Truncated and Infinite Power Series in the Role of Coefficients of Linear Ordinary Differential Equations*

S.A. Abramov[0000−0001−7745−5132], D.E. Khmelnov[0000−0002−4602−2382], and A.A. Ryabenko[0000−0001−5780−7743]

Dorodnicyn Computing Centre,
Federal Research Center "Computer Science and Control"
of the Russian Academy of Sciences,
Vavilova, 40, Moscow 119333, Russia
sergeyabramov@mail.ru, dennis_khmelnov@mail.ru, anna.ryabenko@gmail.com

Abstract. We consider linear ordinary differential equations, each of the coefficients of which is either an algorithmically represented power series, or a truncated power series. We discuss the question of what can be learned from equations given in this way about its Laurent solutions, i.e. solutions belonging to the field of formal Laurent series. We are interested in the information about these solutions, that is invariant with respect to possible prolongations of the truncated series which are the coefficients of the given equation.

Keywords: Differential equations · Truncated power series · Algorithmically represented infinite power series · Laurent series · Computer algebra systems

1 Introduction

We will consider operators and differential equations written using the operation $\theta = x\frac{d}{dx}$. In the original operator

$$L = \sum_{i=0}^{r} a_i(x)\theta^{\,i}, \tag{1}$$

as well as in the equation $L(y) = 0$, for each $a_i(x)$, $i = 0, 1, \ldots, r$, one of two possibilities is allowed: $a_i(x)$ can be

- an infinite series represented algorithmically: the series $\sum a_n x^n$ is defined by an algorithm computing a_n by n,
  or

* Supported in part by RFBR grant, project 19-01-00032.

– a truncated series
$$a_i(x) = \sum_{j=0}^{t_i} a_{ij} x^j + O(x^{t_i+1}) \tag{2}$$
where t_i is an integer such that $t_i \geq -1$ (if $t_i = -1$ then the sum in (2) is 0). We call t_i *the truncation degree* of the coefficient $a_i(x)$ represented in the form (2). Note that a coefficient in (1) can be of the form $O(x^m)$, $m \geq 0$.

We assume that at least one of the constant terms of $a_0(x), \ldots, a_r(x)$ is nonzero.

The coefficients of series belong to a field K of characteristics 0. The following notations are standard:

$K[x]$, $K[x, x^{-1}]$ are the rings of polynomials and, resp., Laurent polynomials with coefficients from K;

$K[[x]]$ is the ring of formal power series with coefficients from K;

$K((x))$ is the quotient field of the ring $K[[x]]$; the elements of this field are formal Laurent series with coefficients from K.

Definition 1. *The degree* $\deg f(x)$ *of a polynomial* $f(x)$ *from* $K[x]$ *or* $K[x, x^{-1}]$*, is defined as the largest degree of* x *belonging to* $f(x)$ *(conventionally,* $\deg 0 = -\infty$*). Note that the degree of a Laurent polynomial is in some cases non-positive, even when this polynomial is not a constant:* $\deg(2x^{-2} + x^{-1}) = -1$*,* $\deg(3x^{-1} + 1) = 0$*, etc.*

The solutions we are interested in, belong to the field of formal Laurent series with coefficients from K. We will call such solutions as *Laurent*. A more exact specification for the problem of finding such solutions will be given later in this introductory section. We will not discuss the questions of convergence of series.

A discussion of the algorithmic aspect of problems of this kind involves considering the question of representing infinite series, in particular the series which play the role of the coefficients of the equation.

In [1–3], an algorithmic representation was considered. It was detected that some problems associated with the solutions of equations given in this way turn out to be algorithmically unsolvable, though, at the same time, the other part is successfully solvable. For example, the problem of finding Laurent solutions is solvable: these solutions can be represented algorithmically in the same sense as the representation of the coefficients of the equation. (In the mentioned papers, not only scalar equations were discussed, but also systems of equations.)

In [3, 8], the authors considered the problems of constructing solutions under the assumption that all series playing the role of the coefficients of a given equation or system are represented in the truncated form. In [8] it was found out which truncation of the system coefficients will be sufficient to calculate a given number of initial terms of the series, included in the exponentially-logarithmic solutions of the system. In [5, 6] we considered this problem as the task of constructing truncated solutions; it was shown how to construct the maximum possible number of initial terms of the series included in the Laurent and regular solutions of the equation.

