Checkpoints in searching for rational solutions of linear ordinary difference and differential systems

Sergei A. Abramov, Denis E. Khmelnov, Anna A. Ryabenko
Dorodnicyn Computing Center,
Federal Research Center "Computer Science and Control"
of the Russian Academy of Sciences
Vavilova 40, Moscow, Russia, 119333
sergeyabramov@mail.ru,
dennis_khmelnov@mail.ru,
anna.ryabenko@gmail.com

Abstract

It is quite common that search algorithms for those solutions of difference and differential equations and systems that belong to a fixed class of functions are designed so that nonexistence of solutions of the desired type is detected only in the last stages of the algorithm. However, performing additional tests on the intermediate results makes it possible to stop the algorithm as soon as these tests imply that no solutions of the desired type exist. This gives an opportunity to save time and other computing resources. So, it makes sense to equip algorithms with checkpoints and some tests. We consider these questions in connection with the search for rational solutions of linear homogeneous difference and differential systems with polynomial coefficients, and propose a scheme equipped with such checkpoints and tests, and also report results of experiments with our implementation of the scheme in Maple.

1 Introduction

Algorithms for finding rational solutions of linear differential and difference equations and systems are one of the basic building blocks of all differential and difference equation solvers in any symbolic computation system. Therefore it is clear that improving efficiency of such algorithms is of great importance. Our goal is to show how it is often possible to detect nonexistence of rational solutions of a linear differential or difference system already in the early stages of the algorithm, thus saving a significant amount of time and other computing resources. For instance, if the indicial polynomial (whose roots include valuations for all nonzero rational solutions) has no integer roots, the algorithm can be terminated immediately, and similarly later on in the computation of the denominator and numerator degree bounds. Concerning this example, it is important to note that the construction of the indicial equations can be performed at different moments. It is desirable to rebuild the algorithm so that the corresponding indicial equations appeared before the laborious steps of the algorithm. Then the absence of integer roots of the indicial equation eliminates the need to perform these laborious steps. Thus, in some cases it makes sense to slightly change the existing algorithm without affecting its main scheme (e.g., to change the order of some operations that is not of importance, etc.).

Equipping with checkpoints algorithms for finding rational solutions in the scalar case (i.e., the case of one equation) was proposed by A.Gheffar in [16, 17]. Later, in [4] some checkpoints and tests were proposed for linear differential systems. In the present paper, we consider in detail such questions for linear difference systems.

It will be appropriate to note that the search for solutions of systems has its own specificity in comparison with the scalar case, and in the case of the systems themselves there is a significant dissimilarity for the differential and difference cases. For example, this takes place in the construction of the so-called universal denominator (the definition is given in Sect. 2). The difference case poses its own problems, and we will discuss them below.

In [16, 17, 4] the possibility of using checkpoints in the parts of algorithms for constructing the numerators of solutions was not considered. Only an upper bounds for the degrees of these polynomials was found. It was assumed that the method of undefined coefficients could be applied further. In our paper, this issue is given some attention.

Our focus in the paper is on linear difference systems. It was said that the differential case was already considered in [4]; more precisely, only the theoretical aspect was discussed, computer implementation was not offered. In Sect. 5 a brief information on the differential case is given. Sect. 6 describes our implementation of the proposed scheme in Maple [21]. The experiments of comparing two versions of the algorithm with and without checkpoints are discussed. It becomes apparent that the version with checkpoints detects the absence of rational solutions significantly faster. In the presence of rational solutions, a slight increase in time is observed. Tests performed at checkpoints do not take much time. Our experiments show that in the absence of rational solutions, time savings are about 75%, and the additional time in the presence of solutions does not exceed 20–25%.

Consideration of the algorithm from some special point of view will allow, possibly, to add some more checkpoints that are not considered in our paper. Similarly, one can ignore some of the points in our set. General rule: there is no sense of stopping if the natural completion of calculations does not require large expenditures.

The introduction of checkpoints in the algorithm may not be associated with the search for rational solutions. One of the general goals of our article is to draw attention to the advisability of using checkpoints in searching for solutions of any kind.

2 Preliminaries

2.1 Rational and polynomial solutions

Let K be a field of characteristic 0. In the sequel, the standard notations are used: the ring of polynomials in x over K and the field of rational functions are denoted by $K[x]$ and $K(x)$. The field of formal Laurent series is denote by $K((x))$. If R is a ring (in particular, a field), then the ring of $m \times m$-matrices whose entries are in R is denoted by $\mathrm{Mat}_m(R)$.

We consider systems of the form

$$A_r(x)\sigma^r y(x) + \cdots + A_1(x)\sigma y(x) + A_0(x)y(x) = 0 \tag{1}$$

where $\sigma y(x) = y(x+1)$, and $A_i(x)$, $i = 0, 1, \dots, r$, are matrices belonging to $\mathrm{Mat}_m(K[x])$; $A_r(x) \neq 0$ is the *leading* matrix (we suppose that it is non-zero), and $y(x) = (y_1(x), \dots, y_m(x))^T$ is a column of unknown functions (T denotes transposition). The number r is called the *order* of the system, we denote it by $\mathrm{ord} L$. The system under study is assumed to be of full rank; i.e., the equations of the system are linearly independent over the ring of operators $K(x)[\sigma]$.

