AAECC (2016) 27:105–122
DOI 10.1007/s00200-015-0269-0

ORIGINAL PAPER

On the complexity of skew arithmetic

Joris van der Hoeven[1]

Received: 4 September 2012 / Revised: 23 June 2015 / Accepted: 25 June 2015 /
Published online: 18 August 2015

Abstract In this paper, we study the complexity of several basic operations on linear differential operators with polynomial coefficients. As in the case of ordinary polynomials, we show that these complexities can be expressed almost linearly in terms of the cost of multiplication.

Keywords Linear differential operators · Algorithm · Complexity · Multiplication · Local solution · Division · gcrd · lclm

Mathematics Subject Classification 68W30 · 68Q15 · 34M03 · 12E15

1 Introduction

Let $\mathbb{K}$ be an effective field of constants of characteristic zero, so that all field operations can be carried out by algorithms. Given an indeterminate x and the derivation $\delta = x\partial$, where $\partial = \partial/\partial x$, it is well known [8,12,20,22,23] that the skew polynomial ring $\mathbb{K}(x)[\delta]$ behaves very much like an ordinary polynomial ring: there are skew analogues for each of the classical operations of division with remainder, greatest common divisors, least common multiples, etc. In this paper, we will study the complexity of these operations. For this purpose, it will be more appropriate to work in the ring $\mathbb{K}[x, \delta]$ instead of $\mathbb{K}(x)[\delta]$. In analogy with the commutative case, we will

This work has been supported by the ANR-09-JCJC-0098-01 MAGIX Project, as well as a Digiteo 2009-36HD grant and Région Ile-de-France.

✉ Joris van der Hoeven
vdhoeven@lix.polytechnique.fr
http://lix.polytechnique.fr/~vdhoeven

[1] LIX, CNRS, École Polytechnique, 91128 Palaiseau Cedex, France

give bounds for the computational complexities of the various operations in terms of the complexity of operator multiplication.

For our complexity measures, we make the customary assumption that all field operations in K can be carried out in constant time $O(1)$. We will try to express the complexities of our algorithms in terms of the following standard complexities:

- The time $\mathsf{M}(n)$ required for the multiplication of two polynomials of degrees $<n$ and coefficients in $\mathbb{K}$. It is classical [9,25,26] that $\mathsf{M}(n) = O(n \log n \log\log n)$ and $\mathsf{M}(n) = O(n \log n)$ if $\mathbb{K}$ admits sufficiently many 2^p-th roots of unity [10].
- The complexity $O(r^\omega)$ of multiplying two $r \times r$ matrices with entries in $\mathbb{K}$. It is classical [11,18,24,28] that $\omega < 2.376$, although $\omega \approx 3$ in practice.

We will denote by $\mathbb{K}[x]_n$ the subset of $\mathbb{K}[x]$ of polynomials of degree $< n$. Likewise, we denote by $\mathbb{K}[x, \delta]_{n,r}$ the set of operators $L \in \mathbb{K}[x, \delta]$ of degree $\deg_x L < n$ in x and degree $\deg_\delta L < r$ in δ.

Now consider two linear differential operators $K, L \in \mathbb{K}[x, \delta]_{n,r}$. We start with studying the following complexities:

- The complexity $\mathsf{SM}(n, r)$ of multiplying K and L.
- The complexity $\mathsf{SV}(n, r)$ of applying L to a vector of r polynomials in $\mathbb{K}[x]_n$.
- The cost $\mathsf{SF}(n, r)$ to compute a fundamental system of r solutions to the monic equation $(\delta^r + L) f = 0$ in $\mathbb{K}[[x]]$, up to order $O(x^n)$, while assuming the existence of such a fundamental system.
- Given a vector V of r truncated power series in $\mathbb{K}[x]$, the cost $\mathsf{SA}(n, r)$ of computing a monic operator in $A = \delta^r + \mathbb{K}[x, \delta]_{n,r}$ with $A(V) = O(x^n)$.

The special case $n = r$ was first studied in [30], where it was shown that $\mathsf{SM}(n, n) = O(n^\omega)$, using evaluation–interpolation techniques. The inverse bound $n^\omega = O(\mathsf{SM}(n, n))$ has been proved in [5]; this paper also contains detailed information on the constant factors involved in these bounds. Recently (and after the writing of a first version of this paper), the quasi-optimal bound $\mathsf{SM}(n, r) = \tilde{O}(nr(nr)^{\omega-2})$ was proved in [2].

For fixed constants $\alpha, \beta > 0$, one has $\mathsf{M}(\alpha n) = O(\mathsf{M}(n))$, $(\beta r)^\omega = O(r^\omega)$, $\mathsf{SM}(\alpha n, \beta r) = O(\mathsf{SM}(n, r))$, etc., by splitting the multiplicands in a finite number of pieces. In this paper, we will freely use this remark without further mention. In order to simplify our complexity estimates, it will be convenient to make a few additional assumptions. First of all, we will assume that $\omega > 2$, whence in particular $\mathsf{M}(n) \log n = O(n^{\omega-1})$. We will also assume that the function $\mathsf{M}(n)/n$ is increasing and that $\mathsf{SM}(n, r)/(nr)$ is increasing in both n and r. This will indeed be the case for the complexity bounds for $\mathsf{SM}(n, r)$ that will be given in Sect. 2.

In Sect. 2, we will first prove (see Theorems 1 and 2) that the problems of multiplication and operator–vector application are essentially equivalent when $n \geqslant r$. We also recall the best existing bounds for operator multiplication.

In Sect. 3, we show that the problems of computing fundamental systems of solutions and its inverse can be reduced to operator multiplication modulo a logarithmic overhead (see Theorems 5 and 6). This provides a dual way to perform operations on differential operators by working on their fundamental systems of solutions. In Sect. 3 and all subsequent sections, we always assume that $n \geqslant r$. This is indeed required for

the truncations of the fundamental systems of solutions at order $O(x^n)$ to be linearly independent.

In Sect. 4, we start with the operations of exact right division and right division with remainder. In Sect. 5, we consider greatest common right divisors (gcrds) and least common left multiples (lclms). Again, we will show how to express the complexities of these operations essentially in terms of the complexity $\mathsf{SM}(n, r)$ of multiplication (see Theorems 7, 8, 9 and 10).

For several of our algorithms, we need to work at a point where certain operators are non singular. If we only need the input operators to be non singular, then it is easy to find a point where this is the case. If we also need the output operators or certain auxiliary operators to be non singular (as in Sect. 5), then we resort to picking random points, which are non singular with probability 1. In Sect. 5.2 we present additional techniques for turning algorithms which rely on random point picking into randomized algorithms of Las Vegas type and into fully deterministic algorithms.

For technical reasons, we found it convenient to work with respect to the Euler derivation δ instead of ∂. Nevertheless, operators L in $\mathbb{K}[x, \delta]$ can be converted efficiently into operators in $\mathbb{K}[x, \partial]$ and *vice versa*, modulo an increase of the degree n in x with the degree r in δ or ∂ (see Lemma 2). Using our assumption that $n \geqslant r$, such increases of the degree n by r only gives rise to constant overheads in the complexity bounds. Hence, the complexity bounds for our main algorithms from Sects. 3, 4 and 5 still hold when replacing δ by ∂. In addition, some of the algorithms can be adapted to directly use ∂ instead of δ, without the need for any conversions (see Remark 3).

To the best of our knowledge, the idea to perform operations on linear differential operators *via* power series solutions was first proposed (but only partially worked out) in [4, Chapter 10]. In this paper, we use a slightly different technique: instead of a single power series solution, we prefer to consider a fundamental system of solutions. This has the advantage of forcing a clean bijection between operators and solution spaces, thereby avoiding part of the randomness in the proposals from [4, Chapter 10].

It is also possible to mimic classical divide and conquer algorithms for right division, greatest common right divisors and least common left multiples, while using adjoints in order to perform the recursive operations on the appropriate side. Such algorithms were partially implemented inside MATHEMAGIX [34] and we plan to analyze this technique in more details in a forthcoming paper.

