Valuations of rational solutions of linear difference equations at irreducible polynomials

A. Gheffar[a], S. Abramov[b]

[a]XLIM, Université de Limoges, CNRS, 123, Av. A. Thomas, 87060, Limoges Cedex, France
[b]Computing Centre of the Russian Academy of Sciences, ul. Vavilova, 40, Moscow 119991, GSP-1, Russia

---

Abstract

We discuss two algorithms which, given a linear difference equation with rational function coefficients over a field k of characteristic 0, construct a finite set M of polynomials, irreducible in $k[x]$, such that if the given equation has a solution $F(x) \in k(x)$ and $\mathrm{val}_{p(x)} F(x) < 0$ for an irreducible $p(x)$, then $p(x) \in M$. After this for each $p(x) \in M$ the algorithms compute a lower bound for $\mathrm{val}_{p(x)} F(x)$, which is valid for any rational function solution $F(x)$ of the initial equation. The algorithms are applicable to scalar linear equations of arbitrary orders as well as to linear systems of first-order equations.

The algorithms are based on a combination of renewed approaches used in earlier algorithms for finding a universal denominator (Abramov, Barkatou), and on a denominator bound (van Hoeij). A complexity analysis of the two proposed algorithms is presented.

Keywords: linear difference equation, rational solution, universal denominator, denominator bound, valuation
2010 MSC: 68W30, 39A06, 68W40

---

1. Introduction

Let k be a field of characteristic 0. We consider systems of the form

$$Y(x+1) = A(x)Y(x), \tag{1}$$

Email addresses: f_gheffar@yahoo.fr (A. Gheffar), sergeyabramov@mail.ru (S. Abramov)

$Y(x) = (Y_1(x), Y_2(x), \dots, Y_n(x))^T$, $A(x) = (a_{ij}(x)) \in \mathrm{Mat}_n(k(x))$. It is assumed that there exists the inverse matrix $A^{-1}(x) = (\tilde{a}_{ij}(x)) \in \mathrm{Mat}_n(k(x))$. If an inhomogeneous system $Y(x+1) = A(x)Y(x) + G(x)$ is given and $A(x)$ is as in (1), $G(x) \in k(x)^n$, then by adding to $Y(x)$ an $(n+1)$-st component with value 1, one can transform the given system into a homogeneous system with an invertible matrix belonging to $\mathrm{Mat}_{n+1}(k(x))$ (see, e.g., [12, Sect. 2.2]). For this reason we restrict our consideration to (1). At the same time we will consider scalar equations of the form

$$y(x+n) + a_{n-1}(x)y(x+n-1) + \cdots + a_1(x)y(x+1) + a_0(x)y(x) = \varphi(x), \quad (2)$$

$\varphi(x), a_1(x), \dots, a_{n-1}(x) \in k(x)$, $a_0(x) \in k(x) \setminus \{0\}$, and such an equation is inhomogeneous if $\varphi(x)$ is a non-zero rational function. By clearing denominators we can rewrite (2) as

$$b_n(x)y(x+n) + \cdots + b_1(x)y(x+1) + b_0(x)y(x) = \psi(x), \quad (3)$$

$\psi(x), b_1(x), \dots, b_{n-1}(x) \in k[x]$, $b_0(x), b_n(x) \in k[x] \setminus \{0\}$.

Currently, a few algorithms for finding rational (i.e., rational function) solutions of equations (2), (3) and systems (1) are known. The algorithms from [3, 5, 6, 10] first construct a *universal denominator*, i.e., a polynomial $U(x)$ such that in the scalar case an arbitrary rational solution $y(x)$ of (2) or (3) can be represented as

$$y(x) = \frac{z(x)}{U(x)}, \quad (4)$$

where $z(x) \in k[x]$ (in other words, if (2) has a rational solution $\frac{f(x)}{g(x)}$ which is in the lowest terms then $g(x)|U(x)$). In the case of system an arbitrary rational solution of (1) can be represented as

$$Y_i(x) = \frac{Z_i(x)}{U(x)}, \quad i = 1, 2, \dots, n, \quad (5)$$

where $Z_1(x), Z_2(x), \dots, Z_n(x) \in k[x]$. When a universal denominator is constructed, one can substitute (4), (5) with undetermined $z(x)$ resp. $Z_i(x)$ into the initial equation resp. system to reduce the problem of searching for rational solutions to the problem of searching for polynomial solutions. After this, e.g., the algorithms from [2, 7] (the scalar case) and the corresponding algorithm from [6, 10, 16] (the case of system) can be used.

The algorithm from [12] is applicable to the system (1) when $k = \mathbb{C}$. It finds n rational functions $R_1(x), R_2(x), \dots, R_n(x) \in \mathbb{C}(x)$ such that for any rational solution of (1) we have

$$Y_i(x) = Z_i(x) R_i(x), \quad i = 1, 2, \dots, n, \tag{6}$$

where $Z_1(x), Z_2(x), \dots, Z_n(x) \in \mathbb{C}[x]$ (the numerator of $R_i(x)$ is a factor of the numerator of the i-th entry $Y_i(x)$ of any rational solution $Y(x)$). The substitution (6) is used instead of (4), (5).

The approach related to [12] can lead to a more "productive" substitution. But the general situation is not so simple. We will return to this question shortly.

The algorithms from [3, 5, 6, 10] are based on computation of gcd's and do not work directly with zeros and poles of rational functions from $k(x)$ (in the general case such zeros and poles belong to an extension of k). By contrast, the algorithm from [12] first finds a finite set $\bar{S} \subset \mathbb{C}$ of candidates for poles of all possible rational solutions and then for each $c \in \bar{S}$ and $1 \leqslant i \leqslant n$ computes a lower bound for $\mathrm{val}_{x-c} Y_i(x)$ (such a bound can be a positive number in specific cases). These bounds are used to construct the rational functions $R_1(x), R_2(x), \dots, R_n(x)$.

The algorithm from [12] gives quite exact lower bounds. However it is based on matrix operations (matrix entries are in $\mathbb{C}(x)$) which are costly. The number of these operations depends on the number of elements of the set $\bar{S}$. Notice in addition that even when the entries of the matrix A belong to $\mathbb{Q}(x)$, in the general situation the algorithm requires computation with algebraic numbers.

