The `TruncatedSeries` Package for Solving Linear Ordinary Differential Equations Having Truncated Series Coefficients*

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 with power series in the role of coefficients. It is assumed that some or all of the series are truncated. A series of the form $\Sigma\, a_i x^i$ can also be given completely using an algorithm that computes a_i from i. The equation may contain both types of coefficients — truncated and represented algorithmically. Algorithms and commands that implement them in Maple as the `TruncatedSeries` package are proposed, which make it possible to find Laurent, regular and exponential-logarithmic solutions. In cases where, due to the presence of truncated coefficients, the information about the equation is incomplete, commands of our package find the maximum possible number of terms of those series that are involved in the solutions. If all the coefficients of the given equation are algorithmically represented series then the commands allow finding any specified number of initial terms of the series involved in the solutions.

Keywords: Differential equations · Truncated power series · Algorithmically represented infinite formal series

1 Introduction

Power and Laurent series are important and convenient tools of representing linear ordinary differential equations with variable coefficients as well as of representing solutions to these equations. This is reflected in theoretical studies (see, e.g., [17–23]) and found numerous application in computer algebra (see, e.g., [1–6, 15, 16, 24]).

Linear ordinary differential equations with coefficients in the form of truncated power series have been considered by us in [7–14]. Concerning the original differential equation we have incomplete information in this case: for a power series, only a finite number of initial terms are known. We are interested in the

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

information on the solutions of the equation given in this form that is invariant under all possible prolongations of all the truncated series that represent the coefficients of the equation (the *prolongation* is a series whose initial terms coincide with the known initial terms of the original truncated series). First, we have investigated what can be learned about the solutions in the field of Laurent formal series (we call them *Laurent solutions*) (see [7, 8]). Then a similar question has been discussed for *regular solutions* in [10]. In both cases, the proposed algorithms construct the maximum possible number of invariant initial terms of the series involved in the solutions.

The approach that we use in the algorithms for computing Laurent and regular solutions, has allowed us, in combination with the well-known algorithm of Newton polygons, to construct *formal exponential-logarithmic solutions* of linear ordinary differential equations having coefficients in the form of truncated power series (see [12, 13]). The series which appear in the solutions have also only a finite number of known initial terms.

Linear ordinary differential equations with the coefficients that are either algorithmically represented power series, or truncated power series have been considered as well in [11]. For such a mixed case, the problem of the construction of the maximum possible number of terms of the involved in the solutions series is algorithmically undecidable (for some such equations, the information is sufficient for computing any number of terms of the series). This undecidability is, so to speak, not too burdensome. If we are interested in all solutions with a truncation degree not exceeding a given integer d then the proposed algorithm allows to construct all of them.

All the developed algorithms are implemented by us as the `TruncatedSeries` package in Maple. Some examples of the use of the package procedures have been already presented in the preceding works [7–14], the corresponding algorithms being presented and justified there as well. In this work, we outline the current state of the package and present more examples to demonstrate its up to date key capabilities. We do not repeat the descriptions and justifications of the implemented algorithms. In the future we plan to extend the package possibilities, in particularly, to the case of the systems of linear ordinary differential equations having truncated series coefficients.

The Maple library with the `TruncatedSeries` package and Maple worksheets with examples of using its commands are available from

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

2 Short Specification of the Package

The `TruncatedSeries` package provides three commands `LaurentSolution`, `RegularSolution` and `FormalSolution`:

```
> with(TruncatedSeries);
```

$$[\,FormalSolution, LaurentSolution, RegularSolution\,]$$

Calling sequence of these three commands:

```
LaurentSolution(ode, var, opts),
RegularSolution(ode, var, opts),
FormalSolution(ode, var, opts)
```

with parameters

`ode` – a homogeneous linear ordinary differential equation;
`var` – a dependent variable, for example $y(x)$;
`opts` – a sequence of optional arguments of the form `keyword=value`.

The equation `ode` for $y(x)$ may be given in the *diff-form*:

$$a_r(x)\,\frac{\mathrm{d}^r}{\mathrm{d}x^r}\,y(x) + \cdots + a_1(x)\,\frac{\mathrm{d}}{\mathrm{d}x}\,y(x) + a_0(x)\,y(x) = 0$$

or in the *theta-form*:

$$a_r(x)\,\theta^r y(x) + \cdots + a_1(x)\,\theta\,y(x) + a_0(x)\,y(x) = 0.$$

where r is a positive integer and $\theta\,y(x) = x\,\frac{\mathrm{d}}{\mathrm{d}x}\,y(x)$. The derivative $\theta^k y(x)$ is specified as `theta(y(x), x, k)` for $k \geq 1$. The derivative $\frac{\mathrm{d}^k}{\mathrm{d}x^k}\,y(x)$ is specified using the ordinary Maple `diff` command.