Various complexity results for computations with linear differential operators and other skew polynomials were previously obtained [4,13–16,19]. Especially the computation of greatest common right divisors and least common left multiples of two or more operators has received particular attention. After the publication of a first version of this paper [33], the complexities of several classical algorithms [17,19,27] for the computation of least common right multiples were studied in great detail in [6], and new improvements were proposed there.

The complexities of most of the algorithms in this paper are stated in terms of the input *and* output sizes. The uncertified randomized algorithms for gcrds and lclms are optimal up to logarithmic factors from this perspective, which yields an improvement with respect to the previously known complexity bounds. In the context of certified randomized algorithms (i.e. of Las Vegas type), the complexity bounds remain quasi-

optimal in terms of the size of a suitable certificate. From the deterministic point of view, the new algorithms for gcrds and lclms are suboptimal.

2 Evaluation and interpolation

The key argument behind the proof from [30] that $\mathsf{SM}(n,n) = O(n^\omega)$ is the observation that an operator $L \in \mathbb{K}[x,\delta]_{n,r}$ is uniquely determined by its images on the vector $x^{;r} = (1,\ldots,x^{r-1})$. This makes it possible to use a similar evaluation–interpolation strategy for the multiplication of differential operators as in the case of FFT-multiplication of commutative polynomials. More precisely, given $L \in \mathbb{K}[x,\delta]_{n,r}$, let $\Phi_L^{r+n,r}$ be the matrix of the mapping $\mathbb{K}[x]_r \to \mathbb{K}[x]_{r+n}$; $P \mapsto L(P)$ with respect to the bases $x^{;r}$ and $x^{;r+n}$:

$$\Phi_L^{r+n,r} = \begin{pmatrix} L(1)_0 & \cdots & L(x^{r-1})_0 \\ \vdots & & \vdots \\ L(1)_{r+n-1} & \cdots & L(x^{r-1})_{r+n-1} \end{pmatrix}.$$

The evaluation and interpolation steps can be done efficiently using the following lemma, which is essentially contained in [5]:

Lemma 1 *Let $L \in \mathbb{K}[x,\delta]_{n,r}$. Then*

(a) We may compute $\Phi_L^{r+n,r}$ as a function of L in time $O(n\mathsf{M}(r)\log r)$.

(b) We may recover L from $\Phi_L^{r+n,r}$ in time $O(n\mathsf{M}(r)\log r)$.

Proof Consider the expansion of L with respect to x

$$L(x,\delta) = L_0(\delta) + \cdots + x^{n-1}L_{n-1}(\delta).$$

For all i, j, we have

$$\begin{aligned} L(x,\delta)(x^j)_{i+j} &= [x^i L_i(\delta)](x^j)_{i+j} \\ &= [x^{i+j} L_i(\delta+j)(1)]_{i+j} \\ &= L_i(j). \end{aligned}$$

In other words, $\Phi_L^{r+n,r}$ is a lower triangular band matrix

$$\Phi_L^{r+n,r} = \begin{pmatrix} L_0(0) & & \\ \vdots & \ddots & \\ L_{n-1}(0) & & L_0(r-1) \\ & \ddots & \vdots \\ & & L_{n-1}(r-1) \end{pmatrix}$$

of bandwidth $\leqslant n$. The coefficients on the i-th subdiagonal of $\Phi_L^{r+n,r}$ are exactly the result of a multipoint evaluation of L_i at $0,\ldots,r-1$. It is classical [3,21,29] that both

multipoint evaluation and the inverse operation of interpolation can be performed in time $\mathcal{O}(\mathsf{M}(r)\log r)$. Doing this for each of the polynomials $L_0, \ldots, L_{n-1}$ yields the result. □

Theorem 1 *If $n \geqslant r$, then*

$$\mathsf{SM}(n,r) = O(\mathsf{SV}(n,r) + n\mathsf{M}(r)\log r) \tag{1}$$

Proof Let $K, L \in \mathbb{K}[x,\delta]_{n,r}$ and assume that we want to compute KL. We may evaluate $L(x^{:2r})$ in time $\mathsf{SV}(\max(n,2r),2r) = O(\mathsf{SV}(n,r))$. We may also evaluate $K(L(x^{:2r}))$ in time $\mathsf{SV}(n+2r,2r) = O(\mathsf{SV}(n,r))$. Using Lemma 1, we may recover KL from $K(L(x^{:2r}))$ in time $O(n\mathsf{M}(r)\log r)$. This completes the proof. □

Theorem 2 *If $n \geqslant r$, then*

$$\mathsf{SV}(n,r) = O(\mathsf{SM}(n,r) + n\mathsf{M}(r)\log r). \tag{2}$$

Proof Assume now that we are given $K(x,\delta) \in \mathbb{K}[x,\delta]_{n,r}$, as well as a vector $V = (V_0, \ldots, V_{r-1}) \in \mathbb{K}[x]_n^r$ and that we want to evaluate $K(V) = (K(V_0), \ldots, K(V_{r-1}))$. This is equivalent to evaluating the operator $K^* = K(x, \delta - r)$ at the vector $x^r V$. It is classical [1] that K^* can be computed in time $O(n\mathsf{M}(r))$. Using Lemma 1, we may compute the unique operator $L \in \mathbb{K}[x,\delta]_{n+r,r}$ with $L(x^{:r}) = x^r V$ in time $O((n+r)\mathsf{M}(r)\log r) = O(n\mathsf{M}(r)\log r)$. We may next compute the product K^*L in time $\mathsf{SM}(n+r,r) = O(\mathsf{SM}(n,r))$. Lemma 1 finally allows us to evaluate K^*L at $x^{:r}$ in time $O(n\mathsf{M}(r)\log r)$, thereby yielding $K(V)$. □

The above results immediately imply the bound $\mathsf{SM}(n,n) = O(n^\omega)$ from [30] by the computation of a product KL to the computation of a matrix product

$$\Phi_{KL}^{2r+2n,2r} = \Phi_K^{2r+2n,2r+n}\Phi_L^{2r+n,2r}.$$

After the publication of a first version of this paper, the following quasi-optimal bound for $\mathsf{SM}(n,r)$ was established in [2, Theorems 3 and 5].

Theorem 3 *(i) For $r \geqslant n$, we have $\mathsf{SM}(n,r) = O\left(n^{\omega-1}r + n\mathsf{M}(r)\log r\right)$.*
(ii) For $n \geqslant r$, we have $\mathsf{SM}(n,r) = O\left(r^{\omega-1}n + r\mathsf{M}(n)\log n\right)$.

The inverse bound $n^\omega = O(\mathsf{SM}(n,n))$ from [5] can also be generalized:

Theorem 4 *If $n \geqslant r$, then the product of an $r \times n$ matrix and an $r \times r$ matrix with coefficients in $\mathbb{K}$ can be computed in time $O(\mathsf{SM}(n,r))$.*

Proof By the result from [5], the problem is equivalent to the computation of $k = \lceil n/r \rceil$ operators $K_0, \ldots, K_{k-1}$ in $\mathbb{K}[x,\delta]_{r,r}$ with a fixed operator $L \in \mathbb{K}[x,\delta]_{r,r}$. Setting $K = K_0 + x^{2r}K_1 + \cdots + x^{2r(k-1)}K_{k-1}$, we may compute KL in time $O(\mathsf{SM}(n,r))$. We may directly read off the products $K_0L, \ldots, K_{k-1}L$ from the result. □

In this paper, we have chosen to work with respect to the derivation δ instead of ∂. The following result from [5, Section 3.3] can be used to efficiently convert between operators in $\mathbb{K}[x, \delta]$ and $\mathbb{K}[x, \partial]$ (in [30], we proved a somewhat weaker result which would also suffice for the purposes of this paper). We have written $K[x, \partial]_{n,r}$ for the set of operators of degree $< n$ in x and degree $< r$ in ∂.

