Puiseux Series and Algebraic Solutions of First Order Autonomous AODEs – A MAPLE Package

François Boulier[1], José Cano[2], Sebastian Falkensteiner[3](✉), and J. Rafael Sendra[4]

[1] Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL, Lille, France
francois.boulier@univ-lille.fr

[2] Dpto. Algebra, análisis matemático, geometría y topología, Universidad de Valladolid, Valladolid, Spain
jcano@agt.uva.es

[3] Research Institute for Symbolic Computation (RISC), Johannes Kepler University Linz, Linz, Austria
falkensteiner@risc.jku.at

[4] Universidad de Alcalá, Dpto. Física y Matemáticas, Alcalá de Henares, Madrid, Spain
rafael.sendra@uah.es

Abstract. There exist several methods for computing exact solutions of algebraic differential equations. Most of the methods, however, do not ensure existence and uniqueness of the solutions and might fail after several steps, or are restricted to linear equations. The authors have presented in previous works a method to overcome this problem for autonomous first order algebraic ordinary differential equations and formal Puiseux series solutions and algebraic solutions. In the first case, all solutions can uniquely be represented by a sufficiently large truncation and in the latter case by its minimal polynomial.

The main contribution of this paper is the implementation, in a `MAPLE` package named `FirstOrderSolve`, of the algorithmic ideas presented therein. More precisely, all formal Puiseux series and algebraic solutions, including the generic and singular solutions, are computed and described uniquely. The computation strategy is to reduce the given differential equation to a simpler one by using local parametrizations and the already known degree bounds.

Keywords: Maple · Symbolic computation · Algebraic differential equation · Formal Puiseux series solution · Algebraic solution

1 Introduction

The problem of finding power series solutions of ordinary differential equations has been extensively studied in the literature. A method to compute generalized formal power series solutions, i.e. power series with real exponents, and to describe their properties is the Newton polygon method. A description of this method is given in [11,12] and more recently in [1,6,13]. In [4], the second author, using the Newton polygon method, gives a theoretical description of all generalized formal power series solutions of a non-autonomous first order ordinary differential equation as a finite set of one parameter families of generalized formal power series. This description of the solutions is in general not algorithmic by several reasons. One of them is that there is no bound on the number of terms which have to be computed in order to guarantee the existence of a generalized formal power series solution when extending a given truncation of a determined potential solution. Also the uniqueness of the extension can not be ensured a-priori.

In [5] this problem has been overcome by the authors for autonomous first order differential equations by using a local version of the algebro-geometric approach introduced in [9].

In [15] they derive an associated differential system to find rational general solutions of non-autonomous first order differential equations by considering rational parametrizations of the implicitly defined curve. We instead consider its places and obtain an associated differential equation of first order and first degree which can be transformed into an equation of a very specific type [3]. Using the known bounds for computing places of algebraic curves (see e.g. [7]), existence and uniqueness of the solutions and the termination of our computations can be ensured.

In [2] the results of [9,10] are generalized to algebraic solutions. It is well known that algebraic solutions can be represented as Puiseux series. The advantage is that they can be fully described by its minimal polynomial. In this package we mainly follow [2], but we use an adapted version of the algorithm there for deciding the existence of algebraic solutions and computing all of them in the affirmative case.

2 Theoretical and Algorithmic Framework

In this section we recall the main notions and results that are used in our implementations. For further details we refer to [5] in the case of formal Puiseux series and to [2] in the case of algebraic solutions.

Let $\mathbb{K}$ be a computable field of characteristic zero such as the rational numbers $\mathbb{Q}$ and let us denote by $\overline{\mathbb{K}}$ its algebraic closure. Let us consider the differential equation

$$F(y, y') = 0, \tag{1}$$

where $F \in \mathbb{K}[y, p]$ is square-free and non-constant in the variables y and p. We are looking for formal Puiseux series and algebraic solutions of (1). In the case of formal Puiseux series solutions we will represent the full series by a sufficiently large truncation such that existence and uniqueness are guaranteed. In the case of algebraic solutions we look for its minimal polynomial.

We associate to (1) the affine algebraic curve $C(F) \subset \overline{\mathbb{K}}^2$ defined by the zero set of $F(y, p)$ in $\overline{\mathbb{K}}^2$. We denote by $\mathscr{C}(F)$ the Zariski closure of $C(F)$ in $\overline{\mathbb{K}}_\infty^2$, where $\overline{\mathbb{K}}_\infty = \overline{\mathbb{K}} \cup \{\infty\}$ denotes the one-point compactification of $\overline{\mathbb{K}}$. In the case of formal Puiseux series solutions we will look for local parametrizations of $\mathscr{C}(F)$ and in the case of algebraic solutions for algebraic parametrizations, respectively.

2.1 Formal Puiseux Series Solutions

Formal Puiseux series can either be expanded around a finite point or at infinity. In the first case, since Eq. (1) is invariant under translation of the independent variable, without loss of generality we can assume that the formal Puiseux series is expanded around zero and it is of the form $\varphi(x) = \sum_{j \geq j_0} a_j \, x^{j/n}$, where $a_j \in \overline{\mathbb{K}}$, $n \in \mathbb{Z}_{>0}$ and $j_0 \in \mathbb{Z}$. In the case of infinity we can use the transformation $x = 1/z$ obtaining the (non-autonomous) differential equation $F(y(z), -z^2 y'(z)) = 0$. In order to deal with both cases in a unified way, we will study equations of the type

$$F(y(x), (1-h)x^h y'(x)) = 0, \tag{2}$$

with $h \in \{0, 2\}$ and its formal Puiseux series solutions expanded around zero. We note that for $h = 0$ Eq. (2) is equal to (1) and for $h = 2$ the case of formal Puiseux series solutions expanded at infinity is treated.

