Algorithm EG as a Tool for Finding Laurent Solutions of Linear Differential Systems with Truncated Series Coefficients

D.E. Khmelnov[1], A.A. Ryabenko[1]

[1]*Federal Research Center "Computer Science and Control" of RAS, Russia*

e-mail: anna.ryabenko@gmail.com, dennis_khmelnov@mail.ru

Abstract

Laurent solutions of systems of linear ordinary differential equations with the truncated power series coefficients are considered. The Laurent series in the solutions are also truncated. We use induced recurrent systems for constructing the solutions and have previously proposed an algorithm for the case when the induced system has a non-singular leading matrix. The algorithm finds the maximum possible number of terms of the series in the solutions that are invariant with respect to any prolongation of the original system. Below we present advances in extending our algorithm to the case when the leading matrix is singular using algorithm EG as an auxiliary tool.

Keywords: differential systems, truncated series coefficients, truncated Laurent series solutions

1. Starting Point

We consider systems of the form

$$A_r(x)\theta^r y(x) + A_{r-1}(x)\theta^{r-1} y(x) + \cdots + A_0(x)y(x) = 0, \tag{1}$$

where $y(x) = (y_1(x), y_2(x), \ldots, y_m(x))^T$ is the vector of unknowns, $A_r(x), \ldots, A_0(x)$ are $m \times m$-matrices of coefficients with entries in the form of power series in x over the field of algebraic numbers, $\theta = x\frac{d}{dx}$.

A solution $y(x) = (y_1(x), \ldots, y_m(x))^T$ of the differential system (1) the components of which are formal Laurent series is referred to as a Laurent solution:

$$y(x) = \sum_{n=v}^{\infty} u(n)x^n, \tag{2}$$

where $v \in \mathbb{Z}$ is a *valuation* of the series, $u(n) = (u_1(n), \ldots, u_m(n))^T$ are vectors of coefficients of Laurent series for $n \in \mathbb{Z}$.

For a full-rank system the coefficients of which are series specified algorithmically (i.e. an algorithm is given which computes the coefficient of any term x^s of any series), the algorithm from [2] finds all its truncated Laurent solutions with any given truncation degree. The algorithm is based on the construction of an *induced recurrent system* $R(u) = 0$ which is satisfied by the sequence of vectors $u(n)$ from (2). The induced recurrent system is constructed with the transformation

$$x \to E^{-1}, \ \ \theta \to n, \tag{3}$$

applied to the original differential system (1). E^{-1} denotes the shift operator: $E^{-1}u(n) = u(n-1)$. Thus $R = B_0(n) + B_{-1}(n)E^{-1} + B_{-2}(n)E^{-2} + \ldots$ and the induced system is written as

$$B_0(n)u(n) + B_{-1}(n)u(n-1) + \ldots = 0, \tag{4}$$

where $u(n) = (u_1(n), \dots, u_m(n))^T$ is the column vector of unknown sequences such that $u_i(n) = 0$ for all negative n with large enough value of $|n|$, $i = 1, \dots, m$; $B_0(n), B_{-1}(n), \dots$ are matrices of polynomials in n; $B_0(n)$ is the *leading matrix* of system (4). If $B_0(n)$ is non-singular, then we can consider the equation $\det B_0(n) = 0$ as an indicial equation of the original differential system: the set of integer roots of this algebraic equation includes the set of all possible valuations of the Laurent solutions of system (1). This makes it possible, in particular, to find the lower bound for the valuations of all Laurent solutions of the system. If $\det B_0(n) = 0$ has no integer roots, then the system has no Laurent solutions. If $B_0(n)$ is singular, then algorithm $\mathrm{EG}_\sigma^\infty$ (it is a version of original EG algorithm from [1] introduced in [2] for infinite recurrent systems) initially applied to transform the induced recurrent system to the embracing recurrent system of the same form with a non-singular leading matrix, and it constructs any given number of its initial terms. The embracing recurrent system supplemented with a set of linear constraints, which are also constructed by $\mathrm{EG}_\sigma^\infty$ algorithm, has the same set of solutions as the system (4).