Lemma 2 *(a) Any operator in $\mathbb{K}[x, \delta]_{n,r}$ can be converted into an operator in $\mathbb{K}[x, \partial]_{n+r,r}$ in time $O((n + r)\mathsf{M}(r) \log r)$.*
(b) Any operator in $x^r \mathbb{K}[x, \partial]_{n,r}$ can be converted into an operator in $\mathbb{K}[x, \delta]_{n+r,r}$ in time $O((n + r)\mathsf{M}(r) \log r)$.

3 Local solutions

From now on, we will assume that $n \geqslant r$. We recall that an operator $L \in \mathbb{K}[x, \partial]$ of order r is said to be *non singular* at x_0, if its leading coefficient L_r does not vanish at x_0. We will say that an operator $L \in \mathbb{K}[x, \delta]$ of order r is non singular (at the origin) if $x^{-r} L \in \mathbb{K}[x, \partial]$ and $x^{-r} L$ is non singular as an operator in ∂.

Given a non singular differential operator $L \in \mathbb{K}[x, \delta]_{n,r+1}$ of order r, the equation $L(H) = 0$ admits a *canonical* fundamental system $H = (H_0, \ldots, H_{r-1})$ of solutions in $\mathbb{K}[[x]]^r$, with the property that $(H_i)_i = 1$ and $(H_i)_j = 0$ for all $i, j < r$ with $i \neq j$. Conversely, given a $\mathbb{K}$-linearly independent vector of power series $H \in \mathbb{K}[[x]]^r$, there exists a unique monic operator $L \in \delta^r + \mathbb{K}[[x]][\delta]$ of order r with $L(H) = 0$. Let us show how to convert efficiently between these two representations.

Theorem 5 *Let $L \in \mathbb{K}[x, \delta]_{n,r+1}$ be a differential operator of order $r \leqslant n$, which is non singular at the origin, and let H be its canonical fundamental system of solutions. Then we may compute H up to order $O(x^n)$ in time $O(\mathsf{SV}(n, r) \log n)$. In other words,*

$$\mathsf{SF}(n, r) = O(\mathsf{SM}(n, r) \log n). \tag{3}$$

Proof Modulo multiplying L on the left by L_r^{-1}, we may assume without loss of generality that L is monic. Since L is non singular at the origin, we have $x^{-r} L \in \mathbb{K}[x, \partial]$. Rewritten in terms of δ, this means that L is of the form

$$\begin{aligned} L &= \Delta_r(\delta) + x C_{r-1} \Delta_{r-1}(\delta) + \cdots + x^r C_0 \Delta_0(\delta). \\ \Delta_k(\delta) &= \delta(\delta - 1) \cdots (\delta - k + 1), \end{aligned}$$

for certain $C_0, \ldots, C_{r-1} \in \mathbb{K}[x]$. Setting $R = \Delta_r(\delta) - L \in x\mathbb{K}[x, \delta]_{n-1,r}$, we observe that R maps $\mathbb{K}[[x]]$ into $x^r \mathbb{K}[[x]]$. We now compute H using the "recursive" formula

$$H = \begin{pmatrix} 1 \\ \vdots \\ x^{r-1} \end{pmatrix} + \Delta_r(\delta)^{-1}(R(H)), \tag{4}$$

where

$$\Delta_r(\delta)^{-1}\left(\sum_{k\geqslant r} A_k x^k\right) = \sum_{k\geqslant r} \frac{A_k}{\Delta_r(k)} x^k.$$

Equation (4) is a schoolbook example for applying the strategy of relaxed resolution of power series equations [31,32]. Since $\Delta_r(\delta)^{-1}$ operates coefficientwise, it can be computed in linear time. The main cost of the computation therefore reduces to the relaxed evaluation of $R(H)$. Using fast relaxed multiplication, this amounts to a cost

$$\mathsf{SF}(n,r) = 2\mathsf{SV}(\lceil n/2\rceil, r) + 4\mathsf{SV}(\lceil n/4\rceil, r) + \cdots + n\mathsf{SV}(1,r).$$

Using the monotonicity assumption and Theorem 2, the result follows. □

In what follows, given a non zero series Y in x, we denote by $v(Y)$ its valuation. Given a vector V of elements in a $\mathbb{K}$-vector space, we will also denote by $\mathrm{Vect}(V)$ the subvector space generated by the entries of V, and

$$v^{\max}(V) = \max\{v(Y) : Y \in \mathrm{Vect}(V)\backslash\{0\}\}.$$

Notice that $v^{\max}(V) \geqslant \dim(\mathrm{Vect}(V)) - 1$.

Theorem 6 *Let $H = (H_0, \ldots, H_{r-1}) \in \mathbb{K}[[x]]^r$ be $\mathbb{K}$-linearly independent. Then there exists a unique monic operator $L = \mathrm{ann}(H) \in \delta^r + \mathbb{K}[[x]][\delta]_r$ with $L(H) = 0$. Moreover, given the truncation of H at order $O(x^n)$, we may compute L at order $O(x^{n-v^{\max}(H)})$ in time $O(\mathsf{SM}(n,r)\log r)$. In other words,*

$$\mathsf{SA}(n,r) = O(\mathsf{SM}(n,r)\log r). \tag{5}$$

Proof Modulo a triangularization of H, we may assume without loss of generality that $v(H_0) < \cdots < v(H_{r-1}) = v^{\max}(H)$. We define operators $L^{[0]}, \ldots, L^{[r]}$ by

$$\begin{aligned} L^{[0]} &= 1\\ L^{[i+1]} &= \left(\delta - \frac{\delta L^{[i]}(H_i)}{L^{[i]}(H_i)}\right) L^{[i]}. \end{aligned}$$

Then $L = L^{[r]}$ annihilates H and for any other operator $\tilde{L} \in \delta^r + \mathbb{K}[[x]][\delta]_r$ with $\tilde{L}(H) = 0$, we would have $(\tilde{L} - L)(H) = 0$, which is in contradiction with the fact that $\dim\ker(\tilde{L} - L) < r$. Moreover, by induction over i, we observe that the coefficient of x^0 in $L^{[i]}$ is given by $(\delta - v(H_0))\cdots(\delta - v(H_{i-1}))$ and the coefficients of $x^0, \ldots, x^{n-1}$ in $L^{[i]}$ can be expressed in terms of the coefficients of $x^0, \ldots, x^{n-1+v(H_{i-1})}$ in $H_0, \ldots, H_{i-1}$. In particular, the truncation of L at order $O(x^{n-v^{\max}(H)})$ is uniquely determined by the truncation of H at order $O(x^n)$.

In order to explicitly compute L up to a given order, it is more efficient to use a divide and conquer approach. More precisely, given $H \in (H_0, \ldots, H_{r-1}) \in \mathbb{K}[x]_n^r$ we compute $\mathrm{ann}_n(H) \in \delta^r + \mathbb{K}[x,\delta]_{n,r}$ using the following method:

– If $r = 1$, then we take $\mathrm{ann}_n(H) = \delta - (\delta H_0/H_0) \bmod x^n$.
– Otherwise, let $r = a + b$ with $a = \lceil r/2 \rceil$.
– Compute $A := \mathrm{ann}_n(H_0, \ldots, H_{a-1})$.
– Evaluate $I := (A(H_a), \ldots, A(H_{r-1})) \bmod x^n$.
– Compute $B := \mathrm{ann}_n(I_0, \ldots, I_{b-1})$.
– Return $L = BA \bmod x^n$.

If $n > v^{\max}(H)$, then it is easy to check that $\mathrm{ann}_n(H)(H) = O(x^{n-v^{\max}(H)})$. For a fixed constant C, we thus have

$$\mathsf{SA}(n, 2r) \leqslant 2\mathsf{SA}(n, r) + C\mathsf{SM}(n, r).$$

The result now follows from the monotonicity assumption. □

Remark 1 If $\mathsf{SM}(n, r)/r^{1+\epsilon}$ is increasing in r for some $\epsilon > 0$, then the bound further simplifies to $\mathsf{SA}(n, r) = O(\mathsf{SM}(n, r))$.

Remark 2 We notice that the operator L in Theorem 6 is singular if and only if $v^{\max}(H) = r - 1$, and if and only if $\{v(Y) : Y \in \mathrm{Vect}(H)\backslash\{0\}\} = \{0, \ldots, r - 1\}$.