In this paper, we admit the presence in the original equation of such coefficients that are of two different kinds indicated below formula (1). We are interested in information on Laurent solutions that is invariant with respect to possible prolongations of the truncated series representing the coefficients of the equation. But everything is not so simple already with the definition of the concept of "solution" for an equation which, in the presence of truncated coefficient, is, in fact, not completely specified. We introduce the concept "truncated Laurent solution".

Definition 2. *Let an operator L have the form (1) and*

$$L(y) = 0 \tag{3}$$

be the equation corresponding to this operator. An expression

$$f(x) + O(x^{k+1}), \tag{4}$$

in which $f(x) \in K[x^{-1}, x] \setminus \{0\}$ and k is an integer greater than or equal to $\deg f(x)$*, is called a* truncated Laurent solution *of (3), if for any specification of all $O(x^{t_i+1})$ (that is, for any replacement of the symbols $O(...)$ by concrete series having corresponding valuation) included in the coefficients $a_i(x)$ of equation (3), such a specification of the series $O(x^{k+1})$ in (4) is possible that the specified expression (4) becomes a Laurent solution to the specified equation. This k in (4) is* the truncation degree of the solution*.*

We propose an algorithm that allows for an equation $L(y) = 0$ represented in the explained form and an integer k to construct all such truncated Laurent solutions of this equation that have a truncation degree not exceeding k. If the equation does not have such truncated Laurent solutions then the result of the algorithm will indicate this.

The algorithm is described in Sect. 5. But first it is shown in Sect. 2 that both checking the finiteness of the set of those k for which the formulated problem has a solution, and finding the maximum possible value of k if it exists, are algorithmically undecidable problems. Our algorithm works with a specific given k.

Sect. 6 describes the implementation of the algorithm in Maple [9].

2 The Equation Threshold

Definition 3. *Let L be of the form (1). Consider the set N of all integers n such that the equation $L(y) = 0$ has a truncated Laurent solution whose truncation degree is n. Let N be nonempty and have the maximal element. We will call this element the* threshold *of the equation $L(y) = 0$. If the set N contains arbitrarily large integers, then we say that the threshold of the equation is ∞. If this set is empty, then the threshold is conventionally $-\infty$.*

Remark 1. In Sect. 3 it will be, in particular, shown that if the set N considered in the previous definition is nonempty, then the subset of its negative elements is finite.

For demonstrating an example, we will need the notions of the series valuation and the prolongation of an equation.

Definition 4. *For a nonzero formal Laurent series* $a(x) = \sum a_i x^i \in K((x))$ *its* valuation *is defined as* $\operatorname{val} a(x) = \min\{i \mid a_i \neq 0\}$, *with* $\operatorname{val} 0 = \infty$. A prolongation *of the operator* L *of the form (1) is defined as an operator*

$$\tilde{L} = \sum_{i=0}^{r} b_i(x)\theta^i \in K[[x]][\theta]$$

such that $b_i(x) = a_i(x)$ *if all terms of* $a_i(x)$ *are known and*

$$b_i(x) - a_i(x) = O(x^{t_i+1}) \tag{5}$$

(i.e. $\operatorname{val}(b_i(x) - a_i(x)) > t_i$*) for truncated* $a_i(x)$, $i = 0, 1, \ldots, r$.

Example 1. Consider the equation

$$(1 + O(x))\theta y + a_0(x)y = 0, \tag{6}$$

where

$$a_0(x) = \sum_{j=k}^{\infty} a_{0j} x^j,$$

Set $k = \operatorname{val} a_0(x)$. For $k = 0$, i.e. for $a_{00} \neq 0$, truncated Laurent solutions exist only when a_{00} is an integer. We will consider the case

$$k = \operatorname{val} a_0(x) \geq 1. \tag{7}$$

Here, any prolongation of equation (6) has Laurent solutions. If the leading coefficient of this equation is $1 + \sum_{j=1}^{\infty} a_{1j} x^j$ then each Laurent solution is of the form

$$C\left(1 - \frac{a_{0k}}{k}x^k + \frac{a_{0k}a_{11} - a_{0,k+1} + a_{01}^2}{k+1}x^{k+1} + O(x^{k+2})\right),$$

where C is an arbitrary constant.

Thus, for equations of the form (6) with $\operatorname{val} a_0(x) > 0$, the coefficient of x^{k+1} in all nonzero Laurent solutions depends on coefficients of prolongation of the original equation (6). Consequently, the number $k = \operatorname{val} a_0(x)$ is the threshold.