Starting from some properties of rational solutions proven in [3, 5, 6] we show in this paper that a proper subset of $\bar{S}$ can be taken as a set of candidates for poles in a large number of cases (Proposition 4 and the ensuing example). Moreover, we use these properties of rational solutions in a general situation: instead of $\mathbb{C}$ we consider an arbitrary field k of characteristic 0, and construct a finite set M of irreducible polynomials from $k[x]$ instead of a set of candidates for poles (Section 3). No computation in extensions of k is used.

Note that besides manipulations with irreducible over $\mathbb{Q}$ polynomials the work with algebraic numbers includes some additional actions. The algorithm working directly with irreducible polynomials is preferable from the standpoint of computer algebra (in addition this version of the algorithm handles not only $\mathbb{Q}$ or $\mathbb{C}$ as a ground field, but an arbitrary field k of characteristic zero).

So the algorithm from [12] is modified in this paper (Sections 3, 4) in two directions: first, instead of complex numbers we consider irreducible polynomials from $k[x]$, and second, following results of [3, 5, 6] the set of the irreducible polynomials which are used to find lowers bounds of valuations is constructed in a specific way. We describe in details the modification $\mathbf{A}_B$ of the algorithm from [12]. A scalar version of the algorithm $\mathbf{A}_B$ is described as well.

A new version $\mathbf{A}_U$ of the algorithms from [5, 6, 10] which is based on consideration of the set M of irreducible polynomials from $k[x]$ is proposed in Section 5. We prove an exact formula for a suitable bound of the exponent of each element of M (Theorem 1). Generally speaking, in the case of system this formula provides one with a universal denominator of smaller degree than the algorithm from [10]. (The algorithm from [10] can be modified in such a way that its application to a system will give the result corresponding to the formula from Theorem 1; in our paper we consider the published version of that algorithm.)

There exist such examples when substitutions (5), (6) are identical, but the algorithm from [12], resp. the algorithm $\mathbf{A}_B$ spends much more time than the algorithms from [5, 6], resp. the algorithm $\mathbf{A}_U$. This is shown in Section 6 (Theorem 2).

It is common knowledge that "...Several algorithms in symbolic computation depend on a subroutine for finding the rational solutions of ordinary linear difference equations (OLDE), and several algorithms are known for implementing of such subroutines ..." [15]. Concerning the algorithms which depend on such a subroutine mention may be made of, e.g.,

— the algorithm for finding hypergeometric solutions of OLDE with rational coefficients and a hypergeometric right-hand side [19],

— the difference version of the Accurate summation algorithm [8], [9], which is a generalization of the well known Gosper algorithm [11],

— the algorithm for finding Liouvillian solution of second order homogeneous irreducible OLDE with rational coefficients [14], etc.

As for algorithms for implementing of such subroutines, in this paper we concentrate on the algorithms based on constructing a set of irreducible polynomials that are candidates for divisors of denominators of rational solutions, and on finding a bound for the exponent of each of these candidates. Such algorithms use the full factorization of polynomials. Note that very fast (not only theoretically) factoring algorithms are known, — see, e.g., [13]. We emphasize that only reducing the problem of finding rational solutions to the problem of finding polynomial solutions is discussed in this paper (some references on publications related to the search for polynomial solutions were mentioned above).

Acknowledgments. The authors are grateful to M. Barkatou, M. Kauers, and M. Petkovšek for interesting and helpful discussions.

2. Preliminaries

Working with polynomial and rational functions over k we will write $f(x) \perp g(x)$ for $f(x), g(x) \in k[x]$ to indicate that $f(x)$ and $g(x)$ are coprime; if $F(x) \in k(x)$, then $\operatorname{den} F(x)$ is the monic polynomial from $k[x]$ such that $F(x) = \frac{f(x)}{\operatorname{den} F(x)}$ for some $f(x) \in k[x], f(x) \perp \operatorname{den} F(x)$. In this case we write $\operatorname{num} F(x)$ for $f(x)$. The set of monic irreducible polynomials of $k[x]$ will be denoted by $\operatorname{Irr}(k[x])$. If $p(x) \in \operatorname{Irr}(k[x])$, $f(x) \in k[x]$, then we define the valuation $\operatorname{val}_{p(x)} f(x)$ as the maximal $m \in \mathbb{N}$ such that $p^m(x) | f(x)$ ($\operatorname{val}_{p(x)} 0 = \infty$), and $\operatorname{val}_{p(x)} F(x) = \operatorname{val}_{p(x)}(\operatorname{num} F(x)) - \operatorname{val}_{p(x)}(\operatorname{den} F(x))$ for $F(x) \in k(x)$.

If $p(x) \in \operatorname{Irr}(k[x])$, $f(x) \in k[x] \setminus \{0\}$ then we define the finite set

$$\mathcal{N}_{p(x)}(f(x)) = \{m \in \mathbb{Z} : p(x+m) | f(x)\}. \tag{7}$$

If $\mathcal{N}_{p(x)}(f(x)) = \varnothing$ then define $\max \mathcal{N}_{p(x)}(f(x)) = -\infty$ and $\min \mathcal{N}_{p(x)}(f(x)) = \infty$.

Let $A(x)$ be as in (1), then we define

$$\operatorname{den} A(x) = \operatorname{lcm}_{i=1}^n \operatorname{lcm}_{j=1}^n \operatorname{den} a_{ij}(x), \quad \operatorname{den} A^{-1}(x) = \operatorname{lcm}_{i=1}^n \operatorname{lcm}_{j=1}^n \operatorname{den} \tilde{a}_{ij}(x).$$

If $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T \in k(x)^n$ then $\operatorname{den} F(x) = \operatorname{lcm}_{i=1}^n \operatorname{den} F_i(x)$, and $\operatorname{val}_{p(x)} F(x) = \min_{i=1}^n \operatorname{val}_{p(x)} F_i(x)$. A solution $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T \in k(x)^n$ of (1) as well as a solution $F(x) \in k(x)$ of (2), (3) is a *rational* solution. If $\operatorname{den} F(x) \neq 1$ then this solution is *non-polynomial*, and *polynomial* otherwise.