The system (1) can be written in the form

$$L(y) = 0 \qquad (2)$$

where

$$L = A_r(x)\sigma^r + \dots + A_1(x)\sigma + A_0(x).$$

A solution $y(x) = (y_1(x), y_2(x), \dots, y_m(x))^T \in K(x)^m$ of (1) is called a *rational* solution. If $y(x) \in K[x]^m$, it is called a *polynomial* solution (a particular case of a rational solution). Below, when we talk about rational solutions of systems or about systems having rational solutions, we will mean nonzero rational solutions and, accordingly, systems having nonzero rational solutions.

Algorithms for finding rational solutions of linear difference systems of the form $\sigma y = Ay$, where A is a non-singular matrix with rational function entries, were proposed, e.g., in [5, 6, 7, 9, 10, 8, 13, 18]. (Similar questions related to differential systems were considered in [6, 11, 12].) In the book [19] the basic tools for working with scalar difference equations are described. Note that the checkpoints and tests of the considered type were not discussed in that literature, and to the best of authors' knowledge in other literature as well except mentioned [16, 17, 4].

2.2 Universal denominator

The problem of finding rational solutions of full-rank systems (1) in the case where the matrix $A_r(x)$ may be singular, was considered in [9]. An appropriate algorithm was suggested. This algorithm is based on finding a *universal denominator* of rational solutions to the original system (for brevity, we call it the universal denominator for the original system), i.e., a polynomial $U(x) \in K[x]$ such that, if the system has a rational solution $y(x) \in K(x)^m$, then it can be represented as $\frac{1}{U(x)} z(x)$, where $z(x) \in K[x]^m$. If a universal denominator is known, we can make the substitution

$$y(x) = \frac{1}{U(x)} z(x) \qquad (3)$$

where $z(x) = (z_1, (x) \dots, z_m(x))^T$ is a vector of new unknowns, and then apply one of the algorithms for finding polynomial solutions (see, e.g., [5, 13, 20]).

Other approaches are also possible. For example, the approach presented in [3] is based on expanding a general solution of the original system (1) into a series whose coefficients linearly depend on arbitrary constants. After multiplication by a universal denominator $U(x)$ the series corresponding to rational solutions turn into polynomials.

2.3 Induced operators and systems

The notions of *induced operators* and *induced systems* are quite important for finding solutions of different kinds [2, 7].

We will consider double-sided series by *factorial powers* of x, i.e., series of the form

$$s(x) = \sum_{n \in \mathbb{Z}} v_n x^{\overline{\underline{n}}}, \tag{4}$$

where

$$x^{\overline{\underline{n}}} = \begin{cases} x(x-1)\dots(x-n+1), & \text{if } n > 0, \\ 1, & \text{if } n = 0, \\ \frac{1}{(x+1)(x+2)\dots(x+|n|)}, & \text{if } n < 0. \end{cases} \tag{5}$$

In [20], it was shown that the map

$$x \to n + \sigma^{-1}, \;\; \sigma \to (n+1)\sigma + 1 \tag{6}$$

transforms an operator $L \in \operatorname{Mat}_m(K\,[x, \sigma])$ into $L^{\textcircled{i}} \in \operatorname{Mat}_m(K\,[n, \sigma, \sigma^{-1}])$

$$L^{\textcircled{i}} = B_l(n)\sigma^l + \dots + B_t(n)\sigma^t, \;\; l > \dots > t,$$

where σ is the shift of n, $B_l \neq 0$, $B_t \neq 0$, l and t are the *leading* and *trailing orders* of the operator $L^{\textcircled{i}}$. The difference $l - t$ is the *order* of $L^{\textcircled{i}}$, we denote it by $\operatorname{ord} L^{\textcircled{i}}$.

For any double-sided sequence $v = (v_n)$, application L to (4) gives the series, whose double-sided sequence of coefficients can be obtained by applying $L^{\textcircled{i}}$ to the sequence v. As a consequence, series (4) satisfies $L(y) = 0$ if and only if $L^{\textcircled{i}}(v) = 0$.

Remark 1 There exists the inverse map for (6): $n \to x - x\sigma^{-1}$, $\sigma \to (x+1)^{-1}\sigma$.

Remark 2 On can rewrite the definition (5) of the factorial power using the well known rising and falling powers $x^{\overline{n}}$, $x^{\underline{n}}$ (see, e.g., [22, Sect. 4.4, 8.3]):

$$x^{\overline{\underline{n}}} = \begin{cases} x^{\underline{n}} & \text{if } n > 0, \\ 1, & \text{if } n = 0, \\ \frac{1}{x^{\overline{-n}}}, & \text{if } n < 0. \end{cases}$$

This combination of rising and falling powers in the form of our factorial power allows to give quite a simple map (6) to transform L into $L^{\textcircled{i}}$.

2.4 Leading and trailing matrices regularization

The algorithms of EG family ([2]) allow to transform, e.g., $L^{\textcircled{i}}$ to the form, where the leading or the trailing matrix is non-singular (such transformation is referred to as *the regularizaation*). This gives, resp., the operators ${}^{+}L^{\textcircled{i}}$ and ${}^{-}L^{\textcircled{i}}$. In the process of the transformation we can obtain additionally *linear constraints* which are equal to zero linear combinations of a finite set of elements of the sequences satisfying the system. Taking these linear constraints into account excludes appearance of parasitic solutions to the transformed systems. In the sequel, we suppose ${}^{-}L^{\textcircled{i}}$

$${}^{-}L^{\textcircled{i}} = {}^{-}B_l(n)\sigma^l + \dots + {}^{-}B_t(n)\sigma^t,$$

where the trailing matrices ${}^{-}B_t(n)$ is non-singular.