If $\operatorname{val} a_0(x) = \infty$, in other words, if $a_0(x) = 0$, then all extensions of the equation (6) will have Laurent solutions $y(x) = C$, and the threshold of the equation (6) is $k = \operatorname{val} a_0(x) = \infty$.

Proposition 1. *There exists no algorithm that, for an arbitrary equation $L(y) = 0$ with an operator L of the form (1), finds out whether its threshold is finite or infinite.*

Proof. It is known that there is no algorithm that allows for an arbitrary series represented algorithmically to check whether this series is zero — this fact follows from the fundamental results of A. Turing [10]. If there was an algorithm that allows us to solve the problem formulated in the condition of the present proposition, then applying it to the equation (6), in which the series $a_0(x)$ is represented algorithmically, would allow us to determine whether the series $a_0(x)$ is zero (first it is necessary to verify that the constant term of this series is zero: see the assumption made by us when considering the Example 1 (7); if this constant term is not equal to zero, then, of course, the series is non-zero). The impossibility of this algorithm follows.

Corollary 1. *There is no algorithm that allows for an arbitrary equation $L(y) = 0$ with the operator L of the form (1), to calculate the value (an integer or one of the symbols ∞, $-\infty$) of its threshold.*

Proposition 2. *Let $L(y) = 0$ be an equation with an operator L of the form (1) and $k \in \mathbb{Z}$. It can be tested algorithmically whether k exceeds the threshold of the equation or not; if the answer is positive then the threshold h of this equation can be found. In addition, all such truncated Laurent solutions whose truncation degree does not exceed h can be constructed.*

Proof. We check the existence of invariant initial segments of solutions up to x^k by actually constructing these segments. The construction is considered as successful, if the resulting coefficients for powers of x do not include unknown coefficients of the prolongation of the equation (this approach was used in [5] when considering equations with all coefficients represented by truncated series). If it was not possible to reach x^k then, first, it was established that the value of k exceeds the threshold of the equation, and, second, it is possible to find the threshold value and find all the invariant initial segments of the Laurent solutions.

Remark 2. Thus, if the considered k is such that for a given equation $L(y) = 0$ with the operator L having the form (1) there does not exist truncated Laurent solutions of truncation degree k, then this circumstance opens, in particular, the opportunity of finding the threshold of the original equation — a quantity which is by Corollary 1 of the Proposition 1 non-computable algorithmically, in case if one is based only on the original equation.

3 Induced Recurrence Equations

Let σ denote the shift operator such that $\sigma c_n = c_{n+1}$ for any sequence (c_n). The transformation

$$x \to \sigma^{-1}, \;\; \theta \to n$$

assigns to a differential equation

$$\sum_{i=0}^{r} a_i(x)\theta^{\,i} y(x) = 0, \tag{8}$$

where $a_i(x) \in K[[x]]$, the induced recurrent equation

$$u_0(n)c_n + u_{-1}(n)c_{n-1} + \dots = 0. \tag{9}$$

The equation (8) has a Laurent solution $y(x) = c_v x^v + c_{v+1}x^{v+1} + \dots$ if and only if the two-sided sequence $\dots, 0,\ 0,\ c_v,\ c_{v+1}, \dots$ satisfies the equation (9) (see [4]). In our assumption, for the given operator (1), some of whose coefficients are truncated series, at least one of the constant terms of $a_0(x), \dots, a_r(x)$ is not equal to zero. Thus

$$u_0(n) = \sum_{i=0}^{r} a_{i,0}\, n^i \tag{10}$$

is a non-zero polynomial which is independent of any prolongations of the given operator L. It can be considered as a version of the *indicial* polynomial of the given equation. The finite set of integer roots of this polynomial contains all possible valuations v of Laurent solutions of all prolongations of the equation $L(y) = 0$.

If the polynomial $u_0(n)$ has no integer roots, then no prolongation of $L(y) = 0$ has nonzero Laurent solutions. In this case, set the threshold of the equation $L(y) = 0$ to be $-\infty$.

Let $\alpha_1 < \dots < \alpha_s$ be all integer roots of the polynomial $u_0(n)$. Then, the set N from Definition 3 has no element which is less than α_1. All prolongations of the equation $L(y) = 0$ have Laurent solutions with valuation α_s (see, e.g., [5]). Thus, $\alpha_s \in N$, and as a consequence, the threshold is greater than or equal to α_s. The threshold is $-\infty$ if and only if the polynomial $u_0(n)$ has no integer root.

