Thèse de doctorat
NNT : 2025UPASM031

Efficient Algorithms for Creative Telescoping using Reductions

Algorithmes efficaces pour le télescopage créatif utilisant des réductions

Thèse de doctorat de l'université Paris-Saclay

École doctorale n◦ 574 : mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques aux interfaces
Graduate School : Mathématiques
Référent : Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche
Inria Saclay-Île-de-France (Université Paris-Saclay, Inria)
sous la direction de **Frédéric Chyzak**, directeur de recherche
et le co-encadrement de **Pierre Lairez,** chargé de recherche

Thèse soutenue à Paris-Saclay, le 5 décembre 2025, par

Hadrien Brochet

Composition du Jury

Membres du jury avec voix délibérative

Shaoshi Chen Professeur, Chinese Academy of Sciences	Rapporteur & Examinateur
Alban Quadrat Directeur de recherche, Inria Paris	Rapporteur & Examinateur
Thomas Cluzeau Professeur des universités, Université de Limoges	Examinateur
Christoph Koutschan Professeur, Johann Radon Institute for Computational and Applied Mathematics	Examinateur
Gleb Pogudin Professeur, École Polytechnique, Institut Polytechnique de Paris	Examinateur
Anna-Laura Sattelberger Professeure, Max Planck Institute for Mathematics in the Sciences	Examinatrice

ÉCOLE DOCTORALE
de mathématiques Hadamard (EDMH)

Title: Efficient Algorithms for Creative Telescoping using Reductions

Keywords: Symbolic integration, Symbolic summation, Creative telescoping, Weyl algebras

Abstract: Creative telescoping is an algorithmic method introduced by Zeilberger for computing linear functional equations satisfied by parametric sums and integrals. Initially developed for sums of hypergeometric functions and integrals of hyperexponential functions, the method has since been extended to broader classes of functions, including ∂-finite and holonomic functions. In recent years, a new family of creative telescoping algorithms based on reductions has emerged, addressing efficiency limitations in earlier approaches. This thesis continues in this direction by introducing two new reduction-based algorithms. The first one is designed for computing univariate sums of ∂-finite functions, and the second one for computing multivariate integrals of holonomic functions.

Creative telescoping algorithms rely on the knowledge of sufficiently many linear functional equations satisfied by the function f to be summed or integrated. These equations are represented by a left ideal in an algebra of operators, known as the annihilator of f. For ∂-finite functions, this ideal resides in an Ore algebra, while for holonomic functions, it lies within a Weyl algebra. In the ∂-finite case, annihilators are typically easy to compute. In contrast, obtaining a complete set of annihilating operators for holonomic functions is significantly more challenging. When a function is both ∂-finite and holonomic, one approach for computing its holonomic annihilator consists in computing its ∂-finite annihilator and performing an operator called the Weyl closure. Although an algorithm for computing the Weyl closure exists, it suffers from practical inefficiency. This thesis proposes a new algorithm for computing holonomic approximations of the Weyl closure of left ideals in the Weyl algebra.

Titre : Algorithmes efficaces pour le télescopage créatif utilisant des réductions

Mots clés : Intégration symbolique, Sommation symbolique, Télescopage créatif, Algèbre de Weyl

Résumé : Le télescopage créatif est une méthode algorithmique introduite par Zeilberger pour calculer des équations fonctionnelles linéaires satisfaites par des sommes et des intégrales à paramètres. Initialement développée pour les sommes de fonctions hypergéométriques et les intégrales de fonctions hyperexponentielles, cette méthode a depuis été étendue à des classes de fonctions plus larges tels que les fonctions ∂-finies et holonomes. Au cours des dernières années, une nouvelle génération d'algorithmes de téléscopage créatif, basée sur les réductions a vu le jour afin de remédier à des problèmes d'efficacités inhérents aux approches précédentes. Cette thèse s'inscrit dans cette lignée en introduisant deux nouveaux algorithmes basés sur des réductions. Le premier est conçu pour calculer des sommes univariées de fonctions ∂-finies, et le second pour calculer des intégrales multivariées de fonctions holonomes.

Les algorithmes de téléscopage créatif reposent sur la connaissance d'un nombre suffisant d'équations fonctionnelles linéaires satisfaites par la fonction f à sommer ou intégrer. Ces équations sont représentées par un idéal à gauche dans une algèbre d'opérateurs, appelé annulateur de f. Pour les fonctions ∂-finies, cet idéal appartient à une algèbre d'Ore, tandis que pour les fonctions holonomes, il se trouve dans une algèbre de Weyl. Dans le cas ∂-fini, les annulateurs sont en général faciles à calculer. En revanche, dans le cas holonome ce problème est considérablement plus difficile. Lorsqu'une fonction est à la fois ∂-finie et holonome, une approche pour calculer son annulateur holonome consiste à déterminer son annulateur ∂-fini puis à effectuer une opération appelée clôture de Weyl. Bien qu'un algorithme permettant de calculer cette clôture de Weyl existe, il se révèle peu efficace en pratique. Cette thèse introduit un nouvel algorithme pour calculer une approximation holonome de la clôture de Weyl d'un idéal à gauche d'une algèbre de Weyl.

Efficient Algorithms for Creative Telescoping using Reductions

Hadrien Brochet

November 19, 2025

Acknowledgements

I would first like to thank my two PhD advisors, Frédéric Chyzak and Pierre Lairez, for their guidance and support throughout these three years and four months of doctoral work. You have always been available and helpful, and you have shown great patience by allowing me to work at my own pace. I am also grateful for the many opportunities you gave me to attend conferences, which is far from common.

My gratitude also goes to my Master's thesis advisor, Bruno Salvy, for the guidance and support he likewise provided. You made me discover research and the field of computer algebra, which motivated me to pursue a PhD in the same direction.

I am thankful to the members of my thesis committee for agreeing to examine my thesis: Shaoshi Chen, Thomas Cluzeau, Christoph Koutschan, Gleb Pogudin, Alban Quadrat, and Anna-Laura Sattelberger. I am especially grateful to Shaoshi and Alban for the time and care they devoted to reviewing my manuscript.

I warmly thank all the colleagues I met at Inria who made daily life there so enjoyable: Albin Ahlbäck, Alin Bostan, Ricardo Buring, Geoffrey Datchanamourtty, Philippe Dumas, Guy Fayolle, Claudia Fevola, Alexandre Goyer, Alexandre Guillemot, Pingchuan Ma, Jaali Mazzaggio, Marc Mezzarobba, Arnaud Minondo, Rafael Mohr, Yulia Mukhina, Hadrien Notarantonio, Eric Pichon-Pharabod, Linus Sommer, Catherine St-Pierre, Aleksandr Storozhenko, Théo Ternier.

I am also grateful to the many colleagues I met during conferences and travels for the stimulating scientific discussions and the enjoyable moments shared (including climbing trips during conferences): Florent Bréhard, Khalil Gorbach, Alaa Ibrahim, Camille Pinto, Rémi Prébet, Carsten Schneider, as well as many others.

My warmest thanks go to my friends for the time spent outside work, and for the many vacations and climbing sessions we enjoyed together: Antoine, Charlotte, Hugoline, Julie, Léna, Odelin, Solène, Valentin.

And finally I would like to thank my family for their constant support.

Contents

1 Introduction

This thesis is devoted to the computation of sums and integrals that depend on one or more parameters. Such quantities arise naturally in many areas of mathematics and physics, including combinatorics, number theory, algebraic geometry, statistical physics, and quantum field theory. In this context, computation takes on a particular meaning, as parametric sums and integrals can generally not be expressed in terms of elementary functions. A well established approach in computer algebra is to represent them as solutions of some systems of linear functional equations with coefficients that are either polynomials or rational functions. Typically, these systems consist of differential equations, recurrence relations or a combination of both. Over the past few decades, a powerful technique known as creative telescoping has emerged for computing such representations, particularly when the summand or integrand is either D-finite or holonomic. More recently, a new family of creative telescoping algorithms based on reduction techniques has been developed to address certain inefficiencies in the earlier methods. This thesis focuses on extending this family by proposing more efficient and more general algorithms.

1.1 D-finite and holonomic functions

The method of creative telescoping has been successfully applied to integrands belonging to the large classes of D-finite functions and holonomic functions. These two classes are very often mixed up, and their names are mistakenly used as synonyms, as they have a large non-empty intersection. In this thesis I need to make a clear distinction between them because first, they do not have the same expressivity and second, they correspond to distinct algorithms with different representations of functions on a computer. This section introduces the algebras of differential operators, and then defines and compares these two classes of functions in the purely differential setting.

1.1.1 Operator algebra

Let $\mathbb{K}$ be a field of characteristic 0 and let W_n be the nth Weyl algebra

$$\mathbb{K}[x_1, \dots, x_n]\langle \partial_1, \dots, \partial_n \rangle,$$

which is the algebra generated by the variables $x_1, \dots, x_n, \partial_1, \dots, \partial_n$ and subject to the commutation rules $x_i x_j = x_j x_i$, $\partial_i \partial_j = \partial_j \partial_i$, $x_i \partial_j = \partial_j x_i$ and $\partial_i x_i = x_i \partial_i + 1$ for any distinct i, j. This algebra acts naturally on functions via $x_i \cdot f = x_i f$ and $\partial_i \cdot f = \frac{\partial f}{\partial x_i}$, inducing a module structure on many classes such as

- the class of meromorphic functions on an open set of $\mathbb{C}^n$,
- the class of formal Laurent series in n variables,
- the class of distributions on an open set of $\mathbb{R}^n$.

This action allows us to represent homogeneous linear differential equations with polynomial coefficients as elements of W_n. Indeed, the differential equation

$$\sum_{\alpha_1,\ldots,\alpha_n} c_{\alpha_1,\ldots,\alpha_n}(x_1,\ldots,x_n)\frac{\partial^{\alpha_1+\cdots+\alpha_n} f}{\partial^{\alpha_1}x_1\cdots\partial^{\alpha_n}x_n} = 0, \tag{1.1}$$

where the sum is taken over a finite set of indices $\alpha_1,\ldots\alpha_n$ and where $c_{\alpha_1,\ldots,\alpha_n}$ are coefficients in $\mathbb{K}[x_1,\ldots,x_n]$, is represented by the element

$$\sum_{\alpha_1,\ldots,\alpha_n} c_{\alpha_1,\ldots,\alpha_n}(x_1,\ldots,x_n)\partial_1^{\alpha_1}\cdots\partial_n^{\alpha_n} \tag{1.2}$$

of W_n. Moreover, the set of all equations obtained from (1.1) by differentiating arbitrarily many times and multiplying by polynomials of $\mathbb{K}[x_1,\ldots,x_n]$ corresponds exactly to the left ideal of W_n generated by the element in (1.2). I will speak of an operator to mean an element of a ring acting on a class of functions, and henceforth view elements of W_n as operators.

Note that the division by polynomials is not permitted in W_n. This limitation motivates the definition of the rational Weyl algebra R_n which extends W_n by allowing rational function coefficients in $x_1,\ldots,x_n$. Formally it is defined as the algebra

$$\mathbb{K}(x_1,\ldots,x_n)\langle\partial_1,\ldots,\partial_n\rangle \tag{1.3}$$

subject to the relations $\partial_i\partial_j = \partial_j\partial_i$ and

$$\partial_i r = r\partial_i + \frac{\partial r}{\partial x_i} \tag{1.4}$$

for any $i,j \in \{1,\ldots,n\}$ and any rational function $r \in \mathbb{K}(x_1,\ldots,x_n)$. The algebra R_n is used to represent linear homogeneous differential equations with rational function coefficients. Besides, it similarly defines a natural action on:

- the class of meromorphic functions on an open set of $\mathbb{C}^n$,
- the class of formal Laurent series in n variables.

However, it does not define an action on distributions, as illustrated by the following example.

Example 1. The dirac distribution δ is defined for any smooth function with compact support φ by the formula

$$\langle\delta,\varphi\rangle = \varphi(0). \tag{1.5}$$

As a distribution, it satisfies the equation

$$x \cdot \delta = \langle \delta, x \rangle = 0. \tag{1.6}$$

If R_n defined an action on the class of distributions, we would have for any $a, b \in R_n$

$$(ab) \cdot \delta = a \cdot (b \cdot \delta), \tag{1.7}$$

but taking $a = 1/x$ and $b = x$ gives a contradiction as δ is not the zero distribution.

The ability to represent objects that are annihilated by polynomials is the main advantage of homology. However, as we will see later, this expressiveness comes at a computational cost.

1.1.2 D-finite functions

The notion of D-finite functions was introduced for univariate power series by Stanley [125], its generalization to multivariate functions was first used by Zeilberger [143] and was later formalized by Lipshitz [92]. A recent book by Kauers [82] provides an up-to-date introduction to the subject.

Let $\mathcal{F}$ be a class of functions on which R_n acts and let $f \in \mathcal{F}$.

Definition 2. The function f is said to be D-finite if the $\mathbb{K}(x_1, \ldots, x_n)$-vector space $R_n \cdot f$ is finite-dimensional.

Definition 2 is equivalent to the existence, for each i, of a linear ordinary differential equation in ∂_i with coefficients in $\mathbb{K}(x_1, \ldots, x_n)$ satisfied by f. The notion of D-finiteness captures the idea that any derivatives of $\partial_1^{\ell_1} \cdots \partial_n^{\ell_n} \cdot f$ can be expressed in terms of a finite number of them, which form a basis of $R_n \cdot f$. From an algorithmic point of view, it is convenient to represent a D-finite function f through its annihilator in R_n.

Definition 3. The annihilator of f in R_n is the left ideal

$$\operatorname{ann}_{R_n}(f) = \{P \in R_n \mid P \cdot f = 0\}. \tag{1.8}$$

This representation is particularly useful, since it allows us to work with operators instead of functions. The relation between f and its annihilator is captured by the following classical isomorphism.

Lemma 4. *The module $R_n \cdot f$ is isomorphic to $R_n / \operatorname{ann}_{R_n}(f)$ as R_n-module. Consequently, the function f is D-finite if and only if the quotient $R_n / \operatorname{ann}_{R_n}(f)$ is a finite-dimensional $\mathbb{K}(x_1, \ldots, x_n)$-vector space.*

Lemma 4 is illustrated on the following example.

Example 5. The function $e^{-x(a+1)} \ln(x)$ is D-finite as it satisfies the following two equations:

$$\frac{\partial f}{\partial a} + xf = 0, \tag{1.9}$$

$$x \frac{\partial^2 f}{\partial x^2} + (2ax + 2x + 1) \frac{\partial f}{\partial x} + (a + 1)(ax + x + 1) f = 0. \tag{1.10}$$

Using multiples of these two operators, all derivatives of f can be expressed as linear combinations with rational function coefficients of f and $\frac{\partial f}{\partial x}$. In other words, we have the following isomorphism of vector spaces:

$$R_n/\operatorname{ann}_{R_n}(f) \simeq \mathbb{K}(x,a)f \oplus \mathbb{K}(x,a)\frac{\partial f}{\partial x}. \tag{1.11}$$

A D-finite holomorphic function f can be uniquely determined by its annihilator in R_n together with finitely many initial conditions, provided that singularities do not interfere (see, e.g., [82, pp. 363-364]). This makes the annihilator a valid and effective data-structure for representing f. Computing a generating set for such an annihilator is a classical problem [94, 115, 131, 45], with implementations available in several computer algebra systems: in MAPLE via the `Mgfun` package [46], in MATHEMATICA via the `HolonomicFunctions` package [85], and in SAGEMATH via the `ore_algebra` package [83].

1.1.3 Holonomic functions

The theory of holonomic D-modules is a branch of algebraic analysis concerned with studying systems of linear partial differential equations from an algebraic perspective. It originated in the 1970s with the work of Bernstein [12, 13] and Kashiwara [81]. As a side note, Kashiwara recently received the Abel prize notably for his foundational work on D-module theory. In the 1990s, it was introduced to the computer algebra community by Zeilberger [144], alongside the method of creative telescoping. An accessible introduction to the theory is provided in Coutinho's book [54], while the algorithmic aspects are introduced in [114]. For a more advanced and comprehensive treatment, see Björk's book [16].

While D-finiteness is defined with respect to the rational Weyl algebra R_n, holonomy is defined in terms of the (polynomial) Weyl algebra W_n. Let $\mathcal{F}$ be a set on which W_n acts and let $f \in \mathcal{F}$. By a slight abuse of terminology, I will refer to f as a function, since in most cases it is one, although in some contexts it could be, for example, a distribution. As before, the function f is represented by its annihilator, but now this annihilator is taken in W_n.

Definition 6. The annihilator of f in W_n is the left ideal

$$\operatorname{ann}_{W_n}(f) = \{P \in W_n \mid P \cdot f = 0\}. \tag{1.12}$$

To define the notion of holonomic function, I first need to define the notion of holonomic modules. A finitely generated W_n-module is called holonomic if its dimension is exactly n. However, defining properly the dimension of such a module requires quite a lot of mathematical preliminaries, so I instead use an equivalent property for modules of the form W_n/I, where I is a left ideal of W_n. This equivalent property is described in Characterization 7. The implication "holonomic $\implies$ characterization" was first proved by Zeilberger [144]. Since then, the characterization has sometimes been taken as the definition itself (see [82, Definition 4.67]). However, I have not found a published proof of the converse implication. A proof will be given in Section 4.2.1.1.

Characterization 7. Let I be a non-zero left ideal of W_n. The module W_n/I is holonomic if and only if for any subset $S \subset \{x_1, \ldots, x_n, \partial_1, \ldots, \partial_n\}$ of cardinality $n+1$, the intersection $\mathbb{K}\langle S\rangle \cap I$ is non-empty, where $\mathbb{K}\langle S\rangle$ denotes the subalgebra of W_n generated by the elements of S.

This type of property is typically used to prove that the dimension of the module W_n/I is at most n. However, a fundamental result due to Bernstein [13] asserts that the dimension of any non-zero W_n-module is always at least n. Together, these results yield the upper and lower bounds required for the characterization.

Definition 8. The function f is holonomic if the module $W_n/\operatorname{ann}_{W_n}(f)$ is holonomic.

Example 9. The rational function $1/(x^2 - y^3)$ is holonomic as its annihilator is generated by the three operators

$$\begin{gathered} \partial_x(x^2 - y^3), \qquad \partial_y(x^2 - y^3), \\ 3x\partial_x + 2y\partial_y + 6, \end{gathered} \tag{1.13}$$

from which two additional operators can be obtained by Gröbner basis computation to apply Characterization 7:

$$\begin{gathered} 9y^3\partial_x^2 - 4y^2\partial_y^2 + -10y\partial_y, \\ 27x^3\partial_x^3 + 8x^2\partial_y^3 + 135x^2\partial_x^2 + 105x\partial_x. \end{gathered} \tag{1.14}$$

A more algorithmic criterion for deducing holonomy directly from Eq. (1.13) will be presented in Section 4.2.1.1.

More generally, it is well-known that every rational function is holonomic.

Computing annihilators in W_n

The computation of annihilators in W_n is more complicated than in R_n. One approach is to start from known annihilators of classical functions (e.g., rational functions, exp, log, sin, cos, …) and apply closure algorithms for sums and products to deduce annihilators of more general functions. These closure properties were first presented in [144], building on Bernstein's results [13]. A concise algorithmic summary is also given in [100, Section 3]. This approach mirrors the standard method for computing D-finite annihilators. However, two main difficulties arise. First the computation of annihilators for rational functions becomes challenging (see Section 4.1 for the state of the art), and second, the closure properties become likewise considerably more computationally expensive.

Another approach exists when a function is both holonomic and D-finite. In this case, the annihilator in W_n can be derived from its counterpart in R_n. To illustrate how it can be done, let us consider again the rational function $1/(x^2 - y^3)$. As a D-finite function, its annihilator in R_2 is generated by $p_1 = \partial_x(x^2 - y^3)$ and $p_2 = \partial_y(x^2 - y^3)$. From these two operators, one can deduce a third:

$$3xp_1 - 2yp_2 = (x^2 - y^3)(3x\partial_x - 2y\partial_y + 6). \tag{1.15}$$

However, without dividing by $x^2 - y^3$ one cannot deduce the last operator in Eq. (1.13), which is required for generating the holonomic annihilator of $1/(x^2-y^3)$. More generally, the holonomic ideal of a function f can be obtained by taking a generating set G with polynomial coefficients of its D-finite annihilator and computing the Weyl closure of $W_n G$:

$$\mathrm{Cl}(W_n G) = R_n G \cap W_n. \tag{1.16}$$

An algorithm for computing the Weyl closure was proposed by Tsai [134]. It proceeds by localizing the ideal $W_n G$ at a polynomial vanishing on the singular locus of f. In practice, however, its performance does not permit handling cutting edge examples, especially when the polynomial vanishing on the singular locus of f is large.

1.1.4 D-finiteness versus holonomy

The notions of D-finiteness and holonomy coincide for many classes of functions.

Theorem 10. *Let $\mathcal{F}$ be a set on which both R_n and W_n act and let $f \in \mathcal{F}$. The function f is holonomic if and only if it is D-finite.*

An elementary proof of this result was given by Takayama [131], who attributes the result to Kashiwara. The key difference between these two notions lies in how functions are represented on a computer. In the D-finite setting, a function is represented by a zero-dimensional ideal, its annihilator in R_n. In contrast, in the holonomic setting, it is represented by its annihilator in W_n which is an ideal of dimension n. In practice, we observe that holonomic annihilators are much more complicated to compute. The most striking examples are rational functions. In the D-finite setting, the annihilator of a rational function R is generated by $R\partial_i - \frac{\partial R}{\partial x_i}$ for all i, which can be computed by hand. Now, consider for example the rational function

$$\frac{1}{1-(1-xy)z-txyz(1-x)(1-y)(1-z)}, \tag{1.17}$$

which appears in Beukers' work [14]. The computation of its holonomic annihilator with Plural [90], currently the fastest software for this purpose, takes around 30 seconds.

I see two main arguments explaining why computing annihilators is more difficult in the holonomic setting. The first is based on an analogy with the complexity of computing Gröbner bases of polynomial ideals. In the polynomial case, it is well-known that the worst-case complexity of computing a Gröbner basis is single-exponential in the number of variables for zero-dimensional ideals [96], and double-exponential for positive-dimensional ideals [95, 59]. Although to the best of my knowledge no analogous complexity results are known for Weyl algebras, it is reasonable to expect a similar behavior. The second argument is that holonomic annihilators contain more information than their D-finite counterpart. As discussed earlier, holonomic annihilators can be expressed as the Weyl closure of some ideal, and the Weyl closure is known to encode all desingularizations of the operators in the input ideal. The desingularization of a univariate differential equation is a classical problem known to be computationally costly. In the language of operators, given an operator $P \in R_1$, it consists in finding a left multiple Q of P which singularities are strictly included in those of P. This concept was extended to D-finite systems by Chen, Kauers, Li and Zhang [41].

Despite this computational drawback, the theory of holonomy has proven particularly fruitful for establishing existence results and proving termination of algorithms. Notably, Kashiwara's theorem guarantees that the integral of a holonomic module is holonomic (see Section 3.2.4). One immediate consequence is the existence of a non-trivial creative telescoping relation in the purely differential setting (see Eq. (1.25) in next section). Another argument in favor of holonomy is its expressivity: it makes it possible to represent differentiable objects that are annihilated by polynomials. This is for example the case of some distributions as mentioned earlier.

1.1.5 Extension to more general settings

The class of D-finite functions generalizes to functions whose variables are associated to other operators than a derivation, such as a shift or more generally an Ore operator. This broader class is known as ∂-finite functions [45], although the term D-finite is sometimes also used to refer to these functions. Precise definitions of Ore algebras and ∂-finite functions are given in Section 2.3.1.

The notion of holonomy was extended to sequences by Zeilberger by means of their generating series [144]. More recently, Kauers generalized this definition to Ore algebras [82, definition 4.67], using an adaptation of Characterization 7. This definition guarantees that holonomic functions always admit creative telescoping relations for summation and integration. However, the equivalence between holonomy and D-finiteness, as established in Theorem 10, does not extend to the general Ore setting. A classical counterexample is the bivariate sequence $(1/(n^2+k^2))_{n,k}$, which is D-finite but not holonomic (see, e.g., [84, Example 2.23]).

1.2 Creative telescoping

1.2.1 The creative telescoping method

Creative telescoping is a powerful method for computing linear functional equations satisfied by sums and integrals. The method consists in finding an equation of a specific form satisfied by the summand (respectively, the integrand), so that it can be summed (respectively, integrated) in a straightforward way.

Let $\mathbb{K}$ be a field of characteristic zero. For a sum

$$S(n) = \sum_{k=a}^{b} f(n,k),$$

the method consists in finding a function g and coefficients $c_0(n), \dots, c_\ell(n) \in \mathbb{K}(n)$ such that

$$c_\ell(n) f(n+\ell,k) + \cdots + c_0(n) f(n,k) = g(n,k+1) - g(n,k). \tag{1.18}$$

Assuming everything is well-defined, this equation can be summed over k to obtain the linear inhomogeneous recurrence

$$c_\ell(n)S(n+\ell) + \cdots + c_0(n)S(n) = g(n, b+1) - g(n, a). \tag{1.19}$$

Similarly, for an integral

$$I(t) = \int_a^b f(t,x)dx,$$

the goal is to find a function g and coefficients $c_0(x), \ldots, c_\ell(x) \in \mathbb{K}(x)$ satisfying

$$c_\ell(x)\frac{\partial^\ell f(t,x)}{\partial t^\ell} + \cdots + c_0(x)f(t,x) = \frac{\partial g(t,x)}{\partial x}. \tag{1.20}$$

Integrating the above equation with respect to x produces the linear inhomogeneous differential equation

$$c_\ell(x)\frac{\partial^\ell I(t)}{\partial t^\ell} + \cdots + c_0(x)I(t) = g(t,b) - g(t,a). \tag{1.21}$$

In many applications, the right-hand sides of (1.19) and (1.21) vanish, resulting in homogeneous equations. This occurs, for instance, in a complex setting when integrating over a loop that does not intersect the singularities of the function g. In general, evaluating these two right-hand sides algorithmically can be complicated due to the presence of singularities. As a consequence, most creative telescoping algorithms focus on computing relations of the form (1.18) or (1.20). When the right-hand sides can be predicted to vanish, some algorithms, including the so-called reduction-based algorithms, can compute exclusively their left-hand sides.

For the method to be fully algorithmic, the form of g must be prescribed. In the discrete case, g is taken as a finite linear combination of shifts of f:

$$g = \sum_{i,j} a_{i,j}(n,k)f(n+i, k+j), \tag{1.22}$$

where each $a_{i,j}(n,k)$ is a rational function in $\mathbb{K}(n,k)$. In the continuous case, g is a finite linear combination of derivatives of f:

$$g = \sum_{i,j} a_{i,j}(x,t)\frac{\partial^{i+j} f(x,t)}{\partial x^i, \partial t^j}, \tag{1.23}$$

where $a_{i,j}(x,t)$ belongs to $\mathbb{K}(x,t)$.

These choices allow the problem to be expressed in operator form. In the discrete setting, the task is to find operators $L \in \mathbb{K}(n)\langle S_n\rangle$ and $G \in \mathbb{K}(n,k)\langle S_n, S_k\rangle$ such that

$$L \cdot f = (S_k - 1)G \cdot f. \tag{1.24}$$

In the continuous setting, the goal is to determine $L \in \mathbb{K}(t)\langle \partial_t\rangle$ and $G \in \mathbb{K}(t,x)\langle \partial_t, \partial_x\rangle$ satisfying

$$L \cdot f = \partial_x G \cdot f. \tag{1.25}$$

In both cases, L is referred to as a telescoper, G as its associated certificate, and the resulting equation as a creative telescoping relation.

More general formulations of creative telescoping exists, allowing the treatment of multiple sums and integrals, as well as sums and integrals depending on several parameters. The first situation will appear in Chapter 3 for integrals, and the second in Chapter 2 for sums.

Apart from creative telescoping, other methods have been developed for handling parametric sums and integrals. A notable line of research, initiated by Karr [78, 79] and subsequently advanced by Schneider [116, 117, 118, 119, 120, 121, 122, 123, 124], is based on the theory of difference fields and allows the use of non-linear expressions. This method has been adapted to the differential setting by Raab [106, 107] based on the work of Risch [110, 111]. A different approach to integration, proposed by Regensburger, Rosenkranz, and collaborators [72, 109, 112, 113], enriches the underlying algebra with integration and evaluation operators, thereby providing a framework for studying algebraically linear differential equations with delay. In the recent years, this line of work has seen significant progress, with many effective results obtained by Cluzeau, Pinto, Quadrat, and their coauthors [105, 52, 53].

1.2.2 A brief history of creative telescoping

The term *creative telescoping* was first introduced by van der Poorten [137] in 1979, as he sought to rigorously justify a claim left unproven by Apéry in his celebrated proof of the irrationality of $\zeta(3)$. The claim required to show that the binomial sum

$$S(n) = \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 \tag{1.26}$$

satisfies the now famous recurrence

$$n^3 u(n) - (34n^3 - 51n^2 + 27n - 5)u(n-1) + (n-1)^3 u(n-2) = 0. \tag{1.27}$$

Later on, this method was brought to the computer algebra community by Doron Zeilberger in 1990 [144]. Since then, a lot of effort has been put into developing efficient algorithms, some of which have become essential tools in symbolic computation. The remainder of this section gives an overview of this history.

1.2.2.1 Algorithms based on elimination

In his foundational article [144], Zeilberger proposed an algorithm for computing univariate integrals of bivariate holonomic functions and univariate sums of bivariate holonomic sequences. The method was influenced by earlier work of Fasenmyer [62, 63], and by its automation by Verbaeten [138, 139]. For integration (the method for summation being analogous), it follows from Characterization 7 that there exist operators L and G satisfying Eq. (1.25) and independent of the integration variable x. Zeilberger's method takes advantage of this property. His algorithm eliminates x from the annihilator of the integrand by using what he calls Sylvester's dialytic elimination [127], yielding creative telescoping relations. However, the requirements that G has no denominator and does not involve x (resp. k) impose a strong restriction on its form. As a result, the algorithm cannot handle complicated integrals, and Zeilberger himself referred [145] to it as "the slow algorithm" [144] in contrast to a faster variant [145] he introduced later.

An alternative elimination method for parametric integration was first proposed by Takayama [129, 128], and later improved by Oaku and Takayama [102]. This approach is based on an infinite-dimensional generalization of Gröbner bases in the Weyl algebras. Unlike Zeilberger's algorithm, Takayama's method allows x to appear in the certificate but does not require its explicit computation. In 2013, Oaku [100] proposed an algorithm for computing annihilators of distributions of the form $f\mathbb{1}_A$ where $\mathbb{1}_A$ is the indicator function of a semi-algebraic set A and f a holonomic function. He showed that Takayama's algorithm could be used with distributions, making it possible to deal with integrals of holonomic function over semi-algebraic domains automatically. Such cases are beyond the reach of the theory of D-finiteness.

A limitation shared by both methods is their reliance on the knowledge of a holonomic annihilator of the integrand, which can be difficult to compute. Both approaches, however, generalize naturally to multiple integrals.

1.2.2.2 Algorithms based on the resolution of linear functional equations

Due to the inefficiencies of his first algorithm, Zeilberger turned to designing methods for more restricted classes of functions. He introduced a "fast algorithm" for sums of bivariate hypergeometric sequences, which was soon extended to integrals of hyperexponential functions by Almkvist and himself [3]. Wilf and Zeilberger [142] introduced a subclass of functions called *proper hypergeometric functions* for which these algorithms could be iterated to deal with multi-sums and integrals. Almkvist's and Zeilberger's ideas were later extended by Chyzak [47] to handle the broader class of ∂-finite functions, for both summation and integration. In the differential case (the summation case being analogous), these algorithms proceed by making an Ansatz for the telescoper L of the form

$$\Big(c_\ell(t)\partial_t^\ell + \cdots + c_0(t)\Big) \cdot f, \tag{1.28}$$

where $c_0, \ldots, c_\ell$ are indeterminate rational function coefficients. By D-finiteness of f, the quotient $R_n/\operatorname{ann}(f)$ is a finite-dimensional $\mathbb{K}(t, x)$-vector space, and a basis V can be chosen. These algorithms then make a second ansatz for the certificate G:

$$G = \sum_{v \in V} g_v v \tag{1.29}$$

with indeterminate coefficients g_v in $\mathbb{K}(t, x)$.

Substituting (1.28) and (1.29) into Eq. (1.25) and expressing the result in the basis V produces a system of coupled differential equations in the coefficients c_i, with an inhomogeneous part involving the g_v. This system is then uncoupled, and rational solutions are sought. If a nontrivial rational solution is found, it yields a creative telescoping relation; otherwise, the order ℓ of the initial ansatz is increased, and the procedure is repeated.

1.2.2.3 Algorithms based on Ansatz and linear algebra

A third family of algorithm constructs, similarly to the previous method, an Ansatz for both the telescoper and the certificate, and then attempts to find a solution by pure linear algebra. A first algorithm was proposed for multi-sums of binomial functions by Wegschaider [141] in his Master thesis.

This idea was taken up again years later by Koutschan, who sought to avoid the computational cost of certain steps in Chyzak's algorithm, particularly the uncoupling phase and the repeated search for rational solutions, together with its lack of generalization for multivariate integration and summation (that is without calling a univariate algorithm multiple times). Unlike Wegschaider, Koutschan found that allowing denominators in the Ansatz for the certificate actually led to a faster algorithm.

Koutschan's algorithm relies on guessing the shape of a common denominator in the certificate by examining the leading terms of a Gröbner basis with polynomial coefficients of the D-finite annihilator of the summand or integrand. Following Kauers' presentation [82, Algorithm 5.48], the algorithm can be stated in the differential setting (the discrete case and generalizations to multi-sums/multi-integrals being analogous) as follows. Let s be an integer and let p be a polynomial such that for any $i+j \leq s$, $p\partial_x^i \partial_t^j$ can be decomposed in V with polynomial coefficients. The algorithm introduces an Ansatz for both the telescoper and the certificate of the form

$$\sum_{i=0}^{s} c_i(t)\partial_t^i = \partial_x \left(1/p \sum_{i=0}^{s+\deg_x(p)} \sum_{v \in V} d_{i,v}(t) x^i v \right) \tag{1.30}$$

where $c_i(t)$ and $d_{v,i}(t)$ are unknown polynomials. It checks for the existence of a nonzero solution of this form by expressing everything in the basis V and solving a linear system. If no solution is found, s is increased and the process is repeated.

Koutschan also proposed several heuristic optimizations, based on experimentation, to speed up the search. For instance, he proposed to test whether a nonzero solution exists for a given Ansatz by evaluating t at a random value and working modulo a large prime. If such a solution exists after specialization, then it should also exist in the general case. Once an Ansatz is confirmed to yield a nonzero solution, the same technique can be applied again to reduce the degrees of the numerators and of the factors of the denominator before performing the final computation. Koutschan's implementation of his algorithm is currently among the most efficient and widely used software packages for creative telescoping.

1.2.2.4 Reduction-based approaches

Recently, a new family of algorithms based on reductions was introduced to overcome two drawbacks of previous methods. First, they require computing the certificate G, even though it is not needed in some situations. As observed in [23], the certificate can be much larger than the telescoper, making its computation a serious bottleneck. Second, methods based on Ansätze or on solving linear functional equations lack incrementality. Indeed, they attempt to find a telescoper of a fixed order, and if unsuccessful, must restart the search from scratch with a higher order.

In the differential setting, given a function $f(x,t)$, reduction-based algorithms proceed by computing iteratively decompositions of the form

$$\partial_t^i \cdot f = h_i + \partial_x \cdot g_i \tag{1.31}$$

where g_i and h_i belong to $R_2 \cdot f$ and h_i satisfies some minimality condition. A telescoper is then obtained by finding a $\mathbb{K}(t)$-linear relation among the h_i's.

The decomposition in Eq. (1.31) is computed by means of a *reduction map*, which is an effective $\mathbb{K}(t)$-linear map $[.] : M \to M$, where $M = R_2 \cdot f$, satisfying the following properties:

1. $h - [h] \in \partial_x M$ (reduction)
2. $[h] = 0$ iff $h \in \partial_x M$ (normal form)

The reduction [.] provides a way to defines normal forms in the $\mathbb{K}(t)$-vector space $M/\boldsymbol{\partial} M$ and, most importantly, enable their computation. The second property of [.] is sometimes relaxed to the weaker condition that $[\partial_x M]$ is a finite-dimensional $\mathbb{K}(t)$-vector space, as it is sufficient for finding a (not-necessarily minimal) telescoper. With the reduction [.], the decomposition in Eq. (1.31) follows by taking $h_i = [\partial_t^i \cdot f]$ and choosing g_i such that $\partial_x \cdot g_i = \partial_t^i \cdot f - [\partial_t^i \cdot f]$. Note that if only the telescoper is required, the functions g_i need not be computed.

The reduction-based approach was first introduced by Bostan, Chen, Chyzak and Li [18] for the integration of bivariate rational functions. Their algorithm relied on the well-known Hermite reduction [74]. This idea was later extended to the integration of multivariate rational functions [23, 86], by using the Griffiths-Dwork reduction [71, 60, 61] instead of Hermite reduction. The Hermite reduction was independently extended in many directions to permit the univariate integration of hyperexponential functions [19], hypergeometric-hyperexponential functions [22], algebraic functions [40], Fuchsian functions [43], D-finite functions [136, 36] and ∂-finite functions [20]. In parallel, this approach has been adapted to summation problems. It has been applied to the univariate summation of hypergeometric functions [39] via a modification of the Abramov-Petkovšek reduction [1], of P-recursive sequences [37] (i.e. ∂-finite involving only shifts) and of general ∂-finite functions[135].

1.3 Contributions and content

The present thesis contains three main contributions, each described in a dedicated chapter. Chapters 2 and 3 are based on research articles, while Chapter 4 is unpublished.

An algorithm for the univariate summation of D-finite functions

Chapter 2 presents a joint work with Bruno Salvy, started as part of my Master thesis and subsequently published as a research article [29]. It introduces a new reduction-based creative telescoping algorithm for the univariate summation of D-finite functions. The approach adapts the integration algorithm by Bostan, Chyzak, Lairez, Salvy [20] to the summation setting and refines van der Hoeven's algorithm for summing ∂-finite functions.

The reduction underlying the algorithm can be summarized in the bivariate case with two shifts as follows. Suppose the summand $f(n,k)$ is a cyclic vector for the shift operator S_k, meaning that every term $S_n^j \cdot f$ rewrites as $M_j \cdot f$, where M_j is a linear operator involving only S_k. For any rational function u and linear operator M in S_k, repeated summation by parts yields Lagrange's identity:

$$(M \cdot f)u - (M^* \cdot u)f = (S_k - 1) \cdot P_M(f,u), \tag{1.32}$$

where M^* denotes the adjoint of M and P_M is linear in backward shifts of u and forward shifts of f. Specializing this identity to $u = 1$ and $M = M_j$ leads to the decomposition:

$$S_n^j \cdot f = (M^* \cdot 1)f + (S_k - 1) \cdot P_{M_j}(f,1). \tag{1.33}$$

The final step of the reduction consists in reducing the rational function $M^* \cdot 1$ modulo the image of $S_k - 1$, thereby obtaining a decomposition analogous to Eq. (1.31). Let L be the minimal operator in $\mathbb{K}(n,k)\langle S_k\rangle$ annihilating f. One can show that for any rational function $R \in \mathbb{K}(n,k)$, the term Rf belongs to $(S_n - 1)\mathbb{K}(n,k)\langle S_n, S_k\rangle \cdot f$ if and only if R is in $L^*(\mathbb{K}(n,k))$. Therefore, the problem reduces to finding a minimal representative R_j such that

$$M^*(1) = R_j \mod L^*(\mathbb{K}(n,k)). \tag{1.34}$$

Our main contribution is the description of a reduction procedure to solve this problem. The procedure proceeds in two successive steps: first a pole reduction, then a polynomial reduction.

Our approach differs from van der Hoeven's in this final reduction step. His reduction modulo $L^*(\mathbb{K}(n,k))$ is constrained enough to ensure the termination of the creative telescoping algorithm, but may fail to produce the minimal telescoper. In his pole reduction, he focuses on reducing the order of the poles first before reducing the dispersion of the rational function, that is the largest integer distance between two poles. By contrast, our reduction first minimizes the dispersion and then reduces the pole order. This change ensures that we obtain the minimal telescoper in all cases. Noting that many applications require information on the poles of the certificates and their evaluation at specific values of k, we propose a compact representation of the certificate in the form of a directed acyclic graph. This structure enables its efficient evaluation and the computation of a multiple of its denominator. We provide an implementation of our algorithm in the CreativeTelescoping package [30]. The algorithm performs well in general, and seems to outperform previous methods when the telescoper has high order.

An algorithm for the multivariate integration of holonomic functions

Chapter 3 is based on [28] and results from joint a work with Frédéric Chyzak and Pierre Lairez. It presents a reduction-based creative telescoping algorithm for the multivariate integration of holonomic functions, the first of its kind in the holonomic setting. Our method prioritizes computational speed over minimality of the resulting telescoper, thereby avoiding the costly computation of the b-function (to be defined in Chapter 4). It also shares similarities with the Griffiths-Dwork method for integration multivariate rational functions [23, 86].

Let $f(t, x_1, \dots, x_n)$ be a holonomic function. Our algorithm computes the left hand-side of a relation of the form

$$a_\ell(t)\frac{\partial^\ell f}{\partial t^\ell} + \cdots + a_1(t)\frac{\partial f}{\partial t} + a_0(t)f = \frac{\partial g_1}{\partial x_1} + \cdots + \frac{\partial g_n}{\partial x_n}, \tag{1.35}$$

where $a_0(t), \dots a_\ell(t)$ are rational function coefficients and $g_1, \dots, g_n$ are functions in the module

$$M = \mathbb{K}(t)[x_1, \dots, x_n]\langle \partial_t, \partial_1, \dots, \partial_n \rangle \cdot \ f. \tag{1.36}$$

We prove that the module M is finitely generated and even holonomic as a $W_{\mathbf{x}}(t)$-module, where $W_{\mathbf{x}}(t)$ denotes the nth Weyl algebra in $x_1, \dots, x_n$ over $\mathbb{K}(t)$. This result makes it possible to represent the function f by the $W_{\mathbf{x}}(t)$-module M and a derivation map $\partial_t : M \mapsto M$ which corresponds to differentiation with respect to t and satisfies the Leibniz rule

$$\partial_t \cdot (am) = \tfrac{\partial a}{\partial t} m + a(\partial_t \cdot m) \tag{1.37}$$

for any $a \in \mathbb{K}(t)$ and $m \in M$.

Since the right-hand side of Eq. (1.35) corresponds to an element of the right module $\boldsymbol{\partial} M = \partial_1 M + \cdots + \partial_n M$, the problem reduces to finding a $\mathbb{K}(t)$-linear relation among the $\partial_t^i(f)$'s in the quotient $M/\boldsymbol{\partial} M$. By holonomy of M, this quotient is a finite-dimensional vector space over $\mathbb{K}(t)$, ensuring that such a relation exists for sufficiently large ℓ. We introduce a family of increasingly stronger reductions for computing normal forms in $M/\boldsymbol{\partial} M$, with the property that, for every holonomic function f, there exists a reduction that yields the minimal telescoper. In practice, our algorithm finds a reduction that guarantees the termination but not necessarily the minimality. However, the user may choose in the input to employ stronger reductions in the hope of obtaining a telescoper of smaller order, at the expense of increased computation time.

The algorithm is implemented in my Julia package MultivariateCreativeTelescoping.jl [27]. As a proof of concept, we successfully applied our algorithm to a challenging problem in combinatorics: computing a recurrence for the number of 8-regular graphs, a case that had previously remained out of reach.