Note that the regularization of leading and trailing matrices can be applied not only to the induced system. This procedure can be applied also to the original system. This is used by the dispersion algorithm [9] for computing a universal denominator (see Sect. 2.2).

2.5 Indicial polynomials and equations

We will call the determinant (a polynomial in n) of the matrix

$$ {}^{-}B_t(n-t) \tag{7}$$

as the *indicial polynomial* for L at infinity and denote this polynomial by $I_{L,\infty}(n)$. Correspondingly, $I_{L,\infty}(n) = 0$ is the *indicial equation* for L at infinity. If a series $s(x) = \sum_{n \le k} v_n x^{\overline{n}}$ represents a solution to the system $L(y) = 0$ and $v_k \ne 0$, then $k = \operatorname{val}_\infty s(x)$. This k is the *valuation* at ∞ of the series $s(x)$. It is an integer root of the algebraic equation $I_{L,\infty}(n) = 0$.

2.6 Notation

For discussing a possible scheme containing some checkpoints, it will be useful to collect together some notions and the corresponding notation in advance.

$L(y) = 0$ is an original system; $L^{\textcircled{i}}(v) = 0$ is the induced recurrence system, ${}^{-}L^{\textcircled{i}}(v) = 0$ is the result of transforming of $L^{\textcircled{i}}(v) = 0$ to the form with the non-singular trailing matrix; l and t are leading and trailing orders of the operator $L^{\textcircled{i}}$;

$I_{L,\infty}(n) = 0$ is an indicial equation for the original system, i.e., an algebraic equation $d(n-t) = 0$, where $d(n)$ is the determinant of the trailing matrix of the system ${}^{-}L^{\textcircled{i}}(v) = 0$;

n^* is the largest integer root of the indicial equation (if there is no integer root then the original system has no rational solutions;

$U(x)$ is a universal denominator for the original system.

2.7 Traditional scheme

We discuss now the traditional scheme [9] of searching for rational solutions of a system of linear difference equations with polynomial coefficients.

1. Computing a universal denominator $U(x)$ (Sect. 2.2):
   - 1a. Regularization the leading and trailing matrices of the system $L(y) = 0$ (Sect. 2.4),
   - 1b. Computing $U(x)$.
2. Substitution of (3) into the system $L(y) = 0$; transition to a system $L_U(z) = 0$ with polynomial coefficients (cleaning denominators).
3. Computing the numerator:
   - 3a. Finding a indicial equation $I_{L_U,\infty}(n) = 0$ (Sect. 2.5). Degrees of polynomial solutions are its integer roots,
   - 3b. Computing polynomial solutions of the system $L_U(z) = 0$.

The detection of the absence of rational solutions can occur at one of steps 3a, 3b. Step 3a stops if the indicial equation $I_{L_U,\infty}(n) = 0$ does not have integer non-negative roots. Otherwise the largest of such roots is used as an upper bound of the degrees of polynomial solutions in step 3b. Thus, the stop occurs when quite costly steps 1 and 2 have already been completed. We propose to change the scheme: find out in advance (before steps 1 and 2) possible obstacles to the existence candidates in the form of non-negative integers for the degree of this numerators of rational solutions. Actually, we are talking about the early determination of the indicial equation, which in the traditional scheme is at step 3a.

One way to find the indicial equations for systems is to construct induced recurrence systems $L^{ⓘ}(v) = 0$. In our case, these are series, into which rational solutions of the original system are expanded. Equality to zero of determinants of the leading and trailing matrices of the system $L^{ⓘ}(v) = 0$ (provided that these determinants are not identically equal to zero) give such kind of indicial equations. Here it should be noted that the induced systems $L^{ⓘ}(v) = 0$ and/or ${L_U}^{ⓘ}(\tilde{v}) = 0$ can be used to construct the polynomial solutions themselves after computing an upper bound for their degrees (step 3b).

The approach used below is such that the indicial equation $I_{L,\infty}(n) = 0$ is constructed for the original system $L(y) = 0$ before computing $U(x)$. In some cases this allows us to conclude that the continuation of the calculations will not lead to success. The new scheme is represented in detail in Sec. 3. It is based on some preliminary calculations, not reducible to simple checking in step 3a of the traditional scheme whether there are non-negative integers among the roots of the indicial equation.

Step 3b goes into a corresponding place of new scheme in a detailed form, including, in particular, an additional control point (see Sect. 3).

3 Scheme equipped with checkpoints and tests

The scheme equipped with checkpoints and tests is as follows (the symbol • marks a checkpoint and a test after which the algorithm can be stopped):

1. Find ${}^{-}L^{ⓘ}$ and $I_{L,\infty}(n)$ • {If $I_{L,\infty}(n)$ has no integer root then STOP.} Let n^* be the largest integer root of the polynomial $I_{L,\infty}(n)$.

