28(4-5), 569-587, 1999.

[39] M. A. Barkatou, E. Pflügel. The ISOLDE package. *A SourceForge Open Source project.* http://isolde.sourceforge.net/, 2006.

[40] M. A. Barkatou, E. Pflügel. Computing super-irreducible forms of systems of linear differential equations via Moser-reduction: a new approach. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 1-8, 2007.

[41] M. A. Barkatou, E. Pflügel. On the Moser- and super-reduction algorithms of systems of linear differential equations and their complexity. *Journal of Symbolic Computation,* 44(8), 1017-1036, 2009.

[42] M. A. Barkatou, F. Richard-Jung. Formal solutions of linear differential and difference equations. *Programming and Computer Software,* 23(1), 17-30, 1997.

[43] M. A. Barkatou, J. Saadé, J.-A. Weil. Formal reduction of singular linear differential systems using eigenrings: A refined approach. *Journal of Symbolic Computation,* 102, 231-258, 2019.

[44] M. A. Barkatou, M. Van Hoeij. Hypergeometric solutions of systems of linear difference equations. *Computer Algebra and Functional Equations CAFE, July* 2006.

[45] M. Birkhof. The generalized Riemann problem. *Riemann Solvers and Numerical Methods for Fluid Dynamics,* 625-653, 1913.

[46] M. Bronstein, Z. Li, M. Wu. Picard-Vessiot extensions for linear functional systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 68-75, 2005.

[47] M. Bronstein, M. Petkovšek. On ore rings, linear operators and factorisation. *Programmirovanie,* 20, 27-45, 1994.

[48] M. Bronstein, M. Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science,* 157, 3-33, 1996.

[49] G. Broughton. Symbolic algorithms for the local analysis of systems of pseudo-linear equations. *PhD Thesis, Kingston University,* 2013.

[50] G. Chen. Solutions formelles de systèmes d'équations différentielles linéaires ordinaires homogènes. *PhD thesis, Université Joseph Fourrier - Grenoble I*, 1990.

[51] R. C. Churchull, J. J. Kovacic. Cyclic vectors. *In Differential Algebra and Related Topics, Proceedings of the International Workshop, pages 191-218, Rutgers University, Newark, November 2002. River Edge, NJ, World Scientific Publishing Company.*

[52] T. Cluzeau. Algorithmique modulaire des équations différentielles linéaires. *PhD Thesis, Université de Limoges,* 2004.

[53] T. Cluzeau, M. van Hoeij. A modular algorithm for computing the exponential solutions of a linear differential operator. *Journal of Symbolic Computation,* 38(3), 1043-1076, 2004.

[54] T. Cluzeau, M. van Hoeij. Computing hypergeometric solutions of linear difference equations. *Applicable Algebra in Engineering, Communication and Computing, Springer Verlag,* 17, 83-115, 2006.

[55] T. Cluzeau, A. Quadrat. Factoring and decomposing a class of linear functional systems. *Linear Algebra and its Applications,* 428(1), 324-381, 2008.

[56] E. Coddington, N. Levinson. Theory of Ordinary Differential Equations. *McGraw-Hill Book Company, Inc., New York,* 1955.

[57] F. Cope. Formal solutions of irregular linear differential equations, Part II. *American Journal of Mathematics,* 58, 130-140, 1936.

[58] V. Dietrich. Zur Reduktion von linearen Differentialgleichungssystemen. *Mathematische Annalen,* 237, 79-95, 1978.

[59] C. El Bacha. Méthodes algébriques pour la résolution d'équations différentielles matricielles d'ordre arbitraire. *PhD Thesis, Université de Limoges,* 2011.

[60] R. Feng, M. F. Singer, M. Wu. An algorithm to compute Liouvillian solutions of prime order linear difference-differential equations. *Journal of Symbolic Computation,* 45(3), 306-323, 2010.

[61] C. Hardouin, M. Singer. Differential Galois theory of linear difference equations. *Mathematische Annalen,* 350(1), 243-244, 2011.

[62] W. Harris. Linear systems of difference equations. *Contributions to Differential Equations 1,* 489-518, 1963.