The first computer algebra algorithm for finding solutions of (3) which belong to $k(x)$ was proposed in [3]. One of the statements proven in [3] can be formulated (using notation (7)) as follows:

Proposition 1. ([3]) *Let $F(x) \in k(x)$ satisfy (3), $p(x) \in \operatorname{Irr}(k[x])$, and $p(x) | \operatorname{den} F(x)$ (i.e., $0 \in \mathcal{N}_{p(x)}(\operatorname{den} F(x))$). Let $l =$*

$\max \mathcal{N}_{p(x)}(\operatorname{den} F(x))$, $m = \min \mathcal{N}_{p(x)}(\operatorname{den} F(x))$. *Then* $p(x+l) | b_n(x-n)$, $p(x+m) | b_0(x)$.

As a consequence, if equation (3) has a non-polynomial rational solution then $\deg \gcd(b_0(x+d), b_n(x-n)) > 0$ for some $d \in \mathbb{N}$. Indeed, let $p(x)$ be as in Proposition 1. Define $d = l - m$, then $p(x+l) = p(x+m+d)$. So $p(x+m) | b_0(x)$ and $p(x+m+d) | b_n(x-n)$.

In [6] this was generalized for system (1).

Proposition 2. ([6]) *Let* $F(x) = (F_1(x), F_2(x), \dots, F_n(x))^T \in k(x)^n$ *satisfy (1),* $p(x) \in \operatorname{Irr}(k[x])$, *and* $p(x) | \operatorname{den} F(x)$. *Let* $l = \max \mathcal{N}_{p(x)}(\operatorname{den} F(x))$, $m = \min \mathcal{N}_{p(x)}(\operatorname{den} F(x))$, $u_0(x) = \operatorname{den} A^{-1}(x)$, $u_1(x) = \operatorname{den} A(x)$. *Then* $p(x+l) | u_1(x-1)$, $p(x+m) | u_0(x)$.

As a consequence, if the system (1) has a non-polynomial rational solution then $\deg \gcd(u_1(x-1), u_0(x+d)) > 0$ for some $d \in \mathbb{N}$.

Concerning denominators, the algorithm from [6] for systems is a generalization of the algorithm from [5] for the scalar case.

Define $V(x) = b_n(x-n)$, $W(x) = b_0(x)$ for equation (3), and $V(x) = u_1(x-1)$, $W(x) = u_0(x)$ where $u_1(x) = \operatorname{den} A(x)$, $u_0(x) = \operatorname{den} A^{-1}(x)$, for system (1). Compute h as the greatest non-negative integer such that $V(x)$ and $W(x+h)$ have a nontrivial common divisor (if such integers do not exist then $h = -\infty$; in this case the initial equation has no non-polynomial rational solution). This is so called *dispersion* of $V(x)$ and $W(x)$, which denoted by $\operatorname{dis}(V(x), W(x))$ and can be computed as the largest integer root of the polynomial $\operatorname{Res}_x(V(x), W(x+m)) \in k[m]$ ([1]). The value

$$h = \operatorname{dis}(V(x), W(x)) \tag{8}$$

also can be obtained from the full factorization of $V(x)$ and $W(x)$ ([18]). This is successfully used, e.g., in Maple [20]: LREtools[dispersion].

We have to comment on our reference to [5]. The fact is that this algorithm for the scalar case firstly was published in [4], but in that publication the loop $i = 0, 1, \dots, h$ instead of $i = h, h-1, \dots, 0$ was mistakenly used (N is h in [4]). We refer to [5] where this mistake was corrected. This algorithm is exploited in current versions of Maple: LREtools[ratpolysols], LinearFunctionSystems[UniversalDenominator].

It is pertinent to note that the idea of Proposition 1 is used in [15] for constructing "aperiodic" factors of a universal denominator of rational solutions for partial linear difference equations. Note also that in [10] a more general problem than the search for rational solutions of system (1) was solved. However the algorithm from [10, Prop. 3] can be used to compute a universal denominator related to (1).

3. The set M

Let k be again a field of characteristic 0. In this section we define a finite set M such that if a given system or scalar equation has a rational non-polynomial solution, whose denominator is divisible by an irreducible polynomial $p(x) \in \mathrm{Irr}(k[x])$ then $p(x) \in M$ (the set M depends on the original system or equation).

Consider first the case of system (1). Let $u_0(x) = \operatorname{den} A^{-1}(x)$, $u_1(x) = \operatorname{den} A(x)$. We start with the finite set

$$Q = \{q_1(x), q_2(x), \ldots, q_s(x)\}, \quad s \geqslant 1, \tag{9}$$

of all elements of $\mathrm{Irr}(k[x])$ such that

$$\min \mathcal{N}_{q_t(x)}(u_0(x)) = 0, \quad \max \mathcal{N}_{q_t(x)}(u_1(x-1)) \geqslant 0, \tag{10}$$

$t = 1, 2, \ldots, s$. For each $q_t(x) \in Q$ consider

$$M_{q_t(x)} = \{q_t(x), q_t(x+1), \ldots, q_t(x+h_t)\}, \tag{11}$$

where

$$h_t = \max \mathcal{N}_{q_t(x)}(u_1(x-1)). \tag{12}$$

Define the set M:

$$M = \bigcup_{t=1}^{s} M_{q_t(x)}. \tag{13}$$

Proposition 3. *Let $F(x) \in k(x)^n$ satisfy (1), $p(x) \in \mathrm{Irr}(k[x])$, and $p(x) | \operatorname{den} F(x)$. Then $p(x) \in M$.*

Proof. As a consequence of Proposition 2 we have $\mathcal{N}_{p(x)}(u_0(x)) \neq \varnothing \neq \mathcal{N}_{p(x)}(u_1(x-1))$. Evidently $l \geqslant 0, m \leqslant 0$ for $l = \max \mathcal{N}_{p(x)}(u_1(x-1)), m = \min \mathcal{N}_{p(x)}(u_0(x))$. Take $q(x) = p(x+m)$, $h = l - m$. Then $\min \mathcal{N}_{q(x)}(u_0(x)) = 0$, $\max \mathcal{N}_{q(x)}(u_1(x-1)) = h$, $p(x) = q(x-m)$, and $0 \leqslant -m \leqslant h$. □

The algorithm from [12] starts with constructing the set S of $c \in \mathbb{C}$ for which A has a pole at c or $\det A(c) = 0$. As it is proven in [12], the finite set

$$\bar{S} = \{c \in \mathbb{C} : \exists_{c_1,\, c_2 \in S}\ c - c_1 - 1 \in \mathbb{N},\ c_2 - c \in \mathbb{N}\}. \tag{14}$$

is such that if a rational solution of the system has a pole at $c \in \mathbb{C}$ then $c \in \bar{S}$.