4 Computing Coefficients of Truncated Laurent Solutions

Computing elements of the sequence (c_n) of coefficients of Laurent solutions can be performed by successively increasing n by 1, starting with $n = \alpha_1$ which is the minimum integer root of the polynomial $u_0(n)$. Set $c_n = 0$ for $n < \alpha_1$. If $u_0(n) \neq 0$ for some integer n then (9) allows us to find c_n by $c_{n-1}, c_{n-2}, \dots$ Since $c_n = 0$ when $n < \alpha_1$, relation (9) has a finite number of non-zero terms. If $u_0(n) = 0$, we declare c_n an *unknown constant*. The previously calculated c_{n-1}, $c_{n-2}, \dots, c_{\alpha_1}$ satisfy the relation

$$u_{-1}(n)c_{n-1} + u_{-2}(n)c_{n-2} + \dots + u_{-n+\alpha_1}(n)c_{\alpha_1} = 0. \tag{11}$$

These relations allow to calculate the values of some previously introduced unknown constants. After the value of n exceeds the greatest integer root of $u_0(n)$, new unknown constants and relations of the form (11) will not occur any longer.

If L has truncated coefficients then it is possible that for some $n \geq \alpha_1$ the left hand side of (9) depends on those unspecified coefficients that are hidden in (1) in the symbols O (some of coefficients of L may be of the form (2)). These unspecified coefficients will be called *literals*. For $u_0(n) \neq 0$, the calculated value of c_n depends on literals. For $u_0(n) = 0$, if the relation (11) depends on literals then computing previously introduced unknown constants is postponed until n reaches α_s. When $n = \alpha_s$, we obtain

(a) the values of the coefficients $c_{\alpha_1}, c_{\alpha_1+1}, \ldots, c_{\alpha_s}$ (all of which depend on unknown constants, some of which may depend on literals as well);
(b) the set of unknown constants;
(c) the set of relations for unknown constants containing literals.

By the set (c) we can find values of unknown constants which are invariant to all prolongations of the given truncated equation (see [6] for details). The unknown constants, that did not get values we declare *arbitrary constants* involved into the Laurent solution of the differential equation.

5 Algorithm

Input data:

- a differential operator L of the form (1), whose each coefficient is either an algorithmically represented power series, or a truncated power series,
- an integer number k.

Output result:

- The answer to the question of the existence of truncated Laurent solutions for the equation $L(y) = 0$. If there are no such solutions, then the output is the empty list [ ].
- If the answer to the question is positive then the algorithm computes all the truncated Laurent solutions, whose truncation degrees do not exceed k; it is possible that some solutions are computed with bigger truncation degree (such solutions are found by the algorithm due to the general computation strategy). If the algorithm finds out that k exceeds the threshold of the equation $L(y) = 0$ then the algorithm computes the value h of the threshold (see Remark 2) and constructs all the truncated Laurent solutions, whose truncation degrees do not exceed h.

The steps:

1. By (10), compute $u_0(n)$. Find the set

$$\alpha_1 < \ldots < \alpha_s$$

   of all integer roots of $u_0(n)$. If the set is empty then there are no truncated Laurent solutions; stop the work with the result [ ].

2. $d := \alpha_s - \alpha_1$; compute the coefficients

$$u_{-j}(n) := \sum_{i=0}^{r} a_{i,j}\ (n-j)^i, \ \ j = 1, \ldots, \max\{d,\, k - \alpha_1\},$$

   of the induced recurrent equation (9).
3. Compute the coefficients c_n, $n = \alpha_1, \alpha_1 + 1, \ldots, \alpha_s$, of the truncated Laurent solution using (9) as it is described in Sect. 4.
4. If $k > \alpha_s$ then continue computing c_n using (9) with n subsequently increased by 1 while the both following conditions hold
   (a) $n \leq k$,
   (b) a non trivial set of the values of the arbitrary constants exists such that c_n is independent of the literals (it is detailed in [6, Sect. 4.1]).

   If (a) is true, but (b) is false for the current n then the threshold of the equation is computed as $h = n - 1$. In the latter case, substitute k by smaller value: $k := h$. Report the substitution with the value h.
5. Construct the list of all truncated Laurent solutions

$$c_v x^v + c_{v+1} x^{v+1} + \ldots + c_m x^m + O(x^{m+1}), \ \ v \in \{\alpha_1, \ldots, \alpha_s\}, \ \ m \leq k, \quad (12)$$

   containing no literals as described in [6, Sect. 4.1]. (Some elements of the set $\{\alpha_1, \ldots, \alpha_s\}$ might be not used in the truncated Laurent solutions (12)).