We use the notations $\mathbb{L}[[x]]$ for the ring of formal power series, $\mathbb{L}((x))$ for its fraction field and $\mathbb{L}((x))^* = \bigcup_{n \geq 1} \mathbb{L}((x^{1/n}))$ for the field of formal Puiseux series expanded at zero with coefficients in some field $\mathbb{L}$. We call the minimal natural number n such that $\varphi(x)$ belongs to $\mathbb{L}((x^{1/n}))$ the *ramification order* of $\varphi(x)$. Moreover, for $\varphi(x) = \sum_{j \geq j_0} a_j \, x^{j/n}$ with $a_{j_0} \neq 0$ we call $j_0/n \in \mathbb{Q}$ the order of φ, denoted by $\operatorname{ord}_x(\varphi(x))$, and set $\operatorname{ord}_x(\varphi(x)) = \infty$ for $\varphi = 0$.

Additionally to (2) we may require that a formal Puiseux series solution $y(x)$ of (2) fulfills the initial conditions $y(0) = y_0, ((1-h)x^h y'(x))(0) = p_0$ for some fixed $\mathbf{p}_0 = (y_0, p_0) \in \overline{\mathbb{K}}_\infty^2$. In the case where $y(0) = \infty$, $\tilde{y}(x) = 1/y(x)$ is a Puiseux series solution of a new first order differential equation of the same type, namely the equation given by the numerator of the rational function $F(1/y, -(1-h)x^h p/y^2)$, and $\tilde{y}(0) \in \overline{\mathbb{K}}$. Therefore, in the sequel, we may assume that $\mathbf{p}_0 \in \overline{\mathbb{K}} \times \overline{\mathbb{K}}_\infty$.

Formal Parametrizations. Let us recall some classical terminology on local parametrizations of algebraic curves and its algorithmic aspects, for further details see e.g. [7,16].

A *formal parametrization* centered at $\mathbf{p}_0 \in \mathscr{C}(F)$ is a pair of formal Puiseux series $A(t) \in \overline{\mathbb{K}}((t))^2 \backslash \overline{\mathbb{K}}^2$ such that $A(0) = \mathbf{p}_0$ and $F(A(t)) = 0$. In the set of all formal parametrizations of $\mathscr{C}(F)$ we introduce the equivalence relation $\sim$ by defining $A(t) \sim B(t)$ if and only if there exists a formal power series $s(t) \in \overline{\mathbb{K}}[[t]]$ of order one such that $A(s(t)) = B(t)$. A formal parametrization is said to be *irreducible* if it is not equivalent to another one in $\overline{\mathbb{K}}((t^m))^2$ for some $m > 1$. An equivalence class of an irreducible formal parametrization $(a(t), b(t))$ is called a *place* of $\mathscr{C}(F)$ centered at the common center point $\mathbf{p}_0$ and is denoted by $[(a(t), b(t))]$.

Let $\mathrm{IFP}(\mathbf{p}_0)$ denote the set of all irreducible formal parametrizations of $\mathscr{C}(F)$ at $\mathbf{p}_0$ and $\mathrm{Places}(\mathbf{p}_0)$ containing the places of $\mathscr{C}(F)$ centered at $\mathbf{p}_0$. Computationally we have to truncate the formal parametrizations. There are bounds presented in [7,14] such that

1. the truncations of the formal parametrizations $(a(t), b(t))$ at $\mathbf{p}_0$ are in one-to-one correspondence to $\mathrm{Places}(\mathbf{p}_0)$;
2. the orders $\mathrm{ord}_t(a(t) - y_0), \mathrm{ord}_t(b(t))$ are determined;
3. no further extension of the ground field for computing the following coefficients have to be done.

More precisely, in [7], $N = 2(\deg_p(F) - 1)\deg_y(F) + 1$ is proved to be a bound satisfying the above requirements, under the hypothesis that the leading coefficient of $F(y, p)$, seen as polynomial in p and denoted by $\mathrm{lc}_p(F)(y)$, is constant. Her proof can be generalized straightforward to the case where $\mathrm{lc}_p(F)(y)$ is of order zero. The general case is reduced to the previous one by a change of variable $q(y) = y^\nu\, p(y)$, where ν is the order of $\mathrm{lc}_p(F)(y)$. In this way, the bound above generalizes as

$$\begin{aligned} N &= (2\ \deg_p F - 1)\,(\deg_y F + \nu\,(\deg_p F - 1)) + 1 \\ &\leq (2\ \deg_p F - 1)\ \deg_p F\ \deg_y F + 1. \end{aligned} \tag{3}$$

In the literature other possible bounds exist such as in [14] given in terms of the Milnor number.

Let us note that the solutions of (2) will be independent of the chosen representative of a place. Hence, regarding uniqueness of the prolongation, number of field extensions, etc. it does not matter which local parametrization we chose (for example classical Puiseux parametrizations or rational Puiseux parametrization [7]). For representing the solution parametrizations, which is not the goal of the current paper, however, it would be relevant.