To have analogy with the algorithm from [12] (and in particular with the above formula (14)) we define the set $S_{k[x]}$ of polynomials $p(x) \in \mathrm{Irr}(k[x])$ such that $p(x) |\, \mathrm{den}\, A(x)$ or $p(x) |\, \mathrm{num}(\det A(x))$, and the set

$$\bar{S}_{k[x]} = \{p \in \mathrm{Irr}(k[x]) : \exists_{p_1, p_2 \in S_{k[x]},\, l, m \in \mathbb{N}}\ p(x+l+1) = p_1(x), p(x-m) = p_2(x)\}$$

(if $k = \mathbb{C}$ and $\bar{S} = \{c_1, c_2, \dots\}$ then $\bar{S}_{k[x]} = \{x - c_1, x - c_2, \dots\}$). Using the same reasoning as in [12] it is not difficult to prove that if $F(x) \in k(x)^n$ satisfies (1), $p(x) \in \mathrm{Irr}(k[x])$, and $p(x) |\, \mathrm{den}\, F(x)$ then $p(x) \in \bar{S}_{k[x]}$. Now we compare the sets $\bar{S}_{k[x]}$ and M.

Proposition 4. $M \subseteq \bar{S}_{k[x]}$.

Proof. Let $M_{q_t(x)}$ of the form (11) be one of the sets belonging to the right-hand side of (13). Prove that

$$q_t(x), q_t(x+1), \dots, q_t(x+h_t) \in \bar{S}_{k[x]}. \tag{15}$$

Notice that by (10), (12) we have

$$q_t(x) |\, \mathrm{den}\, A^{-1}(x), \tag{16}$$

and

$$q_t(x + h_t + 1) |\, \mathrm{den}\, A(x). \tag{17}$$

The relation (17) implies that $q_t(x + h_t + 1) \in S_{k[x]}$.

Consider relation (16). We have

$$A^{-1}(x) = \frac{1}{\det A(x)} \cdot C^T(x),$$

where $C(x)$ is the matrix of cofactors. Each cofactor is a determinant of order $n - 1$, whose entries are some of entries of $A(x)$. So the denominator of each cofactor divides $(\operatorname{den} A(x))^{n-1}$. Therefore the denominator of any of entries of $A^{-1}(x)$ divides the product

$$(\operatorname{num} \det A(x)) \cdot (\operatorname{den} A(x))^{n-1}.$$

Since $q_t(x)$ is irreducible, it follows from (16) that at least one of the relations

$$q_t(x) | \operatorname{num}(\det A(x)), \quad q_t(x) | \operatorname{den} A(x)$$

is valid (in the case $n = 1$ the first relation is valid). This gives $q_t(x) \in S_{k[x]}$.

So $q_t(x), q_t(x + h_t + 1) \in S_{k[x]}$, and (15) is proven. □

However M and $\bar{S}_{k[x]}$ do not coincide in some cases. For example, let $k = \mathbb{C}$, m be a positive integer, and

$$A(x) = \begin{pmatrix} \frac{x+m}{x(x-m)} & 0 \\ 0 & \frac{x+m}{x(x-m)} \end{pmatrix}.$$

In this case

$$A^{-1}(x) = \begin{pmatrix} \frac{x(x-m)}{x+m} & 0 \\ 0 & \frac{x(x-m)}{x+m} \end{pmatrix},$$

$\det A(x) = \frac{(x+m)^2}{x^2(x-m)^2}$, $\operatorname{den} A(x) = x(x-m)$, $\operatorname{den} A^{-1}(x) = x + m$, and

$$S = \{-m, 0, m\}, \quad \bar{S} = \{-m+1, -m+2, \dots, 0, 1, \dots, m\},$$

$$S_{k[x]} = \{x+m, x, x-m\}, \quad \bar{S}_{k[x]} = \{x+m-1, x+m-2, \dots, x, x-1, \dots, x-m\}.$$

But $M = \varnothing$, and by Proposition 3 the system has no non-polynomial rational solution. We do not need substitution (6).

For the scalar case the set M can be constructed similarly, taking $b_0(x), b_n(x-n)$ instead of $u_0(x), u_1(x-1)$.

4. Algorithm $\mathbf{A}_{\boldsymbol{B}}$

Following [12] define $A_N(x) = A(x-1)A(x-2)\dots A(x-N)$ for each $N \in \mathbb{N}$. Then

$$Y(x) = A_N(x)Y(x-N) \tag{18}$$

for each solution $Y(x)$ of (1). As we have mentioned in Section 1, the algorithm from [12] is applicable to a system of the form (1) when $k = \mathbb{C}$. Let the set $\bar{S}$ be as in (14). For each $c \in \bar{S}$ the algorithm takes $N \in \mathbb{N}$ such that $c - N \notin \bar{S}$. If $Y(x) = (Y_1(x), Y_2(x), \dots, Y_n(x))^T$ is a rational solution of (1) then $\mathrm{val}_{x-c} Y_i(x - N) \geqslant 0$, $i = 1, 2, \dots, n$. An investigation of the valuation at $x - c$ of entries of $A_N(x)$ gives some lower bounds (the *left-hand* bounds) for

$$\mathrm{val}_{x-c} Y_i(x), \quad i = 1, 2, \dots, n. \tag{19}$$

Now let N be such that $c + N \notin \bar{S}$. The matrix A is invertible and we get

$$Y(x) = A_{-N}(x) Y(x + N), \tag{20}$$

where $A_{-N} = A^{-1}(x) A^{-1}(x + 1) \dots A^{-1}(x + N - 1)$. This also gives some lower bounds (the *right-hand* bounds) for (19). For each i the algorithm takes the maximum of two bounds.

Below we describe a generalization of the algorithm from [12] for the case of an arbitrary field k of characteristic 0 (we also will use the set M instead of $\bar{S}$, $\bar{S}_{k[x]}$).