Coefficients $a_r(x), \ldots, a_1(x), a_0(x)$ of the equation may be of two types. The first type is an algorithmically represented power series in one of the following forms:

– A polynomial in x over the algebraic number field.
– A finite power sum $\displaystyle\sum_{k=k_0}^{N} f(k)\,x^k$ with a summation index k, a non-negative integer low limit of summation k_0 and a non-negative integer upper limit of summation $N \geq k_0$. It has to be specified by means of the Maple inert `Sum` command. The coefficient $f(k)$ of x^k may be given by an arbitrary expression of the index k which gives an algebraic number for all $k \geq k_0$.
– An infinite power sum $\displaystyle\sum_{k=k_0}^{\infty} f(k)\,x^k$ with k_0, $f(k)$ as described above.
– A sum of a polynomial and power sums described above.

The other type of coefficients is a truncated power series in one of the following forms:

– $O\big(x^{t+1}\big)$, where t is an integer, $t \geq -1$.
– $a(x) + O\big(x^{t+1}\big)$, where $a(x)$ is a polynomial in x over the algebraic number field and t is an integer greater than or equal to the degree of $a(x)$.

The integer t is called the *truncation degree*. In the presented package, all algebraic numbers have to be represented as `RootOf(expr, x, 'index'=i)` where `expr` is an irreducible polynomial in x with rational number coefficients.

The following optional arguments can be used:

– 'top'=d, where d is an integer;
– 'threshold'='h', where h is a name of a variable.

Below we present the use of the commands with optional arguments 'top' and 'threshold'.

3 LaurentSolution

For an equation whose all coefficients are algorithmically represented power series, the LaurentSolution command determines a finite set of all integers i_0 such that the equation has Laurent series solutions with the *valuation* i_0, i.e. the equation has solutions in the form $\sum_{i=i_0}^{\infty} v(i)\, x^i$ where $v(i_0) \neq 0$.

If the option 'top' = d is given, the LaurentSolution command computes the initial terms of Laurent series solutions to the degree d or greater for each valuation i_0. The LaurentSolution command returns a list $[s_1, s_2, \ldots]$ of truncated Laurent series solutions for all found valuations. The elements of the list involve parameters of the form $\_c_1, \_c_2, \ldots$ For each element s_j these parameters can take any such values that the valuation of s_j does not change.

Below is an equations whose all coefficients are algorithmically represented power series:

```
> f := proc(i)
         if i::'integer' then 0 else 'procname'(i) end if;
       end proc:
  Ex1 := x^9*diff(y(x), x$5)+
         (x^7+Sum(k^2*x^k/2, k = 9 .. infinity))*diff(y(x), x$4)+
         (2*x^5+x^2)*diff(y(x), x$2)+
         (2*x^10+x^4+3*x)*diff(y(x), x)+
         Sum(f(k)*x^k, k = 0 .. infinity)*y(x) = 0;
```

$$Ex1 := x^9 \left(\frac{d^5}{dx^5}\, y(x)\right) + \left(x^7 + \left(\sum_{k=9}^{\infty} \frac{k^2 x^k}{2}\right)\right) \left(\frac{d^4}{dx^4}\, y(x)\right) + \left(2\,x^5 + x^2\right) \left(\frac{d^2}{dx^2}\, y(x)\right) + \left(2\,x^{10} + x^4 + 3\,x\right) \left(\frac{d}{dx}\, y(x)\right) + \left(\sum_{k=0}^{\infty} f(k)\, x^k\right) y(x) = 0$$

This equation has polynomial coefficients x^9, $2x^5 + x^2$. The coefficient for the third derivative is zero. There are also two infinite sums: one with the explicitly defined coefficients $\frac{k^2}{2}$ and the other with the coefficients defined by the Maple-procedure f.