In our current research we are focused on the case when the series in the system coefficients are represented in a truncated form, with the truncation degree being different for different coefficients. We refer to such systems as *truncated systems*. Each truncated series is represented as

$$a(x) + O(x^{t+1}), \tag{5}$$

where $a(x)$ is a polynomial, the integer $t \geq \deg a(x)$ is a *truncation degree*.

The *prolongation* of a truncated series is a series (possibly, also truncated) the initial terms of which coincide with known initial terms of the original truncated series. In turn, the prolongation of a truncated equation is an equation the coefficients of which are prolongations of coefficients of the original equation, and the prolongation of a system of equations is a system the equations of which are prolongations of the equations of the original system.

We are interested in finding the maximum possible number of terms of truncated Laurent solutions of a given truncated system that are invariant with respect to any prolongations of the truncated coefficients of the given system. Solutions with arbitrary truncation degree cannot be calculated for a truncated system. This statement was proved in [3] for a particular case, namely, for a scalar equation ($m = 1$). In [3] we also proposed an algorithm which finds such truncated Laurent solutions for scalar equations. We utilized induced recurrent systems and *literals* as a foundation for finding the solutions. Literals are symbols used to represent unspecified coefficients of the truncated series involved in the system. For a series of form (5), we say that the coefficients of terms x^s,$s > t_{kij}$ are *unspecified*. When constructing solutions of the truncated system, these coefficients are represented by symbols, i.e., by literals. Our algorithm for truncated systems is a modification of the algorithms for the systems with the algorithmically specified coefficients. The key idea is to represent the truncated series (5) algorithmically: the algorithm returns the known coefficient of the series if $s \leq t$, and it returns literals if $s > t$.

In [6] we applied the approach from [3] to the systems with $m > 1$ and proposed an algorithm for constructing Laurent solutions of the system for the case when the determinant of the leading matrix of the induced system is not zero (i.e. the leading matrix is non-singular) and does not contain literals.

2. Advances and Further Plans

Our advances in the research of finding Laurent solutions of the systems in hand are related to the use of the algorithm $\mathrm{EG}_\sigma^\infty$ to extend the applicability of the algorithm from [6] to the systems whose induced recurrent systems have singular leading matrix. We continue the adaptation of the algorithm for the systems with the algorithmically specified coefficients by representing the truncated series algorithmically with the help of literals.

The $\mathrm{EG}_\sigma^\infty$ algorithm consists in the successive repetition of reduction and shift steps, which continues until the rows of the leading matrix remain linear dependent. On the reduction step, coefficients of the dependence are found; then, the equation corresponding to one of the dependent rows is replaced with the linear combination of the other equations, hence the row of the leading matrix is set zero. On the shift step, the shift operator E is applied to the new equation. The termination of the algorithm is guaranteed with using a simple rule when selecting equations to be replaced. The reduction steps also lead to a finite set of linear constraints, each of which involves a finite number of elements of a sequential solution and is a linear combination of these elements with constant coefficients. The linear constraints correspond to the integer roots of a polynomial which is the coefficient of the row of the replaced equation in the linear combination (we further refer to the polynomials as the *constraint polynomials*).

The main obstacle is that literals may appear in intermediate calculations. If there are no literals both in the determinant of the leading matrix and in the constraint polynomials after $\mathrm{EG}_\sigma^\infty$ execution then the further calculations with the resulting induced recurrence and the linear constraints in the way as in algorithm from [6] gain desired truncated Laurent solutions which form the extended version of our algorithm in the case. It is preliminary implemented in Maple ([7]) as an updated version of procedure LaurentSolution of package TruncatedSeries [4, 5] (for more information about the package, please visit http://www.ccas.ru/ca/TruncatedSeries).