Let $p(x) \in \mathrm{Irr}(k[x])$, N be a positive integer, $1 \leqslant i \leqslant n$. Define $B(p(x), N, i)$ as the minimum of the valuations at $p(x)$ of the entries in the i-th row of $A_N(x)$. Then the inequality $\mathrm{val}_{p(x)} Y_i(x - N) \geqslant 0$, $i = 1, 2, \dots, n$, implies

$$\mathrm{val}_{p(x)} Y_i(x) \geqslant B(p(x), N, i), \quad i = 1, 2, \dots, n. \tag{21}$$

In the same way we can use a matrix of the form $A_{-N}(x)$ (see (20)). For positive integer N define $B(p(x), -N, i)$ as the minimum of the valuations at $p(x)$ of the entries in the i-th row of $A_{-N}(x)$. If $N \in \mathbb{N}$ is such that $\mathrm{val}_{p(x)} Y_i(x + N) \geqslant 0$, $i = 1, 2, \dots, n$ then

$$\mathrm{val}_{p(x)} Y_i(x) \geqslant B(p(x), -N, i), \quad i = 1, 2, \dots, n. \tag{22}$$

Similarly to [12], bounds (21) are the left-hand bounds, while (22) are the right-hand bounds.

Let the set Q be as in (9) and $M_{q_t(x)}$ be as in (11), $t = 1, 2, \dots, s$. Let

$$h = \max\{h_1, h_2, \dots, h_t\}. \tag{23}$$

The algorithm is as follows.

Computing successively matrices $A_N(x)$ for $N = 1, 2, \dots, h + 1$ we find for each t such that $1 \leqslant t \leqslant s$ and $h_t \geqslant N - 1$ the values

$$B(q_t(x + h_t - N + 1), N, i), \quad i = 1, 2, \dots, n,$$

which give us left-hand bounds for $\mathrm{val}_{q_t(x+h_t-N+1)} Y_i(x)$, $i = 1, 2, \dots, n$. Analogously we compute successively matrices $A_{-N}(x)$ for $N = 1, 2, \dots, h+1$ and find for each t such that $1 \leqslant t \leqslant s$ and $h_t \geqslant N - 1$ the values

$$B(q_t(x + N - 1), -N, i), \quad i = 1, 2, \dots, n,$$

which give us right-hand lower bounds for $\mathrm{val}_{q_t(x+N-1)} Y_i(x)$, $i = 1, 2, \dots, n$. We have two lower bounds for each of the valuations $\mathrm{val}_{q_t(x+j)} Y_i(x)$, $i = 1, 2, \dots, n$, $t = 1, 2, \dots, s$, $j = 0, 1, \dots, h_t$, and can take the maximal one, we denote it by $\alpha_{i,j,t}$. The rational functions

$$R_i(x) = \prod_{\substack{1 \leqslant t \leqslant s \\ 0 \leqslant j \leqslant h_t}} q_t^{\alpha_{i,j,t}}(x + j), \quad i = 1, 2, \dots, n,$$

are used in the substitution (6).

In [12] the algorithm is described only for systems of the form (1). Scalar equations (2), (3) with zero right-hand side are assumed ([12, Sect. 3]) to be transformed to the system with the companion matrix $A(x)$ of the initial scalar equation. But the matrix operations are quite costly. In addition it is not difficult to give a scalar version of the algorithm. We describe this version assuming again that the ground field is an arbitrary field k of characteristic 0.

First we show that for an arbitrary positive integer N we can construct an equation

$$y(x) = v_{N,\,n-1}(x) y(x - N) + \dots + v_{N,\,0}(x) y(x - N - n + 1) + v_{N,\,-1}(x) \quad (24)$$

with $v_{N,\,-1}(x), v_{N,0}(x), \dots, v_{N,n-1}(x) \in k(x)$, which is satisfied by all rational solutions of (2) and (3). Indeed, for $N = 1$ we have the equation

$$y(x) = -a_{n-1}(x - n) y(x - 1) - \dots - a_0(x - n) y(x - n) + \varphi(x - n), \quad (25)$$

which is a consequence of (2). We can define $v_{1,\,-1}(x) = \varphi(x - n)$, and $v_{1,i}(x) = -a_i(x - n)$, $i = 0, 1, \dots, n - 1$. Suppose inductively that equation (25) is constructed for some $N \geqslant 1$. Then we can get the corresponding equation for $N + 1$ using the equality

$$y(x - N) = -a_{n-1}(x - N - n) y(x - N - 1) - \dots$$
$$\dots - a_0(x - N - n) y(x - N - n) + \varphi(x - N - n) \quad (26)$$

for eliminating $y(x-N)$ in the right-hand side of (24).

Similarly to (24) for an arbitrary positive integer N we can construct an equation

$$y(x) = w_{N,\,n-1}(x)y(x+N) + \cdots + w_{N,\,0}(x)y(x+N+n-1) + w_{N,\,-1}(x) \quad (27)$$

with $w_{N,\,-1}(x), w_{N,0}(x), \ldots, w_{N,n-1}(x) \in k(x)$, which is satisfied by all rational solutions of (2) and (3). First, since $a_0(x)$ in (2) is non-zero, we can rewrite this equation as

$$y(x) = c_1(x)y(x+1) + c_2(x)y(x+2) + \cdots + c_n(x)y(x+n) + \chi(x),$$

with $c_1(x), c_2(x), \ldots, c_n(x), \chi(x) \in k(x)$. Therefore for $N = 1$ we can define $w_{1,\,-1}(x) = \chi(x)$, and $w_{1,i}(x) = c_{n-i}(x)$, $i = 0, 1, \ldots, n-1$. Suppose inductively that equation (27) is constructed for some $N \geqslant 1$. Then we can get the corresponding equation for $N+1$ using the equality

$$y(x+N) = c_1(x+N)y(x+N+1) + \cdots + c_n(x+N)y(x+N+n) + \chi(x+N)$$

for eliminating $y(x+N)$ in the right-hand side of (27).

Equations (24), (27) are analogs of equations (18), (20).