2. Find a universal denominator $U(x)$. • {If $n^* + \deg U(x) < 0$, then STOP}.

   [Let $n^* < 0$. Let some intermediate stage of the computation $U(x)$ process allow to get quickly a number u such that $u \geq \deg U(x)$. • { If $n^* + u < 0$ then STOP without finishing the computation $U(x)$.}]

3. Using ${}^{-}L^{ⓘ}$ and $U(x)$ compute the polynomial $P(x)$ which is the numerator of the rational solution $P(x)/U(x)$ of the initial system. Here, finding a solution

$$R(x) = \sum_{n \leq n^*} v_n x^{\overline{n}} \tag{8}$$

   for the operator L is combined with multiplying $R(x)$ by $U(x)$; in fact, the sequence of coefficients of this product

$$U(x)R(x) = \sum_{n \leq \deg U(x) + n^*} \tilde{v}_n x^{\overline{n}} \tag{9}$$

is calculated starting from the upper bound $n^* + \deg U(x)$ for degree of $U(x)R(x)$ and finished when it is proved that $\tilde{v}_n = 0$ for $n < 0$. Then

$$P(x) = \sum_{0 \le n \le \deg U(x) + n^*} \tilde{v}_n x^{\overline{n}}.$$

When calculating the next coefficient v_n, constants can appear that look like arbitrary constants and/or some of the constants that we considered as arbitrary constants can get values. It also takes into account those linear constraints that arise during the regularization of the trailing matrix of the operator $L^{①}$. • {If in computing values of the "arbitrary constants" all the coefficients $\tilde{v}_n$ vanish for non-negative n then STOP}.

Let us comment on the checkpoints of the later scheme.

Step 1. Any rational function belonging to $K(x)$ (factually, any formal series in $\frac{1}{x}$, i.e., an element of $K((\frac{1}{x}))$ allows a representation in the form of a series in the factorial powers $x^{\overline{n}}$: it is not difficult to show that

$$\frac{1}{x^k} = \sum_{n \le -k} \gamma_{k,n} x^{\overline{n}}, \quad \gamma_{k,n} \in K, \quad \gamma_{k,-k} \ne 0. \tag{10}$$

Indeed, $\frac{1}{x^k}$ is a solution for $(x+1)^k y(x+1) - x^k y(x) = 0$. The trailing term of the induced operator for $L = (x+1)^k \sigma - x^k$ is $(n+1)\sigma^{n-k+1}$. Since $n + 1 - (n - k + 1) = k$ the claim follows (see (7)).

Formula (10) allows to prove the existence of the desired expansion of an arbitrary rational function using its Laurent series expansion (see also [15, pp. 15–17], [14, Sect. 1]). It follows also from (10) that for any $F(x), G(x) \in K((x))$ we have

$$\mathrm{val}_\infty(F(x)G(x)) = \mathrm{val}_\infty F(x) + \mathrm{val}_\infty G(x). \tag{11}$$

Step 2. When we multiply the solution by the polynomial $U(x)$ this gives by (11) a polynomial vector of degree $\alpha + \deg U(x)$, this number cannot evidently be negative.

If the dispersion algorithm is used for computing $U(x)$ (for systems of difference equations it is described in [9], where the main idea of [1] for scalar case is generalized), then the universal denominator $U(x)$ is computed as a product of some polynomial factors. Two polynomials $V(x), W(x) \in K[x]$ are preliminary computed. And then only $V(x)$ and $W(x)$ are used for $U(x)$ computing. First, an integer h called the dispersion should be found. It is seen from the algorithm description that $\deg U(x)$ does not exceed the number u that is the product of h and the smallest of degrees of V and W. The computed u is an upper bound for $\deg U(x)$.

Step 3. On this step, the approach [20] for finding polynomial solutions is used.

4 Search for the numerators

We consider in details Step 3 performing, i.e., the constructing polynomial numerators of rational solutions. Remark that it was supposed in [16, 17, 4] that if an upper bound for degrees of such numerators is computed, one can use the undefined coefficient method to find such numerators. Here another method is proposed that does not use the substitution (3). Correspondingly, no need also to apply algorithms for regularizing the trailing matrix of the induced system $L_U^{①}$.

Following Proposition we use to determine a number of coefficients $\tilde{v}_n$ of the series (9) which is sufficient on Step 3.

Proposition 1 *A rational column vector* $\frac{p(x)}{q(x)}$, $p(x) \in K[x]^m$, $q(x) \in K[x]$, *is a solution of* L *if and only if* L *has such a solution* $R(x) = \sum_{n \le n^*} v_n x^{\overline{n}}$, *that, first, the "Laurent polynomial"*

$$S(x) = q(x) \sum_{-(\mathrm{ord} L^{\textcircled{i}} + (\mathrm{ord} L + 1) \deg q(x)\,) \le n \le n^*} v_n x^{\overline{n}}$$

has zero coefficients at $x^{\overline{n}}$, $n = -1, -2, \dots, -(\mathrm{ord} L^{\textcircled{i}} + \mathrm{ord} L \deg q(x))$, *and, second, discarding in* $S(x)$ *all the terms containing* $x^{\overline{n}}$, $n < 0$, *gives* $p(x)$.

Proof: If $\frac{p(x)}{q(x)}$ is a solution of the operator L, then the series $R(x)$ can be taken as the expansion of our rational solution using the factorial powers of x. Then the first and the second conditions of the statement under proof are carried out.

