Symbolic-Numeric Factorization of Differential Operators

Frédéric Chyzak, Alexandre Goyer, Marc Mezzarobba

Symbolic-Numeric Factorization of Differential Operators

Frédéric Chyzak
Inria
Palaiseau, France
frederic.chyzak@inria.fr

Alexandre Goyer
Inria
Palaiseau, France
alexandre.goyer@inria.fr

Marc Mezzarobba
LIX, CNRS, École polytechnique,
Institut polytechnique de Paris
Palaiseau, France
marc@mezzarobba.net

ABSTRACT

We present a symbolic-numeric Las Vegas algorithm for factoring Fuchsian ordinary differential operators with rational function coefficients. The new algorithm combines ideas of van Hoeij's "local-to-global" method and of the "analytic" approach proposed by van der Hoeven. It essentially reduces to the former in "easy" cases where the local-to-global method succeeds, and to an optimized variant of the latter in the "hardest" cases, while handling intermediate cases more efficiently than both.

ACM Reference Format:
Frédéric Chyzak, Alexandre Goyer, and Marc Mezzarobba. 2022. Symbolic-Numeric Factorization of Differential Operators. In *XXX*. ACM, New York, NY, USA, 10 pages. https://doi.org/XX.XXX/XXXXXX.XXXXXX

1 INTRODUCTION

Problem. Can numerical integration of differential equations help finding exact solutions? The present paper revisits one aspect of this question. To a linear ordinary differential equation

$$y^{(r)}(x) + a_{r-1}(x)y^{(r-1)}(x) \cdots + a_0(x)y(x) = 0,$$

one classically associates the differential operator

$$L = \partial^r + a_{r-1}\partial^{r-1} + \cdots + a_1\partial + a_0,$$

where $\partial = \mathrm{d}/\mathrm{d}x$ is the standard derivation. Linear differential operators with coefficients $a_i \in \mathbb{K}(x)$ for some number field $\mathbb{K} \subset \mathbb{C}$ can be viewed as skew polynomials in ∂ over $\mathbb{K}(x)$, subject to the relation $\partial x = x\partial + 1$. They form a skew Euclidean ring which we denote by $\mathbb{K}(x)\langle\partial\rangle$.

An operator L_1 is said to be a right-hand factor of $L \in \mathbb{K}(x)\langle\partial\rangle$ if there exists an operator L_2 such that $L = L_2L_1$; an operator with no proper right-hand factor is called irreducible. Factoring operators is helpful in understanding their solutions. More precisely, when $L = L_2L_1$, the solution space of L_1 is contained in that of L, whereas solutions w of L_2 give rise to solutions y of L via inhomogeneous equations of the form $L_1(y) = w$.

It is well-known that factorization in this setting is not unique. For instance, one has $\partial^2 = (\partial + 1/(x+\alpha))(\partial - 1/(x+\alpha))$ for any α, expressing that the solutions $y(x) = x+\alpha$ of all first-order equations $(x+\alpha)y'(x) = y(x)$ are gathered as solutions of $y''(x) = 0$.

XXX, XXX, XXX

ACM ISBN XXXXXXXXXXXXXXXXXXXX…$15.00
https://doi.org/XX.XXX/XXXXXX.XXXXXX

In the present paper, we are interested in the problem of finding *one* factorization of an operator $L \in \mathbb{Q}(x)\langle\partial\rangle$ (or, more generally, $L \in \mathbb{K}(x)\langle\partial\rangle$) as a product $L = L_\ell \cdots L_1$ of irreducible operators $L_i \in \overline{\mathbb{Q}}(x)\langle\partial\rangle$. Since, once we have written $L = L_2L_1$, we can recursively try to factor L_1 and L_2, we will focus on the problem of finding any proper right-hand factor.

The problem of factoring differential operators can be rephrased using basic notions of differential Galois theory [MS16, vdPS03]. The basic fact here is that the solution space V of the differential operator L is naturally equipped with an action of the *differential Galois group* G of L, and a subspace of V is the space of solutions of a right-hand factor if and only if it is invariant under this action. In other words, right-hand factors correspond bijectively to submodules of V viewed as module over $\mathbb{C}[G]$. This point of view allows one to study factorizations of differential operators using the general theory of modules over finite-dimensional associative algebras [*e.g.*, Pie82]. This is (explicitly or not) the philosophy of many of the algorithms for factoring operators or solving related problems.

Computing the differential Galois group is notoriously difficult [*e.g.*, Sun19]. However, as a linear algebraic group it admits a finite system of generators that can be described explicitly using values of analytic solutions of differential equations. This property suggests a symbolic-numeric approach to the factorization problem. The idea is to compute generators of the Galois group by solving the equations numerically, then search for a common invariant subspace and use it to reconstruct a candidate factor, and finally check one's guess by exact division.

Previous work. The standard general algorithm for factoring differential operators goes back to Beke [Bek94] at the end of the 19th century, with modern improvements due to Schwarz [Sch89], Bronstein [Bro94] and Tsarev [Tsa94]. Beke's method and its modern variants reduce the problem of finding a right-hand factor of order k of L to that of finding a first-order right-hand factor of the kth exterior power of L, which they do by combining "local first-order factors" at each of the singular points of L. This strategy can be slow even in relatively simple cases for a number of reasons, including the size of exterior powers, the need to work over algebraic extensions of the constants, and a possible combinatorial explosion in the recombination phase [vH97b].

The only worst-case complexity bound we are aware of is due to Grigoriev [Gri90], also using an improved variant of Beke's method. In the special case of a monic $L \in \mathbb{Q}[x]\langle\partial\rangle$ of order r and degree d, it states that L can be factored in time polynomial in $(\delta r d)^{r^4}$ where δ is the maximum degree of L_2 in any factorization $L = L_1L_2L_3$ with monic L_2, L_3. Grigoriev's worst-case bound for δ is more than doubly exponential in r (see Bostan *et al.* [BRS19] for more on this).

More practical algorithms are based on two main ideas. One, the *eigenring method*, introduced by Singer [Sin96] and improved by van Hoeij [vH96], applies to operators that can be decomposed as a least common left multiple of two right-hand factors. The other is a local-to-global approach due to van Hoeij [vH97b]. It applies when the structure of local solutions at one of the singular points satisfies certain conditions, and leads in particular to an efficient algorithm for finding first-order factors. These two methods form the basis of the state-of-the-art implementation, due to van Hoeij [vH97a] and available in Maple as `DEtools[DFactor]`. Beyond the case of first-order factors, though, they are incomplete and need to fall back on the exterior power method in "hard" cases (but still benefit from van Hoeij's fast algorithm for first-order factors then).

The symbolic-numeric approach to factorization outlined above was suggested by van der Hoeven, who also gave fast algorithms for the high-precision computation of generators of the Galois group with rigorous error bounds, and a heuristic method for "reconstructing" the group [vdH07a, vdH07b]. Related symbolic-numeric methods have been developed for the problems of finding all first-order right-hand factors [JKM13], and of computing Liouvillian solutions [LM14]. One of the present authors implemented van der Hoeven's approach and studied its practical behavior [Goy21].

Once numeric approximations of the generators are available, the main task of the factorization algorithm is to find a non-trivial invariant subspace or prove that there is none. This is a basic problem in effective representation theory [*e.g.,* LP10, Chap. 1]. Most of the literature in this area deals either with computations over finite fields or with issues specific to exact computations in characteristic zero. An exception is the early work of Gabriel [Gab71]. We note also that Eberly [Ebe89, p. 245] suggested combining symbolic techniques with interval arithmetic for decomposing algebras and representations over number fields; however, no algorithm of this type appears to have been developed since then. Purely numerical methods for decomposing *unitary* representations [*e.g.,* Dix70] are a different subject with is own developments but are of limited relevance to our problem.

Leaving aside the issue of representing complex numbers in an algebraic algorithm, though, the case of complex representation is the simpler one. Speyer [Spe12] explains how to compute invariant subspaces based on classical methods for decomposing finite-dimensional algebras [compare, *e.g.,* Bre10]. More generally, important ideas used in classical exact algorithms adapt to the rigorous numeric setting, including the Holt–Rees variant [HR94] of Norton's irreducibility test [Par84], and the use of splitting elements [Ebe89, BR90].

Van der Hoeven's paper [vdH07a] also contains an algorithm for finding invariant subspaces. Unlike most related methods, this algorithm admits a deterministic version. Unfortunately, no complete proof is provided, and we have not been able to convince ourselves that the deterministic version works in the general case.

Contribution. We present a new symbolic-numeric algorithm for factoring ordinary differential operators with rational function coefficients. We make two simplifying assumptions. Firstly, we restrict ourselves to *Fuchsian* operators, that is, operators with only regular singular points. This restriction makes some details of the description technically simpler, but we expect that a very similar approach works in general. Secondly, we assume that the operator to be factored only admits a finite number of distinct factorizations. We have evidence that this assumption is stronger than necessary, but leave it to future work to determine how much.

