Simple Differentially Definable Functions

Antonio Jiménez-Pastor*
ajpastor@risc.uni-linz.ac.at
Johannes Kepler University Linz
Research Institute for Symbolic Computation
Linz, Austria

ABSTRACT

D-finite functions satisfy linear differential equations with polynomial coefficients. The solutions to this type of equations may have singularities determined by the zeros of their leading coefficient. There are algorithms to desingularize the equations, i.e., remove singularities from the equation that do not appear in its solutions. However, classical computations of closure properties (such as addition, multiplication, etc.) with D-finite functions return equations with extra zeros in the leading coefficient. In this paper we present theory and algorithms based on linear algebra to control the leading coefficients when computing these closure properties and we also extend this theory to the more general class of differentially definable functions.

CCS CONCEPTS

• **Mathematics of computing → Mathematical software**; *Generating functions*; **Ordinary differential equations**.

KEYWORDS

D-finite functions; formal power series; closure properties; differential algebra; generating functions; ordinary differential equations; linear algebra

ACM Reference Format:
Antonio Jiménez-Pastor. 2021. Simple Differentially Definable Functions. In *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation (ISSAC '21), July 18–23, 2021, Virtual Event, Russian Federation.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3452143.3465534

1 INTRODUCTION

D-finite functions, i.e., formal power series that satisfy a linear differential equation with polynomial coefficients, have been widely studied in the last decades. Using these differential equations and some initial conditions we can get a finite representation for these objects. Many algorithms have been developed to work with this representation of D-finite functions [6, 14, 20], and can be used to prove identities for special functions or sequences in enumerative combinatorics [13, 18].

There are also results that characterize the singularities that a solution to a linear differential equation can have. These points are called *singularities of the differential operators*. It is known that the singularities of a differential operator are the singularities of its coefficients and the zeros of its leading coefficient. For D-finite operators, since all coefficients are polynomials, the singularities are just the (finitely many) zeros of the leading coefficients.

Sometimes a differential operator has some singularities that none of its solutions has as a singularity. These singularities are called *apparent*. The problem of finding an equivalent operator (i.e., an operator that contains all the solutions from the original operator) with no apparent singularities is called desingularization [1, 3, 5].

There are algorithms for computing the desingularization of Ore operators [7] (a generalization of differential operators) and differential systems [6]. However, the computations of closure properties over operators without apparent singularities do not preserve this property, meaning that we need to apply again the desingularization process in order to keep the resulting differential operator without apparent singularities.

In this paper we present the concept of *S-simple D-finite functions*. These are functions that satisfy a linear differential equation with a leading coefficient from a fixed set S. We prove that this subclass of D-finite functions is a differential ring in a constructive way, leading to algorithms that compute operations (such as addition, multiplication, etc.) preserving the leading coefficient in the same set. Thus, we can control the zero set of the leading coefficient when computing closure properties of D-finite functions and compute directly operators without new apparent singularities.

We also extend this theory to the general case of differentially definable functions [10]. In particular, we extend the result for D^n-finite functions. These functions are defined recursively as formal power series that satisfy linear differential equations with D^{n-1}-finite coefficients. Using the results of this paper, we can build functions whose singularities are known and then, compute several operations preserving the singularities on the differential operator.

The algorithms described in this paper are implemented in the open source computer algebra system SageMath [21] and are included within the package `dd_functions` [9]. This package is a tool for computing with D-finite, DD-finite and more general classes of differentially definable functions.

In Section 2 we present the main theoretical results of the paper. Then in Section 3 we describe several cases of D-finite functions with special leading coefficients. In Section 4 we show how to actually compute the closure properties described in the previous

*This research was partially funded by the strategic program "Innovatives OO–2010 plus" and the Austrian Science Fund (FWF) under the grants SFB F50-07.

ISSAC '21, July 18–23, 2021, Virtual Event, Russian Federation

ACM ISBN 978-1-4503-8382-0/21/07…$15.00
https://doi.org/10.1145/3452143.3465534

section in the particular case of D-finite functions and in Section 5 we present the extension of these results to the general case of D^n-finite functions.