Let the series $R(x)$ exist. The coefficient sequence $v = (v_n)$ of $R(x)$ is a solution of the recurrence operator $L^{\textcircled{i}}$. The multiplication of entries of $R(x)$ by the polynomial $q(x)$ gives the series

$$q(x)R(x) = \sum_{n \le \deg q(x) + n^*} v_n x^{\overline{n}}$$

which is a solution of the operator L_q and the coefficient sequence $\tilde{v} = (\tilde{v}_n)$ of which is a solution of the recurrence operator $L_q^{\textcircled{i}}$.

The substitution $y(x) = \frac{1}{q(x)} z(x)$ into a system of the form (1) gives the system

$$\frac{1}{q(x+r)} A_r(x) z(x+r) + \cdots + \frac{1}{q(x+1)} A_1(x) z(x+1) + \frac{1}{q(x)} A_0(x) z(x) = 0.$$

After cleaning denominators, we get the system $L_q(z) = 0$. The degrees of polynomial coefficients of L_q exceed the corresponding degrees of L by no more than $r \deg q(x)$. This and the formula $x \to n + \sigma^{-1}$ of transformation (6), imply

$$\mathrm{ord} L_q^{\textcircled{i}} \le \mathrm{ord} L^{\textcircled{i}} + r \deg q(x). \tag{12}$$

As above, $\mathrm{ord} L^{\textcircled{i}} = l - t$. The coefficients $\tilde{v}_n$ of the series $q(x)R(x)$ for $n = -1, -2, \dots, -(l - t + r \deg q(x))$ coincide with the coefficients of the Laurent polynomial $S(x)$ and are uniquely defined by v_n having such indexes n, for which

$$-(l - t + (r+1) \deg q(x)\,) \le n \le n^*$$

holds.

The equalities $\tilde{v}_n = 0$ for $n = -1, -2, \dots, -\mathrm{ord} L_q^{\textcircled{i}}$ guarantee that the coefficient sequence

$$\begin{cases} \tilde{v}_n, & \text{if } n \ge 0, \\ 0, & \text{if } n < 0 \end{cases} \tag{13}$$

is a solution of $L_q^{\textcircled{i}}$. From (12) it implies that the equalities $\tilde{v}_n = 0$ for $n = -1, -2, \dots, -(l - t + r \deg q(x))$ guarantee that (13) is a solution of $L_q^{\textcircled{i}}$. The sequence (13) is a coefficients sequence of the polynomial $p(x)$ that it implies $L_q(p(x)) = 0$ and it implies $L\left(\frac{p(x)}{q(x)}\right) = 0$. □

Coefficients of Laurent polynomial (8) are computed using ${}^{-}L^{\textcircled{i}} \in \mathrm{Mat}_m(K[\sigma, \sigma^{-1}, n])$ with the method from [20], that is from the highest degree of its terms (i.e. n^*) to $-((r+1)\deg U(x)+l-t)$. Let $v = (v_n)$ be a sequence that is a solution of the recurrent operator ${}^{-}L^{\textcircled{i}}$ and that satisfies the condition $\forall_{n>n^*}\, v_n = 0$.

Starting from $j = n^*$, in the case when the trailing matrix ${}^{-}L^{\textcircled{i}}$ is nonsingular for $n = j - t$, one obtains the values of the elements of the vector v_j using ${}^{-}L^{\textcircled{i}}$ and $v_{j+l-t}, \dots, v_{j+1}$.

If the trailing matrix ${}^{-}L^{\textcircled{i}}$ is singular for $n = j - t$ then one obtains the values of some of the arbitrary constants which have been earlier introduced. It is done using the system of linear algebraic equations ${}^{-}L^{\textcircled{i}}(v)\Big|_{n=j-t} = 0$ and allows refining the earlier computed coefficients $v_{n^*}, \dots, v_{j+1}$ and $\tilde{v}_{n^*+\deg U(x)}, \dots, \tilde{v}_{j+\deg U(x)+1}$. New arbitrary constants are to be introduced by that as a part of the vector v_j.

Let the values of the elements of the vector v_j be obtained that is the coefficient of the term containing $x^{\underline{j}}$ in the vector $R(x)$. The coefficient $\tilde{v}_{j+\deg U(x)}$ in the vector (9) is then computed using the induced operator $U^{\textcircled{i}} \in \mathrm{Mat}_m(K[n, \sigma^{-1}])$, whose order is $\deg U(x)$, and $v_{j+\deg U(x)}, \dots, v_{j+1}, v_j$.

The value $\tilde{v}_{j+\deg U(x)}$ is a linear combination of arbitrary constants. The coefficients of the terms of negative degree in $U(x)R(x)$ have to be zero. When $j + \deg U(x) < 0$ the system of the linear algebraic equations $\tilde{v}_{j+\deg U(x)} = 0$ is solved for the earlier introduced arbitrary constants. Using the system the values of some of the arbitrary constants may be calculated, and thus the values of earlier computed coefficients $v_{n^*}, \dots, v_j$ and $\tilde{v}_{n^*+\deg U(x)}, \dots, \tilde{v}_0$ may be refined. If $\tilde{v}_n = 0$ after that for $n = 0, 1, \dots, n^* + \deg U(x)$, i.e. $P(x) = 0$, then the computation is stopped: there is no need to compute v_n and $\tilde{v}_{n+\deg U(x)}$ for $n = -((r+1)\deg U(x) + l - t), \dots, j-1$.