Our algorithm can be viewed as a hybrid of van Hoeij's and van der Hoeven's methods. We point out that van Hoeij's method for exponential parts of multiplicity one can be viewed as a special case of Norton's irreducibility test, which, in a symbolic-numeric setting, applies to any element of the algebra generated by the Galois group. A new extension of van Hoeij's idea allows us to handle a few more cases as efficiently. These two techniques are completed by a streamlined version of van der Hoeven's method, with a more detailed proof. We also present several possible improvements to the reconstruction phase of the symbolic-numeric algorithm, aiming to limit the need for very high numeric precision.

Outline. We first recall some background on the analytic theory of differential equations in Section 2. In Section 3, we specify the model of interval arithmetic used in our algorithms. In Section 4, we discuss the subproblem of reconstructing a factor from numerical initial conditions presumed to lie in a proper invariant subspace. Then, in Section 5, we present several criteria for finding such "seed vectors" or proving that no invariant subspace exists. The main algorithm, combining the tools from the previous two sections, appears in Section 6. Finally, in Section 7, we report on experiments with an implementation of the new algorithm.

Acknowledgements. We thank Alin Bostan, Joris van der Hoeven, and Anne Vaugon for discussions related to the topics of this work.

2 MONODROMY

The main points of the analytic theory of linear differential equations with rational coefficients that we will need are as follows. We refer to [Hil76, Inc26, MS16, vdPS03] for more information.

Singular points. Let $L = \partial^r + a_{r-1}\partial^{r-1} + \cdots + a_0 \in \mathbb{K}(x)\langle\partial\rangle$ be a differential operator. Recall that the *singular points* of L are the poles of $a_0, \ldots, a_{r-1}$ in $\mathbb{P}^1(\mathbb{C})$; denote their set by Σ. Recall also that, on any simply connected domain $U \subset \mathbb{C}\backslash\Sigma$, the space of analytic solutions of the equation $L(y) = 0$ has dimension r. A point $x_0 \in \mathbb{P}^1(\mathbb{C})\backslash\Sigma$ that is not a singular point is called *ordinary*.

A point $\xi \in \Sigma$ is a *regular singular point* if the operator L_ξ obtained by making the change of variable $x \leftarrow \xi+z$ (resp. $x \leftarrow z^{-1}$ if $\xi = \infty$) in L has r linearly independent solutions $y_1, \ldots, y_r$, of the form [Poo36, Chap. V]

$$y_i(z) = z^{\alpha_i}(s_{i,d}(z)\log^d(z) + \cdots + s_{i,0}(z)) \tag{1}$$

for some $\alpha_i \in \overline{\mathbb{Q}}$, $d \in \mathbb{Z}_{\geqslant 0}$, and functions $s_{i,0}, \ldots, s_{i,d}$ analytic on a disk $|z| < \rho$. Thus the y_i are analytic on the slit disk $U = \{z : |z| < \rho,\ z \notin \mathbb{R}_{\leqslant 0}\}$. The α_i occurring in the basis (1) are called the *local exponents* at $x = \xi$ and are the roots of the *indicial polynomial* of L at ξ, a polynomial with coefficients in $\mathbb{K}(\xi)$ that is easily computed from the operator. (By *Fuchs' criterion*, $\xi \in \Sigma$ is a regular singular point if and only if, for $0 \leqslant k < r$, the valuation of a_k at ξ is at least $k - r$. Regularity can hence be checked syntactically.)

We assume from now on that all singular points of L are regular; an operator with this property is also called *Fuchsian*. Note that any factor of a Fuchsian operator is Fuchsian as well.

Right-hand factors and monodromy. Let $Y = (y_1, \dots, y_r)$ be a basis of the solution space V of L on some simply connected domain $U \subset \mathbb{C}\backslash\Sigma$, and consider the associated *Picard–Vessiot extension*, that is, the differential field extension E of $\mathbb{C}(x)$ generated by the y_i. The *differential Galois group* of L can be defined as the group $\mathcal{G} = \mathrm{aut}_{\mathrm{diff}}(E/\mathbb{C}(x))$ of differential automorphisms of E whose restriction to $\mathbb{C}(x)$ is the identity. This is a linear algebraic group [MS16, Theorem 2.10]. The map ψ_Y sending each element $\mathcal{G}$ to the matrix in the basis Y of its action on V is a faithful representation. We denote its image by $\mathrm{Gal}(L, Y)$. For any ordinary point x_0, if Y is the unique basis whose Wronskian matrix $\mathrm{Wr}(y_1, \dots, y_r)$ specializes to the identity matrix at $x = x_0$, then we also write $\mathrm{Gal}(L, x_0)$ in place of $\mathrm{Gal}(L, Y)$.

Solutions of L defined on U can be analytically continued along any path γ drawn in $\mathbb{C}\backslash\Sigma$; for fixed endpoints, the result depends only on the homotopy class of γ in $\mathbb{C}\backslash\Sigma$. The action M_γ of analytic continuation along a loop γ is an element of the differential Galois group. A (local) *monodromy matrix* of L around ξ in the basis Y is a matrix of the form $\psi_Y(M_\gamma)$ where γ is a loop starting from U and going around ξ once, in the positive direction, and enclosing no other singular point. While there can be several homotopy classes with this property, the *monodromy group in the basis* Y, that is, the matrix group generated by local monodromy matrices in the basis Y around each $\xi \in \Sigma$, is defined without ambiguity.

Proposition 2.1. *[vdPS03, Corollary 2.35] A subspace $V_1 \subset V$ is the space of solutions of a right-hand factor of L if and only if it is invariant under the action of the differential Galois group.*

Thus, for any solution f of L, the orbit $\mathbb{C}[\mathcal{G}]f$ is equal to the solution space of the *minimal annihilator* of f, that is, the monic operator R of least order such that $R(f) = 0$. The operator L is reducible if and only if V, viewed as a $\mathbb{C}[\mathcal{G}]$-module, admits a proper submodule. An operator is *decomposable* if it can be written as the least common left multiple (lclm) of operators of lower order, that is, if V is a direct sum of proper submodules.

Theorem 2.2 (Schlesinger). *[MS16, Theorem 2.28] The monodromy group of a Fuchsian operator is a Zariski-dense subset of the differential Galois group.*

Schlesinger's theorem reduces invariance under the differential Galois group to invariance under a finite number of matrices. An entirely similar result holds in the irregular case, using Ramis' generalization of Schlesinger's theorem [vdPS03, Theorem 8.10].

Corollary 2.3. *A subspace $V_1 \subset V$ is the space of solutions of a right-hand factor $L_1 \in \mathbb{C}(x)\langle\partial\rangle$ of L if and only if it is left invariant by the monodromy matrices around all $\xi \in \Sigma$, or equivalently by any choice of all but one of them.*

Proof. Since $\mathcal{G}$ is an algebraic group, a subspace invariant under a Zariski-dense subset is invariant under it. The product of the local monodromy matrices is the identity, so $|\Sigma| - 1$ of them generate the same group as all of them, namely the monodromy group. □

Monodromy matrices typically have transcendental entries. Approximations with rigorous error bounds of the monodromy matrices can be computed using known algorithms for the rigorous numerical integrations of ODEs. The *formal monodromy matrix* at each $\xi \in \Sigma$, that is, the local monodromy matrix around ξ expressed in a suitable local basis of the type (1), though, can be computed exactly. (The computation essentially amounts to changing z^α into $e^{2\pi i\alpha}z^\alpha$ and $\log(z)$ into $\log(z) + 2\pi i$ in (1).) However, this is not enough to express the whole monodromy group in the same basis, as one has to do to get an effective version of Corollary 2.3.

Adjoints. Recall that the *adjoint* of an operator L is the image L^* of L by the anti-morphism of $\mathbb{K}(x)\langle\partial\rangle$ to itself mapping ∂ to $-\partial$.

Lemma 2.4. *Let C denote the companion matrix of L. Define the matrices $B_0, \dots, B_{r-1}$ by $B_0 = I_r$ and $B_{k+1} = B_k' - B_kC^T$. Let P be the matrix whose $(k+1)$th row is the last row of B_k. Then the map $\varphi \mapsto P(x_0)(\varphi^{-1})^T P(x_0)^{-1}$ is a group isomorphism from $\mathrm{Gal}(L, x_0)$ to $\mathrm{Gal}(L^*, x_0)$.*

Proof. Let $W := \mathrm{Wr}(y_1, \dots, y_r)$ where the y_i are solutions of L such that $W(x_0) = I_r$. Note that $W' = CW$. It can be proved [vdPS03, Exercise 2.30] that the matrix $U := (W^{-1})^T$ satisfies $U' = -C^TU$ and the last row $(v_1 \cdots v_r)$ of U is a basis of solutions of L^*. The B_k are defined so that $U^{(k)} = B_kU$. Let $V = \mathrm{Wr}(v_1, \dots, v_r)$ and $Z = \mathrm{Wr}(z_1, \dots, z_r)$ where the z_i are solutions of L^* such that $Z(x_0) = I_r$. Since $V = PU$ and $V = ZP(x_0)$, we have $\sigma(V)(x_0) = P(x_0)(\sigma(W)(x_0)^{-1})^T = \sigma(Z)(x_0)P(x_0)$ and thus $\psi_Z(\sigma) = P(x_0)(\psi_Y(\sigma)^{-1})^T P(x_0)^{-1}$ for any $\sigma \in \mathcal{G}$. □

3 OPTIMISTIC ARITHMETIC

Our algorithms involve algebraic computations, including zero-tests, on complex numbers that are known only approximately (but can be recomputed to higher precision if necessary).