Let $p(x) \in \operatorname{Irr}(k[x])$, N be a positive integer. Define $B(p(x), N)$ as the minimum of the valuations at $p(x)$ of the coefficients $v_{N,\,-1}(x), v_{N,\,0}(x), \ldots, v_{N,n-1}(x)$ in (24), and $B(p(x), -N)$ as the minimum of the valuations at $p(x)$ of the coefficients $w_{N,\,-1}(x), w_{N,\,0}(x), \ldots, w_{N,n-1}(x)$ in (27).

Consider the set M defined for (2) by (13). Let the equalities (9), (11) and (23) be valid. The algorithm for the scalar case is as follows.

Constructing successively equations (24) for $N = 1, 2, \ldots, h+1$ we find for each t such that $1 \leqslant t \leqslant s$ and $h_t \geqslant N-1$ the value $B(q_t(x+h_t-N+1), N)$ which gives us the left-hand bound for $\operatorname{val}_{q_t(x+h_t-N+1)} y(x)$. Similarly we construct successively equations (27) for $N = 1, 2, \ldots, h+1$ and find for each t such that $1 \leqslant t \leqslant s$ and $h_t \geqslant N-1$ the value $B(q_t(x+N-1), -N)$, which gives us right-hand lower bounds for $\operatorname{val}_{q_t(x+N-1)} y(x)$. We have two lower bounds for each of the valuations $\operatorname{val}_{q_t(x+j)} y(x)$, $t = 1, 2, \ldots, s$, $j = 0, 1, \ldots, h_t$, and can take the maximal one, we denote it by $\beta_{j,t}$. The rational function

$$R(x) = \prod_{\substack{1 \leqslant t \leqslant s \\ 0 \leqslant j \leqslant h_t}} q_t^{\beta_{j,t}}(x+j)$$

is used for the substitution $y(x) = z(x)R(x)$ into (2).

We will refer to the given modification (for both system and scalar cases) of the algorithm from [12] as $\mathbf{A}_B$. Thus the novelty of this algorithm as compared with the algorithm from [12] consists in considering irreducible polynomials over k instead of complex numbers, and the set M instead of $\bar{S}$, $\bar{S}_{k[x]}$.

5. Algorithm $\mathbf{A}_U$

The algorithms from [5, 6, 10] for constructing universal denominators use a gcd computations instead of the full factorization of polynomials (but note that these algorithms use a polynomial dispersion computation; we mentioned in Section 2 that currently Maple uses the full polynomial factorization for this). The algorithm given below is represented in the same style as the algorithm $\mathbf{A}_B$.

Theorem 1. *Let $V(x) = b_n(x-n)$, $W(x) = b_0(x)$ for equation (3), and $V(x) = u_1(x-1)$, $W(x) = u_0(x)$ where $u_1(x) = \operatorname{den} A(x)$, $u_0(x) = \operatorname{den} A^{-1}(x)$ for system (1). Let $F(x)$ be a rational solution of equation (3) or system (1). Then*

$$\operatorname{val}_{p(x)} F(x) \geqslant -\min\left\{\sum_{i\in\mathbb{N}} \operatorname{val}_{p(x+i)} V(x),\ \sum_{i\in\mathbb{N}} \operatorname{val}_{p(x-i)} W(x)\right\} \tag{28}$$

for any $p(x) \in \operatorname{Irr}(k[x])$.

Proof. Let $h = \operatorname{dis}(V(x), W(x))$ and $N = h+1$ in (18), (20). Both polynomials $\operatorname{den} A_{h+1}(x)$, $\operatorname{den} A_{-h-1}(x))$ are universal denominators for system (1). Obviously $\operatorname{den} A_{h+1}(x) \mid W(x)W(x+1)\ldots W(x+h)$ and $\operatorname{den} A_{-h-1}(x) \mid V(x)V(x-1)\ldots V(x-h)$. Therefore $\operatorname{den} F(x) \mid V(x)V(x-1)\ldots V(x-h)$ and $\operatorname{den} F(x) \mid W(x)W(x+1)\ldots W(x+h)$. (In the scalar case we get similar relations considering a system $Y(x+1) = A(x)Y(x)$ with the companion matrix.) The inequality (28) follows from the fact that

$$\operatorname{val}_{p(x)} \prod_{i=0}^{h} V(x-i) = \sum_{i\in\mathbb{N}} \operatorname{val}_{p(x)} V(x-i),$$

$$\mathrm{val}_{p(x)} \prod_{i=0}^{h} W(x+i) = \sum_{i \in \mathbb{N}} \mathrm{val}_{p(x)} W(x+i).$$

□

Therefore if for each $q_t(x+j) \in M$ (Section 3) we compute

$$\gamma_{j,t} = \min \left\{ \sum_{i \in \mathbb{N}} \mathrm{val}_{q_t(x+j+i)} V(x),\ \sum_{i \in \mathbb{N}} \mathrm{val}_{q_t(x+j-i)} W(x) \right\}$$

then we get the universal denominator

$$\prod_{\substack{1 \leqslant t \leqslant s \\ 0 \leqslant j \leqslant h_t}} q_t^{\gamma_{j,t}}(x+j).$$

We will refer to this algorithm as algorithm $\mathbf{A}_U$. The novelty of this algorithm as compared with the algorithms from [5, 6, 10] consists in considering the set M and the corresponding exponents of its elements instead of the dispersion and the gcd's.

Remark 1. *Let $d_i(x) = \mathrm{den}\, a_i(x)$, where $a_i(x)$ is the i-th row of the matrix $A(x)$, $i = 1, 2, \ldots, n$. Let*

$$D(x) = \mathrm{diag}(d_1(x), d_2(x), \ldots, d_n(x)). \tag{29}$$

The polynomials $\det D(x)$, $\det(D(x)A(x))$ are used in [10] instead of $\mathrm{den}\, A(x), \mathrm{den}\, A^{-1}(x)$ for constructing a universal denominator in the case of system (1). Let $v_0(x) = \det(D(x)A(x))$, $v_1(x) = \det D(x)$, where $D(x)$ is as in (29). It is proven in [10] that $\gcd\left(\prod_{i=0}^{h} v_0(x-i),\ \prod_{i=0}^{h} v_1(x+i)\right)$ is a universal denominator. This can be used for another proof of Theorem 1. The scalar case of Theorem 1 follows also from [5, Th. 2].