6 Implementation; Examples of Use

We implemented the algorithm in Maple [9] as an extension of `LaurentSolution` procedure from the package `TruncatedSeries` [7]. The first argument of the procedure is a differential equation $L(y) = 0$ where L is an operator of the form (1). Previously, the procedure worked for the case where all the series, which are the coefficients of the equation, are represented as truncated series. Now it is also possible to represent them (or a part of them) algorithmically. The application of θ^k to the unknown function $y(x)$ is written as `theta(y(x),x,k)`. The truncated coefficients of the equation, i.e. the coefficients of the form (2) are written as `a_i(x)+O(x^(t_i+1))`, where `a_i(x)` is a polynomial of the degree not higher than `t_i` over the field of algebraic numbers. Algorithmically represented series might be specified either as a polynomial, or as a finite or infinite power series in integer powers of x, or as a sum of a polynomial and such power series. The power series is written in a usual Maple form as `Sum(f(i)*x^i,i=a..b)`, where `f(i)` is an expression or a function that implements an algorithm for computing the number coefficient of the series with the index `i`, the specified `a` and `b` are the lower and the upper bounds of summation, the upper bound might be infinite which is designated as `infinity`. The coefficients of both polynomials and series, as in the case of truncated series, are the elements of the field of algebraic numbers. Irrational algebraic numbers are represented in Maple as the expression `RootOf(p(_Z), index = k)`, where `p(_Z)` is an irreducible polynomial, whose k-th root is the given algebraic number. For example, `RootOf(_Z^2-2, index=2)` represents $-\sqrt{2}$. An unknown function of the equation is specified as the second argument of the procedure.

Concerning the implementation of the algorithm from Sect. 5, the procedure has got two new optional parameters:

- `'top'=k` — where `k` is an integer number, for which it is needed to determine whether it exceeds the threshold of the given equation (by default, `k` equals the maximum integer root of the indicial polynomial if at least one coefficient of the equation is non-truncated and `k` equals the threshold otherwise);
- `'threshold'='h'` — where `h` specifies the name of the variable, which will be assigned to the value of the threshold in the case if it is computed, or to the value `FAIL`, if the threshold is not determined, i.e. if it exceeds the given value `k`.

The result of the procedure is a list of truncated Laurent solutions with different valuations. Each element of the list is represented as

$$c_{v_i}x^{v_i} + c_{v_i+1}x^{v_i+1} + \ldots + c_{m_i-1}x^{m_i-1} + O(x^{m_i}), \tag{13}$$

where v_i is the valuation for which the existence of a truncated Laurent solution is determined; m_i has the previous meaning, c_i are the calculated coefficients of the truncated Laurent solution, which can be linear combinations of arbitrary constants of the form $\_c_j$.

The implementation and a session of Maple with examples of using the procedure `LaurentSolution` are available at the address

http://www.ccas.ru/ca/truncatedseries

in the section "The next version of the procedure LaurentSolution".

Below we present six examples, which we combine into one, containing paragraphs 1-6.

Example 2.

1. In the equation two coefficients are given as a truncated series and one coefficient is represented algorithmically as the sum of the polynomial and the power series:

```
> eq1 := (-1+x+x^2+O(x^3))*theta(y(x), x, 2)+
>        (-2+O(x^3))*theta(y(x), x, 1)+
>        (1+x+Sum(x^i/i!, i = 2 .. infinity))*y(x);
```

$$eq1 := \left(-1 + x + x^2 + O\left(x^3\right)\right)\theta(y(x), x, 2) + \left(-2 + O\left(x^3\right)\right)\theta(y(x), x, 1) + \left(1 + x + \sum_{i=2}^{\infty} \frac{x^i}{i!}\right) y(x)$$

```
> LaurentSolution(eq1, y(x), 'top' = 2, 'threshold' = h1_2);
```

$$[\,]$$

The output means that the equation has no truncated Laurent solutions. The threshold:

```
> h1_2;
```

$$-\infty$$

The value of the threshold is $-\infty$, and it confirms that there are no truncated Laurent solutions.