Puiseux Solution Place. Let $\mathrm{Sol}_{\overline{\mathbb{K}}((x))^*}(\mathbf{p}_0)$ be the set containing the non-constant formal Puiseux series solutions of Eq. (2), expanded at zero, with coefficients in $\overline{\mathbb{K}}$ and with $\mathbf{p}_0$ as initial values. Then the mapping $\varDelta : \mathrm{Sol}_{\overline{\mathbb{K}}((x))^*}(\mathbf{p}_0) \longrightarrow \mathrm{IFP}(\mathbf{p}_0)$ defined as

$$\varDelta(y(x)) = \left(y(t^n), (1-h)t^{hn}\,\frac{d\,y}{d\,x}(t^n)\right),$$

where n is the ramification order of $y(x)$, is well-defined and injective. Moreover, we denote by $\delta : \mathrm{Sol}_{\overline{\mathbb{K}}((x))^*}(\mathbf{p}_0) \longrightarrow \mathrm{Places}(\mathbf{p}_0)$ the map $\delta(y(x)) = [\varDelta(y(x))]$.

An irreducible formal parametrization $A(t) \in \mathrm{IFP}(\mathbf{p}_0)$ is called a *solution parametrization* of (1) if A is in the image $\mathrm{Im}(\varDelta)$. Similarly, a place in $\mathrm{Im}(\delta)$ is called a *(Puiseux) solution place.*

It can be shown that for solution parametrizations $(a(t), b(t)) \in \mathrm{IFP}(\mathbf{p}_0)$, corresponding to a solution with ramification index n, it holds that

$$n(1-h) = \mathrm{ord}_t(a(t) - y_0) - \mathrm{ord}_t(b(t)). \tag{4}$$

This condition is invariant for the representative of a place. In particular, all Puiseux series solutions in the same solution place have the same ramification order. It turns out that condition (4) is already sufficient for solution places at $\mathbf{p}_0$ with $y_0 \in \overline{\mathbb{K}}$. Let us highlight this statement (see Theorem 10 in [5]):

Theorem 1. *Let $\mathcal{P} = [(a(t), b(t))] \in \text{Places}(\mathbf{p}_0)$ and $h = 0$. Then $\mathcal{P}$ is a solution place if and only if Eq. (4) holds for an $n \in \mathbb{Z}_{>0}$. In the affirmative case the ramification order of $\mathcal{P}$ is equal to n.*

Also the solutions with $h = 2$ can be computed algorithmically. For this purpose let us give in the following some insight into to proof of Theorem 1.

Let $\mathbb{L}$ be a subfield of $\overline{\mathbb{K}}$. For a given parametrization $(a(t), b(t))) \in \mathbb{L}((t))^2$ satisfying (4), our strategy is to find $s(t) \in \overline{\mathbb{K}}[[t]]$ with $\text{ord}_t(s(t)) = 1$ such that $(a(s(t)), b(s(t)))$ satisfies the *associated differential equation*

$$a'(s(t)) \cdot s'(t) = n(1-h)\, t^{n(1-h)-1}\, b(s(t)). \tag{5}$$

Let $k = \text{ord}_t(a(t) - y_0), r = \text{ord}_t(b(t))$ and $n(1-h) = k - r > 0$. By transforming (5) into an equation of Briot-Bouquet type [3], the solutions $s(t) = \sum_{i=1}^{\infty} \sigma_i\, t^i$ fulfill the following items.

1. If $h = 0$, there are exactly n solutions where $\sigma_1^n \in \mathbb{L}$ and $\sigma_i \in \mathbb{L}(\sigma_1)$ are uniquely determined for $i > 1$.
2. If $h = 2$, there is no solution or up to n one-parameter families of solutions with $\sigma_1^n \in \mathbb{L}$, $\sigma_{r-k} \in \mathbb{L}$ is a free parameter; $\sigma_2, \ldots, \sigma_{r-k-1} \in \mathbb{L}(\sigma_1)$ and for $i > r - k$ the coefficients $\sigma_i \in \mathbb{L}(\sigma_1, \sigma_{r-k})$ are uniquely determined.

After computing the solutions $s(t)$ of the associated differential equation, we obtain the solutions of the original differential equation by $a(s(x^{1/n}))$.

Solution Truncations. Since we cannot compute all coefficients of the Puiseux series solution, we have to truncate at some point. A *determined solution truncation* of (2) is an element of $\mathbb{L}[x^{1/n}][x^{-1}]$, for some $n \in \mathbb{Z}_{>0}$, that can be extended uniquely to a formal Puiseux series solution in $\mathbb{L}((x^{1/n}))$ or $\mathbb{L}((x^{-1/n}))$, respectively.

A point $\mathbf{p}_0 = (y_0, p_0) \in \mathscr{C}(F)$ is called a *critical curve point* if $p_0 \in \{0, \infty\}$ or $\frac{\partial F}{\partial p}(\mathbf{p}_0) = 0$ or $y_0 = \infty$. Under our assumptions, the set of critical curve points is finite. The only formal Puiseux series solution with non-critical $\mathbf{p}_0$ as initial tuple is a formal power series and its determined solution truncation is given by $y_0 + p_0 x$.

Assume that $\mathbf{p}_0 \in \mathscr{C}(F)$ is a critical curve point. Then, by the properties of the solutions of the associated differential equations, the bound N as in (3) also holds for the computation of the determined solution truncations. In particular, Eq. (4) can be checked, no further extensions of the ground field for computing the coefficients are necessary and the ramification index is determined.