Weyl closure

Creative telescoping algorithms for integrating holonomic functions rely on prior knowledge of a holonomic annihilator of the integrand. This applies in particular to the algorithm presented in Chapter 3. As explained in Section 1.1.3, such an annihilator can be obtained from a D-finite annihilator by computing the Weyl closure (Eq. (1.16)) of a certain ideal. In Chapter 4, I propose an algorithm for computing a holonomic approximation of the Weyl closure that is asymptotically maximal. Given an ideal I of W_n such that $W_n/\operatorname{Cl}(I)$ is holonomic, a holonomic approximation of $\operatorname{Cl}(I)$ is an ideal $J \subset \operatorname{Cl}(I)$ such that W_n/J is holonomic. The algorithm is called asymptotically maximal as it constructs iteratively a sequence of ideals

$$J_1 \subset J_2 \subset \cdots \subset \operatorname{Cl}(I) \tag{1.38}$$

with the property that there exists $\ell \in \mathbb{N}$ such that $J_\ell = \operatorname{Cl}(I)$, continuing until a prescribed termination criterion is met. Possible termination criteria include: stopping at the first ideal J_i for which W_n/J_i is holonomic, stopping at the first J_i such that $J_i = J_{i+1}$ and W_n/J_i is holonomic, or stopping at the first J_{i+r} such that W_n/J_i is holonomic for some prescribed $r \in \mathbb{N}$. At present, however, I do not know any method to guarantee that the output coincides with $\operatorname{Cl}(I)$ without explicitly computing a b-function.

While Tsai's algorithm computes $\operatorname{Cl}(I)$ by localizing a suitable module at a polynomial p that vanishes on the singular locus of I, this new algorithm proceeds by computing successive saturations with respect to the same polynomial p. The saturation step is done by a non-commutative generalization of Rabinowitsch's trick. More precisely, the algebra W_n is extended with a new non-commutative variable T thought of as $1/p$. This forms an infinite dimensional W_n-module $W_n[T]$, in which one can compute Gröbner bases of finitely generated modules. In particular, this makes it possible to find operators of the form $Tpg \in W_n[T]I$ for which we deduce new operators g in $\operatorname{Cl}(I)$.

This work comes with a preliminary implementation in the package [27], along with preliminary benchmarks. Although further optimization is required, the first results are promising.

2 An algorithm for the univariate summation of D-finite functions

This chapter is based on joint work with the advisor of my Master thesis, Bruno Salvy, which resulted in the publication of an article [29]. Its content closely follows that of the published version, with a shortened introduction to avoid redundancies. Throughout the chapter, I use the pronoun "we" to reflect the collaborative nature of the work.

2.1 Introduction

We present a creative telescoping algorithm for computing telescopers of definite sums of D-finite functions. Our approach is an adaptation of a previous algorithm designed for integration of D-finite function [20], and its extension to summation and q-difference equations [135]. Our contribution departs from [135] in several ways. We concentrate solely on the case of summation and give a simple self-contained presentation of the corresponding algorithm; our algorithm returns telescopers of minimal order[1]; we make the choice to avoid algebraic extensions when possible; we present a Maple implementation that performs well in practice. Note that while in terms of complexity, minimal operators cannot be computed in polynomial time in general, in practice this does not seem to be an obstacle.

In order to compute a definite sum of $F(n, x_1, \ldots, x_m)$ with respect to n, where each x_i is a variable with respect to which one can apply an Ore operator ∂_i (for this work either, differentiation or shift), creative telescoping algorithms construct identities of the form

$$\sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}}(x_1, \ldots, x_m)\boldsymbol{\partial}^{\boldsymbol{\alpha}}(F) = G(n+1, x_1, \ldots, x_m) - G(n, x_1, \ldots, x_m). \tag{2.1}$$

Here, the sum is over a finite number of multi-indices $\boldsymbol{\alpha}$ and we use the multi-exponent notation $\boldsymbol{\partial}^{\boldsymbol{\alpha}} = \partial_1^{\alpha_1} \cdots \partial_m^{\alpha_m}$. The equation Eq. (2.1) can be summed to yield an inhomogeneous linear functional equation satisfied by the sum of F.

Recent reduction-based algorithms use a variant of Hermite reduction to compute an additive decomposition of each monomial in the form

$$\boldsymbol{\partial}^{\boldsymbol{\alpha}}(F) = R_{\boldsymbol{\alpha}}(n, x_1, \ldots, x_m)F + \Delta_n(G_{\boldsymbol{\alpha}}), \tag{2.2}$$

where $R_{\boldsymbol{\alpha}}$ is a rational function with a certain minimality property and where Δ_n is forward difference operator defined by $\Delta_n(h(n)) = h(n+1) - h(n)$. A telescoper is

[1] Remark 5.6 in [135] seems to allude to a way of doing this, but the relevant space E may contain rational functions that are not in the image of L. For example take $L = 1/z + 1/(z-1)\sigma^{-1}$, $\alpha = 0$, and $A = \{\alpha\}$, then one can check that $1/z \notin \mathrm{Im}(L)$ but $1/z \in E$.

found by looking for a linear dependency between these rational functions for a family of monomials $\boldsymbol{\partial}^{\boldsymbol{\alpha}}$. The computation of the rational function $R_{\boldsymbol{\alpha}}$ by Hermite reduction works by getting rid of multiple poles and isolating a polynomial part.

The chapter is structured as follows. Section 2.2 presents a detailed example illustrating the main ideas of the algorithm. Section 2.3 provides the necessary mathematical background and describes the prototypical creative telescoping algorithm based on reductions. It uses a reduction procedure, called canonical form, that reduces a rational function modulo the image of a difference operator. The construction of this canonical map is one of our main contribution and is presented in Section 2.4. Section 2.5 explains how to compute the certificate associated with the minimal telescoper in a compact form using a directed acyclic graph. This representation enables efficient evaluation of the certificate and the computation of a finite set containing its integer poles, two key ingredients for summing a creative telescoping equation. Lastly, Section 2.6 compares our algorithm with existing implementations on a variety of examples.

2.2 Example

The multiplication theorem for Bessel functions of the first kind J_ν states that [97, p. 10.23.1]

$$J_\nu(\lambda z) = \lambda^\nu \sum_{n=0}^{\infty} \frac{(-1)^n(\lambda^2-1)^n(z/2)^n}{n!} J_{\nu+n}(z). \tag{2.3}$$

This can be proved automatically by showing that the left-hand side and the right-hand side satisfy the same set of mixed differential-difference equations with sufficiently many identical initial conditions.

We write F for the summand in Eq. (2.3). It is a function of the four variables ν, n, z, λ. Basic properties of the Bessel function give the following four equations:

$$(\lambda^2-1)zS_\nu(F) + 2(n+1)S_n(F) = 0, \tag{2.4}$$

$$(\lambda^2-1)\partial_\lambda(F) - 2n\lambda F = 0, \tag{2.5}$$

$$(1-\lambda^2)z\partial_z(F) + 2(n+1)S_n(F) + (\lambda^2-1)(2n+\nu)F = 0, \tag{2.6}$$

$$4(n+1)(n+2)S_n^2(F) + 4(\lambda^2-1)(n+1)(n+\nu+1)S_n(F) + z^2(\lambda^2-1)^2F = 0, \tag{2.7}$$

where S_ν denotes the shift with respect to ν: $S_\nu : G(\nu) \mapsto G(\nu+1)$ and similarly for S_n, while ∂_z and ∂_λ denote partial derivatives. These equations show that any shift or derivative $S_\nu^a\partial_\lambda^b\partial_z^c S_n^d F$ of F with nonnegative integers a, b, c, d rewrites as a $\mathbb{Q}(\nu, \lambda, z, n)$-linear combination of F and $S_n(F)$. In particular, this implies that F is D-finite with respect to these variables. The aim of creative telescoping is to find a similar set of equations, in the variables ν, λ, z only, for the sum in Eq. (2.3).

Let Δ_n be the *difference operator* $\Delta_n = S_n - 1$. Any product $\phi(n)S_nF$ with $\phi \in \mathbb{Q}(\nu, \lambda, z, n)$ can be rewritten $\phi(n-1)F + \Delta_n(\phi(n-1)F)$, i.e., as the sum of a rational function times F plus a difference, that would telescope under summation. Consequently, any $S_\nu^a\partial_\lambda^b\partial_z^c S_n^d F$ with nonnegative integers a, b, c, d rewrites in the form

$$\boldsymbol{\partial}^{\boldsymbol{\alpha}}(F) = R_{\boldsymbol{\alpha}}F + \Delta_n(G_{\boldsymbol{\alpha}}), \tag{2.8}$$

with $R_{\boldsymbol{\alpha}}$ a rational function in $\mathbb{Q}(\nu, \lambda, z, n)$. This reduction as a sum of a rational function plus a difference is a general phenomenon (see Section 2.3.2). Reduction-based creative telescoping works by reducing this rational function further by pulling out parts that can be incorporated into the difference $\Delta_n(G_{\boldsymbol{\alpha}})$. Denote by L the recurrence operator such that Eq. (2.7) is $LF = 0$. The *adjoint* of L (see Proposition 13) is

$$L^* = 4(n-1)nS_n^{-2} + 4(\lambda^2-1)n(n+\nu)S_n^{-1} + z^2(\lambda^2-1)^2$$

where $S_n^{-1} : g(n) \mapsto g(n-1)$. Proposition 14 shows that a rational function R is of the form $\Delta_n(M(F))$ for a recurrence operator $M(S_n)$ if and only if R is in the image $L^*(\mathbb{Q}(\nu, \lambda, z, n))$. This is the basis for the computation of relations of the form (2.8) where $R_{\boldsymbol{\alpha}}$ is now a *reduced* rational function (in a sense made precise in Definition 16). The next step is to look for linear combinations of these rational functions that yield telescopers.

The starting point is the monomial 1, which decomposes as

$$1 \cdot F = 1 \cdot F + \Delta_n(0). \tag{2.9}$$

Using Eq. (2.5), the monomial ∂_λ rewrites

$$\partial_\lambda(F) = \frac{2n\lambda}{\lambda^2-1}F + \Delta_n(0) \tag{2.10}$$

and the rational function is reduced. Taking the derivative of this equation and using Eq. (2.5) again gives a similar equation for $\partial_\lambda^2(F)$:

$$\begin{aligned}\partial_\lambda^2(F) &= -\frac{2n(\lambda^2+1)}{(\lambda^2-1)^2}F + \frac{2n\lambda}{\lambda^2-1}\partial_\lambda(F) + \Delta_n(0),\\ &= -\frac{2n(\lambda^2+1)}{(\lambda^2-1)^2}F + \frac{(2n\lambda)^2}{(\lambda^2-1)^2}F + \Delta_n(0).\end{aligned}$$

This time, a reduction is possible. Indeed, Proposition 14 implies that $L^*(1)F$ is a difference $\Delta_n(A_n)$ (where A_n can be computed explicitly). Since

$$L^*(1) = 4\lambda^2n^2 + 4((\lambda^2-1)\nu - 1)n + z^2(\lambda^2-1)^2,$$

we can eliminate the term in n^2 in the expression of $\partial_\lambda^2(F)$ to get

$$\partial_\lambda^2(F) = -\frac{2n(2\nu+1)}{(\lambda^2-1)}F - z^2F + \Delta_n\left(\frac{A_n}{(\lambda^2-1)^2}\right). \tag{2.11}$$

A simple linear combination of Eqs. (2.9) to (2.11) then eliminates the term in n, showing that F satisfies the equation

$$\lambda\partial_\lambda^2F + (2\nu+1)\partial_\lambda F + \lambda z^2F = \Delta_n\left(\frac{-\lambda A_n}{(\lambda^2-1)^2}\right).$$

The left-hand side is a telescoper. The right-hand side is a certificate. It can be written more explicitly as

$$\frac{-\lambda A_n}{(\lambda^2-1)^2} = -\frac{4(n\lambda^2+\lambda^2\nu-\nu-1)n\lambda}{(\lambda^2-1)^2}F - \frac{4\lambda(n+1)n}{(\lambda^2-1)^2}S_n(F).$$

In general, summation and telescoping of the certificate requires verification. Here, we first observe that the certificate does not have integer poles and thus is well-defined at all points over which it is summed. Next, the certificate evaluates to zero at $n = 0$. Finally, it tends to zero when n tends to infinity, as $J_{\nu+n}(z)$ decreases fast as $n \to \infty$ [97, p. 10.19.1].

In summary, we have obtained that the sum S in the right-hand side of Eq. (2.3) satisfies

$$\lambda\partial_\lambda^2(S) + (2\nu+1)\partial_\lambda(S) + \lambda z^2 S = 0.$$

Proceeding similarly with Eqs. (2.4) and (2.6), one gets the equations

$$z\lambda S_\nu(S) + \partial_\lambda(S) = 0, \quad z\partial_z(S) - \lambda\partial_\lambda(S) - \nu S = 0.$$

Substituting $T = J_\nu(\lambda z)/\lambda^\nu$ in these equations and using basic equations for J_ν shows that it is a solution of this system too. The proof of the multiplication theorem is concluded by checking the equality of the initial conditions for T and for the sum on the right-hand side of Eq. (2.3). As ν is associated to the shift, we need to check initial conditions for any ν satisfying $0 \le \mathrm{Re}(\nu) < 1$. Indeed, both terms of the identity equal $J_\nu(1)$ at $z = 1, \lambda = 1$, and $\nu \in [0, 1)$ and both their derivatives with respect to λ equal $-J_{\nu+1}(1)$, which proves the identity.

2.3 Background

In this section, we recall the basic framework for creative telescoping using reduction. Most of this section is identical to the differential case [20, Sec. 4], except for the existence and computation of the cyclic vector and the use of the recurrence variant of Lagrange's identity [9]. More gentle introductions to Ore algebras, creative telescoping and their applications can be found in the references [48, 45].

2.3.1 Telescoping ideal

Ore algebras

Let $\mathbf{k}$ be a field of characteristic 0, $x_0, \dots, x_m$ be variables used to form the fields of rational functions $\mathbb{K} = \mathbf{k}(x_1, \dots, x_m)$ and $\hat{\mathbb{K}} = \mathbb{K}(x_0)$. The *Ore algebra* $\mathbb{A} = \hat{\mathbb{K}}\langle\partial_0, \dots, \partial_m\rangle$ is a polynomial ring over $\hat{\mathbb{K}}$, with $\partial_i\partial_j = \partial_j\partial_i$, and a commutation between the ∂_is and the elements of $\hat{\mathbb{K}}$ ruled by relations

$$\partial_i R = \sigma_i(R)\partial_i + \delta_i(R), \qquad R \in \hat{\mathbb{K}}, \tag{2.12}$$

with σ_i a ring homomorphism of $\hat{\mathbb{K}}$ and δ_i a σ_i-derivation, which means that $\delta_i(ab) = \sigma_i(a)\delta_i(b) + \delta_i(a)b$ for all a, b in $\hat{\mathbb{K}}$ [31, 45]. The typical cases are when ∂_i is the differentiation d/dx_i (then σ_i is the identity and $\delta_i = d/dx_i$) and the shift operator $x_i \mapsto x_i + 1$ (then $\sigma_i(a) = a|_{x_i \leftarrow x_i+1}$ and $\delta_i = 0$).

Annihilating and D-finite ideals

For a given function f in a left $\mathbb{A}$-module, the annihilating ideal of f is the left ideal $\operatorname{ann} f \subseteq \mathbb{A}$ of elements of $\mathbb{A}$ that annihilate f. A left ideal $\mathcal{I}$ of $\mathbb{A}$ is *D-finite* when the quotient $\mathbb{A}/\mathcal{I}$ is a finite-dimensional $\hat{\mathbb{K}}$-vector space. A function is called D-finite when its annihilating ideal is D-finite.

Telescoping ideal

As we focus here on summation, from now on, when we use n and S_n, they stand for x_0 and the corresponding shift operator $\partial_0 : x_0 \mapsto x_0 + 1$.

The *telescoping ideal* $\mathcal{T}_\mathcal{I}$ of the left ideal $\mathcal{I} \subset \mathbb{A}$ with respect to n is

$$\mathcal{T}_\mathcal{I} = (\mathcal{I} + \Delta_n(\mathbb{A})) \cap \mathbb{K}\langle \partial_1, \ldots, \partial_m \rangle, \qquad \text{where} \quad \Delta_n = S_n - 1.$$

In other words, if $\mathcal{I} = \operatorname{ann} F$, the telescoping ideal $\mathcal{T}_\mathcal{I}$ is the set of operators $T \in \mathbb{K}\langle \partial_1, \ldots, \partial_m \rangle$ such that there exists $G \in \mathbb{A}$ such that $T + \Delta_n G \in \mathcal{I}$, or equivalently, such that Eq. (2.1) holds (with $t = n$).

2.3.2 Cyclic vector and Lagrange identity

Cyclic vector

Let $\mathcal{I}$ be a D-finite ideal of $\mathbb{A}$ and let r be the dimension of the $\hat{\mathbb{K}}$-vector space $\mathbb{B} := \mathbb{A}/\mathcal{I}$. An element $\gamma \in \mathbb{B}$ is called *cyclic with respect to* ∂_0 if $\{\gamma, \ldots, \partial_0^{r-1}\gamma\}$ is a basis of $\mathbb{B}$. In the differential case ($\partial_0 = d/dx_0$), such a vector always exists and can be computed efficiently when $\mathcal{I}$ is D-finite [44]. In the shift case ($\partial_0 : x_0 \mapsto x_0 + 1$), even for a D-finite ideal $\mathcal{I}$, it is not the case that there always exists a cyclic vector: in general, $\mathbb{B}$ decomposes as the sum of a vector space where ∂_0 is nilpotent and a part where it is cyclic [76]. However, we have the following.

Proposition 11. *[73, Thm. B2] With the notation above, in the case when ∂_0 is the shift operator $x_0 \mapsto x_0 + 1$, let $E = (e_1, \ldots, e_r)^\intercal$ be a basis of the vector space $\mathbb{B} = \mathbb{A}/\mathcal{I}$ and $A_0 \in \hat{\mathbb{K}}^{r\times r}$ be defined by $\partial_0 E = A_0 E$. If A_0 is invertible, then there exists a cyclic vector with respect to ∂_0 of the form $v = a_1 e_1 + \cdots + a_r e_r$ with polynomial coefficients $a_i \in \mathbb{Z}[x_0]$ of degree at most $r - 1$, and coefficients all in $\{0, \ldots, r\}$.*

Sufficient conditions for the matrix A_0 to be invertible are that $\mathcal{I} = \operatorname{ann} f$ with f in a $\hat{\mathbb{K}}[\partial_0, \partial_0^{-1}]$-module [73] or that $\mathcal{I}$ be a reflexive ideal [135]. In practice, this condition on A_0 can be checked from the input and appears to be always satisfied in the examples we have tried. From this proposition, the computation of a cyclic vector follows the same lines as that of the differential case [44]. Most often, $e_1 = 1$ is a cyclic vector, which simplifies the rest of the computation.

Example 12. In the example of Section 2.2, the operator 1 is cyclic with respect to S_n: by Eqs. (2.4) to (2.7) a basis of $\mathbb{B}$ is $\{1, S_n\}$.

Lagrange's identity

For our purpose, the shift version of Lagrange's identity can be viewed as giving an explicit form of the result of the left Euclidean division by the difference operator Δ_n, when applied to a left multiplication by a rational function.

Proposition 13. *[9] Let $u \in \hat{\mathbb{K}}$ and let $L = \sum_{i=0}^{r} a_i S_n^i$ be an operator of order r with a_i in $\hat{\mathbb{K}}$. The adjoint operator L^* of L is defined as $L^* = \sum_{i=0}^{r} a_i(n-i) S_n^{-i}$ and it satisfies*

$$uL - L^*(u) = \Delta_n P_L(u) \tag{2.13}$$

where

$$P_L(u(n)) = \sum_{i=0}^{r-1} \left(\sum_{j=i+1}^{r} a_j(n+i-j)u(n+i-j) \right) S_n^i. \tag{2.14}$$

Note that the term $L^*(u)$ denotes the evaluation of the operator L^* at the rational function u, rather than the product of L^* by u.

Let γ be a cyclic vector. Then any element of $\mathbb{B}$ is of the form $A\gamma$ with $A \in \hat{\mathbb{K}}\langle S_n \rangle$. Applying Lagrange's identity with $u = 1$, $L = A$ and multiplying on the right by γ shows that this is a rational multiple of γ up to a difference:

$$A\gamma = A^*(1)\gamma + \Delta_n P_A(1)\gamma. \tag{2.15}$$

As in the differential case, all computations in $\mathbb{B}$ then reduce to $\hat{\mathbb{K}}$-linear operations on single rational functions, rather than vectors of them, by the following analogue of [20, Prop. 4.2].

Proposition 14. *With the notation above, let γ be a cyclic vector of $\mathbb{B} = \mathbb{A}/\mathcal{I}$ and for all $i = 0, \dots, m$, let $B_i \in \hat{\mathbb{K}}\langle S_n \rangle$ be such that $\partial_i \gamma = B_i \gamma$. Then for all $R \in \hat{\mathbb{K}}$,*

$$\partial_i R\gamma = \varphi_i(R)\gamma + \Delta_n Q_i(R)\gamma, \tag{2.16}$$

$$\text{with} \quad \begin{cases} \varphi_i(R) = B_i^*(R(x_i+1)), & Q_i(R) = P_{B_i}(R(x_i+1)) \quad \text{if } \partial_i : x_i \mapsto x_i + 1; \\ \varphi_i(R) = B_i^*(R) + \frac{d}{dx_i}(R), & Q_i(R) = P_{B_i}(R(x_i)) \quad \text{if } \partial_i = d/dx_i. \end{cases}$$

Proof. Multiplying Eq. (2.12) by γ on the right and using the definition of B_i gives

$$\partial_i R\gamma = \sigma_i(R) B_i \gamma + \delta_i(R)\gamma.$$

The conclusion follows from Lagrange's identity (2.13) applied with $L = B_i$ and $u = \sigma_i(R)$. □

2.3.3 Canonical form

Proposition 14 shows how, given a cyclic vector γ, all elements of $\mathbb{B}$ can be reduced to the product of γ by a rational function, up to a difference in $\Delta_n\mathbb{B}$. The starting point of the reduction-based creative telescoping is that one can actually identify those multiples that belong to $\Delta_n\mathbb{B}$.

Proposition 15. *With the same hypotheses as in Proposition 14, let L be a minimal-order operator in $\hat{\mathbb{K}}\langle S_n\rangle$ annihilating γ, ie, the product $L\gamma$ is 0 in $\mathbb{A}/\mathcal{I}$, L has order r and no operator of smaller order has that property. Then for all $R \in \hat{\mathbb{K}}$, $R\gamma \in \Delta_n(\mathbb{B}) \iff R \in L^*(\hat{\mathbb{K}})$.*

Proof. First, if $R = L^*(R')$ with $R' \in \hat{\mathbb{K}}$, Lagrange's identity (2.13) with $u = R'$ and $L\gamma = 0$ implies that $R\gamma = L^*(R')\gamma = \Delta_n G\gamma$ for $G = -P_L(R')$. Conversely, if $R\gamma \in \Delta_n(\mathbb{B})$, there exists $M \in \hat{\mathbb{K}}\langle S_n\rangle$ such that $R\gamma = \Delta_n M\gamma$. The operator $\Delta_n M - R$ annihilates γ. By minimality of L, there exists $N \in \hat{\mathbb{K}}\langle S_n\rangle$ such that $\Delta_n M - R = NL$. Taking the adjoint and evaluating at 1 gives $R = M^*\Delta_n^* - L^*N^*$ and finally $R = -L^*(N^*(1))$. □

This proposition motivates the following.

Definition 16. [20] A *canonical form* associated to L^* is a $\mathbb{K}$-linear map $[\cdot] : \mathbb{K}(n) \to \mathbb{K}(n)$ such that for all $R \in \mathbb{K}(n)$, $[L^*(R)] = 0$ and $R - [R] \in L^*(\mathbb{K}(n))$. A rational function $R \in \mathbb{K}(n)$ is called *reduced* when $[R] = R$.

The computation of canonical forms is the object of Section 2.4.

2.3.4 Creative telescoping algorithm via canonical forms

With the notation above, the creative telescoping algorithm from [20] applies verbatim. It is given in Algorithm 1. Its principle is to iterate on every monomial of the form $\partial_1^{\alpha_1} \dots \partial_m^{\alpha_m}$ by increasing order for some monomial order, e.g., the grevlex order, and to compute the reduced rational functions $R_{\boldsymbol{\alpha}}$ such that

$$\boldsymbol{\partial}^{\boldsymbol{\alpha}} = R_{\boldsymbol{\alpha}}\gamma + \Delta_n G_{\boldsymbol{\alpha}} \bmod \mathcal{I}. \tag{2.17}$$

The rational function $R_{\boldsymbol{\alpha}}$ is obtained by

$$\begin{cases} R_{0,\dots,0} = [A_1^*(1)], \\ R_{\boldsymbol{\alpha}} = [\varphi_i(R_{\boldsymbol{\beta}})] \quad \text{if } \boldsymbol{\partial}^{\boldsymbol{\alpha}} = \partial_i \boldsymbol{\partial}^{\boldsymbol{\beta}}, \end{cases} \tag{2.18}$$

where A_1^* is the adjoint of the operator A_1 verifying $1 = A_1(\gamma)$. When a monomial $\boldsymbol{\alpha}$ is dealt with, two situations are possible. The corresponding $R_{\boldsymbol{\alpha}}$ can be a linear combination of the previous $R_{\boldsymbol{\beta}}$. In that case, that linear combination makes the corresponding linear combination of $\boldsymbol{\partial}^{\boldsymbol{\alpha}}$ and the $\boldsymbol{\partial}^{\boldsymbol{\beta}}$ a newly discovered element of the telescoping ideal $\mathcal{T}_{\mathcal{I}}$ and then it is not necessary to visit the multiples of this monomial. Otherwise, $\boldsymbol{\partial}^{\boldsymbol{\alpha}}$ is

Algorithm 1 Creative telescoping algorithm

Input Generators of a D-finite ideal $\mathcal{I}$
Output A basis of the telescoping ideal $\mathcal{T}_{\mathcal{I}}$ of $\mathcal{I}$

```
γ ← a cyclic vector of 𝔹 := 𝔸/ℐ with respect to S_n          {See Section 2.3.2}
L ← a minimal order operator in S_n annihilating γ           {See Section 2.3.3}
φ_1, ..., φ_n the maps described in Proposition 14
Initialize CanonicalForm                                      {See Section 2.4}
R_0 ← CanonicalForm(A_1^*(1), L^*)                            {See Section 2.3.4}
ℒ ← [1]                              {List of monomials in ∂_1, ..., ∂_m to visit}
𝒢 ← {}                                                        {Gröbner basis}
𝒬 ← {}                                                {Basis of the quotient}
ℛ ← {}                                            {Set of reducible monomials}
while ℒ ≠ ∅ do
  Remove the first element ∂^α of ℒ
  if ∂^α is not a multiple of an element of ℛ then
    if ∂^α ≠ 1 then
      Take i such that ∂^α/∂_i ∈ 𝒬; R_α ← CanonicalForm(φ_j(R_{∂^α/∂_i}), L^*)
    if there exists a 𝕂-linear relation between R_α and {R_β | R_β ∈ 𝒬} then
      (λ_β)_{R_β∈𝒬} ← coefficients of the relation R_α = Σ_{R_β∈𝒬} λ_β R_β
      Add ∂^α − Σ_{R_β∈𝒬} λ_β ∂^β to 𝒢; Add ∂^α to ℛ
    else
      Add ∂^α to 𝒬
      for j = 1 to m do Append the monomial ∂_j ∂^α to ℒ
return 𝒢
```

free from the previous ones and thus a new generator of $\mathbb{B}$ has been found. The algorithm terminates when there are no more monomials to visit.

The only difference with the differential case lies in the definition of the canonical form $[\cdot]$ associated to the adjoint L^* of the minimal-order operator $L \in \mathbb{K}(n)\langle S_n\rangle$ annihilating γ. By Propositions 14 and 15 and the definition of a canonical form, Eq. (2.17) is satisfied and the following equivalence holds

$$\lambda_{\boldsymbol{\alpha}_1} R_{\boldsymbol{\alpha}_1} + \cdots + \lambda_{\boldsymbol{\alpha}_s} R_{\boldsymbol{\alpha}_s} = 0 \quad \text{iff} \quad \lambda_{\boldsymbol{\alpha}_1} \boldsymbol{\partial}^{\boldsymbol{\alpha}_1} + \cdots + \lambda_{\boldsymbol{\alpha}_s} \boldsymbol{\partial}^{\boldsymbol{\alpha}_s} \in \mathcal{T}_{\mathcal{I}}. \tag{2.19}$$

for any $s \in \mathbb{N}$ and $\lambda_{\boldsymbol{\alpha}_1}, \ldots, \lambda_{\boldsymbol{\alpha}_s} \in \mathbb{K}$. The following result follows, with the same proof as in [20].

Theorem 17. *Given as input the generators of a D-finite ideal $\mathcal{I}$ and a cyclic vector γ for S_n, Algorithm 1 terminates if and only if $\mathcal{T}_{\mathcal{I}}$ is D-finite. Then, it outputs a Gröbner basis of $\mathcal{T}_{\mathcal{I}}$ for the grevlex order.*

As in the differential situation [20], one can modify the algorithm to compute all elements of $\mathcal{T}_{\mathcal{I}}$ up to a given bound on their degree, or to return as soon as one telescoper is found, thus allowing to recover a generating family of a sub-ideal of $\mathcal{T}_F$.

A non-zero telescoper is not guaranteed to exist in general. An existence criterion was obtained for P-recursive sequences by Du [58], and in general a telescoper is known to exist if the summand is holonomic [82, section 5.2]

2.4 Generalized Abramov-Petkovšek decomposition

The main contribution of this article is an algorithm for the computation of canonical forms as in Definition 16 for the operator

$$L^* = \sum_{i=0}^{r} p_i(n) S_n^{-i}$$

with polynomial coefficients $p_i \in \mathbb{K}[n]$. The modified Abramov-Petkovšek decomposition [39] is a special case of this reduction when L has order 1 and once the shell [39] has been removed [20, Sec. 3.5.3].

The starting point is a decomposition of any rational function $R \in \mathbb{K}(n)$ in the form

$$R(n) = P_\infty(n) + \sum_{i,h} \frac{c_{i,h}(n)}{Q_i(n+h)^{\ell_{i,h}}}, \tag{2.20}$$

with $\ell_{i,h} \in \mathbb{N}^*$, polynomials P_∞ in $\mathbb{K}[n]$ and $Q_i, c_{i,h}$ in $\mathbb{K}[n] \setminus \{0\}$ such that $\deg c_{i,h} < \ell_{i,h} \deg Q_i$ and $\gcd(Q_i(n+k), Q_j(n)) = 1$ for all $k \in \mathbb{Z}$ when $i \neq j$. This is discussed in Section 2.4.1.

The vector spaces

$$\mathbb{K}_{Q_i}(n) \stackrel{\text{def}}{=} \mathrm{Vect}_{\mathbb{K}} \left(\frac{n^\ell}{Q_i(n+h)^j} \mid h \in \mathbb{Z}, j \in \mathbb{N}^*, \ell < j \deg(Q_i) \right)$$

are in direct sum for distinct Q_i and are left invariant by L^* modulo $\mathbb{K}[n]$. This allows the reduction algorithm to operate in each of the $\mathbb{K}_{Q_i}(n)$ independently. This is described in Sections 2.4.2 and 2.4.3, before the reduction of the remaining polynomial part in Sections 2.4.4 and 2.4.5.

Notation 18. For two integers a, b with $a \leq b$, we write $[\![a; b]\!]$ for the set $\{a, a+1, \dots, b\}$.

2.4.1 Decomposition of rational functions

Recall that a polynomial Q is *square-free* when it does not have multiple nontrivial factors. It is *shift-free* when $\gcd(Q(n), Q(n+k)) = 1$ for all $k \in \mathbb{Z}^*$.

A *shiftless decomposition* of a polynomial Q is a factorization of the form

$$Q = \prod_{i=1}^{v} \prod_{j=1}^{n_i} Q_i(n + h_{i,j})^{e_{i,j}},$$

with $e_{i,j} \in \mathbb{N}^*, h_{i,j} \in \mathbb{Z}$, and $Q_i \in \mathbb{K}[n]$ are such that each Q_i is square-free and $\gcd(Q_i(n+k), Q_j(n)) = 1$ for all i, j and all $k \in \mathbb{Z}$ unless $i = j$ and $k = 0$. Such a factorization can be computed using only gcds, resultants and integer root finding [68].

Algorithm 2 Weak reduction of poles $[\cdot]_Q$

Input R, Q

Output a reduced form of R

while there exists $j_m < 0$ (minimal) in J from Eq. (2.21) **do**

$R \leftarrow R - L^*\left(\frac{A(n)}{Q(n-j_m)^{s_{j_m}+\mathrm{ord}_{j_m}(p_0)}}\right)$ with A and s_{j_m} as in Eq. (2.22)

while there exists $j_M \geq r$ (maximal) in J from Eq. (2.21) **do**

$R \leftarrow R - L^*\left(\frac{A'(n+r)}{Q(n-j_M+r)^{s_{j_M}+\mathrm{ord}_{j_M}(p_r)}}\right)$ with A' and s_{j_M} as in Eq. (2.23)

return R

Note that shiftless decompositions are not unique in general. One can be refined when a Q_i is not irreducible, by splitting this factor further. In particular, the linear factors of the Q_i can be isolated and dealt with more easily.

A polynomial Q is *refined with respect to a polynomial* P when it is such that for each $h \in \mathbb{Z}$, there exists $\ell \in \mathbb{N}$ such that $\gcd(P, Q(n+h)^{\ell+1}) = Q(n+h)^{\ell}$. A shiftless decomposition is called *refined with respect to* P when each Q_i is. This refinement can be computed using gcds only and will be used with $P = p_0$ and $P = p_r$, the extreme coefficients of L^*.

Example 19. Let $Q = (n-1)(n-1/2)$, $P_1 = (n-1)(n-1/2)$ and $P_2 = (n-1)$. Then the shiftless decomposition of Q defined by $v = n_1 = e_{1,1} = 1$ and $Q_1 = Q$ is refined with respect to P_1 but not P_2 as for any $\ell \in \mathbb{N}, \gcd(P_2, Q^{\ell+1}) = n-1$, which is not a power of Q. However the shiftless decomposition defined by $v = 2$, $n_1 = n_2 = e_{1,1} = e_{2,1} = 1, Q_1 = n-1$ and $Q_2 = n-1/2$ is refined with respect to both P_1 and P_2.

From a shiftless decomposition, the partial fraction decomposition of Eq. (2.20) is then obtained by standard algorithms [140, p. 5.11].

2.4.2 Weak reduction of the polar part

Lemma 20. *Let $Q \in \mathbb{K}[n]$ be square-free, shift-free and refined with respect to the coefficients p_0 and p_r of L^*. Given a rational function $R \in \mathbb{K}_Q(n)$, Algorithm 2 computes a rational function $[R]_Q \in \mathbb{K}_Q(n)$ with all its poles at zeros of $Q(n-j)$ such that $j \in [\![0; r-1]\!]$ and $R - [R]_Q = P + L^*(T)$ for some $P \in \mathbb{K}[n]$ and $T \in \mathbb{K}_Q(n)$. The algorithm is $\mathbb{K}$-linear, i.e. the function implemented by the algorithm is $\mathbb{K}$-linear.*

Proof. Assume that R decomposes as

$$R = \sum_{j \in J} \frac{\lambda_j(n)}{Q(n-j)^{s_j}} \qquad \text{with} \quad \lambda_j \neq 0,\ \deg(\lambda_j(n)) < s_j \deg(Q). \tag{2.21}$$

Let $j_m = \min(J)$ and $\mathrm{ord}_j(p_0)$ be the largest integer ℓ such that $Q(n-j)^{\ell} \mid p_0$. Then,

$$Q(n-j_m)^{s_{j_m}} L^*\left(\frac{1}{Q(n-j_m)^{s_{j_m}+\mathrm{ord}_{j_m}(p_0)}}\right) = \tilde{p}_0(n) \bmod Q(n-j_m)^{s_{j_m}},$$

where $\tilde{p}_0(n)$ is the remainder in the Euclidean division of $p_0/Q(n-j_m)^{\mathrm{ord}_{j_m}(p_0)}$ by $Q(n-j_m)^{s_{j_m}}$. The poles of the rational function $L^*\left(Q(n-j_m)^{-s_{j_m}-\mathrm{ord}_{j_m}(p_0)}\right)$ are at zeros of $Q(n-j)$ with $j \in (j_m + [\![0,r]\!])$.

Since Q is refined with respect to p_0, the polynomial $\tilde{p}_0(n)$ is relatively prime with $Q(n-j_m)$. Thus, an extended gcd computation yields two polynomials A and B such that

$$\lambda_{j_m}(n) = A(n)\tilde{p}_0(n) + B(n)Q(n-j_m)^{s_{j_m}}. \tag{2.22}$$

Then

$$\frac{A(n)\tilde{p}_0(n)}{Q(n-j_m)^{s_{j_m}}} = \frac{\lambda_{j_m}(n)}{Q(n-j_m)^{s_{j_m}}} - B(n),$$

so that

$$R - L^*\left(\frac{A(n)}{Q(n-j_m)^{s_{j_m}+\mathrm{ord}_{j_m}(p_0)}}\right)$$

is equivalent to R modulo $L^*(\mathbb{K}_Q(n))$ and with all its poles at zeros of $Q(n-j)$ with $j \in J \setminus \{j_m\} \cup (j_m + [\![1,r]\!])$. This operation can be repeated a finite number of times until all poles are at zeros of $Q(n-j)$ with $j \geq 0$.

Similarly, let $j_M = \max(J)$. Then

$$Q(n-j_M)^{s_{j_M}} L^*\left(\frac{1}{Q(n-j_M+r)^{s_{j_M}+\mathrm{ord}_{j_M}(p_r)}}\right) = \tilde{p}_r(n) \bmod Q(n-j_M)^{s_{j_M}},$$

where $\tilde{p}_r(n)$ is the remainder in the Euclidean division of $p_r/Q(n-j_M)^{\mathrm{ord}_{j_M}(p_r)}$ by $Q(n-j_M)^{s_{j_M}}$. The poles of the rational function $L^*\left(Q(n-j_M+r)^{-s_{j_M}-\mathrm{ord}_{j_M}(p_r)}\right)$ are at zeros of $Q(n-j)$ with $j \in (j_M - [\![0,r]\!])$. Again, since Q is refined with respect to p_r, the polynomial $\tilde{p}_r$ is relatively prime with $Q(n-j_M)$. Thus there exist two polynomials A' and B' such that

$$\lambda_{j_M}(n) = A'(n)\tilde{p}_r(n) + B'(n)Q(n-j_M)^{s_{j_M}} \tag{2.23}$$

so that

$$R - L^*\left(\frac{A'(n+r)}{Q(n-j_M+r)^{s_{j_M}+\mathrm{ord}_{j_M}(p_r)}}\right)$$

is equivalent to R modulo $L^*(\mathbb{K}_Q(n))$ and with all its poles at zeros of $Q(n-j)$ with $j \in J \setminus \{j_M\} \cup (j_M - [\![1,r]\!])$. This operation can be repeated a finite number of times until all poles are at zeros of $Q(n-j)$ with $j \in [\![0, r-1]\!]$.

Each step being $\mathbb{K}$-linear, so is the algorithm. □

Example 21. Let

$$R = 8nx + \frac{2x^3}{n+3} + \frac{4x(x^2-x-8)}{n+2} + \frac{(x+4)(2x^2-3x-4)n + 2x^3 + x^2 - 16x - 16}{(n+1)^2} - \frac{4(x-1)x^2}{n} + \frac{5nx^2 - 9x^2}{(n-1)^2}$$

and

$$L^* = x^2(n-2)S_n^{-3} - n(4n^2 - x^2 - 4n)S_n^{-2} + n(4n^2 - x^2 + 4n)S_n^{-1} - x^2(n+2).$$

The poles of R are at $\{1, 0, -1, -2, -3\}$. We take $Q = n+1$ and follow the steps of the algorithm.

The pole at -3 is easy: from

$$L^*\left(\frac{1}{n+3}\right) = 4n + \frac{x^2}{n+3} + \frac{2x^2-16}{n+2} + \frac{-x^2+8}{n+1} - \frac{2x^2}{n}$$

and the coefficient $2x^3$ of $(n+3)^{-1}$ in R, the algorithm performs the subtraction

$$R \leftarrow R - 2xL^*\left(\frac{1}{n+3}\right) = -\frac{4x^2}{n+2} - \frac{(5x^2-16)n + x^2 - 16}{(n+1)^2} + \frac{4x^2}{n} + \frac{5nx^2 - 9x^2}{(n-1)^2}.$$

Next, the pole -2 is a simple root of the constant coefficient of L^*, leading to the computation of

$$L^*\left(\frac{1}{(n+2)^2}\right) = -\frac{x^2}{n+2} - \frac{(x^2-4)n - 4}{(n+1)^2} + \frac{x^2}{n} + \frac{x^2 n - 2x^2}{(n-1)^2}$$

so that the pole is removed by

$$R \leftarrow R - 4L^*\left(\frac{1}{(n+2)^2}\right) = -\frac{x^2}{(n+1)} + \frac{x^2}{(n-1)}. \tag{2.24}$$

R now has all its poles in $\{-1, 0, 1\}$ and the weak reduction is completed.

2.4.3 Strong reduction of the polar part

By Lemma 20, the weak reduction produces rational functions all whose poles differ from those of Q by an integer in $[\![0, r-1]\!]$. The next step of the reduction is to subtract rational functions in $L^*(\mathbb{K}_Q(n))$ that have this property.

It turns out to be possible to focus on a finite-dimensional subspace of $L^*(\mathbb{K}_Q(n))$ thanks to the following.

Lemma 22. *If $j < 0$, $s > \operatorname{ord}_j(p_0)$ and $\ell < s\deg(Q)$ or if $j \geq 0$, $s > \operatorname{ord}_{j+r}(p_r)$ and $\ell < s\deg(Q)$ then*

$$\left[L^*\left(\frac{n^\ell}{Q(n-j)^s}\right)\right]_Q = 0.$$

Proof. Let j, s, ℓ be three integers that satisfy the first assumption. Then $L^*(n^\ell/Q(n-j)^s)$ has a denominator that is divisible by $Q(n-j)$ with $j < 0$ by assumption. No smaller k is such that $Q(n-k)$ divides the denominator. Thus the first pass through the first loop of the weak reduction subtracts $L^*(n^\ell/Q(n-j)^s)$ to itself and reduces it to zero. When the second assumption is satisfied, then $L^*(n^\ell/Q(n-j)^s)$ has a denominator that is divisible by $Q(n-(j+r))$ with $j+r \geq r$ by assumption. No larger k is such that $Q(n-k)$ divides the denominator. Thus again, the second loop reduces that fraction to 0. ☐

Algorithm 3 Weak reduction of polynomials $[\cdot]_\infty$

Input P and (σ,p) from Eq. (2.25)
Output a reduced form of P
$a \leftarrow 0$
while $\deg(P) \geq \sigma$ **do**
  if $\deg(P) - \sigma$ is a root of p **then** $a \leftarrow a + \operatorname{lt}(P)$; $P \leftarrow P - \operatorname{lt}(P)$
  else $P \leftarrow P - \frac{\operatorname{lc}(P)}{p(\deg(P)-\sigma)} L^*(n^{\deg(P)-\sigma})$
return $a + P$

Corollary 23. *Let* $I_0 := \{j \in \mathbb{Z}_{<0} \mid \gcd(p_0(n), Q(n-j)) \neq 1\}$ *and* $I_r := \{j \in \mathbb{Z}_{\geq 0} \mid \gcd(p_r(n), Q(n-(j+r))) \neq 1\}$*. The* $\mathbb{K}$*-vector space* $[L^*(\mathbb{K}_Q(n))]_Q$ *is generated by the fractions*

$$\left[L^*\left(\frac{n^\ell}{Q(n-j)^{s_j}}\right)\right]_Q, \qquad \textit{with} \quad \begin{cases} j \in I_0 \textit{ and } 1 \leq s_j \leq \operatorname{ord}_j(p_0) \textit{ and } 0 \leq \ell < s_j \deg Q \\ \textit{or} \\ j \in I_r \textit{ and } 1 \leq s_j \leq \operatorname{ord}_{j+r}(p_r) \textit{ and } 0 \leq \ell < s_j \deg Q. \end{cases}$$

Corollary 23 provides a generating family of the finite-dimensional $\mathbb{K}$-vector space $[L^*(\mathbb{K}_Q(n))]_Q$. These rational functions can be written in the basis $(n^i/Q(n-j)^k)_{i,k\in\mathbb{N},j\in\mathbb{Z}}$ and one can then compute an echelon basis of this finite-dimensional space. This precomputation step corresponds to the computation of the B_{Q_i}'s in Algorithm 4. The *strong reduction* of a rational function $R \in \mathbb{K}_Q$ then consists in reducing $[R]_Q$ with this echelon basis. By this process, we obtain the following.