In order to get ${}^{-}L^{\textcircled{i}}$ the algorithm EG (see Sect. 2.4) is applied to $L^{\textcircled{i}}$. The finite set of the linear constraints for the values of the elements of the vectors $v_{n^*}, v_{n^*-1}, \dots, v_{-((r+1)\deg U(x)+l-t)}$ may appear as well. If the computation of $\tilde{v}$ is completed and $\tilde{v} \neq 0$ then it is needed to take into account the linear constraints, i.e., to solve the corresponding system of the linear algebraic equations with respect to arbitrary constants.

5 The differential case

In this section we consider systems of the form

$$A_r(x)D^r y(x) + \cdots + A_1(x)Dy(x) + A_0(x)y(x) = 0 \tag{14}$$

where $D = \frac{d}{dx}$, and as before $A_i(x)$, $i = 0, 1, \dots, r$, are $m \times m$-matrices with entries belonging to $K[x]$. The system (14) can be written in the form (2), with

$$L = A_r(x)D^r + \cdots + A_1(x)D + A_0(x). \tag{15}$$

In the sequel, it will be useful to consider formal Laurent series, i.e., for example, elements of the field $K((x))$ (or the field $\bar{K}((x))$, where $\bar{K}$ is the algebraic closure of K). Recall that the *valuation* $\mathrm{val}\, y(x)$ of $y(x) \in K((x))$ is the minimal integer n such that the coefficient of x^n in $y(x)$ is non-zero. If $y(x)$ is the zero series then we set $\mathrm{val}\, y(x) = +\infty$. For $\alpha \in \bar{K}$, we can also consider the field $K((x-\alpha))$ of formal Laurent series in $x - \alpha$ and, correspondingly, $\mathrm{val}_{x-\alpha} t(x)$ for $t(x) \in K((x-\alpha))$.

We consider also the formal series in terms of decreasing negative powers (this can also be viewed as expansion at ∞); the field of such series is denoted by $K((x^{-1}))$. Each series of this kind contains only a finite number of powers of x with nonnegative exponents and, possibly, an infinite number of powers with negative ones. The greatest exponent of x with a nonzero coefficient occurring in a series $y(x)$ is the valuation $\mathrm{val}_\infty y(x)$. If $y(x) \in K((x^{-1}))$ is the zero series, then we set $\mathrm{val}_\infty y(x) = -\infty$.

For a vector $f(x) = (f_1(x), \ldots, f_m(x))^T \in K((x))^m$ we set $\mathrm{val}\, f(x) = \min_{i=1}^m \mathrm{val}\, f_i(x)$ (similarly for $\mathrm{val}_{x-\alpha} f(x)$). For $g(x) = (g_1(x), \ldots, g_m(x))^T \in K((x^{-1}))^m$ we set $\mathrm{val}_\infty g(x) = \max_{i=1}^m \mathrm{val}_\infty g_i(x)$. It is easy to see that $\mathrm{val}_\infty p(x) = \deg p(x)$ for a polynomial $p(x)$ and $v(\frac{f(x)}{g(x)}) = v(f(x)) - v(g(x))$ for $f(x), g(x) \in K[x]$, $v \in \{\mathrm{val}, \mathrm{val}_{x-\alpha}, \mathrm{val}_\infty\}$. It is also significant that the valuation of any type under consideration of a product is the sum of the valuations of the factors.

A rational solution of a system of the form (14) can be represented by formal Laurent series both at an arbitrary finite point α and at ∞.

It is well known (see, e.g., [10, Sect. 7.2]) that it is possible to construct for (14) a finite set of irreducible polynomials over K

$$p_1(x), \ldots, p_k(x) \tag{16}$$

such that if for some $\alpha \in \overline{K}$ there exists a solution $F \in \overline{K}((x - \alpha))^m$ such that $\mathrm{val}_{x-\alpha} F < 0$ then $p_i(\alpha) = 0$ for some $1 \leq i \leq k$, and for each $p_i(x)$ a polynomial $I_{L,p_i}(n) \in K[n]$ can be constructed such that for a solution $F \in K((x - \alpha))^m$, $p_i(\alpha) = 0$, one has $I_{L,p_i}(\mathrm{val}_{x-\alpha} F) = 0$ [10]. It is also possible to construct such a polynomial $I_{L,\infty}(n) \in K[n]$ that if a system $L(y) = 0$ has a solution $y \in K((x^{-1}))$ then $I_{L,\infty}(\mathrm{val}_\infty y(x)) = 0$. In particular, the degree of a polynomial solution is a root of $I_{L,\infty}(n)$. The polynomials $I_{L,\infty}(n), I_{L,p_1}(n), \ldots, I_{L,p_k}(n)$ are the *indicial* polynomials connected with L.

Remark 3 In the context of this paper, by the indicial polynomial for a given operator L we mean a certain polynomial, a root of which may give useful information on solutions of the initial differential system. Absence of roots of a certain type also gives information on solutions of the original differential system. Similarly to the difference case (Sect. 2.3, 2.4, 2.5), for constructing the needed polynomials we can use induced recurrence system and bring its leading or trailing matrix to non-singular form.