Algorithm 1. PuiseuxSolve

```
Input: A first-order AODE $F(y, y') = 0$, where $F \in \mathbb{K}[y, p]$ is square-free with no factor in $\mathbb{K}[y]$ or $\mathbb{K}[p]$.
Output: A set consisting of all determined solution truncations of $F(y, y') = 0$ (expanded around zero and around infinity).
1: If $(\infty, \infty) \in \mathscr{C}(F)$, then perform the transformation $\tilde{y} = 1/y$ and apply the following steps additionally to the numerator of $F(1/y, -p/y^2)$ and $\mathbf{p}_0 = (0, 0)$ in order to obtain the solutions of negative order.
2: Compute the set of critical curve points $\mathcal{B}(F)$ (for $y_0 \in \overline{\mathbb{K}}$) and $\mathbb{V}(F(y, 0))$ (for $y_0 = \infty$).
3: For every point $(y_0, p_0) \in \mathscr{C}(F) \setminus \mathcal{B}(F), y_0 \neq \infty$, a determined solution truncation is $y_0 + p_0 x$.
4: Add to the output the constant solutions $y(x) = y_0$ corresponding to $(y_0, 0) \in \mathscr{C}(F), y_0 \neq \infty$.
5: For every place centered at a critical curve point $\mathbf{p}_0 = (y_0, p_0) \in \mathcal{B}(F)$, compute the first $N$ terms of a formal parametrization $(a(t), b(t))$.
6: For solutions expanded around zero (resp. around infinity), check equation (4) with $h = 0$ (resp. $h = 2$).
7: In the negative case, $[(a(t), b(t)]$ is not a solution place. In the affirmative case, compute the first $N$ terms of the solutions $s(t)$ of (5).
8: If $h = 0$ and $n > 0$, there exist exactly $n$ solutions. If $h = 2$ and $n > 0$, the associated differential equation is either unsolvable or contains a free parameter.
9: The first $N$ terms of $a(s(x^{1/n}))$ are the solution truncations with $\mathbf{p}_0$ as initial tuple.
```

For solutions expanded around zero we are able to ensure uniqueness of the extension of the truncated Puiseux series solutions (see also [5] [Theorem 14]). In the case where the expansion point is infinity, some truncations may coincide for some specific values of the free parameter coming from the solution of the associated differential equation.

2.2 Algebraic Solutions

In this section we consider a subclass of formal Puiseux series, namely algebraic series. These are $y(x) \in \overline{\mathbb{K}}((x))^*$ such that there exists a non-zero $G \in \overline{\mathbb{K}}[x, y]$ with $G(x, y(x)) = 0$. Since the field of formal Puiseux series is algebraically closed, all algebraic solutions can be represented as (formal) Puiseux series.

In [2] a bound on the degree of algebraic general solutions is given. There the authors indicate how to use these results in order to compute all algebraic solutions of such a given differential equation. A more detailed proof of this fact can be found in [8].

The first important observation is that if there exists one non-constant algebraic solution of (1), then all of them can be found easily by a shift in the minimal polynomial (see [8] [Theorem 4.1.22]).

Theorem 2. *Let $F \in \mathbb{K}[y, y']$ be irreducible and let $y(x)$ be a non-constant solution of $F = 0$ algebraic over $\overline{\mathbb{K}}(x)$ with minimal polynomial $G \in \overline{\mathbb{K}}[x, y]$.*

Then all formal Puiseux series solutions $\mathrm{Sol}_{\overline{\mathbb{K}((x))^*}}(F)$ *are algebraic and given by* $G(x+c,y)$*, where* $c \in \overline{\mathbb{K}}$*.*

The second important computational aspect is the degree bound on the solutions [2] [Theorem 3.4, Theorem 3.8]:

Theorem 3. *Let* $F \in \mathbb{K}[y,y']$ *be irreducible and let* $y(x)$ *be a non-constant solution of* $F = 0$ *algebraic over* $\overline{\mathbb{K}}(x)$ *with minimal polynomial* $G \in \overline{\mathbb{K}}[x,y]$*. Then*

$$\deg_x(G) = \deg_p(F), \quad \deg_y(G) \leq \deg_y(F) + \deg_p(F).$$

The third result is used to construct candidates of algebraic solutions:

Lemma 1. *Let* $G(x,y) \in \mathbb{K}[x,y]$ *be an irreducible polynomial with* $d_x = \deg_x G, d_y = \deg_y G$*. Let* $y(x)$ *be a Puiseux series solution of* $G(x,y) = 0$ *expanded at* $x = 0$ *with* $\mathrm{ord}_x(y(x)) = \nu$*. Let* $\nu' = \min\{\nu, 0\}$ *and write* $y(x) = \bar{y}(x) + \varphi(x)$ *with* $\mathrm{ord}_x(\varphi(x)) > N > 0$ *where*

$$N \geq 2\, d_x\, d_y - 2\, \nu'\, (d_y - 1). \tag{6}$$

Assume that $A(x,y) \in \mathbb{K}[x,y], \deg_x A \leq d_x$*,* $\deg_y A \leq d_y$ *is minimal with respect to the lexicographical order* $y > x$ *such that*

$$\mathrm{ord}_x(A(x,\bar{y}(x)) > 2\, d_x\, d_y - \nu'(d_y - 1), \tag{7}$$

holds. Then $A(x,y)$ *is, up to a constant factor, equal to* $G(x,y)$*.*

Proof. Let $R(x)$ be the resultant of $G(x,y)$ and $A(x,y)$ with respect to y. It is well known that there exist polynomials $B(x,y)$, $C(x,y)$ with $\deg_y B < d_y$, $\deg_y C < d_y$ such that

$$G(x,y)\, B(x,y) + A(x,y)\, C(x,y) = R(x).$$

Evaluating at $\bar{y}(x)$ we obtain

$$G(x,\bar{y}(x))\, B(x,\bar{y}(x)) + A(x,\bar{y}(x))\, C(x,\bar{y}(x)) = R(x). \tag{8}$$

