Fast algorithms for differential equations in positive characteristic

Alin Bostan, Éric Schost

▶ **To cite this version:**

Alin Bostan, Éric Schost. Fast algorithms for differential equations in positive characteristic. 2009. inria-00355818

HAL Id: inria-00355818
https://inria.hal.science/inria-00355818

Preprint submitted on 24 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast algorithms for differential equations in positive characteristic

Alin Bostan
Algorithms Project
INRIA Rocquencourt
France
78153 Le Chesnay Cedex France

Éric Schost
ORCCA and Computer Science Department
The University of Western Ontario
London, ON, Canada
eschost@uwo.ca

ABSTRACT

We address complexity issues for linear differential equations in characteristic $p > 0$: resolution and computation of the p-curvature. For these tasks, our main focus is on algorithms whose complexity behaves well with respect to p. We prove bounds linear in p on the degree of polynomial solutions and propose algorithms for testing the existence of polynomial solutions in sublinear time $\tilde{O}(p^{1/2})$, and for determining a whole basis of the solution space in quasi-linear time $\tilde{O}(p)$; the $\tilde{O}$ notation indicates that we hide logarithmic factors. We show that for equations of arbitrary order, the p-curvature can be computed in subquadratic time $\tilde{O}(p^{1.79})$, and that this can be improved to $O(\log(p))$ for first order equations and to $\tilde{O}(p)$ for classes of second order equations.

Categories and Subject Descriptors:
I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation – *Algebraic Algorithms*

General Terms: Algorithms, Theory

Keywords: Algorithms, complexity, differential equations, polynomial solutions, p-curvature.

1. INTRODUCTION

We study several algorithmic questions related to linear differential equations in characteristic p, where p is a prime number: resolution of such equations and computation of their p-curvature. Our emphasis is on the complexity viewpoint.

Let thus $\mathbb{F}_p$ be the finite field with p elements, and let $\mathbb{F}_p(x)\langle\partial\rangle$ be the algebra of differential operators with coefficients in $\mathbb{F}_p(x)$, with the commutation relation $\partial x = x\partial + 1$. One of the important objects associated to a differential operator L of order r in $\mathbb{F}_p(x)\langle\partial\rangle$ is its p-curvature, hereafter denoted $\mathbf{A}_p$. By definition, this is the $(r \times r)$ matrix with coefficients in $\mathbb{F}_p(x)$, whose (i, j)-entry is the coefficient of ∂^i in the remainder of the Euclidean (right) division of ∂^{p+j} by L, for $0 \leq i, j < r$.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2001 ACM 0-89791-88-6/97/05 ...$5.00.

The concept of p-curvature originates in Grothendieck's work in the late 1960s, in connection to one of his famous (still unsolved) conjectures. In its simplest form, this conjecture is an arithmetic criterion of algebraicity, which states that a linear differential equation with coefficients in $\mathbb{Q}(x)$ has a basis of algebraic solutions over $\mathbb{Q}(x)$ if and only if its reductions modulo p have zero p-curvature, for almost all primes p. The search of a proof of this criterion motivated the development of a theory of differential equations in characteristic p by Katz [20], Dwork [14], Honda [19], etc.

There are two basic differences between differential equations in characteristic zero and p: one concerns the dimension of the solution space, the other, the form of the solutions. While in characteristic zero, a linear differential equation of order r admits exactly r linearly independent solutions, this is no longer true in positive characteristic: for $L \in \mathbb{F}_p(x)\langle\partial\rangle$, the dimension of the solution space of the equation $Ly = 0$ over the field of constants $\mathbb{F}_p(x^p)$ is generally less than the order r. Moreover, by a theorem of Cartier and Katz (see Lemma 2 below), the dimension is exactly r if and only if the p-curvature matrix $\mathbf{A}_p$ is zero. Thus, roughly speaking, the p-curvature measures to what extent the solution space of a differential equation modulo p has dimension close to its order.

On the other hand, the form of the solutions is simpler in characteristic p than in characteristic zero. Precisely, the existence of polynomial solutions is equivalent to the existence of solutions which are either algebraic over $\mathbb{F}_p(x)$, or power series in $\mathbb{F}_p[[x]]$, or rational functions in $\mathbb{F}_p(x)$ [19]. Therefore, in what follows, by solving $Ly = 0$ we simply understand finding its polynomial solutions.

In computer algebra, the p-curvature was publicised by van der Put [25, 26], who used it as a central tool in designing algorithms for factoring differential operators in $\mathbb{F}_p(x)\langle\partial\rangle$. Recently, his algorithms were analyzed from the complexity perspective and implemented by Cluzeau [11], who extended them to the case of systems. Cluzeau also took in [12] a first step towards a systematic modular approach to the algorithmic treatment of differential equations.

Improving the complexity of the p-curvature computation is an interesting problem in its own right. Our main motivation for studying this question comes, however, from concrete applications. First, in a combinatorial context, the use of the p-curvature served in the automatic classification of restricted lattice walks [8] and notably provided crucial help in the treatment of the notoriously difficult case of Gessel's walks [7]. Also, intensive p-curvature computations were

needed in [4], where the question is to decide whether various differential operators arising in statistical physics have nilpotent, or zero, p-curvature.

In the latter questions, the prime p was "large", typically of the order of 10^4. This remark motivates our choice of considering p as the most important parameter: our primary objective is to obtain complexity estimates featuring a low exponent in p.

Previous work. The non-commutativity of $\mathbb{F}_p(x)\langle\partial\rangle$ prevents one from straightforwardly using binary powering techniques for the computation of $\mathbf{A}_p$ via that of $\partial^p \bmod L$. Thus, the complexity of all currently known algorithms for computing the p-curvature is quadratic in p.

Katz [21] gave the first algorithm, based on the following matrix recurrence: define

$$\mathbf{A}_1 = \mathbf{A}, \quad \mathbf{A}_{k+1} = \mathbf{A}_k' + \mathbf{A}\mathbf{A}_k, \tag{1}$$

where $\mathbf{A} \in \mathscr{M}_r(\mathbb{F}_p(x))$ is the companion matrix associated to L; then, $\mathbf{A}_p$ is the p-curvature matrix (hence our notation).

It was observed in [27, §13.2.2] that it is slightly more efficient to replace (1) by the recurrence $\mathbf{v}_{k+1} = \mathbf{v}_k' + \mathbf{A}\mathbf{v}_k$ which computes the first column $\mathbf{v}_k$ of $\mathbf{A}_k$, by taking for $\mathbf{v}_0$ the first column of $\mathbf{I}_r$. Then $\mathbf{v}_p, \dots, \mathbf{v}_{p+r-1}$ are the columns of $\mathbf{A}_p$. This alternative requires only matrix-vector products, and thus saves a factor of r, but still remains quadratic in p. Cluzeau proposed in [11, Prop. 3.2] a fraction-free version of (1) having essentially the same complexity, but incorrectly stated that the method in [27] works in linear time in p.

Concerning polynomial and rational solutions of differential equations modulo p, very few algorithms can be found in the literature. Cluzeau proposes in [11, §2] an algorithm of cubic complexity in p and, in the special case when $\mathbf{A}_p = 0$, a different algorithm of quadratic complexity in p, based on a formula due to Katz which is the nub of Lemma 2 below.

Our contribution. We prove in Section 3 a linear bound in p on the degree for a basis of the solution space of polynomial solutions of an equation $Ly = 0$. Then, we adapt the algorithm in [1] and its improvements [6] to the case of positive characteristic; we show how to test the existence of polynomial solutions in time nearly proportional to $p^{1/2}$, and how to determine a full basis of the solution space in time quasi-linear in p.

Regarding the p-curvature, we first focus on two particular cases: first order operators, where the cost is polynomial in $\log(p)$ (Section 4), and second order ones, for which we obtain a cost quasi-linear in p in some cases (Section 5).

In general, a useful way to see (1) is to note that the p-curvature is obtained by applying the operator $(\partial + \mathbf{A})^{p-1}$ to $\mathbf{A}$. In Section 6 we exploit this observation. As a side result, we give a baby steps / giant steps algorithm for computing the image Lu of an operator L applied to a polynomial u; this is inspired by Brent-Kung's algorithm for power series composition [9].

Complexity measures. Time complexities are measured in terms of arithmetic operations in $\mathbb{F}_p$.

We let $\mathsf{M} : \mathbb{N} \to \mathbb{N}$ be such that polynomials of degree at most n in $\mathbb{F}_p[x]$ can be multiplied in time $\mathsf{M}(n)$. Furthermore, we assume that $\mathsf{M}(n)$ satisfies the usual assumptions of [18, §8.3]; using Fast Fourier Transform, $\mathsf{M}(n)$ can be taken in $O(n \log n \log\log n)$ [23, 10]. We suppose that $2 \le \omega \le 3$ is a constant such that two matrices in $\mathscr{M}_n(\mathbb{F}_p)$ can be multiplied in time $O(n^\omega)$. The current tightest upper bound is $\omega < 2.376$ [13].

The precise complexity estimates of our algorithms are sometimes quite complex; to highly their main features, we rather give simplified estimates. Thus, we use the notation $f \in \tilde{O}(g)$ for $f, g : \mathbb{N} \to \mathbb{N}$ if f is in $O(g \log(g)^m)$ for some $m \ge 1$. For instance, $\mathsf{M}(n)$ is in $\tilde{O}(n)$.