Let's consider the following system

$$\begin{pmatrix} 3x+O(x^2) & 7x^2+O(x^4) \\ O(x^2) & 17x^2+O(x^4) \end{pmatrix} \theta^2 y(x) + \begin{pmatrix} -1+2x+O(x^2) & x+5x^2+O(x^4) \\ O(x^2) & 11x^2+O(x^4) \end{pmatrix} \theta y(x) + \\ + \begin{pmatrix} O(1) & x-3x^2+O(x^4) \\ 1+O(x^2) & -6x^2+O(x^4) \end{pmatrix} y(x) = 0. \quad (6)$$

The leading matrix of its induced recurrent system is singular:

$$\begin{pmatrix} U_{[1,1],[0,0]} - n & 0 \\ 1 & 0 \end{pmatrix}.$$

$U_{[i,j][k,s]}$ is the representation of the literal denoting an unspecified coefficient of x^s in (i,j)-th element of matrix coefficient $A_k(x)$ of the system (1). After execution of $\mathrm{EG}_\sigma^\infty$ the transformed leading matrix becomes non-singular:

$$\begin{pmatrix} 3n^2+2n+U_{[1,1],[0,1]} & n+1 \\ 1 & 0 \end{pmatrix},$$

and its determinant $(-n-1)$ contains no literals. No linear constraints constructed by $\mathrm{EG}_\sigma^\infty$, so there is no constraint polynomials with literals as well. It is the case when our new algorithm is applicable and it computes the truncated Laurent solution (c_1 is an arbitrary constant):

$$\begin{pmatrix} 6x^2c_1+O(x^3) \\ \frac{c_1}{x}+c_1+O(x) \end{pmatrix}.$$

Let's consider another system:

$$\begin{pmatrix} O(x^5) & -1+O(x^5) \\ 1+O(x^5) & O(x^5) \end{pmatrix} \theta y(x) + \begin{pmatrix} O(x^5) & O(1) \\ 2+O(x^5) & O(x^5) \end{pmatrix} y(x). \quad (7)$$

The leading matrix of its induced recurrent system is already non-singular:

$$\begin{pmatrix} 0 & U_{[1,2],[0,0]} - n \\ 2+n & 0 \end{pmatrix}.$$

However there is a literal in its determinant $(n - U_{[1,2],[0,0]})(2+n)$. It is seen that the determinant has the roots -2 and $U_{[1,2],[0,0]}$. It means that the set of integer roots of the determinant may be different for different integer values of the literal $U_{[1,2],[0,0]}$. It is easy to check that there is no desired Laurent solutions of the system. It may be done by constructing various prolongations of the original system with substituting various integer values of the literal $U_{[1,2],[0,0]}$ and finding Laurent solutions of the prolongations with our algorithm from [6] (the algorithm is applicable to the prolongations since for each of them the leading matrix of the induced recurrence is still non-singular and there are no literals in its determinant already). For example, for $U_{[1,2],[0,0]} = 5$ the solution of the prolongation is

$$\begin{pmatrix} O(x^{10}) \\ c_1x^5 + O(x^6) \end{pmatrix}$$

and for $U_{[1,2],[0,0]} = 6$ the solution of the prolongation is

$$\begin{pmatrix} O(x^{11}) \\ c_1x^6 + O(x^7) \end{pmatrix}.$$

Since the solutions of the prolongations has no coinciding initial terms of the series, there is no desired Laurent solution of the original system.