For the equation Ex1 we obtain the list of two elements of solutions with the truncation degree 3 which is set by the option 'top':

```
> LaurentSolution(Ex1, y(x), 'top' = 3);
```

$$\left[\frac{\_c_1}{x^2} + \_c_2 - \frac{130\,x\_c_1}{3} + 90\,\_c_1 x^2 - 324\,x^3\_c_1 + O\big(x^4\big),\ \_c_2 + O\big(x^4\big)\right]$$

The truncation degree of the result may be `d1` which is greater than `d` if it is needed to compute initial terms up to the degree `d1` to determine if Laurent solutions with valuation i_0 exist. Below we obtain the list of two elements of Laurent solutions with the truncation degree 0. This is needed to determine if there are Laurent solutions with the valuation -2 and with the valuation 0:

```
> LaurentSolution(Ex1, y(x), 'top' = -1);
```

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

The same result will be obtained if the option `'top' = d` is not given:

```
> LaurentSolution(Ex1, y(x));
```

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

For an equation whose all coefficients are truncated series, the `LaurentSolution` command investigates what can be learned from the equation about its Laurent solutions. The command constructs the maximum possible number of initial terms of solutions which are defined uniquely by known terms of coefficients of the given equation. The maximum truncation degree may be different in the solutions with different valuations, that is why the `LaurentSolution` command forms the solution for each valuation separately. The greatest truncation degree of Laurent solutions is called the *threshold* of the given equation.

For example, the equation whose all coefficients are truncated power series with various truncation degrees:

```
> Ex2 := O(x^9)*diff(y(x), x$5)+
         (x^7+81/2*x^9+50*x^10+O(x^11))*diff(y(x), x$4)+
         O(x^7)*diff(y(x), x$3)+
         (2*x^5+x^2+O(x^7))*diff(y(x), x$2)+
         (x^4+3*x+O(x^5))*diff(y(x), x)+
         O(x^6)*y(x) = 0;
```

$$\begin{aligned} Ex2 := O\big(x^9\big)\left(\frac{\mathrm{d}^5}{\mathrm{d}x^5}\,y(x)\right) + \left(x^7 + \frac{81\,x^9}{2} + 50\,x^{10} + O\big(x^{11}\big)\right)\left(\frac{\mathrm{d}^4}{\mathrm{d}x^4}\,y(x)\right) \\ + O\big(x^7\big)\left(\frac{\mathrm{d}^3}{\mathrm{d}x^3}\,y(x)\right) + \big(2\,x^5 + x^2 + O\big(x^7\big)\big)\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\,y(x)\right) + \\ \big(x^4 + 3\,x + O\big(x^5\big)\big)\left(\frac{\mathrm{d}}{\mathrm{d}x}\,y(x)\right) + O\big(x^6\big)\,y(x) = 0 \end{aligned}$$

We know only several initial terms of all coefficients. The coefficients of $\frac{\mathrm{d}^5}{\mathrm{d}x^5}\,y(x)$, $\frac{\mathrm{d}^3}{\mathrm{d}x^3}\,y(x)$ and $y(x)$ are $O\big(x^9\big)$, $O\big(x^7\big)$ and $O\big(x^6\big)$, and we don't know if they are zero or not. For `Ex2` we obtain:

```
> LaurentSolution(Ex2, y(x));
```

$$\left[\frac{\_c_1}{x^2} + \_c_2 - \frac{130\,x\_c_1}{3} + O\big(x^2\big),\ \_c_2 + O\big(x^6\big)\right]$$

The first element of the returned list has the valuation -2. The maximum possible number of initial terms for it is equal to 4, the truncation degree is equal to 1. The second one has the valuation 0 and the maximum possible number of the initial terms is equal to 6, the truncation degree is equal to 5. So, the threshold for `Ex2` is equal to 5.

If the option `'top' = d` is given, then the `LaurentSolution` command handles `d` in the same way as described for equations whose all coefficients are algorithmically represented power series.

```
> LaurentSolution(Ex2, y(x), 'top' = 3, 'threshold' = 'h');
```

$$\left[\frac{\_c_1}{x^2} + \_c_2 - \frac{130\,x\_c_1}{3} + O\big(x^2\big),\ \_c_2 + O\big(x^4\big)\right]$$