Remark 3 The algorithm from the proof can be adapted so as produce a vanishing operator in $x^r\partial^r + \mathbb{K}[[x]][\partial]_r$ instead of $\delta^r + \mathbb{K}[[x]][\delta]_r$. Indeed, for this, it suffices to take

$$L^{[i+1]} = x\left(\partial - \frac{\partial L^{[i]}(H_i)}{L^{[i]}(H_i)}\right) L^{[i]},$$

and carefully adapt the truncation orders. □

Although a general operator $L \in \mathbb{K}[x, \delta]$ can be singular at the origin, many operations on operators (such as right division and greatest common right divisors) commute with translations $x \mapsto x + x_0$, and Lemma 2 may be used in conjunction with the following lemma in order to reduce to the case when L is non singular at the origin.

Lemma 3 *Given a non zero operator $L \in \mathbb{K}[x, \partial]_{n,r}$, we may find a point $x_0 \in \mathbb{K}$ where L is non singular in time $O(\mathsf{M}(n))$.*

Proof Let L_k be the leading coefficient of L. Since $\deg_x L_k < n$, we have $L_k(x_0) \neq 0$ for some $x_0 \in \{0, \ldots, n\}$. Using fast multipoint evaluation [7], we may find such a point x_0 in time $O(\mathsf{M}(n))$. □

4 Right division

Both the degrees in x and δ are additive for the multiplication of operators $K, L \in \mathbb{K}[x, \delta]$. In particular, if $K, L \in \mathbb{K}[x, \delta]_{n,r}$ and L is left or right divisible by K, then the quotient is again in $\mathbb{K}[x, \delta]_{n,r}$.

Theorem 7 *Let $K, L \in \mathbb{K}[x,\delta]_{n,r}$ be such that $L = QK$ for some $Q \in \mathbb{K}[x,\delta]$ and $n \geqslant r$. Then we may compute Q in time $O(\mathsf{SM}(n,r)\log n)$.*

Proof By Lemmas 2 and 3, and modulo a shift $x \mapsto x + x_0$, we may assume without loss of generality that K and L are non singular at the origin. We now use the following algorithm:

- We first compute the canonical fundamental system of solutions H to $L(H) = 0$ up to order $O(x^{n+r})$. By Theorem 5, this can be done in time $O(\mathsf{SM}(n,r)\log n)$.
- We next evaluate $I = K(H)$ and compute a $\mathbb{K}$-basis G for $\mathrm{Vect}(I)$ at order $O(x^{n+r})$. This can be done in time $O(\mathsf{SM}(n,r))$, by Theorems 2 and 4, and using linear algebra. Since K is non singular, we have $v(Y) \geqslant \deg_\delta K \Rightarrow v(K(Y)) = v(Y)$ for all $Y \in \mathbb{K}[[x]]$. In particular, the $\deg_\delta Q = \deg_\delta L - \deg_\delta K$ elements of H of valuations $\deg_\delta K, \ldots, \deg_\delta L - 1$ are mapped to set which spans a vector space of dimension $\deg_\delta Q$. This shows that $s = \dim(\mathrm{Vect}(I) \bmod x^r) = \deg_\delta Q$.
- We now compute the monic annihilator $\Omega = \mathrm{ann}(G)$ of G at order $O(x^n)$. This can be done in time $O(\mathsf{SM}(n,r)\log r) = O(\mathsf{SM}(n,r)\log n)$, by Theorem 6.
- We return the truncation of $Q_s\Omega$ at order $O(x^n)$, where $Q_s = L_{\deg_\delta L}/K_{\deg_\delta K}$.

Since each of the steps can be carried out in time $O(\mathsf{SM}(n,r)\log n)$, the result follows. □

It is classical that euclidean division generalizes to the skew polynomial ring $\mathbb{K}(x)[\delta]$. In other words, given operators $A, B \in \mathbb{K}(x)[\delta]$ where $B \neq 0$, there exist unique operators $Q = \mathrm{quo}(A,B)$ and $R = \mathrm{rem}(A,B)$ in $\mathbb{K}(x)[\delta]$ with

$$A = QB + R,$$

and $\deg_\delta R < \deg_\delta B$. If $A, B \in \mathbb{K}[x,\delta]$ and I is the leading term of B with respect to δ, then left multiplication of A by $I^{\deg_\delta A - \deg_\delta B + 1}$ allows us to remain in the domain $\mathbb{K}[x,\delta]$: there exist unique $Q = \mathrm{pquo}(A,B)$ and $R = \mathrm{prem}(A,B)$ in $\mathbb{K}[x,\delta]$ with

$$I^{\deg_\delta A - \deg_\delta B + 1} A = QB + R, \tag{6}$$

and $\deg_\delta R < \deg_\delta B$. The operators Q and R are usually called pseudo-quotients and pseudo-remainders. In some cases, a non trivial polynomial can be factored out in the relation (6). Let J be monic, of maximal degree, such that $J^{-1}QB$, $J^{-1}R \in \mathbb{K}[x,\delta]$. Then we call $J^{-1}Q = \mathrm{quo}^*(A,B)$ and $J^{-1}R = \mathrm{rem}^*(A,B)$ the "simplified" pseudo-quotient and pseudo-remainder of A and B.

Lemma 4 *Let $H = (H_0, \ldots, H_{r-1}) \in \mathbb{K}[[x]]^r$ be $\mathbb{K}$-linearly independent and define $p = v^{\max}(\mathrm{Vect}(H)) + 1$. Given $G \in (x^p\mathbb{K}[[x]])^r$, there exists a unique operator $L \in \mathbb{K}[[x]][\delta]_r$ of order $< r$ with $L(H) = G$ and we may compute its first n terms with respect to x in time $O(\mathsf{SM}(n+p,r)\log n)$.*

Proof Let $\alpha_i = v(H_i)$ for each i. Modulo a base change, we may assume without loss of generality that $\alpha_0 < \cdots < \alpha_{r-1}$. Let $\Phi : \mathbb{K}[[x]]^r \to \mathbb{K}[[x]]^r$ be the operator with

$$\Phi(V_0, \ldots, V_{r-1}) = (x^{\alpha_0}V_0, \ldots, x^{\alpha_{r-1}}V_{r-1}),$$

and let Φ^{-1} denote the inverse operator. Let $\Psi : \mathbb{K}[[x]][\delta]_r \to \mathbb{K}[[x]]^r$ be the operator with

$$\Psi(K) = \Phi^{-1}(K(\Phi(1))).$$

Writing $K = \sum_{i,k} K_{i,k} x^k \delta^i$ and $\Psi(K)_{i,k} = (\Psi(K)_i)_k$, we have

$$\begin{pmatrix} \Psi(K)_{0,k} \\ \vdots \\ \Psi(K)_{r-1,k} \end{pmatrix} = \begin{pmatrix} 1 & k+\alpha_0 & \cdots & (k+\alpha_0)^{r-1} \\ \vdots & \vdots & & \vdots \\ 1 & k+\alpha_{r-1} & \cdots & (k+\alpha_{r-1})^{r-1} \end{pmatrix} \begin{pmatrix} K_{0,k} \\ \vdots \\ K_{r-1,k} \end{pmatrix}.$$

In other words, Ψ and its inverse Ψ^{-1} operate coefficientwise and n coefficients can be computed in time $O(r^\omega n)$.

Putting $H_i = x^{\alpha_i} + E_i$ with $E_i = o(x^{\alpha_i})$ for each i, we may rewrite the equation $L(H) = G$ as

$$L = \Psi^{-1}(\Phi^{-1}(G - L(E)))$$

and we observe that the coefficient of x^k in the righthand side only depends on earlier coefficients of $1, \ldots, x^{k-1}$ in L. In particular, we may solve the equation using a relaxed algorithm. Then the main cost is concentrated in the relaxed evaluation of $L(E)$. As in the proof of Theorem 5, this evaluation can be done in time $O(\mathsf{SM}(n + p, r) \log n)$. □

Theorem 8 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ with $n \geqslant r$ and $s = \deg_\delta K > 0$. Right pseudo-division of L by K and simplification yields a relation*

$$AL = QK + R,$$

where $A, Q = \mathrm{quo}^*(L, K), R = \mathrm{rem}^*(L, K) \in \mathbb{K}[x, \delta]$. If $n' \geqslant n$ is such that $A, Q, R \in \mathbb{K}[x, \delta]_{n',r}$, then A, Q and R can be computed in time $O(\mathsf{SM}(n', r) \log n')$.