Since $\nu' \leq 0$, it follows that $\mathrm{ord}_x\, C(x,\bar{y}(x)) \geq \nu' \deg_y C \geq \nu'(d_y - 1)$ and similarly for $B(x,\bar{y}(x))$. Hence, by (7), we have that

$$\mathrm{ord}_x(A(x,\bar{y}(x))\, C(x,\bar{y}(x))) > 2\, d_x\, d_y.$$

Let us proof that $\mathrm{ord}_x(G(x,\bar{y}(x)) > \nu'(d_y - 1) + N$. Taking the Taylor series of $G(x,\bar{y}(x) + \varphi(x))$ and because $G(x,\bar{y}(x) + \varphi(x)) = 0$, we have:

$$G(x,\bar{y}(x)) = -\sum_{j=1}^{d_y} \frac{1}{j!} \frac{\partial^j G}{\partial y^j}(x,\bar{y}(x))\, \varphi(x)^j.$$

The order in x of each term on the right hand side of above equation is greater than $\nu'\,(d_y-1)+N$, so it is for the left hand side. Now, because of (6), we have that

$$\mathrm{ord}_x(G(x,\bar{y}(x))\,B(x,\bar{y}(x))) > \nu'(d_y-1)+N+\nu'(d_y-1) \geq 2\,d_x\,d_y.$$

Hence, the left hand side of (8) has order greater than $2\,d_x\,d_y$ and the right hand side is a polynomial of degree less than or equal to $2\,d_x\,d_y$. Hence, $R(x)=0$, and therefore, $G(x,y)$ and $A(x,y)$ have a common factor. Since $G(x,y)$ is an irreducible polynomial, it is a factor of $A(x,y)$. Then, by the degree conditions on $A(x,y)$, the statement follows. □

In [2] the method of detecting candidates $G(x,y)$ for algebraic solutions of the differential equations $F(y,y')=0$ consists by computing $\bar{y}(x)$, the first N terms of a power series solution $y(x)$ of the differential equations $F(y,y')=0$, with a regular curve point of $\mathscr{C}(F)$ as initial tuple. Hence, in this case the solution $y(x)$ is of order 0 and $\nu'=0$. Choose $N>d_x\,d_y$ and construct, by solving a linear system of equations, a polynomial A fulfilling the properties (6) and (7). This approach reduced the number of formal power series solutions that we can use to construct a candidate. Lemma 1 allows to choose any Puiseux series solutions of the differential equations and reduce the computational cost.

Once a candidate $A(x,y)$ is detected, we can check whether it is an actual algebraic solution of the differential equation $F(y,y')=0$ by checking whether the differential pseudo remainder of $F(y,y')$ with respect $A(x,y)$ is zero. These results lead to the following algorithm.

Algorithm 2. AlgebraicSolve

```
Input: A first-order AODE $F(y,y')=0$, where $F\in\mathbb{K}[y,p]$ is irreducible over $\overline{\mathbb{K}}(y)$.
Output: The minimal polynomial of an algebraic solution of $F(y,y')=0$, describing
    all solutions, if it exists.
 1: Compute the minimal number of terms of all Puiseux solutions of $F(y,y')=0$
    using PuiseuxSolve and choose one of them, denote it by $\hat{y}(x)$. Let $\nu$ be its order,
    $\nu'=\min(\nu,0)$ and $n$ its ramification index.
 2: Let $d_x=\deg_p F$ and $d_y=\deg_y F+\deg_p F$.
 3: Compute the prolongation $\bar{y}(x)$ of $\hat{y}(x)$ up to order $N=2\,d_x\,d_y-2\,\nu'\,(d_y-1)+1/n$.
 4: Compute $A(x,y)\in\overline{\mathbb{K}}[x,y]$ fulfilling the required conditions from Lemma 1 by an
    ansatz of unknown coefficients and solving the resulting linear system.
 5: Check whether $\mathrm{prem}(F,A)=0$. If so, then $A(x,y)$ is an actual solution. Otherwise
    there exists no algebraic solution.
```

3 The Package FirstOrderSolve

In this section, we present the structure and content of the `MAPLE` package `FirstOrderSolve`. It consists several procedures that implement in particular the algorithms PuiseuxSolve and AlgebraicSolve described above. This package computes the Puiseux series solutions and algebraic solutions of first order autonomous AODEs with coefficients in an algebraic extension field of $\mathbb{Q}$.

3.1 Overview of the Software Structure

The created `MAPLE` package is initialized by the command

```
> with(FirstOrderSolve):
```

The main procedures are

- `SolutionTruncations`: for computing all formal Puiseux series solutions (Algorithm PuiseuxSolve);
- `AlgebraicSolution`: for computing the minimal polynomial of the algebraic solutions (Algorithm AlgebraicSolve);
- `GenericSolutionTruncation`: for computing a truncation of the solutions with non-critical initial tuple;
- `ProlongSolutionTruncation`: for prolonging the solution truncations up to a higher degree.

These four commands are public to the user. The package is divided into several sub-packages `BriotBouquetSolve, LocalSolve, AlgebraicSolve`, which are not accessible for the user, and uses the hierarchy sketched below (Fig. 1).

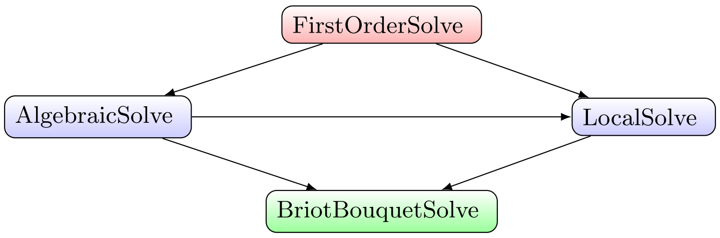

Fig. 1. The hierarchy of the package.