2. PRELIMINARIES

Basic properties of the p-curvature. We first give degree bounds on the p-curvature of an operator. Consider

$$L = \ell_0(x) + \ell_1(x)\partial + \cdots + \ell_r(x)\partial^r, \tag{2}$$

with all ℓ_i in $\mathbb{F}_p[x]$ of degrees at most d and $\ell_r \neq 0$. As in (1), we define $\mathbf{A}_1 = \mathbf{A}$ and $\mathbf{A}_{k+1} = \mathbf{A}_k' + \mathbf{A}\mathbf{A}_k$ for $k \ge 1$.

LEMMA 1. *For $k \ge 0$, let $\mathbf{B}_k = \ell_r^k \mathbf{A}_k$. Then $\mathbf{B}_k$ is in $\mathscr{M}_r(\mathbb{F}_p[x])$, with entries of degree at most dk.*

PROOF. Explicitly, we have

$$\mathbf{A} = \begin{bmatrix} & & & -\frac{\ell_0}{\ell_r} \\ 1 & & & -\frac{\ell_1}{\ell_r} \\ & \ddots & & \vdots \\ & & 1 & -\frac{\ell_{r-1}}{\ell_r} \end{bmatrix}.$$

From this, we see that the sequence $\mathbf{B}_k$ satisfies the equation

$$\mathbf{B}_{k+1} = \ell_r \mathbf{B}_k' + (\mathbf{B}_1 - k\ell_r' \mathbf{I}_r)\mathbf{B}_k,$$

where $\mathbf{I}_r$ is the $r \times r$ identity matrix. The claim follows. □

In particular, the p-curvature $\mathbf{A}_p$ has the form $\mathbf{B}_p/\ell_r^p$, with $\mathbf{B}_p$ a polynomial matrix of degree at most dp.

A second useful result is the following lemma, attributed to Katz. It relates the solution space of $Ly = 0$ to the p-curvature and generalizes a theorem of Cartier. A proof can be found in [11, Th. 3.8].

LEMMA 2. *The dimension over $\mathbb{F}_p(x^p)$ of the vector space of rational solutions of L is equal to the dimension over $\mathbb{F}_p(x)$ of the kernel of $\mathbf{A}_p$. In particular, L has a basis of polynomial solutions if and only if its p-curvature is zero.*

Operator algebras. In what follows, we mainly consider operators with coefficients in $\mathbb{F}_p(x)$, but also sometimes more generally in the $(n \times n)$ matrix algebra $\mathscr{M}_n(\mathbb{F}_p(x))$; as has been done up to now, we will write matrices in bold face. If L is in $\mathscr{M}_n(\mathbb{F}_p(x))\langle\partial\rangle$ of the form

$$L = \boldsymbol{\ell}_0(x) + \boldsymbol{\ell}_1(x)\partial + \cdots + \boldsymbol{\ell}_r(x)\partial^r,$$

with coefficient matrices $\boldsymbol{\ell}_i$ in $\mathscr{M}_n(\mathbb{F}_p[x])$ of maximal degree d, we say that L has *bidegree* (d, r).

Regularization. For most of our algorithms, we must assume that the origin $x = 0$ does not cancel the leading term $\ell_r \in \mathbb{F}_p[x]$ of the operator L.

If we can find $x' \in \mathbb{F}_p$ such that $\ell_r(x') \neq 0$, we can ensure this property by translating the origin to x'. To ensure that we can find x', we must make the following hypothesis, written **H**: ℓ_r does not vanish identically on $\mathbb{F}_p$.

LEMMA 3. *Given L of bidegree (d, r), testing whether* **H** *holds can be done in time $O(\mathsf{M}(d)) \subset \tilde{O}(d)$. If so, one can find x' such that $\ell_r(x') \neq 0$ and translate the coordinates' origin to x' in time $O(r\mathsf{M}(d)\log(d)) \subset \tilde{O}(rd)$.*

PROOF. Testing $\mathbf{H}$ amounts to verify whether $x^p - x$ divides ℓ_r. If $\deg(\ell_r) < p$, $\mathbf{H}$ obviously holds. Else, we have $p \le d$; then, it is enough to reduce ℓ_r modulo $x^p - x$, which takes time $O(\mathsf{M}(d))$.

If $\mathbf{H}$ holds, we know that we can find $x' \in \{0, \dots, \deg(\ell_r)\}$ such that $\ell_r(x') \neq 0$; so it is enough to evaluate ℓ_r at this set of points, which by [18, §10.1] takes time $O(\mathsf{M}(d)\log(d))$. Once x' is known, we shift all coefficients of L by x'. Using fast algorithms for polynomial shift [17], the time is $O(\mathsf{M}(d)\log(d))$ per coefficient; the conclusion follows. □

As a consequence, in all the following algorithms, we will assume that $\mathbf{H}$ holds. If not, one could actually work in a low-degree extension of $\mathbb{F}_p$ to find x'; we do not consider this generalization here.

3. POLYNOMIAL SOLUTIONS

We start with the study of the polynomial solutions of a linear differential equation; aside from its own interest, this question will arise in our algorithm for order two operators in Section 5.

THEOREM 1. *Let L be as in (2), with $r \le d$ and $r \le p$, and such that $\mathbf{H}$ holds. Then, one can test whether the equation $Lu = 0$ has non-zero solutions in $\mathbb{F}_p(x)$ in time*

$$\tilde{O}(d^\omega r^{1/2} p^{1/2} + d^{\omega+1} r^{\omega-1}).$$

If so, one can determine a basis of the solution set consisting of polynomials of degree at most $dp-1$ in extra time

$$\tilde{O}(d^{\omega+1} r p + d^2 r^{\omega+3} p).$$

The main point here is that for fixed d and r, *testing* the existence of solutions takes time $\tilde{O}(p^{1/2})$, whereas finding a basis of the solution space takes time $\tilde{O}(p)$.

In all this section, L is fixed, and the assumptions of Theorem 1 are satisfied. The assumptions on the relative order of magnitude of p, d, r help us obtain simple cost estimates and rule out some possible overlaps in indices modulo p. The assumption $r \le d$ is here mostly for convenience; the assumption $r \le p$ is necessary.

3.1 Degree bounds

Let $\mathcal{F}$ be the $\mathbb{F}_p(x^p)$-vector space of rational solutions of the equation $Lu = 0$. The following proposition proves a bound linear in p on the degree of a basis of $\mathcal{F}$. To our knowledge, such linear bounds were previously available only in two particular cases: (a) when the equation has a basis of polynomial solutions and under the additional hypotheses $0 \le \deg(\ell_0) - r \le p-1$ and $p \ge r$ [19, Th. 7]; (b) when $r = 2$ and the equation has exactly one nonzero polynomial solution [14, Lemma 10.1]. These bounds are respectively $(p-r)d + \binom{r}{2}$ for (a) and $\frac{1}{2}(p-1)(d-1)$ for (b). In the general case, the analysis in [11, 12] suggests a bound quadratic in p of type $p(p+d)$. Our result refines this approach.

PROPOSITION 1. *If $Lu = 0$ has at least one nonzero solution in $\mathbb{F}_p(x)$, then $\mathcal{F}$ admits a basis consisting of polynomial solutions of degree at most $pd-1$ each.*

PROOF. The map $\varphi_L : \mathbb{F}_p(x) \to \mathbb{F}_p(x)$ defined by $y \mapsto L(y)$ is $\mathbb{F}_p(x^p)$-linear. Let $\mathbf{M} \in \mathscr{M}_p(\mathbb{F}_p(x^p))$ be the matrix of this map with respect to the basis $(1, x, \dots, x^{p-1})$. Write $\mathbf{M} = (m_{i,j})_{0 \le i,j \le p-1}$ for some $m_{i,j}$ in $\mathbb{F}_p[x^p]$. Then, $u \in \mathbb{F}_p[x]$ is in $\mathcal{F}$ if and only if $\mathbf{M} \times [u_0 \cdots u_{p-1}]^t = 0$, with u_i in $\mathbb{F}_p[x^p]$ such that $u = u_0 + u_1 x + \cdots + u_{p-1} x^{p-1}$.

Since $L(x^i) = \sum_{j \le p-1} m_{i,j} x^j$ is a sum of p polynomials of pairwise distinct degrees $\deg(m_{i,j}) + j$, we deduce that for all i, j, $\deg(m_{i,j}) + j \le \deg(L(x^i))$.

Since $Lu = 0$ has a non-zero solution in $\mathbb{F}_p(x)$, it has also a non-zero solution in $\mathbb{F}_p[x]$, by clearing denominators. Let thus v be in $\mathbb{F}_p[x] \setminus \{0\}$ such that $Lv = 0$, or equivalently $\ell_0 v = -\sum_{1 \le j \le r} \ell_j v^{(j)}$. Since all terms in the right-hand side have degree at most $d + \deg(v) - 1$, we deduce that $\deg(\ell_0) \le d-1$. This implies that $L(x^i) = \ell_0 x^i + \sum_{1 \le j \le r} i \cdots (i-j+1) \ell_i x^{i-j}$ has degree at most $d+i-1$.