Proof Modulo a shift $x \mapsto x + x_0$, we may assume without loss of generality that K and L are non singular at the origin. We now use the following algorithm:

- We compute the canonical fundamental system H of solutions to $K(H) = 0$ up to order $O(x^{2n'+r})$. This requires a time $O(\mathsf{SM}(n', s) \log n')$.
- We compute $G = L(H)$ with $R(H) = AG$ up to order $O(x^{2n'+r})$. This requires a time $O(\mathsf{SM}(n', r))$.
- We determine the operator $\Omega \in \mathbb{K}[[x]][\delta]_s$ with $\Omega(H) = x^s G$ up to order $O(x^{2n'+r})$. Lemma 4 shows that this can be done in time $O(\mathsf{M}(n', s) \log n')$.
- By Theorem 6, we have $R = x^{-s} A\Omega$ and $x^{-s}\Omega$ is known up to order $O(x^{2n'})$. Now $x^{-s}\Omega_0, \ldots, x^{-s}\Omega_{s-1}$ are truncated rational functions, for which the degrees of the numerators and denominators are bounded by n'. Using rational function reconstruction [35], we may thus compute $N_k / D_k = x^{-s}\Omega_k$ with $\gcd(N_k, D_k) = 1$ in time $s\,O(\mathsf{M}(n) \log n)$. Taking $A = \mathrm{lcm}(D_0, \ldots, D_{s-1})$, we find R.

– Once A and R are known, we compute Q using the algorithm from Theorem 7.

The total complexity of this algorithm is bounded by $O(\mathsf{SM}(n', r) \log n')$. □

Remark 4 In the above proof, we have assumed that n' is known beforehand. In general, we may still apply the above algorithm for a trial value n^*. Then the algorithm may either fail (for instance, if $\deg \operatorname{lcm}(D_0, \ldots, D_{s-1}) \geqslant n^*$), or return the triple (A, Q, R) under the assumption that $A, Q, R \in \mathbb{K}[x, \delta]_{n^*, r}$. We may then check whether the triple is correct in time $O(\mathsf{SM}(n^*, r))$. Applying this procedure for successive guesses $n^* = n, 2n, 4n, \ldots$, the algorithm ultimately succeeds for an n^* with $n^* \leqslant 2n'$. Using the monotonicity hypothesis, the total running time thus remains bounded by $O(\mathsf{SM}(n^*, r) \log n^*) = O(\mathsf{SM}(n', r) \log n')$.

5 Euclidean operations

5.1 Randomized algorithms

It is classical that greatest common right divisors and least common left multiples exist in the skew euclidean domain $\mathbb{K}(x)[\delta]$: given two operators $K, L \in \mathbb{K}(x)[\delta]$, the greatest common right divisor $\Gamma = \operatorname{gcrd}(K, L)$ and the least common left multiple $\Lambda = \operatorname{lclm}(K, L)$ are the unique monic operators with

$$\mathbb{K}(x)[\delta]\Gamma = \mathbb{K}(x)[\delta]K + \mathbb{K}(x)[\delta]L$$
$$\mathbb{K}(x)[\delta]\Lambda = \mathbb{K}(x)[\delta]K \cap \mathbb{K}(x)[\delta]L.$$

Assume now that $K, L \in \mathbb{K}[x, \delta]$ and let A and B be monic polynomials of minimal degrees, such that $A\Gamma$ and $B\Lambda$ are in $\mathbb{K}[x, \delta]$. Then we call $\Gamma^* = \operatorname{gcrd}^*(K, L) = A\Gamma$ and $\Lambda^* = \operatorname{lclm}^*(K, L) = B\Lambda$ the (simplified) pseudo-gcrd and pseudo-lclm of K and L.

Lemma 5 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ be such that K and L are non singular at the origin, as well as* $\operatorname{gcrd}^*(K, L)$ *or* $\operatorname{lclm}^*(K, L)$. *Let G and H be the canonical fundamental systems of solutions to $K(G) = 0$ and $L(H) = 0$. Then*

$$\deg_\delta \operatorname{lclm}^*(K, L) = \dim([\operatorname{Vect}(G) + \operatorname{Vect}(H)] \bmod x^{2r})$$
$$\deg_\delta \operatorname{gcrd}^*(K, L) = \dim([\operatorname{Vect}(G) \cap \operatorname{Vect}(H)] \bmod x^{2r}).$$

Proof Let $\Gamma^* = \operatorname{gcrd}^*(K, L)$, $\Lambda^* = \operatorname{lclm}^*(K, L)$, $s = \deg_\delta \Gamma^*$ and $t = \deg_\delta \Lambda^* \leqslant 2r$. If Λ^* is non singular, then it admits a canonical fundamental system of solutions $M = (M_0, \ldots, M_{t-1})$ with $(M_i)_i = 1$ and $(M_i)_j = 0$ for all $i, j < t$ with $i \neq j$. In particular, $\dim(\operatorname{Vect}(M) \bmod x^{2r}) = t$. Since Λ^* is the least common left multiple of K and L, we also have $\operatorname{Vect}(M) = \operatorname{Vect}(G) + \operatorname{Vect}(H)$, which completes the proof of the first equality. If Γ^* is non singular, then we obtain the second equality in a similar way.

If Λ^* is non singular, then we also have $\dim(\operatorname{Vect}(K) \bmod x^{2r}) = \deg_\delta K$ and $\dim(\operatorname{Vect}(L) \bmod x^{2r}) = \deg_\delta L$, since K and L are non singular. Now

$\dim([\mathrm{Vect}(G) \cap \mathrm{Vect}(H)] \bmod x^{2r}) = \dim(\mathrm{Vect}(K) \bmod x^{2r}) + \dim(\mathrm{Vect}(L) \bmod x^{2r}) - \dim([\mathrm{Vect}(G) + \mathrm{Vect}(H)] \bmod x^{2r})$, whence $\dim([\mathrm{Vect}(G) \cap \mathrm{Vect}(H)] \bmod x^{2r}) = \deg_\delta K + \deg_\delta L - \deg_\delta \Lambda^* = \deg_\delta \Gamma^*$. If Γ^* is non singular, then we obtain the first equality in a similar way. □

Theorem 9 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ and $n' \geqslant n$ be such that $\Gamma^* = \mathrm{gcrd}^*(K, L) \in \mathbb{K}[x, \delta]_{n',r}$ and $n \geqslant r$. Assume that K, L and $\mathrm{gcrd}^*(K, L)$ (or $\mathrm{lclm}^*(K, L)$) are non singular at the origin. Then we may compute Γ^* in time $O(\mathsf{SM}(n', r) \log n')$.*

Proof We compute Γ^* using the following algorithm:

- We compute the canonical fundamental systems of solutions G and H to $K(G) = 0$ and $L(H) = 0$ at order $O(x^{2n'+r})$. This can be done in time $O(\mathsf{SM}(n', r) \log n')$.
- Using linear algebra, we compute a basis B for $V = \mathrm{Vect}(G) \cap \mathrm{Vect}(H)$ at order $O(x^{2n'+r})$. This can be done in time $O(n' r^{\omega-1})$. By Lemma 5, we have $s := \dim(V \bmod x^{2r}) = \deg_\delta \Gamma^*$. We also notice that $v^{\max}(B) < r$.
- We compute $\Omega = \mathrm{ann}(B) = \mathrm{gcrd}(K, L)$ at order $O(x^{2n'})$. By Theorem 6, this can be done in time $O(\mathsf{SM}(n', r) \log n')$.
- We compute Γ^* from $\Omega \bmod x^{2n'}$ using rational function reconstruction.