Proposition 24. *Strong reduction reduces every rational function* $R \in L^*(\mathbb{K}_Q(n))$ *to a polynomial in* $\mathbb{K}[n]$*.*

Example 25. With the same notation as in example 21, Corollary 23 shows that the vector space $[L^*(\mathbb{K}_{n+1}(n))]_{n+1}$ is generated by

$$\left[L^*\left(\frac{1}{n+2}\right)\right]_{n+1} = 4n + \frac{x^2}{n+1} - \frac{x^2}{n-1}, \qquad \left[L^*\left(\frac{1}{n+1}\right)\right]_{n+1} = 4n + \frac{x^2}{n-1} - \frac{x^2}{n+1}.$$

Thus the strong reduction of the rational function R from Eq. (2.24) is the polynomial

$$R + L^*((n-2)^{-1})/(2x) = -2n/x,$$

concluding the reduction.

2.4.4 Weak reduction of polynomials

The weak reduction of polynomials is a direct adaptation of the differential case [20]. The indicial polynomial of L^* at infinity is the polynomial $p \in \mathbb{K}[s]$ defined by

$$L^*(n^s) = n^{s+\sigma}(p(s) + O(1/n)), \tag{2.25}$$

with $\sigma \in \mathbb{N}$. The ensuing weak reduction is presented in Algorithm 3.

Example 26. In Examples 21 and 25, the indicial equation at infinity is

$$L^*(n^s) = n^{s+2}(8 + 4s + O(1/n)).$$

The polynomial $-2n/x$ found in Example 25 cannot be reduced further by weak reduction since its degree in n is smaller than 2.

The following properties are proved exactly as those for weak reduction at a pole.

Lemma 27. *Algorithm* $[\cdot]_\infty$ *terminates and is* $\mathbb{K}$*-linear. For all* $P \in \mathbb{K}[n]$*, there exists* $Q \in \mathbb{K}[n]$ *such that* $P - [P] = L^*(Q)$*. If* $s \in \mathbb{N}$ *is not a root of* p*, then* $[L^*(n^s)]_\infty = 0$*.*

2.4.5 Strong reduction of polynomials

The final step is to subtract polynomials in $L^*(\mathbb{K}(n))$. Here again, a finite number of generators can be obtained thanks to the following.

Lemma 28. *Let* $Q_1, \dots, Q_v$ *be the polynomials that occur in a shiftless decomposition of* $p_0 p_r$ *and let* P *be a polynomial in* $L^*(\mathbb{K}(n))$*. Then*

$$P \in E_{pol} \overset{def}{=} L^*(\mathbb{K}[n]) + \sum_{i=1}^{v} [L^*(\mathbb{K}_{Q_i}(n))]_{Q_i} \cap \mathbb{K}[n].$$

Proof. If $R \in \mathbb{K}(n)$ is such that $L^*(R)$ is a polynomial, then the poles of R must be cancelled by the zeros of $p_0 p_r$ or their shifts. It follows that R decomposes as

$$R = R_\infty + \sum_{i=1}^{v} R_i$$

with $R_i \in \mathbb{K}_{Q_i}(n)$ and $R_\infty \in \mathbb{K}[n]$. Each $L^*(R_i)$ has to be a polynomial and thus invariant by $[\cdot]_{Q_i}$. This concludes the proof. □

By Lemma 27, the vector space $[L^*(\mathbb{K}[n])]_\infty$ is generated by

$$\{[L^*(n^s)]_\infty \mid s \in \mathbb{N} \text{ and } p(s) = 0\},$$

where p is the indicial polynomial of L^* at infinity. Generators of each $[L^*(\mathbb{K}_{Q_i}(n))]_{Q_i} \cap \mathbb{K}[n]$ are obtained from the echelon basis used in the strong reduction with respect to Q_i. This gives a finite set of generators for $[E_{\mathrm{pol}}]_\infty$, which is easily transformed into a basis by a row echelon computation. Strong reduction consists in reducing modulo this basis. The following consequence is as in the polar case.

Lemma 29. *The strong reduction of polynomials reduces every polynomial* $P \in L^*(\mathbb{K}(n))$ *to zero.*

Example 30. Continuing Examples 21, 25 and 26, the polynomial $p(s) = 8 + 4s$ has no root in $\mathbb{N}$, therefore $[L^*(\mathbb{K}[n])]_\infty = \{0\}$. A basis of $[L^*(\mathbb{K}_{n+1})]_{n+1} \cap \mathbb{K}[n]$ is $\{n\}$ according to Example 25. Therefore $2n/x$ reduces to 0.

Algorithm 4 PrecomputeBases

Input $Q_1, \dots, Q_v$ polynomials that occur in the shiftless decomposition of $p_0 p_r$
Output The echelon bases $B_{Q_1}, \dots, B_{Q_v}, B_{pol}$

  $B_{pol} \leftarrow \{\}$
  for $i = 1$ **to** v **do**
    $B_{Q_i} \leftarrow \text{Echelon}\left(\left\{ \left[L^*\left(\frac{n^\ell}{Q(n-j)^{s_j}}\right)\right]_{Q_i} \mid \ell, j, s_j \text{ as in Cor.1} \right\}\right)$
    $B_{pol} \leftarrow B_{pol} \cup (B_{Q_i} \cap \mathbb{K}[n])$
  $B_{pol} \leftarrow \text{Echelon}(B_{pol} \cup \{[L^*(n^s)]_\infty \mid s \in \mathbb{N} \text{ root of } p\})$
  return $B_{Q_1}, \dots, B_{Q_v}, B_{pol}$

Algorithm 5 Reduction of rational functions $[\cdot]$

Input R and $B_{Q_1}, \dots, B_{Q_v}, B_{pol}$ computed by Algorithm 4
Output a reduced form of R by $L^*(\mathbb{K}(n))$.

  Decompose R as $P_\infty + \sum_{i=1}^{v} R_i$ with $R_i \in \mathbb{K}_{Q_i}(n)$ as in Eq. (2.20)
  for $i = 1$ **to** v **do** $R_{Q_i} \leftarrow \text{ReduceWithEchelon}([R_i]_{Q_i}, B_{Q_i})$
  $R \leftarrow P_\infty + \sum_{i=1}^{v} R_{Q_i}$
  Write $R = P + \tilde{R}$ with P a polynomial and $\tilde{R}$ a proper fraction
  $P \leftarrow \text{ReduceWithEchelon}([P]_\infty, B_{pol})$
  return $P + \tilde{R}$

2.4.6 Canonical form

Algorithm 4 and Algorithm 5 combine the previous algorithms to produce a canonical form.

Theorem 31. *Algorithm 5 computes a canonical form.*

Proof. Algorithm 5 is linear as every step is linear. By Proposition 24 and Lemma 29, $[L^*(\mathbb{K}(n))]$ reduces to 0 and $[R] - R \in L^*(\mathbb{K}(n))$ as only functions in this image were subtracted to R. □

2.5 Certificates

Reduction-based creative telescoping algorithms allow to find a telescoper without having to compute an associated certificate. This has led to faster algorithms as certificates are known to be larger than telescopers [23]. This approach makes sense in the differential case when it is known in advance that the integral of a certificate over a cycle that avoids singularities is equal to zero. The framework is not as favorable for sums. Indeed, it is necessary to detect whether the certificate has poles in the range of summation and it is often unclear whether the certificate becomes 0 at the boundaries of the summation interval.

It is however possible to compute the certificates *in a compact way* during the execution of our algorithm, with almost no impact on the execution time. The idea is to make the computation and storage of certificates efficient by storing them as directed acyclic graphs (dags) rather than operators with normalized rational function coefficients. These dags have a number of internal nodes of the same order as the number of operations performed when computing the telescoper, so that their computation does not burden the complexity. They can then be evaluated at the endpoints of the range of summation, or expanded in Laurent series there. In particular, they can be used for a probabilistic test of the telescoping identity that has been computed, by checking that it vanishes at random points (our implementation provides a function `testcert` for this.)

2.5.1 Computation and structure of certificates

Equations (2.17) and (2.18) show how one can compute telescopers without computing certificates. We now show how to compute the certificates simultaneously. These are obtained as sums of monomials in the ∂_i multiplied by rational functions. These rational functions have denominators of the form $Q(n-j)^s$, with integers j, s and polynomials Q that occur either in the shiftless decomposition of $p_0 p_r$ or in the denominator of the operator $A_1 \in \hat{\mathbb{K}}\langle S_n \rangle$ such that $1 = A_1\gamma$, or in the denominator of one of the operators B_i from Proposition 14. Those sums are not reduced to a common denominator. They share many common coefficients and denominators that are efficiently compacted into dags by sharing common subexpressions (this is how Maple stores them by default).

The starting point is the cyclic vector $\gamma \in \mathbb{A}/\mathcal{I}$ and the operator $A_1 \in \hat{\mathbb{K}}\langle S_n \rangle$ such that $1 = A_1\gamma$. By Euclidean division by Δ_n on the left,

$$A_1 = R_0 + \Delta_n g_0.$$

In general, the certificate $G_{\boldsymbol{\alpha}}$ in Eq. (2.17) is stored as an (unreduced) element $g_{\boldsymbol{\alpha}}$ of $\mathbb{A}$ such that $G_{\boldsymbol{\alpha}} = g_{\boldsymbol{\alpha}}\gamma \bmod \mathcal{I}$. The computation is incremental. It is initialized with R_0 and g_0 as above, corresponding to $\boldsymbol{\alpha} = \mathbf{0}$. In many cases, the vector 1 is cyclic, so that one can take $\gamma = 1 = A_1 = R_0$ and $g_0 = 0$. Otherwise, the denominators of R_0 and g_0 are shifts of the denominator of A_1.

Next, Eq. (2.18) leads to the computation of the canonical form of the rational function $\varphi_i(R_{\boldsymbol{\beta}})$. This computation is performed via a sequence of reductions which consist of subtractions of elements in $L^*(\mathbb{K}(n))$. Keeping track of these rational functions (without normalizing them) gives the canonical form $R_{\boldsymbol{\alpha}}$ as

$$\varphi_i(R_{\boldsymbol{\beta}}) = R_{\boldsymbol{\alpha}} - L^*(c_{\boldsymbol{\alpha}})$$

for some rational function $c_{\boldsymbol{\alpha}} \in \hat{\mathbb{K}}$. Both $R_{\boldsymbol{\alpha}}$ and $c_{\boldsymbol{\alpha}}$ have denominators that are shifts of those of $R_{\boldsymbol{\beta}}$ or of factors of $p_0 p_r$. In view of Eq. (2.16), it follows that

$$\boldsymbol{\partial}^{\boldsymbol{\alpha}} = \partial_i \boldsymbol{\partial}^{\boldsymbol{\beta}} = R_{\boldsymbol{\alpha}}\gamma + \Delta_n \left(\partial_i G_{\boldsymbol{\beta}} + P_{B_i}(\sigma_i(R_{\boldsymbol{\beta}}))\gamma + P_L(c_{\boldsymbol{\alpha}})\gamma\right) \bmod \mathcal{I}.$$

Thus, the certificate $G_{\boldsymbol{\alpha}}$ is obtained as $g_{\boldsymbol{\alpha}}\gamma$ with

$$g_{\boldsymbol{\alpha}} = \partial_i g_{\boldsymbol{\beta}} + P_{B_i}(\sigma_i(R_{\boldsymbol{\beta}})) + P_L(c_{\boldsymbol{\alpha}}). \tag{2.26}$$

This proves the claim concerning the structure of the certificates and the factors of their denominators. In our implementation, ∂_i is commuted with the coefficients of the certificate $g_{\boldsymbol{\beta}}$ only. If desired, one can further use $\partial_i = B_i \bmod \mathcal{I}$ so as to write $g_{\boldsymbol{\alpha}}$ as an operator in S_n only.

2.5.2 Evaluation of the certificates

The output of Algorithm 1 is a set of elements T of the telescoping ideal, which means that

$$T(F) = g(F)(n+1, x_1, \dots, x_m) - g(F)(n, x_1, \dots, x_m),$$

with g a certificate as described above. Summing over n, the right-hand side telescopes and only the values of the certificate $g(F)$ at the endpoints are needed.

It is possible to prove that these evaluations are zero without any evaluation in two important cases. First, if the summand F has finite support (e.g., binomial sums), then the sum of any certificate over $\mathbb{Z}$ will be zero provided it has no pole in the summation range. The second case is when one can prove that $R(n)\gamma(F), \dots, R(n)S_n^{r-1}\gamma(F)$ tend to zero as n tends to $\pm\infty$ for any rational function $R \in \mathbb{K}(n)$ (as in the introductory example). Then again the sum of any certificate over $\mathbb{Z}$ will be zero provided it has no pole in the summation range.

2.5.3 Integer pole detection

By its very nature, the method of creative telescoping requires the certificate not to have poles in the range of summation, so that telescoping can occur. The structure of the certificates described above does not allow the efficient computation of its denominator exactly. However it is possible to compute a multiple of it by taking the least common multiple of the denominators of every rational function in the representation. This can be done efficiently by performing the computation on the dag representation of the certificates.

From this multiple of the denominator of the certificate, one can compute the set of roots that lie in the summation range; this amounts to computing the roots that differ from the endpoints of the summation range by an integer. If that set is not empty, then one can compute a Laurent series expansion of the certificate at any point to check whether it is a pole or not, again by exploiting the dag representation of the rational function coefficients.

2.5.4 Examples

2.5.4.1 Neumann's Addition Theorem for Bessel functions

On input $S(x) = \sum_{n=1}^{\infty} J_n(x)^2$ where $J_n(x)$ is the Bessel function of the first kind, Algorithm 1 outputs the telescoper ∂_x and a certificate G in dag form that exhibits poles at $n \in \{-1, 0, 1, 2\}$. Our implementation produces the polynomial $n(n+1)(n-2)(n-1)$

containing this information. In this example, the certificate is small enough that it can easily be normalized and one gets its value as

$$-\frac{x}{4(n+1)}S_n^2+\frac{n+1}{x}S_n-\frac{8n^2-x^2+8n}{4x(n+1)},$$

showing that the poles at 1 and 2 vanish in the normalization. Without normalizing the certificate, one can still evaluate the series expansions of the certificate at those points to establish that it has a finite limit there, making the summation legitimate. At $n=1$, the evaluation is found to be

$$-J_0(x)J_1(x)=\frac{1}{2}(J_0(x)^2)'.$$

Thus we have proved

$$\frac{d}{dx}\left(\frac{1}{2}J_0^2(x)+J_1(x)^2+J_2(x)^2+\cdots\right)=0.$$

This shows that the sum is constant and the value is revealed by its value at 0, which follows from $J_k(0)=0$ for $k>0$ and $J_0(0)=1$, so that in the end, we recover the classical identity [97, p. 10.23.3]

$$1=J_0(x)^2+2\sum_{k\geq 1}J_k(x)^2.$$

2.5.4.2 Apéry's Sequence

The classical sum

$$A_n=\sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2, \tag{2.27}$$

used by Apéry in his proof of the irrationality of $\zeta(3)$, has telescoper

$$\mathcal{T}:=(n+2)^3S_n^2-(2n+3)\left(17n^2+51n+39\right)S_n+(n+1)^3. \tag{2.28}$$

The singularities of the certificate obtained by our implementation are at $k\in\{n+1,n+2\}$. Indeed, once normalized, the certificate is found to be

$$\mathcal{C}:=\frac{4(3k-4n-8)k^4}{(k-n-2)^2}S_n-\frac{4k^4(3k+4n+4)}{(k-n-1)^2}.$$

Let $U_{n,k}$ denote the product of binomials in the sum. Summing $\mathcal{T}U_{n,k}$ from $k=0$ to $k=n+2$ gives $\mathcal{T}(A_n)$. If telescoping is legitimate, then the values of the endpoints are the values of $\mathcal{C}U_{n,k}$ at $k=0$ and $k=n+3$, that are both easily checked to be 0. For this to allow to conclude that $\mathcal{T}(A_n)=0$, it is then sufficient to check that $\mathcal{C}U_{n,k}$ is not singular at $k=n+1$ and $k=n+2$, even though $\mathcal{C}$ is. Indeed, a series expansion of the evaluation of $\mathcal{C}U_{n,k}$ at $k=n+1$ and $k=n+2$ is possible for our implementation, and finds that the sequence has a finite limit there, which concludes the proof that the telescoper Eq. (2.28) cancels the sum Eq. (2.27); see also [49] for more on these issues.

2.5.5 A larger example

The computation of telescoper and certificate for the sum

$$\sum_{n=0}^{\infty} \frac{(4n+1)(2n)!}{n!^2 2^{2n}\sqrt{x}} J_{2n+1/2}(x) P_{2n}(u) \tag{2.29}$$

takes less than 15 sec. with our current implementation (see Table 2.1). The telescopers are quite small:

$$(1-u^2)\partial_u + xu\partial_x, \quad (u^3-u)\partial_u^2 + (1+u^2)\partial_u - u^3x^2.$$

In this example, not normalizing the certificates during their computation has a cost. The actual certificates, once reduced by the Gröbner basis of the annihilating ideal of the summand, are not very large. They are easily computed by Koutschan's program. Still, the corresponding dags are large. Nonetheless, it takes less than 1 sec. to compute a multiple of the denominators of the certificates and detect that they do not have integer roots. Evaluating the certificates at $n = 0$ using their dag representation takes less than 2 min. and proves that the telescopers cancel the sum in Eq. (2.29).

2.6 Implementation

This algorithm is implemented in Maple[2]. Table 2.1 gives a comparison of our code with Koutschan's heuristic (HF-FCT) and Chyzak's algorithm (HF-CT)[3]. They are both implemented in Koutschan's HolonomicFunctions package in Mathematica [84]. The column 'redctsum' corresponds to our algorithm.

These programs have been executed on a list of 21 easy examples that were compiled by Koutschan, as well as more difficult ones given in Eqs. (2.30) to (2.36) and (2.39) to (2.41) below. Eq. (2.30) comes from recent identities involving determinants [4], Eqs. (2.31), (2.32), (2.39) and (2.40) have been chosen because they looked natural to experiment with, Eq. (2.34) is a harder example found in Koutschan's list, Eq. (2.33) as well as Eq. (2.36) and its special case Eq. (2.35) come from the classical book of integral and series by Prudnikov *et al.* [104], and finally Eq. (2.41) is an example where Koutschan's heuristic does not stop as it does not guess correctly the form of the ansatz to use [38].

$$\sum_{j=1}^{n} \binom{m+x}{m-i+j} c_{n,j} \quad \text{where } c_{n,j} \text{ satisfies recurrences of order 2 [4, p. 6]} \quad [\text{shift } n, i] \tag{2.30}$$

$$\sum_{n=0}^{\infty} C_n^{(k)}(x) C_n^{(k)}(y) \frac{u^n}{n!} \qquad [\text{shift } k, \text{ diff } x, y, u] \tag{2.31}$$

[2] The implementation is available at https://github.com/HBrochet/CreativeTelescoping.git, together with sessions of examples.

[3] The code was run on a Intel Core i7-1265U with 32 GB of RAM.

	HF-CT	HF-FCT	redctsum
easy examples	6.7s	7s	0.9s
Eq. (2.30)	101s	49s	0.8s
Eq. (2.31)	52s	4s	1.4s
Eq. (2.32)	62s	1.7s	5.7s
Eq. (2.33)	4.9s	1.4s	10.3s
Eq. (2.34)	4.9s	1.4s	13.5s
Eq. (2.35)	1200s	13s	205s
Eq. (2.36)	> 6h	108s	3338s
Eq. (2.37)	976s	9s	123s
Eq. (2.38)	> 1h	19s	278s
Eq. (2.39)	1703s	4.7s	580s
Eq. (2.40)	> 1h	3.2s(*)	> 1h
Eq. (2.41)	> 1h	> 1h	0.4s

Table 2.1: Columns HF-CT and HF-FCT report the runtimes of Chyzak's and Koutschan's algorithms in the `HolonomicFunctions` package, while column redctsum shows the runtimes of our Maple implementation. An entry marked with (*) indicates that we could not verify whether the telescopers returned by HF-FCT were minimal.

$$\sum_{n=0}^{\infty} J_n(x)C_n^{(k)}(y)\frac{u^n}{n!} \qquad \text{[shift } k\text{, diff } x, y, u] \tag{2.32}$$

$$\sum_{k=0}^{\infty}(-1)^k(4k+1)J_{2k+1/2}(w)P_{2k}(z) \qquad \text{[diff } w, z] \tag{2.33}$$

$$\sum_{n=0}^{\infty}\frac{(4n+1)(2n)!}{n!^2 2^{2n}\sqrt{x}}J_{2n+1/2}(x)P_{2n}(u) \qquad \text{[diff } x, u] \tag{2.34}$$

$$\sum_{k=0}^{\infty}\frac{(b+3/2)_k}{(3/2)_k(b+1)_k}P_k^{(1/2,b)}(x)P_k^{(1/2,b)}(y) \qquad \text{[shift } b\text{, diff } x, y] \tag{2.35}$$

$$\sum_{k}\frac{(a+b+1)_k}{(a+1)_k(b+1)_k}P_k^{(a,b)}(x)P_k^{(a,b)}(y) \qquad \text{[shift } a, b\text{, diff } x, y] \tag{2.36}$$

$$\sum_{k}\frac{(b+3/2)_k k! t^k}{(3/2)_k(b+1)_k}P_k^{(1/2,b)}(x)P_k^{(1/2,b)}(y) \qquad \text{[shift } b\text{, diff } t, x, y] \tag{2.37}$$

$$\sum_{k}\frac{(a+b+1)_k k! t^k}{(a+1)_k(b+1)_k}P_k^{(a,b)}(x)P_k^{(a,b)}(y) \qquad \text{[shift } a, b\text{, diff } t, x, y] \tag{2.38}$$

$$\sum_{n=0}^{\infty}P_n(x)P_n(y)P_n(1/2) \qquad \text{[diff } x, y] \tag{2.39}$$

	HF-CT	HF-FCT	redctsum
S_6	11s	64s	0.4s
S_7	32s	331s	0.6s
S_8	106s	1044s	1.0s
S_9	325s	3341s	2.5s
S_{10}	1035s	>1h	5.7s

Table 2.2: Timings on the family S_r from Eq. (2.42). Columns HF-CT and HF-FCT report the runtimes of Chyzak's and Koutschan's algorithms in the `HolonomicFunctions` package, while column redctsum shows the runtimes of our Maple implementation. An entry marked with (*) indicates that we could not verify whether the telescopers returned by HF-FCT were minimal.

$$\sum_{n=0}^{\infty} P_n(x)P_n(y)P_n(z) \qquad [\text{diff } x, y, z] \tag{2.40}$$

$$\sum_{y} \frac{4x+2}{(45x+5y+10z+47)(45x+5y+10z+2)(63x-5y+2z+58)(63x-5y+2z-5)} \qquad [\text{shift } x, z] \tag{2.41}$$

The family (S_r) is defined by [69]

$$S_r = \sum_{k=0}^{n} \frac{(-1)^k (rn-(r-1)k)!(r!)^k}{(n-k)!^r k!}. \tag{2.42}$$

For any r, our algorithm produces a minimal telescoper of order r and degree $r(r-1)/2$. The timings are reported in Table 2.2. It is unclear why the heuristic HF-FCT does not perform well on this family.

On most of these examples, the main part of the time of the computation is spent in the reductions in the call to CanonicalForm in Algorithm 1. For the two similar sums of Eqs. (2.33) and (2.34), almost half of the time is spent in Algorithm 4 performing the reductions needed to compute the bases for the strong reduction. This step is crucial to ensure that the minimal order elements in the telescoping ideal are found.

There are cases, like Eq. (2.36) and the family S_r, where the intermediate rational functions $R_{\boldsymbol{\alpha}}$ in Eq. (2.17) become much larger than the telescopers found after linear algebra on them. In such situations, the direct, non-incremental approach taken by HF-CT and HF-FCT can be more efficient, by avoiding an unnecessarily large basis of rational functions.

3 An algorithm for the multivariate integration of holonomic functions

The research presented in this chapter was conducted in collaboration with my PhD advisors, Frédéric Chyzak and Pierre Lairez, and resulted in the submission of an article [28]. The chapter closely follows the submitted version, with a shortened introduction to avoid redundancies. Throughout the chapter, I use the pronoun "we" to reflect the collaborative nature of the work.

3.1 Introduction

We present a creative telescoping algorithm for computing telescopers of multiple integrals of holonomic functions depending on a single parameter. The method naturally generalizes to integrals with multiple parameters. Our algorithm uses a reduction based-approach, a novelty in the holonomic setting. Let $f(t, x_1, \dots, x_n)$ be a holonomic function and let

$$I(t) = \oint f(t, x_1, \dots, x_n) \mathrm{d}x_1 \dots \mathrm{d}x_n \tag{3.1}$$

be an integral we aim to compute. In the multivariate case, the method of creative telescoping seeks to compute an equation of the form

$$a_\ell(t) \frac{\partial^\ell f}{\partial t^\ell} + \dots + a_1(t) \frac{\partial f}{\partial t} + a_0(t) f = \frac{\partial g_1}{\partial x_1} + \dots + \frac{\partial g_n}{\partial x_n}, \tag{3.2}$$

where $a_\ell, \dots, a_0$ are rational functions in $\mathbb{K}(t)$ and $g_1, \dots g_n$ are functions in $W_{t,\mathbf{x}} \cdot f$, with $W_{t,\mathbf{x}}$ denoting the Weyl algebra in t and $\mathbf{x}$.

We prove that the module

$$M = \mathbb{K}(t)[x_1, \dots, x_n]\langle \partial_t, \partial_1, \dots, \partial_n \rangle \cdot \ f \tag{3.3}$$

is finitely generated and even holonomic as a $W_{\mathbf{x}}(t)$-module, where $W_{\mathbf{x}}(t)$ denotes the nth Weyl algebra in $x_1, \dots, x_n$ over $\mathbb{K}(t)$. This result makes it possible to represent the function f by the $W_{\mathbf{x}}(t)$-module M and a derivation map $\partial_t : M \mapsto M$ which corresponds to differentiation with respect to t and satisfies the Leibniz rule

$$\partial_t \cdot (am) = \tfrac{\partial a}{\partial t} m + a(\partial_t \cdot m) \tag{3.4}$$

for any $a \in \mathbb{K}(t)$ and $m \in M$. Noting that the right-hand side of Eq. (3.2) corresponds to an element of the right module $\boldsymbol{\partial} M = \partial_1 M + \dots + \partial_n M$, the problem of creative telescoping reduces to computing $\mathbb{K}(t)$-linear relations among elements of M, in this specific case the $\partial_t^i(f)$'s, in the quotient $M/\boldsymbol{\partial}M$. The holonomy of M implies that the quotient $M/\boldsymbol{\partial}M$ is a finite dimensional vector-space over $\mathbb{K}(t)$ [16, Theorem 6.1 of Chapter 1, combined with the example that precedes it, for $p = 2n$], which guarantees the existence of such a relation for sufficiently large ℓ.

The chapter is structured as follows. In Section 3.3, we propose a new algorithm for computing relations in $M/\boldsymbol{\partial}M$. In some aspects, this generalizes the Griffiths-Dwork reduction method for homogeneous rational functions [71, 23, 86] to the holonomic setting (see Section 3.3.4). This yields a new algorithm for multivariate creative telescoping that is presented in Section 3.4. In Section 3.5 we present a Julia [15] implementation of our algorithm. Although the new algorithm does not yet match the performance of the best D-finite-based implementations, it improves the state of the art in the holonomic setting. As an application, we present in Section 3.6 the computation of a differential equation for the generating series of 8-regular graphs, a case for which D-finite approaches are theoretically not suited, and which was also previously unattainable by dedicated methods.

3.2 Computing with Weyl algebras

3.2.1 Weyl algebras

Let $\mathbb{K}$ be a field of characteristic zero, typically $\mathbb{Q}$ or $\mathbb{Q}(t)$. Let $W_{\mathbf{x}}$ denote the *nth Weyl algebra* $\mathbb{K}[\mathbf{x}]\langle\boldsymbol{\partial}_{\mathbf{x}}\rangle$ with generators $\mathbf{x} = (x_1 \dots, x_n)$ and $\boldsymbol{\partial}_{\mathbf{x}} = (\partial_1, \dots, \partial_n)$, and relations $\partial_i x_i = x_i\partial_i + 1$, $x_i x_j = x_j x_i$, $\partial_i\partial_j = \partial_j\partial_i$ and $x_i\partial_j = \partial_j x_i$ whenever $i \neq j$. We refer to [54, Chapters 1–10] for a complete introduction to these algebras covering most needs of the present article, or to [17, Chapter 5] for a denser alternative. We often need to highlight one variable with a specific role, in which case we use the name t for the distinguished variable. Correspondingly, we will write $W_{t,\mathbf{x}}$ for the $(1+n)$th Weyl algebra, and we will write W_t for the special case $n = 0$. We also define $W_{t,\mathbf{x}}(t)$ as the algebra $\mathbb{K}(t) \otimes_{\mathbb{K}[t]} W_{t,\mathbf{x}}$ where the variable t is rational and the variables $\mathbf{x}$ are polynomial. For non-zero $r \in \mathbb{N}$, we also consider Cartesian powers of these algebras, $W_{\mathbf{x}}^r$, $W_{\mathbf{x}}(t)^r$, etc., which we view as modules over $W_{\mathbf{x}}$ or $W_{\mathbf{x}}(t)$, as relevant. Each element of the module $W_{\mathbf{x}}^r$ decomposes uniquely in the basis of $\mathbb{K}$-vector space

$$M_{\mathbf{x},r} = \{\mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}} e_i \mid \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^n,\ i \in \{1, \dots, r\}\}.$$

Given an element $p = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta},i} a_{\boldsymbol{\alpha},\boldsymbol{\beta},i}\mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}^{\boldsymbol{\beta}} e_i$ of $W_{t,\mathbf{x}}^r$, we define the *degree* of p as

$$\deg(p) = \max\{|\boldsymbol{\alpha}| + |\boldsymbol{\beta}| \mid \exists i \in \{1, \dots, r\},\ a_{\boldsymbol{\alpha},\boldsymbol{\beta},i} \neq 0\},$$

where $|\boldsymbol{\alpha}|$ and $|\boldsymbol{\beta}|$ denote the sums $\alpha_1 + \cdots + \alpha_n$ and $\beta_1 + \cdots + \beta_n$. Definitions for the algebra $W_{\mathbf{x}}$ mimic the case $r = 1$, just not considering any e_i, and definitions for the module $W_{t,\mathbf{x}}^r$ are just a special notation when n is replaced with $n + 1$: a vector basis of $W_{\mathbf{x}}$ is

$$M_{\mathbf{x}} = \{\mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}} \mid \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^n\},$$

and given an element $p = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}} a_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\alpha}} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}}$ of $W_{\mathbf{x}}$, its degree is

$$\deg(p) = \max\{|\boldsymbol{\alpha}| + |\boldsymbol{\beta}| \mid a_{\boldsymbol{\alpha},\boldsymbol{\beta}} \neq 0\};$$

a vector basis of $W_{t,\mathbf{x}}^r$ is

$$M_{t,\mathbf{x},r} = \{t^\alpha \mathbf{x}^{\boldsymbol{\beta}} \partial_t^\gamma \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\delta}} e_i \mid \alpha, \gamma \in \mathbb{N},\ \boldsymbol{\beta}, \boldsymbol{\delta} \in \mathbb{N}^n,\ i \in \{1, \ldots, r\}\},$$

and given an element $p = \sum_{\alpha,\boldsymbol{\beta},\gamma,\boldsymbol{\delta},i} a_{\alpha,\boldsymbol{\beta},\gamma,\boldsymbol{\delta},i} t^\alpha \mathbf{x}^{\boldsymbol{\beta}} \partial_t^\gamma \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\delta}} e_i$ of $W_{t,\mathbf{x}}^r$, its degree is

$$\deg(p) = \max\{\alpha + |\boldsymbol{\beta}| + \gamma + |\boldsymbol{\delta}| \mid \exists i \in \{1, \ldots, r\},\ a_{\alpha,\boldsymbol{\beta},\gamma,\boldsymbol{\delta},i} \neq 0\}.$$

3.2.2 Holonomic modules

Let S be a submodule of $W_{\mathbf{x}}^r$. We recall the classical definition of a *holonomic* $W_{\mathbf{x}}$-*module* by means of the *dimension* of the quotient module $M = W_{\mathbf{x}}^r / S$. We point out that any $W_{\mathbf{x}}$-module of finite type is isomorphic to a module of this form. The *Bernstein filtration* [12] of the algebra $W_{\mathbf{x}}$ is the sequence of $\mathbb{K}$-vector spaces $\mathcal{F}_m$ defined by

$$\mathcal{F}_m = \{P \in W_{\mathbf{x}} \mid \deg(P) \leq m\}.$$

A filtration of the module M that is adapted to $(\mathcal{F}_m)_{m \geq 0}$ is the sequence of $\mathbb{K}$-linear subspaces $\Phi_m \subseteq M$ defined by

$$\Phi_m = \text{image in } M \text{ of } \{P \cdot e_i \mid P \in \mathcal{F}_m,\ 1 \leq i \leq r\}.$$

Those filtrations are compatible with the algebra and module structures: both $\mathcal{F}_m \mathcal{F}_{m'} \subseteq \mathcal{F}_{m+m'}$ and $\mathcal{F}_m \Phi_{m'} \subseteq \Phi_{m+m'}$ hold. There exists a polynomial $p \in \mathbb{K}[m]$ called the *Hilbert polynomial* of M that satisfies $\dim_{\mathbb{K}}(\Phi_k) = p(k)$ for any sufficiently large k. The *dimension* of the module M is the degree d of the polynomial p. The integer d clearly lies between 0 and $2n$. It was proved by Bernstein that if M is non-zero, then d is larger than or equal to n [54, Theorem 9.4.2]. When the dimension of M is exactly n or when M is the zero module, we say that the module M is *holonomic*. (Here, we follow the tradition in [17, Chapter 5] and in [54] to consider the zero module as holonomic. By way of comparison, [16] speaks of a module “in the Bernstein class” to refer to a non-zero holonomic module.)

3.2.3 Gröbner bases in Weyl algebras and their modules

Despite their non-commutative nature, by a *monomial* we will mean an element of the vector bases $M_{\mathbf{x},r}$, $M_{\mathbf{x}}$, and $M_{t,\mathbf{x},r}$. A *monomial order* $\preccurlyeq$ on $W_{\mathbf{x}}^r$ is a well-ordering on $M_{\mathbf{x},r}$ that satisfies for any $i, j \in \{1, \ldots, r\}$ and any exponents $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2 \in \mathbb{N}^n$

$$\mathbf{x}^{\boldsymbol{\alpha}_1} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}_1} e_i \preccurlyeq \mathbf{x}^{\boldsymbol{\alpha}_2} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}_2} e_j \implies \mathbf{x}^{\boldsymbol{\alpha}_1+\boldsymbol{\alpha}} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}_1+\boldsymbol{\beta}} e_i \preccurlyeq \mathbf{x}^{\boldsymbol{\alpha}_2+\boldsymbol{\alpha}} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}_2+\boldsymbol{\beta}} e_j.$$

Given an operator $P \in W_{\mathbf{x}}^r$, we define its *support* $\operatorname{supp}(P)$ as the set of all monomials appearing with non-zero coefficient in the decomposition of P with coefficients on the left in the basis $M_{\mathbf{x},r}$. We then define its *leading monomial* $\mathrm{lm}(P)$ as the largest monomial for $\preccurlyeq$ in $\mathrm{supp}(P)$, its *leading coefficient* $\mathrm{lc}(P)$ as the coefficient of $\mathrm{lm}(P)$ in this decomposition, and its *leading term* $\mathrm{lt}(P)$ as $\mathrm{lc}(P)\,\mathrm{lm}(P)$. We stress that our definition of a leading monomial makes $\mathrm{lm}(P)$ an element of $W_{\mathbf{x}}^r$, whereas some authors choose to see leading monomials as commutative objects in an auxiliary commutative polynomial algebra, introducing commutative variables ξ_i to replace the ∂_i in monomials. For example, we have $\mathrm{lm}(\partial_1 x_1 e_i) = x_1\partial_1 e_i$. Note that an essentially equivalent theory could be developed by choosing monomials as elements of the basis consisting of the products $\boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\alpha}} e_i$, instead of monomials in $M_{\mathbf{x},r}$.

Computations in Weyl algebras rely heavily on a non-commutative generalization of Gröbner bases. After the original introduction [32, 34, 67] there have been a number of presentations of such a theory, including [130, 77, 90]. A first textbook presentation is [114, Chapter 1]. A recent and simpler introduction can be found in [5]. We now adapt this to our need. A Gröbner basis of a left (resp. right) ideal I of $W_{\mathbf{x}}$ with respect to a monomial order $\preccurlyeq$ is a finite set G of generators of I such that for any $a \in I$ there exist $g \in G$ and $q \in W_{\mathbf{x}}$ satisfying $\mathrm{lm}(a) = \mathrm{lm}(qg)$ (resp. $\mathrm{lm}(a) = \mathrm{lm}(gq)$). Note that the non-commutativity of the monomials makes them lack divisibility properties that a commutative variant would: we may have $\mathrm{lm}(qg) \neq \mathrm{lm}(q)\,\mathrm{lm}(g)$ and $\mathrm{lm}(a)$ may not be a multiple of $\mathrm{lm}(g)$, be it on the left or on the right. However, we always have $\mathrm{lm}(a) = \mathrm{lm}(\mathrm{lm}(q)\,\mathrm{lm}(g))$. (The variables ξ_i introduced by other authors is to avoid this formula.) A Gröbner basis allows us to define and compute for any $a \in W_{\mathbf{x}}$ a unique representative of $a + I$ in the quotient $W_{\mathbf{x}}/I$ by means of a non-commutative generalization of polynomial division. We call this unique representative the *remainder of the division of a by the Gröbner basis G* or more shortly *remainder of a modulo the Gröbner basis G*. We denote this remainder $\mathrm{LRem}(a, G)$ when I is a left ideal and $\mathrm{RRem}(a, G)$ when I is a right ideal. The concept of Gröbner bases for ideals of $W_{\mathbf{x}}$ extends to submodules of $W_{\mathbf{x}}^r$ in the same way as the notion of Gröbner bases for polynomial ideals generalizes to submodules of a polynomial algebra (see [11, Chapter 10.4], [55, Chapter 5], or [2, Section 3.5]). The noetherianity of Weyl algebras implies that any left or right submodule of $W_{\mathbf{x}}^r$ admits a (finite) Gröbner basis.

Let A be a subvector space of $W_{\mathbf{x}}^r$. We define $\boldsymbol{\partial}A$ as the $\mathbb{K}$-vector space $\sum_{i=1}^n \partial_i A$. If A is a right $W_{\mathbf{x}}$-module, then $\boldsymbol{\partial}A$ is also a right $W_{\mathbf{x}}$-module.

3.2.4 Integration

The *integral of a $W_{\mathbf{x}}$-module $M \simeq W_{\mathbf{x}}^r/S$* is the $\mathbb{K}$-vector space

$$M/\boldsymbol{\partial}M \simeq W_{\mathbf{x}}^r/(S + \boldsymbol{\partial}W_{\mathbf{x}}^r). \tag{3.5}$$

As already mentioned, it is classical that, if M is holonomic, then the integral (3.5) of M is a finite-dimensional $\mathbb{K}$-linear space [17, Theorem 6.1 of Chapter 1]. Computing relations modulo $\boldsymbol{\partial}M$ is the main matter of this article: given a family in M, we want to find a linear dependency relation on its image in the integral module $M/\boldsymbol{\partial}M$, if any such relation exists.

3.2.5 Data structure for holonomic modules

Algorithmically, we only deal with holonomic $W_{\mathbf{x}}$-modules. They are finitely presented: for such a module M, there exist $W_{\mathbf{x}}$-linear homomorphisms a and b forming an exact sequence

$$W_{\mathbf{x}}^s \xrightarrow{a} W_{\mathbf{x}}^r \xrightarrow{b} M \to 0.$$

Equivalently, this means that $M \simeq W_{\mathbf{x}}^r/S$, where S is the left submodule generated by the image under a of the canonical basis of $W_{\mathbf{x}}^s$. The module S consists of $W_{\mathbf{x}}$-linear combinations of the canonical basis of $W_{\mathbf{x}}^r$, which we always denote $(e_1, \dots, e_r)$. This gives a concrete data structure for representing holonomic modules. It is well known that holonomic modules are *cyclic*, that is generated by a single element [54, Corollary 10.2.6]. This means that we could in principle always assume that $r = 1$. However, some modules have a more natural description with $r > 1$ and transforming the presentation to achieve $r = 1$ has an algorithmic cost that we are not willing to pay. Therefore, we will not assume $r = 1$.

In order to integrate an infinitely differentiable function $f(\mathbf{x})$, we may consider the $W_{\mathbf{x}}$-module generated by f under the natural action of $W_{\mathbf{x}}$ on C^∞ functions. Of course, the holonomic approach to symbolic integration will only work if this module is holonomic. Instead of $W_{\mathbf{x}} \cdot f$, we can also consider any holonomic module that contains it as a submodule.

For example, to integrate a rational function $A/F \in \mathbb{K}(\mathbf{x})$, we can consider the module $\mathbb{K}[\mathbf{x}][F^{-1}]$, which is holonomic. However, finding a finite presentation $W_{\mathbf{x}}^r/S \simeq \mathbb{K}[\mathbf{x}][F^{-1}]$ is not trivial. There are algorithms [103] to solve this problem, but, in terms of efficiency, it is still a practical issue that we do not address in this work. Fortunately, in many cases we can easily construct a holonomic module for integration. See for example Section 3.6.

3.3 Reductions

We consider the Weyl algebra $W_{\mathbf{x}}$ over a field $\mathbb{K}$ and a finitely presented $W_{\mathbf{x}}$-module M given in the form $W_{\mathbf{x}}^r/S$ for some $r \geq 1$ and some submodule S of $W_{\mathbf{x}}^r$ (see Section 3.2.2). The main objective of the section is to compute normal forms in M modulo $\boldsymbol{\partial} M$, or, equivalently, normal forms in $W_{\mathbf{x}}^r$ modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. In other words, we want an algorithm that given some $a \in W_{\mathbf{x}}^r$ computes some $[a] \in W_{\mathbf{x}}^r$, and such that $[a] = [b]$ if and only if $a - b \in S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. This goal is only partially reached with a family of reductions $[.]_\eta$ such that for each pair (a, b), there exists η such that $[a]_\eta = [b]_\eta$ if and only if $a - b \in S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. The existence of a monomial η is not effective, similarly to the maximal total degree to be considered in Takayama's algorithm [128]. This is a step backwards compared to previous methods [102], but computing weaker normal forms allows for more efficient computational methods. Concretely, we do not rely on the computation of b-functions.

The present section is organized as follows. In Section 3.3.1 we define a reduction procedure $[.]$ that partially reduces elements of $W_{\mathbf{x}}^r$ by $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$, in the sense that the procedure will in general not reduce every element of $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$ to zero. In Section 3.3.2

we define a filtration $(F_{\preccurlyeq\eta})_{\eta\in M_{\mathbf{x},r}}$ of the vector space $S+\boldsymbol{\partial}W_{\mathbf{x}}^r$ and we give an algorithm to compute a basis of each vector space $[F_{\preccurlyeq\eta}]$ of reduced forms. Using this basis we define a new reduction $[.]_\eta$ that enhances the first one. In Section 3.3.3, we provide, for some infinite families $(a_i)_{i\geq 0}$ in $W_{\mathbf{x}}^r$, an algorithm for computing an η such that all the $[a_i]_\eta$ lie in a finite-dimensional subspace. In Section 3.3.4, we consider the case where S is the annihilator of e^f for some homogeneous multivariate polynomial f, and we compare our reduction procedures with variants of the Griffiths–Dwork reduction.

3.3.1 Reduction $[.]$ and irreducible elements

Reduction rules.

Let $\preccurlyeq$ be a monomial order on $W_{\mathbf{x}}^r$ and let G be a Gröbner basis of S for this order. We define two binary relations $\to_1$ and $\to_2$ on $W_{\mathbf{x}}^r \times W_{\mathbf{x}}^r$ as follows:

- Given $a \in W_{\mathbf{x}}^r$, $\lambda \in \mathbb{K}$, $m \in M_{\mathbf{x}}$, and $g \in G$, we write

  $$a \to_1 a - \lambda mg$$

  if $\operatorname{lm}(mg)$ is in the support of a but not in the support of $a - \lambda mg$.