To summarize, for all $0 \le i, j \le p-1$, we obtain the inequality $\deg(m_{i,j}) \le (d-1) + (i-j)$. This implies that for any permutation σ of $\{0, \dots, p-1\}$,

$$\deg(\textstyle\prod_{i=0}^{p-1} m_{i,\sigma(i)}) = \sum_{i=0}^{p-1} \deg(m_{i,\sigma(i)}) \le p(d-1),$$

since the sum of the terms $i - \sigma(i)$ is zero. This implies that all minors of $\mathbf{M}$ have degree at most $p(d-1)$, since any term appearing in the expansion of such minors can be completed to form one of the form $\prod_{0 \le i \le p-1} m_{i,\sigma(i)}$.

The nullspace of $\mathbf{M}$ admits a basis $[\mathbf{v}_1, \dots, \mathbf{v}_k]$, all of whose entries are minors of $\mathbf{M}$. By what was said above, they all have degree at most $p(d-1)$. A basis of $\mathcal{F}$ is easily deduced: to $\mathbf{v}_i = [v_{i,0} \cdots v_{i,p-1}]^t$ corresponds the polynomial $v_i = v_{i,0} + \cdots + v_{i,p-1} x^{p-1}$. We deduce that $\deg(v_i) \le p - 1 + p(d-1) = pd - 1$, as claimed. □

3.2 Solutions of bounded degree

Let $\mathcal{G} \subset \mathbb{F}_p[x]$ be the $\mathbb{F}_p$-vector space of polynomial solutions of $Lu = 0$ of degree at most $pd-1$. We are interested in computing either the dimension of $\mathcal{G}$, or an $\mathbb{F}_p$-basis of it. In view of the former proposition, this will be sufficient to prove Theorem 1. Proposition 2 gives cost estimates for these tasks, adapting the algorithm in [1] and its improvements [6].

PROPOSITION 2. *Under the assumptions of Theorem 1, one can compute $\dim_{\mathbb{F}_p}(\mathcal{G})$ in time*

$$\tilde{O}(d^\omega r^{1/2} p^{1/2} + d^{\omega+1} r^{\omega-1}).$$

One can deduce a basis of $\mathcal{G}$ in extra time $\tilde{O}(d^{\omega+1} r p)$.

For r and d fixed, the main feature of this result is that the cost of computing the dimension of $\mathcal{G}$ is the sublinear $\tilde{O}(p^{1/2})$, whereas the cost of computing a basis of it is $\tilde{O}(p)$.

PROOF. Let $u_0, \dots, u_{pd-1}$ be unknowns and let u be the polynomial $u = \sum_{n < pd} u_n x^n$; for $n < 0$ or $n \ge pd$, we let $u_n = 0$. There exist $c_0, \dots, c_{d+r}$ in $\mathbb{F}_p[n]$, of degree at most r, such that for $n \ge 0$, the coefficient of degree n of

$$\ell_0(x) u + \cdots + \ell_r(x) u^{(r)} \tag{3}$$

is $C_n = c_0(n) u_{n-d} + \cdots + c_{d+r}(n) u_{n+r}$; note for further use that

$$C_{n-r} = c_0(n-r) u_{n-r-d} + \cdots + c_{d+r}(n-r) u_n. \tag{4}$$

The polynomial u is in $\mathcal{G}$ if and only if $C_n = 0$ for $0 \le n \le (p+1)d - 1$. Shifting indices, we obtain the system of linear equations $C_{n-r} = 0$, with $r \le n \le (p+1)d + r - 1$, in the unknowns $u_0, \dots, u_{pd-1}$.

The matrix of this system is band-diagonal, with a band of width $d+r+1$. In characteristic zero or large enough, one can

eliminate each unknown u_n, with $n \geq r$, using C_{n-r}. Here, some equations C_{n-r} become deficient, in the sense that the coefficient of u_n vanishes; this induces a few complications.

Outline of the computation. Since $\lambda = \ell_r(0)$ is not zero, $c_{d+r}(n)$ is the non-zero polynomial $\lambda(n+1)\cdots(n+r)$, and $c_{d+r}(n-r) = \lambda(n-(r-1))\cdots n$. Let then $R = [0,\ldots,r-1]$ be the set of roots of the latter polynomial. For $r \leq n \leq pd-1$, if $(n \bmod p)$ is not in R, then u_n is the highest-index unknown appearing with a non-zero coefficient in C_{n-r}; we can then eliminate it, by expressing it in terms of the previous u_m's.

The unknowns we cannot eliminate this way are u_n, with n in

$$A = [\ n \mid 0 \leq n \leq pd-1,\ (n \bmod p) \in R\ ];$$

the residual equations are $C_{n-r} = 0$, for n in $B = B_1 \cup B_2$, with

$$B_1 = [\ n \mid r \leq n \leq pd-1 \text{ and } (n \bmod p) \in R\ ]$$

and

$$B_2 = [\ n \mid pd \leq n \leq (p+1)d + r - 1\ ].$$

To determine the dimension of $\mathcal{G}$, and later on find a basis of it, we rewrite the residual equations using the residual unknowns.

For $n = ip + j$ in B_1, the unknowns present in C_{n-r} are $u_{ip+j-r-d},\ldots,u_{ip+j}$. Of those, only $u_{ip+j-r-d},\ldots,u_{ip-1}$ need to be rewritten in terms of $[u_n \mid n \in A]$; the others already belong to this set. Thus, it is enough to express all $u_{ip-r-d},\ldots,u_{ip-1}$ in terms of $[u_n \mid n \in A]$, for $1 \leq i < d$.

For n in B_2, the unknowns in C_{n-r} are $u_{n-r-d},\ldots,u_{pd-1}$ (the higher index ones are zero). So, it is enough to compute $u_{pd-r-d},\ldots,u_{pd-1}$ in terms of $[u_n \mid n \in A]$. This is thus the same problem as above, for index $i = d$.

Expressing all needed unknowns using A. Let $A' = [0,\ldots,pd-1] - A$. For n in A', one can rewrite the equation $C_{n-r} = 0$ as the first order recurrence

$$\begin{bmatrix} u_{n-r-d+1} \\ \vdots \\ u_n \end{bmatrix} = \mathbf{A}(n) \begin{bmatrix} u_{n-d-r} \\ \vdots \\ u_{n-1} \end{bmatrix} \tag{5}$$

with

$$\mathbf{A}(n) = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \ldots & 1 \\ -\frac{c_0(n-r)}{c_{d+r}(n-r)} & \ldots & \ldots & -\frac{c_{d+r-1}(n-r)}{c_{d+r}(n-r)} \end{bmatrix};$$

note that for $n \neq 0 \bmod p$, $\mathbf{A}(n+p) = \mathbf{A}(n)$. Let next $\mathbf{B}$ be the matrix factorial $\mathbf{A}(p-1)\cdots\mathbf{A}(r)$. Then we have the equalities, for $1 \leq i \leq d$:

$$\begin{bmatrix} u_{ip-r-d} \\ \vdots \\ u_{ip-1} \end{bmatrix} = \mathbf{B} \begin{bmatrix} u_{(i-1)p-d} \\ \vdots \\ u_{(i-1)p+r-1} \end{bmatrix}.$$

Note that $|A| = dr$; we let $\mathbf{u}$ be the $dr \times 1$ column-vector consisting of all u_n, for n in A. Let further $\mathbf{C}_0$ be the $(d+r) \times dr$ zero matrix. For $1 \leq i \leq d$, suppose that we have determined $(d+r) \times dr$ matrices $\mathbf{C}_1,\ldots,\mathbf{C}_{i-1}$ such that, for $1 \leq j < i$, we have

$$\begin{bmatrix} u_{jp-r-d} \\ \vdots \\ u_{jp-1} \end{bmatrix} = \mathbf{C}_j\mathbf{u} \quad \text{and} \quad \begin{bmatrix} u_{jp-r-d} \\ \vdots \\ u_{jp+r-1} \end{bmatrix} = \mathbf{D}_j\mathbf{u}, \tag{6}$$

with

$$\mathbf{D}_j = \begin{bmatrix} \mathbf{C}_j \\ \mathbf{0}_{r\times\ell_j}\ \mathbf{I}_r\ \mathbf{0}_{r\times\ell'_j} \end{bmatrix},\ \ell_j = (j-1)r \quad \text{and } \ell'_j = (d-j-1)r.$$

Letting $\mathbf{C}'_{i-1}$ be the matrix made of the last d rows of $\mathbf{C}_{i-1}$, we define

$$\mathbf{C}_i = \mathbf{B} \begin{bmatrix} \mathbf{C}'_{i-1} \\ \mathbf{0}_{r\times\ell_{i-1}}\ \mathbf{I}_r\ \mathbf{0}_{r\times\ell'_{i-1}} \end{bmatrix};$$

then, (6) is satisfied at index i as well.

Rewriting all residual equations using A. Combining all previous information, we obtain a matrix equality of the form $\mathbf{u}' = \mathbf{D}\mathbf{u}$, where $\mathbf{u}'$ is the column vector with entries $u_{ip-r-d},\ldots,u_{ip+r-1}$, for $1 \leq i \leq d$, and where $\mathbf{D}$ is the matrix obtained by stacking up $\mathbf{D}_1,\ldots,\mathbf{D}_d$.