The main commands in the sub-packages are the following.

- `ParametrizationSetAlgCurve`: for computing truncations of the formal parametrizations of an implicitly defined algebraic curve by using the command `algcurves:-puiseux`;
- `ReparametrizationSet`: for computing the solutions of the associated differential equation by using `BriotBouquetSolve`;
- `BriotBouquetSolve`: for computing the unique solution of a first-order differential equation in quasi-solved form (which is called an equation of Briot-Bouquet type [3] [Section 80,86]); this procedure is using a Newton type algorithm for solving the resulting linear system in several variables;

In the following, we give a description of the procedures in the package `FirstOrderSolve`, available at https://risc.jku.at/sw/firstordersolve/. There can be found more detailed information on the commands in the included help.

3.2 Description of the Software Components

`> SolutionTruncations`

Computes all Puiseux series solutions of a given first order autonomous ordinary differential equation.

Since the equation is autonomous, the translation of the independent variable by any constant in a solution is again a solution. Hence, the only relevant expansion points are 0 and infinity.

The solutions expanded at 0 can be split into two sets: a generic solution and a set of particular solutions. The generic solution consists of all solutions starting with a non-critical curve point and is addressed in `GenericSolutionTruncation`. Each critical curve point corresponds to a set (that could be empty) of particular Puiseux series solutions.

The command computes the generic solution, all particular solutions expanded at 0 and all solutions expanded at infinity. The solutions are represented as truncations such that existence and uniqueness is ensured. In other words, the truncations are in one-to-one correspondence to the solutions. By setting the optional arguments genericsolution, const, computeFinite, computeInf to false, the corresponding subsets of the solution set can be suppressed. The remaining option iv $= y_0$ represents an initial condition of the format $y(0) = y_0$, where y_0 is an element of the ground field or an algebraic extension field of it, which is additionally taken into account.

- ◇ Calling Sequence: `> SolutionTruncations(F, N, options)`
- ◇ Input: a polynomial F in y, y', a rational number N (by default set to zero) and several optional arguments: genericsolution, const, computeFinite, computeInf (all boolean) and a constant iv.
- ◇ Output: a list consisting of three components: the generic solutions, the solutions expanded at 0 and the solutions expanded at infinity represented as truncated Puiseux series (modulo x^N).

`> GenericSolutionTruncation`

The first order differential equation has a generic local solution $y(x) = y_0 + y_1 x + \mathcal{O}(x^2)$, where $F(y_0, y_1) = 0$. If F is irreducible as polynomial and (y_0, y_1) is a regular affine point of the curve implicitly defined by F, the extension of $y_0 + y_1 x$ to a solution $y(x)$ is guaranteed and unique. The command `GenericSolutionTruncation` computes the first terms of the generic (formal) power series solutions, expanded around 0, of the given differential equation.

Note that for every irreducible component one generic solution is computed. Thus, all generic solutions of $F = 0$ are given by the union of the generic solutions of the components. If the given differential equation is known to be irreducible, the optional argument irreducible = true (see below) can be used in order to speed up computations.

The output of the command is a set of lists with two entries: a polynomial in x representing the solution computed modulo x^N involving an unspecified parameter $\_CC$ and a set of exceptional values for $\_CC$. For these values the generic solution would in general not lead to a solution of the given differential equation or might involve fractional exponents. Finally, let us mention that, if the precision of the output is not high enough, it is possible to use the command `ProlongSolutionTruncation`; see below.

- ◇ Calling Sequence: `> GenericSolutionTruncation(F, N, options)`
- ◇ Input: a polynomial F in y, y', a rational number N (by default set to zero), and optionally `irreducible` as boolean.
- ◇ Output: is a set of lists with two entries: a polynomial in x representing the solution computed modulo x^N involving an unspecified parameter $\_CC$ and a set of exceptional values for $\_CC$.

`> ProlongSolutionTruncation`
For the given first order differential equation, if an appropriate change of variables $z(x) = y(x) + s(x)$ is performed, the resulting equation

$$G(x, z(x), z'(x)) = F(y(x) + s(x), y'(x) + s'(x)) = 0$$

might be of Briot-Bouquet type. In case that $s(x)$ is such a solution truncation of $y(x)$, existence and uniqueness of the solution $z(x)$ of G are ensured and the following coefficients can be found by a Newton type algorithm. In particular, this is the case when $s(x)$ is an output element of `GenericSolutionTruncation` or `SolutionsTruncations`.

In this situation, the command `ProlongSolutionTruncation` prolongs the first terms of a truncated Puiseux series solution $s(x)$ of $F(y(x), y'(x)) = 0$.

- ◇ Calling Sequence: `> ProlongSolutionTruncation(F, s, N, x0)`
- ◇ Input: a polynomial F in y, y', a polynomial s, a rational number N and $x0$ equals 0 or infinity (by default set to zero)
- ◇ Output: it is again a truncated Puiseux series computed until the order x^N (or $1/x^N$)

`> AlgebraicSolution`
Algebraic solutions of the first order autonomous differential equation are represented by its minimal polynomisl, say $G(x, y)$. In this case, all the functions $y(x)$ with $G(x, y(x)) = 0$ are solutions of this differential equation and can be represented as Puiseux series.

Assuming that F is an irreducible polynomial, the existence of algebraic solutions can be decided and, in the affirmative case, all solutions are algebraic and are given as shift of the independent variable, namely by $G(x+c, y)$. Therefore, by factorizing the given differential equation, all algebraic solutions can be found using this procedure for every component.