Using the option `'threshold'='h'` we can obtain the information whether the given `d` is greater than the threshold of the `ode`. If it isn't then `h` is set equal to FAIL:

```
> h;
```

$$FAIL$$

Otherwise, `h` is set equal to the threshold. Below `h` is set equal to 5:

```
> LaurentSolution(Ex2, y(x), 'top' = 8, 'threshold' = 'h');
  'h' = h;
```

$$\left[\frac{\_c_1}{x^2} + \_c_2 - \frac{130\,x\_c_1}{3} + O\big(x^2\big),\ \_c_2 + O\big(x^6\big)\right]$$

$$h = 5$$

For an equation whose coefficients are of both types, in general, it's impossible to determine the greatest degree of truncated Laurent solutions. The threshold may be a finite number or infinity. Then if the option `'top'` is not given, the `LaurentSolution` command computes exactly as many initial terms of the solutions as needed to find a set of all valuations of the Laurent solutions of the given equation. For the equation

```
> Ex3 := O(x^9)*diff(y(x), x$5)+
     (x^7+Sum((1/2)*k^2*x^k, k = 9 .. infinity))*diff(y(x), x$4)+
     (2*x^5+x^2)*diff(y(x), x$2)+(x^4+3*x+O(x^5))*diff(y(x), x)+
     Sum(f(k)*x^k, k = 0 .. infinity)*y(x) = 0;
```

$$Ex3 := O(x^9)\left(\frac{d^5}{dx^5}\,y(x)\right) + \left(x^7 + \left(\sum_{k=9}^{\infty} \frac{k^2 x^k}{2}\right)\right)\left(\frac{d^4}{dx^4}\,y(x)\right) + \left(2\,x^5 + x^2\right)\left(\frac{d^2}{dx^2}\,y(x)\right) + \left(x^4 + 3\,x + O(x^5)\right)\left(\frac{d}{dx}\,y(x)\right) + \left(\sum_{k=0}^{\infty} f(k)\,x^k\right) y(x) = 0$$

we obtain

```
> LaurentSolution(Ex3, y(x));
```

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

If the option `'top' = d` is given, then the `LaurentSolution` command tries to compute all Laurent solutions to the truncation degree `d`. For `Ex3` it's only possible for the solutions having valuation 0 (see the second element of the returned list):

```
> LaurentSolution(Ex3, y(x), 'top' = 4);
```

$$\left[\frac{\_c_1}{x^2} + \_c_2 - \frac{130\,x\,\_c_1}{3} + O(x^2),\ \_c_2 + O(x^5)\right]$$

If the threshold of the equation is greater than or equal to `d` and the option `'threshold'='h'` is given, then `h` is set equal to FAIL:

```
> LaurentSolution(Ex3, y(x), 'top' = 4, 'threshold' = 'h'):
  h;
```

$$FAIL$$

In fact, the threshold of `Ex3` is equal to ∞. This equation has the solution $y(x) = \_c_2$, where $\_c_2$ is an arbitrary constant. The trailing coefficient of `Ex3` is $\sum_{k=0}^{\infty} f(k)\,x^k$ and the `LaurentSolution` command can check any finite number of values of $f(k)$:

```
> {seq(f(k), k = 0 .. 100)};
```

$$\{0\}$$

but there is no algorithm to check that $f(k) = 0$ for all integer $k \geq 0$.

If the given equation has no nonzero Laurent solution (the set of valuations of Laurent solutions is empty), then the `LaurentSolution` command returns the empty list:

```
> LaurentSolution(x^2*diff(y(x), x)+y(x), y(x));
```

$$[\,]$$

And it returns FAIL if the known terms of the coefficients of the given equation are not sufficient to find a set of valuations of Laurent solutions:

```
> LaurentSolution(O(x)*diff(y(x), x)+y(x), y(x));
```

$$FAIL$$

4 RegularSolution

For an equation with power series coefficients a formal regular solution is a finite sum of expressions in this form:

$$x^{\lambda}\left(\sum_{k=0}^{m}\left(\sum_{i=i_{k,0}}^{\infty} v_k(i)\,x^i\right)\ln^k x\right)$$

where λ is an algebraic number, m is a non-negative integer, $i_{0,0}, \ldots, i_{m,0}$ are integers and $v_k(i_{k,0}) \neq 0$ for $k = 0, 1, \ldots, m$.

Same as for the case of Laurent solutions, the definition of the threshold of the equation is introduced. For the `'top'` and `'threshold'` options, the `RegularSolution` command works in the same way as the `LaurentSolution` command.