- Given $a \in W_{\mathbf{x}}^r$, $\lambda \in \mathbb{K}$, $m \in M_{\mathbf{x},r}$, and $i \in \{1,\dots,n\}$, we write

  $$a \to_2 a - \lambda\partial_i m$$

  if $\operatorname{lm}(\partial_i m)$ is in the support of a but not in the support of $a - \lambda\partial_i m$.

The relation $\to_1$ corresponds to the reduction by the Gröbner basis G of the left module S and the relation $\to_2$ corresponds to the reduction by the Gröbner basis $\{\partial_i e_j \mid i = 1,\dots,n,\ j = 1,\dots,r\}$ of the right module $\boldsymbol{\partial}W_{\mathbf{x}}^r$. Next, we define $\to$ as the relation $\to_1 \cup \to_2$. That is, $a \to b$ if either $a \to_1 b$ or $a \to_2 b$. The relation $\to^+$ is the transitive closure of $\to$: $a \to^+ b$ if there exist $s \geq 1$ and a sequence of s reductions

$$a \to c_1 \to \cdots \to c_s = b \tag{3.6}$$

for some $c_1,\dots,c_s \in W_{\mathbf{x}}^r$. The relation $\to^*$ is the reflexive closure of $\to^+$: $a \to^* b$ if either $a \to^+ b$ or $a = b$. In this situation we say that a *reduces to* b.

Irreducible elements.

We say that an element b is *irreducible* if there is no c such that $b \to c$ and we say that b is a *reduced form of* a if b is irreducible and $a \to^* b$.

Lemma 32. *Let $a, b \in W_{\mathbf{x}}^r$. If $a \to^* b$ then $a - b \in S + \boldsymbol{\partial}W_{\mathbf{x}}^r$.*

Proof. This follows from the definition of $\to_1$ and $\to_2$ since the terms mg and $\partial_i m$ are in S and $\boldsymbol{\partial}W_{\mathbf{x}}^r$, respectively. □

However, the converse of Lemma 32 is not true in general, even when $b = 0$: there may be nonzero irreducible elements in $S + \boldsymbol{\partial}W_{\mathbf{x}}^r$.

Lemma 33. *The irreducible elements of $W_{\mathbf{x}}^r$ form a $\mathbb{K}$-vector space.*

Proof. The set V of all irreducible elements contains 0 and is stable by multiplication by $\mathbb{K}$. Let $a, b \in V$ and assume by contradiction that $a+b$ is not irreducible. Then, there exists a monomial $m \in M_{\mathbf{x},r}$ in the support of $a+b$ that can be reduced by $\to$. Because $a+b=b+a$, we can without loss of generality assume that m is in the support of a. This contradicts the irreducibility of a. Thus $a+b \in V$. □

The vector space of Lemma 33 can be infinite-dimensional, as we now exemplify.

Example 34. Let $S = W_{x_1}\partial_1$ be the left ideal generated by ∂_1 in the Weyl algebra in one pair of generators, (x_1, ∂_1). Note that $W_{x_1}/S \simeq \mathbb{K}[x_1]$ as W_{x_1}-module. Let $\preccurlyeq$ be the lexicographic order for which $\partial_1 \preccurlyeq x_1$. Then, any element of $\mathbb{K}[x_1]$ (as a subspace of W_{x_1}) is irreducible.

Irreducible forms can be computed by alternating left reductions with respect to a Gröbner basis of S (representing the rule $\to_1$) and right reductions with respect to a Gröbner basis of $\boldsymbol{\partial} W_{\mathbf{x}}^r$ (representing the rule $\to_2$). This leads to Algorithm 1. Correctness is clear. The algorithm terminates since the largest reducible monomial in a, if any, decreases at each iteration of the loop.

Algorithm 1 Computation of a reduced form

Input:
- $a \in W_{\mathbf{x}}^r$
- a Gröbner basis G of S

Output:
- a reduced form of a

```
while a is not irreducible
    a ← RRem(a, {∂_i e_j | i = 1, ..., n,  j = 1, ..., r})
    a ← LRem(a, G)
return a
```

Definition 35. We denote by $[a]$ the reduced form of $a \in W_{\mathbf{x}}^r$ that is computed by Algorithm 1.

Proposition 36. *The map* $[.]$ *is* $\mathbb{K}$*-linear.*

Proof. The maps RRem and LRem are $\mathbb{K}$-linear by the uniqueness of the remainder of a division by a Gröbner basis. Let $\tau(a)$ denote the number of iterations of the while loop in Algorithm 1 on input a. Given $\tau \in \mathbb{N}$, let V_τ denote the set of all a for which $\tau(a) \leq \tau$. The restriction of $[.]$ on V_τ takes the same values as the composition of τ copies of RRem and τ copies of LRem in alternation, in which some of the final copies effectively act by the identity as they input irreducible elements. So the restriction of $[.]$ on V_τ is $\mathbb{K}$-linear as a composition of linear maps. The result follows because $W_{\mathbf{x}}^r = \bigcup_{\tau \geq 0} V_\tau$. □

3.3.2 Computation of the irreducible elements of $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$

Again, we fix an order $\preccurlyeq$ and a submodule S of $W_{\mathbf{x}}^r$ by considering a Gröbner basis G of it. Let E be the vector space of all irreducible elements of $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. This vector space can be infinite-dimensional hence we cannot hope to compute all of it. We therefore define a vector-space filtration $(F_{\preccurlyeq \eta})_{\eta \in M_{\mathbf{x},r}}$ of $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$ by

$$F_{\preccurlyeq \eta} = \{ s + d \in W_{\mathbf{x}}^r \mid s \in S,\ d \in \boldsymbol{\partial} W_{\mathbf{x}}^r, \text{ and } \max(\operatorname{lm}(s), \operatorname{lm}(d)) \preccurlyeq \eta \} ,$$

and a vector-space filtration of E by $E_{\preccurlyeq \eta} := F_{\preccurlyeq \eta} \cap E$. We define $F_{\prec \eta}$ and $E_{\prec \eta}$ similarly, by requiring a strict inequality on the maximum of the leading monomials.

Our goal is to obtain an efficient computation of a $\mathbb{K}$-basis of $E_{\preccurlyeq \eta}$. Let us give an intuitive description of our algorithm. By general properties of Gröbner bases, a non-zero element reduces to zero using the relation $\to_1$ (resp. $\to_2$) if and only if it belongs to S (resp. $\boldsymbol{\partial} W_{\mathbf{x}}^r$). The difficulty arises when both reduction rules can be applied to reduce a monomial. For example, take an element s in S such that $\operatorname{lm}(s) \in \operatorname{lm}(S) \cap \operatorname{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$ and, assuming it can be reduced so as to cancel its leading monomial by using $\to_2$, perform this reduction, that is, find s' such that $s \to_2 s'$ and $\operatorname{lm}(s') \prec \operatorname{lm}(s)$. In this case, it is possible that s' is neither in S nor in $\boldsymbol{\partial} W_{\mathbf{x}}^r$, making it a good candidate for an element that does not reduce to 0 by $\to$. The following theorem shows more precisely how generators of E can be obtained.

Theorem 37. *Let $\eta \in M_{\mathbf{x},r}$.*

1. *If $\eta \notin \operatorname{lm}(S) \cap \operatorname{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$, then $E_{\preccurlyeq \eta} = E_{\prec \eta}$.*
2. *If $\eta \in \operatorname{lm}(S) \cap \operatorname{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$, then $E_{\preccurlyeq \eta} = E_{\prec \eta} + \mathbb{K}a$, for any reduced form a of $mg - \partial_i w$, where $w \in W_{\mathbf{x}}^r$, $m \in W_{\mathbf{x}}$ and $g \in G$ are any elements such that $\eta = \operatorname{lm}(mg) = \operatorname{lm}(\partial_i w)$ and $\operatorname{lc}(mg) = \operatorname{lc}(\partial_i w)$. Moreover, such m and g exist because G is a Gröbner basis of S.*

Proof. For the first point, we prove by contradiction that for any $a \in E_{\preccurlyeq \eta}$ and any $s \in S$ and $d \in \boldsymbol{\partial} W_{\mathbf{x}}^r$ satisfying $a = s + d$ and $\max(\operatorname{lm}(s), \operatorname{lm}(d)) \preccurlyeq \eta$, we have in fact $\max(\operatorname{lm}(s), \operatorname{lm}(d)) \prec \eta$. This will imply the equality $E_{\preccurlyeq \eta} = E_{\prec \eta}$. Let us assume that the equality $\max(\operatorname{lm}(s), \operatorname{lm}(d)) = \eta$ holds. Therefore, either $\operatorname{lm}(s) = \operatorname{lm}(d) = \eta$, or $\operatorname{lm}(s) \prec \operatorname{lm}(d) = \eta$, or $\operatorname{lm}(d) \prec \operatorname{lm}(s) = \eta$. The first case is excluded because we assumed $\eta \notin \operatorname{lm}(S) \cap \operatorname{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$. In both remaining cases it is possible to reduce $\operatorname{lm}(a) = \eta$ with one of the two reduction rules. This contradicts the fact that a is irreducible.

For the second point, let m, g, w, i, and a be given as in the statement. We first check that $E_{\prec \eta} + \mathbb{K}a \subseteq E_{\preccurlyeq \eta}$. It is enough to prove that $a \in E_{\preccurlyeq \eta}$. By definition, $mg - \partial_i w \in F_{\preccurlyeq \eta}$, and we check easily that $F_{\preccurlyeq \eta}$ is stable under $\to$. So $a \in F_{\preccurlyeq}$. Since a is also irreducible, we have $a \in E_{\preccurlyeq \eta}$.

Let us prove the other inclusion. Let $f \in E_{\preccurlyeq \eta}$. Then f is irreducible and of the form $s + d$ for $s \in S$ and $d \in \boldsymbol{\partial} W_{\mathbf{x}}^r$ satisfying $\max(\operatorname{lm}(s), \operatorname{lm}(d)) \preccurlyeq \eta$. If this inequality is strict, then $f \in E_{\prec \eta}$, proving $f \in E_{\prec \eta} + \mathbb{K}a$. Otherwise, we have the equality $\max(\operatorname{lm}(s), \operatorname{lm}(d)) = \eta$. Let us remark the equality $\operatorname{lm}(s) = \operatorname{lm}(d)$, for otherwise either $\mathrm{lm}(s) \succ \mathrm{lm}(d)$ and f could be reduced using $\to_1$, or $\mathrm{lm}(s) \prec \mathrm{lm}(d)$ and f could be reduced using $\to_2$. So $\eta = \mathrm{lm}(s) = \mathrm{lm}(d)$. This monomial cannot be $\mathrm{lm}(f)$, for otherwise f could be reduced using any of $\to_1$ and $\to_2$. Hence $\mathrm{lm}(f) \prec \eta$ and $\mathrm{lt}(s) = -\mathrm{lt}(d)$. We decompose s and d as $s = \lambda mg + s'$ and $d = -\lambda\partial_i w + d'$ with $\lambda \in \mathbb{K}$, $s' \in S$, $d' \in \boldsymbol{\partial} W_{\mathbf{x}}^r$, and $\max(\mathrm{lm}(s'), \mathrm{lm}(d')) \prec \eta$. Let h denote $mg - \partial_i w$, which, by hypothesis, has a as a reduced form. This implies an equality of the form $h = a + s'' + d''$ with $s'' \in S$, $d'' \in \boldsymbol{\partial} W_{\mathbf{x}}^r$, and $\max(\mathrm{lm}(s''), \mathrm{lm}(d'')) \prec \eta$. We obtain $f = s + d = \lambda(mg - \partial_i w) + s' + d' = \lambda a + b$ with $b = s' + \lambda s'' + d' + \lambda d''$. Since both f and a are irreducible so is b, thus $b \in E_{\prec\eta}$, proving that f is in $E_{\prec\eta} + \mathbb{K}a$. □

The meaning of Theorem 37 is that the dimension of the filtration $(E_{\preccurlyeq\eta})_\eta$ is susceptible to increase at η only if $\eta \in \mathrm{lm}(S) \cap \mathrm{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$. But this is not necessary as the element a may well be in $E_{\prec\eta}$. The following lemma describes a sufficient condition for this situation.

Lemma 38. *Let $\eta \in \mathrm{lm}(S) \cap \mathrm{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$. If there exist $g \in G$, $m \in M_{\mathbf{x}}$, and some i such that $\eta = \mathrm{lm}(\partial_i mg)$, then*

$$E_{\prec\eta} = E_{\preccurlyeq\eta}.$$

Proof. By Theorem 37, the result reduces to proving that $E_{\prec\eta}$ contains the reduced form a of some $h = \partial_i mg - \partial_j w$ with $\mathrm{lm}(h) \prec \eta$. We choose $j = i$ and $w = mg$, so that $h = 0$, which already is irreducible and in $E_{\prec\eta}$. □

Corollary 39. *Let $\eta \in M_{\mathbf{x},r}$. Let H be the set of monomials $m \preccurlyeq \eta$ such that $m \in \mathrm{lm}(S) \cap \mathrm{lm}(\boldsymbol{\partial} W_{\mathbf{x}}^r)$ and $m \neq \mathrm{lm}(\partial_i pg)$ for any $i \in \{1, \ldots, n\}$, $g \in G$, and $p \in M_{\mathbf{x}}$. For $m \in H$, let $a_m \in W_{\mathbf{x}}^r$ be any reduced form of some $\mathbf{x}^{\boldsymbol{\gamma}} g - \mathrm{lc}(g)\boldsymbol{\partial}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{\gamma}} e_j$, where $g \in G$, $\mathrm{lm}(g) = \mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}^{\boldsymbol{\beta}} e_j$, and $m = \mathrm{lm}(\mathbf{x}^{\boldsymbol{\gamma}} g)$. Then*

$$E_{\preccurlyeq\eta} = \sum_{m\in H} \mathbb{K}a_m. \tag{3.7}$$

Proof. Note that for each $m \in H$, the corresponding $\boldsymbol{\beta}$ is nonzero. Indeed, by definition, $m \in \mathrm{lm}(\boldsymbol{\partial} W_x^r)$ so there is some ∂_i such that $m = \mathrm{lm}(\partial_i m')$ for another monomial m'. Moreover, $m = \mathrm{lm}(\mathbf{x}^{\boldsymbol{\gamma}} g)$, so $\mathrm{lm}(g)$ also contains ∂_i. In particular, the term $\mathrm{lc}(g)\boldsymbol{\partial}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{\gamma}} e_j$ has the form $\partial_i w$. Therefore, Theorem 37 applies and $E_{\preccurlyeq m} = E_{\prec m} + \mathbb{K}a_m$. For a monomial m not in H, either Theorem 37 or Lemma 38 shows that $E_{\preccurlyeq m} = E_{\prec m}$. Then the statement follows by well-founded induction on η. □

To turn Corollary 39 into an algorithm, we introduce a finiteness property of the monomial order $\preccurlyeq$.

Hypothesis 40. *For any two monomials γ and η of $M_{\mathbf{x},r}$, the set of $\boldsymbol{\alpha}$ for which $\mathbf{x}^{\boldsymbol{\alpha}}\gamma \preccurlyeq \eta$ is finite.*

This hypothesis is always satisfied by orders graded by total degree, because a monomial η has a finite number of predecessors in $M_{\mathbf{x},r}$. It is also satisfied by orders eliminating $\mathbf{x}$, in the sense that

$$\boldsymbol{\alpha}' - \boldsymbol{\alpha} \in \mathbb{N}^n \setminus \{0\} \Rightarrow \mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}} e_i \prec \mathbf{x}^{\boldsymbol{\alpha}'}\boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}'} e_{i'}, \tag{3.8}$$

as long as the set of $\boldsymbol{\alpha}$ for which $\mathbf{x}^{\boldsymbol{\alpha}} \preccurlyeq x_i$ is finite for each $i \in \{1, \dots, n\}$. For example, this contains "elimination orders" [55] or "block orders" [11] that first order by total degree in $\mathbf{x}$, but not a lexicographical order that has $x_1 > x_2 > \partial_1 > \partial_2$.

Algorithm 2 Computation of $E_{\preccurlyeq \eta}$

Input:

- a Gröbner basis G of S
- $\eta \in M_{\mathbf{x},r}$

Output:

- a generating family of the $\mathbb{K}$-vector space $E_{\preccurlyeq \eta}$

1 $G' \leftarrow \{g \in G \mid \mathrm{lm}(g) \text{ involves some } \partial_i\}$
2 $H \leftarrow \{\mathrm{lm}(\mathbf{x}^{\boldsymbol{\gamma}} g) \mid \boldsymbol{\gamma} \in \mathbb{N}^n,\ g \in G',\ \text{and } \mathrm{lm}(\mathbf{x}^{\boldsymbol{\gamma}} g) \preccurlyeq \eta\}$ # finite by Hypothesis 40
3 $H \leftarrow H \setminus \{\mathrm{lm}(\partial_i m g) \mid m \in M_{\mathbf{x}},\ g \in G,\ 1 \leq i \leq n\}$
4 $B \leftarrow \varnothing$
5 **for** $m \in H$
6   pick $g \in G'$ and $\boldsymbol{\gamma}$ such that $m = \mathrm{lm}(\mathbf{x}^{\boldsymbol{\gamma}} g)$
7     $\mathbf{x}^{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\beta}} e_i \leftarrow \mathrm{lm}(g)$ # by construction $\beta \neq 0$
8     $B \leftarrow B \cup \left\{ \left[\mathbf{x}^{\boldsymbol{\gamma}} g - \mathrm{lc}(g) \boldsymbol{\partial}^{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{\gamma}} e_i \right] \right\}$
9 **return** B

Theorem 41. *Under Hypothesis 40 Algorithm 2 is correct and terminates.*

Proof. Termination is obvious since the set on line 2 is finite, by hypothesis. For the correction, we observe that H computed in the algorithm is the same as the set H described in Corollary 39. □

Definition 42. Let B_η be an echelon form of the generating family returned by Algorithm 2 on input η. We define a reduction $[.]_\eta$ from $W_{\mathbf{x}}^r$ into itself by

$$[a]_\eta = \mathrm{Reduce}([a], B_\eta)$$

where $[.]$ is the map defined by Algorithm 1 and $\mathrm{Reduce}(., B_\eta)$ is the reduction algorithm by the echelon form B_η.

Proposition 43. *The map $[.]_\eta$ is $\mathbb{K}$-linear.*

Proof. This follows from Proposition 36 and the $\mathbb{K}$-linearity of $\mathrm{Reduce}(., B_\eta)$. □

Theorem 44. *For any $a \in S + \boldsymbol{\partial} W_{\mathbf{x}}^r$ there exists $\eta \in M_{\mathbf{x},r}$ such that for all $\eta' \succcurlyeq \eta$, the remainder $[a]_{\eta'}$ is zero.*

Proof. The element $[a]$ is congruent to a modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$, so it is in $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$, like a itself. Moreover, it is irreducible, and so it is in E by the definition of E. Because of the equality $E = \bigcup_{\eta \in M_{\mathbf{x},r}} E_{\preccurlyeq \eta}$, there exists η such that $[a] \in E_{\preccurlyeq \eta}$ and thus

$\operatorname{Reduce}([a], B_\eta) = 0$. For $\eta' \succcurlyeq \eta$, the vector space $\operatorname{Span}_{\mathbb{K}}(B_\eta)$ is included in $\operatorname{Span}_{\mathbb{K}}(B_{\eta'})$, so $[a]_{\eta'} = \operatorname{Reduce}([a], B_{\eta'}) = \operatorname{Reduce}(\operatorname{Reduce}([a], B_\eta), B_{\eta'}) = \operatorname{Reduce}(0, B_{\eta'}) = 0$. □

Definition 45. The *normal form* of an element $a \in W_{\mathbf{x}}^r$ modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$ is the unique element $a' \in W_{\mathbf{x}}^r$ such that $a \equiv a' \pmod{S + \boldsymbol{\partial} W_{\mathbf{x}}^r}$ and no monomial of a' is the leading monomial of an element of $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$.

Corollary 46. *For any $a \in W_{\mathbf{x}}^r$, there exists $\eta \in M_{\mathbf{x},r}$ such that for all $\eta' \succcurlyeq \eta$, the remainder $[a]_{\eta'}$ is the normal form of a modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$.*

Proof. Let a' be the normal form of a. Let η such that $[a - a']_\eta = 0$, given by Theorem 44. By definition, $[.]_\eta$ replaces monomials by smaller ones, but only if this is possible, so that we have $[a']_\eta = a'$. By linearity of $[.]_\eta$, we obtain $[a]_\eta = a'$. □

3.3.3 Confinement

"Computing" in the quotient $M/\boldsymbol{\partial} M \simeq W_{\mathbf{x}}^r/(S + \boldsymbol{\partial} W_{\mathbf{x}}^r)$ can take on several forms, with various levels of potency. In the strongest interpretation, we want to compute a basis of the quotient, as a $\mathbb{K}$-linear space, and we want to be able to compute normal forms in $W_{\mathbf{x}}^r$ modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. In a weaker sense, we merely want to be able to capture the finiteness of the quotient space, without ensuring the linear independence of a finite generating set or even producing it explicitly. In view of our needs for integration algorithms in the next sections, there is an even weaker sense: given $a \in W_{\mathbf{x}}^r$ (which will designate a function to be integrated) and a $W_{\mathbf{x}}$-linear map from $W_{\mathbf{x}}^r$ to itself (which will be related to taking derivatives with respect to a parameter $t \in \mathbb{K}$), we need to testify the finite-dimensionality of the span over $\mathbb{K}$ of the orbit $\{L^i(a) \mid i \in \mathbb{N}\}$ modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. In this section, we show that the reductions $[.]_\eta$ can be used to find, for any a and L, a finite set B that witnesses this finite-dimensionality.

Algorithm 3 Computation of a confinement

Input:
- a module $S \subseteq W_{\mathbf{x}}^r$ given by a Gröbner basis
- $a \in W_{\mathbf{x}}^r$
- a $W_{\mathbf{x}}$-linear map $L : W_{\mathbf{x}}^r \to W_{\mathbf{x}}^r$ given by an $r \times r$ matrix
- $\rho \in \mathbb{N}$

Output:
- an effective confinement for a and L modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$

$s \leftarrow \rho$
$\eta \leftarrow$ the largest monomial of degree s
$B \leftarrow \varnothing$
$Q \leftarrow \operatorname{supp}([a]_\eta)$
while $Q \setminus B \neq \varnothing$
  $m \leftarrow$ an element of $Q \setminus B$
  if $\deg m > s - \rho$
    $s \leftarrow s + 1$
    goto line 2
  $Q \leftarrow Q \cup \operatorname{supp}([L(m)]_\eta)$
  $B \leftarrow B \cup \{m\}$
return (η, B)

Definition 47. An *effective confinement* for $a \in W_{\mathbf{x}}^r$ and a $W_{\mathbf{x}}$-linear map $L : W_{\mathbf{x}}^r \to W_{\mathbf{x}}^r$ is a pair (η, B) consisting of a monomial η and of a finite subset $B \subseteq M_{\mathbf{x},r}$, and satisfying:

1. the support of $[a]_\eta$ is included in B;
2. the support of $[L(m)]_\eta$ is included in B for any $m \in B$.

An effective confinement is *free* if the elements of B are $\mathbb{K}$-linearly independent modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$.

Theorem 48. *Algorithm 3 is correct. It terminates if $M/\boldsymbol{\partial} M$ is finite-dimensional, for example if M is holonomic. Moreover, if the input parameter ρ is large enough, then Algorithm 3 outputs a free effective confinement.*

Proof. We first address correctness. Consider the sets B and Q after any iteration of the while loop. By construction, we have $B \subseteq Q$, $\operatorname{supp}([a]_\eta) \subseteq Q$, and $\operatorname{supp}([L(m)]_\eta) \subseteq Q$ for any $m \in B$. If the halting condition $Q \setminus B \neq \varnothing$ of the while loop is reached, that is, equivalently, if $Q \subseteq B$ holds at the end of an iteration, then we have $B = Q$. In conclusion, the returned value (η, B) is an effective confinement.

As for termination, let $C \subseteq M_{\mathbf{x}}^r$ be the set of monomials that are normal forms modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. As a consequence of Definition 45, the set of normal forms is the vector space $\operatorname{Span}_{\mathbb{K}}(C)$. Moreover, a basis of the quotient space $M/\boldsymbol{\partial} M$ is formed by the classes modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$ of all elements of C, so in particular, C is finite by the hypothesis of finite dimension. By Corollary 46, there is therefore some η_∞ such that for any $\eta \succcurlyeq \eta_\infty$,

$$\operatorname{supp}([a]_\eta) \subseteq C \text{ and } \forall m \in C,\ \operatorname{supp}([L(m)]_\eta) \subseteq C. \tag{3.9}$$

Since each iteration of the *while* loop treats a different monomial m, and since there are finitely many monomials of degree at most $s-\rho$, the *while* loop terminates. It terminates either because $Q \setminus B = \varnothing$, in which case the algorithm terminates, or because Q contains an element of degree larger that $s-\rho$, in which case we increase s. So, either the algorithm terminates, or s tends to ∞.

Assume $s \to \infty$. At some point, we will have $s \geq \deg \eta_\infty$, so after line 2 is executed, we have the inequalities $\eta \succcurlyeq \eta_\infty x_1^{s-\deg(\eta_\infty)} \succcurlyeq \eta_\infty$, because η is the largest monomial of degree s and by the definition of a monomial order. In these circumstances, the set Q is a subset of C at every iteration of the main loop, because of (3.9), and so is B because of the invariant $B \subseteq Q$. Since $s \to \infty$, we also reach a point where $\rho + \deg m \leq s$ for all $m \in C$. After this point, s is not increased anymore. This contradiction shows that the algorithm terminates.

If the input ρ satisfies $\rho \geq \deg \eta_\infty$, we have $s \geq \rho \geq \deg \eta_\infty$, and by the same reasoning as in the previous paragraph, we have again $\eta \succcurlyeq \eta_\infty$. This is so during the whole execution of the algorithm. Therefore, like in the preceding paragraph, we have $B \subseteq Q \subseteq C$ during the execution of the while loop. So, the output set B is a subset of C, which is a free family modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. □

3.3.4 Comparison with the Griffiths–Dwork reduction

Let $f \in \mathbb{K}[\mathbf{x}]$ be a homogeneous polynomial and let M be the $W_{\mathbf{x}}$-module $\mathbb{K}[\mathbf{x}]e^f$, where ∂_i acts by $\partial_i \cdot e^f = \frac{\partial f}{\partial x_i} e^f$. When f defines a smooth variety, we can compute in $M/\boldsymbol{\partial} M$ using the Griffiths–Dwork reduction [60, 61, 71]. This is usually presented with rational functions in $\mathbb{K}[\mathbf{x}, f^{-1}]$ but the exponential formulation is equivalent (for example, see [57, 86]). The module M admits the presentation

$$M \simeq \frac{W_{\mathbf{x}}}{\sum_i W_{\mathbf{x}}(\partial_i - f_i)},$$

where f_i denotes the partial derivative $\frac{\partial f}{\partial x_i}$. (This presentation is what makes the exponential formulation easier in our setting. A presentation of the holonomic $W_{\mathbf{x}}$-module $\mathbb{K}[\mathbf{x}, f^{-1}]$ is much harder to compute [102].)

We briefly present the Griffiths–Dwork reduction and observe that irreducible elements for the Griffiths–Dwork reduction are exactly the irreducible elements for our reduction $\to$.

The Griffiths–Dwork reduction.

Let $\preccurlyeq_0$ be a monomial order on $\mathbb{K}[\mathbf{x}]$, and, for this monomial ordering, let G_0 be the minimal Gröbner basis of the polynomial ideal $I = (f_1, \ldots, f_n)$. Given a homogeneous polynomial $a \in \mathbb{K}[\mathbf{x}]$, we can compute the remainder r of the multivariate division of a by G_0 and the cofactors $b_1, \dots, b_n \in \mathbb{K}[\mathbf{x}]$ such that

$$a = r + \sum_{i=1}^{n} b_i f_i. \tag{3.10}$$

By homogeneity, $\deg b_i = \deg a - \deg f + 1$ (unless $b_i = 0$). Then, the rule for the derivative of a product yields

$$ae^f = re^f - \underbrace{\sum_{i=1}^{n} \frac{\partial b_i}{\partial x_i} e^f}_{\text{degree } \deg a - \deg f} + \underbrace{\sum_{i=1}^{n} \partial_i \cdot \left(b_i e^f\right)}_{\in \boldsymbol{\partial} M}. \tag{3.11}$$

The last term is in $\boldsymbol{\partial} M$, so we ignore it, and the second term has lower degree than a, so we can apply the same procedure recursively, which will terminate by induction on the degree. In the end, we obtain a reduced form $ae^f \equiv re^f \pmod{\boldsymbol{\partial} M}$ where $r \in \mathbb{K}[\mathbf{x}]$ is irreducible with respect to G_0. These are the irreducible elements for the Griffiths–Dwork reduction.

This reduction is defined for any homogeneous polynomial f, but it enjoys special properties when $f_1, \dots, f_n$ do not have any non-trivial common zero in an algebraic closure of $\mathbb{K}$. Geometrically, this means that f defines a smooth hypersurface in $\mathbb{P}^{n-1}(\mathbb{K})$. Griffiths [71] proved, under this smoothness assumption, that the reduced form of any ae^f vanishes if and only if $ae^f \in \boldsymbol{\partial} M$.

Comparison with the reduction $\to$.

We consider the reduction rule $\to$ applied to the left ideal S of $W_\mathbf{x}$ generated by $\partial_i - f_i$, so that $\mathbb{K}[\mathbf{x}]e^f \simeq W_\mathbf{x}/S$ (Section 3.3.1). Let $\preccurlyeq$ be a monomial order on $W_\mathbf{x}$ that eliminates $\mathbf{x}$ (see (3.8)) and agrees with $\preccurlyeq_0$ on $\mathbb{K}[\mathbf{x}]$. By following the steps of Buchberger's algorithm, we observe that there is a Gröbner basis G of S in which each element is:

1. either an element of the form $r - \sum_{i=1}^{n} b_i \partial_i$, with $r \in G_0$, $b_i \in \mathbb{K}[\mathbf{x}]$ of degree $\deg(r) - \deg(f_i)$, and $r = \sum_i b_i f_i$,

2. or an element of $W_\mathbf{x}\partial_1 + \cdots + W_\mathbf{x}\partial_n$.

We will call such elements respectively of the *first kind* and of the *second kind*.

We now characterize irreducible elements in $W_\mathbf{x}$ with respect to $\to$. Let $a \in W_\mathbf{x}$ be an irreducible element. Since a cannot be reduced with $\to_2$, it contains no ∂_i, so it is a polynomial. Since a cannot be reduced with $\to_1$, no monomial in a is divisible by the leading term of an element of G. By considering the elements of the first kind, we see that the monomials of a are not divisible by the leading monomial of any element of G_0. So a is irreducible with respect to G_0. The converse also holds: if $a \in \mathbb{K}[\mathbf{x}]$ is irreducible with respect to G_0, then a is irreducible in $W_\mathbf{x}$ with respect to S. In this sense, we can regard Algorithm 1 as a generalization of the Griffiths–Dwork reduction.

As we observed, this reduction is not enough to compute in $M/\boldsymbol{\partial} M$, since there may be nonzero irreducible elements in $S + \boldsymbol{\partial} W_{\mathbf{x}}$. In the case of rational functions, Lairez [86] gave an algorithm to compute them efficiently. The algorithm that we have given in Section 3.3.2 behaves differently. In short, the algorithm in [86] would only consider elements of the second kind with degree 1 in the ∂_i, whereas we consider all elements of the second kind. On the one hand, this seems to give more reduction power, on the other hand the cost of computing them is higher. This indicates room for improvement in future work.

3.4 Creative Telescoping by Reduction

In the previous section, we obtained an algorithm for normalizing modulo derivatives in a holonomic $W_{\mathbf{x}}$-module. In this section, we introduce a parameter t and differentiation with respect to t. It would be natural to work with a holonomic $W_{t,\mathbf{x}}$-module, but in view of the previous section, we need a finitely presented module over a Weyl algebra in the derivatives with respect to $\mathbf{x}$ only. This motivates the following context.

We consider the Weyl algebra $W_{\mathbf{x}}(t) = \mathbb{K}(t) \otimes_{\mathbb{K}} W_{\mathbf{x}}$ (which is just a Weyl algebra over the field $\mathbb{K}(t)$), and a holonomic $W_{\mathbf{x}}(t)$-module M with a compatible derivation ∂_t, that is, a $\mathbb{K}$-linear map $\partial_t : M \to M$ such that for any $a \in W_{\mathbf{x}}(t)$ and any $m \in M$,

$$\partial_t \cdot am = \tfrac{\partial a}{\partial t} m + a \partial_t \cdot m,$$

where $\frac{\partial a}{\partial t}$ is the coefficient-wise differentiation in $W_{\mathbf{x}}(t)$. In other words, M is a $W_{t,\mathbf{x}}(t)$-module that is holonomic as a $W_{\mathbf{x}}(t)$-module.

It is also convenient to fix a finite presentation $W_{\mathbf{x}}(t)^r/S$ of M and assume that there is a $W_{\mathbf{x}}(t)$-linear map $L : W_{\mathbf{x}}(t)^r \to W_{\mathbf{x}}(t)^r$ such that for any $a \in W_{\mathbf{x}}(t)^r$,

$$\partial_t \cdot \operatorname{pr}_S(a) = \operatorname{pr}_S\left(\tfrac{\partial a}{\partial t} + L(a)\right), \tag{3.12}$$

where pr_S is the canonical map $W_{\mathbf{x}}(t)^r \to M$. In particular, note that S is stable under $\frac{\partial}{\partial t} + L$. From the algorithmic point of view, we represent M by its finite presentation and the derivation ∂_t by the $r \times r$ matrix of the endomorphism L. We explain in Section 3.4.3 how to obtain this setting from a holonomic $W_{t,\mathbf{x}}$-module.

Using the algorithm of the previous section, we aim at describing an algorithm that performs integration with respect to $x_1, \ldots, x_n$, in the following sense. Given $f \in M$, we want to compute a nonzero operator $P(t, \partial_t) \in W_t(t)$ such that

$$P(t, \partial_t) \cdot f \in \boldsymbol{\partial} M \tag{3.13}$$

with the motivation that $P(t, \partial_t)$ is then an annihilating operator of the integral of f with respect to $x_1, \ldots, x_n$. The principle of integration by reduction is described in Section 3.4.1 and an algorithm is presented in Section 3.4.2.

3.4.1 Integration by reduction

We utilize the family of reductions $[.]_\eta$ defined in Section 3.3.2. Let f be an element of $W_{\mathbf{x}}(t)^r$, let η be some monomial in $M_{\mathbf{x},r}$ and let $(g_i)_{i\geq 0}$ be the sequence in $W_{\mathbf{x}}(t)^r$ defined by

$$g_0 = [f]_\eta \quad \text{and} \quad g_{i+1} = \tfrac{\partial g_i}{\partial t} + [L(g_i)]_\eta \text{ for all } i \geq 0. \tag{3.14}$$

As usual with integration-by-reduction algorithms, we relate the dependency relations between the reduced forms g_i to the operators $P \in W_t(t)$ such that $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$, which, as is traditional, we call *telescopers* for f.

Lemma 49. *For any $i \geq 0$,*

$$\mathrm{pr}_S(g_i) \equiv \partial_t^i \cdot \mathrm{pr}_S(f) \pmod{\boldsymbol{\partial} M}. \tag{3.15}$$

Proof. For $i = 0$, this means that $[f]_\eta \equiv f \pmod{S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r}$, which holds by construction of $[.]_\eta$. By property of $[.]_\eta$, again, there is some $s_i \in S$ and some $\Delta_i = \sum_j \partial_j a_{i,j} \in \boldsymbol{\partial} W_{\mathbf{x}}(t)^r$ such that

$$[L(g_i)]_\eta = L(g_i) + s_i + \Delta_i.$$

Therefore, using the $W_{\mathbf{x}}(t)$-linearity of pr_S and $\mathrm{pr}_S(s_i) = 0$, we obtain

$$\begin{aligned} \mathrm{pr}_S(g_{i+1}) &= \mathrm{pr}_S\left(\tfrac{\partial g_i}{\partial t} + L(g_i)\right) + \mathrm{pr}_S(\Delta_i) \\ &= \partial_t \cdot \mathrm{pr}_S(g_i) + \sum_j \partial_j \, \mathrm{pr}_S(a_{i,j}), \qquad \text{using (3.12)}, \\ &\equiv \partial_t \cdot \mathrm{pr}_S(g_i) \pmod{\boldsymbol{\partial} M}. \end{aligned}$$

The claim follows by induction on i, using that $\partial_t \cdot \boldsymbol{\partial} M \subset \boldsymbol{\partial} M$, since ∂_t commutes with $\boldsymbol{\partial}$. □

Lemma 50. *Let $P = \sum_{i=0}^N c_i \partial_t^i \in W_t(t)$.*

1. *If $c_0 g_0 + \cdots + c_N g_N = 0$, then $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$.*
2. *If $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$, then $c_0 g_0 + \cdots + c_N g_N \in S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r$.*
3. *If η is large enough and if $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$, then $c_0 g_0 + \cdots + c_N g_N = 0$. (Note that "large enough" depends on f and P.)*

Proof. The first assertion follows directly from Lemma 49. Conversely, assume that $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$. This implies, again by Lemma 49, that

$$\sum_{i=0}^N c_i g_i \in S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r,$$

proving the second assertion. Now, we observe that, if in addition η is large enough, the g_i are normal forms modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r$. Indeed, by Corollary 46, the $[L(g_i)]_\eta$ are normal forms; and since being a normal form is a condition on the monomial support, it is stable under coefficient-wise differentiation, so the $\frac{\partial g_i}{\partial t}$ are normal forms, by induction on i. So the linear combination $\sum_i c_i g_i$ is also a normal form modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r$. This implies $\sum_i c_i g_i = 0$. □

3.4.2 An algorithm for integrating by reduction

To turn Lemma 50 into an algorithm to compute a telescoper, it only remains to find a suitable η. We use the idea of *confinement* (Section 3.3.3). Using Algorithm 3, we can compute an effective confinement for f and L. Recall that this is a monomial η and a finite set B of monomials in $M_{\mathbf{x},r}$ such that $\operatorname{supp}([f]_\eta) \subseteq B$ and $\operatorname{supp}([L(b)]_\eta) \subseteq B$ for all $b \in B$. The following statement explains that reduced forms of successive derivatives with respect to t therefore lie in the finite-dimensional vector-space $\operatorname{Span}_{\mathbb{K}(t)}(B)$.

Lemma 51. *Let (η, B) be an effective confinement for f and L. Let $(g_i)_{i\geq 0}$ be the sequence defined by (3.14). Then, for all $i \geq 0$, $g_i \in \operatorname{Span}_{\mathbb{K}(t)}(B)$.*

Proof. By definition of an effective confinement, $[f]_\eta \in \operatorname{Span}_{\mathbb{K}(t)}(B)$, and the space $\operatorname{Span}_{\mathbb{K}(t)}(B)$ is stable under $[L(.)]_\eta$. Moreover, $\operatorname{Span}_{\mathbb{K}(t)}(B)$ is stable under $\frac{\partial}{\partial t}$. So the claim follows from the definition of g_i. ☐

Algorithm 3 and Lemma 51 combine into Algorithm 4, whose main properties are provided in the following theorem.

Algorithm 4 Integration using reductions

Input:
- a holonomic module $W_{\mathbf{x}}(t)^r/S$
- a derivation map $\partial_t : W_{\mathbf{x}}(t)^r/S \to W_{\mathbf{x}}(t)^r/S$ given by the matrix of an endomorphism L, as in (3.12).
- an element $f \in W_{\mathbf{x}}(t)^r$
- an integer $\rho \geq 0$

Output:
- $P = c_0 + \cdots + c_N \partial_t^N$ such that $c_i \in \mathbb{K}(t)$, $c_N \neq 0$ and $P \cdot \operatorname{pr}_S(f) \in \boldsymbol{\partial} M$

$(\eta, B) \leftarrow$ an effective confinement obtained from (S, f, L, ρ) by Algorithm 3
$g_0 \leftarrow [f]_\eta$
$N \leftarrow 0$
while $g_0, \ldots, g_N$ are linearly independent over $k(t)$
    $g_{N+1} \leftarrow \frac{\partial g}{\partial t} + [L(g)]_\eta$
    $N \leftarrow N + 1$
return $c_0 + \cdots + c_N \partial_t^N$ s.t. $c_0 g_0 + \cdots + c_N g_N = 0$, $c_i \in \mathbb{K}(t)$ and $c_N \neq 0$.

Theorem 52. *Algorithm 4 is correct and terminates. Moreover, if ρ is large enough, then it outputs a minimal telescoper for the input.*

Proof. Correctness follows from Lemma 50. As to termination, it follows from Lemma 51: because the set B is finite, the infinite family of elements $g_0, g_1, \ldots$ is linearly dependent, so the main loop terminates for some N less than or equal to the cardinality of B.

As for the minimality, it is clear that the algorithm outputs a non-trivial relation $c_0 g_0 + \dots + c_N g_N = 0$ with minimal possible N among those available for the sequence $(g_i)_{i \geq 0}$. Besides, consider any telescoper $P = c_0 + \ldots + c_\Omega \partial_t^\Omega$. By point 2 of Lemma 50, we have

$$c_0 g_0 + \dots + c_\Omega g_\Omega \in S + \boldsymbol{\partial} W_{\mathbf{x}}(t)^r. \tag{3.16}$$

Assume that ρ is large enough, in the sense of Theorem 48, so that the confinement is free, meaning that the elements of B are independent modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}^r$. The linear combination in (3.16) is a linear combination of elements of B, so it must be zero: $c_0 g_0 + \dots + c_\Omega g_\Omega = 0$. So Algorithm 4 will output a relation for an N that is at most the minimal order of telescopers. □

Remark 53. Algorithm 4 can be modified to compute a system of linear differential equations satisfied by an integral depending on multiple parameters $t_1, \ldots, t_p$. These parameters can also be associated to other Ore operators [45] than the differentiation provided they define a map on $W_{\mathbf{x}}^r(t_1, \ldots, t_p)$.

3.4.3 Scalar extension

Let P be a holonomic $W_{t,\mathbf{x}}$-module and let $M = \mathbb{K}(t) \otimes_{\mathbb{K}[t]} P$. This space M is a $W_{\mathbf{x}}(t)$-module in a natural way. Moreover, we can define a derivation ∂_t by

$$\partial_t \cdot (a \otimes m) = \tfrac{\partial a}{\partial t} \otimes m + a \otimes (\partial_t \cdot m).$$

This derivation commutes with the action of the ∂_i.

In this section, we aim to compute M from P, so that we can apply the integration algorithm (Section 3.4.2). Let us first make explicit what we mean. We assume that P is given by a finite presentation, that is, $P = W_{t,\mathbf{x}}^s / J$ for some $s \geq 0$ and some submodule $J \subseteq W_{t,\mathbf{x}}^s$ given by a finite set of generators. Computing M means computing a finite presentation $M \simeq W_{\mathbf{x}}(t)^r / S$, and a $W_{\mathbf{x}}(t)$-linear map $L : W_{\mathbf{x}}(t)^r \to W_{\mathbf{x}}(t)^r$ such that (3.12) holds.

It is not obvious that such a finite presentation exists because M does not have any obvious finite set of generators. However, this existence is implied by the holonomy of M. Here, we give a proof based on restriction of D-modules.

Lemma 54. *If P is a $W_{t,\mathbf{x}}$-holonomic module, then $M = \mathbb{K}(t) \otimes_{\mathbb{K}[t]} P$ is a $W_{\mathbf{x}}(t)$-holonomic module.*

The statement is similar in nature to the well-known statement that “holonomic implies D-finite”.