We formalize the way of performing these computations by the following variant of complex interval arithmetic. Complex numbers are replaced by exactly representable closed complex intervals, or *balls* [vdH10], containing them. We denote by $\mathbb{C}_\bullet$ the set of balls. Given a ball $\boldsymbol{z} \in \mathbb{C}_\bullet$, we write $z \in \boldsymbol{z}$ to mean that z is a complex number contained in $\boldsymbol{z}$, and $\mathrm{rad}(\boldsymbol{z})$ to denote the radius of $\boldsymbol{z}$. We extend this notation to lists, vectors, matrices, and polynomials over $\mathbb{C}_\bullet$. A ball is *exact* when its radius is zero.

As with usual interval arithmetic, versions operating on balls of basic operations $* \in \{+, -, \times, /\}$ are defined so that $x * y \in \boldsymbol{x} * \boldsymbol{y}$ for all $x \in \boldsymbol{x}$, $y \in \boldsymbol{y}$, and we assume that $\mathrm{rad}(\boldsymbol{x} * \boldsymbol{y})$ tends to zero when $\boldsymbol{x}$ tends to a point x_0 and $\boldsymbol{y}$ tends to a point y_0 (and both $(\boldsymbol{x}, \boldsymbol{y})$ and (x_0, y_0) are contained in the domain of continuity of $*$, *i.e.*, no division by zero occurs). However, the comparison $\boldsymbol{x} = \boldsymbol{y}$ returns "true" if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ intersect.

Thus, *when the working precision is large enough*, all tests involved in the execution of a particular algorithm on a given exact input yield the same outcome as they would in infinite precision, and the output is a rigorous enclosure of the exact result. At a smaller working precision, equality tests may incorrectly return "true", but we can still rigorously decide that two numbers are distinct provided that the control flow of their computation was not affected by previous incorrect tests. We call this model *optimistic arithmetic*. It is close to the one based on computable complex numbers used in [vdH07a], but more explicit about precision management.

Convention 3.1. We say that an algorithm satisfies some property *at high precision* when the property holds given an accurate enough

input. More precisely, if $\boldsymbol{x}$ is the input of the algorithm, "at high precision, $P(x, \boldsymbol{x})$" means $\forall x, \exists \varepsilon, \forall \boldsymbol{x} \ni x,\ (\mathrm{rad}(\boldsymbol{x}) < \varepsilon \implies P(x, \boldsymbol{x}))$.

Roughly speaking, using optimistic arithmetic is legitimate in our context because (1) our irreducibility criteria are based on "open" conditions like checking that certain vectors span the whole ambient space, where the optimistic zero-test can do no worse than *underestimate* the dimension; (2) in the reducible case, candidate factors can be validated by an *a posteriori* divisibility check carried out in exact arithmetic.

More precisely, inspecting the behavior of key algebraic algorithms shows that they satisfy the following properties. The optimistic version can also fail when the algebraic analogue would not, typically by trying to divide by an interval containing zero. This manifests by an error that can be caught by the caller.

Lemma 3.2. *(Row echelon form.) Given $\boldsymbol{M} \in \mathbb{C}_\bullet^{m\times n}$, one can compute $\boldsymbol{R} \in \mathbb{C}_\bullet^{m\times n}, \boldsymbol{T} \in \mathbb{C}_\bullet^{n\times n}$ such that*

1. *$\boldsymbol{R}$ is row-reduced, in the sense that there is $0 \leqslant r \leqslant \min(m, n)$ and a list $j_0 < j_1 < \cdots < j_{r+1}$ where $j_0 = 0$ and $j_{r+1} = n + 1$, such that*
   - *for all $1 \leqslant i \leqslant r$, the j_ith column of $\boldsymbol{R}$ is exact, with the ith entry equal to one and all other entries equal to zero,*
   - *for all $0 \leqslant i \leqslant r$ and $j_i < j < j_{i+1}$, each of the $m - i$ last entries of the jth column of $\boldsymbol{R}$ is a ball that contains zero,*
2. *r cannot exceed the rank of any $M \in \boldsymbol{M}$,*
3. *for all $M \in \boldsymbol{M}$, there exist $R \in \boldsymbol{R}$ and an invertible $T \in \boldsymbol{T}$ such that $R = TM$,*
4. *at high precision, r is equal to the rank of M and the reduced row echelon form of M belongs to $\boldsymbol{R}$.*

In particular, at high precision, we can verify that an $M \in \mathbb{C}^{m\times n}$ has full rank.

Lemma 3.3. *(Kernel.) Given $\boldsymbol{M} \in \mathbb{C}_\bullet^{m\times n}$, one can compute $\boldsymbol{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell) \in \left(\mathbb{C}_\bullet^n\right)^\ell$ such that*

1. *any $v_1, \dots, v_\ell$ with $v_i \in \boldsymbol{v}_i$ are linearly independent,*
2. *for all $M \in \boldsymbol{M}$, there exists $V \in \boldsymbol{V}$, that is, $V = (v_1, \dots, v_\ell)$ and $v_i \in \boldsymbol{v}_i$ for all i, such that $\ker(M) \subset \mathrm{span}(V)$,*
3. *at high precision, the last inclusion is an equality.*

In particular, at high precision, we can verify the nullity of a kernel.

Lemma 3.4. *(Spin-up.) Given a list $\boldsymbol{A} \in (\mathbb{C}_\bullet^{n\times n})^k$ of matrices and a vector $\boldsymbol{v} \in \mathbb{C}_\bullet^n$, one can compute $\boldsymbol{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_\ell) \in \left(\mathbb{C}_\bullet^n\right)^\ell$ such that*

1. *any $u_1, \dots, u_\ell$ with $u_i \in \boldsymbol{u}_i$ are linearly independent,*
2. *for all $M \in \boldsymbol{A}$ and $v \in \boldsymbol{v}$, there exists $U \in \boldsymbol{U}$ such that $\mathbb{C}[A]v \supset \mathrm{span}(U)$,*
3. *at high precision, the last inclusion is an equality.*

In particular, at high precision, we can verify that $\mathbb{C}[A]v = \mathbb{C}^n$ when this is the case.

Lemma 3.5. *(Root isolation.) Given a monic polynomial $\boldsymbol{P}$, one can compute pairs $(\boldsymbol{\lambda}_1, m_1), \dots, (\boldsymbol{\lambda}_\ell, m_\ell)$ such that*

1. *the $\boldsymbol{\lambda}_i \in \mathbb{C}_\bullet$ are pairwise disjoint and the m_i are positive,*
2. *for all $P \in \boldsymbol{P}$, each $\boldsymbol{\lambda}_i$ contains exactly m_i roots (counted with multiplicities) of P, and all roots of P are contained in $\bigcup_i \boldsymbol{\lambda}_i$,*
3. *at high precision, no two distinct roots of P are contained in the same λ_i.*

In particular, at high precision, we can verify that a root is simple.

4 MINIMAL ANNIHILATORS

Like both van Hoeij's and van der Hoeven's, our factoring algorithm works by searching for a solution that belongs to a proper invariant subspace, and reconstructing an annihilator of that solution. In this section, we discuss the problem of reconstructing an invariant subspace, and a corresponding right-hand factor, from a seed vector.

We fix a monic differential operator $L \in \mathbb{K}(x)\langle\partial\rangle$ of order r, and an ordinary point $x_0 \in \mathbb{Q}$ of L. We denote $G = \mathrm{Gal}(L, x_0)$. A solution f of L is represented by the vector $v = (f(x_0), \dots, f^{(r-1)}(x_0))^T$ (so that the action of $\mathcal{G}$ on f corresponds to a left action of G on v), and we sometimes abusively identify f with v.

We use Algorithm 1 to compute a right-hand factor of L from an approximate seed vector $\boldsymbol{v}$.

Proposition 4.1. *Fix $\varphi_0, \dots, \varphi_k \in \mathbb{C}[G]$. Let $\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k \in \mathbb{C}_\bullet^{r\times r}$ be such that $\varphi_i \in \boldsymbol{\varphi}_i$ for all i.*

1. *Annihilator(L, $\boldsymbol{v}$, $(\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k)$, t) returns either the special value* Inconclusive *or a right-hand factor $R \in \overline{\mathbb{Q}}(x)\langle\partial\rangle$ of L.*
2. *If the output is L, no $v \in \boldsymbol{v}$ admits an annihilator of order $< r$.*

Assume further that exact initial conditions $v \in \mathbb{C}^r$ are fixed and $\boldsymbol{v}$ is chosen such that $\boldsymbol{v} \ni v$. Let M be the minimal annihilator of v.