We have seen that all indeterminates appearing in the residual equations C_{n-r}, with r in B, are actually in $\mathbf{u}'$. By evaluating the coefficients $c_0,\ldots,c_{d+r}$ at $n-r$, for n in B, we obtain the matrix $\mathbf{D}'$ of the residual equations, expressed in terms of the unknowns in $\mathbf{u}'$. Hence, the matrix $\mathbf{E} = \mathbf{D}'\mathbf{D}$ expresses the residual equations in terms of u_n, for n in A.

By construction, the dimension of $\mathcal{G}$ equals the dimension of the nullspace of $\mathbf{E}$. Knowing a basis of the nullspace of $\mathbf{E}$, one deduces a basis of $\mathcal{G}$ using (5), to compute all u_n for n in A'.

Cost analysis. By [6, Lemma 7], one can compute $\mathbf{B}$ in time $T_1 = O(d^\omega \mathsf{M}(r^{1/2}p^{1/2})\log(rp))$, which is $\tilde{O}(d^\omega r^{1/2}p^{1/2})$. Computing a matrix $\mathbf{C}_i$ requires one matrix multiplication of size $(d+r,d) \times (d,dr)$. In view of the inequality $r \leq d$, using block matrix multiplication, this can be done in time $O(d^\omega r)$. Thus, computing all needed matrices $\mathbf{C}_i$ takes time $T_2 = O(d^{\omega+1}r)$.

The matrix $\mathbf{D}$ has size $(d(d+2r) \times dr$; no more computations are needed to fill its entries. The matrix $\mathbf{D}'$ has size $d(r+1) \times d(d+2r)$. Its entries are obtained by evaluating $c_0,\ldots,c_{r+d}$ at all n in B. Since $\deg(c_i) \leq r$ and $|B| = d(r+1)$, this takes time $O(\mathsf{M}(dr)\log(dr))$ per polynomial. Since $r \leq d$, the total time is $T_3 = O(d^2\mathsf{M}(r)\log(r)) \in \tilde{O}(d^2r)$.

The matrix $\mathbf{E} = \mathbf{D}'\mathbf{D}$ has size $d(r+1) \times dr$; using block matrix multiplication with blocks of size dr, it can be computed in time $T_4 = O(d^{\omega+1}r^{\omega-1})$. A basis of its nullspace can be computed in time $T_5 = O(d^\omega r^\omega)$.

Given a vector $[u_n \mid n \in A]$ in the nullspace of $\mathbf{E}$, one can reconstruct $[u_n \mid 0 \leq n < pd]$ using (5). This first requires evaluating all coefficients of all equations C_{n-r}, for n in $A' = [0,\ldots,pd-1] - A$, which takes time $T_6 = O(d\mathsf{M}(s)\log(s))$, with $s = \max(r,pd)$.

Then, for a given $[u_n \mid n \in A]$ in the nullspace of $\mathbf{E}$, deducing $[u_n \mid 0 \leq n < pd]$ requires $|A'| < pd$ matrix-vector products in size $d+r$. The dimension of the nullspace is $O(dr)$; we process all vectors in the nullspace basis simultaneously, so that we are left to do pd matrix products in size $(d+r) \times (d+r)$ by $(d+r) \times dr$. The cost of each product is $O(d^\omega r)$, so the total cost is $T_7 = O(d^{\omega+1}rp)$.

Summing $T_1,\ldots,T_5$ proves the first part of the proposition. Adding to this T_6 and T_7 gives the second claim.

3.3 Proof of Theorem 1

Let $\mathcal{F}$ and $\mathcal{G}$ be as above. By Proposition 1, $\dim_{\mathbb{F}_p}(\mathcal{G}) = 0$ if and only if $\dim_{\mathbb{F}_p(x^p)}(\mathcal{F}) = 0$. Hence, the first estimate of Proposition 2 proves our first claim.

Suppose that $\dim_{\mathbb{F}_p}(\mathcal{G}) \neq 0$, and let $u_1, \ldots, u_k$ be an $\mathbb{F}_p$-basis of $\mathcal{G}$. Proposition 1 implies that $u_1, \ldots, u_k$ generates $\mathcal{F}$ over $\mathbb{F}_p(x^p)$. We deduce an $\mathbb{F}_p(x^p)$-basis $\mathscr{B}$ of $\mathcal{F}$ in a naive way: starting from $\mathscr{B} = [u_1]$, we successively try to add $u_2, \ldots$ to $\mathscr{B}$. Independence tests are performed at each step, using the following lemma.

LEMMA 4. *Given $u_1, \ldots, u_\ell$ in $\mathbb{F}_p[x]$ of degree less than pd, one can determine whether they are linearly independent over $\mathbb{F}_p(x^p)$ in time $\tilde{O}(\ell^{\omega+2}dp)$.*

PROOF. It suffices to compute their Wronskian determinant. The determinant of a matrix of size ℓ can be computed using $O(\ell^{\omega+1})$ sums and products [2]; since here all products can be truncated in degree ℓdp, the cost is $O(\ell^{\omega+1}\mathsf{M}(\ell dp))$. □

At all times, there are at most r elements in $\mathscr{B}$, so we always have $\ell \leq r+1$. Since we also have $k \leq dr$, the overall time is $\tilde{O}(d^2 r^{\omega+3} p)$, as claimed.

4. P-CURVATURE: FIRST ORDER

For first order operators, there is a closed form formula for the p-curvature. Let $L = \partial - u$, with u in $\mathbb{F}_p(x)$; then, by [25, Lemma 1.4.2], the p-curvature of L is the 1×1 matrix with entry $u^{(p-1)} + u^p$, where the first term is the derivative of order $p-1$ of u. In this case, we do not distinguish between the p-curvature and its unique entry.

The case of first order operators stands out as the only one where a cost polynomial in $\log(p)$ can be reached; this is possible since in this case, we only compute $O(d)$ non-zero coefficients. As per our convention, in the following statement, we take L not necessarily monic, but with polynomial coefficients.

THEOREM 2. *Given $L = a\partial - b$ in $\mathbb{F}_p[x]\langle\partial\rangle$ of bidegree $(d, 1)$ that satisfies* **H**, *one can compute its p-curvature in time $O(d\mathsf{M}(d)\log(p)) \subset \tilde{O}(d^2\log(p))$.*

PROOF. Since the p-curvature belongs to $\mathbb{F}_p(x^p)$, it suffices to compute its pth root. Computing the p-curvature itself requires no extra arithmetic operation, since taking p-powers is free over $\mathbb{F}_p$, as far as arithmetic operations are concerned. Hence, we claim that the rational function

$$\left(\left(\frac{b}{a}\right)^{(p-1)} + \left(\frac{b}{a}\right)^{p}\right)^{\frac{1}{p}} = \left(\left(\frac{b}{a}\right)^{(p-1)}\right)^{\frac{1}{p}} + \frac{b}{a}$$

can be computed in time $O(d\mathsf{M}(d)\log(p))$. Of course, the only non-trivial point is to compute $s = (u^{(p-1)})^{1/p}$, with $u = b/a$.

Observe that $a^p u^{(p-1)}$ is a polynomial of degree less than dp, so as is a polynomial of degree less than d. Hence, it is enough to compute the power series expansion $s \bmod x^d$. From this, we deduce the polynomial as by a power series multiplication in degree d, and finally s by division by a.

Let us write the power series expansion $u = \Sigma_{i\geq 0} u_i x^i$. Then, the series s equals $-\Sigma_{i\geq 0} u_{ip} x^i$, so it is enough to compute the coefficients $(u_{ip})_{i<d}$.

We start by computing the first coefficients $u_0, \ldots, u_{d-1}$ by power series division, in time $O(\mathsf{M}(d))$. From these initial conditions, the coefficients $u_p, \ldots, u_{p+d-1}$ can be deduced for $O(\mathsf{M}(d)\log(p))$ operations using binary powering techniques, see [16] or [3, Sect. 3.3.3]. Iterating this process d times, we obtain the values $u_{ip}, \ldots, u_{ip+d-1}$, for $i < d$, in time $O(d\mathsf{M}(d)\log(p))$. □

As an aside, note that by Lemma 2, a rational function u is a logarithmic derivative in $\mathbb{F}_p(x)$ if and only if $u^{(p-1)} + u^p = 0$. This point also forms the basis of Niederreiter's algorithm for polynomial factoring [22].

5. P-CURVATURE: SECOND ORDER

For second order operators, it is possible to exploit a certain linear differential system satisfied by the entries of the p-curvature matrix: already in [15, 26], one finds a third order linear differential equation satisfied by an anti-diagonal entry of the p-curvature, for the case of operators of the form $\partial^2 + s$, or more generally $\partial^2 + r\partial + s$, when $r^{(p-1)} + r^p = 0$.

In this section, we let L have the form $v\partial^2 + w\partial + u$, with u, v, w in $\mathbb{F}_p[x]$ of degree at most d. We assume that $d \geq 2$ and $p > 2$, and that **H** holds (we do not repeat these assumptions in the theorems); we let $\mathbf{A}$ be the companion matrix of L and let $\mathbf{A}_p$ be its p-curvature.