In [10] two following statements were proven

1. *If equation (3) has a solution $F(x) \in k(x)$ and $m \in \mathbb{N}$ is such that $b_n(x-n) \perp b_0(x+l)$ for any integer $l > m$, then $\mathrm{den}\, F(x) | \prod_{i=0}^{m} b_n(x-n-i)$ and $\mathrm{den}\, F(x) | \prod_{i=0}^{m} b_0(x+i)$.*

2. *If the system (1) has a solution $F(x) \in k(x)^n$, $v_0(x) = \det(D(x)A(x))$, $v_1(x) = \det D(x)$, where $D(x)$ is as in (29), and $m \in \mathbb{N}$ is such that $v_1(x-1-l) \perp v_0(x)$ for any integer $l > m$, then $\mathrm{den}\, F(x) | \prod_{i=0}^{m} v_1(x-1-i)$ and $\mathrm{den}\, F(x) | \prod_{i=0}^{m} v_0(x+i)$.*

The second statement can be strengthened.

Proposition 5. *Let the system (1) have a solution* $F(x) \in k(x)^n$. *Let* $u_0(x) = \operatorname{den} A^{-1}(x)$, $u_1(x) = \operatorname{den} A(x)$, *and* $m \in \mathbb{N}$ *be such that* $u_1(x-1) \perp u_0(x+l)$ *for any integer* $l > m$. *Then* $\operatorname{den} F(x) | \prod_{i=0}^{m} u_1(x-1-i)$ *and* $\operatorname{den} F(x) | \prod_{i=0}^{m} u_0(x+i)$.

Proof follows from Theorem 1. □

It is well known that if a differential system $Y'(x) = A(x)Y(x)$ with $A(x) \in \operatorname{Mat}_n(k(x))$ has a rational solution $F(x)$ then $\operatorname{den} F(x) | (\operatorname{den} A(x))^m$ for all integer values of m large enough. Proposition 5 gives a difference analog of this.

It is not difficult to show that $\operatorname{den} A(x) | \det D(x)$, $\operatorname{den} A^{-1}(x) | \det(D(x)A(x))$ where $D(x)$ is as in (29). In some cases the strong inequalities $\deg(\operatorname{den} A(x)) < \deg(\det D(x))$, $\deg(\operatorname{den} A^{-1}(x)) < \deg(\det(D(x)A(x)))$ are valid and the reason for this is obvious: $\det D(x) = d_1(x)d_2(x)\dots d_n(x)$, whereas $\operatorname{den} A(x) = \operatorname{lcm}(d_1(x), d_2(x), \dots, d_n(x))$. If, e.g., $A(x)$ of order n is

$$A(x) = \operatorname{diag}\left(\frac{x(x+1)}{(x+3)(x+4)}, \frac{x(x+1)}{(x+3)(x+4)}, \dots, \frac{x(x+1)}{(x+3)(x+4)}\right), \quad (30)$$

then $D(x) = \operatorname{diag}((x+3)(x+4), (x+3)(x+4), \dots, (x+3)(x+4))$, $\det D(x) = (x+3)^n(x+4)^n$, while $\operatorname{den} A(x) = (x+3)(x+4)$; similarly $\det(D(x)A(x)) = x^n(x+1)^n$, while $\operatorname{den} A^{-1}(x) = x(x+1)$. For the system $Y(x+1) = A(x)Y(x)$ with $A(x)$ as in (30) we get the universal denominator $x(x+1)^2(x+2)^2(x+3)$ using $\mathbf{A}_U$ as well as the algorithm from [6], while the universal denominator computed by the algorithm from [10] is, resp. $x^n(x+1)^{2n}(x+2)^{2n}(x+3)^n$. But as we mentioned in the Introduction, the algorithm from [10] allows a modification to avoid this excessiveness.

6. Complexity analysis

We have noticed that the algorithm from [12] often gives quite exact lower bounds. By Theorem 1 from [12] for $k = \mathbb{C}$ these bounds are even sharp if the system (1) has a fundamental solution matrix which consists of rational functions. In some sense, the fact that this algorithm requires a significant amount of time is vindicated by decreasing the degrees of polynomial solutions of the equation that appears after the corresponding substitution into the initial system (equation). But as we show below, the situation is not so simple.

We will compare the algorithms $\mathbf{A}_B$, $\mathbf{A}_U$ as they were described above. Each of these algorithms achieves some "speed-up" in certain situations (concerning the original versions, see the last paragraph of [12, Sect. 2.1], and [6, Sect. 3.3], respectively; observe that after this "speed-up", Theorem 1 from [12] no longer applies to the algorithm of that paper).

First we consider the scalar case. If equation (3) of order n is such that $\max\{\deg b_0(x), \deg b_1(x), \ldots, \deg b_n(x), \psi(x)\} = l$ and the set M defined by (13) consists of m elements then the triple (l, m, n) will be called the *combined size* of the equation.

The process of applying each of the algorithms $\mathbf{A}_B$, $\mathbf{A}_U$ can be divided into two steps. In the first step, each of these algorithms constructs the set M. In the second step, the algorithms compute the exponents $\beta_{j,t}$ (the algorithm $\mathbf{A}_B$) and $\gamma_{j,t}$ (the algorithm $\mathbf{A}_U$) of the factors $q_t(x+j) \in M$. The cost of the first step is the same for both algorithms. We will consider the number of exponents which must be computed to be the cost of the second step of $\mathbf{A}_U$ (therefore this cost is equal to m; thus we suppose that all $\gamma_{j,t}$ are computed independently, although many of them may be equal). As for $\mathbf{A}_B$, the cost of the second step is m plus the cost of constructing all needed equations (24), (27). Therefore the difference $T_B(l, m, n) - T_U(l, m, n)$ of the complexities of $\mathbf{A}_B$ and $\mathbf{A}_U$ can be considered as the cost of constructing these equations in the worst case. We see that the number of such equations is maximal when $m = h + 1$, where h is the dispersion corresponding to the original equation. Therefore in the worst case we have

$$M = \{q(x), q(x+1), \ldots, q(x+m-1)\} \tag{31}$$

where $q(x) \in \mathrm{Irr}(k[x])$.

In the next theorem we use the Ω-notation which is very common in complexity theory ([17]). Unlike O-notation which is used for describing upper asymptotical bounds, the Ω-notation is used for describing lower asymptotical bounds.