Proof. Let ξ be a new variable. Consider the field $\mathbb{L} = \mathbb{K}(\xi)$. Introduce the $(1+n)$th Weyl algebra with coefficients in $\mathbb{L}$, which we denote $W_{t,\mathbf{x}}(\xi)$. Consider as well the $W_{t,\mathbf{x}}(\xi)$-module $P' = \mathbb{L} \otimes_{\mathbb{K}} P$. This scalar extension of the base field this preserves holonomy. So P' is holonomic. Consider now the embedding map $F : \mathbb{L}^n \to \mathbb{L}^{1+n}$ defined by

$$(x_1, \ldots, x_n) \mapsto (\xi, x_1, \ldots, x_n),$$

and the inverse image F^*P', that is, the restriction of P' at $t = \xi$. We just need to know that F^*P' is:

- a $W_{\mathbf{x}}(\xi)$-module, which is by [54, construction of §14.1 and §14.2],
- holonomic as a $W_{\mathbf{x}}(\xi)$-module, which is by [54, Theorem 18.1.4],
- isomorphic to $P'/(t-\xi)P'$, which is obtained as a suitable variant of [54, §15.1], by making $Y = t$ in that reference before specializing at ξ instead of 0.

In particular, we have

$$F^*P' \simeq P'/(t-\xi)P' \simeq \mathbb{L}[t]/(t-\xi) \otimes_{\mathbb{L}[t]} P'.$$

Next, we check that

$$\begin{aligned} P' &= \mathbb{L} \otimes_{\mathbb{K}} P, && \text{by definition} \\ &\simeq \mathbb{L} \otimes_{\mathbb{K}} (\mathbb{K}[t] \otimes_{\mathbb{K}[t]} P), && \text{because } P \text{ is a } \mathbb{K}[t] \text{ module} \\ &\simeq \mathbb{L}[t] \otimes_{\mathbb{K}[t]} P, && \text{by associativity of } \otimes, \end{aligned}$$

and therefore, using associativity of $\otimes$ again,

$$F^*P' \simeq \mathbb{L}[t]/(t-\xi) \otimes_{\mathbb{K}[t]} P.$$

Finally, we observe the isomorphism $\mathbb{K}(t) \simeq \mathbb{L}[t]/(t-\xi)$ as $\mathbb{K}[t]$-algebras under the map $f(t) \mapsto f(\xi)$, so we obtain $F^*P' \simeq \mathbb{K}(t) \otimes_{\mathbb{K}[t]} P = M$, by definition of M. Since F^*P' is holonomic, this gives the claim. □

We now describe an algorithm for computing M. Recall that $P = W_{t,\mathbf{x}}^s/J$, so $M \simeq W_{t,\mathbf{x}}(t)^s/J(t)$ where $J(t)$ is the submodule $\mathbb{K}(t) \otimes_{\mathbb{K}[t]} J$ of $W_{t,\mathbf{x}}(t)^s$ generated by J. We can compute normal forms in M using Gröbner bases in $W_{t,\mathbf{x}}(t)^s$ after fixing a monomial order on all monomials $\mathbf{x}^\alpha \boldsymbol{\partial}^\beta \partial_t^k e_i$ [e.g., 45]. We choose a monomial order that eliminates ∂_t, that is, any monomial order such that

$$k < k' \Rightarrow \mathbf{x}^{\boldsymbol{\alpha}} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}} \partial_t^k e_i \prec \mathbf{x}^{\boldsymbol{\alpha}'} \boldsymbol{\partial}_{\mathbf{x}}^{\boldsymbol{\beta}'} \partial_t^{k'} e_{i'}.$$

Let G denote a Gröbner basis of $J(t)$ for such an elimination order.

As a $W_{\mathbf{x}}(t)$-module, $W_{t,\mathbf{x}}(t)^s$ is generated by the set

$$\left\{ \partial_t^i e_j \;\middle|\; i \geq 0,\ 1 \leq j \leq s \right\}.$$

So M is generated by the image of this set. This is an infinite family, but, since M is $W_{\mathbf{x}}(t)$-holonomic, M is actually a Noetherian $W_{\mathbf{x}}(t)$-module, finitely generated in particular. To describe M, we need to find an explicit finite generating set and the module of relations between the generators.

For $a \in W_{t,\mathbf{x}}(t)^s$, let $\operatorname{ind}(a)$ denote the degree of a with respect to ∂_t, which we will call the index of a. In other words, this is the smallest integer $k \geq 0$ such that a is in the sub-$W_{\mathbf{x}}(t)$-module generated by

$$B_k = \left\{ \partial_t^i e_j \;\middle|\; 0 \leq i \leq k,\ 1 \leq j \leq s \right\}.$$

Moreover, for $a \in W_{t,\mathbf{x}}(t)^s$, let $\operatorname{ind}_{J(t)}(a)$ denote

$$\operatorname{ind}_{J(t)}(a) = \min \{\operatorname{ind}(b) \mid b \in W_{t,\mathbf{x}}(t)^s \text{ and } a \equiv b \pmod{J(t)}\}. \tag{3.17}$$

Given a, we can compute $\operatorname{ind}_{J(t)}(a)$ using the Gröbner basis G:

$$\operatorname{ind}_{J(t)}(a) = \operatorname{ind}(\operatorname{LRem}(a, G)).$$

Indeed: we have $a \equiv \operatorname{LRem}(b, G)$ if $a \equiv b \pmod{J(t)}$ and the elimination property shows $\operatorname{ind}(\operatorname{LRem}(b, G)) \le \operatorname{ind}(b)$, so that $\operatorname{ind}(b)$ can be replaced with $\operatorname{ind}(\operatorname{LRem}(b, G))$ in (3.17); then the Gröbner basis property shows $\operatorname{LRem}(a, G) = \operatorname{LRem}(b, G)$ if $a \equiv b \pmod{J(t)}$.

Lemma 55. *There is $\ell \ge 0$ such that $\operatorname{ind}_{J(t)}(\partial_t^{\ell+1} e_i) \le \ell$ for any $1 \le i \le s$. Moreover, for any such ℓ:*

1. *M is generated as a $W_{\mathbf{x}}(t)$-module by the image of B_ℓ in it,*
2. *$\operatorname{ind}_{J(t)}(a) \le \ell$ for any $a \in W_{t,\mathbf{x}}(t)^s$.*

Proof. Since M is Noetherian, the increasing sequence of the $W_{\mathbf{x}}(t)$-modules S_k generated by the images of the B_k in M is stationary: there exists $\ell \ge 0$ for which the $W_{\mathbf{x}}(t)$-module S_ℓ contains all the S_k for $k \ge 0$, and is therefore equal to M. For such an integer ℓ, any $k > \ell$, and any j, the image of $\partial_t^k e_j$ in M is in S_k, therefore in $S_\ell = M$. Consequently, there exist coefficients $c_{h,i} \in W_{\mathbf{x}}(t)$ satisfying

$$\partial_t^k e_j \equiv \sum_{h \le \ell,\ i} c_{h,i} \partial_t^h e_i \pmod{J(t)}.$$

By the definition (3.17), $\operatorname{ind}_{J(t)}(\partial_t^k e_j)$ is less than or equal to the index of the right-hand side, which by construction is less than or equal to ℓ. We obtain that ℓ is a uniform bound on all $\operatorname{ind}_{J(t)}(\partial_t^k e_j)$. This proves in particular the first part of the statement, on the existence of ℓ. For the second part, we fix such an ℓ. We have already proved $M = S_\ell$, which is the first itemized statement. We have also already proved, for any $k > \ell$ and any j, the existence of some $r_{k,j}$ of index at most ℓ such that $\partial_t^k e_j \equiv r_{k,j} \pmod{J(t)}$. This also holds by the definition of the index for $k \le \ell$. Now, any $a \in W_{t,\mathbf{x}}^s$ writes in the form $\sum_{k,j} c_{k,j} \partial_t^k e_j$ for coefficients $c_{k,j} \in W_{\mathbf{x}}(t)$. Taking linear combinations of congruences modulo the $W_{\mathbf{x}}(t)$-module $J(t)$, we obtain $a \equiv \sum_{k,j} c_{k,j} r_{k,j} \pmod{J(t)}$, and as a consequence,

$$\operatorname{ind}_{J(t)}(a) \le \operatorname{ind}\left(\sum_{k,j} c_{k,j} r_{k,j}\right) \le \ell,$$

where the first inequality is by (3.17) and the second by the definition of the index as a degree. We have proved the second itemized statement. □

An algorithm for computing the smallest ℓ as in the statement above follows directly from (3.17), simply by testing increasing values of ℓ.

Now that we have a finite generating set for M, it remains to characterize the relations between the generators. To this end, for the rest of the section we fix ℓ as provided by Lemma 55 and we let J_ℓ be the sub-$W_{\mathbf{x}}(t)$-module of $W_{t,\mathbf{x}}(t)^s$ generated by

$$\left\{\partial_t^k g \;\middle|\; g \in G \text{ and } k + \operatorname{ind}(g) \leq \ell\right\}.$$

It is, by construction, a submodule of $W_{\mathbf{x}}(t)B_\ell$.

Lemma 56. *The inclusion $W_{\mathbf{x}}(t)B_\ell \to W_{t,\mathbf{x}}(t)^s$ induces an isomorphism*

$$M \simeq \frac{W_{\mathbf{x}}(t)B_\ell}{J_\ell},$$

with inverse induced by the map $W_{t,\mathbf{x}}(t)^s \to W_{\mathbf{x}}(t)B_\ell$ given by $a \mapsto \operatorname{LRem}(a, G)$.

Proof. First, $J_\ell \subseteq J(t)$, so the inclusion $W_{\mathbf{x}}(t)B_\ell \to W_{t,\mathbf{x}}(t)^s$ induces a morphism of $W_{\mathbf{x}}(t)$-modules

$$\phi : W_{\mathbf{x}}(t)B_\ell/J_\ell \to W_{t,\mathbf{x}}(t)^s/J(t).$$

Next, the $\mathbb{K}(t)$-linear map $a \in W_{t,\mathbf{x}}(t)^s \mapsto \operatorname{LRem}(a, G)$ has values in $W_{\mathbf{x}}(t)B_\ell$, because $\operatorname{ind}(\operatorname{LRem}(a, G)) = \operatorname{ind}_{J(t)}(a) \leq \ell$, by Lemma 55. This map vanishes on $J(t)$, because G is a Gröbner basis of $J(t)$, so it induces a $\mathbb{K}(t)$-linear map

$$\psi : W_{t,\mathbf{x}}(t)^s/J(t) \to W_{\mathbf{x}}(t)B_\ell/J_\ell.$$

The maps $\phi \circ \psi$ and $\psi \circ \phi$ are both induced by $a \mapsto \operatorname{LRem}(a, G)$. The first is the identity on $W_{t,\mathbf{x}}(t)^s/J(t)$ because for all $a \in W_{t,\mathbf{x}}^s$, $a \equiv \operatorname{LRem}(a, G) \pmod{J(t)}$ as a property of the Gröbner basis G. The second is the identity on $W_{\mathbf{x}}(t)B_\ell/J_\ell$ because of the elimination property: the computation of $\operatorname{LRem}(a, G)$ only involves multiples of G of index at most $\operatorname{ind}(a)$, which are all in J_ℓ, so that for all a of index at most ℓ, $a \equiv \operatorname{LRem}(a, G) \pmod{J_\ell}$. This shows that ϕ is an isomorphism. □

At this point we are able to define the wanted dimension r and module S. In view of Lemma 56, set r to $(\ell + 1)s$, so as to have a trivial isomorphism $W_{\mathbf{x}}(t)B_\ell \simeq W_{\mathbf{x}}(t)^r$. Call S the image of the submodule J_ℓ of $W_{\mathbf{x}}(t)B_\ell$ under this isomorphism, so that, summarizing,

$$\frac{W_{t,\mathbf{x}}^s}{J(t)} \simeq M \simeq \frac{W_{\mathbf{x}}(t)B_\ell}{J_\ell} \simeq \frac{W_{\mathbf{x}}(t)^r}{S}.$$

It remains to describe an endomorphism L of $W_{\mathbf{x}}(t)^r$ such that

$$\partial_t \cdot \operatorname{pr}_S(a) = \operatorname{pr}_S\left(\tfrac{\partial a}{\partial t} + L(a)\right),$$

for any $a \in W_{\mathbf{x}}(t)^r$. Introduce the canonical maps $\operatorname{pr}_{J(t)}$ and pr_S to the relevant quotients. Recall that ∂_t is defined for any h in $W_{t,\mathbf{x}}(t)^s/J(t)$ by left-multiplication by ∂_t:

$$\partial_t \cdot \operatorname{pr}_{J(t)}(h) = \operatorname{pr}_{J(t)}(\partial_t h).$$

Therefore, the isomorphism of Lemma 56 transfers ∂_t on $W_{\mathbf{x}}(t)B_\ell/J_\ell$ as

$$\partial_t \cdot \mathrm{pr}_{J_\ell}(a) = \mathrm{pr}_{J_\ell}\left(\mathrm{LRem}(\partial_t a, G)\right). \tag{3.18}$$

Lastly, take $a \in W_{\mathbf{x}}(t)B_\ell$ and write it $a = \sum_{i=1}^n a_i e_i$. The Leibniz rule in $W_{t,\mathbf{x}}^s$ gives $\partial_t a = \frac{\partial a}{\partial t} + \sum_{i=1}^n a_i \partial_t e_i$. By linearity of LRem and since $W_{\mathbf{x}}(t)B_\ell$ is stable under the coefficient-wise differentiation $\frac{\partial}{\partial t}$, we obtain

$$\begin{aligned} \mathrm{LRem}(\partial_t a, G) = \mathrm{LRem}(\tfrac{\partial a}{\partial t}, G) + \mathrm{LRem}\left(\sum_{i=1}^n a_i \partial_t e_i, G\right) \\ = \tfrac{\partial a}{\partial t} + \mathrm{LRem}\left(\sum_{i=1}^n a_i \partial_t e_i, G\right) + h \end{aligned}$$

for some $h \in J_\ell$, then, upon applying pr_{J_ℓ} and combining with (3.18),

$$\partial_t \cdot \mathrm{pr}_{J_\ell}(a) = \mathrm{pr}_{J_\ell}\left(\tfrac{\partial a}{\partial t} + \mathrm{LRem}\left(\sum_{i=1}^n a_i \partial_t e_i, G\right)\right).$$

So, the endomorphism L we want is obtained by transferring the endomorphism

$$\sum_{i=1}^n a_i e_i \mapsto \mathrm{LRem}\left(\sum_{i=1}^n a_i \partial_t e_i, G\right)$$

of $W_{\mathbf{x}}(t)B_\ell$ to an endomorphism of $W_{\mathbf{x}}(t)^r$.

3.5 Implementation

This section outlines specific algorithmic and implementation choices made in our Julia implementation of the algorithms presented in this paper. We begin in Section 3.5.1 with a general presentation of our package. In Section 3.5.2 we present two ideas based on memoization and on the use of a tracer to efficiently compute the image of the reduction map $[.]_\eta$ on a finite-dimensional space. Lastly, we present in Section 3.5.3 a modified version of Algorithm 4 that utilizes the evaluation/interpolation paradigm to avoid the growth of intermediate coefficients. This is of particular importance when reducing operators with coefficients in $\mathbb{K}(t)$ using the reduction $[.]_\eta$.

3.5.1 General comments

The algorithm described in Algorithm 5 has been implemented in the Julia package `MultivariateCreativeTelescoping.jl`[1]. The package includes an implementation of Weyl algebras in which operators have a sparse representation by a pair of vectors, one

[1] See `https://hbrochet.github.io/MultivariateCreativeTelescoping.jl/`.

for exponents and one for the corresponding coefficients. The currently supported coefficient fields are the field $\mathbb{Q}$ of rational numbers, the finite fields $\mathbb{F}_p$ with $p \leq 2^{31}$, and extensions of those fields with symbolic parameters. When such parameters are present, the implementation interfaces with FLINT for defining and manipulating commutative polynomials, by means of the Julia packages `AbstractAlgebra.jl` and `Nemo.jl`. Our package also provides an implementation of non-commutative generalizations of algorithms for computing Gröbner bases: Buchberger's algorithm (see e.g. [5]), the F4 algorithm [5], and the F5 algorithm [126, 87]. The F4 implementation seems to be the most efficient one for grevlex orders while the F5 implementation seems to be the most efficient one for block and lexicographical orders.

3.5.2 Efficient computation of the reduction map $[.]_\eta$ on a finite-dimensional vector space

We first explain how to compute efficiently the sequence $(g_i)_{i\geq 0}$ that is contained in a finite-dimensional vector space obtained by an effective confinement. We next discuss the use of a tracer to compute the vector space $E_{\preccurlyeq\eta}$. We finally provide comments on the dimension of $E_{\preccurlyeq\eta}$ for various examples.

Computation of the sequence $(g_i)_i$ using memoization.

The confinement (η, B) required by Algorithm 4 is constructed so that the sequence $(g_i)_i$ defined in (3.14) is contained in $\mathrm{Span}_{\mathbb{K}(t)}(B)$. Properties of the reduction $[.]_\eta$ imply a refined formula: after decomposing g_i as a sum $\sum_{m\in B} a_m m$ with $a_m \in \mathbb{K}(t)$, g_{i+1} can be obtained by

$$g_{i+1} = \sum_{m\in B} \left(\frac{\partial a_i}{\partial t} m + a_m [L(m)]_\eta \right). \tag{3.19}$$

Early in the execution of Algorithm 4, when it computes the confinement (η, B) by Algorithm 3, the image $[L(m)]_\eta$ of every monomial $m \in B$ has to be computed. We therefore choose to store these images in memory to allow for a more efficient computation of the sequence $(g_i)_i$ by using (3.19) at a later stage.

The vector space $E_{\preccurlyeq\eta}$.

In Algorithm 2, a generating set of the vector space $E_{\preccurlyeq\eta}$ is computed by reducing for each $\eta' \preccurlyeq \eta$ a term of the form $mg - \partial_i w$ satisfying $\eta' = \mathrm{lm}(mg) = \mathrm{lm}(\partial_i w)$. However, not all such terms contribute to an increase in the dimension of the space, as their reductions may be linearly dependent. Since such reductions are repeated multiple times for different primes p and evaluation points of t (see the next subsection), we use a tracer [132] during the computation for the first pair (p, t) to record all η' corresponding to a non-contributing term and skip the corresponding terms in subsequent computations. Assuming that the first pair (p, t) is not unlucky, we know that all the skipped pairs would also lead to unnecessary elements if used in later computations.

Dimension of the vector space $E_{\preccurlyeq\eta}$.

By the finite dimensionality of $M/\boldsymbol{\partial}M$, only finitely many monomials of $M_{\mathbf{x},r}$ are irreducible modulo $S+\boldsymbol{\partial}W_{\mathbf{x}}^r$. As a consequence, every monomial $m \in M_{\mathbf{x},r}$ except for the finitely many irreducible ones is either reducible by G or by an echelon form of $E_{\preccurlyeq\eta}$ for some η. If an infinite number of monomials is not reducible by G, as in Example 34, the dimension of $E_{\preccurlyeq\eta}$ will tend to infinity when η increases indefinitely, making the computation of $E_{\preccurlyeq\eta}$ increasingly expensive. As a consequence, the computational cost of Algorithm 2 depends on the structure of the staircase formed by the leading monomials of G. We present two extreme scenarios: in one, $E_{\preccurlyeq\eta}$ is equal to $\{0\}$ for any η and in the other, no monomial of $\mathbb{K}[\mathbf{x}]$ is reducible by G. Naturally, intermediate cases also exist.

Example 57. Let S be the left ideal of $W_{\mathbf{x}}(t)$ generated by the Gröbner basis

$$(t-1)\underline{x_1} - t\partial_1, \qquad \underline{x_2} - t.$$

Up to renaming variables, this is the left ideal used for the computation of the generating series of 2-regular graphs in Section 3.6. Every operator in this Gröbner basis has its leading monomial in $\mathbb{K}[\mathbf{x}]$, therefore Theorem 37 implies that $E_{\preccurlyeq\eta}$ is $\{0\}$ for any η. In this very special case we obtain that the reduction $[.]$ computes normal forms. That is, $[a]=0$ if and only if $a \in S+\boldsymbol{\partial}W_{\mathbf{x}}^r$. We observed the same phenomenon with the ideals S defined in Theorem 60 for k-regular graphs up to $k=8$.

Example 58. Let $f(\mathbf{x},t) = 1-(1-x_1x_2)x_3 - tx_1x_2x_3(1-x_1)(1-x_2)(1-x_3)$ and set the context for $n=3$. The integral of $1/f$ is related to the generating function of Apéry numbers [14]. We were able to compute, by a method that we do not describe here, a Gröbner basis for the grevlex order of a $W_{\mathbf{x}}(t)$-ideal S included in $\operatorname{ann}(1/f)$ such that $W_{\mathbf{x}}(t)/S$ is holonomic. This Gröbner basis contains 26 operators but all of their leading monomials contain a ∂_i. Hence, no monomial of $\mathbb{K}[\mathbf{x}]$ is reducible by the Gröbner basis of S. We observed the same phenomenon for every rational function that we tried.

Lastly, we present the simplest example on which our reduction $[.]_\eta$ is inefficient.

Example 59. We continue Example 34, in which we set S to $W_{x_1}\partial_1$ and $\preccurlyeq$ to the lexicographic order $\partial_1 \preccurlyeq x_1$. We remarked that $E \subset \mathbb{K}[x_1]$, and reciprocally the equality $(i+1)x_1^i = \partial_1 x_1^{i+1} - x_1^{i+1}\partial_1$ proves that $\mathbb{K}[x_1] \subset E$. The reduction $[.]$ does not see that elements of $\mathbb{K}[x_1]$ are reducible by $S+\boldsymbol{\partial}W_{x_1}$ and Algorithm 2 ends up calculating a basis of $\mathbb{K}[x_1]_{\preccurlyeq\eta}$.

3.5.3 Modular methods and evaluation/interpolation

In this subsection, we fix $\mathbb{K}=\mathbb{Q}$. We first recall the principle of the evaluation and interpolation paradigm, then we present a modified version of Algorithm 4 that incorporates this paradigm.

The principle.

Let $r_1,\ldots,r_\ell \in \mathbb{Q}(t)$ be rational functions and let $F : \mathbb{Q}(t)^\ell \to \mathbb{Q}(t)$ be a function computable using only additions, multiplications, and inversions. The rational function $F(r_1, \ldots, r_\ell)$ can be reconstructed as an element of $\mathbb{Q}(t)$ from its evaluations $F(r_1(a_i), \ldots, r_\ell(a_i)) \bmod p_j$ at several points $a_i \in \mathbb{F}_{p_j}$ and for several prime integers p_j in three steps:

1. for each j, reconstruct $F(r_1, \ldots, r_\ell) \bmod p_j$ in $\mathbb{F}_{p_j}(t)$ by Cauchy interpolation [140, Chapter 5.8],
2. reconstruct $F(r_1, \ldots, r_\ell) \bmod N$ in $\mathbb{F}_N(t)$ where $N = \prod_j p_j$ by the Chinese remainder theorem [140, Chapter 5.4],
3. lift the integer coefficients of $F(r_1, \ldots, r_\ell) \bmod N$ in $\mathbb{Q}$ by rational reconstruction [140, Chapter 5.10].

The first (resp. third) step requires a bound on the degree in t of the result (resp. on the size of its coefficients) to determine the number of evaluations needed to ensure correctness. Since we do not have access to such bounds, we rely instead on a probabilistic approach: after successfully obtaining a result with a certain number of evaluations in step 1 (resp. 3), we compute one additional evaluation and check consistency with the previously obtained result.

Finally, this method can fail if some bad evaluation points or some bad primes are chosen. However, these situations are very rare, provided the prime numbers are sufficiently large and the evaluation points are selected uniformly at random over $\mathbb{F}_p$.

Implementation.

The evaluation/interpolation scheme described above cannot be directly applied to Algorithm 4 as it involves not only additions, multiplications and inversions, but also differentiations. Indeed, recall that the sequence $(g_i)_i$ is defined for $\eta \in M_{\mathbf{x},r}$ by $g_0 = [f]_\eta$ and

$$g_{i+1} = \frac{\partial g_i}{\partial t} + [L(g_i)]_\eta$$

where $\partial g/\partial t$ denotes the coefficient-wise differentiation of g in the basis $M_{\mathbf{x},r}$. This differentiation does not commute with the evaluation of t, which prevents the sequence $(g_i)_i$ from being computed by evaluation/interpolation.

Algorithm 5 computes the matrix of the $\mathbb{K}(t)$-linear map $[L(.)]_\eta$ using evaluation/interpolation, and uses it to find a linear relation among the elements of the sequence $(g_i)_i$. The reconstruction of the coefficients from $\mathbb{F}_p(t)$ to $\mathbb{Q}(t)$ is delayed to the end of the algorithm. We adopt the following conventions: the projection of an element $g \in W_{\mathbf{x}}(t)^r$ in $\mathbb{F}_p \otimes_{\mathbb{Z}} W_{\mathbf{x}}^r$ by evaluation at a point a and reduction modulo p is stored in a variable $\tilde{g}$ and its projection in $\mathbb{F}_p \otimes_{\mathbb{Z}} W_{\mathbf{x}}^r(t)$ by reduction modulo p is stored in a variable $\bar{g}$.

Algorithm 5 Integration with evaluation/interpolation

Input:

- a holonomic module $W_{\mathbf{x}}(t)^r/S$
- a derivation map $\partial_t : W_{\mathbf{x}}(t)^r/S \to W_{\mathbf{x}}(t)^r/S$ given by the matrix of an endomorphism L, as in (3.12)
- an element $f \in W_{\mathbf{x}}(t)^r$
- an integer $\rho \geq 0$

Output:

- $P = c_0 + \cdots + c_N\partial_t^N$ such that $c_i \in \mathbb{Q}(t)$, $c_N \neq 0$ and $P \cdot \mathrm{pr}_S(f) \in \boldsymbol{\partial} M$

1 **for** large random prime numbers $p = p_1, p_2, \ldots$
2   $\bar{L} \leftarrow$ the endomorphism of $\mathbb{F}_p \otimes_{\mathbb{Z}} W_{\mathbf{x}}^r(t)$ induced by L via the reduction modulo p
3   **for** random numbers $a = a_1, a_2, \ldots$ in $\mathbb{F}_p$
4     $\tilde{S} \leftarrow$ image of S under evaluation at $t = a$ and reduction modulo p
5     $\tilde{L} \leftarrow$ the endomorphism of $\mathbb{F}_p \otimes_{\mathbb{Z}} W_{\mathbf{x}}^r$ induced by $\bar{L}$ via the same evaluation
6     $(\eta, B) \leftarrow$ an effective confinement obtained from $(\tilde{S}, \tilde{f}, \tilde{L}, \rho)$ by Algorithm 3 over $\mathbb{K} = \mathbb{F}_p$
7     $\tilde{g}_0 \leftarrow [\tilde{f}]_\eta$ where the reduction is over $\mathbb{K} = \mathbb{F}_p$
8   store $\tilde{g}_0$ as well as the images $[\tilde{L}(m)]_\eta$ that have been computed at line 6 for all $m \in B$
9   interpolate the coefficients of $\bar{g}_0$ and of each $[\bar{L}(m)]_\eta$ by elements of $\mathbb{F}_p(t)$
10   $N \leftarrow 0$
11   **while** $\bar{g}_0, \ldots, \bar{g}_N$ are linearly independent over $\mathbb{F}_p(t)$
12     $\bar{g}_{N+1} \leftarrow \frac{\partial \bar{g}_N}{\partial t} + \left[\bar{L}(\bar{g}_N)\right]_\eta$ where the reduction is over $\mathbb{K} = \mathbb{F}_p(t)$
13     $N \leftarrow N + 1$
14   store coefficients of the minimal non-trivial relation $\bar{c}_0\bar{g}_0 + \cdots + \bar{c}_N\bar{g}_N = 0$ for $\bar{c}_i \in \mathbb{F}_p(t)$
15 reconstruct the coefficients $c_0, \ldots, c_N$ in $\mathbb{Q}(t)$ from their values in the $\mathbb{F}_{p_j}(t)$
16 **return** $c_0 + \cdots + c_N\partial_t^N$

3.6 Application to the computation of ODEs satisfied by k-regular graphs

In this section, we illustrate our new algorithm with computations on a family of multivariate integrals of combinatorial origin. We compute linear ODEs satisfied by various models of k-regular graphs and generalizations. A distinguishing feature of these integration problems is that they cannot be solved by classical creative telescoping algorithms, which perform computations over the field of rational functions in all variables: because the objects to be integrated have polynomial torsion, they are not functions, and such calculations would erroneously result in a zero integral.

ODEs for $k \leq 5$ were obtained 20 years ago by naive linear algebra and elimination by Euclidean divisions [51]. This has recently been extended to $k \leq 7$ by a multivariate analog of the reduction-based algorithm [20] in which the reduction is modulo the polynomial image of several differential operators. Here we use our new algorithm to achieve $k = 8$.

3.6.1 Statement of the integration problem

We first briefly introduce the problem and its solution by an integral representation. We refer the reader to [51] for further motivation, history, and details. Given a fixed integer $k \geq 2$, a k-regular graph is a graph whose vertices all have degree k, that is, all have exactly k neighbors. We are interested in the enumerative generating function

$$R_k(t) = \sum_{n\geq 0} r_{k,n} \frac{t^n}{n!},$$

where $r_{k,n}$ is the number of k-regular labeled graphs on n vertices. It is well known that the generating function $R_k(t)$ satisfies a linear ODE with polynomial coefficients in t.

In this section, we show how such a differential equation can be obtained from Algorithm 4. To this end, we use the classical formulation of $R_k(t)$ as a scalar product of two exponentials,

$$R_k(t) = \langle e^f, e^{tg} \rangle,$$

where f and g are explicit polynomials in indeterminate $p_1, \dots, p_k$ [50, Algorithm 1, Step a.], and where the scalar product is a classic tool in the combinatorics of symmetric functions. The scalar product is first defined on monomials by

$$\langle p_1^{r_1} \cdots p_k^{r_k}, p_1^{s_1} \cdots p_k^{s_k} \rangle = z_{\mathbf{r}} \delta_{\mathbf{r},\mathbf{s}} \qquad \text{for} \qquad z_{\mathbf{r}} = r_1!\, 1^{r_1} r_2!\, 2^{r_2} \cdots r_k!\, k^{r_k}. \tag{3.20}$$

With $z_{\mathbf{r}}$ indexed by exponents in $\mathbb{N}^k$, we depart from the equivalent indexing z_λ by partitions $\lambda_1 \geq \lambda_2 \geq \dots$ that is used classically as well as in [50]. The scalar product is then extended by bilinearity to left arguments in $\mathbb{Q}[[\mathbf{p}]]$ and right arguments in $\mathbb{Q}[\mathbf{p}]((t))$, making the scalar product live in $\mathbb{Q}((t))$. Note that, because in the symmetric-function theory the power function p_i denotes the sum $x_1^i + x_2^i + \cdots$, we use the more traditional $\mathbf{p}$ instead of $\mathbf{x}$ for the variables, in accordance with existing literature. Also, by $\mathbb{Q}[\mathbf{p}]((t))$ we mean the ring of formal sums with coefficients in $\mathbb{Q}[\mathbf{p}]$, with finitely many exponents towards $-\infty$ and potentially infinitely many towards $+\infty$. This is not a field. The use of Algorithm 4 is justified by the following statement.

Theorem 60. *Let $f, g \in \mathbb{Q}[\mathbf{p}]$. Let S be the left ideal of $W_{\mathbf{p}}(t)$ generated by*

$$p_i - t\frac{\partial g}{\partial p_i}(u_1, \dots, u_k),$$

for $1 \leq i \leq k$, where $u_i = i\big(\frac{\partial f}{\partial p_i} - \partial_i\big)$. Then, $W_{\mathbf{p}}(t)/S$ is holonomic as a $W_{\mathbf{p}}(t)$-module. Write pr_S for the canonical projection $\mathrm{pr}_S : W_{\mathbf{p}}(t) \to W_{\mathbf{p}}(t)/S$. Then, $W_{\mathbf{p}}(t)/S$ can be endowed with a derivation ∂_t commuting with $\mathbf{p}$ and $\boldsymbol{\partial}_{\mathbf{p}}$ satisfying

$$\partial_t \cdot \mathrm{pr}_S(a) = \mathrm{pr}_S\left(\tfrac{\partial a}{\partial t} + a g(u_1, \dots, u_k)\right). \tag{3.21}$$

On input the module $W_{\mathbf{p}}(t)/S$ (for $r = 1$), the derivation (3.21) (for the implied endomorphism $L : a \mapsto a g(u)$), the element $f = 1 \in W_{\mathbf{p}}(t)$, and any $\rho \geq 0$, Algorithm 4 outputs a nonzero differential operator $P(t, \partial_t)$ such that

$$P(t, \partial_t) \cdot \langle e^f, e^{tg} \rangle = 0.$$

The rest of the section is a proof of this statement. In particular, we fix the ideal S as in theorem, and the holonomy of $W_{\mathbf{p}}(t)/S$ will be proven as Lemma 64, the existence of the derivation ∂_t will be proven as Lemma 62, and the correctness Theorem 52 will prove the final result on the operator P.

We begin with a few preliminary definitions. First, given $h = h(p_1, \ldots, p_k) \in \mathbb{Q}[\mathbf{p}]$, we write $\tilde{h}$ for $h(1p_1, 2p_2, \ldots, kp_k)$. Second, we define a formal Laplace transform on monomials by

$$\mathcal{L}(p_1^{r_1} \ldots p_k^{r_k}) = r_1!\, p_1^{-r_1-1} \ldots r_k!\, p_k^{-r_k-1} \tag{3.22}$$

and extend it by linearity into a map from $\mathbb{Q}[\mathbf{p}]$ to $\mathbb{Q}[\mathbf{p}^{-1}]$. Third, we introduce the ring

$$\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle := \mathbb{Q}[[\mathbf{p}]][\mathbf{p}^{-1}] = \bigcup_{\ell \geq 0} (p_1 \cdots p_k)^{-\ell} \mathbb{Q}[[\mathbf{p}]]$$

and a formal residue on it by the formula

$$\operatorname{Res}\left(\sum_{\mathbf{r} \in \mathbb{Z}^k} c_{\mathbf{r}} \mathbf{p}^{\mathbf{r}} \right) = c_{-1,\ldots,-1}. \tag{3.23}$$

Lastly, we extend $h \mapsto \tilde{h}$ to a map from $\mathbb{Q}[\mathbf{p}]((t))$ to itself, $\mathcal{L}$ to a map from $\mathbb{Q}[\mathbf{p}]((t))$ to $\mathbb{Q}[\mathbf{p}^{-1}]((t))$, and Res to a map from $\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle((t))$ to $\mathbb{Q}((t))$, by making each of those maps act coefficient-wise.

Lemma 61. *For any polynomials f and g in $\mathbb{Q}[\mathbf{p}]$,*

$$\langle e^f, e^{tg} \rangle = \operatorname{Res}(e^f \mathcal{L}(e^{t\tilde{g}})).$$

Proof. For $U \in \mathbb{Q}[[\mathbf{p}]]$ and for $\mathbf{r} \in \mathbb{N}^k$, $U\mathcal{L}(\tilde{\mathbf{p}}^{\mathbf{r}})$ is an element of $\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle$, so that using (3.22), (3.20), and (3.23) in order, we derive:

$$\begin{aligned} \operatorname{Res}(U\mathcal{L}(\tilde{\mathbf{p}}^{\mathbf{r}})) &= \operatorname{Res}\left(U \times 1^{r_1} \cdots k^{r_k} r_1!\, p_1^{-r_1-1} \cdots r_k!\, p_k^{-r_k-1} \right) \\ &= z_{\mathbf{r}} \operatorname{Res}(U p_1^{-r_1-1} \cdots p_k^{-r_k-1}) = z_{\mathbf{r}} \times [\mathbf{p}^{\mathbf{r}}]U = \langle U, \mathbf{p}^{\mathbf{r}} \rangle. \end{aligned}$$

This formula extends by linearity to $\langle U, h \rangle = \operatorname{Res}(U\mathcal{L}(\tilde{h}))$ for any $h \in \mathbb{Q}[\mathbf{p}]$. Upon specializing to $U = e^f$ and $h = g^n/n!$ before taking series in t, this makes the informal integral formula provided in [51, end of 7.1] completely algebraic, in the form of the formula

$$\langle e^f, e^{tg} \rangle = \sum_{\ell \geq 0} \langle e^f, g^\ell \rangle \frac{t^\ell}{\ell!} = \sum_{\ell \geq 0} \operatorname{Res}\left(e^f \mathcal{L}(\tilde{g}^\ell) \right) \frac{t^\ell}{\ell!} = \operatorname{Res}\left(e^f \mathcal{L}(e^{t\tilde{g}}) \right). \qquad \square$$

The ring $\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle((t))$ is a $W_{\mathbf{p}}(t)$-module with the usual actions: p_i acts by multiplication and ∂_i by partial differentiation with respect to p_i. Let K be the subspace of all elements of $\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle((t))$ that do not contain any monomial $\mathbf{p}^{\mathbf{r}} t^m$ with $r_1, \ldots, r_k$ all negative. In other words,

$$K = \sum_{i=1}^{k} \mathbb{Q}[[\mathbf{p}]][p_1^{-1}, \ldots, p_{i-1}^{-1}, p_{i+1}^{-1}, \ldots, p_k^{-1}]((t)).$$

This subspace has the property to be a sub-$W_{\mathbf{p}}(t)$-module of $\mathbb{Q}\langle\langle\mathbf{p}\rangle\rangle((t))$ and to be contained in the kernel of the residue map $\mathbb{Q}\langle\langle\mathbf{p}\rangle\rangle((t)) \to \mathbb{Q}((t))$.

Now, let f and g be two polynomials in $\mathbb{Q}[\mathbf{p}]$. We provide in Lemma 62 an explicit construction of a derivation ∂_t satisfying (3.21). We remark that this construction is simpler than the general approach presented in Section 3.4.3.

Lemma 62. *The $W_{\mathbf{x}}(t)$-linear map $L : a \mapsto a\,\tilde{g}(\frac{\partial f}{\partial p_1} - \partial_1, \ldots, \frac{\partial f}{\partial p_k} - \partial_k)$ defines a derivation on $W_{\mathbf{x}}(t)/S$ by*

$$\partial_t \cdot \mathrm{pr}_S(a) = \mathrm{pr}_S\left(\tfrac{\partial a}{\partial t} + L(a)\right). \tag{3.24}$$

This derivation commutes with $\mathbf{p}$ and $\boldsymbol{\partial}_{\mathbf{p}}$.

Proof. Let ϕ be the $W_{\mathbf{p}}$-linear endomorphism of $W_{\mathbf{p}}(t)$ defined by $\phi(a) = \frac{\partial a}{\partial t} + L(a)$. To show that ∂_t is well-defined, it suffices to verify that $\phi(S) \subseteq S$. Consider the generators

$$s_i := p_i - t\frac{\partial \tilde{g}}{\partial p_i}\left(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k\right)$$

of S. Using the commutation rule $p_i(\frac{\partial f}{\partial p_j} - \partial_j) = (\frac{\partial f}{\partial p_j} - \partial_j)p_i + \delta_{i,j}$, one obtains for any polynomial $q(X_1, \ldots, X_k)$

$$\begin{aligned} &p_i\, q(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k) \\ &\qquad = q(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k)\, p_i + \tfrac{\partial q}{\partial X_i}(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k). \end{aligned}$$

So, for each i, a computation shows that

$$\phi(s_i) = \tilde{g}(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k)\, s_i \in S.$$

Next, for any rational function $R(t)$ and any i, we check

$$\phi(R(t)s_i) = R(t)\phi(s_i) + R'(t)s_i \in S,$$

by $\mathbb{Q}(t)$-linearity of L. Now, S is generated as a $W_{\mathbf{x}}$-module by the family $(R(t)s_i)_{R,i}$, so, by $W_{\mathbf{x}}$-linearity of L, and thus of ϕ, we get the inclusion $\phi(S) \subseteq S$. The $W_{\mathbf{x}}(t)$-linearity of L and the definition (3.24) imply, for any $R(t) \in \mathbb{Q}(t)$,

$$\begin{aligned} \partial_t R(t) \cdot \mathrm{pr}_S(a) = \partial_t \cdot \mathrm{pr}_S(R(t)a)) = \mathrm{pr}_S\left(\tfrac{\partial(R(t)a)}{\partial t} + R(t)L(a)\right) = \\ R(t)\partial_t \cdot \mathrm{pr}_S(a) + R'(t)\,\mathrm{pr}_S(a). \end{aligned}$$

In other words, ∂_t is a derivation. A similar but simpler calculation shows that it commutes with $\mathbf{p}$ and $\boldsymbol{\partial}_{\mathbf{p}}$. □

For the same polynomials f and g, let $\Xi_{f,g}$ be the class of $e^f \mathcal{L}(e^{t\tilde{g}})$ modulo K.

Lemma 63. *For any $a \in S$, the relation $a \cdot \Xi_{f,g} = 0$ holds. Moreover,*

$$\tfrac{\partial}{\partial t} \cdot \Xi_{f,g} = \tilde{g}(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k) \cdot \Xi_{f,g}.$$

Proof. The definition (3.22) of the formal Laplace transform implies the following formulas, valid for $\mathbf{r} \in \mathbb{N}^k$:

$$\mathcal{L}(p_i \cdot \mathbf{p}^{\mathbf{r}}) = (r_i + 1)p_i^{-1}\mathcal{L}(\mathbf{p}^{\mathbf{r}}) = -\partial_i \cdot \mathcal{L}(\mathbf{p}^{\mathbf{r}}), \tag{3.25}$$

$$\mathcal{L}(\partial_i \cdot \mathbf{p}^{\mathbf{r}}) = \mathcal{L}(r_i p_i^{-1}\mathbf{p}^{\mathbf{r}}) = \begin{cases} p_i \cdot \mathcal{L}(\mathbf{p}^{\mathbf{r}}), & \text{if } r_i \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.26}$$

In turn, for $h \in \mathbb{Q}[\mathbf{p}]$, this implies the formulas

$$\mathcal{L}(p_i \cdot h) = -\partial_i \cdot \mathcal{L}(h), \tag{3.27}$$

$$\mathcal{L}(\partial_i \cdot h) = p_i \cdot \mathcal{L}(h) - p_i \cdot \mathcal{L}(h|_{p_i=0}). \tag{3.28}$$

The last formula is not convenient because of the term involving $h|_{p_i=0}$. Fortunately, this term is in K, so we have the nicer formula

$$\mathcal{L}(\partial_i \cdot h) \equiv p_i \cdot \mathcal{L}(h) \pmod{K}. \tag{3.29}$$

Moreover, we have for any $h \in \mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle((t))$,

$$(\tfrac{\partial f}{\partial p_i} - \partial_i) \cdot e^f h = -e^f \partial_i \cdot h \tag{3.30}$$

Therefore, we have

$$\begin{aligned} &\left(p_i - t\tfrac{\partial \tilde{g}}{\partial p_i}(\tfrac{\partial f}{\partial p_1} - \partial_1, \ldots, \tfrac{\partial f}{\partial p_k} - \partial_k)\right) \cdot e^f \mathcal{L}(e^{t\tilde{g}}) \qquad (3.31)\\ &\quad = e^f \left(p_i - t\tfrac{\partial \tilde{g}}{\partial p_i}(-\partial_1, \ldots, -\partial_k)\right) \cdot \mathcal{L}(e^{t\tilde{g}}), \qquad \text{using (3.30)} \\ &\quad \equiv e^f \mathcal{L}\left(\left(\partial_i - t\tfrac{\partial \tilde{g}}{\partial p_i}\right) \cdot e^{t\tilde{g}}\right) \qquad \text{using (3.27) and (3.29)} \\ &\quad \equiv e^f \mathcal{L}(0) \equiv 0 \pmod{K}. \end{aligned}$$

This proves the first statement about all $a \in S$ by $W_{\mathbf{p}}(t)$-linearity. The second statement is proved similarly, starting with $\frac{\partial}{\partial t} - \tilde{g}(\frac{\partial f}{\partial p_1} - \partial_1, \ldots, \frac{\partial f}{\partial p_k} - \partial_k)$ instead of the operator in (3.31). □

Lemma 64. *The $W_{\mathbf{p}}(t)$-module $W_{\mathbf{p}}(t)/S$ is holonomic.*

Proof. Let τ be the automorphism of the $\mathbb{Q}(t)$-algebra $W_{\mathbf{p}}(t)$ defined by

$$\tau(p_i) = \tfrac{\partial f}{\partial p_i} - \partial_i \quad \text{and} \quad \tau(\partial_i) = p_i - t\tau\left(\tfrac{\partial \tilde{g}}{\partial p_i}\right).$$

(Note that the definition is not recursive since $\frac{\partial \tilde{g}}{\partial p_i}$ is a polynomial in $\mathbf{p}$. Also, p_i commutes with $\frac{\partial \tilde{g}}{\partial p_i}$, making their images under τ commute as well. This justifies $\tau(\partial_i)\tau(p_i) = \tau(p_i)\tau(\partial_i) + 1$ to have an algebra morphism.) The inverse morphism is given by

$$\tau^{-1}(p_i) = \partial_i + t\tfrac{\partial \tilde{g}}{\partial p_i} \quad \text{and} \quad \tau^{-1}(\partial_i) = \tau^{-1}\left(\tfrac{\partial f}{\partial p_i}\right) - p_i.$$

By definition, S is generated by $\tau(\partial_1), \dots, \tau(\partial_k)$. Therefore, as $W_{\mathbf{p}}(t)$-modules,

$$W_{\mathbf{p}}(t)/S \simeq W_{\mathbf{p}}(t)/(W_{\mathbf{p}}(t)\partial_1 + \cdots + W_{\mathbf{p}}(t)\partial_k) \simeq \mathbb{Q}(t)[\mathbf{p}].$$

Since the latter is holonomic, so is $W_{\mathbf{p}}(t)/S$. □

Let M denote the holonomic module $W_{\mathbf{p}}(t)/S$. All in all, we have a commutative diagram of $\mathbb{Q}(t)$-linear spaces, with all arrows commuting with the derivation ∂_t

$$\begin{array}{ccccc} M & \longrightarrow & W_{\mathbf{p}}(t) \cdot \Xi_{f,g} & \hookrightarrow & \frac{\mathbb{Q}\langle\langle \mathbf{p} \rangle\rangle((t))}{K} \\ \downarrow & & \downarrow \scriptstyle{\text{Res}} & & \downarrow \scriptstyle{\text{Res}} \\ \frac{M}{\boldsymbol{\partial} M} & \longrightarrow & W_{\mathbf{p}}(t) \cdot \langle e^f, e^{tg} \rangle & \hookrightarrow & \mathbb{Q}((t)). \end{array}$$

The class of 1 in M is mapped to $\langle e^f, e^{tg} \rangle$ in $\mathbb{Q}((t))$. What Algorithm 4 computes, is an operator $L(t, \partial_t)$ such that $L(t, \partial_t) \cdot 1 \in \boldsymbol{\partial} M$. This implies that $L(t, \partial_t) \cdot \langle e^f, e^{tg} \rangle = 0$.