We give partial results regarding the computation of $\mathbf{A}_p$: we give algorithms of cost $\tilde{O}(p^{1/2})$ or $\tilde{O}(p)$ to test properties of $\mathbf{A}_p$, or compute it in some cases, up maybe to some indeterminacy. Though these algorithms do not solve all questions, they are still substantially faster than the ones for the general case in the next section.

The trace of the p-curvature. We start by an easy but useful consequence of the result of the previous section: the trace of $\mathbf{A}_p$ can be computed fast.

THEOREM 3. *One can compute the trace τ of $\mathbf{A}_p$ in time $O(d\,\mathsf{M}(d)\log(p))$.*

PROOF. The p-curvature of a determinant connection is the trace of the p-curvature of the original connection [21, 28]. Concretely, this means that the trace of $\mathbf{A}_p$ is equal to the p-curvature of $v\partial + w$. By Theorem 2, it can be computed in time $O(d\,\mathsf{M}(d)\log(p))$. □

Testing nilpotence. As a consequence of the previous theorems, we obtain a decision procedure for nilpotence.

COROLLARY 1. *One can decide whether $\mathbf{A}_p$ is nilpotent in time $\tilde{O}(d^\omega p^{1/2} + d^{\omega+1})$.*

PROOF. The p-curvature $\mathbf{A}_p$ is nilpotent if and only if its trace and determinant are both zero. By Theorem 3, the condition on the trace can be checked in time logarithmic in p. By Lemma 2, the second condition $\det(\mathbf{A}_p) = 0$ is equivalent to the fact that $Lu = 0$ has a non-zero solution, which can be tested in the requested time by Theorem 1. □

The eigenring. To state our further results, we need an extra object: the *eigenring* $\mathcal{E}(L)$ of L. This is the set of matrices $\mathbf{B}$ in $\mathscr{M}_2(\mathbb{F}_p(x))$ that satisfy the matrix differential equation

$$\mathbf{B}' = \mathbf{BA} - \mathbf{AB} \tag{7}$$

(our definition differs slightly from the usual one in the sign convention). By construction, the eigenring $\mathcal{E}(L)$ is a $\mathbb{F}_p(x^p)$-vector space of dimension at most 4, which contains the p-curvature $\mathbf{A}_p$. Then, we let γ be its dimension over $\mathbb{F}_p(x^p)$; we will prove later on that γ is in $\{2, 4\}$.

Let further $\mathcal{F}$ be the set of solutions of $Ly = 0$ in $\mathbb{F}_p(x)$ and let β be its dimension over $\mathbb{F}_p(x^p)$. Then, our main results are the following.

THEOREM 4. *One can compute in time $\tilde{O}(d^{\omega+1}p)$:*

1. *the dimensions $\gamma \in \{2, 4\}$ of $\mathcal{E}(L)$ and $\beta \in \{0, 1, 2\}$ of $\mathcal{F}$;*
2. *$\mathbf{A}_p$, if $\gamma = 4$ or $\beta = 2$.*
3. *$\mathbf{A}_p$, up to a multiplicative constant in $\mathbb{F}_p[x^p]$ of degree at most pd, if $\gamma = 2$ and the trace $\tau = 0$.*
4. *a list of two candidates for $\mathbf{A}_p$, if $\gamma = 2$ and $\beta = 1$.*

The rest of this section is devoted to prove this theorem.

The dimension of the eigenring. The following lemmas restrict the possible dimension γ of $\mathcal{E}(L)$.

LEMMA 5. *If $\mathbf{A}_p$ has the form $\lambda \mathbf{I}_2$, then $\gamma = 4$.*

PROOF. In this case, the commutator of $\mathbf{A}_p$ in $\mathscr{M}_2(\mathbb{F}_p(x))$ is $\mathscr{M}_2(\mathbb{F}_p(x))$ itself, so it has dimension 4 over $\mathbb{F}_p(x)$. Then, [12, Prop. 3.5] implies that $\mathcal{E}(L)$ has dimension 4 over $\mathbb{F}_p(x^p)$. □

LEMMA 6. *Either $\gamma = 2$, or $\gamma = 4$. In the second case, $\mathbf{A}_p$ is equal to $\frac{\tau}{2}\mathbf{I}_2$, where τ is the trace of $\mathbf{A}_p$.*

PROOF. Corollary 1 of [12] shows that if the minimal and characteristic polynomials of $\mathbf{A}_p$ coincide, then $\mathcal{E}(L)$ equals $\mathbb{F}_p(x^p)[\mathbf{A}_p]$. In this case, $\mathbb{F}_p(x^p)[\mathbf{A}_p]$ has dimension 2 over $\mathbb{F}_p(x^p)$. Else, the minimal polynomial of $\mathbf{A}_p$ must have degree 1, so $\mathbf{A}_p$ is necessarily equal to $\frac{\tau}{2}\mathbf{I}_2$, and we are under the assumptions of the previous lemma. □

Computing γ and β. The equality (7) gives a system of four linear differential equations of order one for the entries $b_{1,1}, \ldots, b_{2,2}$ of $\mathbf{B}$. An easy computation shows that (7) is equivalent to the system

$$\begin{aligned} v^3 b'''_{2,1} + Ab'_{2,1} + Bb_{2,1} &= 0, && (8)\\ v^2 b_{1,2} + Rb''_{2,1} + Sb'_{2,1} + Tb_{2,1} &= 0, && (9)\\ v(b_{1,1} - b_{2,2}) + vb'_{2,1} - wb_{2,1} &= 0, && (10)\\ b'_{1,1} + b'_{2,2} &= 0, && (11) \end{aligned}$$

where A, B, R, S, T belong to $\mathbb{F}_p[x]$, and are given by

$$A = v(-2w'v + 2wv' + 4uv - w^2),$$

$$B = vw(v'' - w') + v'w(w - 2v') + 2u'v^2 - 2vuv' - w''v^2 + 2v'w'v$$

and

$$R = v^2/2, \quad S = -vw/2, \quad T = v'w/2 - vw'/2 + uv.$$

Since Equation (11) is equivalent to $b_{1,1} + b_{2,2} \in \mathbb{F}_p(x^p)$, we readily deduce that the dimension γ of $\mathcal{E}(L)$ equals $\gamma' + 1$, where γ' is the dimension of the solution-set of (8).

Computing both γ and β can be done using Theorem 1, with respectively $r = 3$ or $r = 2$, and in degree respectively at most $4d$ or d. This proves point 1 of Theorem 4.

If $\gamma = 4$, we are in the second case of Lemma 6. Since the trace can be computed in time $\tilde{O}(d^2 \log(p))$ by Theorem 3, point 2 of Theorem 4 is established in this case. If $\beta = 2$, then $\mathbf{A}_p$ is zero by Lemma 2, so point 2 of Theorem 4 is established as well.

Eigenrings of dimension 2. The rest of this section is devoted to analyze what happens if $\mathcal{E}(L)$ has dimension $\gamma = 2$ over $\mathbb{F}_p(x^p)$, so that the dimension γ' of the solution-space of (8) is 1. In this case, the information provided by the eigenring is not sufficient to completely determine the p-curvature. However, it is still possible to recover some useful partial information. To fix notation, we write the p-curvature as

$$\mathbf{A}_p = \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \end{bmatrix}.$$

LEMMA 7. *If $\gamma = 2$, $F = v^p f_{2,1}$ is a nonzero polynomial solution of degree at most pd of Equation (8).*

PROOF. Since the p-curvature $\mathbf{A}_p$ belongs to the eigenring, its entries $f_{1,1}, \ldots, f_{2,2}$ satisfy (8) to (11). Lemma 1 shows that $F = v^p f_{2,1}$ is a polynomial solution of degree at most pd of Equation (8). Moreover, F cannot be 0, since otherwise Equations (9) to (11) would imply that $\mathbf{A}_p$ has the form $\lambda \mathbf{I}_2$ for some λ in $\mathbb{F}_p(x^p)$. By Lemma 5, this would contradict the assumption $\gamma = 2$. □

LEMMA 8. *Suppose that $\gamma = 2$ and let $u \in \mathbb{F}_p[x]$ be the nontrivial polynomial solution of minimal degree of Equation (8). There exists a nonzero polynomial c in $\mathbb{F}_p[x^p]$ of degree at most pd, such that the entries of $\mathbf{A}_p$ are given by*

$$\begin{aligned} f_{1,1} &= \frac{1}{2}\left(\tau + \frac{c}{v^p}\left(\frac{w}{v}u - u'\right)\right),\\ f_{1,2} &= -\frac{c}{v^{p+2}}\left(Ru'' + Su' + Tu\right),\\ f_{2,1} &= \frac{c}{v^p}u,\\ f_{2,2} &= \frac{1}{2}\left(\tau - \frac{c}{v^p}\left(\frac{w}{v}u - u'\right)\right). \end{aligned}$$

PROOF. By Lemma 7, the polynomials F and u both satisfy Equation (8); thus, they differ by an element c in $\mathbb{F}_p(x^p)$. Moreover, the minimality of the degree of u implies that $c = F/u$ actually belongs to $\mathbb{F}_p[x^p]$ and has degree at most $\deg(F) \leq pd$. The rest of the assertion follows from the relations $F = v^p f_{2,1}$, $\tau = f_{1,1} + f_{2,2}$ and the equalities (9) and (10). □

Concluding the proof of Theorem 4. To conclude, we consider two special cases. If $\tau = 0$, as in [26], the previous lemma shows that $\mathbf{A}_p$ is known up to a multiplicative constant in $\mathbb{F}_p[x^p]$, as soon as the polynomial u has been computed. In this case, Corollary 1 shows that one can compute a non-zero solution u_0 of (8) in the required time. The minimal degree solution u by clearing out the factor in $\mathbb{F}_p[x^p]$ in u_0 using [18, Ex. 14.27], in negligible time $O(\mathsf{M}(dp)\log(dp)) \subset \tilde{O}(dp)$, and the substitution in the former formulas takes time $\tilde{O}(dp)$ as well.