Below, we obtain the truncated regular solutions with $\lambda = 0$ (the truncation degree is 4) and $\lambda = \frac{1}{3}$ (the truncation degree is 1). The threshold is computed, it is equal to 4:

```
> Sol := RegularSolution((-3+x+O(x^2))*theta(y(x), x, 2)+
                 (1+x+O(x^2))*theta(y(x), x, 1)+
                 (x^4+O(x^5))*y(x), y(x), 'threshold' = 'h');
  'h' = h;
```

$$Sol := \left[\_c_1 + \frac{x^4\_c_1}{44} + O\big(x^5\big) + x^{1/3}\left(\_c_2 + \frac{x\_c_2}{9} + O\big(x^2\big)\right)\right]$$

$$h = 4$$

Note that if the result `Sol` is combined in one series, for example using the `series` command, then the number of the initial terms is less then the maximum possible one (the term $\dfrac{x^4\_c_1}{44}$ is lost):

```
> series(Sol[1], x, infinity);
```

$$\_c_1 + \_c_2\,x^{1/3} + \frac{\_c_2\,x^{4/3}}{9} + O\Big(x^{7/3}\Big)$$

Below is an equation which has regular solutions with $\ln x$:

```
> RegularSolution((4+x+O(x^2))*theta(y(x), x, 2)+
                  O(x^2)*theta(y(x), x, 1)+
                  (x^3+O(x^4))*y(x), y(x), 'threshold' = 'h');
  'h' = h;
```

$$\left[\_c_2 + O(x^2) + \ln(x)\left(\_c_1 - \frac{x^3\_c_1}{36} + O(x^4)\right), \_c_2 - \frac{x^3\_c_2}{36} + O(x^4), O(x^2) + \ln(x)\left(\_c_1 - \frac{x^3\_c_1}{36} + O(x^4)\right)\right]$$

$$h = 3$$

The first element of the returned list is the truncated regular solutions with $\lambda = 0$ and $m = 1$, having two series with the valuation 0 and the truncation degrees 1 and 3. The second one is the truncated regular solutions with $\lambda = 0$ and $m = 0$, having one series with the valuation 0 and the truncation degree 3. The third one has the logarithm-free part with the valuation which is greater than 1. The threshold is computed, it is equal to 3.

If the equation has at least one completely given coefficient (below it is the coefficient of $\theta(y(x), x, 1)$ which is equal to 0) then we can use the command with different values of `d` in the option `'top' = d` to obtain the threshold.

Below we obtain that `h` is equal to FAIL if `'top' = 2`:

```
> Ex4 := (4+x+O(x^2))*theta(y(x), x, 2)+(x^3+O(x^4))*y(x):
  RegularSolution(Ex4, y(x), 'top' = 2, 'threshold' = 'h');
  'h' = h;
```

$$\left[\_c_2 + O(x^3) + \ln(x)\left(\_c_1 + O(x^3)\right), \_c_2 + O(x^3), O(x^3) + \ln(x)\left(\_c_1 + O(x^3)\right)\right]$$

$$h = \mathit{FAIL}$$

and we obtain that the threshold is equal to 3 if `'top' = 4`:

```
> RegularSolution(Ex4, y(x), 'top' = 4, 'threshold' = 'h');
  'h' = h;
```

$$\left[\_c_2 + x^3\left(-\frac{\_c_2}{36} + \frac{\_c_1}{54}\right) + O(x^4) + \ln(x)\left(\_c_1 - \frac{x^3\_c_1}{36} + O(x^4)\right), \_c_2 - \frac{x^3\_c_2}{36} + O(x^4), \frac{x^3\_c_1}{54} + O(x^4) + \ln(x)\left(\_c_1 - \frac{x^3\_c_1}{36} + O(x^4)\right)\right]$$

$$h = 3$$

The following equation has no nonzero formal regular solution (the set of possible λ of regular solutions is empty), the result is the empty list:

```
> RegularSolution(x*theta(y(x), x, 2)+y(x), y(x));
```

$$[\,]$$

For the following one, the known terms of the coefficients are not sufficient to determine the set of λ, the result is FAIL:

```
> RegularSolution(x*theta(y(x), x, 2)+
                  O(1)*theta(y(x), x, 1)+y(x), y(x));
```

$$FAIL$$

5 FormalSolution

A formal exponential-logarithmic solution has the form

$$\mathrm{e}^{Q(x)}\, x^{\lambda} \left(\sum_{k=0}^{m} \left(\sum_{i=i_{k,0}}^{\infty} v_k(i)\, x^{i/q} \right) \ln^k x \right)$$

where q is a positive integer, $Q(x)$ is a polynomial in $x^{-1/q}$, λ is an algebraic number, m is a non-negative integer, $i_{0,0}$, ..., $i_{m,0}$ are integers and $v_k(i_{k,0}) \neq 0$ for $k = 0, 1, \ldots, m$. Laurent and regular solutions are special cases of formal exponential-logarithmic solutions.