[63] A. Hilali, A. Wazner. Un algorithme de calcul de l'invariant de Katz d'un système différentiel linéaire. *Annales de l'institut Fourier,* 36(3), 67-81, 1986.

[64] A. Hilali. Solutions formelles de systèmes différentiels linéaires au voisinage d'un point singulier. *PhD thesis, L'université scientifique, technologique et médicale de Grenoble,* 1987.

[65] A. Hilali. Contribution à l'étude des points singuliers des systèmes différentiels linéaires. *PhD thesis, L'université scientifique, technologique et médicale de Grenoble,* 1987.

[66] A. Hilali. Calcul des invariants de Malgrange et de Gérard-Levelt d'un système différentiel linéaire en un point singulier irrégulier. *Journal of Differential Equations,* 69, 404-421, 1987.

[67] A. Hilali, A. Wazner. Formes super-irréductibles des systèmes différentiels linéaires. *Numerische Mathematik,* 50, 429-449, 1987.

[68] E. L. Ince. Ordinary Differential Equations. *Dover Publications, Inc.*, 1926.

[69] N. Jacobson. Pseudo-linear transformations. *Annals of Mathematics,* 38(2), 484-507, 1937.

[70] M. Kauers, C. Schneider. Partial denominator bounds for partial linear difference equations. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 211-218, 2010.

[71] M. Kauers, C. Schneider. A refined denominator bounding algorithm for multivariate linear difference equations. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 201-208, 2011.

[72] D. E. Khmelnov. Improved algorithms for solving difference and q-difference equations. *Programming and Computer Software,* 26(2), 107-115, 2000. Vol. 26, No. 2: 107-115 (translated from Programmirovanie No. 2), 2000.

[73] A. Levelt. Stabilizing differential operators: a method for computing invariants at irregular singularities. *In M. Singer, editor, Differential Equations and Computer Algebra,* 181-228, 1991.

[74] Z. Li, M. F. Singer, M. Wu, D. Zheng. A recursive method for determining the one-dimensional submodules of Laurent-Ore modules. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 200-208, 2006.

[75] Z. Li, M. Wu. On solutions of linear systems and factorization of Laurent-Ore modules. *Computer Algebra 2006, World Scientific,* 109-136, 2007.

[76] Y. O. Macuatn. Formal Solutions of Scalar Singularly-Perturbed Linear Differential Equations. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 113-120, 1999.

[77] Y. K. Man, F. J. Write. Fast polynomial dispersion computation and its application to indefnite summation. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 175-180, 1994.

[78] J. Moser. The order of a singularity in Fuchs theory. *Mathematische Zeitschrift,* 72, 379-398, 1960.

[79] O. Ore. Formale theorie der linearen differentialgleichungen. *Journal für die reine und angewandte Mathematik,* 167, 221-234, 1932.

[80] S. V. Paramonov. *On rational solutions of linear partial differential or difference equations. Programming and Computer Software,* 39(2), 57-60, 2013.

[81] S. V. Paramonov. Checking existence of solutions of partial differential equations in the fields of Laurent series. *Programming and Computer Software,* 40, 58-62, 2014.

[82] M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation,* 14(2-3), 243-264, 1992.

[83] M. Petkovšek, B. Salvy. Finding all hypergeometric solutions of linear differential equations. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 27-33, 1993.

[84] E. Pflügel. An algorithm for computing exponential solutions of first order linear differential systems. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation,* 146-171, 1997.

[85] E. Pflügel. Résolution symbolique des sysètmes différentiels linéaires. *PhD thesis, Université Joseph Fourrier,* 1998.

[86] E. Pflügel. Effective formal reduction of linear differential systems. *Applicable Algebra in Engineering, Communication and Computing,* 10(2), 153-187, 2000.

[87] J. Saadé. Méthodes symboliques pour les systèmes différentiels linéaires à singularité irrégulière. *PhD Thesis, Université de Limoges,* 2019.

[88] M. F. Singer. Liouvillian Solutions of Linear Differential Equations with Liouvillian Coefficients. *Computers and Mathematics. Springer, New York,* 182-191, 1989.