If $\beta = 1$, L has a non-trivial polynomial solution, so by Lemma 2 the determinant of $\mathbf{A}_p$ is zero; the additional equation $f_{1,1}f_{2,2} = f_{1,2}f_{2,1}$, in conjunction with the formulas in Proposition 7, uniquely determines the polynomial c^2 and thus leaves us with only two possible candidates for $\mathbf{A}_p$.

6. P-CURVATURE: HIGHER ORDER

In this final section, we study operators of higher order, and we prove that the p-curvature can be computed in time subquadratic in p.

THEOREM 5. *Given L in $\mathbb{F}_p[x]\langle\partial\rangle$ of bidegree (d, r), one can compute its p-curvature in time*

$$\tilde{O}(r^\omega d^2 p^{2\omega/3} + r^\omega d p^{1+\omega/3}).$$

Hence, the exponent in p is $1+\omega/3 < 1.79 < 2$; in the best possible case $\omega = 2$, we would obtain an exponent $5/3$ in p, unfortunately still not optimal.

As a result of independent interest, we also give an algorithm for computing the image of a matrix of rational functions by an differential operator similar in spirit to Brent and Kung's algorithm for modular composition [9]; to our knowledge, no prior non-trivial algorithm existed for this task.

6.1 Preliminaries

Euler's operator. Besides operators in the usual variables x, ∂, it will also be convenient to consider operators in $\mathbb{F}_p(x)\langle\theta\rangle$ or $\mathscr{M}_n(\mathbb{F}_p(x))\langle\theta\rangle$, where θ is Euler's operator $x\partial$, which satisfies the commutation rule $\theta x = x\theta + x$. To avoid confusion, we may say that L has bidegree (d, r) *in* ∂ or *in* θ, if L is written respectively on the bases (x, ∂) or (x, θ).

Conversion. Given an operator L in $\mathscr{M}_n(\mathbb{F}_p[x])\langle\partial\rangle$ of bidegree (d, r), $L' = x^r L$ can be rewritten as an operator in θ with polynomial coefficients. The operator L' has bidegree $(d+r, r)$ in θ. By [5, Section 3.3], computing the coefficients of L' takes time $O(n^2(d+r)\mathsf{M}(r)\log(r))$. Since representing all coefficients of L' requires $O(n^2(d+r)r)$ elements, this is quasi-linear, up to logarithmic factors.

Multiplication. Next, we give an algorithm for the multiplication of operators with rational coefficients of a special type, inspired by that of [5] (which handles polynomial coefficients). The algorithm relies on an evaluation / interpolation idea originally due to [24], and introduces fast matrix multiplication to solve the problem.

LEMMA 9. *Let $b \in \mathbb{F}_p[x]$ be of degree at most d, with $b(0) \neq 0$ and let γ, μ be in $\mathbb{F}_p(x)\langle\partial\rangle$, with*

$$\gamma = \sum_{j=0}^{h} \frac{g_j}{b^{h-j}}\partial^j, \quad \mu = \sum_{j=0}^{h} \frac{m_j}{b^{h-j}}\partial^j,$$

where g_j and $m_j \in \mathbb{F}_p[x]$ have degrees at most $d(h-j)$. Then if $2h \leq p-1$, one can compute $\eta = \gamma\mu$ in time $O(h^\omega d^2)$.

PROOF. Define $\eta^\star = b^{2h}\eta, \gamma^\star = b^h\gamma$ and $\mu^\star = b^h\mu$. A quick verification shows that these operators are in $\mathbb{F}_p[x]\langle\partial\rangle$, of respective bidegrees bounded by $(2dh, 2h)$, (dh, h) and (dh, h).

We first compute $\gamma(x^j)$ and $\mu(x^j)$ mod $x^{2dh+h+1}$ for $j \leq 2h$. This is done by computing the corresponding values of $\gamma^\star$ and $\mu^\star$, and dividing the results by b^h. The former computation takes time $O(\mathsf{M}(dh^2))$ using algorithm Eval of [5]; the latter $O(h\mathsf{M}(dh))$ by Newton iteration for power series division. Our assumption $2h \leq p-1$ ensures that divisions performed in the evaluation algorithm (and in the interpolation below) are well-defined.

From the values of γ and μ, the values $\eta(x^j)$ mod x^{2dh+1} are obtained as in [5, Th. 3]; the cost is $O(h^\omega d^2)$. We can then compute the values of $\eta^\star$ in time $O(h\mathsf{M}(dh))$ by fast polynomial multiplication. Knowing its values, we recover $\eta^\star$ using algorithm Interpol of [5]; this takes time $O(\mathsf{M}(dh^2))$. Finally, we deduce η by division by b^{2h}; this takes time $O(h\mathsf{M}(dh)\log(dh))$, using fast gcd computation. □

Left and right forms. Let $L \in \mathscr{M}_n(\mathbb{F}_p[x])\langle\theta\rangle$ have the form

$$L = \boldsymbol{\ell}_0(x) + \boldsymbol{\ell}_1(x)\theta + \cdots + \boldsymbol{\ell}_r(x)\theta^r,$$

with $\boldsymbol{\ell}_i \in \mathscr{M}_n(\mathbb{F}_p[x])$ of degrees at most d. It can be rewritten

$$L = \boldsymbol{\ell}_0^\star(x) + \theta^\star\boldsymbol{\ell}_1(x) + \cdots + \theta^r\boldsymbol{\ell}_r^\star(x),$$

with $\boldsymbol{\ell}_i^\star$ in $\mathscr{M}_n(\mathbb{F}_p[x])$ of degrees at most d as well. The former expression will be called the *right-form* of L; the latter is its *left-form*.

LEMMA 10. *Let L have bidegree (d, r) in $\mathscr{M}_n(\mathbb{F}_p[x])\langle\theta\rangle$, given in its right-form (resp. in its left-form). Then one can compute its left-form (resp. right-form) in time $O(n^2 d\mathsf{M}(r)\log(r)) \subset \tilde{O}(n^2 dr)$.*

PROOF. We prove one direction only; the other is similar. Given the right-form of L, we can (without performing any operation) rewrite $L = \sum_{j\leq d} x^j L_i$, where L_j has constant coefficients and order at most r. Since $x^j L_i(\theta) = L_i(\theta - j)x^j$, the result follows by using algorithms for polynomial shift by j [17]. □

The number of elements needed to represent L in either left- or right-form is $O(n^2 dr)$, so the previous algorithm is quasi-linear, up to logarithmic factors.

6.2 Evaluation

For L in $\mathscr{M}_r(\mathbb{F}_p[x])\langle\partial\rangle$ or $\mathscr{M}_r(\mathbb{F}_p[x])\langle\theta\rangle$ and $\mathbf{A}$ in $\mathscr{M}_r(\mathbb{F}_p(x))$, $L\mathbf{A}$ denotes the matrix in $\mathscr{M}_r(\mathbb{F}_p(x))$ obtained by applying L to $\mathbf{A}$. In this subsection, we give cost estimates on the computation of $L\mathbf{A}$.

The polynomial case. We start with the case of an operator with polynomial coefficients, which we apply to a matrix with polynomial entries. We use an operator in θ, since this makes operations slightly more convenient than in ∂. As in Section 3, we make assumptions on the relative sizes of the input parameters (here $\delta, \rho, \varepsilon$), for simplicity's sake.

LEMMA 11. *Given $L \in \mathscr{M}_r(\mathbb{F}_p[x])\langle\theta\rangle$ of bidegree (δ, ρ) and $\mathbf{E} \in \mathscr{M}_r(\mathbb{F}_p[x])$ of degree ε, one can compute $L\mathbf{E}$ in time $\tilde{O}(r^\omega\rho\varepsilon^{\omega-2}\delta^{3-\omega})$, assuming $\delta \in O(\varepsilon)$ and $\varepsilon \in O(\rho^{1/2}\delta)$.*

The cost can be rewritten as $\tilde{O}(r^\omega\rho\,\varepsilon\,(\delta/\varepsilon)^{3-\omega})$. Since $\omega \leq 3$ and $\delta \in O(\varepsilon)$, this is always better than $\tilde{O}(r^\omega\rho\varepsilon)$: the cost ranges from $\tilde{O}(r^\omega\rho\delta)$ for a hypothetical $\omega = 2$ to $\tilde{O}(r^\omega\rho\varepsilon)$ for $\omega = 3$. As a matter of comparison, let us write

$$L = \textstyle\sum_{i\leq\rho} \boldsymbol{\ell}_i\theta^i, \quad \boldsymbol{\ell}_i \in \mathscr{M}_r(\mathbb{F}_p[x]).$$

Computing $L\mathbf{E}$ naively amounts to computing all $\theta^i\mathbf{E}$ for $i \leq \rho$, multiplying them by the respective coefficients $\boldsymbol{\ell}_i$, and summing the results; the cost is in $\tilde{O}(r^\omega\rho\varepsilon)$, so our estimate is better.