To construct all formal solutions for an equation with completely given coefficients, the `DEtools[formal_sol]` command can be used. For an equation whose coefficients may be truncated series the `FormalSolution` command of the presented `TruncatedSeries` package computes the maximum possible terms of the exponent $Q(x)$. If $Q(x)$ is obtained completely, the `FormalSolution` command then computes λ and initial terms of series which are components of solutions (if they are invariant to all possible prolongations of the given equation).

For the following equation, the given initial terms are only sufficient to obtain the one term of the exponent $Q(x)$. The unknown part of solutions is denoted by $y_1(x)$:

```
> Ex5 := (x^3 + O(x^4))*diff(y(x), x)+(2 + O(x))*y(x):
  FormalSolution(%, y(x));
```

$$\left[\mathrm{e}^{\frac{1}{x^2}}\, y_1(x) \right] \tag{1}$$

The following equation is a prolongation of `Ex5` (extra new known terms are added to the series coefficients). We obtain the second term of the exponent $Q(x)$. The notation $y_{reg}(x)$ in the result means that the exponential part of formal solutions is obtained completely:

```
> (x^4+x^3+O(x^5))*diff(y(x), x)+(2+x+O(x^2))*y(x):
  FormalSolution(%, y(x));
```

$$\left[e^{\frac{1}{x^2} - \frac{1}{x}} \, y_{reg}(x) \right] \tag{2}$$

Another prolongation of `Ex5` leads to another result:

```
> (x^3+(1/2)*x^4+O(x^5))*diff(y(x), x)+(2+x+O(x^2))*y(x):
  FormalSolution(%, y(x));
```

$$\left[e^{\frac{1}{x^2}} \, y_{reg}(x) \right] \tag{3}$$

Both (2) and (3) are prolongations of (1). It shows that (1) presents the maximum possible information about the solution which is invariant to all possible prolongations of `Ex5`.

Again and again, increasing the number of known terms in `Ex5` we obtain more information about solutions:

```
> (x^5+x^4+x^3+O(x^6))*diff(y(x), x)+(-x^2+x+2+O(x^3))*y(x):
  FormalSolution(%, y(x));
```

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

```
> (x^5+x^4+x^3+(3/2)*x^6+O(x^7))*diff(y(x), x)+
     (-x^2+x+2+O(x^4))*y(x):
  FormalSolution(%, y(x));
```

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

```
> (x^5+x^4+x^3+(3/2)*x^6+(1/4)*x^7+O(x^8))*diff(y(x), x)+
      (-x^2+x+2+O(x^5))*y(x):
  FormalSolution(%, y(x));
```

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

The result of the `FormalSolution` command may contain the following expressions: $y_{reg}(x^{1/q})$, $y_{irr(p)}(x)$, $y_{irr}(x)$, $y_i(x)$, where y, x are given via the second parameter of the command, q and i are positive integers, p is a rational number.

As mentioned above, the notation $y_{reg}(x^{1/q})$ in the result means that the exponent $Q(x)$ (together with the number q) is obtained completely but the algebraic number λ is not invariant to all possible prolongations of the given equation (see (2) and (3) where q is equal to 1).

If the result has a term in the form

$$e^{Q_1(x)} \, y_{irr(p)}(x)$$

then it means that all prolongations of the given equation have formal solutions with the exponential

$$Q(x) = Q_1(x) + \frac{b}{x^p} + Q_2(x) \tag{4}$$

where $Q_1(x)$ and p are invariant to all possible prolongations (and the command computes them) but $b \neq 0$ is not invariant.

If the result has a term in the form

$$\mathrm{e}^{Q_1(x)}\, y_{irr}(x)$$

then it means that all prolongations of the given equation have formal solutions with the exponential (4) but p is not invariant as well as $b \neq 0$.

If the result has a term in the form

$$\mathrm{e}^{Q_1(x)}\, y_i(x)$$

where i is an integer it means that there are prolongation of the given equation having solutions with the exponent (4) and $b \neq 0$, and there are other ones having solutions with the exponent (4) and $b = 0$ and $Q_2(x) = 0$.