2. The equation has both truncated and algorithmically represented coefficients. The separate function `f` is used to specify power series:

```
> f := (i -> i^2+2*i+1-(i+1)^2):
> eq2 := (-1+x+x^2+O(x^3))*theta(y(x), x, 2)+
>        (-2+O(x^3))*theta(y(x), x, 1)+
>        (Sum(f(i)*x^i, i = 0 .. infinity))*y(x);
```

$$eq2 := \left(-1+x+x^2+O\left(x^3\right)\right)\theta(y(x),x,2)+\left(-2+O\left(x^3\right)\right)\theta(y(x),x,1)$$
$$+\left(\sum_{i=0}^{\infty}\left(i^2+2i+1-(i+1)^2\right)x^i\right)y(x)$$

```
> LaurentSolution(eq2, y(x), 'top' = 2, 'threshold' = h2_2);
```

$$\left[\frac{\_c_1}{x^2}-\frac{4\_c_1}{x}+\_c_2+O(x),\_c_2+O\left(x^3\right)\right]$$

The truncated Laurent solutions with valuations -2 and 0 and with different truncation degrees are found. The threshold:

```
> h2_2;
```

$$\mathrm{FAIL}$$

It means that the given value $k=2$ does not exceed the threshold.

Apply the procedure to the given equation again with $k=5$:

```
> LaurentSolution(eq2, y(x), 'top' = 5, 'threshold' = h2_5);
```

$$\left[\frac{\_c_1}{x^2}-\frac{4\_c_1}{x}+\_c_2+O(x),\_c_2+O\left(x^6\right)\right]$$

It is seen that the truncated solution with the valuation -2 is not changed, and the truncation degree of the one with the valuation 0 is increased. The threshold:

```
> h2_5;
```

$$\mathrm{FAIL}$$

It means that the given value $k=5$ does not exceed the threshold. The function `f`, which is used to specify the series coefficient of $y(x)$, computes 0 coefficient for any index value i. Therefore the coefficient of $y(x)$ equals 0. The threshold of the equation is ∞, and any value k will not exceed the threshold. Note that the zero series might be specified just as the polynomial 0, or the term with $y(x)$ might be absent in the equation.

3. The equation has also both truncated and algorithmically represented coefficients. The algorithmically represented coefficient of $y(x)$ is written as the polynomial:

```
> eq3 := (-1+x+x^2+O(x^3))*theta(y(x), x, 2)+
>        (-2+O(x^3))*theta(y(x), x, 1)+(x+6*x^2)*y(x);
```

$$eq3 := \left(-1 + x + x^2 + O\left(x^3\right)\right)\theta(y(x), x, 2) + \left(-2 + O\left(x^3\right)\right)\theta(y(x), x, 1) + \left(x + 6x^2\right)y(x)$$

```
> LaurentSolution(eq3, y(x), 'top' = 2, 'threshold' = h3_2);
```

$$\left[\frac{\_c_1}{x^2} - \frac{5\_c_1}{x} + \_c_2 + O(x), \_c_2 + \frac{1}{3}x\_c_2 + \frac{5}{6}x^2\_c_2 + O\left(x^3\right)\right]$$

The truncated Laurent solutions with valuations -2 and 0 and with different truncation degrees are found again. The threshold:

```
> h3_2;
```

FAIL

It means that the given value $k = 2$ does not exceed the threshold.

Apply the procedure to the given equation with $k = 5$:

```
> LaurentSolution(eq3, y(x), 'top' = 5, 'threshold' = h3_5);
```

$$\left[\frac{\_c_1}{x^2} - \frac{5\_c_1}{x} + \_c_2 + O(x), \_c_2 + \frac{1}{3}x\_c_2 + \frac{5}{6}x^2\_c_2 + \frac{13}{30}x^3\_c_2 + O\left(x^4\right)\right]$$

The threshold:

```
> h3_5;
```

3

It is seen that the threshold is achieved in the computed truncated solutions with the valuation 0.