PROOF. Our result is achieved using a baby steps / giant steps strategy inspired by Brent-Kung's algorithm for power series composition [9]. Let $k = \lfloor\rho^{1/2}\rfloor$ and $h = \lceil\rho/k\rceil$. First, we rewrite L in left-form, as

$$L = \textstyle\sum_{i\leq\rho} \theta^i\boldsymbol{\ell}_i^\star(x);$$

by Lemma 10, the cost is $T_1 = O(r^2\delta\mathsf{M}(\rho)\log(\rho)) \subset \tilde{O}(r^2\delta\rho)$. Next, L is cut into h slices of the form

$$L_0 + \theta^k L_1 + \cdots + \theta^{(h-1)k}L_{h-1}, \quad i.e. \quad L = \textstyle\sum_{j<h} \theta^{jk}L_j.$$

Each L_j has order less than k and can be written as

$$L_j = \textstyle\sum_{i<k} \theta^i\boldsymbol{\ell}_{jk+i}^\star(x),$$

where for $jk+i > \rho$, $\boldsymbol{\ell}^\star_{jk+i}$ is zero. Finally, we rewrite each L_j in right-form:

$$L_j = \sum_{i<k} \boldsymbol{\ell}^\dagger_{j,i}(x)\theta^i, \tag{12}$$

where all $\boldsymbol{\ell}^\dagger_{j,i}$ have degree at most δ. By Lemma 10, the cost is $T_2 = O(hr^2\delta\mathsf{M}(k)\log(k))$, which is in $\tilde{O}(r^2\delta\rho)$ as before. To apply L to $\mathbf{E}$, we first compute the baby steps

$$\mathbf{E}_0 = \mathbf{E},\ \mathbf{E}_1 = \theta\mathbf{E},\ \ldots,\ \mathbf{E}_{k-1} = \theta^{k-1}\mathbf{E};$$

then, we deduce all $L_j\mathbf{E}$, for $j < h$; finally, we do the giant steps

$$L\mathbf{E} = \sum_{j<h} \theta^{jk} L_j\mathbf{E}.$$

All $\mathbf{E}_i$ can be computed in time $T_3 = O(r^2\rho^{1/2}\varepsilon)$, by successive applications of θ. The cost T_4 of deducing the polynomials $L_j\mathbf{E}$ is detailed below. Finally, one recovers $L\mathbf{E}$ by first computing all $\theta^{jk}L_j\mathbf{E}$, for $j < h$, and then summing them. Since $\theta^i(x^j) = j^i x^j$, θ^{jk} can be applied to $L_j\mathbf{E}$ in time $O(r^2\varepsilon\log(\rho))$, so the total cost of this final step is $T_5 = O(r^2\varepsilon h\log(\rho)) \subset \tilde{O}(r^2\rho^{1/2}\varepsilon)$.

It remains to compute all $L_j\mathbf{E}$, given all $\mathbf{E}_i$; we compute them all at once. In view of Equation (12), we have

$$L_j\mathbf{E} = \sum_{i<k} \boldsymbol{\ell}^\dagger_{j,i}\mathbf{E}_i,$$

where the $\mathbf{E}_i$ are known. We cut $\mathbf{E}_i$ into slices of length δ:

$$\mathbf{E}_i = \sum_{u<s} \mathbf{E}_{i,u}x^{\delta u},$$

where $\mathbf{E}_{i,u}$ has degree less than δ and $s = \lceil \varepsilon/\delta \rceil \le 2\varepsilon/\delta$. This gives

$$L_j\mathbf{E} = \sum_{i<k} \boldsymbol{\ell}^\dagger_{j,i} \sum_{u<s} \mathbf{E}_{i,u}x^{\delta u} = \sum_{u<s} x^{\delta u} \sum_{i<k} \boldsymbol{\ell}^\dagger_{j,i}\mathbf{E}_{i,u}.$$

We will compute all inner sums

$$\sum_{i<k} \boldsymbol{\ell}^\dagger_{j,i}\mathbf{E}_{i,u}$$

at once, for $j < h$ and $u < s$; from this, one can recover all $L_j\mathbf{E}$ in time $O(r^2\varepsilon\rho^{1/2})$.

The computation of these sums amounts to perform a $(h\times k)\times(k\times s)$ matrix multiplication, with entries that are polynomial matrices of size r and degree at most δ. Since $\varepsilon \in O(\rho^{1/2}\delta)$, we have $s \in O(\rho^{1/2})$. Hence, we divide the previous matrices into blocks of size s and we are left to do a $(O(\rho^{1/2}/s)\times O(\rho^{1/2}/s))\times(O(\rho^{1/2}/s)\times O(1))$ product of such blocks, where $\rho^{1/2}/s$ is lower-bounded by a constant. Multiplying a single block takes time $O(r^\omega s^\omega \mathsf{M}(\delta))$, so the total time T_4 is $O(r^\omega\rho s^{\omega-2}\mathsf{M}(\delta))$, which is $\tilde{O}(r^\omega\rho\varepsilon^{\omega-2}\delta^{3-\omega})$.

The conclusion of Lemma 11 comes after a few simplifications, which shows that the dominant cost is T_4, for the final linear algebra step. □

The rational function case. Next, we study the application of an operator to a matrix of rational functions $\mathbf{A}$ (we make some simplifying assumptions on the denominators in $\mathbf{A}$, which will be satisfied in the cases in §6.3 where we apply this result). Besides, our operator is now in $\mathscr{M}_r(\mathbb{F}_p[x])\langle\partial\rangle$ rather than in $\mathscr{M}_r(\mathbb{F}_p[x])\langle\theta\rangle$.

Because of the larger number of parameters appearing in the construction, the cost estimate unfortunately becomes more complex than in the polynomial case.

LEMMA 12. *Let $L \in \mathscr{M}_r(\mathbb{F}_p[x])\langle\partial\rangle$ be of bidegree (δ,ρ). Let $\mathbf{A} \in \mathscr{M}_r(\mathbb{F}_p(x))$ be of the form $\mathbf{B}/b^\kappa$, with $b \in \mathbb{F}_p[x]$ of degree at most d and $\mathbf{B} \in \mathscr{M}_r(\mathbb{F}_p[x])$ of degree at most κd. Define*

$$\delta' = \delta + \rho \quad \text{and} \quad \varepsilon = (\kappa+\rho)d + \delta' + 1.$$

If $b(0) \neq 0$ and $\varepsilon \in O(\rho^{1/2}\delta')$, one can compute $L\mathbf{A}$ in time $\tilde{O}(r^\omega\rho\varepsilon^{\omega-2}\delta'^{3-\omega})$.

PROOF. Let $L' = x^\rho L$. Given L as an operator in ∂, we saw that we can write L' as an operator in θ, of bidegree (δ',ρ); the coefficients of L' in θ can be computed in time $O(r^2\delta\mathsf{M}(\rho)\log(\rho)) \subset \tilde{O}(r^2\delta\rho)$. To conclude, it is enough to compute $L'\mathbf{A}$, since then $L\mathbf{A}$ is deduced by a division by x^ρ, which is free.

For any $i \ge 0$, $\theta^i\mathbf{A}$ has the form $\mathbf{B}_i/b^{\kappa+i}$, with $\mathbf{B}_i$ in $\mathscr{M}_r(\mathbb{F}_p[x])$ of degree at most $(\kappa+i)d$. Thus, $L'\mathbf{A}$ has the form $\mathbf{B}^\star/b^{\kappa+\rho}$, with $\mathbf{B}^\star \in \mathscr{M}_r(\mathbb{F}_p[x])$ of degree less than ε, with $\varepsilon = (\kappa+\rho)d + \delta' + 1$.

Knowing $L'\mathbf{A} \bmod x^\varepsilon$, one can recover the numerator matrix $\mathbf{B}^\star$ through multiplication by $b^{\kappa+\rho}$; a gcd computation finally gives $L'\mathbf{A}$ in normal form. These latter steps take time $O(r^2\mathsf{M}(\varepsilon)\log(\varepsilon)) \subset \tilde{O}(r^2\varepsilon)$.