This algorithm requires a total running time $O(\mathsf{SM}(n', r) \log n')$. □

Remark 5 In the above proof, we have again assumed that n' is known beforehand. Below, we will discuss ways to check the correctness of the computed result for a trial value n^*, after which a similar strategy as in remark 4 can be applied. During the relaxed computation of G and H, we may also check whether $V = \varnothing$ at each next coefficient. In the particular case when $\Gamma = 1$, the running time of the algorithm will then be bounded by $O(\mathsf{SM}(n^*, r) \log n^*)$, where n^* is the smallest order at which common solutions no longer exist. This kind of early termination only works for this very special case.

Remark 6 Notice that Γ^* might be singular at the origin, even if K, L and $\mathrm{lclm}^*(K, L)$ are not. This happens for instance when K is the minimal annihilator of the vector $(1, x)$ and L the minimal annihilator of the vector (e^x, x), so that $\Gamma = \delta - 1$.

Theorem 10 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ and $n' \geqslant n$ be such that $\Lambda^* = \mathrm{lclm}^*(K, L) \in \mathbb{K}[x, \delta]_{n',2r}$ and $n \geqslant r$. If K, L and $\mathrm{lclm}^*(K, L)$ (or $\mathrm{gcrd}^*(K, L)$) are non singular at the origin, then we may compute Λ^* in time $O(\mathsf{SM}(n', r) \log n')$.*

Proof Similar to the proof of Theorem 9, by taking $V = \mathrm{Vect}(K) + \mathrm{Vect}(L)$ instead of $V = \mathrm{Vect}(K) \cap \mathrm{Vect}(L)$. □

5.2 Certifying correctness

The assumption that $\mathrm{lclm}^*(K, L)$ should be non singular is still a bit unsatisfactory in Theorems 9 and 10, even though the probability that a randomly chosen point is singular is infinitesimal. If we drop this assumption, then we still have $s \geqslant \deg_\delta \Gamma^*$ in the proof of Theorem 9. Consequently, "candidate" pseudo-gcrds Γ^* found by

the algorithm are genuine pseudo-gcrds whenever Γ^* pseudo-divides both K and L. Using the right division algorithms from the previous section, this can be checked in time $O(\mathsf{SM}(n'r, r)\log n')$ in the case of gcrds and $O(\mathsf{SM}(nr, r)\log n')$ in the case of lclms.

Remark 7 Using the polynomial linear algebra techniques from [6,16], it is likely that one may prove that $PK = A\Gamma^*$ for some $P \in \mathbb{K}[x]_{nr}$ and $A \in \mathbb{K}[x, \delta]_{nr,r}$. If this is indeed the case, then the trial divisions of K and L by Γ^* can actually be carried out in time $O(\mathsf{SM}(nr, r)\log n')$.

An alternative way to check whether candidate gcrds and lclms are correct is to compute Bezout and Ore relations. More precisely, given $K, L \in \mathbb{K}(x)[\delta]$ with $L \notin \mathbb{Q}(x)K$, there exist operators $A, B, C, D \in \mathbb{K}(x)[\delta]$ with

$$\begin{pmatrix} \Gamma \\ 0 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} K \\ L \end{pmatrix},$$

$\deg_\delta AK, \deg_\delta BL < \deg_\delta \Lambda$ and $CK = -DL = \Lambda$. The 2×2 matrix at the righthand side will be called the Euclidean matrix $\mathrm{E} = \mathrm{Eucl}(K, L)$ of K and L. In a similar way as above, we may define a (simplified) pseudo-Euclidean matrix $\mathrm{E}^* = \mathrm{Eucl}^*(K, L)$ with entries A^*, B^*, C^*, D^* in $\mathbb{K}[x, \delta]$, whenever $K, L \in \mathbb{K}[x, \delta]$. We will say that $\mathrm{Eucl}(K, L)$ is non singular at x_0, if the denominators of A, B, C and D do not vanish at x_0.

Theorem 11 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ and $n' \geqslant n$ be such that $\mathrm{E}^* = \mathrm{Eucl}^*(K, L) \in \mathbb{K}[x, \delta]_{n',r}^{2\times 2}$ and $n \geqslant r$. If $K, L, \mathrm{lclm}^*(K, L)$ and $\mathrm{Eucl}(K, L)$ are non singular at the origin, then we may compute Λ^* in time $O(\mathsf{SM}(n', r)\log n') = \tilde{O}(n'r^{\omega-1})$.*

Proof Assuming n' known, we compute $\mathrm{Eucl}(K, L) = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ at order $O(x^{2n'})$ as follows:

- We compute the canonical fundamental systems of solutions G and H to $K(G) = 0$ and $L(H) = 0$ at order $O(x^{2n'+3r})$.
- We compute a basis X for $\mathrm{Vect}(G) \cap \mathrm{Vect}(H)$ at order $O(x^{2n'+3r})$, together with bases $\hat{G}$ and $\hat{H}$ for the supplements of $\mathrm{Vect}(X)$ in $\mathrm{Vect}(G)$ resp. $\mathrm{Vect}(H)$. We also compute $\Gamma = \mathrm{ann}(X)$ at order $O(x^{2n'+2r})$.
- We solve the systems $A(K(\hat{H})) = \Gamma(\hat{H})$ and $B(L(\hat{G})) = \Gamma(\hat{G})$ in A resp. B at order $O(x^{2n'})$, using Lemma 4.
- We compute a basis Y for $\mathrm{Vect}(G) + \mathrm{Vect}(H)$ at order $O(x^{2n'+2r})$, as well as bases $\tilde{H}$ and $\tilde{G}$ for the vector spaces $\mathrm{Vect}(K(Y))$ resp. $\mathrm{Vect}(L(Y))$ at order $O(x^{2n'+2r})$.
- We compute $C = K^{-1}_{\deg_\delta K}\,\mathrm{ann}(\tilde{H})$ and $D = -L_{\deg_\delta L}\,\mathrm{ann}(\tilde{G})$ at order $O(x^{2n'})$.

We finally compute E^* from A, B, C and D using rational function reconstruction. The complexity analysis and the remainder of the proof is done in a similar way as in the proofs of Theorems 8 and 9. □

With the above techniques, we may at least verify whether computed pseudo-gcrds or pseudo-lclms are correct. For a fully deterministic algorithm, we still need a way

to find a point where $\mathrm{lclm}^*(K, L)$ is non singular. This can be done by brute force. Let us state the result in the most general setting of pseudo-Euclidean matrices.

Theorem 12 *Let $K, L \in \mathbb{K}[x, \delta]_{n,r}$ and $n' \geqslant n$ be such that $\mathrm{E}^* = \mathrm{Eucl}^*(K, L) \in \mathbb{K}[x, \delta]_{n',r}^{2\times 2}$ and $n \geqslant r$. Then we may compute E^* in time $O(\mathsf{SM}(n', r)\log n' + n'(\mathsf{M}(n)r + r^\omega \log r)) = \tilde{O}(n'(r^\omega + nr))$.*

Proof Let $k = \deg_\delta K, l = \deg_\delta L$, and assume first that we know n'. Let $x_0, \ldots, x_{n'+n}$ be $n' + n + 1$ be pairwise distinct, randomly chosen points in $\mathbb{K}$ at which K and L are non singular. At each x_i, we compute canonical fundamental systems of solutions G and H for K and L at order $O(x^{k+l})$. We claim that this can be done in time $O\left(\mathsf{M}(n')r\log n' + n'\left(\mathsf{M}(n)r + r^\omega \log r\right)\right)$.

Indeed, it requires a time $O((n + r)\mathsf{M}(r)\log r)$ to rewrite each operator with respect to ∂. We next perform a multipoint evaluation of the coefficients of these operators to obtain the shifted operators at $x_0, \ldots, x_{n'+n}$ (this requires a time $O\left(\mathsf{M}(n')r\log n'\right)$). The truncations of these operators at order $O(x^{k+l+r})$ are then converted back to the respresentation with respect to δ. This can be done in time $O\left(n'r\mathsf{M}(r)\log r\right)$. Using Theorem 5, we finally compute the required fundamental systems of solutions in time $O\left(n'\mathsf{SM}(r, r)\log r\right) = O(n'r^\omega \log r)$.