3. *If $M \in \overline{\mathbb{Q}}(x)\langle\partial\rangle$, then, at high precision and for large enough t, the output is M.*
4. *If $M = L$ and $\varphi_0, \dots, \varphi_k$ generate $\mathbb{C}[G]$, the output at high precision is L with no assumption on t.*

Proof. Assertion (1) is straightforward. If L is returned on line 2, the fact that no $v \in \boldsymbol{v}$ has an annihilator of smaller order is ensured by Lemma 3.4. Step 6 amounts to a kernel computation, so the same conclusion holds if the algorithm terminates on line 11, by Lemma 3.3. When termination happens on line 9, the returned R has order less than r. This proves (2). Let v and M be as in the statement and $V = \mathbb{C}[G]v$. Note that $L = M$ if and only if $V = \mathbb{C}^r$. At high precision, this is correctly decided on line 2 when $\mathbb{C}[\varphi_1, \dots, \varphi_k] = \mathbb{C}[G]$ thanks to Lemma 3.4, proving (4). At high precision, Lemma 3.3 ensures that, after line 6 is executed with $s = r - 1$, the resulting $\boldsymbol{R}$ contains an operator R of order at most $r - 1$ and minimum degree such that $R(f) = O((x - x_0)^t)$. When $M = L$, it follows that line 11 is eventually reached as $t \to \infty$. Assume now that $M \in \overline{\mathbb{Q}}(x)\langle\partial\rangle$ and $\mathrm{ord}\, M \neq L$. At high precision, by Lemma 3.2, step 1 yields a tuple $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_d)$ with $d \leqslant \mathrm{ord}\, M$. Line 6 with $s = \mathrm{ord}\, M$ then finds an $\boldsymbol{R}$ with $M \in \boldsymbol{R}$. By assumption, M has coefficients in $\overline{\mathbb{Q}}$, so that the LLL algorithm eventually recovers M from $\boldsymbol{R}$ as the radii of the coefficients of $\boldsymbol{R}$ tend to zero. This proves (3). □

The assumption in Proposition 4.1 that the minimal annihilator of f has algebraic coefficients is automatically satisfied when L has a finite number of factorizations. Indeed, the (L_1, L_2) with $L = L_2L_1$ form an algebraic variety defined over $\mathbb{K}$, which is then zero-dimensional. In the presence of parameterized families of right-hand factors (like in the example of ∂^2 mentioned in the Introduction), however, some choices of f lead to an annihilator with transcendental coefficients[1].

[1] The algorithm from [vdH07a] is incorrect as stated for this reason: when $\dim K_i > 1$ at step 5 of `Invariant_subspace`, the vector $\boldsymbol{v}$ chosen from K_i may correspond to a minimal annihilator with transcendental coefficients, in which case `Right_factor` will loop indefinitely.

Algorithm 1: Annihilator(L, $\boldsymbol{v}$, $(\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k)$, t)

Input: $L \in \mathbb{K}(x)\langle\partial\rangle$ of order r, $\boldsymbol{v} \in \mathbb{C}_\bullet^r$, $\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k \in \mathbb{C}_\bullet^{r\times r}$, $t \in \mathbb{Z}_{>0}$

Output: a right-hand factor R of L, or *Inconclusive*

1 Compute the dimension d and a basis $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_d) \in (\mathbb{C}_\bullet^r)^d$ of $\mathbb{C}[\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k]\boldsymbol{v}$ in reduced echelon form (Lemma 3.4);
2 if $d = r$ then return L;
3 Compute the first $t + r$ terms of the solution $\boldsymbol{f} \in \mathbb{C}_\bullet[[x - x_0]]$ of L defined by $(\boldsymbol{f}(x_0), \dots, \boldsymbol{f}^{(r-1)}(x_0))^T = \boldsymbol{v}$;
4 Compute B such that any monic right-hand factor of L has degree $\leqslant B$ [vH97b, Section 9]; // precomputable
5 for $s = d, \dots, r-1$:
6 Compute a monic $\boldsymbol{R} \in \mathbb{C}_\bullet(x)\langle\partial\rangle$ of minimum degree such that $\operatorname{ord} \boldsymbol{R} \leqslant s$ and $\boldsymbol{R}(\boldsymbol{f}) = O((x - x_0)^t)$ using Hermite–Padé approximation;
7 if $\deg \boldsymbol{R} < t/(s+1)$ then
8 Compute $R \in \boldsymbol{R} \cap \overline{\mathbb{Q}}(x)\langle\partial\rangle$ using the LLL algorithm[2];
9 if R divides L from the right then return R;
10 if $\deg \boldsymbol{R} > B$ then // can only happen for large t
11 Return L;
12 Return *Inconclusive*;

Remark 4.2. Due to interval blow-up, getting a precise enough $\boldsymbol{R}$ at step 6 to be able to proceed may require a large working precision. Instead, we can compute a minimal approximant basis of

$$(y_0, y_0', \dots, y_0^{(r-1)}, y_1, y_1', \dots, y_1^{(r-1)}, \dots, y_{r-1}, y_{r-1}', \dots, y_{r-1}^{(r-1)})$$

where $(y_0, \dots, y_{r-1})$ is the local basis (consisting of exact series) such that $\boldsymbol{f} = \boldsymbol{u}_0 y_0 + \cdots + \boldsymbol{u}_{r-1} y_{r-1}$. We then search for elements (of a certain maximum degree) of the form

$$(\boldsymbol{u}_0 q_0, \dots, \boldsymbol{u}_0 q_{r-1}, \boldsymbol{u}_1 q_0, \dots, \boldsymbol{u}_1 q_{r-1}, \dots, \boldsymbol{u}_{r-1} q_0, \dots, \boldsymbol{u}_{r-1} q_{r-1})$$

in the module of relations. The latter step reduces to solving a linear system over $\mathbb{C}_\bullet[x]$ given by a matrix mixing exact polynomials and constant ball entries [*cf.* CDDM].

Remark 4.3. Another way of limiting the need for Hermite–Padé approximants with high-precision ball coefficients is as follows. Between lines 2 and 3 of Algorithm 1, we insert a step that attempts to reconstruct a vector $e_1 \in \boldsymbol{e}_1 \cap \overline{\mathbb{Q}}^r$ using the LLL algorithm. If this succeeds, we compute the power series solution f_1 of L associated to e_1 and attempt to recover a factor from it.

In the notation of Proposition 4.1, this strategy yields a proper right-hand factor at high precision when $\mathbb{C}[\varphi_0, \dots, \varphi_k] = \mathbb{C}[G]$. Indeed, $\boldsymbol{e}_1$ contains the first vector e_1 of the exact reduced echelon basis of V, and V is the image in $\mathbb{C}^r$ of the whole solution space of M. As M has coefficients in $\overline{\mathbb{Q}}(x)$, it admits a basis of solutions whose series expansions at x_0 have coefficients in $\overline{\mathbb{Q}}$. A solution g of M is represented in V by the vector $(g(x_0), \dots, g^{(r-1)}(x_0)) \in \overline{\mathbb{Q}}^r$, hence V is generated by vectors with entries in $\overline{\mathbb{Q}}$, and therefore the elements of its reduced echelon basis belong to $\overline{\mathbb{Q}}^r$ as well. Note that the resulting factor might not be an annihilator of f (though this is easy to fix if desired).

Remark 4.4. If a proper right-hand factor of $L \in \mathbb{K}(x)\langle\partial\rangle$ has coefficients in $\mathbb{L}(x)$ for some an extension $\mathbb{L}$ of $\mathbb{K}$, then its conjugates under the action of $\mathrm{Gal}(\mathbb{L}/\mathbb{K})$ are right-hand factors of L as well. Their lclm R is a right-hand factor with coefficients in $\mathbb{K}(x)$. As noted by van Hoeij [vH97b, Section 8], if we have failed to factor L by the eigenring method before, we know that L is indecomposable, so that R must be a proper factor.

In the setting of the previous remark, we can take advantage of this observation as follows. When the reconstruction of e_1 from $\boldsymbol{e}_1$ yields a vector with entries in some $\mathbb{L} \supset \mathbb{K}$, we replace e_1 by the average of its Galois conjugates for the Hermite-Padé step. The average is nonzero since one of the entries of e_1 is equal to one, and it defines a power series solution of L with coefficients in $\mathbb{K}$, whose minimal annihilator hence lies in $\mathbb{K}(x)\langle\partial\rangle$.

5 SUBMODULES AND IRREDUCIBILITY

Let L, x_0, and G be as in the previous section. We now discuss three different ways of finding proper invariant subspaces under the monodromy action (G-submodules) or proving that none exists.

All three tests follow the same pattern. We start with a possibly incomplete set of (approximate or exact) generators of the monodromy group. When one of the tests is applicable, either we exploit error bounds to certify the absence of any proper submodule, which implies that L is irreducible, or we find an approximation $\boldsymbol{v}$ of a candidate v such that $\mathbb{C}[G]v$, from which we attempt to reconstruct a factor of L by Algorithm 1.

Let $\mathcal{A} \subset \mathbb{C}^{r\times r}$ be a matrix algebra. In our applications, $\mathcal{A}$ will be the algebra $\mathbb{C}[G]$ considered in the previous section or a subalgebra of it. We will consider both the left action of $\mathcal{A}$ on the column space $\mathbb{C}^{r\times 1}$ and the right action of $\mathcal{A}$ on the row space $\mathbb{C}^{1\times r}$. If nothing is specified, $\mathbb{C}^r$ stands for the left $\mathcal{A}$-module $\mathbb{C}^{r\times 1}$.