Since $b(0) \neq 0$, the matrix $\mathbf{E} = \mathbf{A} \bmod x^\varepsilon$ is well-defined; it can be computed in time $O(r^2\mathsf{M}(\varepsilon)) \subset \tilde{O}(r^2\varepsilon)$ by power series division. Lemma 11 gives complexity estimates for computing $L'\mathbf{E} \bmod x^\varepsilon$. Since this matrix coincides with $L'\mathbf{A}$ modulo x^ε, this concludes the proof of the lemma, as all previous costs are negligible compared to the one of Lemma 11. □

6.3 Computing the p-curvature

Let L be in $\mathbb{F}_p[x]\langle\partial\rangle$ of bidegree (d,r) and let $\mathbf{A}$ be its companion matrix. We define the operator $\Lambda \in \mathscr{M}_r(\mathbb{F}_p(x))\langle\partial\rangle$ as $\Lambda = \partial + \mathbf{A}$; thus, as pointed out in the introduction, the p-curvature of L is obtained by applying Λ^{p-1} to $\mathbf{A}$.

To obtain a cost better than $O(p^2)$, we first compute a high enough power $\Lambda' = \Lambda^k$ of Λ; then, we apply Λ' to $\mathbf{A}$ k' times, with $k' \simeq (p-1)/k$. Since $p-1$ may not factor exactly as kk', a few iterations of this process are needed.

Computing Λ^k. Let $\ell = \ell_r \in \mathbb{F}_p[x]$ be the leading coefficient of L. Then, Λ has the form $\partial + \boldsymbol{\lambda}/\ell$, with $\boldsymbol{\lambda}$ in $\mathscr{M}_r(\mathbb{F}_p[x])$ and $\ell \in \mathbb{F}_p[x]$ of degree at most d and $\ell(0) \neq 0$. More generally, for $k \ge 0$, we can write Λ^k as

$$\Lambda^k = \sum_{j=0}^k \boldsymbol{\lambda}_{k,j}\partial^j, \quad \text{with} \quad \boldsymbol{\lambda}_{k,j} = \frac{\boldsymbol{\ell}_{k,j}}{b^{k-j}}$$

and $\boldsymbol{\ell}_{k,j}$ in $\mathscr{M}_r(\mathbb{F}_p[x])$ of degree at most $d(k-j)$.

LEMMA 13. *If $k \le p-1$, one can compute Λ^k in time $O(r^\omega k^\omega d^2)$.*

PROOF. We use a divide-and-conquer scheme. Let $h = \lfloor k/2 \rfloor$; we assume for simplicity that $k = 2h$; if k is odd, an extra (cheaper) multiplication by Λ is needed. We assume that Λ^h is known, and we see it as an $r\times r$ matrix with entries that are scalar operators; hence, to compute Λ^k, we do $O(r^\omega)$ products of such scalar operators. All these products have the form $\eta = \gamma\mu$ of the form seen in Lemma 9, so each of their costs is $O(h^\omega d^2) = O(k^\omega d^2)$. □

Computing $\Lambda^{kk'}\mathbf{A}$. We fix $k \le p$, and we compute the operators $\Gamma = \Lambda^k$ and $\Gamma' = d^k\Gamma$. Writing $k' = \lfloor (p-1)/k \rfloor$, we compute the sequence

$$\mathbf{A}_{(1)} = \mathbf{A}, \quad \mathbf{A}_{(i)} = \Gamma\mathbf{A}_{(i-1)}, \quad i = 2,\ldots,k',$$

so that $\mathbf{A}_{(k')} = \Lambda^{kk'}\mathbf{A}$. Thus, we have $\mathbf{A}_{(k')} = \mathbf{A}_{kk'}$, where the latter matrix is defined in Equation (1). Using the subroutines seen before, a quick analysis not reproduced here shows that the optimal choice is $k = \lfloor (p-1)^{2/3} \rfloor$. Then, computing Γ takes time $O(r^\omega k^\omega d^2) = O(r^\omega p^{2\omega/3} d^2)$ by Lemma 13.

By Lemma 1, each matrix $\mathbf{A}_{(i)}$ has the form $\mathbf{B}_{(i)}/b^{ik}$, with $\mathbf{B}_{(i)} \in \mathscr{M}_r(\mathbb{F}_p[x])$ of degree at most dki. Given $\mathbf{A}_{(i)}$, we compute $\mathbf{A}_{(i+1)}$ by first applying Γ' to $\mathbf{A}_{(i)}$ and dividing the result by d^k.

The first step, applying Γ', is the more costly. We obtain its cost by applying Lemma 12, with $(\delta, \rho) = (dk, k)$ and $\kappa = ik$. Then, we have $\delta' \in O(dk)$ and $\varepsilon \in O(idk)$. For all $i \leq k'$, we are under the assumptions of that lemma; after a few simplifications, the cost becomes $\tilde{O}(r^\omega i^{\omega-2} dk^2)$. Summing over all $i \leq k'$, we obtain an overall cost of $\tilde{O}(r^\omega {k'}^{\omega-1} dk^2)$. Taking into account that $k \in O(p^{2/3})$ and $k' \in O(p^{1/3})$, this finally gives a cost of $\tilde{O}(r^\omega d p^{1+\omega/3})$ for computing $\Lambda^{kk'}\mathbf{A}$.

Computing the p-curvature. The definitions of k, k' imply that $p - (p-1)^{2/3} \leq kk' \leq p-1$. To obtain the p-curvature $\Lambda^{p-1}\mathbf{A}$, we iterate the previous process, replacing the required number of steps $p-1$ by $p-1-kk'$, until the required number of steps is $O(1)$. Since $p-1-kk' \leq (p-1)^{2/3}$, it takes $O(\log\log p)$ iterations; hence, the overall time is still in $\tilde{O}(r^\omega d p^{1+\omega/3})$.

Acknowledgments. We wish to acknowledge financial support from the French National Agency for Research (ANR Project "Gecko"), the joint Inria-Microsoft Research Centre, NSERC and the Canada Research Chair program.

7. REFERENCES

[1] S. A. Abramov, M. Bronstein, and M. Petkovšek. On polynomial solutions of linear operator equations. In *ISSAC'95*, pages 290–296. ACM Press, 1995.

[2] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18(3):147–150, 1984.

[3] A. Bostan. *Algorithmique efficace pour des opérations de base en calcul formel*. PhD thesis, École polytechnique, 2003.

[4] A. Bostan, S. Boukraa, S. Hassani, J. M. Maillard, J. A. Weil, and N. Zenine. Globally nilpotent differential operators and the square Ising model. Preprint, available at arXiv:abs/0812.4931, 2008.

[5] A. Bostan, F. Chyzak, and N. Le Roux. Products of ordinary differential operators by evaluation and interpolation. In *ISSAC'08*, pages 23–30. ACM, 2008.

[6] A. Bostan, T. Cluzeau, and B. Salvy. Fast algorithms for polynomial solutions of linear differential equations. In *ISSAC'05*, pages 45–52. ACM Press, 2005.

[7] A. Bostan and M. Kauers. The complete generating function for Gessel walks is algebraic. In preparation.

[8] A. Bostan and M. Kauers. Automatic classification of restricted lattice walks. Preprint, available at arXiv:abs/0811.2899, 2008.

[9] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. ACM*, 25(4):581–595, 1978.

[10] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28(7):693–701, 1991.

[11] T. Cluzeau. Factorization of differential systems in characteristic p. In *ISSAC'03*, pages 58–65. ACM Press, 2003.

[12] T. Cluzeau. *Algorithmique modulaire des équations différentielles linéaires*. PhD thesis, Université de Limoges, 2004.

[13] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, Mar. 1990.

[14] B. Dwork. *Lectures on p-adic differential equations*, volume 253 of *Grundlehren der mathematischen Wissenschaften*. Springer–Verlag, New York, Heidelberg, Berlin, 1982.

[15] B. Dwork. Differential operators with nilpotent p-curvature. *Amer. J. Math.*, 112(5):749–786, 1990.

[16] C. M. Fiduccia. An efficient formula for linear recurrences. *SIAM Journal on Computing*, 14(1):106–112, 1985.

[17] J. von zur Gathen and J. Gerhard. Fast algorithms for Taylor shifts and certain difference equations. In *ISSAC'97*, pages 40–47. ACM, 1997.

[18] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.

[19] T. Honda. Algebraic differential equations. In *Symposia Mathematica, Vol. XXIV (Sympos., INDAM, Rome, 1979)*, pages 169–204. Academic Press, London, 1981.

[20] N. M. Katz. Nilpotent connections and the monodromy theorem: Applications of a result of Turrittin. *Publ. Math. Inst. Hautes Études Sci.*, (39):175–232, 1970.

[21] N. M. Katz. A conjecture in the arithmetic theory of differential equations. *Bull. Soc. Math. France*, (110):203–239, 1982.

[22] H. Niederreiter. A new efficient factorization algorithm for polynomials over small finite fields. *Appl. Algebra Engrg. Comm. Comput.*, 4(2):81–87, 1993.

[23] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[24] J. van der Hoeven. FFT-like multiplication of linear differential operators. *J. Symb. Comp.*, 33(1):123–127, 2002.

[25] M. van der Put. Differential equations in characteristic p. *Compositio Mathematica*, 97:227–251, 1995.

[26] M. van der Put. Reduction modulo p of differential equations. *Indag. Mathem.*, 7(3):367–387, 1996.

[27] M. van der Put and M. Singer. *Galois theory of linear differential equations*. Springer, 2003.

[28] J. F. Voloch. A note on the arithmetic of differential equations. *Indag. Mathem.*, 11(44):617–621, 2000.