From $\mathrm{E}^* \in \mathbb{K}[x, \delta]_{n',r}^{2\times 2}$, we get $\Lambda^* = \mathrm{lclm}^*(K, L) \in \mathbb{K}[x, \delta]_{n'+n,2r}$. Since we assumed n' to be sufficiently large, it follows that $\Lambda^* = \mathrm{lclm}^*(K, L)$ is non singular at one of the points x_i. At such a point x_i, the canonical fundamental systems of solutions G and H generate a vector space $V = \mathrm{Vect}(G) + \mathrm{Vect}(H)$ of maximal dimension $s := \deg_\delta \Lambda^*$, and with a basis $y_0, \ldots, y_{s-1}$ such that $v(y_k) = k$ for all $0 \leqslant k < s$. We finally apply Theorem 11 in order to obtain E^*. If n' is unknown, then we use a sequence of guesses $n' = n, 2n, 4n, \ldots$, as in the previous proofs. □

Remark 8 In the case of least common left multiples, we may directly compute Λ^* using Theorem 10 and certify the result using trial division by K and L. This allows us to use the weaker assumption $\Lambda^* \in \mathbb{K}[x, \delta]_{n',2r}$ instead of $\mathrm{E}^* \in \mathbb{K}[x, \delta]_{n',r}^{2\times 2}$, whereas the complexity bound becomes $O\left(\mathsf{SM}(nr, r)\log n' + n'(\mathsf{M}(n)r + r^\omega \log r)\right) = \tilde{O}(n'(r^\omega + nr))$.

5.3 Summary of the complexity bounds for Euclidean operations

We have summarized our complexity bounds for Euclidean operations on two operators $K, L \in \mathbb{K}[x, \delta]_{n,r}$ in Table 1. We systematically write n' for the degree in x of the result. We also write n^* for the degree of the Euclidean matrix in x.

The algorithms in the first line correspond to applying Theorems 9, 10 and 11 at a randomly chosen point, without checking the result. The second line corresponds to the Las Vegas randomized algorithm for which the answers are certified through trial division (the bound for gcrds might further drop to $\tilde{O}(nr^\omega)$ in view of Remark 7; more generally, the bounds can be restated in terms of sizes of certificates). In the third line, we rather use Euclidean matrices for the certification. The fourth line shows complexity bounds for the deterministic versions of our algorithms.

Table 1 Complexity bounds for the Euclidean operations on two operators K and L

Algorithm	gcrd	lclm	Euclidean matrix
Randomized, uncertified	$\tilde{O}(n'r^{\omega-1})$	$\tilde{O}(n'r^{\omega-1})$	$\tilde{O}(n'r^{\omega-1})$
Certified *via* division	$\tilde{O}(n'r^{\omega})$	$\tilde{O}(nr^{\omega})$	
Euclidean certification	$\tilde{O}(n^*r^{\omega-1})$	$\tilde{O}(n^*r^{\omega-1})$	$\tilde{O}(n'r^{\omega-1})$
Deterministic	$\tilde{O}(n^*(r^{\omega}+nr))$	$\tilde{O}(n'(r^{\omega}+nr))$	$\tilde{O}(n'(r^{\omega}+nr))$

In comparison, several randomized Las Vegas algorithms were given in [6] that achieve the complexity bound $\tilde{O}(nr^{\omega})$ for lclms. This is in particular the case for Heffter's algorithm [17], when using Theorem 3. The non determinism is due to the use of a fast Las Vegas randomized algorithm for the computation of kernels of matrices with polynomial entries [6, Theorem 2]. Grigoriev established complexity bounds for gcrds which rely on a similar reduction to polynomial linear algebra. Along the same lines as in [6], this should lead to a Las Vegas randomized algorithm of complexity $\tilde{O}(nr^{\omega})$, although we did not check this in detail.

In summary, the new algorithms do not achieve any improvements in the worst case. Nevertheless, the uncertified versions of our algorithms admit optimal running times up to logarithmic factors in terms of the combined input *and* output size. The certified randomized versions satisfy similar complexity bounds in terms of the size of a suitable certificate; such bounds can sometimes be better than the previously known worst case bounds. When performing our expansions at a randomly chosen point in $\mathbb{K}$, we also recall that the probability of failure is exponentially small as a function of the bitsize of this point.

5.4 Generalizations

The algorithms from Sect. 5.1 extend in a straightforward way to the computation of greatest common right divisors and least common left multiples of more than two operators. For instance, using obvious notations, we obtain the following generalizations of Theorems 10 and 9.

Theorem 13 *Let $L_1, \ldots, L_k \in \mathbb{K}[x,\delta]_{n,r}$ with $n \geqslant r$ and $r' \geqslant r$, $n' \geqslant \max(n, r')$ be such that $\Lambda^* = \mathrm{lclm}^*(L_1, \ldots, L_k) \in \mathbb{K}[x,\delta]_{n',r'}$. Assume that $L_1, \ldots, L_k$ and Λ^* are all non singular at the origin. Then we may compute Λ^* in time $O(\mathsf{SM}(n', r')\log n' + k\mathsf{SM}(n', r)\log n' + kr(r')^{\omega-2}n')$.*

Proof We compute Λ^* using the following algorithm:

- We first compute the canonical fundamental systems of solutions H_i to $L_i(H_i) = 0$ at order $O(x^{2n'+r'})$. This can be done in time $O(k\mathsf{SM}(n', r)\log n')$.
- Let $V_{i,j} = \mathrm{Vect}(H_i) + \cdots + \mathrm{Vect}(H_j)$ for all $1 \leqslant i \leqslant j \leqslant k$. Using linear algebra, we may recursively compute a basis $B_{i,j}$ for $V_{i,j}$ from bases $B_{i,m}$ and $B_{m+1,j}$ for $V_{i,m}$ and $V_{m+1,j}$, where $m = \lfloor (i+j)/2 \rfloor$. This algorithm yields a basis B for

$V_{1,k}$ in time $O(kr(r')^{\omega-2}n')$. Using a suitable generalization of Lemma 5, we also notice that $\dim(V \bmod x^{r'}) = \deg_\delta \Lambda^*$.
- We compute $\Omega = \operatorname{ann}(B) = \operatorname{lclm}(L_1, \ldots, L_k)$ at order $O(x^{2n'})$. By Theorem 6, this can be done in time $O(\mathsf{SM}(n', r') \log n')$.
- We compute Λ^* from $\Omega \bmod x^{2n'}$ using rational function reconstruction.

We obtain the result by adding up all complexity bounds. □

Remark 9 When taking $r' = kr \leqslant n'$ and using [2], the complexity bound simplifies to $O\left(\mathsf{SM}(n', kr) \log n'\right) = O\left(k^{\omega-1} r^{\omega-1} n' \log n' + kr\mathsf{M}(n') \log^2 n'\right)$. By [6, Theorem 6], we may always take $n' = nrk^2$, after which the bound further reduces to $\tilde{O}(k^{\omega+1} r^\omega n)$. In our randomized setting, this improves upon the bounds from [6, Figure 1].

Remark 10 If we also require a certification of the result, then we may use the trial division technique. This amounts to k exact divisions of operators in $\mathbb{K}[x, \delta]_{n'+nr',r'}$ by $L_1, \ldots, L_k$. Using the division algorithm from Sect. 4, and taking $r' = kr \leqslant n'$ and $n' = nrk^2$ as above, this can be done in time

$$O\left(k\mathsf{SM}(n' + nr', r') \log(nr')\right) = \tilde{O}(k(n' + nr')(r')^{\omega-1}) = \tilde{O}(k^{\omega+2} r^\omega n).$$

This is slightly better than the new bound from [6].