By [Lam98], the $\mathcal{A}$-module $\mathbb{C}^r$ admits a proper submodule if and only if $\mathcal{A} \neq \mathbb{C}^{r\times r}$. Note that this criterion provides no proper $\mathcal{A}$-submodule, even if one exists.

Norton's criterion. The following result is a special case of Norton's irreducibility test, in the form used in the Holt–Rees variant of the "Meataxe" algorithm for testing the irreducibility of modules over finite fields [HR94]. In our context, the infiniteness of the base field $\mathbb{C}$ precludes the use of Norton's test in its generality, that is, with an eigenvalue that is not necessarily simple.

Proposition 5.1. *[Par84, HR94] Assume that there is $M \in \mathcal{A}$ having a simple eigenvalue λ. Introduce nonzero vectors $v \in \mathbb{C}^{r\times 1}$ and $w \in \mathbb{C}^{1\times r}$ such that $Mv = \lambda v$ and $wM = \lambda w$. Then, equivalently: (i) the left $\mathcal{A}$-module $\mathbb{C}^{r\times 1}$ is irreducible; (ii) both $\mathcal{A}v = \mathbb{C}^{r\times 1}$ and $w\mathcal{A} = \mathbb{C}^{1\times r}$ hold; (iii) the right $\mathcal{A}$-module $\mathbb{C}^{1\times r}$ is irreducible.*

Proof. For $w \in \mathbb{C}^{1\times r}$ and $v \in \mathbb{C}^{r\times 1}$, write $\langle w, v\rangle$ for $\sum_{i=1}^r w_i v_i$. For a subspace $F \subset \mathbb{C}^{r\times 1}$, we denote by $F^\perp := \{w \in \mathbb{C}^{1\times r} \mid \forall u \in F, \langle w, u\rangle = 0\}$ the orthogonal of F. We define symmetrically the orthogonal $G^\perp$ of a subspace $G \subset \mathbb{C}^{1\times r}$. For any subspace $F \subset \mathbb{C}^{r\times 1}$, $F = F^{\perp\perp}$ holds and F is a left $\mathcal{A}$-module if and only $F^\perp$ is a right $\mathcal{A}$-module; similarly for subspaces $G \subset \mathbb{C}^{1\times r}$.

Assume (ii) does not hold. If $0 \subsetneq w\mathcal{A} \subsetneq \mathbb{C}^{1\times r}$, then $\mathbb{C}^{r\times 1} \supsetneq (w\mathcal{A})^\perp \supsetneq 0$, and $(w\mathcal{A})^\perp$ is a proper submodule of $\mathbb{C}^{r\times 1}$. Otherwise,

[2] To reconstruct an element $z \in \overline{\mathbb{Q}}$ from a ball $\boldsymbol{z}$, we search for an algebraic number $z \in \boldsymbol{z} \cap \overline{\mathbb{Q}}$ of degree at most $\delta \sim (-\log(\mathrm{rad}(\boldsymbol{z})))^{1/2}$.

$0 \neq \mathcal{A}v \neq \mathbb{C}^{r\times 1}$, making $\mathcal{A}v$ a proper submodule. So $\mathbb{C}^{r\times 1}$ is a reducible module in all cases.

Conversely, assume (i) does not hold, and let U be a proper $\mathcal{A}$-submodule of $\mathbb{C}^{r\times 1}$. The equality $\ker(M-\lambda I_r) = \mathcal{A}v$ holds because λ is a simple eigenvalue. If $\mathcal{A}v \subset U$, then $\mathcal{A}v \neq \mathbb{C}^{r\times 1}$. Otherwise, $\ker(M-\lambda I_r)\cap U = \{0\}$. Since $(M-\lambda I_r)U \subset U$, we have $(M-\lambda I_r)U = U$ by finite dimension. Hence, for all $u \in U$, there is $u' \in U$ such that $\langle w, u\rangle = \langle w, (M-\lambda I_r)u'\rangle = \langle w(M-\lambda I_r), u'\rangle = \langle 0, u'\rangle = 0$. Therefore $w \in U^\perp$, so $w\mathcal{A} \subset U^\perp$ and $w\mathcal{A} \neq \mathbb{C}^{1\times r}$. □

In the special case where M is a formal monodromy matrix, we recover van Hoeij's local-to-global method. Indeed, at a regular singular point, the *exponential parts* defined in [vH97b, Section 3] correspond to the eigenvalues of the formal monodromy matrix. Van Hoeij observes that one can find a factorization or prove that there is none as soon as there is an exponential part e of multiplicity 1 at some singular point, because e is then an exponential part of either L_1 or L_2 but not both in a factorization $L = L_2L_1$. To decide whether e is an exponential part of a right-hand factor, van Hoeij computes a series solution f associated to e and searches for an annihilator of f of order smaller than r using Hermite–Padé approximants. Thanks to degree bounds, it is possible to ensure that e is not an exponential part of any right-hand factor. As noted in Section 2, this is equivalent to $\mathbb{C}[G]f$ being $\mathbb{C}^r$. One can decide if e is an exponential part of a left-hand factor in a similar way, by passing to the adjoint operator.

In the setting where $\mathcal{A} = \mathbb{C}[G]$, we can test point (ii) of Proposition 5.1 in two different ways: we either compute bases of $\mathcal{A}v$ and $w\mathcal{A}$ by saturation, or search for annihilators satisfying certain degree bounds as in van Hoeij's method. The first method is typically more efficient when a full basis of $\mathcal{A}$ is available, but the second has the advantage of being applicable even if only part of the monodromy matrices have been computed. Compared to van Hoeij's method, the numerical test applies to a larger class of operators because an element of $\mathcal{A}$ can have a simple eigenvalue even if the generators only have multiple eigenvalues.

As we will now show, performing either variant of this test using optimistic arithmetic can prove irreducibility, or provide a candidate invariant subspace, depending on the reducibility of the operator.

Proposition 5.2. *Suppose that L is a monic Fuchsian operator admitting finitely many distinct right-hand factors. Fix $\varphi_0, \dots, \varphi_k \in \mathbb{C}[G]$ and let R be the output of* SimpleEigenvalue$(L, (\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k), t)$ *where $\varphi_i \in \boldsymbol{\varphi}_i$. If $R =$* Irreducible*, then L is irreducible. If R is an operator, then R is a proper right-hand factor of L. Assume further that φ_0 has a simple eigenvalue. Then, at high precision: (1) R is either a factor or* Irreducible *for large t; (2) if L is irreducible and the φ_i generate $\mathbb{C}[G]$, the output is* Irreducible.

Proof. Assume that the computation of the eigenvalues of $\boldsymbol{\varphi}_0$ finds an eigenvalue $\boldsymbol{\lambda}$ of multiplicity 1. Lemma 3.5 ensures that φ_0 admits a simple eigenvalue $\lambda \in \boldsymbol{\lambda}$, and, by Lemma 3.3, the eigenvectors v of φ_0, w of χ_0 for λ belong to the respective computed eigenvectors $\boldsymbol{v}$ of $\boldsymbol{\varphi}_0$, $\boldsymbol{w}$ of $\boldsymbol{\chi}_0$ for $\boldsymbol{\lambda}$. By Proposition 4.1, the call to Annihilator on line 2 (if it succeeds) either yields a proper factor or proves that the minimal annihilator of v is L. Since the group generated by the φ_i is the same as the one generated by the φ_i^{-1}, the χ_i are elements of $\mathbb{C}[\mathrm{Gal}(L^*, x_0)]$ by Lemma 2.4, so a similar reasoning applies to line 7. If the minimal annihilators turn out to be L and L^*, then point (ii) of Proposition 5.1 holds with $\mathcal{A} = \mathbb{C}[\varphi_0, \dots, \varphi_k]$, hence also with $\mathcal{A} = \mathbb{C}[G]$, and we can conclude that L is irreducible. Finally, at high precision, when φ_0 does have a simple eigenvalue, all numerical steps succeed, and assertions (1)–(2) follow from assertions (3)–(4) in Proposition 4.1. □

Algorithm 2: SimpleEigenvalue$(L, (\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k), t)$

Input: $L \in \mathbb{K}(x)\langle\partial\rangle$ of order r, $\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k \in \mathbb{C}_\bullet^{r\times r}$, $t \in \mathbb{Z}_{>0}$
Output: a right-hand factor of L, *Irreducible*, or *Inconclusive*

1 Compute a simple eigenvalue $\boldsymbol{\lambda}$ of $\boldsymbol{\varphi}_0$ and an eigenvector $\boldsymbol{v}$ of $\boldsymbol{\varphi}_0$ associated to $\boldsymbol{\lambda}$; // may fail
2 R = Annihilator$(L, \boldsymbol{v}, (\boldsymbol{\varphi}_0, \dots, \boldsymbol{\varphi}_k), t)$;
3 if $\mathrm{ord}(R) < \mathrm{ord}(L)$ then return R;
4 Compute $P(x_0)$ as in Lemma 2.4; // precomputable
5 Compute $\boldsymbol{\chi}_0, \dots, \boldsymbol{\chi}_k$ for $\chi_i := P(x_0)\varphi_i^T P(x_0)^{-1}$, $0 \leqslant i \leqslant k$;
6 Compute an eigenvector $\boldsymbol{w}$ of $\boldsymbol{\chi}_0$ associated to $\boldsymbol{\lambda}$; // may fail
7 Q = Annihilator$(L^*, \boldsymbol{w}, (\boldsymbol{\chi}_0, \dots, \boldsymbol{\chi}_k), t)$;
8 if $\mathrm{ord}(Q) < \mathrm{ord}(L)$ then return $(L^*/Q)^*$;
9 else if $R = L$ and $Q = L^*$ then return *Irreducible*;
10 else return *Inconclusive*; // R or Q is *Inconclusive*

One-dimensional eigenspaces. It is not unusual in applications to encounter operators whose local monodromy matrices have a single eigenvalue, yet with a one-dimensional eigenspace ("MUM points"). The following test is useful in particular for dealing with combinations of such operators. As Norton's criterion adapts to van Hoeij's method, so too does this next test sometimes apply to a formal monodromy matrix. It can therefore be used in a purely symbolic factoring algorithm.