[89] M. F. Singer. Testing reducibility of linear differential operators : a group theoretic perspective. *Applicable Algebra in Engineering, Communication and Computing,* 7(2), 77-104, 1996.

[90] E. Tournier. Solutions formelles d'équations Différentielles, le logiciel de calcul formel: DÉSIR, étude théorique et réalisation. *PhD thesis, L'université scientifique, technologique et médicale de Grenoble,* 1987.

[91] W. Trjitzinsky. Analytic theory of linear q-difference equations. *ACTA Mathematica,* 61,1-38, 1933.

[92] H. Turritin. The formal theory of systems of irregular homogeneous linear difference equations. *Boltin de la Soc. Mexicana,* 5, 255-264, 1960.

[93] M. Van Hoeij. Formal solutions and factorization of differential operators with power series coefficients. *Journal of Symbolic Computation,* 24, 1-30, 1997.

[94] J. Von zur Gathen, J. Gerhard. Modern computer algebra. *Cambridge University Press, New York,* 2003.

[95] M. Wu. On solutions of linear functional systems and factorization of modules over Laurent-Ore algebras. *PhD Thesis, University of Nice-Sophia Antipolis,* 2005.

Abstract: This thesis is concerned with the development of symbolic algorithms in Computer algebra. We study systems of pseudo-linear equations: a large class of linear functional systems including differential, difference and q-difference systems.

The thesis consists of three major parts. In the first part, we are interested in the local analysis of pseudo-linear system near a singularity. We first develop a direct algorithm for computing simple forms of pseudo-linear systems. Whilst direct algorithms for computing simple forms have been already proposed for differential and difference systems, no unifying one for pseudo-linear systems was known prior to our work. Then we show how the reduction to a simple form can be used to compute efficiently local data for pseudo-linear systems.

The second part deals with the computation of closed-form solutions. Firstly, we present a generic and efficient algorithm for computing rational solutions of first order pseudo-linear systems. Then we develop two new recursive algorithms for computing rational and hypergeometric solutions of partial pseudo-linear systems with arbitrary number of variables.

An important contribution of this thesis arises in the implementation of all the algorithms in Maple as part of our freely available package PseudoLinearSystems. In the last part of the thesis, we provide a demonstration of several procedures contained in the package.

Keywords: Computer algebra, Functional equations, Pseudo-linear systems, Simple forms, Rational solutions, Implementation.

Resumé: Cette thèse porte sur le développement d'algorithmes symboliques en calcul formel. Nous étudions des systèmes d'équations pseudo-linéaires: une grande classe de systèmes fonctionnels linéaires comprenant les systèmes différentiels, différences et q-différences.

La thèse se compose de trois grandes parties. Dans la première partie, nous nous intéressons à l'analyse locale d'un système pseudo-linéaire au voisinage d'une singularité. Nous développons d'abord un algorithme direct pour le calcul de formes simples de systèmes pseudo-linéaires. Alors que des algorithmes directs pour le calcul de formes simples ont déj été proposés pour les systèmes différentiels et différences, aucun unificateur pour les systèmes pseudo-linéaires n'était connu avant nos travaux. Ensuite, nous montrons comment la réduction à une forme simple peut être utilisée pour calculer efficacement des données locales pour des systèmes pseudo-linéaires.

La deuxième partie traite du calcul des solutions de forme fermée. Tout d'abord, nous présentons un algorithme générique et efficace pour le calcul de solutions rationnelles de systèmes pseudo-linéaires du premier ordre. Ensuite, nous développons deux nouveaux algorithmes récursifs pour le calcul de solutions rationnelles et hypergéométriques de systèmes pseudo-linéaires avec un nombre arbitraire de variables..

Une contribution importante de cette thèse se pose dans l'implémentation de tous les algorithmes dans Maple dans le cadre de notre package `PseudoLinearSystems` disponible gratuitement. Dans la dernière partie de la thèse, nous proposons une démonstration de plusieurs procédures contenues dans le package.

Mots-clés: Calcul formel, Équations fonctionnelles, Systèmes pseudo-linéaires, Formes simples, Solutions rationnelles, Implémentation.