If different terms with the same expressions $y_{reg}(x^{1/q})$, or $y_{irr(p)}(x)$, or $y_{irr}(x)$ appear in the result, then such expressions are additionally indexed as follows: $y_{reg,1}(x^{1/q})$, $y_{reg,2}(x^{1/q})$, etc.

For example,

```
> Ex6 := (x^5 + O(x^6))*diff(y(x), x$3) +
         (-3*x^3 + O(x^4))*diff(y(x), x$2) +
         O(x)*diff(y(x), x) + (2 + O(x))*y(x) = 0:
  FormalSolution(Ex6, y(x));
```

$$\left[y_1(x) + y_{irr}(x) + y_{irr(1)}(x)\right]$$

The given initial terms are only sufficient to obtain the following information:

- all prolongations of `Ex6` have a three-dimensional linear space of formal exponential-logarithmic solutions;
- the first term $y_1(x)$ of the result means that there are prolongations of `Ex6` that have a one-dimensional space of regular solutions, and there are prolongations that do not have regular solutions;
- the second term $y_{irr}(x)$ means that all prolongations of `Ex6` have such irregular solutions that the exponent $Q(x)$ has no invariant terms;
- the last term $y_{irr(1)}(x)$ means that all prolongations of `Ex6` have at least a one-dimensional space of irregular solutions with an exponent (4), where $Q_1(x) = 0$ and $p = 1$ but b is not invariant.

Below are two prolongations of `Ex6` confirming the above:

```
> (x^5 + O(x^6))*diff(y(x), x$3) +
    (-3*x^3 + O(x^4))*diff(y(x), x$2) +
      (2*x + O(x^2))*diff(y(x), x) + (1 + O(x))*y(x) = 0:
  FormalSolution(%, y(x));
```

$$\left[\frac{\_c_1 + O(x)}{\sqrt{x}} + \mathrm{e}^{-\frac{2}{x}} y_{reg,1}(x) + \mathrm{e}^{-\frac{1}{x}} y_{reg,2}(x)\right]$$

```
> (x^5 + O(x^6))*diff(y(x), x$3) +
     (-3*x^3 + O(x^4))*diff(y(x), x$2) +
       O(x^2)*diff(y(x), x) + (1 + O(x))*y(x) = 0:
  FormalSolution(%, y(x));
```

$$\left[\mathrm{e}^{-\frac{2 RootOf(3\_Z^2-1, index=1)}{\sqrt{x}}} y_{reg,1}\left(\sqrt{x}\right) + \mathrm{e}^{-\frac{2 RootOf(3\_Z^2-1, index=2)}{\sqrt{x}}} y_{reg,2}\left(\sqrt{x}\right) + \mathrm{e}^{-\frac{3}{x}} y_{reg,3}(x)\right]$$

The following equation is also a prolongation of `Ex6`. It contains enough information to construct the exponential parts of the solutions completely. The solution involves series in fractional powers of x:

```
> (x^5 + x^6 + O(x^7))*diff(y(x), x$3) +
     (-3*x^3 - x^4 + O(x^5))*diff(y(x), x$2) +
       (1 + x + O(x^2))*y(x) = 0:
  FormalSolution(%, y(x));
```

$$\Bigg[\mathrm{e}^{-\frac{2 RootOf(3\_Z^2-1, index=1)}{\sqrt{x}}} x^{29/36}\Bigg(\_c_1 + \frac{191\, RootOf\left(3\_Z^2 - 1, index = 1\right) \_c_1\sqrt{x}}{432} - \frac{82679\, \_c_1 x}{1119744} + O\left(x^{3/2}\right)\Bigg) + \mathrm{e}^{-\frac{2 RootOf(3\_Z^2-1, index=2)}{\sqrt{x}}} x^{29/36}\Bigg(\_c_2 + \frac{191\, RootOf\left(3\_Z^2 - 1, index = 2\right) \_c_2\sqrt{x}}{432} - \frac{82679\, \_c_2 x}{1119744} + O\left(x^{3/2}\right)\Bigg) + \mathrm{e}^{-\frac{3}{x}} x^{17/9}\left(\_c_3 + O(x)\right)\Bigg]$$