Remark 65. At this point we can make explicit our remark that earlier creative telescoping algorithms could not deal with our integrals. From the explicit definition of g in the formula $R_k(t) = \langle e^f, e^{tg} \rangle$, we can prove that $\tilde{g}$ is always in the form $p_k + h(p_1, \dots, p_{k-1})$, for some polynomial h. So $p_k - t$ is always in S. Any integration algorithm that would work over $\mathbb{Q}(t, p_k)$, as many that are designed to apply to functions, would therefore consider that 1 is in the annihilator of the function to be integrated, which would lead to a wrong result.

3.6.2 Experimental results

We consider graph models that are either some model of k-regular (simple) graphs or some generalization with loops and/or multiple edges and/or degrees in the set $\{1, 2, \dots, k\}$ instead of $\{k\}$. Given such a graph model, the theory in [50] provides immediate formulas for the polynomials f and g. Obtaining the ideal of the lemma is easily implemented as a simple non-commutative substitution. To this end, we used Maple's `OreAlgebra` package. After converting[2] from Maple notation to the notation of `MultivariateCreativeTelescoping.jl`, we could use the latter to obtain the wanted ODEs, appealing to the implementation of our optimized Algorithm 5. This was done for $2 \leq k \leq 8$ and the degree sets $K = \{k\}$ and $K = \{1, 2, \dots, k\}$. We collected the computed ODEs and made them available on the web[3]. For $k \leq 7$, they are the same as with the Maple implementation that accompanies Chyzak and Mishna's article [50]. To the best of our knowledge, the ODEs for $k = 8$ are obtained for the first time. Table 3.1 displays some parameters related to the calculations and to the results we obtained.

[2] Maple uses its commutative product `*` to represent monomials, so that both Maple inputs `t*dt` and `dt*t` represent the element $t\partial_t$. This is no problem inside Maple, where non-commutative products are computed by the command `skew_product`. But naively serializing an operator from Maple by `lprint` can lead to strings with a different interpretation in Brochet's Julia implementation. We automated a rewrite of those strings to move all derivatives to the right.

[3] See https://files.inria.fr/chyzak/kregs/.

graph			ODE		time and memory			
k	l	e	ord	deg	f5	mct	total	rss
2	ll	se	1	2	0.04	0.05	18	0.63
3	ll	se	2	11	0.04	0.05	18	0.62
4	ll	se	2	14	0.05	0.05	19	0.62
5	ll	se	6	125	0.05	0.59	17	0.65
6	ll	se	6	145	0.23	1.0	20	0.66
7	ll	se	20	1683	8.6	303	330	4.6
7	ll	me	20	1683	8.4	300	326	4.3
7	la	se	20	1683	8.3	301	328	4.3
7	la	me	20	1683	8.3	299	326	4.4
7	lh	se	20	1683	8.6	310	337	5.9
7	lh	me	20	1683	8.3	322	349	5.8
8	ll	se	19	1793	244	832	1095	6.5
8	ll	me	19	1793	247	831	1097	6.7
8	la	se	19	1793	244	831	1094	6.0
8	la	me	19	1793	244	829	1093	6.0
8	lh	se	35	6204	393	23069	23481	5.4
8	lh	me	35	6200	393	23111	23524	5.4

graph			ODE		time and memory			
k	l	e	ord	deg	f5	mct	total	rss
2	ll	se	1	3	0.04	0.04	17	0.62
3	ll	se	2	11	0.04	0.13	18	0.64
4	ll	se	3	29	0.04	0.07	18	0.62
5	ll	se	6	125	0.06	0.69	19	0.65
6	ll	se	10	425	0.31	11	31	0.86
7	ll	se	20	1683	8.3	316	343	5.8
7	ll	me	20	1683	8.5	321	348	5.8
7	la	se	20	1683	8.7	324	352	5.8
7	la	me	20	1683	8.8	322	349	5.8
7	lh	se	20	1683	8.7	310	337	5.6
7	lh	me	20	1683	8.2	323	349	5.8
8	ll	se	35	6201	389	23198	23605	5.4
8	ll	me	35	6200	386	23586	23991	5.4
8	la	se	35	6204	401	23495	23915	5.4
8	la	me	35	6205	393	23188	23600	5.4
8	lh	se	35	6205	387	22745	23152	5.5
8	lh	me	35	6205	394	23440	23853	5.4

Table 3.1: Computation of ODEs for k-regular graph models $K = \{k\}$ (left) and $K = \{1, \dots, k\}$ (right). A graph model is input by the triple (k, l, e), where l and e describe if loops and edges are allowed. For each output ODE, the order ('ord') and the coefficient degree ('deg') are given. All times are given in seconds. The two main computation steps are preparing a Gröbner basis ('f5') and running Algorithm 5 on it ('mct'). The maximum memory used is given in GB ('rss'). See details in Section 3.6.2.

There, each graph model is described by a triple (k, l, e) where: k determines the set of allowed degrees, either by $K = \{k\}$ for the left part of Table 3.1 or by $K = \{1, \dots, k\}$ for the right part; l is one of 'll' for loopless graphs, 'la' for graphs with loops allowed and contributing 2 to the degree, and 'lh' for graphs with loops allowed and contributing 1 to the degree; e is either 'se' for graphs with simple edges or 'me' for graphs with multiple edges allowed. For each graph model, the resulting ODE has order and degree given in columns 'ord' and 'deg'. The total time for computing the ODE is given in column 'total' and decomposes as follows: the time to prepare generators for the ideal S is negligible; the time to make a Gröbner basis out of them is given in column 'f5'; the time to extract next the ODE by our new algorithm is given in column 'mct'. We note that the time for 'mct' always dominates. The total time includes the compilation time, which explains why its value exceeds the sum of 'f5' and 'mct'. The maximal peak of memory usage is listed in column 'rss'.

By way of comparison, the model (se, ll, 7) could be computed by the method and implementation of [50] in $3.22 \cdot 10^4$ seconds (almost 9 hours), which is roughly 100 times as much as the 330 seconds (5.5 minutes) needed by our new algorithm and implementation. This can partly be explained by the lack of efficient evaluation/interpolation methods in the implementation of [50] and by the choice in [50] to continue the calculation by factoring the polynomial coefficients of the ODE.

4 Weyl closure

The Weyl closure of a left module $S \subseteq W_{\mathbf{x}}^r$ is defined as

$$\mathrm{Cl}(S) = W_{\mathbf{x}}(\mathbf{x})S \cap W_{\mathbf{x}}^r. \tag{4.1}$$

The problem of computing generators of $\mathrm{Cl}(S)$ given S was first introduced by Chyzak and Salvy [45] for the case $r = 1$, under the name of extension/contraction problem. The objective was to compute the holonomic annihilator $\mathrm{ann}_{W_{\mathbf{x}}}(f)$ of a holonomic function f from its D-finite annihilator $\mathrm{ann}_{W_{\mathbf{x}}(\mathbf{x})}(f)$, as a preliminary step before applying Takayama's algorithm to integrate f. This problem was later solved by Tsai for general r in his PhD thesis [133, 134] using a localization algorithm by Oaku, Takayama, and Walther [103].

Like Chyzak and Salvy, my goal is to compute a presentation of a holonomic module starting from the presentation of a D-finite module. However, experiments show that computing Weyl closures with current implementations is expensive, notably because they compute a b-function. For this reason, I instead aim to compute a holonomic approximation of the closure, that is I aim to compute a module S' such that $S' \subseteq \mathrm{Cl}(S)$ and $W_{\mathbf{x}}^r/S'$ is holonomic. This holonomy condition is sufficient to ensure the termination of the integration algorithm presented in Chapter 3. The main contribution of this section is a new algorithm to solve this problem, which is asymptotically maximal. That is, the algorithm constructs iteratively a sequence of modules

$$S_1 \subseteq S_2 \subseteq \cdots \subseteq \mathrm{Cl}(S) \tag{4.2}$$

with the property that there exists $\ell \in \mathbb{N}$ such that $S_\ell = \mathrm{Cl}(S)$. The algorithm continues until a prescribed termination criterion is met. Possible termination criteria include: stopping at the first ideal S_i for which W_n/S_i is holonomic, stopping at the first S_i such that $S_i = S_{i+1}$ and W_n/S_i is holonomic, or stopping at the first S_{i+r} such that W_n/S_i is holonomic for some prescribed $r \in \mathbb{N}$. At present, however, I do not know any method to guarantee that the output coincides with $\mathrm{Cl}(S)$ without explicitly computing a b-function.

The algorithm is based on a generalization of Rabinowitsch's trick to D-modules. This approach does not require computing any localization of the module S, instead it computes successive saturations with respect to a well-chosen polynomial. For some applications, it is needed to extend the notion of holonomy to modules of the form $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$, which involve both polynomial and rational variables. This extension enables the computation of annihilators in this algebra and yields a variant of the algorithm described in Section 3.4.3, that produces the input data structure for the integration algorithm of Chapter 3. In contrast to the algorithm of Section 3.4.3, which requires a holonomic module, this variant directly computes the presentation M and the derivation ∂_t from a D-finite module.

Relation with desingularization

Another motivation for the Weyl closure is its application to the desingularization of systems of linear differential equations. By definition, the singularities of an operator $L \in W_1$ are the roots of its leading coefficient. Any singularity of a solution f of L is also a singularity of L, but the converse is not true. For example the operator $L = x\partial - 1$ has a one-dimensional solution space generated by $f = x$ which does not have a singularity at $x = 0$. A singularity α is said to be apparent if no solution of L admits a singularity at α. In this case there exists an operator $M \in W_x(x)$ such that ML has no singularity at α and the process of finding such an operator M is called desingularization. One application of desingularization is the computation of a basis of formal power series solutions of L at an apparent singularity. I refer to [8, 42, 75, 146] for desingularization algorithms for univariate operators and first-order linear systems.

The definition of singularity was extended to D-finite systems by Chen, Kauers, Li and Zhang [41]. Let $G \subseteq W_{\mathbf{x}}$ be a reduced Gröbner basis defining a D-finite ideal, with all operators in G having their content equal to one. A singularity of G is any point $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha}$ is a root of $\operatorname{lc}(g)$ for some $g \in G$. A singularity of G is called apparent if no solution of $W_{\mathbf{x}}(\mathbf{x})G$ admits a singularity at $\boldsymbol{\alpha}$ and in this case there exists a Gröbner basis $G' \subset W_{\mathbf{x}}(\mathbf{x})$ composed of left multiples of elements of G such that G' has no singularity at $\boldsymbol{\alpha}$ (but we don't have $W_{\mathbf{x}}G' \subseteq W_{\mathbf{x}}G$). The nature of a singularity depends on the monomial order chosen for the Gröbner basis G, that is an apparent singularity for some order may not be a singularity at all for another order. A stricter notion of singularity is given by the singular locus (see Section 4.1.3) of a $W_{\mathbf{x}}$-module. The Weyl closure of $W_{\mathbf{x}}G$ is the smallest module containing G and all desingularizations of G.

Contents

The chapter is organized as follows. Section 4.1.1 provides a brief review of the state-of-the-art, recalling Tsai's algorithm for computing Weyl closures of holonomic modules, along with the classical D-modules algorithms used in the process. Section 4.2 introduces the new algorithm for approximating Weyl closures, and finally, Section 4.3 presents the benchmarks of a preliminary implementation.

4.1 State-of-the-art algorithms

In this section, I begin by recalling the classical algorithms for computing the annihilator of f^s for a symbolic variable s, the global b-function of f, and the annihilator of f^α for $\alpha \in \mathbb{C}$ (Section 4.1.1). Next, I recall two localization algorithms that build upon these methods (Section 4.1.2). Finally, I recall Tsai's algorithm for computing the Weyl closure $\operatorname{Cl}(I)$ of a holonomic module $W_{\mathbf{x}}/I$ (Section 4.1.3).

4.1.1 Annihilator of the power of a polynomial and global b-function

Let $f \in \mathbb{K}[\mathbf{x}]$ be a polynomial and s be a symbolic parameter. I describe in this subsection an algorithm due to Oaku [99] and an algorithm due to Briançon-Maisonobe [25] to compute the annihilator of f^s following an idea by Malgrange [93]. Here, f^s should be thought of as a symbol that behaves like the complex valued function f^s. This computation is the first step before computing the global b-function (Section 4.1.1.2) and the annihilator of f^α (Section 4.1.1.3) for a complex number α in a suitable complex open subset of $\mathbb{C}$.

4.1.1.1 Annihilator of f^s

The content of this subsection is based on Chapter 5.3 of the book [114]. Let $\delta(t-f)$ be the distribution defined for any smooth function $\varphi(t,\mathbf{x})$ with compact support by

$$\langle \delta(t-f), \varphi \rangle = \int_{\mathbb{R}^n} \varphi(f(\mathbf{x}), \mathbf{x}) d\mathbf{x} \tag{4.3}$$

Malgrange described explicitly a generating set for the annihilator of this distribution and related it to the annihilator of f^s by means of the Mellin transform.

Theorem 66 (Malgrange [93])**.** *Let $f \in \mathbb{K}[\mathbf{x}]$ be a polynomial. The annihilator of $\delta(t-f)$ is generated in $W_{\mathbf{x},t}$ by*

$$t - f(\mathbf{x}) \text{ and } \partial_i + \frac{\partial f}{\partial x_i}\partial_t \text{ for } i \in \{1,\dots,n\}. \tag{4.4}$$

The *Mellin transform* $\tilde{M}$ is the linear map defined for piece-wise continuous functions F (whenever the integral converges) by

$$\tilde{M} : F(t) \mapsto G(s) = \int_0^\infty F(t) t^s dt. \tag{4.5}$$

Although a generalization of the Mellin transform to distributions exists, to illustrate the connection between the distribution $\delta(t-f)$ and the expression f^s, I present the following intuitive, though not rigorous, computation:

$$\tilde{M}(\delta(t-f)) = \int_{\mathbb{R}} \delta(t-f)\, t^s\, dt = \int_{t=f} t^s\, dt = f^s. \tag{4.6}$$

If F decays rapidly and s belongs to a suitable open subset of $\mathbb{C}$, an algebraic counterpart of the Mellin transform exists for operators. It is given by the multiplicative $W_{\mathbf{x}}$-linear map

$$M : \begin{cases} W_{\mathbf{x},t} & \to W_{\mathbf{x}}\langle s, S, S^{-1}\rangle \\ t & \mapsto S \\ \partial_t & \mapsto -sS^{-1} \end{cases}, \tag{4.7}$$

where S denotes the shift operator with respect to s, satisfying $sS = Ss + S$ while S^{-1} is its inverse. This map is well-defined as $M(\partial_t t) = -s = M(t\partial_t + 1)$. Besides, this formula shows that the map M induces an isomorphism from the subalgebra $W_{\mathbf{x}}[t\partial_t]$ of $W_{\mathbf{x},t}$ onto the subalgebra $W_{\mathbf{x}}[s]$ of $W_{\mathbf{x}}\langle s, S, S^{-1}\rangle$. This isomorphism leads to the following theorem and to Algorithm 6 for computing the annihilator of f^s in $W_{\mathbf{x}}[s]$.

Theorem 67. *The annihilator of f^s in $W_{\mathbf{x}}[s]$ equals* $M(\operatorname{ann}_{W_{\mathbf{x},t}}(\delta(t-f)) \cap W_{\mathbf{x}}[t\partial_t])$.

Algorithm 6 Annihilator of f^s in $W_{\mathbf{x}}[s]$ (Oaku [99])

Input: $f \in W_{\mathbf{x}}$
Output: generators of $\operatorname{ann}_{W_{\mathbf{x}}[s]}(f^s)$

Let u, v be two new commutative polynomial variables.
$G \leftarrow \{tu - f, uv - 1\} \cup \left\{\partial_i + \frac{\partial f}{\partial x_i} v \partial_t \mid i \in [\![1, n]\!]\right\}$
Compute the intersection $I = W_{\mathbf{x},t}[u, v]G \cap W_{\mathbf{x},t}$ by Gröbner basis computation.
Each generator of I has the form $t^a p(t\partial_t)\partial_t^b$ for $a, b \in \mathbb{N}$ and $p \in W_{\mathbf{x}}[t\partial_t]$.
Multiply each $t^a p(t\partial_t)\partial_t^b$ by $t^b \partial^a$ so that they belong to $\operatorname{ann}_{W_{\mathbf{x},t}}(\delta(t-f)) \cap W_{\mathbf{x}}[t\partial_t]$.
Replace $t\partial_t$ by $-s-1$ in each generator and output the results.

This algorithm requires introducing two additional variables u, v to compute the intersection with $W_{\mathbf{x}}[t\partial_t]$. This is a source of inefficiency which led Briançon and Maisonobe to propose another algorithm [25]. Observe that the variable t appears in only one generator of $\operatorname{ann}_{W_{\mathbf{x},t}}(\delta(t-f))$ (Eq. (4.4)). Consequently, this generator, $t - f$, can contribute to the intersection $\operatorname{ann}_{W_{\mathbf{x},t}}(\delta(t-f)) \cap W_{\mathbf{x}}[t\partial_t]$ only if it is multiplied on the left by ∂_t at least once. In that case, the problem reduces to computing the intersection with $W_{\mathbf{x}}[t\partial_t]$ of the ideal generated by

$$\begin{gathered} \partial_t t - f\partial_t, \\ \partial_i + \frac{\partial f}{\partial x_i}\partial_t \quad \text{for } i = 1, \dots, n. \end{gathered} \tag{4.8}$$

Applying the Mellin transform gives:

$$\begin{aligned} M(\partial_t t - f\partial_t) &= -s - fM(\partial_t) \\ M(\partial_i + f\partial_t) &= \partial_i + fM(\partial_t) \quad \text{for } i = 1, \dots, n. \end{aligned} \tag{4.9}$$

Briançon and Maisonobe showed that the annihilator of f^s could be obtained by eliminating $M(\partial_t)$ in Eq. (4.9). Note that s and $M(\partial_t)$ satisfy the commutation rule of a backward shift (since $M(\partial_t) = -sS^{-1}$). Hence, by renaming $M(\partial_t)$ into $\widetilde{S^{-1}}$ we obtain Algorithm 7. I was not able to find Briançon and Maisonobe's original article, hence I refer to [**andresPrincipalIntersectionBernsteinsato2009a**] for a proof of correctness. According to [89], Algorithm 7 is between 1.5 and 3 times faster than Algorithm 6 across a variety of examples.

Algorithm 7 Annihilator of f^s (Briançon and Maisonobe [25])

Input: $f \in W_{\mathbf{x}}$
Output: generators of $\mathrm{ann}_{W_{\mathbf{x}}[s]}(f^s)$
$G \leftarrow \left\{ s + f\widetilde{S^{-1}} \right\} \cup \left\{ \partial_i + \frac{\partial f}{\partial x_i}\widetilde{S^{-1}} \mid i \in [\![1, n]\!] \right\}$
Compute $\mathrm{ann}_{W_{\mathbf{x}}[s]} = W_{\mathbf{x}}\langle s, \widetilde{S^{-1}}\rangle G \cap W_{\mathbf{x}}[s]$ by Gröbner basis computation
return $\mathrm{ann}_{W_{\mathbf{x}}[s]}$

4.1.1.2 Global b-function

The global b-function of f, also known as the Bernstein-Sato polynomial, is the monic polynomial $b_f(s) \in \mathbb{K}[s]$ of smallest degree satisfying

$$P \cdot f^{s+1} = b_f(s) f^s \tag{4.10}$$

for some operator $P \in W_{\mathbf{x}}[s]$. The existence of such a polynomial was first proved by Bernstein [13]. It was originally introduced as a tool for defining the analytic continuation of the function $\lambda \mapsto f^\lambda$, but it has since found numerous other applications in singularity theory and quantum field theory. An important result due to Kashiwara [80] states that the roots of the global b-function are negative rational numbers. As we will see in the next sections, its minimal integer root serves as an upper bound in many algorithms on D-modules.

A direct computation shows that the ideal of polynomials $b(s)$ satisfying Eq. (4.10) without the minimality condition coincides with the ideal

$$(\mathrm{ann}_{W_{\mathbf{x}}[s]}(f^s) + W_{\mathbf{x}}[s]f) \cap \mathbb{K}[s]. \tag{4.11}$$

This naturally leads to an algorithm to compute b_f, which is described in Algorithm 8. It was observed [**andresPrincipalIntersectionBernsteinsato2009a**] that the intersection in step 2 can be computed efficiently by a variant of the FGLM algorithm [64], as the intersection is a zero-dimensional polynomial ideal.

Algorithm 8 b-function (Oaku) [99]

Input: $f \in W_{\mathbf{x}}$
Output: the b-function associated to f
Compute generators of $\mathrm{ann}_{W_{\mathbf{x}}[s]}(f^s)$ by e.g. Algorithm 7
return the monic generator of the ideal $(\mathrm{ann}_{W_{\mathbf{x}}[s]}(f^s) + W_{\mathbf{x}}[s]f) \cap \mathbb{K}[s]$

Remark 68. There exists also a notion of b-function of a holonomic ideal associated to a weight vector w [114, chapter 5.1]. This b-function is useful for computing the integral of a holonomic module and its restriction [114, chapter 5.2 and 5.5].

4.1.1.3 Annihilator of f^α and deformation methods

Let α be a complex number. The annihilator of f^α can be obtained from the annihilator of f^s by evaluating s at α, provided α does not belong to the set $\alpha_0 + 1 + \mathbb{N}$, where α_0 is the smallest integral root of the b-function b_f [114, Theorem 5.3.13]. This minimum is well-defined as -1 is always a root of b_f. If α belongs to this exceptional set, an additional step involving a syzygy computation must be performed to ensure that the full annihilator is computed. This algorithm, due to Oaku and Takayama, is to the best of my knowledge only described in the book [114, Algorithm 5.3.15].

I propose a new algorithm for computing a holonomic approximation of $\operatorname{ann}(1/f)$, based on an idea suggested by Pierre Lairez. It corresponds to an algebraic equivalent of deformation techniques and is also reminiscent of Theorem 66. Let ε be a new variable and let us remark that $1/\varepsilon$ is annihilated in $W_{\mathbf{x},\varepsilon}$ by

$$J = \{\varepsilon\partial_\varepsilon + 1\} \cup \{\partial_i \mid i \in [\![1, n]\!]\}.$$

Moreover, the function

$$\phi : \begin{cases} W_{\mathbf{x},\varepsilon} & \to W_{\mathbf{x},\varepsilon} \\ \varepsilon & \mapsto \varepsilon + f \\ \partial_\varepsilon & \mapsto \partial_\varepsilon \\ \partial_i & \mapsto \partial_i + \frac{\partial f}{\partial x_i}\partial_\varepsilon \end{cases}$$

defines a morphism of $W_{\mathbf{x},\varepsilon}$-module. As a consequence, $\phi(W_{\mathbf{x},\varepsilon}J)$ annihilates the function $1/(\varepsilon + f)$, and we can find a holonomic approximation of $\operatorname{ann}_{W_\mathbf{x}}(1/f)$ by computing the restriction of $\phi(W_{\mathbf{x},\varepsilon}J)$ at $\varepsilon = 0$. The restriction of a $W_{\mathbf{x},\varepsilon}$-module at $\varepsilon = 0$ is the module $M/(\varepsilon M)$. It can be computed using Takayama's algorithm and the Fourier transform [102] (see also [114, chapter 5.2 and 5.5]).

Although I have not seriously tested this algorithm, it has the advantage of making a link between the problems of restriction/integration and the problem of computing the annihilator of a rational function.

Remark 69. The annihilator of a rational function a/f^α can be deduced from the annihilator of $\operatorname{ann}(1/f^\alpha)$ by computing the intersection $I = W_\mathbf{x}a \cap \operatorname{ann}(1/f^\alpha)$ and returning Ia^{-1}.

4.1.1.4 Polynomial Gröbner basis approaches for approximating $\operatorname{ann}(1/f^\alpha)$

The high computational cost of non-commutative Gröbner basis computations in Algorithms 6 and 7 has driven researchers to explore alternative methods for approximating this ideal using polynomial Gröbner bases. One approach, proposed in [35], constructs new generators of $\operatorname{ann}(1/f^\alpha)$ by finding logarithmic derivatives of f, that is relations of the form

$$af = \sum_{i=1}^{n} b_i\partial_i(f), \tag{4.12}$$

where $a, b_1, \ldots, b_n$ are polynomials in $\mathbb{K}[\mathbf{x}]$. Such relations directly yield annihilating operators for $1/f^\alpha$:

$$\left(\sum_{i=1}^{n} b_i \partial_i + \alpha a\right) \cdot \frac{1}{f^\alpha} = 0. \tag{4.13}$$

Another approach, developed in [33], looks for polynomial syzygies between the successive derivatives $\boldsymbol{\partial}^{\boldsymbol{\beta}} \cdot f$ of f. If $\sum_i b_i f_i = 0$, where $f_i = \partial_i \cdot f$, then one obtains

$$\sum_i b_i (f \partial_i + \alpha f_i) = f \sum_i b_i \partial_i \tag{4.14}$$

from which one deduces a new annihilating operator by dividing by f.

Both approaches lead to algorithms that are asymptotically maximal.

4.1.2 Localization of a D-module

Let M be a finitely presented $W_{\mathbf{x}}$-module and $f \in \mathbb{K}[\mathbf{x}]$ be a polynomial. The localization of the module M at f is the $W_{\mathbf{x}}$-module

$$\mathbb{K}[\mathbf{x}, 1/f] \otimes_{\mathbb{K}[\mathbf{x}]} M. \tag{4.15}$$

Kashiwara [81] proved that this module is holonomic whenever M is holonomic and that $\mathbb{K}[\mathbf{x}, 1/f]$ is isomorphic to $W_{\mathbf{x}} / \operatorname{ann}(f^{-k})$ as $W_{\mathbf{x}}$-modules, where k is the smallest integer root of $b_f(s)$.

By combining Algorithm 7 for computing the annihilator of f^s and Algorithm 8 for computing the b-function of f, with the tensor product algorithm I recall in the next paragraph, Oaku and Takayama [102] derived an algorithm for computing the localization module $\mathbb{K}[\mathbf{x}, 1/f] \otimes_{\mathbb{K}[\mathbf{x}]} M$.

Tensor product

Let I and J be presentations of two $W_{\mathbf{x}}$-modules M and N. The tensor product $M \otimes_{\mathbb{K}[\mathbf{x}]} N$ has a natural structure of $W_{\mathbf{x}}$-module given by

$$\partial_i \cdot (a \otimes b) = (\partial_i \cdot a) \otimes b + a \otimes (\partial_i \cdot b). \tag{4.16}$$

A presentation of this tensor product can be computed from I and J as follows.

First rename the variables $\mathbf{x}$ into $\mathbf{z}$ in one of the presentation, e.g. J, and then quotient the $W_{\mathbf{x},\mathbf{z}}$-module $I \otimes_{\mathbb{K}[\mathbf{x},\mathbf{z}]} J$ by the right $W_{\mathbf{x},\mathbf{z}}$-ideal generated by $x_i - z_i$ for all i. This step corresponds to computing a restriction [102] (see also [114, chapter 5.2 and 5.5]). Conceptually, the algorithm mirrors the definition of the tensor product: it introduces two independent copies of the objects I and J and then quotient by what should be equal. The algorithm is described in Algorithm 9 in a slightly modified form, with the restriction taken at $\mathbf{z} = 0$ to match the common practice in the literature.

This method requires doubling the number of variables in the algebra and then eliminating half of them through a restriction. As a consequence, its computational cost is often prohibitive, and its use should be avoided whenever more efficient alternatives are available.

Algorithm 9 Tensor product (Oaku-Takayama [102])

Input: presentations $I, J \subseteq W_{\mathbf{x}}$ of two modules M and N
Output: a presentation of a module M' isomorphic to $M \otimes_{\mathbb{K}[\mathbf{x}]} N$

Replace the variable $\mathbf{x}$ by $\mathbf{z}+\mathbf{x}$ in J and $\partial_{\mathbf{x}}$ by $\partial_{\mathbf{x}} - \partial_{\mathbf{z}}$ in I
Compute the restriction of the $W_{\mathbf{x},\mathbf{z}}$-module $I + J$ at $\mathbf{z} = 0$ with Oaku-Takayama's algorithm [102]
return generators of the restriction

Remark 70. This algorithm can be used to compute the annihilator $\operatorname{ann}_{W_{\mathbf{x}}}(fg)$ of the function fg starting from the annihilator of f and g (e.g. section 3 [100]).

Localization without tensor product computation

A more efficient algorithm for computing a presentation of the localization module

$$M \otimes_{\mathbb{K}[\mathbf{x}]} \mathbb{K}[\mathbf{x}, 1/f]$$

with a single use of Takayama's algorithm was given by Oaku, Takayama and Walther [103]; see also [101] for an alternative presentation.

Recall that $\mathbb{K}[\mathbf{x}, 1/f]$ is isomorphic to $W_{\mathbf{x}}/\operatorname{ann}(f^{-k})$ where $-k$ is the smallest integer root of $b_f(s)$. One can show, similarly to Theorem 66, that the annihilator of f^{-s} is generated in $W_{\mathbf{x},t}[s, 1/f]$ by

$$ft - 1, \quad \partial_i - \frac{\partial f}{\partial x_i}\partial_t t^2, \tag{4.17}$$

where the module structure in t, ∂_t is the one induced by the Mellin transform (Eq. (4.7)). That is, $t \cdot f^{-s} = ff^{-s}$ and $\partial_t \cdot f^{-s} = \frac{-s}{f} f^{-s}$, and the commutation rules are $ts = (s+1)t$ and $\partial_t s = (s-1)\partial_t$.

Now consider the tensor product

$$M \otimes_{\mathbb{K}[\mathbf{x}]} W_{\mathbf{x},t}[s, 1/f] \cdot f^s \tag{4.18}$$

which naturally inherits a $W_{\mathbf{x},t}$-module structure. Assuming M is of the form $W_{\mathbf{x}}/I$, Oaku, Takayama, and Walther explicitly described the annihilator of the class of $1 \otimes 1$, thereby obtaining a presentation of the tensor product in Eq. (4.18) as a $W_{\mathbf{x},t}$-module.

Notation 71. Let $P \in W_{\mathbf{x}}$, we denote by $P(\boldsymbol{\partial} - \frac{\partial f}{\partial \mathbf{x}}\partial_t t^2)$ the operator obtained from P by replacing each variable ∂_i with $\partial_i - \frac{\partial f}{\partial x_i}\partial_t t^2$.

Lemma 72. *(Oaku, Takayama, Walther [103]) Let $P_1, \ldots, P_\ell$ be generators of the $W_{\mathbf{x}}$-ideal I. Then the $W_{\mathbf{x},t}$-module $M \otimes_{\mathbb{K}[\mathbf{x}]} W_{\mathbf{x},t}/\operatorname{ann}_{W_{\mathbf{x},t}[s,1/f]}(f^{-s})$ is isomorphic to the $W_{\mathbf{x},t}$-module $W_{\mathbf{x},t}/J$ where J is generated by*

$$ft - 1, \quad P_i(\boldsymbol{\partial} - \frac{\partial f}{\partial \mathbf{x}}\partial_t t^2) \text{ for } i \in \{1, \ldots, \ell\}. \tag{4.19}$$

They then proved that a presentation of the localization module $M \otimes_{\mathbb{K}[\mathbf{x}]} \mathbb{K}[\mathbf{x}, 1/f]$ could be obtained by computing the integral of the module $W_{\mathbf{x},t}/J$ with respect to t:

$$M/(J + \partial_t W_{\mathbf{x},t}). \tag{4.20}$$

This leads to the following theorem and to Algorithm 10.

Theorem 73. *(Oaku, Takayama, Walther [103]) The module $M \otimes_{\mathbb{K}[\mathbf{x}]} \mathbb{K}[\mathbf{x}, 1/f]$ is isomorphic to*

$$W_{\mathbf{x},t}/\,(J + \partial_t W_{\mathbf{x},t})$$

as $W_{\mathbf{x}}$-module. Besides, the natural map $M \mapsto M \otimes_{\mathbb{K}[\mathbf{x}]} \mathbb{K}[\mathbf{x}, 1/f]$ sends $\bar{1}$ to $\bar{1} \otimes 1/f^2$.

Algorithm 10 Localization (Oaku-Takayama-Walther [103])

Input: a polynomial $f \in \mathbb{K}[\mathbf{x}]$ and a presentation $I \subseteq W_{\mathbf{x}}$ of a module M.
Output:

- a presentation L of the module $M \otimes_{\mathbb{K}[\mathbf{x}]} \mathbb{K}[\mathbf{x}, 1/f]$
- an integer $k \in \mathbb{N}$ such that $\bar{1} \otimes 1/f^{k+2}$ generates $\mathbb{K}[\mathbf{x}, 1/f] \otimes_{\mathbb{K}[\mathbf{x}]} M$

Let $P_1, \ldots, P_\ell$ be generators of I and define J as in Eq. (4.19)
Compute the b-function B associated to the integration ideal [114, chap. 5.1]
$$K = W_{\mathbf{x},t}/(J + \partial_t W_{\mathbf{x},t})$$
if B has no non-negative integer root
  return I and 0
else
  Let k be the largest integer root of B so that L is generated by the class of t^k
  Compute the annihilator L of t^k in K
  return L and k
end

4.1.3 Tsai's algorithm for the Weyl closure of a holonomic module

This section presents Tsai's algorithm for computing the Weyl closure of a holonomic module. To simplify the notation, I present it for holonomic ideals exclusively, but Tsai also presented in for holonomic modules. I first recall the definition of the singular locus of an ideal, which corresponds to the set of all singularities of the solution functions of I. Its definition involves the notions of weight vectors and initial ideals, which although classical in D-module theory are scarcely used in this thesis.

Singular locus

Let I be an ideal of $W_{\mathbf{x}}$ and let $(\mathbf{0}, \mathbf{1})$ denote the weight vector consisting of n zeros followed by n ones [114, Chapter 1.2]. Let $P = \sum_{\alpha,\beta} a_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\beta}}$ be an operator in $W_{\mathbf{x}}$ and let $c = \max\left(|\boldsymbol{\beta}| \mid \exists, \boldsymbol{\alpha}\, a_{\boldsymbol{\alpha},\boldsymbol{\beta}} \neq 0\right)$. The initial form of P with respect to $(\mathbf{0}, \mathbf{1})$ is the polynomial in $\mathbb{K}[\mathbf{x}, \boldsymbol{\xi}]$ given by

$$\mathrm{in}_{(\mathbf{0},\mathbf{1})}(P) = \sum_{\boldsymbol{\alpha}, |\boldsymbol{\beta}|=c} a_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\alpha}} \boldsymbol{\xi}^{\boldsymbol{\beta}}. \tag{4.21}$$

The initial ideal $\mathrm{in}_{(\mathbf{0},\mathbf{1})}(I)$ is the ideal of $\mathbb{K}[\mathbf{x}, \boldsymbol{\xi}]$ containing all the initial forms of element of I. The singular locus of I is then the ideal

$$\mathrm{Sing}(I) = \left(\mathrm{in}_{(\mathbf{0},\mathbf{1})}(I) : \langle \xi_1, \ldots, \xi_n \rangle^{\infty}\right) \cap \mathbb{C}[\mathbf{x}],$$

where $A : B^{\infty}$ denotes the saturation of the ideal A with respect to the ideal B which is defined by

$$A : B^{\infty} = \left\{ a \in \mathbb{K}[\mathbf{x}, \boldsymbol{\xi}] \mid \exists k \in \mathbb{N},\ B^k a \subseteq A \right\}. \tag{4.22}$$

An algorithm to compute the singular locus was proposed by Oaku [98].

Tsai's algorithm

Tsai proved that the Weyl closure of an ideal I of $W_{\mathbf{x}}$ could be obtained by localizing the ideal I at a polynomial vanishing on the singular locus of I.

Theorem 74. *(Tsai [134, Theorem 2.2.1]) Let I be an ideal of $W_{\mathbf{x}}$ and f be a polynomial vanishing on the singular locus of $W_{\mathbf{x}}/I$. The Weyl closure* $\mathrm{Cl}(I)$ *of I is equal to*

$$\mathbb{K}[\mathbf{x}, 1/f] I \cap W_{\mathbf{x}}.$$

This leads directly to an algorithm for the computation of the Weyl closure of ideals I such that $W_{\mathbf{x}}^r / \mathrm{Cl}(I)$ is holonomic, which is described in Algorithm 11.

Algorithm 11 Weyl closure (Tsai [134, algorithm 2.2.4])

Input: an ideal $I \subseteq W_{\mathbf{x}}$ such that $W_{\mathbf{x}}^r / \mathrm{Cl}(I)$ is holonomic.
Output: The Weyl closure $\mathrm{Cl}(I)$
Compute a polynomial f vanishing on the singular locus of I
Compute L and k such that $W_{\mathbf{x}}/L \simeq W_{\mathbf{x}}/I \otimes \mathbb{K}[\mathbf{x}, 1/f]$ by Algorithm 10
return the intersection $W_{\mathbf{x}} f^{k+2} \cap L$

Remark 75. When $W_{\mathbf{x}}/\mathrm{Cl}(I)$ is not holonomic, the localization may not be finitely generated. Tsai presented another algorithm for this setting which is described in his PhD thesis [134, Chapter 2.3].

4.2 Computing an approximation of the partial Weyl closure

Notation 76. I denote $W_{\mathbf{x},\mathbf{t}}$ the Weyl algebra with n variables $x_1, \ldots, x_n$ and m variables $t_1, \ldots, t_m$.

Let S be a submodule of $W_{\mathbf{x},\mathbf{t}}^r$ such that $W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})^r / W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})S$ is a D-finite module. The *partial Weyl closure* of S with respect to $\mathbf{x}$ is the module

$$\mathrm{Cl}_{\mathbf{x}}(S) = W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})S \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r.$$

This definition coincides with the standard Weyl closure when no variable $\mathbf{t}$ are present. I begin this section by presenting the notion of holonomy for finitely presented $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-modules. Then I propose a new algorithm for computing a holonomic approximation of $\mathrm{Cl}_{\mathbf{x}}(S)$. This algorithm is based on a non-commutative extension of Rabinowitsch's trick [108] and on Gröbner bases of infinite rank modules [128]. Unlike Tsai's algorithm, which proceeds by computing the localization of the module S, my method proceeds by computing the saturation with respect to a suitable polynomial f iteratively until a holonomic approximation of $\mathrm{Cl}_{\mathbf{x}}(S)$ is found.

4.2.1 Holonomy

I first recall in Theorem 80 the algorithmic criterion for testing, given a Gröbner basis of the W_x-ideal I, whether the module $W_{\mathbf{x}}/I$ is holonomic. This criterion naturally generalizes to finitely generated modules. The criterion has been used as a definition for holonomy in the literature (e.g., [82, Definition 4.67]), but I have not been able to find a proof of it. I recall several definitions from Section 3.2.2 to make the presentation self-contained.

4.2.1.1 Algorithmic criterion of holonomy

Let $(\mathcal{F}_d)_d$ be the *Bernstein filtration* of $W_{\mathbf{x}}$ defined by

$$\mathcal{F}_d = \{P \in W_{\mathbf{x}} \mid \deg(P) \leq d\}.$$

Let $M = W_{\mathbf{x}}/I$ be a left $W_{\mathbf{x}}$-module, the Bernstein filtration induces a filtration $(\Phi_d)_d$ on M defined by

$$\Phi_d = \{P + I \mid P \in \mathcal{F}_d\} \subseteq W_{\mathbf{x}}/I.$$

There exists a unique polynomial p called the *Hilbert polynomial* of M such that $\dim(\Phi_k) = p(k)$ for any sufficiently large k.

Definition 77. The dimension of the $W_{\mathbf{x}}$-module M is the degree of its Hilbert polynomial. In another word, the module M has dimension ℓ if and only if

$$\dim(\Phi_d) = \Theta(d^\ell) \text{ when } d \to +\infty.$$

Let $A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$ be a set of variables. The $\mathbb{K}$-algebra $\mathbb{K}\langle A\rangle$ is the algebra generated by A and the usual commutation rules between $\mathbf{x}$ and $\boldsymbol{\partial}$.

Lemma 78. *Let $A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$ be a set of cardinality $\ell > 0$. The dimension of the vector space $\mathbb{K}\langle A\rangle_{\leq d}$ is $\Theta(d^\ell)$ as d tends to infinity.*

Proof. It amounts to counting the number of commutative monomials in ℓ variables of degree at most d, which is known to be equal to $\binom{d+\ell}{\ell}$. □

I will prove the following theorem.

Theorem 79. *Let I be a non-trivial ideal of $W_{\mathbf{x}}$, let $\ell \in [\![0, 2n-1]\!]$, let $\preccurlyeq$ be a total degree monomial order on $W_{\mathbf{x}}$ and let G be a Gröbner basis of I for this order. The following statements are equivalent:*

1. *The module $W_{\mathbf{x}}/I$ has dimension $\leq \ell$.*
2. $\forall A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$, $|A| = \ell + 1 \implies$ *the vector space* $I \cap \mathbb{K}\langle A\rangle$ *is non-zero.*
3. $\forall A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$, $|A| = \ell + 1 \implies \exists g \in G$, $\mathrm{lm}(g) \in \mathbb{K}\langle A\rangle$.

Proof. I first prove the implication (1) $\implies$ (2). Let $A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$ be a subset of cardinality $\ell + 1$ and consider the canonical projection

$$\pi : \mathbb{K}\langle A\rangle_{\leq d} \to \Phi_d.$$

By assumption, the vector space Φ_d has dimension $\Theta(d^{\ell'})$ for some $\ell' \leq \ell$ and the vector space $\mathbb{K}\langle A\rangle_{\leq d}$ has dimension $\Theta(d^{\ell+1})$ by Lemma 78. Therefore, the map π can not be injective for large d and its kernel $\mathbb{K}\langle A\rangle_{\leq d} \cap I$ must be non-zero.