Let $p(n)$ be the determinant of the leading matrix (either of the induced recurrent system if its leading matrix is non-singular, or of the embraced recurrent system after the application of $\mathrm{EG}_\sigma^\infty$ otherwise). If $p(n)$ contains literals then it may be represented in the general case as $p(n) = a(u_1,\dots,u_s)(n-r_1)\dots(n-r_k)(b_q(u_1,\dots,u_s)n^q + \cdots + b_1(u_1,\dots,u_s)n + b_0(u_1,\dots,u_s))$, where $u_1,\dots,u_s$ are literals involved in $p(n)$, $a(u_1,\dots,u_s)$, $b_0(u_1,\dots,u_s)$, $b_1(u_1,\dots,u_s), \cdots, b_q(u_1,\dots,u_s)$ are polynomials in the literals, $r_1 \dots r_k$ are integer roots of $p(n)$ independent from the literals. If $a(u_1,\dots,u_s)$ does contain literals (i.e. it is not a number) then there are such values of the literals $u_1,\dots,u_s$ that $a(u_1,\dots,u_s) = 0$ and, hence $p(n) = 0$, i.e. the leading matrix is singular for the values of the literals. If $a(u_1,\dots,u_s)$ is a number, then the solution of the algebraic equation $(b_q(u_1,\dots,u_s)r_0^q + \cdots + b_1(u_1,\dots,u_s)r_0 + b_0(u_1,\dots,u_s)) = 0$ in respect to $u_1,\dots,u_s$ allows getting such values of the literals that $p(n)$ has any desired root r_0 in addition to the roots $r_1 \dots r_k$. It means that in all cases the set of integer roots of $p(n)$ is not invariant in respect to the prolongations of the differential system in hand. The same reasoning is true for the constraint polynomials as well.

$\mathrm{EG}_\sigma^\infty$ may have variability in its execution. In spite of the fact that the special rule should be used for choosing the equation to be replaced to guarantee the termination of the computation, it is still possible to have more than one option to choose. It leads to the fact that for the same recurrent system $\mathrm{EG}_\sigma^\infty$ may result in different embraced systems. For example, for the system (6) $\mathrm{EG}_\sigma^\infty$ may be executed in another way that leads to the embraced recurrent system with another leading matrix:

$$\begin{pmatrix} U_{[1,1],[0,0]} - n & 0 \\ 3n^2 + 2n + U_{[1,1],[0,1]} & n+1 \end{pmatrix}$$

with the determinant $\left(U_{[1,1],[0,0]} - n\right)(n+1)$ which contains the literal $U_{[1,1],[0,0]}$. It can be seen that the determinant is very similar to the determinant of the leading matrix of the induced recurrence for the system (7) that has no desired truncated Laurent solution. In addition, the second variant of $\mathrm{EG}_\sigma^\infty$ execution gives the constraint polynomial $-U[[1,1],[0,0]]+n$ which also contains the literal. Still, we know from the first variant of $\mathrm{EG}_\sigma^\infty$ execution, that the system has desired truncated Laurent solution. It gives us the counterexample to the conjecture that there is no desired Laurent solution as soon as the determinant of the leading matrix and/or the constraint polynomials contain literals.

We experiment with the modification of our algorithm for the case of determinants and constraint polynomials containing literals, which takes into account only invariant integer roots of the determinant and constraint polynomial (i.e. those roots which are independent of literals). The experiments show that the modification of the algorithm gives correct answers for the system (6) for both variants of $\mathrm{EG}_\sigma^\infty$ execution and for the system (7), as well as for more other systems. Our further plans is either to prove that the approach is always correct or to identify the limitations of its applicability.

References

[1] *Abramov S.A.* EG-eliminations. J. Difference Equations Appl. 1999. Vol. 5. P. 393–433.

[2] *Abramov S.A., Barkatou M.A., Khmelnov D.E.* On full rank differential systems with power series coefficients. J. Symbolic Comput. 2015. Vol. 68. P. 120–137.

[3] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Linear ordinary differential equations and truncated series. Comput. Math. and Math. Phys. 2019. Vol. 59, N. 10. P. 1649–1659.

[4] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Procedures for constructing truncated solutions of linear differential equations with infinite and truncated power series in the role of coefficients. Program. Comput. Software. 2021. Vol. 47. P. 144–152.

[5] *Abramov S.A., Khmelnov D.E., Ryabenko A.A.,* The TruncatedSeries package for solving linear ordinary differential equations having truncated series coefficients. Maple in Mathematics Education and Research, Springer Nature Switzerland. 2021. P. 19–33.

[6] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Searching for Laurent solutions of systems of linear differential equations with truncated power series in the role of coefficients. Program. Comput. Software. 2023 (to be published).

[7] *Maplesoft* Maple online help. http://www.maplesoft.com/support/help/