4. The equation is a prolongation of the equation eq3:

```
> eq4 := (-1+x+x^2+9*x^3+O(x^4))*theta(y(x), x, 2)
         +(-2+(x^3)/2+O(x^4))*theta(y(x), x, 1)+(x+6*x^2)*y(x);
```

$$eq4 := \left(-1 + x + x^2 + 9x^3 + O\left(x^4\right)\right)\theta(y(x), x, 2) + \left(-2 + \frac{1}{2}x^3 + O\left(x^4\right)\right)\theta(y(x), x, 1) + \left(x + 6x^2\right)y(x)$$

```
> LaurentSolution(eq4, y(x), 'top' = 5, 'threshold' = h4_5);
```

$$\left[\frac{\_c_1}{x^2} - \frac{5\_c_1}{x} + \_c_2 + \frac{1}{3}x\_c_2 + O\left(x^2\right), \_c_2 + \frac{1}{3}x\_c_2 + \frac{5}{6}x^2\_c_2 + \frac{13}{30}x^3\_c_2 + \frac{95}{144}x^4\_c_2 + O\left(x^5\right)\right]$$

The truncated Laurent solutions with valuations −2 and 0 and with different truncation degrees are found again. These truncated solutions are the prolongations of the computed truncated solutions of the equation `eq3`. The threshold:

```
> h4_5;
```

$$4$$

It is seen that the threshold is achieved in the computed truncated solutions with the valuation 0.

5. The equation is another prolongation of the equation `eq3`:

```
> eq5 := (-1+x+x^2+RootOf(z^2-2, z, index = 2)*x^3+O(x^4))*
> theta(y(x), x, 2)+(-2+2*RootOf(z^2-2, z, index = 2)*x^3+
> O(x^4))*theta(y(x), x, 1)+(x+6*x^2)*y(x);
```

$$eq5 := \left(-1 + x + x^2 + \mathrm{RootOf}\left(\_Z^2 - 2, index = 2\right)x^3 + O\left(x^4\right)\right)\theta(y(x), x, 2)$$
$$+\left(-2 + 2\mathrm{RootOf}\left(\_Z^2 - 2, index = 2\right)x^3 + O\left(x^4\right)\right)\theta(y(x), x, 1)$$
$$+\left(x + 6x^2\right)y(x)$$

```
> LaurentSolution(eq5, y(x), 'top' = 5, 'threshold' = h5_5);
```

$$\left[\frac{\_c_1}{x^2} - \frac{5\_c_1}{x} + \_c_2 + x\left(\frac{1}{3}\_c_2 - \frac{35}{3}\_c_1\right) + O\left(x^2\right), \_c_2 + \frac{1}{3}x\_c_2 + \frac{5}{6}x^2\_c_2\right.$$
$$\left. + \frac{13}{30}x^3\_c_2 + x^4\left(\frac{19}{36}\_c_2 + \frac{1}{24}\mathrm{RootOf}\left(\_Z^2 - 2, index = 2\right)\_c_2\right) + O\left(x^5\right)\right]$$

The truncated Laurent solutions with valuations −2 and 0 and with different truncation degrees are found again. These truncated solutions are the prolongations of the computed truncated solutions of the equation `eq3`, but are different from the computed truncated solutions of the equation `eq4`. The threshold:

```
> h5_5;
```

$$4$$

It is seen that the threshold is achieved again in the computed truncated solutions with the valuation 0.

The results of the application of the procedure to the equations `eq4` and `eq5` show that the earlier computed truncated Laurent solutions of the equation `eq3` contain the maximum possible number of initial terms, since two different prolongations of the equation `eq3` have different truncated Laurent solutions, which are the prolongations of the found truncated solutions of the equation `eq3`.

6. The equation is a prolongation of the equation `eq3` as well, and all its coefficients are represented algorithmically:

```
> eq6 := (-1+x+x^2+Sum((-1)^i*x^i/i!, i = 3 .. infinity))*
> theta(y(x), x, 2)+(-2+2*(Sum((-1)^i*x^i/i!, i = 3 ..
```

```
> infinity)))*theta(y(x), x, 1)+(x+6*x^2)*y(x);
```

$$eq6 := \left(-1 + x + x^2 + \sum_{i=3}^{\infty} \frac{(-1)^i x^i}{i!}\right)\theta(y(x), x, 2)+$$
$$\left(-2 + 2\left(\sum_{i=3}^{\infty} \frac{(-1)^i x^i}{i!}\right)\right)\theta(y(x), x, 1) + \left(x + 6x^2\right)y(x)$$

```
> LaurentSolution(eq6, y(x), 'top' = 5, 'threshold' = h6_5);
```

$$\left[\frac{\_c_1}{x^2} - \frac{5\_c_1}{x} + \_c_2 + x\left(\frac{1}{3}\_c_2 - \frac{35}{3}\_c_1\right)\right.$$
$$+x^2\left(\frac{5}{6}\_c_2 - \frac{145}{48}\_c_1\right) + x^3\left(\frac{13}{30}\_c_2 - \frac{103}{16}\_c_1\right) + x^4\left(\frac{25}{48}\_c_2 - \frac{2131}{576}\_c_1\right)$$
$$+x^5\left(\frac{2057}{5040}\_c_2 - \frac{4303}{960}\_c_1\right) + O\left(x^6\right), \_c_2 + \frac{1}{3}x\_c_2 + \frac{5}{6}x^2\_c_2$$
$$\left.+\frac{13}{30}x^3\_c_2 + \frac{25}{48}x^4\_c_2 + \frac{2057}{5040}x^5\_c_2 + O\left(x^6\right)\right]$$