Now, I prove the implication (2) $\implies$ (3). Let $A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$ such that $|A| = \ell + 1$. By (2), there exists a non-zero element a in $\mathbb{K}\langle A\rangle \cap I$, and by property of Gröbner bases there exists $g \in G$ such that $\mathrm{lm}(g) \mid \mathrm{lm}(a)$. But a is in $\mathbb{K}\langle A\rangle$, hence $\mathrm{lm}(g)$ is also in $\mathbb{K}\langle A\rangle$.

Lastly, I prove the implication (3) $\implies$ (1). Let $A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$ and let

$$B_d^A = \left\{\mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}^{\boldsymbol{\beta}} \in \mathbb{K}\langle A\rangle_{\leq d} \mid \forall g \in G,\, \mathrm{lm}(g) \nmid \mathbf{x}^{\boldsymbol{\alpha}}\boldsymbol{\partial}^{\boldsymbol{\beta}}\right\}.$$

A basis of the vector space Φ_d is given by the image of $B_d^{\{\mathbf{x},\boldsymbol{\partial}\}}$, as it corresponds to the standard vector basis on the quotient $W_{\mathbf{x}}/I$ with respect to the Gröbner basis G. I prove by induction on $|A|$ that $|B_d^A| = O(d^\ell)$, which will be enough to conclude $\dim(\Phi_d) = O(d^\ell)$.

I start with the base case $|A| \leq \ell$. The elements of the set B_d^A form a free family of the vector space $\mathbb{K}\langle A\rangle_{\leq d}$, which has dimension $\Theta(d^{|A|})$ by Lemma 78, hence $|B_d^A| = O(d^\ell)$.

Then I prove the induction step. I assume $|A| \geq \ell + 1$. Let $g \in G$ be an element satisfying $\mathrm{lm}(g) \in \mathbb{K}\langle A\rangle$ and let $m \in B_d^A$. By definition g does not divide m, so there exists $a_0 \in A$ such that $\deg_{a_0}(g) > \deg_{a_0}(m)$. Noting $s = \deg_{a_0}(m)$, I remark that m/a_0^s belongs to $B_d^{A\setminus\{a_0\}}$. Hence,

$$m \in a_0^s B_d^{A\setminus\{a_0\}} \subseteq \bigcup_{i=0}^{\deg_{a_0}(g)} a_0^i B_d^{A\setminus\{a_0\}} \subseteq \bigcup_{a\in A} \bigcup_{i=0}^{\deg_a(g)} a^i B_d^{A\setminus\{a\}}.$$

Since the last set is independent of m, I obtain

$$|B_d^A| \le \sum_{a\in A} \sum_{i=0}^{\deg_a(g)} |B_d^{A\setminus\{a\}}|.$$

Remark that the summation ranges are independent of d and that by induction hypothesis each $B_d^{A\setminus\{a\}}$ has cardinality $O(d^\ell)$, hence $|B_d^A| = O(d^\ell)$. This concludes the induction and the proof. □

The theorem can be extended to non-total degree order in a similar fashion as in the polynomial case [11, Notes of Chapter 9.3]. I only sketch the proof.

Theorem 80. *Let I be a non-trivial ideal of $W_{\mathbf{x}}$, let G be a Gröbner basis of I and let $\ell \in [\![0, 2n-1]\!]$. The following statements are equivalent.*

1. *The module $W_{\mathbf{x}}/I$ has dimension $\le \ell$*
2. *$\forall A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$, $|A| = \ell + 1 \implies$ the vector space $I \cap \mathbb{K}\langle A\rangle$ is non-zero.*
3. *$\forall A \subseteq \{\mathbf{x}, \boldsymbol{\partial}\}$, $|A| = \ell + 1 \implies \exists g \in G$, $\mathrm{lm}(g) \in \mathbb{K}\langle A\rangle$*

Proof. The proofs of the implications (1) $\implies$ (2) and (2) $\implies$ (3) remain valid. For the last implication, remark that the proof can be repeated with a weight order associated to a weight containing strictly positive weights only. The end of the proof follows from the following theorem.

Theorem 81. *[114, Proposition 2.1.5]. Let $\preccurlyeq$ be any monomial order and G be a reduced Gröbner basis of I for this order. There exists a weight order $\preccurlyeq_w$ associated to a weight $w > 0$ for which any reduced Gröbner basis G_w of I with respect to $\preccurlyeq_w$ satisfies $\mathrm{lm}(G) = \mathrm{lm}(G_w)$.*

□

An algorithm for computing the dimension of a module M is obtained by searching for the smallest ℓ satisfying the third point of Theorem 80. Bernstein's inequality states that the dimension of a non-trivial $W_{\mathbf{x}}$-module is at least n. Hence, to prove that a module is holonomic it suffices to check the third point of Theorem 80 with $\ell = n$.

4.2.1.2 Holonomic $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-modules

In light of Theorem 80 applied with $\ell = n$, the definition of holonomy can be extended to $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-modules as follows.

Definition 82. Let $M = W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r/S$ be a $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module and let $e_1, \dots, e_r$ be the canonical basis of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$. The module M is said to be holonomic if for every $A \subseteq \{\mathbf{x}, \partial_{\mathbf{x}}, \partial_{\mathbf{t}}\}$ such that $|A| = n + 1$ and for every $i \in [\![1, r]\!]$, the intersection $S \cap \mathbb{K}(\mathbf{t})\langle A\rangle e_i$ is non-empty.

When $m = 0$, this definition coincides with the standard definition of holonomic modules on $W_{\mathbf{x}}$ and when $n = 0$ this definition is equivalent to the definition of D-finite modules. The third point of Theorem 80, which gave an algorithmic criterion for testing holonomy, naturally extends to this setting:

Lemma 83. *Let $M = W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r/S$ be a $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module and let G be a Gröbner basis of S. The module M is holonomic if and only if for every $I \subseteq \{\mathbf{x}, \partial_{\mathbf{x}}, \partial_{\mathbf{t}}\}$ such that $|I| = n + 1$ and for every $i \in [\![1, r]\!]$, there exists $g \in G$ such that $\mathrm{lm}(g) \in \mathbb{K}(\mathbf{t})\langle I\rangle e_i$.*

The usual equivalence between D-finiteness and holonomy, which was first stated by Takayama [131], also generalizes.

Theorem 84. *Let S be a submodule of $W_{\mathbf{x},\mathbf{t}}(\mathbf{x}, \mathbf{t})^r$. The following statements are equivalent:*

1. *$W_{\mathbf{x},\mathbf{t}}(\mathbf{x}, \mathbf{t})^r/S$ is D-finite*
2. *$W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r/(S \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r)$ is holonomic*
3. *$W^r_{\mathbf{x},\mathbf{t}}/(S \cap W^r_{\mathbf{x},\mathbf{t}})$ is holonomic*

Proof. I prove the result for $r = 1$, the general proof is similar but involves more notations.

It is well-known that (1) and (3) are equivalent: the implication (1) $\implies$ (3) is proved by constructing a suitable filtration [131, Appendix], and the converse is a straightforward application of Definition 82 when no variable $\mathbf{t}$ is present.

Thus, it suffices to prove (3) $\implies$ (2) and (2) $\implies$ (1), which are again applications of Definition 82. As an example I prove (3) $\implies$ (2). Let $A \subseteq \{\mathbf{x}, \partial_{\mathbf{x}}, \partial_{\mathbf{t}}\}$ be a set of cardinality $n+1$, then $A \cup \{\mathbf{t}\}$ is of cardinality $n+m+1$. By applying Definition 82 to $S \cap W^r_{\mathbf{x},\mathbf{t}}$ and $A \cup \{\mathbf{t}\}$, I deduce that the $\mathbb{K}$-vector space $\mathbb{K}\langle A \cup \{\mathbf{t}\}\rangle \cap \left(S \cap W^r_{\mathbf{x},\mathbf{t}}\right)$ is non-empty, which implies that the $\mathbb{K}(\mathbf{t})$-vector space $\mathbb{K}(\mathbf{t})\langle A\rangle \cap (S \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r)$ is non-empty. This proves (2) by Definition 82. □

4.2.2 Partial Weyl closure by saturation

The left saturation module of $S \subseteq W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$ with respect to the polynomial $f \in \mathbb{K}[\mathbf{x}, \mathbf{t}]$ is the left module

$$S : (f)^\infty = \left\{L \in W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r \mid \exists i \in \mathbb{N},\ f^i L \in S\right\}. \tag{4.23}$$

Theorem 85. *The set $S : (f)^\infty$ is a left submodule of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$*

Proof. The set $S : (f)^\infty$ is clearly non-empty, contained in $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$ and stable under addition. It remains to prove the stability under left multiplication by elements of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$. The stability under multiplication by $\mathbb{K}(\mathbf{t})[\mathbf{x}]$ is immediate. We now check the stability under the action of ∂_{x_i} (the case ∂_{t_i} is analogous). Let $L \in W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$ and $j \in \mathbb{N}$ such that $f^j L \in S$, then a computation shows that

$$f^{j+1}\partial_{x_i} L = \left(\partial_{x_i} f + (j+1)\frac{\partial f}{\partial x_i}\right) f^j L. \tag{4.24}$$

The last term belongs to S as it as a left module and $f^j L \in S$ by assumption. Hence, $\partial_{x_i} L \in S : (f)^\infty$, which concludes the proof. □

Theorem 74 shows that the computation of the Weyl closure is actually a saturation computation. This result extends naturally to the partial Weyl closure.

Theorem 86. *Let S be a submodule of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$, G be a finite Gröbner basis of $W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})S$ with polynomial coefficients and contents equal to one, and $f \in \mathbb{K}[\mathbf{x},\mathbf{t}]$ be a multiple of* $\operatorname{lcm}(\{\operatorname{lc}(g) \mid g \in G\})$ *then*

$$\operatorname{Cl}_{\mathbf{x}}(S) = S : (f)^\infty. \tag{4.25}$$

Proof. The implication $S : (f)^\infty \subseteq \operatorname{Cl}_{\mathbf{x}}(S)$ is straightforward by Eq. (4.23) and Eq. (4.1). Let $\preccurlyeq$ be a monomial order on $W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})^r$ and let us prove by induction on $\operatorname{lm}(L)$ that $L \in \operatorname{Cl}_{\mathbf{x}}(S)$ implies $L \in S : (f)^\infty$. Let $L \in \operatorname{Cl}_{\mathbf{x}}(S)$ and let us assume the result is true for any operator with leading monomial strictly less than $\operatorname{lm}(L)$. By properties of Gröbner basis, there exist $g \in G, \alpha \in \mathbb{N}^m, \beta \in \mathbb{N}^n$ and $q \in \mathbb{K}(\mathbf{t},\mathbf{x})$ such that

$$\operatorname{lm}\left(L - q\partial_{\mathbf{t}}^{\alpha}\partial_{\mathbf{x}}^{\beta} g\right) \prec \operatorname{lm}(L).$$

By identifying the leading coefficients of L and $q\partial_{\mathbf{t}}^{\alpha}\partial_{\mathbf{x}}^{\beta} g$ in $W_{\mathbf{x},\mathbf{t}}(\mathbf{x},\mathbf{t})^r$, we obtain that $\operatorname{lc}(g)/\operatorname{den}(q) \in \mathbb{K}(\mathbf{t})[\mathbf{x}]$, that is $\operatorname{den}(q)$ divides $\operatorname{lc}(g)$ in $\mathbb{K}(\mathbf{t})[\mathbf{x}]$ which itself divides f by hypothesis. This implies that $fq\partial_{\mathbf{t}}^{\alpha}\partial_{\mathbf{x}}^{\beta} g$ is in $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$, which proves that $q\partial_{\mathbf{t}}^{\alpha}\partial_{\mathbf{x}}^{\beta} g \in S : (f)^\infty$. By induction hypothesis, $L - q\partial_{\mathbf{t}}^{\alpha}\partial_{\mathbf{x}}^{\beta} g$ is in $S : (f)^\infty$ therefore L is also in $S : (f)^\infty$. □

Remark 87. The polynomial $\operatorname{lcm}(\{\operatorname{lc}(g) \mid g \in G\})$ is not necessarily the smallest polynomial vanishing on the singular locus of S. An example of this phenomenon is given in Saito, Sturmfels, Takayama's book [114, Example 1.4.24].

I now present a new algorithm for computing a holonomic approximation of the partial Weyl closure $\operatorname{Cl}_{\mathbf{x}}(S)$. Following Rabinowitsch's trick [108], I introduce a new variable T that can be thought of as $1/f$. This variable commutes with $\mathbf{x}, \mathbf{t}$ and satisfies for all $\ell \in \{\mathbf{x}, \mathbf{t}\}$

$$\partial_\ell T = T\partial_\ell - f_\ell T^2 \tag{4.26}$$

where f_ℓ is $\frac{\partial f}{\partial \ell}$. This defines a new algebra $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$, whose quotient by

$$U_f = \sum_{i=1}^{r} W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T](fT-1)e_i \tag{4.27}$$

is isomorphic to $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[1/f]^r$. Lemma 88 shows that left saturation modules can be computed by eliminating the variable T in a suitable submodule of this new algebra.

Lemma 88. *Let S be a submodule of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$ and f be a polynomial of $\mathbb{K}(\mathbf{t})[\mathbf{x}]$, then*

$$S : (f)^\infty = (W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]S + U_f) \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r.$$

The proof is based on the following lemma, whose immediate consequence is

$$U_f = \sum_{i=1}^{r} (fT-1) W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T] e_i \tag{4.28}$$

Lemma 89. *The ideal generated by $fT-1$ is two-sided, that is*

$$W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T](fT-1) = (fT-1)W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]$$

Proof. The direct inclusion is a consequence of the equality

$$\partial_\ell(Tf-1) = (Tf-1)(\partial_\ell + Tf_\ell)$$

for $\ell \in \{\mathbf{x}, \mathbf{t}\}$ and the reverse inclusion follows from the analog formula

$$(Tf-1)\partial_\ell = (\partial_\ell - Tf_\ell)(Tf-1).$$

□

I can now prove Lemma 88.

Proof. To simplify the notation, I prove the result for $r = 1$ only, omitting e_1. In this case, U_f becomes $U_f = W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T](fT-1)$. I start with the direct inclusion. Let $L \in S : (f)^\infty$, then there exists $i \in \mathbb{N}^*$ such that $f^i L \in S$. Hence, $T^i f^i L$ decomposes as

$$T^i f^i L = (T^i f^i - 1)L + L \in W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]S.$$

Besides $fT-1$ divides $T^i f^i - 1$, which proves by the remark before Lemma 89 that $(T^i f^i - 1)$ is in U_f. This proves that L is in $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]S + U_f$. Because L is in $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$ by definition, it belongs to the intersection.

I prove now the reverse inclusion. Let $L \in (W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]S + U_f) \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$, then L decomposes as

$$L = \sum_{i=1}^{p} T^i g_i + a$$

with $p \in \mathbb{N}$, $g_1, \dots, g_p \in S$ and $a \in U_f$. Then

$$f^p L = \sum_{i=1}^{p} (T^i f^i - 1) f^{p-i} g_i + \sum_{i=1}^{p} f^{p-i} g_i + f^p a = g + \tilde{a}$$

with $g = \sum_i f^{p-i} g_i \in S$ and $\tilde{a} = f^p a + \sum_{i=1}^{p} (T^i f^i - 1) f^{p-i} g_i$. Using again that $fT-1$ divides $T^i f^i - 1$ and Lemma 89, I obtain that $\tilde{a}$ is in U_f. However, neither $f^p L$ nor g involves the variable T, thus $\tilde{a}$ is in $U_f \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$ and must be zero. This proves that $f^p L$ is in S. □

I would like to use Lemma 88 to compute saturations. In the commutative case this would be done by fixing a monomial order eliminating T and computing a Gröbner basis with respect to this order. This is however not possible in this setting as there is no order on $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ eliminating T that is compatible with multiplication, because of the anti-commutation rule in Eq. (4.26). This problem is tackled by seeing $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ as an infinite-rank $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module and performing Gröbner basis computation on truncations of this infinite rank module. This idea of computing Gröbner bases of truncations of an infinite-rank module has already been used by Takayama in his integration algorithm [128]. Similarly to his algorithm, the termination criterion that I will use is based on holonomy, and it does not guarantee that the whole saturation module will be obtained.

A basis of the $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ is $\{T^j e_i \mid j \in \mathbb{N}, i \in [\![1,r]\!]\}$. The basis elements $T^0 e_i$ will be identified with e_i. I define the degree of $g \in W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ in T, denoted $\deg_T(g)$, as the largest integer j that appears in the decomposition of g with non-zero coefficient in this basis. Let G be a finite subset of $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ and ℓ be the largest degree in T appearing in G. Then $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})G$ is included in the finitely generated $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module $\sum_{i=0}^{\ell}\sum_{j=1}^{r} W_{\mathbf{x},\mathbf{t}}(\mathbf{t})T^i e_j$ in which it is possible to compute Gröbner bases over $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$. Coupled with Lemma 88 this leads to Algorithm 12.

Algorithm 12 Partial Weyl closure

Input:
- a presentation $S \subseteq W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$ of a D-finite module
- a polynomial $f \in \mathbb{K}[\mathbf{x},\mathbf{t}]$ vanishing on the singular locus of S

Output: a holonomic approximation of the partial Weyl closure $\mathrm{Cl}_{\mathbf{x}}(S)$

1 Fix an order on $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]^r$ eliminating T
2 $H \leftarrow \mathrm{gens}(S) \cup \{(fT-1)e_i \mid i = 1,\dots,r\}$
3 $G \leftarrow H;\quad s \leftarrow 0$
4 **while** $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r/(W_{\mathbf{x},\mathbf{t}}(\mathbf{t})G \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r)$ is not holonomic
5 $\quad G \leftarrow \mathrm{GröbnerBasis}\big(\{T^i g \mid i \in \mathbb{N},\, g \in H,\, \deg_T(T^i g) \le s\}\big)$
6 $\quad s \leftarrow s+1$
7 **return** $G \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$

Theorem 90. *Algorithm 12 terminates and returns generators G' of a $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module included in $\mathrm{Cl}_{\mathbf{x}}(S)$ such that $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r/W_{\mathbf{x},\mathbf{t}}(\mathbf{t})G'$ is holonomic.*

Proof. Let us first remark that $\mathrm{Cl}_{\mathbf{x}}(S)$ is equal to $S:(f)^\infty$ by Theorem 86, and that is admits a finite Gröbner basis. By Lemma 88 this Gröbner basis can be obtained from a finite number of generators of the $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})$-module $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})[T]S + U_f$. Besides $S:(f)^\infty$ is holonomic by Theorem 84 which proves that the algorithm terminates. Its correction is also a consequence of Theorem 84, Theorem 86 and Lemma 88. □

Remark 91. Let $(G_s)_{s\in\mathbb{N}}$ be the sequence of Gröbner bases computed on line 5 of Algorithm 12 and let $(G'_s)_{s\in\mathbb{N}}$ be the sequence defined by $G'_s = G_s \cap W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r$. Two other possible termination criteria for Algorithm 12 are:

- stopping at the first s' such that $G'_{s'} = G'_{s'+1}$ and $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})^r / W_{\mathbf{x},\mathbf{t}}(\mathbf{t})G'_{s'}$ is holonomic,
- stopping at the first $s'+h$ for which $W_{\mathbf{x},\mathbf{t}}(\mathbf{t})/W_{\mathbf{x},\mathbf{t}}(\mathbf{t})G'_{s'}$ is holonomic, where $h \in \mathbb{N}$ is a prescribed integer.

At present, however, I do not know any criterion that guarantees the output coincides with $\mathrm{Cl}_{\mathbf{x}}(S)$ without explicitly computing a b-function.

4.2.3 Application

The integration algorithm presented in Chapter 3 requires computing from a holonomic $W_{\mathbf{x},t}$-module P a presentation of the $W_{\mathbf{x}}(t)$-module $M = \mathbb{K}(t) \otimes_{\mathbb{K}[t]} P$ with a derivation map δ_t on M. An algorithm to perform this computation was described in Section 3.4.3. In practice, however, a holonomic $W_{\mathbf{x},t}$-module P is often obtained from its D-finite counterpart $W_{\mathbf{x},t}(\mathbf{x},t)P$ via Weyl closure. By using partial Weyl closure, one can instead compute M and the derivation δ_t directly from $W_{\mathbf{x},t}(\mathbf{x},t)P$. The algorithm is presented in Algorithm 13. When Algorithm 12 does not return the full partial closure, the algorithm computes instead a module M' containing M as a submodule.

Algorithm 13 Computation of M and δ_t

Input: a presentation $S \subseteq W_{\mathbf{x},t}(\mathbf{x},t)^s$ of the D-finite module $W_{\mathbf{x},t}(\mathbf{x},t)P$.
Output:

- a presentation $S' \subseteq W_{\mathbf{x}}(t)^r$ of the $W_{\mathbf{x}}(t)$-module $M = \mathbb{K}(t) \otimes_{\mathbb{K}[t]} P$
- a derivation $\delta_t : M \to M$ such that $\varphi(\partial_t m) = \delta_t(\varphi(m))$ for all $m \in M$, where $\varphi : P \to M$ is the natural inclusion

Let G be a generating set of S with coefficients in $\mathbb{K}(t)[\mathbf{x}]$
Compute the partial Weyl closure $\mathrm{Cl}_{\mathbf{x}}(W_{\mathbf{x},t}(t)G)$ by Algorithm 12
Let G' be a reduced Gröbner basis of $\mathrm{Cl}_{\mathbf{x}}(W_{\mathbf{x},t}(t)G)$ for an order eliminating ∂_t
For $i \in \{1,\dots,s\}$, let g_i be the unique element of G' s.t. $\mathrm{lm}(g) = \partial_t^{\ell_i} e_i$ for some $\ell_i \in \mathbb{N}$
Let $\ell = \max(\ell_1,\dots,\ell_s)$
$G'' \leftarrow \emptyset$
for each g in $G' \setminus \{g_1,\dots,g_s\}$
    $G'' \leftarrow G'' \cup \{g,\dots,\partial_t^{\ell-\deg_{\partial_t}(g)} g\}$
Define δ_t by $\delta_t(m) = \mathrm{LRem}(m, g_1,\dots,g_s)$
return G'' and δ_t

Theorem 92. *Algorithm 13 terminates. Moreover, if Algorithm 12 computes the full partial Weyl closure, then Algorithm 13 is correct.*

Proof. The termination follows directly from the termination of Algorithm 12. For correctness, observe that

$$W_{\mathbf{x},t}(t)^r / \operatorname{Cl}_{\mathbf{x}}(W_{\mathbf{x},t}(t)G) \simeq \mathbb{K}(t) \otimes_{\mathbb{K}[\mathbf{x}]} P$$

is holonomic as a $W_{\mathbf{x}}(t)$-module by Lemma 54. The existence of the g_i then follows from the noetherianity of this module (see also Lemma 55), while uniqueness follows from the property of a reduced Gröbner basis. Finally, the fact that δ_t satisfies the required property, and that $W_{\mathbf{x}}(t)^s/W_{\mathbf{x}}(t)G$ is isomorphic to M', are consequences of Lemma 56. □

4.3 Timings

This section presents benchmarks of a preliminary implementation [27] of my partial Weyl closure algorithm (Algorithm 12). The results are compared with two existing implementations of Tsai's algorithm, the first one is part of Singular's [56] Plural package [90], while the second one is part of Macaulay2's [70] Dmodules package [91]. Since the optimization of my code is still a work in progress, the results reported here are provisional and will likely change with future versions of the package. Nevertheless, these preliminary experiments suggest that Algorithm 12 will likely have meaningful applications.

Comparing different algorithms implemented by different authors in different programming languages is always delicate. In this case, the comparison is even more delicate as both approaches rely on Gröbner basis computations, whose efficiency is highly dependent on the underlying implementation. For this reason, I first benchmark the Gröbner basis implementations used in these three systems.

Benchmarks were performed on a cluster, where each node is equipped with two Intel(R) Xeon(R) E5-2660 v2 CPUs and 256 GB of RAM. Each CPU has 10 cores with hyper-threading (20 threads). All computations were run on a single thread when the cluster was not under heavy load, although the workload manager may have scheduled multiple threads on the same CPU, making the effective memory available per thread difficult to estimate.

4.3.1 Gröbner bases

In Plural, Gröbner bases are computed using slimgb [26], an algorithm specifically designed to limit intermediate expression growth. Macaulay2's Dmodules package, on the other hand, uses the standard Buchberger algorithm, invoked through the "gbw" command. To the best of my knowledge, users may specify only a weight vector, but not a general monomial order. The Julia implementation relies on the F4 algorithm [65]. All three implementations are tested over the finite field $\mathbb{F}_{536870909}$ as well as over $\mathbb{Q}$.

Gröbner bases over a finite field

The performances of the three implementations are compared in Table 4.1 over the field $\mathbb{F}_{536870909}$ on the following set of examples. 7reg and 8reg correspond to the $W_{\mathbf{x}}(t)$-module S described in Theorem 60 where t was evaluated at 42 for $k = 7$ and $k = 8$ respectively. Beukers3 and Beukers4 correspond to the computation of the annihilator of e^f where

$$f = x_4^6 - (x_4^2 - x_1x_2)x_3x_4^3 - 4049x_1x_2x_3(x_4 - x_1)(x_4 - x_2)(x_4 - x_3) \tag{4.29}$$

is the homogenized polynomial appearing in Eq. (1.17). In Beukers3, the variable x_4 has been evaluated at 1987. Exp1 corresponds to the computation of the annihilator of e^g where

$$g = (x_1^2 + x_2^2 + x_3^2)(x_1^4 + x_2^4 + x_3^4). \tag{4.30}$$

For the above examples, the parenthesis (gl) in Table 4.1 indicates that the order

$$\text{grevlex}(x_1, \ldots, x_\ell, \partial_1, \ldots, \partial_\ell) \tag{4.31}$$

was used in Julia and Singular, while the weight vector $(\mathbf{1}, \mathbf{1})$ was chosen in Macaulay2. Similarly, the parenthesis (elim) indicates that the block order

$$\text{lex}(x_1, \ldots, x_\ell) > \text{lex}(\partial_1, \ldots, \partial_\ell) \tag{4.32}$$

was used in Julia and Singular, while the weight vector $(\mathbf{1}, \mathbf{0})$ was chosen in Macaulay2.

The examples ha2, reiffen11, and ucha2 correspond to the computation in line 3 of Algorithm 6 on input the polynomials

$$x_1x_2x_3x_4(x_1 + x_2)(x_1 + x_3)(x_1 + x_4) \tag{4.33}$$
$$x_1^{11} + x_2^{11} + x_1x_2^{10} \tag{4.34}$$
$$x_1^4 + x_2^5 + x_1x_2^4 \tag{4.35}$$
$$\tag{4.36}$$

respectively. The first two examples are taken from [89], and the third from [88]. All three are computed with the block order

$$\text{grevlex}(u, v) > \text{grevlex}(s, t, x_1, \ldots, x_\ell, \partial_t, \partial_1, \ldots, \partial_\ell). \tag{4.37}$$

Since Julia uses Just-In-Time (JIT) compilation, the compilation overhead of approximately 7-8 seconds is included in the timings. We observe that Macaulay's implementation falls behind both Julia and Singular, which achieve comparable performances. Julia's f4 tends to perform slightly better with grevlex orders, whereas Singular's slimgb is more efficient for elimination block orders. The sizes reported in Table 4.1 correspond to the number of elements in the reduced Gröbner basis returned as output. This is the only check performed to verify that the outputs are the same. Note that Macaulay2 may return different Gröbner bases than the other two making the comparison less meaningful.

Table 4.1: Comparison of three Gröbner basis implementation over the field $\mathbb{F}_{536870909}$. The column time indicates the computation time and the column size the number of operators in the output reduced Gröbner basis.

	Julia-f4		Macaulay2		Singular	
	time	size	time	size	time	size
7reg (elim)	8s	23	<1s	23	<1s	23
7reg (gl)	9s	23	<1s	23	<1s	23
8reg (elim)	9s	42	1s	42	4s	42
8reg (gl)	4min 49s	59	1h 38min 10s	59	5min 8s	59
Beukers3 (elim)	13s	30	<1s	29	1s	30
Beukers3 (gl)	6s	28	<1s	28	<1s	28
Beukers4 (elim)	1min 25s	75	20h 26min 49s	162	31s	75
Beukers4 (gl)	6min 9s	164	1h 38min 15s	164	31min 27s	164
Exp1 (elim)	10s	121	<1s	27	10s	121
Exp1 (gl)	7s	27	<1s	27	<1s	27
ha2	1h 56min 40s	334	1h 49min 35s	499	1h 43min 56s	334
ucha2	7s	23	<1s	38	<1s	23
reiffen11	7s	11	<1s	24	<1s	11

Gröbner basis over $\mathbb{Q}$

The same set of examples as in the previous paragraph is used in Table 4.2 to compare the three implementations over $\mathbb{Q}$. Singular and Macaulay2 perform the computation directly over the integers, whereas my Julia implementation proceeds modulo several primes and then reconstructs the result in $\mathbb{Q}$. To my surprise, this does not yield good performance. It will have to be investigated and possibly changed in future versions of my package.

We observe that Singular's implementation performs significantly better than Macaulay2, which in turn outperforms my Julia implementation. The label "INPUT OVFL" indicates that an integer in the input ideal exceeded 32 bits, causing an overflow and an incorrect result. The label "MEM" indicates that the program ran out of memory.

4.3.2 Weyl closure

The three implementations of the Weyl closure are now compared in Table 4.3. Recall that Algorithm 12 only computes an approximation of the Weyl closure. The benchmark set consists in computing the Weyl closure of the D-finite annihilator of the following

Table 4.2: Comparison of three Gröbner basis implementation over the field $\mathbb{Q}$. The column time indicates the computation time and the column size the number of operators in the output reduced Gröbner basis

	Julia-f4		Macaulay2		Singular	
	time	size	time	size	time	size
7reg (elim)	9s	23	<1s	23	1s	23
7reg (gl)	9s	23	<1s	23	<1s	23
8reg (elim)	11s	42	1s	42	21s	42
8reg (gl)	14h 5min 24s	59	1h 45min 57s	59	14min 43s	59
Beukers3 (elim)	3min 22s	30	<1s	29	INPUT	OVFL
Beukers3 (gl)	9s	28	<1s	28	INPUT	OVFL
Beukers4 (elim)	>4j		MEM		39s	75
Beukers4 (gl)	26h 35min 38s	164	2h 19min 57s	164	1h 29min 33s	164
Exp1 (elim)	4h 22min 19s	194	<1s	27	12s	121
Exp1 (gl)	9s	27	<1s	27	<1s	27
ha2	>4j		1h 48min 32s	499	1h 48min 42s	334
ucha2	8s	23	<1s	38	<1s	23
reiffen11	9s	11	<1s	24	<1s	11

integrands:

$$\int_0^\infty x \ \mathrm{J}_1(ax) \ \mathrm{I}_1(ax) \ \mathrm{Y}_0(x) \ \mathrm{K}_0(x) \, dx \qquad \mathrm{grevlex}(a, x, \partial_a, \partial_x) \tag{4.38}$$

$$\int_0^\infty \int_0^\infty \mathrm{J}_1(x) \ \mathrm{J}_1(y) \ \mathrm{J}_2\big(t\sqrt{xy}\big) \frac{dx\,dy}{e^{x+y}} \qquad \mathrm{grevlex}(t, x, y, \partial_t, \partial_x, \partial_y) \tag{4.39}$$

$$\int_a^b \arccos\left(\frac{x}{\sqrt{(a+b)x - ab}}\right) dx \qquad \mathrm{grevlex}(x, a, b, \partial_x, \partial_a, \partial_b) \tag{4.40}$$

$$\int_0^\infty \mathrm{Ai}\big(2^{2/3}(x^2+u)\big)\, dx \qquad \mathrm{grevlex}(x, u, \partial_x, \partial_u) \tag{4.41}$$

$$\int \frac{dx\,dy}{\big((y^2+1)(x^2+1)\,t - xy\big)} \qquad \mathrm{grevlex}(t, x, y, \partial_t, \partial_x, \partial_y) \tag{4.42}$$

$$\int \frac{dx\,dy\,dz}{1-(1-xy)z - txyz(1-x)(1-y)(1-z)} \qquad \mathrm{grevlex}(t, x, y, z, \partial_t, \partial_x, \partial_y, \partial_z) \tag{4.43}$$

$$\int \frac{dxdydzdw}{q(x,y,z,w)t - xyzw} \qquad \mathrm{grevlex}(t, x, y, z, w, \partial_t, \partial_x, \partial_y, \partial_z, \partial_w) \tag{4.44}$$

Table 4.3: Comparison of three algorithms for the Weyl closure. The column time indicates the computation time and the column size the number of operators in the output Gröbner basis. Julia and Singular return reduced Gröbner bases but not Macaulay2.

	Julia		Macaulay2		Singular	
	time	size	time	size	time	size
Eq. (4.38)	4min 38s	19	14s	396	8s	32
Eq. (4.39)	27s	8	50s	223	5min 57s	13
Eq. (4.40)	20s	10	ERR	ERR	1min 50s	10
Eq. (4.41)	20s	4	<1s	29	<1s	4
Eq. (4.42)	20s	13	MEM	MEM	3s	13
Eq. (4.43)	22s	37	MEM	MEM	1min 50s	15
Eq. (4.44)	97h 38min 54s	143	MEM	MEM	MEM	MEM

where on the last equation

$$q = (x^2y^2z^2w^2+x^2y^2zw^2+x^2yz^2w^2+xy^2zw^2+xy^2w^2+xyzw^2+xyz+xz^2+xy+xz+y+z).$$

The symbols $\mathrm{J}, \mathrm{I}, \mathrm{Y}, \mathrm{K}$ denote the usual Bessel functions and Ai denotes the Airy function of the first kind. Eqs. (4.38) and (4.39) were taken from Chyzak's habilitation thesis [48], Eq. (4.43) appeared in [14], Eq. (4.42) belongs to a family of integrals related to the enumeration of small step walks in the quarter plane [21] (with a polynomial factor in the numerator removed), Eq. (4.40) is from the example base of [20], Eq. (4.41) is from Koutschan's database [85], and Eq. (4.44) is the period of a Calaby-Yau 3-fold appearing in the database[1] associated to the article [10] (Topology #42, polytope v25.59).

The compilation time included in the timings of the Julia implementation seems to be around 20 seconds. If we disregard this overhead, Julia outperforms the other two algorithms on all examples except Eq. (4.38). However, on both examples involving Bessel functions, Julia does not compute the complete Weyl closure. My Julia implementation computes the output Gröbner basis modulo several primes and then reconstructs its coefficients in $\mathbb{Q}$. Since the same strategy was used for Gröbner basis computations over $\mathbb{Q}$ and led to poor timings, the speed-ups obtained on Eqs. (4.39), (4.40), (4.43) and (4.44) suggest that this algorithm holds promising potential for practical applications.

Note that the holonomic annihilator of rational functions can be obtained by other means. For example, the annihilator of the integrand in Eq. (4.43) can be obtained in roughly 30 seconds in Singular.

[1] http://hep.itp.tuwien.ac.at/~kreuzer/math/0802/

5 Perspectives

The time spent on a PhD always feels too short, and inevitably there are ideas or projects I didn't have time to fully explore, or even to start. This chapter is meant to share some of them:

1. Implementation. Much work remains to be done on the implementation side. In computer algebra, the success of an algorithm almost always goes hand in hand with a user-friendly and efficient implementation.

2. Reductions. I have had several ideas for alternative reductions for multivariate integration that could potentially lead to more efficient algorithms. The first might allow us to recover the same performance for the integration of rational functions as Lairez's algorithm. The second could address an efficiency issue that is observed when integrating polynomials. The third and final idea proposes a reduction that could compute minimal telescopers.

3. Binomial sums. One natural extension of my PhD, suggested by my advisor Pierre Lairez, concerns the class of binomial sums. With Alin Bostan and Bruno Salvy [24], they proposed an algorithm for rewriting these sums as integrals of rational functions, which could then be computed with Lairez's integration algorithm. Since the multivariate integration algorithm presented in Chapter 3 is more general than Lairez's algorithm, it is natural to explore whether this approach can be extended to a broader class of sums.

5.1 Implementation

One of the first directions I plan to pursue after my PhD is to provide a more efficient implementation of Algorithm 12, which in turn requires a better implementation of Gröbner bases over $\mathbb{Q}$ and over parametric fields. For Gröbner basis over $\mathbb{Q}$, it is not clear at this stage whether the main source of inefficiency lies in the reconstruction strategy from modular computations or in shortcomings of the current implementation; this question will require further investigation. For Gröbner bases over parametric fields, the current implementation relies on a generic representation of the coefficient field, which can cause severe coefficient swell. To address this, specialized subroutines should be designed for this case, as well as for computations over $\mathbb{Q}$. In addition, I plan to extend the benchmarks presented in Section 4.3 by incorporating a wider range of examples. Finally, I intend to redesign my package `MultivariateCreativeTelescoping.jl` [27] to make it more robust and user-friendly.

5.2 Improving reductions for multivariate integration

The family of reduction procedures described in Section 3.3 and used in Algorithm 5 is not entirely satisfactory for two reasons. First, it does not perform as well as the Griffiths-Dwork reduction (see Section 3.3.4) and second, it performs very poorly for integrating polynomials, as shown by Example 59. Although employing creative telescoping for integrating polynomials is excessive, comparable inefficiencies may well occur in other applications yet to be identified. This section presents three ideas to build more efficient reductions or family of reductions, two of which are based on the following theorem.

Theorem 93. *Let S be a submodule of $W_{\mathbf{x}}^r$, let $g_1, \dots, g_\ell$ be generators of S and let J be the left $\mathbb{K}[\mathbf{x}]$-module $\sum_{i=1}^{\ell} \mathbb{K}[\mathbf{x}] g_i$. Then we have the following equality of $\mathbb{K}$-vector space:*

$$S + \boldsymbol{\partial} W_n^r = J + \boldsymbol{\partial} W_n^r. \tag{5.1}$$

Proof. Clearly $J + \boldsymbol{\partial} W_n^r \subseteq S + \boldsymbol{\partial} W_n^r$. Let $a \in S + \boldsymbol{\partial} W_n^r$, then $a = \sum_{i=1}^{\ell} (q_i + d_i) g_i + d$ with $q_i \in \mathbb{K}[\mathbf{x}]$ and $d, d_i \in \boldsymbol{\partial} W_n^r$. Reordering the sum as

$$a = \left(\sum_{i=1}^{\ell} q_i d_i g_i \right) + \left(d + \sum_{i=1}^{\ell} d_i g_i \right) \tag{5.2}$$

we obtain $a \in J + \boldsymbol{\partial} W_n^r$. □

A direct corollary of this theorem is an alternative description of the integral of the module $M = W_{\mathbf{x}}^r / S$ as

$$M / \boldsymbol{\partial} M \simeq W_{\mathbf{x}}^r / (J + \boldsymbol{\partial} W_{\mathbf{x}}^r). \tag{5.3}$$

As a result it seems possible to perform integration using a Gröbner basis of the $\mathbb{K}[\mathbf{x}]$-module J instead of a Gröbner basis of the $W_{\mathbf{x}}$-module S. Experiments indicate that the former is easier to compute when S is the annihilator of the exponential of a homogeneous polynomial, as in the Griffiths-Dwork reduction, but not when S is the module defined in Theorem 60. This suggests that both formulations may be useful, depending on the situation.

5.2.1 A variant of the family of reductions $([\,.\,]_\eta)_\eta$

The isomorphism in Theorem 93 offers an alternative way to represent the integral of a module. In this representation, the $W_{\mathbf{x}}$-module S is replaced by the $\mathbb{K}[\mathbf{x}]$-module J. It is then natural to consider how the reduction family defined in Section 2.4 might be adapted to J. Let G be a Gröbner basis of the $\mathbb{K}[\mathbf{x}]$-module J. The binary relation $\to_1$ introduced in Section 2.4 can be adapted as follows: given $a \in W_{\mathbf{x}}^r$, $\lambda \in \mathbb{K}$, $m \in M_{\mathbf{x}}$, and $g \in G$, we write

$$a \to_1 a - \lambda m g \tag{5.4}$$

if $\operatorname{lm}(mg)$ is in the support of a but not in the support of $a - \lambda m g$.

With this modified version of $\to_1$, the definition of $[.]$ extends naturally, as does the statement of Theorem 37. However, the criterion provided by Lemma 38, which enables more efficient computation of irreducible elements, does not appear to generalize.

When S is the annihilator of the exponential of a homogeneous polynomial p, as in the Griffiths-Dwork reduction, and the generators $g_1, \dots, g_n$ are chosen to be $\partial_1 - \frac{\partial p}{\partial x_1}, \dots, \partial_n - \frac{\partial p}{\partial x_n}$, computing a Gröbner basis of J is equivalent to computing a Gröbner basis of the ideal generated by the partial derivatives of p: $\frac{\partial p}{\partial x_1}, \dots, \frac{\partial p}{\partial x_n}$. Hence, we retrieve the input of the Griffiths-Dwork reduction, thus solving the issue highlighted at the end of Section 3.3.4. With some effort, it should be possible to adapt the reduction algorithms to recover the same performance as in the Griffiths-Dwork reduction.

5.2.2 A reduction using shift algebras

This section presents preliminary ideas for a new reduction procedure with calculations in shift algebras. It builds on the isomorphism presented in Theorem 93 and can be regarded as a multivariate generalization of the polynomial reduction used in several D-finite creative telescoping algorithms, including the one presented in Chapter 2 (see Section 2.4.4). A related approach has already been employed in physics for the computation of Feynman integrals [7, 6]. For better readability, I restrict to the case $r = 1$; the general case follows analogously.

When computing normal forms modulo $J + \boldsymbol{\partial} W_{\mathbf{x}}$, terms involving ∂_i can be easily reduced using operators from $\boldsymbol{\partial} W_{\mathbf{x}}$. The main difficulty lies in finding the polynomials that belong to $J + \boldsymbol{\partial} W_{\mathbf{x}}$. For $\boldsymbol{a} \in \mathbb{N}^n$, we remark that reducing $\mathbf{x}^{\boldsymbol{a}} g_i$ modulo $\boldsymbol{\partial} W_{\mathbf{x}}$ amounts to evaluate the adjoint [66, p.231] of g_i denoted g_i^* at $\mathbf{x}^{\boldsymbol{a}}$, that is

$$\operatorname{Rem}(\mathbf{x}^{\boldsymbol{a}} g_i, \boldsymbol{\partial} W_{\mathbf{x}}) = g_i^*(\mathbf{x}^{\boldsymbol{a}}). \tag{5.5}$$

The following theorem characterizes the polynomials in $J + \boldsymbol{\partial} W_{\mathbf{x}}$ by means of adjoints.