Theorem 2. *Let l, m, n be positive integer numbers. In this case*

(i) for the worst-case complexities $T_U(l, m, n)$ and $T_B(l, m, n)$ of the algorithms $\mathbf{A}_B$ and $\mathbf{A}_U$ the difference $T_U(l, m, n) - T_B(l, m, n)$ is non-negative and $T_U(l, m, n) - T_B(l, m, n) = \Omega(lmn)$;

(ii) there exists a homogeneous equation $E_{l,m,n}$ of combined size (l, m, n) such that (31) is valid for this equation, and applying $\mathbf{A}_B$ to $E_{l,m,n}$ results in a rational function $R(x) = \frac{1}{g(x)}$, $g(x) \in k[x]$, such that $U(x)|g(x)$, where $U(x)$ is the result of applying $\mathbf{A}_U$ to $E_{l,m,n}$.

Proof. (i) Let, e.g., the equation (24) be constructed for some $1 \leqslant N < m$. Then constructing the equation (24) for $N+1$ requires in the worst case more than nl field operations in k. We have to construct such equations for $N = 1, 2, \dots, m$.

(ii) First consider the case $n = 1$ and define $E_{l,m,1}$ as

$$x^l y(x+1) - (x-m)^l y(x) = 0. \tag{32}$$

This equation has the rational solution

$$F(x) = \frac{1}{((x-1)(x-2)\dots(x-m))^l}.$$

We define $E_{l,m,n}$ for an arbitrary $n > 1$ as

$$(x+n-1)^l y(x+n) + \sum_{i=1}^{n-1} (2x+2i-m-1)^l y(x+i) - (x-m)^l y(x) = 0. \tag{33}$$

This equation is satisfied by $F(x)$. Indeed, let ϕ be the shift operator: $\phi(y(x)) = y(x+1)$. Then the operator $x^l\phi - (x-m)^l$ corresponds to equation (32). If we left-multiply this operator by $\phi^{n-1} + \phi^{n-2} + \dots + 1$ then we get the operator which correspond to (33). So $F(x)$ satisfies (33). Therefore, if applying $\mathbf{A}_B$ to $E_{l,m,n}$ we obtain $R(x) \in k(x)$, then

$$\operatorname{num} R(x) = 1, \quad \operatorname{den} F(x) | \operatorname{den} R(x). \tag{34}$$

The set M for (33) is as for (32), i.e. $\{x-1, x-2, \dots, x-m\}$. It is easy to check that $\mathbf{A}_U$ gives the denominator of $F(x)$. □

Informally speaking, for any combined size (l, m, n) there exists such a “bad" equation $E_{l,m,n}$ for which $\mathbf{A}_B$ spends a large amount of time (the maximal for the given combined size!) but the output is not better than the output of $\mathbf{A}_U$, and $\mathbf{A}_U$ spends a much smaller amount of time on this equation.

The case of system is analogous (we can transform any scalar equation into a system, using the companion matrix).

Remark 2. *It is easy to see that $T_U(l,m,n)$ in contrast to $T_B(l,m,n)$ does not depend on n (factually if we define the complexities $T'_U(l,m)$, $T'_U(l,m)$ then $T'_U(l,m) = T_U(l,m,n)$, while $T'_U(l,m) = \infty$ for all non-negative l,m). In addition $T_U(l,m,n)$ will not be changed if we define $l = \min\{\deg b_0(x), \deg b_n(x)\}$, while $T_B(l,m,n)$ will be equal to ∞ for all $n > 1$.*

References

[1] S. Abramov, On the summation of rational functions, USSR Comput. Math. Math. Phys. 11 (1971), 324–330 . Transl. from Zh. Vychisl. Mat. Mat. Fiz. 11 (1971), 1071–1075.

[2] S. Abramov, Problems of computer algebra involved in the search for polynomial solutions of linear differential and difference equations. *Moscow Univ. Comput. Math. Cybernet.*, No 3, 63–68 (1989).

[3] S. Abramov, Rational solutions of linear difference and differential equations with polynomial coefficients, Comput. Math. Math. Phys. 29 (1989), 7–12. Transl. from Zh. Vychisl. Mat. Mat. Fiz. 29 (1989), 1611–1620.

[4] S. Abramov, Rational solutions of linear difference and q-difference equations with polynomial coefficients, Proc. ISSAC'95 (1995), 303–308.

[5] S. Abramov, Rational solutions of linear difference and q-difference equations with polynomial coefficients. Program. Comput. Softw. 21 (1995), 273–278. Transl. from Programmirovanie 6 (1995), 3–11.

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

[7] S. Abramov, M. Bronstein, M. Petkovšek, On polynomial solutions of linear operator equations, Proc. ISSAC'95 (1995), 290–295.

[8] S. Abramov, M. van Hoeij, A method for the integration of solutions of Ore equations, Proc. ISSAC'97 (1997), 172–175.

[9] S. Abramov, M. van Hoeij, Integration of solutions of linear functional equations, Integral Transforms Spec. Funct. 8 (1999), 3–12 .

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

[11] R.W. Gosper, Decision procedure for indefinite hypergeometric summation, Proc. Natl. Acad. Sci. USA 75 (1978), 40–42.

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

[13] M. van Hoeij, Factoring polynomials and the knapsack problem, J. Number Theory 95 (2002), 167–189.

[14] M. van Hoeij, G. Levy, Liouvillian solutions of irreducible second order linear difference equations, Proc. ISSAC'10 (2010), 297–301.

[15] M. Kauers, C. Schneider, Partial denominator bounds for partial linear differential equations, Proc. ISSAC'10 (2010), 211–218.

[16] D.E. Khmelnov, Search for polynomial solutions of linear functional systems by means of induced recurrences, Program. Comput. Softw. 30 (2004), 61–67. Transl. from Programmirovanie 2 (2004), 8-16.

[17] D.E. Knuth, Big omicron and big omega and big theta, ACM SIGACT News 8 (1976), 18–23 .

[18] Y.K. Man, F.J. Wright, Fast polynomial dispersion computation and its application to indefinite summation, Proc. ISSAC'94 (1994), 175–180.

[19] M. Petkovšek, Hypergeometric solutions of linear recurrences with polynomial coefficients, J. Symbolic Comput. 14 (1992), 243–264.

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