The two truncated Laurent solutions with the same truncation degree are found, which are the prolongations of the computed truncated solutions of the equation `eq3`. The threshold:

```
> h6_5;
```

FAIL

Thus $k = 5$ does not exceed the threshold. The case when all the coefficients are represented algorithmically is the case when the threshold of the equation is ∞, and any value k does not exceed the value of the threshold.

7 Concluding Remarks

This study is a continuation of the studies started in [2, 7], in which it was assumed that either all the coefficients of a differential equation are represented algorithmically, and in this sense, are given completely, or are represented in the truncated form. In the current paper, the presence of both types coefficients is allowed.

The presence of infinite series in the input data of a problem is a source of difficulties (the algorithmic impossibility of answering certain natural questions). This is, e.g., related to the fact that if sequences of coefficients of series can be specified by arbitrary algorithms, then it is impossible to test algorithmically the equality of such series to zero (this is a consequence of the classical results of A. Turing on the undecidability of the problem of terminating of an algorithm [10]).

There is nowhere to go from this in the problem considered above, — see Proposition 1. However, along with this, we must admit that in the situation, we are faced, in a certain sense, with a lighter version of the algorithmic undecidability. This undecidability is, so to speak, not too burdensome.

Indeed, we cannot indicate the greatest degree of the truncated Laurent solution existing for a given equation (the threshold of the equation). However, if we are interested in all solutions of a truncation degree not exceeding a given integer k then the algorithm proposed in Sec. 5, allows us to construct all of them.

It would be interesting to try to obtain similar results for solutions of a more general form — the so-called regular and exponentially logarithmic solutions, and generalize this to systems of differential equations. We will continue to investigate this line of enquiry.

Acknowledgement

The authors are grateful to anonymous referees for their helpful comments.

References

1. Abramov, S., Barkatou, M.: Computable infinite power series in the role of coefficients of linear differential systems. In: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) CONFERENCE 2014, LNCS, vol. 8660, pp. 1–12. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10515-4_1
2. Abramov, S., Barkatou, M., Khmelnov, D.: On full rank differential systems with power series coefficients. Journal of Symbolic Computation **68**, 120–137 (2015)
3. Abramov, S., Barkatou, M., Pflügel, E.: Higher-order linear differential systems with truncated coefficients. In: Gerdt, V.P., Koepf, W., Mayr, E.W., Vorozhtsov, E.V. (eds.) CONFERENCE 2011, LNCS, vol. 6885, pp. 1024. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23568-9_2
4. Abramov, S., Bronstein, M., Petkovšek, M.: On polynomial solutions of linear operator equations. In: ISSAC '95: Proceedings of the 1995 international symposium on Symbolic and algebraic computation, pp. 290–296. Machinery, New York, NY, United States (1995)
5. Abramov, S., Khmelnov, D., Ryabenko, A.: Linear ordinary differential equations and truncated series. Comput. Math. Math. Phys. **49**(10), 1649–1659 (2019)
6. Abramov, S., Khmelnov, D., Ryabenko, A.: Regular solutions of linear ordinary differential equations and truncated series. Comput. Math. Math. Phys. **60**(1), 2–15 (2020)
7. Abramov, S., Khmelnov, D., Ryabenko, A.: Procedures for searching Laurent and regular solutions of linear differential equations with the coefficients in the form of truncated power series. Progr. and Comp. Soft. **46**(2), 67–75 (2020)
8. Lutz, D.A., Schäfke, R. On the identification and stability of formal invariants for singular differential equations. Linear Algebra and Its Applications **72** 1–46 (1985)
9. Maple online help, http://www.maplesoft.com/support/help/
10. Turing, A.: On computable numbers, with an application to the Entscheidungsproblem. In: the London Mathematical Society **42**(2), 230–265 (1937)