Proposition 2 [4, Sect. 1.3] *Let L, $p_1(x), \ldots, p_k(x)$ be as in (15), (16). Let*

$$I_{L,\infty}(n),\ I_{L,p_1}(n), \ldots, I_{L,p_k}(n)$$

be the corresponding indicial polynomials. In this case

(i) *if $I_{L,\infty}(n)$ has no integer root then the system $L(y) = 0$ has no rational solution;*

(ii) *if at least one of the polynomials $I_{L,p_1}(n), \ldots, I_{L,p_k}(n)$ has no integer root then $L(y) = 0$ has no rational solution;*

(iii) *if $b_1, \ldots, b_k \in \mathbb{Z}$ are lower bounds for integer roots of polynomials $I_{L,p_1}(n), \ldots, I_{L,p_k}(n)$ (e.g., $b_1, \ldots, b_k$ can be equal to the minimal integer roots of those polynomials), N is an upper bound for integer roots of the polynomial $I_{L,\infty}(n)$ (e.g., N can be equal to the maximal integer root of that polynomial), and $N - \sum_{i=1}^k b_i \deg p_i < 0$, then $L(y) = 0$ has no rational solution;*

(iv) *if $N - \sum_{i=1}^k b_i \deg p_i \geq 0$ (see (iii)) and the system $L(y) = 0$ has a rational solution then that solution is of the form $p_1^{b_1}(x) \ldots p_k^{b_k}(x) f(x)$, where $f(x) = (f_1(x), \ldots, f_m(x))^T \in K[x]^m$ with $\deg f_j(x) \leq N - \sum_{i=1}^k b_i \deg p_i$, $j = 1, \ldots, m$.*

This proposition is used as the basis for reconfiguring the scheme given in Sect. 3 to the differential case.

6 Implementation, experiments

The algorithm presented in the paper was implemented in Maple 2019 ([21]) as a modification of the procedure `RationalSolution` ([9, Sct.4]). The first argument of the procedure is a full rank difference or differential system. The system is specified as a linear equation with matrix coefficients. Elements of a matrix coefficient are rational functions of one variable (for example, `x`) over the rational number field. For example, the system $L(y) = 0$

$$\begin{pmatrix} x^2+102x+101 & x^3+104x^2+305x+202 \\ x^2-x-2 & x^3+x^2-4x-4 \end{pmatrix} y(x+2)+$$

$$+\begin{pmatrix} -x^2-99x+202 & -x+2 \\ x-2 & \dfrac{x-2}{x+101} \end{pmatrix} y(x+1)+\begin{pmatrix} -x-101 & -\dfrac{x+101}{x+100} \\ -x & -\dfrac{x}{x+100} \end{pmatrix} y(x) = 0$$

is represented by means of standard objects Matrix of the Maple system

```
> eq1 := Matrix([[x^2+102x+101, x^3+104x^2+305x+202],
                 [x^2-x-2, x^3+x^2-4x-4]]).y(x+2) +
         Matrix([[-x^2-99x+202, -x+2],
                 [x-2, (x-2)/(x+101)]]).y(x+1) +
         Matrix([[-x-101, -(x+101)/(x+100)],
                 [-x,  -x/(x+100)]]).y(x) = 0:
```

The second argument of the procedure is a name of a vector of unknowns (for example, `y(x)`). The third argument is optional. It is `'earlyterminate' = true` or `'earlyterminate' = false`. The default is `'earlyterminate' = true`. For `'earlyterminate' = true`, the presented algorithm with checkpoints is used. For `'earlyterminate' = false`, the algorithm from [9] is used.

If there is no rational solution, `RationalSolution` returns the empty list, i.e. [ ]:

```
> st := time():
> RationalSolution(eq1, y(x));
> time() - st;
```

[ ]

0.047

Here, 0.047 is the time taken to evaluate the result[1]. For this system, the indicial polynomial is $I_{L,\infty}(n) = -1$, there is no root. The algorithm with checkpoints stops at the first step. For this system, the time of evaluation with the argument `'earlyterminate' = false` is 3.097 sec. The algorithm from [9] finds a universal denominator $U(x)$ (here, $\deg U(x) = 205$), makes the

[1]In seconds, CPU time. Computations were carried out in Maple 2019, Ubuntu 8.04.4 LTS, AMD Athlon(tm) 64 Processor 3700+, 3GB RAM

substitution (3) in the given system, finds the indicial polynomial for the new system $L_U(z) = 0$, and stops because $I_{L_U,\infty}(n)$ has no root. For this sample, most of the time (3.023 sec) is spent building the indicial polynomial for $L_U(z) = 0$.

The next differential system has no rational solution too:

```
> eq2 := Matrix([[2, 0], [0, x^2+x]]) . diff(y(x), x$2) +
         Matrix([[1, 1], [x, x^2+5x+2]]) . diff(y(x), x) +
         Matrix([[-1, 1], [x+1,  2x+4]]) . y(x) = 0:
> RationalSolution(eq2, y(x));
```

$$[\,]$$

The indicial polynomial for this system is $n + 3$ and has the integer root $n^* = -3$, then the algorithm with checkpoints finds a universal denominator $U(x) = x(x + 1)$ and after that stops because $n^* + \deg U(x) < 0$.