Proposition 5.3. *Assume that there is $M \in \mathcal{A}$ whose eigenspaces $E_1, \dots, E_\ell$ are all 1-dimensional. Let $v_i \in \mathbb{C}^r$ satisfy $E_i = \mathbb{C}v_i$ for each $1 \leqslant i \leqslant \ell$. Then $\mathbb{C}^r$ is an irreducible $\mathcal{A}$-module if and only if $\mathcal{A}v_i = \mathbb{C}^r$ for all $1 \leqslant i \leqslant \ell$.*

Proof. The eigenvalues of the restriction of M to an invariant subspace are eigenvalues of M, so any nonzero invariant subspace must intersect at least one eigenspace of M in a nontrivial way. □

Let us explain why this test can again prove the irreducibility at high precision. We denote by $\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_\ell$ the eigenvalues of a ball approximation $\boldsymbol{M}$ of an element $M \in \mathcal{A}$, and we assume that, for each $1 \leqslant i \leqslant \ell$: (1) the optimistic computation of $\ker(\boldsymbol{M} - \boldsymbol{\lambda}_i I_r)$ returns a single vector $\boldsymbol{v}_i$, and (2) the optimistic computation of the orbit of $\boldsymbol{v}_i$ returns r independent vectors. Then all the eigenspaces of M are 1-dimensional and one has $\mathcal{A}v = \mathbb{C}^r$ for each eigenvector v of M. Indeed, consider an eigenvalue μ of M. Since $\mu \in \boldsymbol{\lambda}_i$ for some i, there exists $v \in \boldsymbol{v}_i$ such that $\ker(M - \mu I_r) \subset \mathbb{C}v$. But $\ker(M - \mu I_r) \neq \{0\}$ so $\ker(M - \mu I_r) = \mathbb{C}v$. Next, $\mathcal{A}v = \mathbb{C}^r$ thanks to the computation of the orbit of $\boldsymbol{v}_i$. Note that all the distinct eigenvalues of M do not need to be isolated in different $\boldsymbol{\lambda}_i$.

This leads to a procedure OneDimEigenspaces, which we omit, with the same signature as SimpleEigenvalue and similar correctness properties.

Van der Hoeven's algorithm revisited. The following result is based on the ideas introduced in [vdH07a]. It allows us to deal with the cases that cannot be handled by the two previous criteria.

Proposition 5.4. *Assume that all the matrices of $\mathcal{A}$ have at least one multiple eigenvalue. Consider $M \in \mathcal{A}$ with a maximal number of eigenvalues. Denote by λ one of its multiple eigenvalues, by E the generalized eigenspace of M for λ, that is, $E = \ker((M - \lambda I_r)^r)$, and by F the sum of the other generalized eigenspaces of M, so that $\mathbb{C}^r = E \oplus F$. Let $K := \{v \in E \mid \forall N \in \mathcal{A}, PNv \in \mathbb{C}v\}$ where $P \in \mathcal{A}$ denotes the projection onto E along F.*

Then $K \neq \{0\}$ and $\mathcal{A}v$ is a proper $\mathcal{A}$-submodule of $\mathbb{C}^r$ for any nonzero $v \in K$. In particular, the $\mathcal{A}$-module $\mathbb{C}^r$ is reducible.

Proof. Let $N \in \mathcal{A}$ and φ be the endomorphism of E defined by $\varphi(v) := PNv$. Note that $PNv = PNPv$ for any $v \in E$. Let us show that φ has a unique eigenvalue. Otherwise, take a nonzero eigenvalue μ of φ. Hence μ is also an eigenvalue of $PNP \in \mathcal{A}$. Denote by E_μ the generalized eigenspace of PNP for μ, by G the sum of the other generalized eigenspaces of PNP and by Q the projector onto E_μ along G. It is then classical [Bou90, A.VII.31, Prop. 3] that the projector P, respectively Q, can be written as a polynomial in M, respectively in PNP, so P and Q belong to $\mathcal{A}$. Hence QP is the projector onto $E \cap E_\mu$ along $(E \cap G) \oplus F$. Since $E \cap E_\mu \subsetneq E$, we observe that $M + \alpha QP$ has more eigenvalues than M for any α such that $\lambda + \alpha$ is not an eigenvalue of M; this is in contradiction with the assumption made on M.

Define $\mathcal{A}_E := \{\varphi_N - \lambda_N \operatorname{id}_E ; N \in \mathcal{A}\}$, where φ_N is the endomorphism of E defined by $\varphi_N(v) := PNv$ and λ_N is its unique eigenvalue, so that $K = \bigcap_{n \in \mathcal{A}_E} \ker(n)$. Owing to a result of Levitski [Kap72, Theorem 35, p. 135] that states that a semigroup of nilpotent endomorphisms is simultaneously triangularizable, showing $K \neq \{0\}$ reduces to showing that $\mathcal{A}_E$ is stable by composition. For all $N, R \in \mathcal{A}$, we have $(\varphi_N - \lambda_N \operatorname{id}_E)(\varphi_R - \lambda_R \operatorname{id}_E) = \varphi_S + \lambda_N \lambda_R \operatorname{id}_E$ where $S := NPR - \lambda_N R - \lambda_R N \in \mathcal{A}$. Applying the equality of endomorphisms to any nonzero eigenvector v of φ_R shows $\lambda_S = -\lambda_N \lambda_R$.

For the statement on $\mathcal{A}v$, we proceed by contraposition. If $v \in K$ satisfies $\mathcal{A}v = \mathbb{C}^r$, there is $N \in \mathcal{A}$ such that $Nv \in E \backslash \mathbb{C}v$ because the dimension of E is at least 2, so $v \notin K$. □

This proposition also implies an irreducibility criterion ("if some $M \in \mathcal{A}$ has only simple eigenvalues and $\mathcal{A}v = \mathbb{C}^r$ for each v in a basis of eigenvectors, then $\mathbb{C}^r$ is irreducible"), but this criterion is weaker than Norton's. Again we omit the corresponding procedure MultipleEigenvalue, which computes the space K of Proposition 5.4, then calls Annihilator (Algorithm 1) on any of its nonzero elements. Since K can be written as an intersection of kernels, the convergence of its computation at high precision is ensured by Lemma 3.3.

Proposition 5.5. *Suppose that L is a monic Fuchsian operator admitting finitely many distinct right-hand factors. Fix $\varphi_0, \ldots, \varphi_k \in \mathbb{C}[G]$ and let $R = \text{MultipleEigenvalue}(L, (\boldsymbol{\varphi}_0, \ldots, \boldsymbol{\varphi}_k), t)$ where $\varphi_i \in \boldsymbol{\varphi}_i$. Then R is either the special value* Inconclusive *or a proper right-hand factor of L. Assume additionally that L is reducible and that φ_0 has a maximal number of eigenvalues among the elements of $\mathbb{C}[G]$. Then, at high precision, if $\varphi_0, \ldots, \varphi_k$ generate $\mathbb{C}[G]$ and t is large enough, the algorithm neither fails nor returns* Inconclusive.

Algorithm 3: RightFactor(L)

Input: $L \in \mathbb{K}(x)\langle\partial\rangle$ of order r
Output: a proper right-hand factor R of L or *Irreducible*

1 Choose an ordinary base point $x_0 \in \mathbb{Q}$;
2 Compute the finite singular points $\xi_1, \ldots, \xi_\nu$ of L;
3 Set some initial working precision p and truncation order t;
4 try:
5 for $i = 1, \ldots, \nu$:
6 Compute a rigorous enclosure $\boldsymbol{\varphi}_i \in \mathbb{C}_\bullet^{r \times r}$ of a local monodromy matrix $\varphi_i \in \mathrm{Gal}(L, x_0)$ around ξ_i, working at precision p; // may fail
7 $\boldsymbol{\varphi}$ = a random combination of $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_i$;
8 if $\boldsymbol{\varphi}$ has a simple eigenvalue then
9 R = SimpleEigenvalue(L, $(\boldsymbol{\varphi}, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_i)$, t);
10 else if all eigenspaces of $\boldsymbol{\varphi}$ are 1-dimensional then
11 R = OneDimEigenspaces(L, $(\boldsymbol{\varphi}, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_i)$, t);
12 else
13 R = MultipleEigenvalue(L, $(\boldsymbol{\varphi}, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_i)$, t);
14 if $R \neq$ *Inconclusive* then return R;
15 catch: *Failure* // *e.g.*, division by 0 in a basic subroutine
16 Increase p and go to line 4;
17 Increase t and p and go to line 4;

6 FACTORING

The three previous tests combine into a factorization procedure described in Algorithm 3. Since no bounds for a sufficient numeric precision are known, the strategy consists in increasing the precision p every time it turns out to be insufficient until getting either a proper factor or an irreducibility certificate.