Theorem 14 *Let $L_1, \ldots, L_k \in \mathbb{K}[x, \delta]_{n,r}$ and $n' \geqslant n \geqslant r$ be such that $\Gamma^* = \operatorname{gcrd}^*(L_1, \ldots, L_k) \in \mathbb{K}[x, \delta]_{n',r}$. Assume that $L_1, \ldots, L_k$ and Γ^* are all non singular at the origin. Then we may compute Γ^* in time $O(\mathsf{SM}(n', r) \log n' + k\mathsf{SM}(r, r) \log r)$.*

Proof The proof is similar to the one of Theorem 13, except for the way how we compute a basis for $V = \operatorname{Vect}(H_1) \cap \cdots \cap \operatorname{Vect}(H_k)$. Indeed, we first compute a basis $B \bmod x^r$ for $V \bmod x^r$. This requires a time $O\left(k\mathsf{SM}(r, r) \log r\right)$ for the computation of $H_1, \ldots, H_k$ modulo x^r and a time $O(kr^\omega)$ for the remaining linear algebra. We next compute the unique constant matrix C such that $B = CH_1$ modulo x^r. Since Γ^* is non singular, we have $B = CH_1$ at any order, so it suffices to compute H_1 up to order $x^{2n'+r}$ in order to obtain B up to order $x^{2n'+r}$. □

Remark 11 An interesting question is whether there exists a faster algorithm to compute the orders s and t of $\Gamma^* = \operatorname{gcrd}^*(L_1, \ldots, L_k)$ and $\Lambda^* = \operatorname{lclm}^*(L_1, \ldots, L_k)$, without computing Γ^* and Λ^* themselves. For this, it suffices to compute the dimensions of $\operatorname{Vect}(H_1) \cap \cdots \cap \operatorname{Vect}(H_k)$ and $\operatorname{Vect}(H_1) + \cdots + \operatorname{Vect}(H_k)$. Assuming that we are at a "non singular point", the answer is therefore yes: using the techniques from the proofs of Theorems 14 and 13, we may compute s in time $O\left(k\mathsf{SM}(r, r) \log r\right) = \tilde{O}(kr^\omega)$ and t in time $O\left(k\mathsf{SM}(t, r) \log t + krt^{\omega-1}\right) = \tilde{O}(krt^{\omega-1})$.

Acknowledgments The author is grateful to the second referee whose questions and remarks led to several improvements with respect to the first version of this paper. The article was originally written by the author using GNU TeXmacs, and Springer acknowledges the assistance of the author with the conversion into Springer's LaTeX format.

References

1. Aho, A.V., Steiglitz, K., Ullman, J.D.: Evaluating polynomials on a fixed set of points. SIAM J. Comput. **4**, 533–539 (1975)
2. Benoit, A., Bostan, A., van der Hoeven, J.: Quasi-optimal multiplication of linear differential operators. In Proceedings of FOCS '12, pp. 524–530. IEEE, New Brunswick (2012)
3. Borodin, A., Moenck, R.T.: Fast modular transforms. J. Comput. Syst. Sci. **8**, 366–386 (1974)
4. Bostan, A.: Algorithmique efficace pour des opérations de base en calcul formel. Ph.D. Thesis, École polytechnique (2003)
5. Bostan, A., Chyzak, F., Le Roux, N.: Products of ordinary differential operators by evaluation and interpolation. In: Rafael Sendra, J., González-Vega, L. (eds.) ISSAC, pp. 23–30. ACM Press, Linz/Hagenberg (2008)
6. Bostan, A., Chyzak, F., Salvy, B., Li, Z.: Fast computation of common left multiples of linear ordinary differential operators. In: van der Hoeven, J., van Hoeij, M. (eds.) Proceedings of ISSAC '12, pp. 99–106. Grenoble, France (2012)
7. Bostan, A., Schost, É.: Polynomial evaluation and interpolation on special sets of points. J. Complex. **21**(4), 420–446 (2005). Festschrift for the 70th Birthday of Arnold Schönhage
8. Brassine, E.: Analogie des équations différentielles linéaires à coefficients variables, avec les équations algébriques, pp. 331–347. Note III du Tome 2 du Cours d'analyse de Ch. Sturm. École polytechnique (1864)
9. Cantor, D.G., Kaltofen, E.: On fast multiplication of polynomials over arbitrary algebras. Acta Inf. **28**, 693–701 (1991)
10. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Math. Comput. **19**, 297–301 (1965)
11. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. In: Proceedings of the 19th Annual Symposium on Theory of Computing, pp. 1–6. New York City (1987)
12. Demidov, S.S.: On the history of the theory of linear differential equations. Arch. Hist. Exact. Sci. **28**(4), 369–387 (1983)
13. Giesbrecht, M.: Factoring in skew polynomial rings over finite fields. In: Proceedings of LATIN '92, Volume 583 of LNCS, pp. 191–203 (1992)
14. Giesbrecht, M.: Factoring in skew polynomial rings over finite finite fields. JSC **26**, 463–486 (1998)
15. Giesbrecht, M., Zhang, Y.: Factoring and decomposing ore polynomials over $\mathbb{F}_q(t)$. In: Bronstein, M. (ed.) Proceedings of ISSAC '03, pp. 127–134. Philadelphia, USA (2003)
16. Grigoriev, D.Y.: Complexity of factoring and calculating the GCD of linear ordinary differential operators. J. Symb. Comput. **10**(1), 7–37 (1990)
17. Heffter, L.: Über gemeinsame Vielfache linearer Differentialausdrücke und lineare Differentialgleichungen derselben Klasse. J. Reine Angew. Math. **116**, 157–166 (1896)
18. Le Gall, F.: Powers of tensors and fast matrix multiplication. In: Proceedings of ISSAC 2014, pp. 296–303. Kobe, Japan (2014)
19. Li, Z.: A subresultant theory for ore polynomials with applications. In: Gloor, O. (ed.) Proceedings of ISSAC '98, pp. 132–139. Rostock, Germany (1998)
20. Libri, G.: Mémoire sur la résolution des équations algébriques dont les racines ont entre elles un rapport donné, et sur l'intégration des équations différentielles linéaires dont les intégrales particulières peuvent s'exprimer les unes par les autres. J. Reine Angew. Math. **10**, 167–194 (1833)
21. Moenck, R.T., Borodin, A.: Fast modular transforms via division. In: Thirteenth Annual IEEE Symposium on Switching and Automata Theory, pp. 90–96. Univ. Maryland, College Park, MD (1972)
22. Ore, O.: Theorie der linearen Differentialgleichungen. J. Reine Angew. Math. **167**, 221–234 (1932)
23. Ore, O.: Theory of non-commutative polynomials. Ann. Math. **34**(3), 480–508 (1933)
24. Pan, V.: How to Multiply Matrices Faster, Volume 179 of Lect. Notes in Math. Springer, Berlin (1984)
25. Schönhage, A.: Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. Acta Inf. **7**, 395–398 (1977)
26. Schönhage, A., Strassen, V.: Schnelle Multiplikation großer Zahlen. Computing **7**, 281–292 (1971)
27. Stanley, R.P.: Differentially finite power series. Eur. J. Comb. **1**, 175–188 (1980). MR #81m:05012
28. Strassen, V.: Gaussian elimination is not optimal. Numer. Math. **13**, 352–356 (1969)
29. Strassen, V.: Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. Numer. Math. **20**, 238–251 (1973)
30. van der Hoeven, J.: FFT-like multiplication of linear differential operators. JSC **33**(1), 123–127 (2002)

31. van der Hoeven, J.: Relax, but don't be too lazy. JSC **34**, 479–542 (2002)
32. van der Hoeven, J.: Relaxed multiplication using the middle product. In: Bronstein, M. (ed.) Proceedings of ISSAC '03, pp. 143–147. Philadelphia, USA (2003)
33. van der Hoeven, J.: On the complexity of skew arithmetic. Technical Report, HAL (2011). http://hal.archives-ouvertes.fr/hal-00557750
34. van der Hoeven, J., Lecerf, G., Mourrain, B., et al.: Mathemagix (2002). http://www.mathemagix.org
35. von zur Gathen, J., Gerhard, J.: Mod. Comput. Algebra, 2nd edn. Cambridge University Press, Cambridge (2002)