If there are rational solutions, the procedure `RationalSolution` builds a basis of their linear space, returns a list of basis elements. For example:

```
> eq3 := Matrix([[0, 0], [x^3 + 5x^2 + 9x + 5, x^3 + 5x^2 + 9x + 5]]).y(x+2) +
         Matrix([[2x^2 - 2, 2(x^2 - 1)/(x + 101)],
                 [x^3 - x^2 - x + 1, (x^3 - x^2 - x + 1)/(x + 101)]]).y(x+1) +
         Matrix([[-2x^2 + 2x, -2x(x - 1)/(x + 100)],
                 [-2x^3 + x^2 - 2x - 1,
                  -(x^4 + 102x^3 + 99x^2 + 102x + 100)/(x + 100)]]).y(x):
> RationalSolution(eq3, y(x));
```

$$\left[\left[\begin{array}{c} -\dfrac{1}{(x^2+1)(x+99)} \\ \\ \dfrac{x+100}{(x^2+1)(x+99)} \end{array}\right], \left[\begin{array}{c} -\dfrac{x^3+100x^2-59600x+100}{x(x^2+1)(x+99))} \\ \\ \dfrac{(x^3-59501x^2-5960099x+100}{x(x^2+1)(x+99)} \end{array}\right]\right]$$

Our experiments show that in the absence of rational solutions, time savings are about 75%, and the additional time in the presence of solutions does not exceed 20-25%.

The implementation and a session of Maple with examples of using the procedure `RationalSolution` are available at the address http://www.ccas.ru/ca/lfs.

Acknowledgments

This work is supported in part by the Russian Foundation for Basic Research, project no. 19-01-00032.

References

[1] S.A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. *Programming and Computer Software*, 6:273–278, 1995.

[2] S.A. Abramov. EG–eliminations. *J. Diff. Equ. Appl*, 5:393–433, 1999.

[3] S.A. Abramov. Search of rational solutions to differential and difference systems by means of formal series. *Programming and Computer Software*, 41(2):65–73, 2015.

[4] S.A. Abramov. When the search for solutions can be terminated. *In the book "Algorithmic Combinatorics: Enumerative Combinatorics, Special Functions, and Computer Algebra — in Honour of Peter Paule on his 60th Birthday"* (in print).

[5] S.A. Abramov, M. Barkatou. Rational solutions of first order linear difference systems. *Proc. ISSAC'98*, 124–131, 1998.

[6] S.A. Abramov, M. Bronstein. On solutions of linear functional systems. *Proc. ISSAC'2001*, 1–6, 2001.

[7] S.A. Abramov, M. Bronstein. Linear algebra for skew-polynomial matrices. *Rapport de Recherche INRIA RR-4420, March 2002*, `https://hal.inria.fr/inria-00072168`, 2002.

[8] S.A. Abramov, A. Gheffar, D.E. Khmelnov. Factorization of polynomials and gcd computations for finding universal denominators. *Proc. CASC'2010, LNCS* 6244:4–18, 2010.

[9] S.A. Abramov, D.E. Khmelnov. Denominators of rational solutions of linear difference systems of an arbitrary order. *Programming and Computer Software*, 38(2):84–91, 2012.

[10] S.A. Abramov, D.E. Khmelnov. Linear differential and difference systems: EG_δ- and EG_σ-eliminations. *Programming and Computer Software* 39(2):91–109, 2013.

[11] M. Barkatou. A fast algorithm to compute the rational solutions of systems of linear differential equations. *RR 973–M– Mars 1997, IMAG–LMC, Grenoble*, 1997.

[12] M. Barkatou. On rational solutions of systems of linear differential equations. *J. of Symbolic Computation*, 28:547–567, 1999.

[13] M. Barkatou. Rational solutions of matrix difference equations: problem of equivalence and factorization. *Proc. ISSAC'99*, 277–282, 1999.

[14] M. Barkatou, T. Cluzeau, C. El Bacha. A new approach for computing regular solutions of linear difference systems. *Proc. CASC'2014*, 72–86, 2015.

[15] A. Duval. Équations aux différences dans le champ complexe. *PhD thesis, Université de Strasbourg*, 1984.

[16] A. Gheffar. Linear differential, difference and q-difference homogeneous equations having no rational solutions. *ACM Commun. Comput. Algebra*, 44(3):78–83, 2011.

[17] A. Gheffar. Detecting nonexistence of rational solutions of linear difference equations in early stages of computation. *ACM Commun. Comput. Algebra*, 48(3):90–97, 2015.

[18] van Hoeij M. Rational solutions of linear difference equations. *Proc. ISSAC'98*, 120–123, 1998.

[19] M. Kauers, P. Paule. The concrete tetrahedron. Symbolic sums, recurrence equations, generating functions, asymptotic estimates. *Texts and Monographs in Symbolic Computation. SpringerWienNewYork, Vienna*, 2011.

[20] D.E. Khmelnov. Search for polynomial solutions of linear functional systems by means of induced recurrences. *Programming and Computer Software*, 30:61–67, 2004.

[21] Maple online help: `http://www.maplesoft.com/support/help/`

[22] M. Petkovšek, H.S. Wilf, D. Zeilberger. $A = B$. *Peters*, 1996.