Bounds on the possible degrees of right-hand factors exist, but these bounds can be large even when the operator is irreducible. We hence increase also the series truncation order t progressively, in the hope of proving irreducibility by purely numerical methods ($t \approx 0$) or finding factors of low degree ($t \gtrsim \deg(L) \operatorname{ord}(L)$) before reaching the bound. Increasing t requires increasing p as well, to compensate for both loss of precision in larger computations and the expected larger bit-size of coefficients of high-degree factors.

Computing monodromy matrices (though asymptotically of cost softly linear in p) tends to be the most expensive step in practice; therefore, for given p and t, we try to use as few of them as possible.

Line 7 of Algorithm 3 needs additional explanations. The idea is that taking a random element of $\mathbb{C}[\varphi_1, \ldots, \varphi_i]$ will immediately provide a φ satisfying the assumptions of Proposition 5.1 or Proposition 5.4 if any such element exists. This is made precise in the following result. In practice, rather than maintaining a basis of $U = \mathbb{C}[\varphi_1, \ldots, \varphi_i]$, we can multiply together a few linear combinations of $\varphi_1, \ldots, \varphi_i$, increasing that number if necessary. At worst, multiplying $\dim U$ random linear combinations of generators will yield a "generic" element.

Lemma 6.1. *[Ebe91, Lemma 2.1]. Let $U \subset \mathbb{C}^{r \times r}$ be a vector space. Let m be the maximum cardinality of the spectrum of any element of U. The elements of U with less than m distinct eigenvalues form a proper algebraic subset of U.*

Another subtlety is that the increase of p on line 17 is important to ensure termination: without it the working precision might not suffice to compensate for the additional work due to a larger t, and interval computations could fail at every iteration.

Proposition 6.2. *Let $L \in \mathbb{K}(x)\langle\partial\rangle$ be a Fuchsian operator. Assume that L admits a finite number of factorizations as a product of irreducible elements of $\overline{\mathbb{Q}}(x)\langle\partial\rangle$. There exists a proper algebraic subset $X \subsetneq \mathrm{Gal}(L, x_0)$ such that Algorithm 3 terminates provided that $\boldsymbol{\varphi} \cap X = \varnothing$ at step 7 of every iteration. Algorithm 3 then returns* Irreducible *if and only if L is irreducible, and returns a proper right-hand factor of L otherwise.*

Heuristically, when L is irreducible, we expect the algorithm to conclude as soon as $\mathbb{C}[\varphi_1, \dots, \varphi_i]$ contains a matrix with a simple eigenvalue and enough other elements of $\mathbb{C}[G]$ that Norton's test passes. Verifying irreducibility this way should require only a moderate p and does not depend on t. In the reducible case, t and p need to reach at least the total arithmetic size (resp. the bit size of the coefficients) of at least one right-hand factor before the computation has any chance of finishing. Once t and p are large enough, we can expect again the computation to finish as soon as $\mathbb{C}[\varphi_1, \dots, \varphi_i]$ contains a matrix to which either SimpleEigenvalue or OneDimEigenspaces applies[3]. MultipleEigenvalue, in contrast, provides no guarantee of finding a factor before the last iteration, but may still do so in a number of situations involving left-hand factors of low order.

The version presented here is but a simple illustration of how the tests described above can be combined, and many improvements are possible in practice. First of all, at the price of minor technical complications, we can take x_0 to be a well-chosen singular point and $\xi_1 = x_0$. The first iteration of the loop on i then require no numerical monodromy computation and parts of it can be performed in exact arithmetic if desired, essentially reducing to van Hoeij's method. Like in the exact case [vH97b, Section 8], it is likely worth trying the eigenring method before using Algorithm 3. Obviously, one should compute information such as degree bounds only once, and, when computing a complete factorization, reuse the monodromy matrices from the caller in recursive calls. Finally, one needs reasonable heuristics to decide how to increase p and t and skip some steps which one expects to fail or to be too costly.

7 EXPERIMENTAL RESULTS

We are working on an implementation of Algorithm 3 in SageMath. Our code is available in an experimental branch of the ore_algebra package[4], under the GNU GPL. It currently implements none of the tricks described outside the pseudo-code blocks, except for the technique of Remark 4.3, which in fact completely replaces lines 5–9 of Algorithm 1, so that irreducibility results are based on monodromy matrices only.

To extensively test our implementation, we developed a generator of random Fuchsian operators, following the theory in [Inc26, §15.4]. After fixing the order r and finite singularities $\xi_1, \dots, \xi_\nu$, the coefficients of a Fuchsian operator $L = \partial^r + \sum_{m=1}^{r} p_m(x)\partial^{r-m}$ can always be written in the form

$$p_m(x) = \sum_{s=1}^{\nu} \frac{P_{m,s}}{(x-\xi_s)^m} + \frac{A_m x^{m\nu-m-\nu} + O(x^{m\nu-m-\nu-1})}{(x-\xi_s)^{m-1}} \qquad (2)$$

for constants $P_{m,s}$ and A_m with $A_1 = 0$. Those constants depend polynomially on the exponents $\alpha_{s,1}, \dots, \alpha_{s,r}$ at the singularity ξ_s and the exponents $\beta_1, \dots, \beta_r$ at ∞, and they satisfy the so-called Fuchs relation, which states that the sum of all the exponents $\alpha_{s,k}$ and β_k is $\frac{1}{2}r(r-1)(\nu-1)$. So any random choice of rational exponents $\alpha_{s,k}$ and β_k satisfying Fuchs's relation provides coefficients $P_{m,s}$ and A_m, while the $\frac{1}{2}(r-1)(r\nu-r-2)$ coefficients hidden under the $O(x^{m\nu-m-\nu-1})$ of (2) can be taken as independent random rational numbers. Generically, the resulting operator is irreducible. We produce reducible operators with constrained exponent patterns by forming products or lclms.

We limit ourselves here to experiments with indecomposable operators with rational singularities and small rational exponents.

Table 1 shows timings for factoring products of two irreducible operators L_1, L_2 of order $r/2$ with the same singularities, using Maple's `DEtools[DFactor]`[5] and using Algorithm 3 recursively. Factors are drawn either to have all their exponential parts of multiplicity one and not repeated across factors or to have, at each singularity, a single exponential part shared by both factors, thus making it have multiplicity r in the product. Unsurprisingly, `DEtools` performs well in the first scenario, whereas our implementation is systematically faster in the second one. Since, when all exponential parts have multiplicity one, our algorithm reduces to van Hoeij's, it is expected that the timings are often comparable. The observed differences may be due to our use of an ordinary base point as well as to time spent computing the eigenring or trying to recursively factor L_1 and L_2. It should be noted that `DEtools` outputs the warning 'factorization may be incomplete' whenever $r \geqslant 6$.

Table 2 compares irreducibility testing on random operators of order r with ν finite singularities and rational exponents, either with all exponential parts of multiplicity 1 or with a single exponential part of multiplicity r at each singularity. Our implementation becomes almost systematically faster than `DEtools[DFactor]` for orders $r > 4$, in relation with the latter having dedicated algorithms for low orders. The warning 'factorization may be incomplete' also appears whenever $r \geqslant 5$.

A cooked-up example will amplify conditions that make the monodromy-based approach win. We chose two operators P and Q with singularities at $0, 1, 2, \infty$, order 2, and integer exponents, thus exponential parts all of multiplicity 2. The product QPP is reducible but indecomposable. We obtain an irreducible operator by considering $QPP + R$ for $R = (x(x-1)(x-2))^{-5}$. Our implementation finds another factorization of QPP in 4.4 seconds and proves the irreducibility of $QPP + R$ in 8 seconds, while `DEtools` fails to find any factor of QPP in about 3 minutes and asserts the irreducibility of $QPP + R$ in a non-certified way in about the same time, in both cases admitting that 'factorization may be incomplete'.

[3] This holds true also in the irreducible case if t is large not only compared to the degrees of actual factors but compared to van Hoeij's bound.

[4] https://github.com/a-goyer/ore_algebra/tree/facto. The experiments reported here use commit `b096ff91`.