Theorem 94. *Let $g_1, \dots, g_\ell \in W_{\mathbf{x}}$ be generators of the $\mathbb{K}[\mathbf{x}]$-module J. The $\mathbb{K}$-vector space $(J + \boldsymbol{\partial} W_{\mathbf{x}}) \cap \mathbb{K}[\mathbf{x}]$ is generated by $g_i^*(\mathbf{x}^{\boldsymbol{a}})$ for $i \in \{1, \dots, \ell\}$ and $\boldsymbol{a} \in \mathbb{N}^n$.*

Proof. First observe that $g_i^*(\mathbf{x}^{\boldsymbol{a}})$ belongs to $(J + \boldsymbol{\partial} W_{\mathbf{x}}) \cap \mathbb{K}[\mathbf{x}]$ for any i and $\boldsymbol{a} \in \mathbb{N}^n$. Conversely, let $p \in (J + \boldsymbol{\partial} W_{\mathbf{x}}) \cap \mathbb{K}[\mathbf{x}]$. Then p admits a decomposition of the form

$$p = \sum_{i,\boldsymbol{a}} \lambda_{i,\boldsymbol{a}} \mathbf{x}^{\boldsymbol{a}} g_i + d \tag{5.6}$$

where $\lambda_{i,\boldsymbol{a}} \in \mathbb{K}$ and $d \in \boldsymbol{\partial} W_{\mathbf{x}}$. Reducing (5.6) modulo $\boldsymbol{\partial} W_{\mathbf{x}}$ gives

$$p = \sum_{i,\boldsymbol{a}} \lambda_{i,\boldsymbol{a}} g_i^*(\mathbf{x}^{\boldsymbol{a}}) \tag{5.7}$$

since p is in $\mathbb{K}[\mathbf{x}]$ and d vanishes modulo $\boldsymbol{\partial} W_{\mathbf{x}}$. □

The idea of the reduction is to obtain indicial equations by replacing $\boldsymbol{a}$ with a symbolic n-tuple $\boldsymbol{\alpha}$. Consider the $\mathbb{K}$-vector space

$$N = \sum_{\boldsymbol{a}} \sum_{i=1}^{\ell} \mathbb{K} g_i^*(\mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{a}}). \tag{5.8}$$

One verifies that N has a natural structure of $\mathbb{K}[\boldsymbol{\alpha}]\langle S_{\boldsymbol{\alpha}}\rangle$-module, and that there exists $\boldsymbol{b} \in \mathbb{N}^n$ such that

$$N \subseteq \mathbb{K}[\boldsymbol{\alpha}][\mathbf{x}] \cdot \mathbf{x}^{\boldsymbol{\alpha}-\boldsymbol{b}}. \tag{5.9}$$

Let $\preccurlyeq$ be a monomial order on $\mathbb{K}[\mathbf{x}]$. Any element of $g \in N$ gives rise to an equation of the form:

$$g(\mathbf{x}^{\boldsymbol{\alpha}}) = (c_g(\boldsymbol{\alpha})\mathbf{x}^{\boldsymbol{a_g}} + \text{lower-order terms}) \cdot \mathbf{x}^{\boldsymbol{\alpha}-\boldsymbol{b}}, \tag{5.10}$$

where $c_g(\boldsymbol{\alpha}) \in \mathbb{K}[\boldsymbol{\alpha}]$ plays the role of an indicial equation and $\boldsymbol{a_g} \in \mathbb{N}^n$. This equation can be used to find polynomials in $(J + \boldsymbol{\partial} W_{\mathbf{x}})$ by replacing $\boldsymbol{\alpha}$ with an n-tuple of integer values. The leading term of $g(\boldsymbol{a})$ can be predicted, except when its leading coefficient $c_g(\boldsymbol{a})$ vanishes. A natural idea to define a reduction modulo $(J + \boldsymbol{\partial} W_{\mathbf{x}}) \cap \mathbb{K}[\mathbf{x}]$ is to compute a Gröbner basis of N for the shift structure and to use the indicial equations obtained from this Gröbner basis to build a reduction in a similar fashion as in Algorithm 3.

I face two difficulties in turning this idea into an algorithm. The first one is theoretical: finding the values $\boldsymbol{a}$ for which the coefficients $c_g(\boldsymbol{\alpha}) \in \mathbb{K}[\boldsymbol{\alpha}]$ vanish amounts to finding the integer solutions of a multivariate polynomial, which is known to be undecidable, as stated by Matiyasevich's theorem. The second is computational: preliminary experiments in Maple indicated that computing a Gröbner basis of the module N may be prohibitively expensive. Similar observations were reported in [7, 6].

The first problem may not be as serious in practice as it initially appears: the polynomials $c_g(\boldsymbol{\alpha})$ could have a nice structure for some restricted classes of integrands, or cancellations might be sufficiently rare to be ignored, leading to a viable heuristic algorithm. The second problem requires further investigation to determine whether these Gröbner bases are intrinsically hard to compute, and if not, whether it would be possible to improve existing Gröbner basis implementations for shift algebras.

5.2.3 Reduction using order associated to a weight vector

Another idea for constructing a reduction modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}$ (again assuming $r = 1$) is to use Gröbner bases with respect to an order defined by a weight vector $\boldsymbol{w} \in (\mathbb{R}_+^*)^n$ [114, p.6], combined with a suitable monomial order as a tie-breaker. More precisely, let $\preccurlyeq$ be a monomial order eliminating $\mathbf{x}$ and define $\preccurlyeq_{(\boldsymbol{w},-\boldsymbol{w})}$ by

$$\mathbf{x}^{\boldsymbol{a}}\boldsymbol{\partial}^{\boldsymbol{b}} \preccurlyeq_{(\boldsymbol{w},-\boldsymbol{w})} \mathbf{x}^{\boldsymbol{a}'}\boldsymbol{\partial}^{\boldsymbol{b}'} \quad \Leftrightarrow \quad \begin{cases} (\boldsymbol{a}-\boldsymbol{b}) \cdot \boldsymbol{w} < (\boldsymbol{a}'-\boldsymbol{b}') \cdot \boldsymbol{w}, \\ \text{or } (\boldsymbol{a}-\boldsymbol{b}) \cdot \boldsymbol{w} = (\boldsymbol{a}'-\boldsymbol{b}') \cdot \boldsymbol{w} \;\text{ and }\; \mathbf{x}^{\boldsymbol{a}}\boldsymbol{\partial}^{\boldsymbol{b}} \preccurlyeq \mathbf{x}^{\boldsymbol{a}'}\boldsymbol{\partial}^{\boldsymbol{b}'}. \end{cases} \tag{5.11}$$

Here the variable x_i is assigned a weight w_i, while ∂_i is assigned a weight $-w_i$. The presence of negative weights makes this order not well-founded, that is there exists strictly decreasing sequences. One such example is the sequence $(\partial_1^\ell)_{\ell\in\mathbb{N}}$. Nevertheless, finite Gröbner bases still exist and can be computed by homogeneization of the Weyl algebra $W_{\mathbf{x}}$ [114, Chap. 1.2]. The next theorem shows that, given such a Gröbner basis, one can define a reduction procedure that reduces modulo $S + \boldsymbol{\partial} W_{\mathbf{x}}$ up to a finite-dimensional vector space.

Theorem 95. *Let G be a Gröbner basis of S for $\preccurlyeq_{(\boldsymbol{w},-\boldsymbol{w})}$. There exists an integer s_0 such that for any $\boldsymbol{a} \in \mathbb{N}^n$ with $\boldsymbol{a}\cdot\boldsymbol{w} > s_0$, there exists $g \in S$ and $m \in M_{\mathbf{x}}$ such that $\mathrm{lm}_{\preccurlyeq_{(\boldsymbol{w},-\boldsymbol{w})}}(mg) = \mathbf{x}^{\boldsymbol{a}}$.*

The proof is a direct adaptation of a very similar lemma found in Saito, Sturmfels and Takayama's book [114, Lemma 5.1.8]. The integer s_0 in the theorem corresponds to the largest root of the b-function associated to the ideal I with respect to the weight vector $\boldsymbol{w}$ [114, Definition 5.1.1]. Whenever no reducer $g \in G$ exists for a monomial $\mathbf{x}^{\boldsymbol{a}}$, the corresponding number $\boldsymbol{w}\cdot\boldsymbol{a}$ is a root of b. The converse implication, however does not hold. As a consequence, if one is willing to compute the b-function, then one can also obtain a reduction that computes normal forms. This, in turn, yields a creative telescoping algorithm capable of computing minimal telescopers.

The efficiency of this reduction depends entirely on the performance of Gröbner basis computations for weight orders. I have not investigated this subject during my PhD, and I therefore cannot assess the practical performance of such a reduction.

5.3 Extending the class of binomial sums that can be represented as integrals

Creative telescoping algorithms based on reductions do not compute the certificate in general. While this approach is often justified in the case of integration, it becomes problematic for summation, where singularities in the certificate frequently arise. An idea of Bostan, Lairez, and Salvy [24] consists in expressing a certain class of sums as formal residues of rational functions, thereby turning a summation problem into an integration problem with natural boundaries. They called the class of sums for which their method applies the class of binomial sums, and defined it inductively as follows.

Definition 96. The class of binomial sum is defined inductively by:

- the sequence $(\delta_n)_{n\in\mathbb{Z}}$ defined by $\delta_0 = 1$ and $\delta_n = 0$ for $n \neq 0$ is a binomial sum,
- the geometric sequences $(C^n)_{n\in\mathbb{Z}}$ for all $C \in \mathbb{K}\setminus\{0\}$ are binomial sums,
- the binomial sequence $\left(\binom{n}{k}\right)_{n,k\in\mathbb{Z}}$ is a binomial sum,
- the product and any $\mathbb{K}$-linear combination of two binomial sums are binomial sums,
- if $\lambda : \mathbb{Z}^d \to \mathbb{Z}^e$ is an affine map and if u is a binomial sum, then the sequence $\boldsymbol{n} \in \mathbb{Z}^d \to u_{\lambda(\boldsymbol{n})}$ is a binomial sum,

- if u is a binomial sum, then the following indefinite sum is also a binomial sum

$$(\boldsymbol{n}, m) \in \mathbb{Z}^d \times \mathbb{Z} \mapsto \sum_{k=0}^{m} u_{\boldsymbol{n},k}. \tag{5.12}$$

Two examples of binomial sums taken from the authors' article are:

$$\sum_{k=0}^{2n} (-1)^k \binom{2n}{k}^3, \quad \sum_{i=0}^{n} \sum_{j=0}^{n} \binom{i+j}{i}^2 \binom{4n-2i-2j}{2n-2i}. \tag{5.13}$$

While their method allows computing many sums arising in the literature very efficiently, the class of binomial sums remains rather restrictive. A natural research direction is to enlarge this class by allowing more general integral representations: one could consider holonomic functions in the integrand (instead of only rational functions) and one could allow boundaries defined by polynomial inequalities. Indeed, Oaku [100] showed that such integrals can be handled automatically by creative telescoping. Our integration algorithm presented in Section 3.3 would be a good candidate form computing such integrals. Some possibly useful integral representations are for instance:

$$\frac{1}{n} = \int_0^1 x^{n-1} dx, \tag{5.14}$$

$$n! = \int_0^\infty x^{n-1} e^{-x} dx, \tag{5.15}$$

$$\frac{1}{n!} = \frac{1}{2i\pi} \oint \frac{e^x}{x^{n+1}} dx, \tag{5.16}$$

which would allow respectively the occurrence in binomial sums of polynomials in the denominator, factorials and the inverse of factorials.

Bibliography

[1] S. A. Abramov and M. Petkovsek. “Minimal Decomposition of Indefinite Hypergeometric Sums”. In: *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’01. London Ontario Canada: ACM, July 2001, pp. 7–14. DOI: 10.1145/384101.384103.

[2] W. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*. Graduate Studies in Mathematics v. 3. Providence, R.I: American Mathematical Society, 1994.

[3] G. Almkvist and D. Zeilberger. “The Method of Differentiating under the Integral Sign”. In: *Journal of Symbolic Computation* 10.6 (Dec. 1990), pp. 571–591. DOI: 10.1016/S0747-7171(08)80159-9.

[4] T. Amdeberhan, C. Koutschan, and D. Zeilberger. *Yay for Determinants!* Sept. 2023. DOI: 10.48550/arXiv.2307.01912.

[5] R. Bahloul. “Gröbner Bases in D-Modules: Application to Bernstein-Sato Polynomials”. In: *Two Algebraic Byways from Differential Equations: Gröbner Bases and Quivers*. Ed. by K. Iohara, P. Malbos, M.-H. Saito, and N. Takayama. Cham: Springer International Publishing, 2020, pp. 75–93. DOI: 10.1007/978-3-030-26454-3_2.

[6] M. Barakat, R. Brüser, C. Fieker, T. Huber, and J. Piclum. “Feynman Integral Reduction Using Gröbner Bases”. In: *Journal of High Energy Physics* 2023.5 (May 2023), p. 168. DOI: 10.1007/JHEP05(2023)168.

[7] M. Barakat, R. Brüser, T. Huber, and J. Piclum. “IBP Reduction via Gröbner Bases in a Rational Double-Shift Algebra”. In: *PoS* LL2022 (2022), p. 043. DOI: 10.22323/1.416.0043.

[8] M. A. Barkatou and S. S. Maddah. “Removing Apparent Singularities of Systems of Linear Differential Equations with Rational Function Coefficients”. In: *Proceedings of the 2015 ACM International Symposium on Symbolic and Algebraic Computation*. Bath United Kingdom: ACM, June 2015, pp. 53–60. DOI: 10.1145/2755996.2756668.

[9] L. C. Barrett and F. Dristy. “Particular Solutions for Nonhomogeneous, Linear, Ordinary Difference Equations”. In: *The American Mathematical Monthly* 67.1 (Jan. 1960), p. 71. DOI: 10.2307/2308936.

[10] V. Batyrev and M. Kreuzer. “Constructing New Calabi-Yau 3-Folds and Their Mirrors via Conifold Transitions”. In: *Advances in Theoretical and Mathematical Physics* 14.3 (2010), pp. 879–898. DOI: 10.4310/ATMP.2010.v14.n3.a3.

[11] T. Becker, V. Weispfenning, and H. Kredel. *Gröbner Bases: A Computational Approach to Commutative Algebra*. Graduate Texts in Mathematics 141. New York: Springer-Verlag, 1993.

[12] I. N. Bernshtein. "Modules over a Ring of Differential Operators. Study of the Fundamental Solutions of Equations with Constant Coefficients". In: *Functional Analysis and Its Applications* 5.2 (1971), pp. 89–101. DOI: 10.1007/BF01076413.

[13] I. N. Bernshtein. "The Analytic Continuation of Generalized Functions with Respect to a Parameter". In: *Functional Analysis and Its Applications* 6.4 (Oct. 1972), pp. 273–285. DOI: 10.1007/BF01077645.

[14] F. Beukers. "Irrationality of π^2, Periods of an Elliptic Curve and $\Gamma_1(5)$". In: *Birkhäuser Boston* 31. Progr. Math. Mass.Diophantine approximations and transcendental numbers (Luminy, 1982). ().

[15] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing". In: *SIAM Review* 59.1 (Jan. 2017), pp. 65–98. DOI: 10.1137/141000671.

[16] J.-E. Björk. *Rings of Differential Operators*. North-Holland Mathematical Library v. 21. Amsterdam New York: North-Holland Pub. Co. Sole distributors for the U.S.A. and Canada, Elsevier North-Holland Pub. Co, 1979.

[17] A. Borel. *Algebraic D-modules*. Perspectives in mathematics vol. 2. Boston: Academic Press, 1987.

[18] A. Bostan, S. Chen, F. Chyzak, and Z. Li. "Complexity of Creative Telescoping for Bivariate Rational Functions". In: *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. ISSAC '10. New York, NY, USA: ACM, July 2010, pp. 203–210. DOI: 10.1145/1837934.1837975.

[19] A. Bostan, S. Chen, F. Chyzak, Z. Li, and G. Xin. "Hermite Reduction and Creative Telescoping for Hyperexponential Functions". In: *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. ISSAC '13. New York, NY, USA: ACM, June 2013, pp. 77–84. DOI: 10.1145/2465506.2465946.

[20] A. Bostan, F. Chyzak, P. Lairez, and B. Salvy. "Generalized Hermite Reduction, Creative Telescoping and Definite Integration of D-Finite Functions". In: *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC '18. New York, NY, USA: ACM, July 2018, pp. 95–102. DOI: 10.1145/3208976.3208992.

[21] A. Bostan, F. Chyzak, M. van Hoeij, M. Kauers, and L. Pech. "Hypergeometric Expressions for Generating Functions of Walks with Small Steps in the Quarter Plane". In: *European Journal of Combinatorics* 61 (Mar. 2017), pp. 242–275. DOI: 10.1016/j.ejc.2016.10.010.

[22] A. Bostan, L. Dumont, and B. Salvy. “Efficient Algorithms for Mixed Creative Telscoping”. In: *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC ’16. New York, NY, USA: ACM, July 2016, pp. 127–134. DOI: 10.1145/2930889.2930907.

[23] A. Bostan, P. Lairez, and B. Salvy. “Creative Telescoping for Rational Functions Using the Griffiths: Dwork Method”. In: *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. Boston Maine USA: ACM, June 2013, pp. 93–100. DOI: 10.1145/2465506.2465935.

[24] A. Bostan, P. Lairez, and B. Salvy. “Multiple Binomial Sums”. In: *Journal of Symbolic Computation* 80 (May 2017), pp. 351–386. DOI: 10.1016/j.jsc.2016.04.002.

[25] J. Briançon and P. Maisonobe. *Remarques Sur l’idéal de Bernstein Associé à Des Polynômes*. Univ. Nice Sophia-Antipolis, 2002.

[26] M. Brickenstein. “Slimgb: Gröbner Bases with Slim Polynomials”. In: *Revista Matemática Complutense* 23.2 (July 2010), pp. 453–466. DOI: 10.1007/s13163-009-0020-0.

[27] H. Brochet. *The Julia Package MultivariateCreativeTelescoping*. Mar. 2025. URL: https://github.com/HBrochet/MultivariateCreativeTelescoping.jl.

[28] H. Brochet, F. Chyzak, and P. Lairez. *Faster Multivariate Integration in D-modules*. Apr. 2025. DOI: 10.48550/arXiv.2504.12724.

[29] H. Brochet and B. Salvy. “Reduction-Based Creative Telescoping for Definite Summation of D-finite Functions”. In: *Journal of Symbolic Computation* 125 (Nov. 2024), p. 102329. DOI: 10.1016/j.jsc.2024.102329.

[30] H. Brochet and B. Salvy. *The Maple Package CreativeTelescoping*. Jan. 2024. URL: https://github.com/HBrochet/CreativeTelescoping.

[31] M. Bronstein and M. Petkovšek. “An Introduction to Pseudo-Linear Algebra”. In: *Theoretical Computer Science* 157.1 (Apr. 1996), pp. 3–33. DOI: 10.1016/0304-3975(95)00173-5.

[32] F. Castro. “Théorème de division pour les opérateurs différentiels et calcul des multiplicités”. PhD thesis. Université Paris 7, 1984.

[33] F. J. Castro Jiménez and A. Leykin. “Computing localizations Iteratively”. In: *Harmony of Gröbner Bases and the Modern Industrial Society, 2012, ISBN 9789814383455, págs. 26-40*. World Scientific, 2010. Chap. Harmony of Gröbner Bases and the Modern Industrial Society, pp. 26–40. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=8452385.

[34] C. J. Castro-Jiménez Francisco. “Calculs Effectifs Pour Les Idéaux d’opérateurs Différentiels”. In: *in Actas de la II Conferencia Internacional de Geometria Algebraic* (1987).

[35] F. J. Castro-Jiménez and J. M. Ucha-Enríquez. "Gröbner Bases and Logarithmic D-modules". In: *Journal of Symbolic Computation*. Logic, Mathematics and Computer Science: Interactions in Honor of Bruno Buchberger (60th Birthday) 41.3 (Mar. 2006), pp. 317–335. DOI: 10.1016/j.jsc.2004.04.011.

[36] S. Chen, L. Du, and M. Kauers. "Hermite Reduction for D-finite Functions via Integral Bases". In: *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*. ISSAC '23. New York, NY, USA: ACM, July 2023, pp. 155–163. DOI: 10.1145/3597066.3597092.

[37] S. Chen, L. Du, M. Kauers, and R.-H. Wang. "Reduction-Based Creative Telescoping for P-recursive Sequences via Integral Bases". In: *Journal of Symbolic Computation* 126.C (Jan. 2025). DOI: 10.1016/j.jsc.2024.102341.

[38] S. Chen, Q.-H. Hou, H. Huang, G. Labahn, and R.-H. Wang. "Constructing Minimal Telescopers for Rational Functions in Three Discrete Variables". In: *Advances in Applied Mathematics* 141 (Oct. 2022), p. 102389. DOI: 10.1016/j.aam.2022.102389.

[39] S. Chen, H. Huang, M. Kauers, and Z. Li. "A Modified Abramov-Petkovsek Reduction and Creative Telescoping for Hypergeometric Terms". In: *Proceedings of the 2015 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC '15. New York, NY, USA: ACM, June 2015, pp. 117–124. DOI: 10.1145/2755996.2756648.

[40] S. Chen, M. Kauers, and C. Koutschan. "Reduction-Based Creative Telescoping for Algebraic Functions". In: *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC '16. New York, NY, USA: ACM, July 2016, pp. 175–182. DOI: 10.1145/2930889.2930901.

[41] S. Chen, M. Kauers, Z. Li, and Y. Zhang. "Apparent Singularities of D-finite Systems". In: *Journal of Symbolic Computation* 95 (Nov. 2019), pp. 217–237. DOI: 10.1016/j.jsc.2019.02.009.

[42] S. Chen, M. Kauers, and M. F. Singer. "Desingularization of Ore Operators". In: *Journal of Symbolic Computation* 74 (May 2016), pp. 617–626. DOI: 10.1016/j.jsc.2015.11.001.

[43] S. Chen, M. van Hoeij, M. Kauers, and C. Koutschan. "Reduction-Based Creative Telescoping for Fuchsian D-finite Functions". In: *Journal of Symbolic Computation*. 41th International Symposium on Symbolic and Alge-braic Computation (ISSAC'16) 85 (Mar. 2018), pp. 108–127. DOI: 10.1016/j.jsc.2017.07.005.

[44] R. C. Churchill and J. J. Kovacic. "CYCLIC VECTORS". In: *Differential Algebra and Related Topics*. The State University of New Jersey: WORLD SCIENTIFIC, May 2002, pp. 191–218. DOI: 10.1142/9789812778437_0007.

[45] F. Chyzak and B. Salvy. "Non-Commutative Elimination in Ore Algebras Proves Multivariate Identities". In: *Journal of Symbolic Computation* 26.2 (Aug. 1998), pp. 187–227. DOI: 10.1006/jsco.1998.0207.

[46] F. Chyzak. "Fonctions holonomes en calcul formel". PhD thesis. Ecole Polytechnique X, May 1998. URL: https://theses.hal.science/tel-00991717.

[47] F. Chyzak. "An Extension of Zeilberger's Fast Algorithm to General Holonomic Functions". In: *Discrete Mathematics* 217.1 (Apr. 2000), pp. 115–134. DOI: 10.1016/S0012-365X(99)00259-9.

[48] F. Chyzak. "The ABC of Creative Telescoping — Algorithms, Bounds, Complexity". Thesis. Ecole Polytechnique X, Apr. 2014. URL: https://theses.hal.science/tel-01069831.

[49] F. Chyzak, A. Mahboubi, T. Sibut-Pinote, and E. Tassi. "A Computer-Algebra-Based Formal Proof of the Irrationality of $\zeta(3)$". In: *Interactive Theorem Proving*. Ed. by G. Klein and R. Gamboa. Cham: Springer International Publishing, 2014, pp. 160–176. DOI: 10.1007/978-3-319-08970-6_11.

[50] F. Chyzak and M. Mishna. *Differential Equations Satisfied by Generating Functions of 5-, 6-, and 7-Regular Labelled Graphs: A Reduction-Based Approach*. June 2025. DOI: 10.48550/arXiv.2406.04753.

[51] F. Chyzak, M. Mishna, and B. Salvy. "Effective Scalar Products of D-finite Symmetric Functions". In: *Journal of Combinatorial Theory, Series A* 112.1 (Oct. 2005), pp. 1–43. DOI: 10.1016/j.jcta.2005.01.001.

[52] T. Cluzeau, C. Pinto, and A. Quadrat. "Further Results on the Computation of the Annihilators of Integro-Differential Operators". In: *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*. Tromsø Norway: ACM, July 2023, pp. 182–190. DOI: 10.1145/3597066.3597083.

[53] T. Cluzeau, C. Pinto, and A. Quadrat. "Effective Characterization of Evaluation Ideals of the Ring of Integro-Differential Operators". In: *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation*. Raleigh NC USA: ACM, July 2024, pp. 117–125. DOI: 10.1145/3666000.3669682.

[54] S. C. Coutinho. *A Primer of Algebraic D-Modules*. London Mathematical Society Student Texts. Cambridge: Cambridge University Press, 1995. DOI: 10.1017/CBO9780511623653.

[55] D. A. Cox, J. B. Little, and D. O'Shea. *Using Algebraic Geometry*. 2nd ed. Graduate Texts in Mathematics 185. New York: Springer, 2005.

[56] W. Decker, G.-M. Greuel, G. Pfister, and H. Schöonemann. *Singular 4-4-0 — A Computer Algebra System for Polynomial Computations*. 2024. URL: %5Ctextbackslash%20url%5Ctextbraceleft%20http://www.singular.uni-kl.de%5Ctextbraceright.

[57] A. Dimca. "On the Milnor Fibrations of Weighted Homogeneous Polynomials". In: *Algebraic Geometry: Proceedings of the Conference at Berlin 9–15 March 1988*. Ed. by H. Kurke and J. H. M. Steenbrink. Dordrecht: Springer Netherlands, 1990, pp. 19–47. DOI: 10.1007/978-94-009-0685-3_3.

[58] L. Du. “On the Existence of Telescopers for P-recursive Sequences”. In: *Journal of Symbolic Computation* 130 (Sept. 2025), p. 102423. DOI: 10.1016/j.jsc.2025.102423.

[59] T. W. Dubé. “The Structure of Polynomial Ideals and Gröbner Bases”. In: *SIAM Journal on Computing* 19.4 (Aug. 1990), pp. 750–773. DOI: 10.1137/0219053.

[60] B. Dwork. “On the Zeta Function of a Hypersurface”. In: *Publications Mathématiques de l’Institut des Hautes Études Scientifiques* 12.1 (Dec. 1962), pp. 5–68. DOI: 10.1007/BF02684275.

[61] B. Dwork. “On the Zeta Function of a Hypersurface: II”. In: *The Annals of Mathematics* 80.2 (Sept. 1964), p. 227. DOI: 10.2307/1970392.

[62] M. C. Fasenmyer. “Some Generalized Hypergeometric Polynomials”. In: *Bulletin of the American Mathematical Society* 53.8 (1947), pp. 806–812. DOI: 10.1090/S0002-9904-1947-08893-5.

[63] M. C. Fasenmyer. “A Note on Pure Recurrence Relations”. In: *The American Mathematical Monthly* 56.1 (1949), pp. 14–17. DOI: 10.2307/2305810.

[64] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. “Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering”. In: *Journal of Symbolic Computation* 16.4 (Oct. 1993), pp. 329–344. DOI: 10.1006/jsco.1993.1051.

[65] J.-C. Faugére. “A New Efficient Algorithm for Computing Gröbner Bases ($F4$)”. In: *Journal of Pure and Applied Algebra* 139.1 (June 1999), pp. 61–88. DOI: 10.1016/S0022-4049(99)00005-5.

[66] G. B. Folland. *Introduction to Partial Differential Equations: Second Edition*. Princeton University Press, May 2020. DOI: 10.2307/j.ctvzsmfgn.

[67] A. Galligo. “Some Algorithmic Questions on Ideals of Differential Operators”. In: *EUROCAL ’85*. Ed. by B. F. Caviness. Berlin, Heidelberg: Springer, 1985, pp. 413–421. DOI: 10.1007/3-540-15984-3_301.

[68] J. Gerhard, M. Giesbrecht, A. Storjohann, and E. V. Zima. “Shiftless Decomposition and Polynomial-Time Rational Summation”. In: *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’03. New York, NY, USA: ACM, Aug. 2003, pp. 119–126. DOI: 10.1145/860854.860887.

[69] J. Gillis, B. Reznick, and D. Zeilberger. “On Elementary Methods in Positivity Theory”. In: *SIAM Journal on Mathematical Analysis* 14.2 (Mar. 1983), pp. 396–398. DOI: 10.1137/0514031.

[70] D. R. Grayson and M. E. Stillman. *Macaulay2, a Software System for Research in Algebraic Geometry*. URL: http://www2.macaulay2.com.

[71] P. A. Griffiths. “On the Periods of Certain Rational Integrals: I”. In: *Annals of Mathematics* 90.3 (1969), pp. 460–495. DOI: 10.2307/1970746.

[72] L. Guo, G. Regensburger, and M. Rosenkranz. “On Integro-Differential Algebras”. In: *Journal of Pure and Applied Algebra* 218.3 (Mar. 2014), pp. 456–473. DOI: 10.1016/j.jpaa.2013.06.015.

[73] P. A. Hendricks and M. F. Singer. "Solving Difference Equations in Finite Terms". In: *Journal of Symbolic Computation* 27.3 (Mar. 1999), pp. 239–259. DOI: 10.1006/jsco.1998.0251.

[74] C. Hermite. "Sur l'intégration Des Fractions Rationnelles". In: *Annales scientifiques de l'École normale supérieure* 1 (1872), pp. 215–218. DOI: 10.24033/asens.82.

[75] E. L. Ince. *Ordinary Differential Equations*. Dover, 1926.

[76] N. Jacobson. "Pseudo-Linear Transformations". In: *Annals of Mathematics* 38.2 (1937), pp. 484–507. DOI: 10.2307/1968565.

[77] A. Kandri-Rody and V. Weispfenning. "Non-Commutative Gröbner Bases in Algebras of Solvable Type". In: *Journal of Symbolic Computation* 9.1 (Jan. 1990), pp. 1–26. DOI: 10.1016/S0747-7171(08)80003-X.

[78] M. Karr. "Summation in Finite Terms". In: *Journal of the ACM* 28.2 (Apr. 1981), pp. 305–350. DOI: 10.1145/322248.322255.

[79] M. Karr. "Theory of Summation in Finite Terms". In: *Journal of Symbolic Computation* 1.3 (Sept. 1985), pp. 303–315. DOI: 10.1016/S0747-7171(85)80038-9.

[80] M. Kashiwara. "B-Functions and Holonomic Systems". In: *Inventiones mathematicae* 38.1 (Feb. 1976), pp. 33–53. DOI: 10.1007/BF01390168.

[81] M. Kashiwara. "On the Holonomic Systems of Linear Differential Equations, II". In: *Inventiones mathematicae* 49.2 (June 1978), pp. 121–135. DOI: 10.1007/BF01403082.

[82] M. Kauers. *D-Finite Functions*. Vol. 30. Algorithms and Computation in Mathematics. Cham: Springer Nature Switzerland, 2023. DOI: 10.1007/978-3-031-34652-1.

[83] M. Kauers, M. Jaroschek, and F. Johansson. "Ore Polynomials in Sage". In: *Computer Algebra and Polynomials*. Ed. by J. Gutierrez, J. Schicho, and M. Weimann. Vol. 8942. Cham: Springer International Publishing, 2015, pp. 105–125. DOI: 10.1007/978-3-319-15081-9_6.

[84] C. Koutschan. "Advanced applications of the holonomic systems approach". PhD thesis. June 2010. URL: https://dl.acm.org/doi/10.1145/1823931.1823954.

[85] C. Koutschan. "Holonomic Functions in Mathematica". In: *ACM Commun. Comput. Algebra* 47.3/4 (Jan. 2014), pp. 179–182. DOI: 10.1145/2576802.2576831.

[86] P. Lairez. "Computing Periods of Rational Integrals". In: *Mathematics of Computation* 85.300 (July 2016), pp. 1719–1752. DOI: 10.1090/mcom/3054.

[87] P. Lairez. "Axioms for a Theory of Signature Bases". In: *Journal of Symbolic Computation* 123 (2024), p. 102275. DOI: 10.1016/j.jsc.2023.102275.

[88] V. Levandovskyy. "Non-Commutative Computer Algebra for Polynomial Algebras: Gröbner Bases, Applications and Implementation". PhD thesis. Technische Universität Kaiserslautern, 2005.

[89] V. Levandovskyy and J. M. Morales. "Computational D-module Theory with Singular, Comparison with Other Systems and Two New Algorithms". In: *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation*. Linz/Hagenberg Austria: ACM, July 2008, pp. 173–180. DOI: 10.1145/1390768.1390794.

[90] V. Levandovskyy and H. Schönemann. "Plural: A Computer Algebra System for Noncommutative Polynomial Algebras". In: *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*. ISSAC '03. New York, NY, USA: ACM, Aug. 2003, pp. 176–183. DOI: 10.1145/860854.860895.

[91] A. Leykin. "D-MODULES FOR MACAULAY 2". In: *Mathematical Software*. Beijing, China: WORLD SCIENTIFIC, July 2002, pp. 169–179. DOI: 10.1142/9789812777171_0017.

[92] L. Lipshitz. "D-Finite Power Series". In: *Journal of Algebra* 122.2 (May 1989), pp. 353–373. DOI: 10.1016/0021-8693(89)90222-6.

[93] B. Malgrange. "Le Polynome de Bernstein d'une Singularite Isolee". In: *Fourier Integral Operators and Partial Differential Equations*. Ed. by J. Chazarain. Vol. 459. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 98–119. DOI: 10.1007/BFb0074194.

[94] C. Mallinger. "Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequence". MA thesis. Linz: J. Kepler University, Aug. 1996.

[95] E. W. Mayr and A. R. Meyer. "The Complexity of the Word Problems for Commutative Semigroups and Polynomial Ideals". In: *Advances in Mathematics* 46.3 (Dec. 1982), pp. 305–329. DOI: 10.1016/0001-8708(82)90048-2.

[96] E. W. Mayr and S. Ritscher. "Dimension-Dependent Bounds for Gröbner Bases of Polynomial Ideals". In: *Journal of Symbolic Computation* 49 (Feb. 2013), pp. 78–94. DOI: 10.1016/j.jsc.2011.12.018.

[97] National Institute of Standards and Technology. *NIST Handbook of Mathematical Functions*. Ed. by F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark. Cambridge New York Melbourne: Cambridge University Press, 2010.

[98] T. Oaku. "Computation of the Characteristic Variety and the Singular Locus of a System of Differential Equations with Polynomial Coefficients". In: *Japan Journal of Industrial and Applied Mathematics* 11.3 (Oct. 1994), pp. 485–497. DOI: 10.1007/BF03167233.

[99] T. Oaku. "An Algorithm of Computing B-Functions". In: *Duke Mathematical Journal* 87.1 (Mar. 1997). DOI: 10.1215/S0012-7094-97-08705-6.

[100] T. Oaku. "Algorithms for Integrals of Holonomic Functions over Domains Defined by Polynomial Inequalities". In: *Journal of Symbolic Computation* 50 (Mar. 2013), pp. 1–27. DOI: 10.1016/j.jsc.2012.05.004.

[101] T. Oaku. “Localization, Local Cohomology, and the b-Function of a D-module with Respect to a Polynomial”. In: *Current Trends on Gröbner Bases: The 50th Anniversary of Gröbner Bases*. Osaka, Japan, pp. 353–398. DOI: 10.2969/aspm/07710353.

[102] T. Oaku and N. Takayama. “Algorithms for D-modules — Restriction, Tensor Product, Localization, and Local Cohomology Groups”. In: *Journal of Pure and Applied Algebra* 156.2-3 (Feb. 2001), pp. 267–308. DOI: 10.1016/S0022-4049(00)00004-9.

[103] T. Oaku, N. Takayama, and U. Walther. “A Localization Algorithm forD-modules”. In: *Journal of Symbolic Computation* 29.4-5 (May 2000), pp. 721–728. DOI: 10.1006/jsco.1999.0398.

[104] A. P. Prudnikov, J. A. Bryčkov, and O. I. Maričev. “Integrals and Series. 2: Special Functions”. In: 4. print. New York, NY: Gordon and Breach Science Publ, 1998.

[105] A. Quadrat and G. Regensburger. “Computing Polynomial Solutions and Annihilators of Integro-Differential Operators with Polynomial Coefficients”. In: *Algebraic and Symbolic Computation Methods in Dynamical Systems*. Ed. by A. Quadrat and E. Zerz. Cham: Springer International Publishing, 2020, pp. 87–114. DOI: 10.1007/978-3-030-38356-5_3.

[106] C. Raab. “Definite Integration in Differential Fields”. PhD thesis. Johannes Kepler University, 2012.

[107] C. G. Raab. “Generalization of Risch’s Algorithm to Special Functions”. In: *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions*. Ed. by C. Schneider and J. Blümlein. Vienna: Springer, 2013, pp. 285–304. DOI: 10.1007/978-3-7091-1616-6_12.

[108] J. L. Rabinowitsch. “Zum Hilbertschen Nullstellensatz”. In: *Mathematische Annalen* 102.1 (Dec. 1930), pp. 520–520. DOI: 10.1007/BF01782361.

[109] G. Regensburger, M. Rosenkranz, and J. Middeke. “A Skew Polynomial Approach to Integro-Differential Operators”. In: *Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’09. New York, NY, USA: ACM, July 2009, pp. 287–294. DOI: 10.1145/1576702.1576742.

[110] R. H. Risch. “The Problem of Integration in Finite Terms”. In: *Transactions of the American Mathematical Society* 139.0 (1969), pp. 167–189. DOI: 10.1090/S0002-9947-1969-0237477-8.

[111] R. H. Risch. “The Solution of the Problem of Integration in Finite Terms”. In: *Bulletin of the American Mathematical Society* 76.3 (1970), pp. 605–608. DOI: 10.1090/S0002-9904-1970-12454-5.

[112] M. Rosenkranz and G. Regensburger. “Integro-Differential Polynomials and Operators”. In: *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation*. Linz/Hagenberg Austria: ACM, July 2008, pp. 261–268. DOI: 10.1145/1390768.1390805.

[113] M. Rosenkranz and G. Regensburger. "Solving and Factoring Boundary Problems for Linear Ordinary Differential Equations in Differential Algebras". In: *Journal of Symbolic Computation* 43.8 (Aug. 2008), pp. 515–544. DOI: 10.1016/j.jsc.2007.11.007.

[114] M. Saito, B. Sturmfels, and N. Takayama. *Gröbner Deformations of Hypergeometric Differential Equations*. Ed. by E. Becker, M. Bronstein, H. Cohen, D. Eisenbud, and R. Gilman. Vol. 6. Algorithms and Computation in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. DOI: 10.1007/978-3-662-04112-3.

[115] B. Salvy and P. Zimmermann. "GFUN: A Maple Package for the Manipulation of Generating and Holonomic Functions in One Variable". In: *ACM Transactions on Mathematical Software* 20.2 (June 1994), pp. 163–177. DOI: 10.1145/178365.178368.

[116] C. Schneider. "Symbolic Summation in Difference Fields". PhD thesis. Johannes Kepler University, 2001.

[117] C. Schneider. "Symbolic Summation with Single-Nested Sum Extensions". In: *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*. Santander Spain: ACM, July 2004, pp. 282–289. DOI: 10.1145/1005285.1005326.

[118] C. Schneider. "Finding Telescopers with Minimal Depth for Indefinite Nested Sum and Product Expressions". In: *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation*. Beijing China: ACM, July 2005, pp. 285–292. DOI: 10.1145/1073884.1073924.

[119] C. Schneider. "Product Representations in ΠΣ-Fields". In: *Annals of Combinatorics* 9.1 (Apr. 2005), pp. 75–99. DOI: 10.1007/s00026-005-0242-2.

[120] C. Schneider. "SIMPLIFYING SUMS IN ΠΣ*-EXTENSIONS". In: *Journal of Algebra and Its Applications* 06.03 (June 2007), pp. 415–441. DOI: 10.1142/S0219498807002302.

[121] C. Schneider. "A Refined Difference Field Theory for Symbolic Summation". In: *Journal of Symbolic Computation* 43.9 (Sept. 2008), pp. 611–644. DOI: 10.1016/j.jsc.2008.01.001.

[122] C. Schneider. "Simplifying Multiple Sums in Difference Fields". In: *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions*. Ed. by C. Schneider and J. Blümlein. Vienna: Springer, 2013, pp. 325–360. DOI: 10.1007/978-3-7091-1616-6_14.

[123] C. Schneider. "A Difference Ring Theory for Symbolic Summation". In: *Journal of Symbolic Computation* 72 (Jan. 2016), pp. 82–127. DOI: 10.1016/j.jsc.2015.02.002.

[124] C. Schneider. "Summation Theory II: Characterizations of $R\Pi\Sigma$ -Extensions and Algorithmic Aspects". In: *Journal of Symbolic Computation* 80 (May 2017), pp. 616–664. DOI: 10.1016/j.jsc.2016.07.028.

[125] R. P. Stanley. “Differentiably Finite Power Series”. In: *European Journal of Combinatorics* 1.2 (June 1980), pp. 175–188. DOI: 10.1016/S0195-6698(80)80051-5.

[126] Y. Sun, D. Wang, X. Ma, and Y. Zhang. “A Signature-Based Algorithm for Computing Gröbner Bases in Solvable Polynomial Algebras”. In: *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*. Grenoble France: ACM, July 2012, pp. 351–358. DOI: 10.1145/2442829.2442879.

[127] J. Sylvester. “XXIII. A Method of Determining by Mere Inspection the Derivatives from Two Equations of Any Degree”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 16.101 (Feb. 1840), pp. 132–135. DOI: 10.1080/14786444008649995.

[128] N. Takayama. “An Algorithm of Constructing the Integral of a Module–an Infinite Dimensional Analog of Gröbner Basis”. In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation*. ISSAC ’90. New York, NY, USA: ACM, July 1990, pp. 206–211. DOI: 10.1145/96877.96929.

[129] N. Takayama. “Gröbner Basis, Integration and Transcendental Functions”. In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation*. Tokyo Japan: ACM, July 1990, pp. 152–156. DOI: 10.1145/96877.96916.

[130] N. Takayama. “Gröbner Basis and the Problem of Contiguous Relations”. In: *Japan Journal of Applied Mathematics* 6.1 (Feb. 1989), pp. 147–160. DOI: 10.1007/BF03167920.

[131] N. Takayama. “An Approach to the Zero Recognition Problem by Buchberger Algorithm”. In: *Journal of Symbolic Computation* 14.2-3 (Aug. 1992), pp. 265–282. DOI: 10.1016/0747-7171(92)90039-7.

[132] C. Traverso. “Gröbner Trace Algorithms”. In: *Symbolic and Algebraic Computation*. Ed. by P. Gianni. Berlin, Heidelberg: Springer, 1989, pp. 125–138. DOI: 10.1007/3-540-51084-2_12.

[133] H. Tsai. “Weyl Closure of a Linear Differential Operator”. In: *Journal of Symbolic Computation* 29.4-5 (May 2000), pp. 747–775. DOI: 10.1006/jsco.1999.0400.

[134] H. Tsai. “Algorithms for Algebraic Analysis”. PhD thesis. University of California at Berkelet.

[135] J. van der Hoeven. “Creative Telescoping Using Reductions”. Apr. 2018. URL: https://hal.science/hal-01773137.

[136] J. van der Hoeven. “Constructing Reductions for Creative Telescoping”. In: *Applicable Algebra in Engineering, Communication and Computing* 32 (Jan. 2021), pp. 575–602. DOI: 10.1007/s00200-020-00413-3.

[137] A. van der Poorten and R. Apéry. “A Proof That Euler Missed ...” In: *The Mathematical Intelligencer* 1.4 (Dec. 1979), pp. 195–203. DOI: 10.1007/BF03028234.

[138] P. Verbaeten. “The Automatic Construction of Pure Recurrence Relations”. In: *SIGSAM Bull.* 8.3 (Aug. 1974), pp. 96–98. DOI: 10.1145/1086837.1086854.

[139] P. Verbaeten. "Rekursiebetrekkingen Voor Lineaire Hypergeometrische Funkties". PhD thesis. Leuven: KU Leuven.

[140] J. Von Zur Gathen and J. Gerhard. *Modern Computer Algebra*. 3rd ed. Cambridge University Press, Apr. 2013. DOI: 10.1017/CBO9781139856065.

[141] K. Wegschaider. "Computer Generated Proofs of Binomial Multi-Sum Identities." MA thesis. Johannes Kepler University, 1997.

[142] H. S. Wilf and D. Zeilberger. "An Algorithmic Proof Theory for Hypergeometric (Ordinary and "q") Multisum/Integral Identities". In: *Inventiones mathematicae* 108.1 (Dec. 1992), pp. 575–633. DOI: 10.1007/BF02100618.

[143] D. Zeilberger. "Sister Celine's Technique and Its Generalizations". In: *Journal of Mathematical Analysis and Applications* 85.1 (Jan. 1982), pp. 114–145. DOI: 10.1016/0022-247X(82)90029-4.

[144] D. Zeilberger. "A Holonomic Systems Approach to Special Functions Identities". In: *Journal of Computational and Applied Mathematics* 32.3 (Dec. 1990), pp. 321–368. DOI: 10.1016/0377-0427(90)90042-X.

[145] D. Zeilberger. "The Method of Creative Telescoping". In: *Journal of Symbolic Computation* 11.3 (Mar. 1991), pp. 195–204. DOI: 10.1016/S0747-7171(08)80044-2.

[146] Y. Zhang. "Contraction of Ore Ideals with Applications". In: *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC '16. New York, NY, USA: ACM, July 2016, pp. 413–420. DOI: 10.1145/2930889.2930890.