The command `AlgebraicSolution` decides the existence of algebraic solutions of the given differential equation. Furthermore, if a solution exists the output is the minimal polynomial of the solution. The other solutions then can be easily found by shifting x. The solutions are found by checking whether a particular solution is algebraic. Efficiency of the algorithm highly depends on the chosen initial value. The procedure is using a formal power series solution, which means non-negative integer exponents for the solution, with a relatively small number of algebraic extensions of the ground field. Similarly to the command `GenericSolutionTruncation`, if the given differential equations is known to be irreducible, this can be specified by the optional argument irreducible = true.

- ◇ Calling Sequence: `> AlgebraicSolution(F, options)`
- ◇ Input: F is a first-order differential polynomial, and `irreducible` is a boolean option.
- ◇ Output: the decision on the existence of algebraic solutions of the differential equation. If a solution exists the output is the minimal polynomial of the solution.

3.3 Usage of the Package

In order to use the package, download the file FirstOrderSolve.m from https://risc.jku.at/sw/firstordersolve/ and save it as your local folder. After starting Maple you redefine the variable libname as

```
> libname:=libname, 'path of user local folder';
```

Then, after executing the command

```
> with(FirstOrderSolve);
```

The package can be used. In the appendix we provide a Maple Worksheet illustrating the usage of our package for the computation of all Puiseux series solutions of first-order autonomous ordinary differential equations. We provide at https://risc.jku.at/sw/firstordersolve/ an extended version of this file.

```
> with(FirstOrderSolve);
```

$$[AlgebraicSolution, GenericSolutionTruncation, ProlongSolutionTruncation, SolutionTruncations] \tag{1}$$

A list of possible examples is the following.

```
> Examples := [diff(y(x), x) * y(x)^2 + y(x) - 1,
        diff(y(x), x) - y(x)^4 - y(x)^2,
        diff(y(x), x) - y(x)^3 - y(x)^2,
        diff(y(x), x)^3 + y(x)^2 - diff(y(x), x),
        diff(y(x), x)^2-4 * y(x)^3,
        diff(y(x), x)^3 * (y(x)^6 + 2 * y(x) + 1) - (12 * y(x)^5 + 9 * y(x)^4 - 1) * diff(y(x),
      x)^2 + 27 * y(x)^8 + 54 * y(x)^7 + 27 * y(x)^6 + 4 * y(x)^3
   ]:
```

We start with the first entry of the list and compute all solutions and use the option of minimal output length by not specifying the truncation order.

```
> F1 := Examples[1];
```

$$F1 := \left(\frac{d}{dx}\, y(x)\right) y(x)^2 + y(x) - 1 \tag{2}$$

```
> sol1 := SolutionTruncations(F1);
```

$$sol1 := \left[\left\{\left[\_CC - \frac{(-1 + \_CC)\, x}{\_CC^2}, \{0\}\right]\right\}, \{1, RootOf(\_Z^3 - 3)\, x^{1/3}\}, \varnothing\right] \tag{3}$$

Let us prolong the non-constant solution.

```
> ProlongSolutionTruncation(F1, sol1[2, 2], 7/3);
```

$$RootOf(\_Z^3 - 3)\, x^{1/3} - \frac{RootOf(\_Z^3 - 3)^2\, x^{2/3}}{4} - \frac{3\, x}{80} + \frac{RootOf(\_Z^3 - 3)\, x^{4/3}}{320} + \frac{67\, RootOf(\_Z^3 - 3)^2\, x^{5/3}}{22400} + \frac{603\, x^2}{179200} + \frac{163\, RootOf(\_Z^3 - 3)\, x^{7/3}}{179200} \tag{4}$$

The generic solution can be either prolonged by the same command or by GenericSolutionTruncation itself. The exceptional value is _CC=0.

```
> GenericSolutionTruncation(F1, 3);
```

$$\left\{\left[\_CC - \frac{(-1 + \_CC)\, x}{\_CC^2} - \frac{(\_CC^2 - 3\,\_CC + 2)\, x^2}{2\,\_CC^5}, \{0\}\right]\right\} \tag{5}$$

For the second example we prolong the solution expanded around infinity.

```
> F2 := Examples[2];
```

$$F2 := \frac{d}{dx}\, y(x) - y(x)^4 - y(x)^2 \tag{6}$$

```
> sol2 := SolutionTruncations(F2);
```

$$sol2 := \left[\{[\_CC + (\_CC^4 + \_CC^2)\, x, \varnothing]\}, \left\{0, -I, I, -\frac{RootOf(\_Z^3 + 3)^2}{3\, x^{1/3}}\right\}, \left\{-\frac{1}{x} + \frac{\_CC}{x^2}\right\}\right] \tag{7}$$

```
> ProlongSolutionTruncation(F2, sol2[3, 1], 4, infinity);
```

$$\frac{\_CC}{x^2} - \frac{1}{x} + \frac{-\_CC^2 - 1}{x^3} + \frac{\_CC^3 + 3\,\_CC}{x^4} \qquad \textbf{(8)}$$

The standard procedure of Maple cannot find all of the solutions, since for example the series expansion of the Puiseux series is not covered.

```
> dsolve({F2=0}, y(x));
```

$$\{y(x) = \tan(\mathit{RootOf}(\_C1\tan(\_Z) + \_Z\tan(\_Z) + x\tan(\_Z) + 1))\} \qquad \textbf{(9)}$$