[5] With `_Env_eigenring_old` set to true, resulting in significantly better performance.

r ν	mult. = 1		mult. = r	
	classic	new	classic	new
4 2	0.58	5.9	184	2.7
4 3	1.1	4.7	172	6.6
4 4	2.8	6.5	577	10
4 5	16	6.0	2547	15
4 6	58	8.1	1240	7.2
6 2	34	21	1844	58
6 3	50	102	796	43
6 4	141	44	–	76
6 5	357	78	–	265
6 6	1813	63	–	73
8 2	79	3.6	–	1330
8 3	82	91	–	4.6
8 4	402	143	–	5.5
8 5	3042	2780	–	8.6
10 2	574	593	–	–
10 3	593	2414	–	–
10 4	2133	–	–	–

Table 1: Comparison of the classic `DEtools[DFactor]` with our new implementation on products of pairs of operators of order $r/2$ and ν finite singularities[6].

r ν	mult. = 1		mult. = r	
	classic	new	classic	new
2 2	0.15	1.5	49	1.3
2 3	0.17	1.4	0.17	1.3
2 4	0.18	1.3	0.49	1.6
2 5	0.27	1.3	0.89	2.7
2 6	0.35	1.4	0.33	1.4
2 7	0.56	2.4	0.41	4.3
2 8	0.96	3.5	1.7	4.5
3 2	0.20	3.1	35	2.7
3 3	0.35	2.4	4.3	3.0
3 4	0.68	4.1	197	2.2
3 5	1.7	5.4	702	4.0
3 6	5.1	4.9	495	2.5
3 7	17	8.5	340	19
3 8	64	16	1411	13
4 2	0.41	3.5	170	13
4 3	1.1	6.0	144	17
4 4	2.5	9.8	673	6.4
4 5	8.9	11	2131	54
4 6	34	5.8	–	11
4 7	159	77	–	343
4 8	891	59	–	119
5 2	47	3.5	124	187
5 3	57	16	1387	50
5 4	165	24	–	19
5 5	582	42	–	132
5 6	1169	22	–	50
5 7	–	270	–	–
5 8	–	987	–	1759
6 2	114	3.7	101	452
6 3	175	107	2030	344
6 4	404	118	–	88
6 5	1490	376	–	415
6 6	–	87	–	303
6 7	–	1889	–	–
6 8	–	1658	–	–
7 2	167	3.6	463	–
7 3	768	113	–	2037
7 4	1279	229	–	270
7 5	3350	382	–	294
7 6	–	288	–	924
7 7	–	–	–	–
7 8	–	–	–	–
8 2	348	3.7	1633	–
8 3	1146	414	–	–
8 4	2594	261	–	3016
8 5	–	1295	–	–
8 6	–	186	–	–
8 7	–	–	–	–
8 8	–	–	–	–
9 2	815	265	–	–
9 3	2643	573	–	–
9 4	–	1477	–	80
9 5	–	–	–	–
9 6	–	–	–	–
9 7	–	–	–	–

Table 2: Comparison of the classic `DEtools[DFactor]` with our new implementation on irreducible operators of order r and ν finite singularities[6].

REFERENCES

[Bek94] Emanuel Beke. Die Irreducibilität der homogenen linearen Differentialgleichungen. *Mathematische Annalen*, 45(2):278–294, 1894.

[Bou90] Nicolas Bourbaki. *Algebra. II. Chapters 4–7.* Elements of Mathematics (Berlin). Springer-Verlag, Berlin, 1990. Translated from the French by P. M. Cohn and J. Howie.

[BR90] László Babai and Lajos Rónyai. Computing irreducible representations of finite groups. *Mathematics of computation*, 55(192):705–722, 1990.

[Bre10] Murray R. Bremner. How to compute the Wedderburn decomposition of a finite-dimensional associative algebra. *arXiv:1008.2006 [math-ph]*, August 2010.

[Bro94] Manuel Bronstein. An improved algorithm for factoring linear ordinary differential operators. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 336–340, 1994.

[BRS19] Alin Bostan, Tanguy Rivoal, and Bruno Salvy. Explicit degree bounds for right factors of linear differential operators. *Bulletin of the London Mathematical Society*, 2019.

[CDDM] Frédéric Chyzak, Thomas Dreyfus, Philippe Dumas, and Marc Mezzarobba. First-order factors of linear Mahler operators. In preparation.

[Dix70] John D. Dixon. Computing irreducible representations of groups. *Mathematics of Computation*, 24(111):707–712, 1970.

[Ebe89] Wayne Michael Eberly. *Computations for Algebras and Group Representations.* PhD thesis, University of Toronto, 1989.

[Ebe91] Wayne Michael Eberly. Decomposition of algebras over finite fields and number fields. *Computational complexity*, 1(2):183–210, June 1991.

[Gab71] John R. Gabriel. Numerical methods for reduction of group representations. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation - SYMSAC '71*, pages 180–182, Los Angeles, California, United States, 1971. ACM Press.

[Goy21] Alexandre Goyer. A Sage package for the symbolic-numeric factorization of linear differential operators. *ACM Communications in Computer Algebra*, 55(2):44–48, 2021.

[Gri90] Dmitrii Yur'evich Grigor'ev. Complexity of factoring and calculating the GCD of linear ordinary differential operators. *Journal of Symbolic Computation*, 10(1):7–37, 1990.

[Hil76] Einar Hille. *Ordinary Differential Equations in the Complex Domain.* John Wiley & Sons, 1976.

[HR94] Derek F. Holt and Sarah Rees. Testing modules for irreducibility. *Journal of the Australian Mathematical Society*, 57(1):1–16, 1994.

[Inc26] Edward Lindsay Ince. *Ordinary Differential Equations.* Dover Publications, New York, 1926.

[JKM13] Fredrik Johansson, Manuel Kauers, and Marc Mezzarobba. Finding hyperexponential solutions of linear ODEs by numerical evaluation. In *International Symposium on Symbolic and Algebraic Computation*, 2013.

[Kap72] Irving Kaplansky. *Fields and Rings.* University of Chicago Press, 1972.

[Lam98] Tsit Yuen Lam. A theorem of Burnside on matrix rings. *The American Mathematical Monthly*, 105(7):651–653, August 1998.

[LM14] Alberto Llorente Mediavilla. *Métodos numérico-simbólicos para calcular soluciones liouvillianas de ecuaciones diferenciales lineales.* PhD thesis, Universidad de Valladolid, 2014.

[LP10] Klaus Lux and Herbert Pahlings. *Representations of groups: a computational approach*, volume 124. Cambridge University Press, 2010.

[MS16] Claude Mitschi and David Sauzin. *Divergent Series, Summability and Resurgence II. Monodromy and Resurgence.* Springer, 2016.

[Par84] Richard A. Parker. The computer calculation of modular characters (the Meat-Axe). In *Proceedings of the London Mathematical Society Symposium on Computational Group Theory*, pages 267–274, 1984.

[Pie82] Richard S. Pierce. *Associative Algebras.* Springer, 1982.

[Poo36] Edgar Girard Croker Poole. *Introduction to the Theory of Linear Differential Equations.* Clarendon Press, Oxford, 1936.

[Sch89] Fritz Schwarz. A factorization algorithm for linear ordinary differential equations. In *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, pages 17–25, 1989.

[Sin96] Michael F. Singer. Testing reducibility of linear differential operators: a group theoretic perspective. *Applicable Algebra in Engineering, Communication and Computing*, 7(2):77–104, 1996.

[Spe12] David E Speyer. Invariant subspaces if f is defined by more than one matrix. Mathematics Stack Exchange, 2012.

[6] All computations on a processor Intel i9-10885H (8 cores, 16 threads), with concurrent jobs but all BIOS settings that could induce variations in performance disabled. Calculations stopped after 3600 seconds.

URL:https://math.stackexchange.com/q/185001 (version: 2017-04-13).

[Sun19] Mengxiao Sun. *On the Complexity of Computing Galois Groups of Differential Equations*. PhD thesis, City University of New York, 2019.

[Tsa94] Sergey Petrovich Tsarev. Problems that appear during factorization of ordinary linear differential operators. *Programming and Computer Software*, 20(1), 1994.

[vdH07a] Joris van der Hoeven. Around the numeric-symbolic computation of differential Galois groups. *Journal of Symbolic Computation*, pages 236–264, 2007.

[vdH07b] Joris van der Hoeven. Efficient accelero-summation of holonomic functions. *Journal of Symbolic Computation*, 42(4):389–428, 2007.

[vdH10] Joris van der Hoeven. Ball arithmetic. In Arnold Beckmann, Christine Gaßner, and Bededikt Löwe, editors, *Logical approaches to Barriers in Computing and Complexity*, number 6 in Preprint-Reihe Mathematik, pages 179–208. Ernst-Moritz-Arndt-Universität Greifswald, February 2010. International Workshop.

[vdPS03] Marius van der Put and Michael F. Singer. *Galois Theory of Linear Differential Equations*. Springer-Verlag Berlin Heidelberg, 2003.

[vH96] Mark van Hoeij. Rational solutions of the mixed differential equation and its application to factorization of differential operators. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 219–225, 1996.

[vH97a] Mark van Hoeij. The diffop package, 1997. https://www.math.fsu.edu/~hoeij/daisy/lib/DEtools/diffop/.

[vH97b] Mark van Hoeij. Factorization of differential operators with rational functions coefficients. *Journal of Symbolic Computation*, 24(5):537–561, 1997.