References

1. Abramov, S., Barkatou, M.: Computable infinite power series in the role of coefficients of linear differential systems. Lecture Notes in Computer Science **8660**, 1–12 (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**(1), 120–137 (2015). https://doi.org/https://doi.org/10.1016/j.jsc.2014.08.010
3. Abramov, S., Barkatou, M., Pflügel, E.: Higher-order linear differential systems with truncated coefficients. Lecture Notes in Computer Science **6885**, 10–24 (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: Proceedings ISSAC'95. pp. 290–296 (1995). https://doi.org/10.1145/220346.220384
5. Abramov, S., Khmelnov, D.: Regular solutions of linear differential systems with power series coefficients. Programming and Computer Software **40**(2), 98–106 (2014). https://doi.org/10.1134/S0361768814020029
6. Abramov, S., Khmelnov, D., Ryabenko, A.: Procedures for searching local solutions of linear differential systems with infinite power series in the role of coefficients. Programming and Computer Software **42**(2), 55–64 (2016). https://doi.org/10.1134/S036176881602002X
7. Abramov, S., Khmelnov, D., Ryabenko, A.: Laurent solutions of linear ordinary differential equations with coefficients in the form of truncated power series. In: COMPUTER ALGEBRA, Moscow, June 17–21, 2019, International Conference Materials. pp. 75–82 (2019)
8. Abramov, S., Khmelnov, D., Ryabenko, A.: Linear ordinary differential equations and truncated series. Computational Mathematics and Mathematical Physics **59**, 1649–1659 (2019). https://doi.org/10.1134/S0965542519100026
9. 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. Programming and Computer Software **46**(2), 67–75 (2020). https://doi.org/10.1134/S0361768820020024
10. Abramov, S., Khmelnov, D., Ryabenko, A.: Regular solutions of linear ordinary differential equations and truncated series. Computational Mathematics and Mathematical Physics **60**, 1–14 (2020). https://doi.org/10.1134/S0965542520010029
11. Abramov, S., Khmelnov, D., Ryabenko, A.: Truncated and infinite power series in the role of coefficients of linear ordinary differential equations. Lecture Notes in Computer Science **12291**, 63–76 (2020). https://doi.org/10.1007/978-3-030-60026-6_4
12. Abramov, S., Khmelnov, D., Ryabenko, A.: Truncated series and formal exponential-logarithmic solutions of linear ordinary differential equations. Computational Mathematics and Mathematical Physics **60**, 1609–1620 (2020). https://doi.org/10.1134/S0965542520100024
13. Abramov, S., Khmelnov, D., Ryabenko, A.: Truncated series (in Russian). In: Differential equations and related questions of mathematics, works of the XII Prioksky scientific conference, June 19-20, 2020. pp. 8–19 (2020), http://kolomnamath.ru/download/Kolomna_Sbornik_2020.pdf
14. Abramov, S., Khmelnov, D., Ryabenko, A.: Procedures for constructing truncated solutions of linear differential equations with infinite and truncated power series in the role of coefficients. Programming and Computer Software **47**(2), 144–152 (2021). https://doi.org/10.1134/S036176882102002X
15. Abramov, S., Petkovšek, M.: Special power series solutions of linear differential equations. In: Proceedings FPSAC96. pp. 1–8 (1996)
16. Barkatou, M.: Rational Newton algorithm for computing formal solutions of linear differential equations. Lecture Notes in Computer Science **358**, 183–195 (1989). https://doi.org/https://doi.org/10.1007/3-540-51084-2_17

17. Coddington, E., Levinson, N.: Theory of Ordinary Differential Equations. Krieger (1984)
18. Frobenius, G.: Über die Integration der linearen Differentialgleichungen durch Reihen. Journal für die reine und angewandte Mathematik **76**, 214–235 (1873). https://doi.org/10.1515/crll.1873.76.214
19. Heffter, L.: Einleitung in die Theorie der linearen Differentialgleichungen mit einer unabhängigen Variablen. BG Teubner, Leipzig (1894)
20. Ince, E.: Ordinary Differential Equations. London, New York, Bombay: Longmans (1926)
21. Kamke, E.: Differentialgleichungen. Lösungsmethoden und Lösungen I. Gewöhnliche Differentialgleichungen. Leipzig (1942)
22. Malgrange, B.: Sur la réduction formelle des équations différentielles à singularités irrégulières. Université Scientifique et Médicale de Grenoble (1979)
23. Schlesinger, L.: Handbuch der Theorie der linearen Differentialgleichungen, Bd. 1. Teubner, Leipzig (1895)
24. Singer, M.F.: Formal solutions of differential equations. Journal of Symbolic Computation **10**(1), 59–94 (1990). https://doi.org/10.1016/S0747-7171(08)80037-5