Solutions at infinity with fractional exponents are found in the next example. The generic solution, constant solutions and the solutions expanded around infinity are suppressed and the precision is set to N=2.

```
> SolutionTruncations(Examples[3], 2, computeInf=false, genericsolution=false, const=false);
```

$$\left[\varnothing, \left\{-\frac{1}{3} - \frac{\mathit{RootOf}(\_Z^2+2)}{2\sqrt{x}} + \frac{\mathit{RootOf}(\_Z^2+2)\sqrt{x}}{12} + \frac{4x}{135} - \frac{\mathit{RootOf}(\_Z^2+2)\,x^{3/2}}{432}\right\}, \varnothing\right] \qquad \textbf{(10)}$$

Alternatively one may directly specify the initial value by iv=infinity.

```
> SolutionTruncations(Examples[3], 2, iv=infinity);
```

$$\left\{-\frac{1}{3} - \frac{\mathit{RootOf}(\_Z^2+2)}{2\sqrt{x}} + \frac{\mathit{RootOf}(\_Z^2+2)\sqrt{x}}{12} + \frac{4x}{135} - \frac{\mathit{RootOf}(\_Z^2+2)\,x^{3/2}}{432}\right\} \qquad \textbf{(11)}$$

In the next example we compute the solutions with the initial value y(0)=0. The non-constant solutions are not detected by 'dsolve'.

```
> SolutionTruncations(Examples[4], iv=0);
```

$$\{0, x, -x\} \qquad \textbf{(12)}$$

```
> dsolve({Examples[4]=0, y(0)=0}, y(x));
```

$$y(x) = 0 \qquad \textbf{(13)}$$

For obtaining algebraic solutions we run the corresponding command over all examples. We see that not all of them have algebraic solutions.

```
> map(a→AlgebraicSolution(a), Examples);
```

$$[\varnothing, \varnothing, \varnothing, \varnothing, \{x^2y - 2xy + y - 1\}, \{yx^3 + y^2 + x + y\}] \qquad \textbf{(14)}$$

For the last two differential equations we obtain other results by using 'dsolve'. In the latter example the output is very complicated and lengthy whereas the algebraic solution obtained by our code is very compact.

```
> dsolve(Examples[5]=0);
```

$$y(x) = \mathrm{WeierstrassP}(x + \_C1, 0, 0) \qquad \textbf{(15)}$$

```
> dsolve(Examples[6]=0) :
```

Acknowledgements. The first author was supported by the bilateral project ANR-17-CE40-0036 and DFG-391322026 SYMBIONT. The second author was partially supported by Agencia Estatal de Investigación PID2019-105621GB-I00. The third and fourth authors were partially supported by FEDER/Ministerio de Ciencia, Innovación y Universidades Agencia Estatal de Investigación/MTM2017-88796-P (Symbolic Computation: new challenges in Algebra and Geometry together with its applications). The third author was also supported by the Austrian Science Fund (FWF): P 31327-N32. The fourth author is member of the Research Group ASYNACS (Ref. CT-CE2019/683).

References

1. Aroca, J.: Puiseux Solutions of Singular Differential Equations, pp. 129–145. Birkhäuser Basel, Basel (2000)
2. Aroca, J., Cano, J., Feng, R., Gao, X.S.: Algebraic general solutions of algebraic ordinary differential equations. In: Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, pp. 29–36. ACM (2005)
3. Briot, C., Bouquet, J.: Recherches sur les proprietés des équations différentielles. J. de l'Ecole Polytechnique **21**(36), 133–198 (1856)
4. Cano, J.: The newton polygon method for differential equations. In: Li, H., Olver, P.J., Sommer, G. (eds.) GIAE/IWMM -2004. LNCS, vol. 3519, pp. 18–30. Springer, Heidelberg (2005). https://doi.org/10.1007/11499251_3
5. Cano, J., Falkensteiner, S., Sendra, J.R.: Existence and convergence of Puiseux series solutions for first order autonomous differential equations. J. Symb. Comput. (2020). https://doi.org/10.1016/j.jsc.2020.06.010
6. Della Dora, J., Richard-Jung, F.: About the newton algorithm for non-linear ordinary differential equations. In: Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, ISSAC 1997, pp. 298–304. ACM, New York (1997). https://doi.org/10.1145/258726.258817
7. Duval, D.: Rational Puiseux expansion. Compositio Mathematica **70**(2), 119–154 (1989)
8. Falkensteiner, S.: Power series solutions of AODEs - existence, uniqueness, convergence and computation. Ph.D. thesis, RISC Hagenberg, Johannes Kepler University Linz (2020)
9. Feng, R., Gao, X.S.: Rational general solutions of algebraic ordinary differential equations. In: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation, pp. 155–162. ACM (2004)
10. Feng, R., Gao, X.S.: A polynomial time algorithm for finding rational general solutions of first order autonomous ODEs. J. Symb. Comput. **41**(7), 739–762 (2006)
11. Fine, H.: On the functions defined by differential equations, with an extension of the Puiseux polygon construction to these equations. Am. J. Math. **11**, 317–328 (1889). https://doi.org/10.2307/2369347
12. Fine, H.: Singular solutions of ordinary differential equations. Am. J. Math. **12**, 295–322 (1890). https://doi.org/10.2307/2369621
13. Grigoriev, D., Singer, M.: Solving ordinary differential equations in terms of series with real exponents. Trans. A.M.S. **327**, 329–351 (1991). https://doi.org/10.2307/2001845
14. Stadelmeyer, P.: On the computational complexity of resolving curve singularities and related problems. Ph.D. thesis, RISC, Johannes Kepler University Linz (2000)
15. Vo, N., Grasegger, G., Winkler, F.: Deciding the existence of rational general solutions for first-order algebraic ODEs. J. Symb. Comput. **87**, 127–139 (2018)
16. Walker, R.: Algebraic Curves. Princeton University Press, Princeton (1950)