2 THEORETICAL RESULTS

In this section we prove that we can control the nature of the leading coefficient of the differential equations with which we are computing. In order to present this theory, we first recall some classical definitions:

Definition 2.1 ([2, Chapter 6]). Let R be a ring and M an R-module. We say that R is Noetherian if all ideals of R are finitely generated. We say that M is Noetherian if all R-submodules are finitely generated.

In this sense, R is Noetherian as a ring if it is Noetherian as an R-module.

Noetherian rings and modules have been widely studied and we know plenty of operations that preserve this property. For example, if M and N are Noetherian, then $M \oplus N$ and $M \otimes N$ are also Noetherian modules. We will use extensively these properties and all can be found in [2, Chapter 6].

Definition 2.2 ([4, Chapter 3]). Let R be a ring. We say that an additive map $\partial : R \to R$ is a derivation if it satisfies the Leibniz rule, i.e., for all $r, s \in R$, $\partial(rs) = \partial(r)s + r\partial(s)$. We say that (R, ∂) is a differential ring.

If $E \supset R$ is a ring extension and $\tilde{\partial} : E \to E$ is a derivation such that $\tilde{\partial}|_R \equiv \partial$, we say that $(E, \tilde{\partial})$ is a differential extension of (R, ∂).

We simply denote the extended derivation by ∂ again, i.e., (E, ∂) is a differential extension of (R, ∂). We also denote the set of linear differential operators over R by $R[\partial]$. Its elements $C = r_0 + \ldots + r_d\partial^d$ act over any differential extension E by

$$C \cdot e = r_0 e + \ldots + r_d \partial^d(e).$$

Definition 2.3 ([2, Chapter 3]). Let R be a ring and $S \subset R$. We say that S is *multiplicatively closed* if $1 \in S$, $0 \notin S$ and for all $s_1, s_2 \in S$ we have that $(s_1 s_2) \in S$.

Given a multiplicatively closed set S, we define the *localization of R w.r.t. S* as the set $R \times S$ with the equivalence relation $(r, s) \sim (r', s')$ if and only if there is $t \in S$ such that $t(rs' - r's) = 0$. We denote the equivalence class of (r, s) with r/s. We denote this ring by R_S.

The localization ring is usually studied when we consider prime or maximal ideals of a ring and also when we build the field of fractions of an integral domain. Moreover, if R is Noetherian, then R_S is also Noetherian and if R is an integral domain, then R_S is a differential extension of R.

In [10], the concept of *differentially definable* elements was given. Namely, if (R, ∂) is a differential integral domain and E a differential extension, we say that $f \in E$ is *differentially definable over R* if there is $\mathcal{A} \in R[\partial] \setminus \{0\}$ such that $\mathcal{A} \cdot f = 0$. We denote the set of these elements by $\mathrm{D}_E(R)$.

This definition leads to the set of D-finite functions when taking $R = \mathbb{K}[x]$ and $E = \mathbb{K}[[x]]$, where $\mathbb{K}$ is a field of characteristic 0. Here, we propose a slightly different variation of it, where we put some emphasis on the leading coefficient of the differential equation:

Definition 2.4. Let (R, ∂) be a differential integral domain, $E \supset R$ a differential extension and $S \subset R$ a multiplicatively closed set. We say that $f \in E$ is *S-simple differentially definable over R* if there is $\mathcal{A} \in R[\partial] \setminus \{0\}$ with $\mathrm{lc}(\mathcal{A}) \in S$ such that $\mathcal{A} \cdot f = 0$.

We denote the set of all these elements by $\mathrm{D}_E(R, S)$.

When we consider $S = R \setminus \{0\}$ this definition yields the usual differentially definable elements over R. It is known that differentially definable elements can be characterized with finite dimensional vector spaces [12]. A similar characterization can be proven for the S-simple differentially definable elements by using finitely generated R_S-modules instead.

Theorem 2.5. *Let (R, ∂) be a differential integral domain, $E \supset R$ a differential extension and $S \subset R$ a multiplicatively closed set. For $f \in E$, the following are equivalent:*

1. *$f \in \mathrm{D}_E(R, S)$.*
2. *$\exists g \in \mathrm{D}_E(R, S)$ and $\mathcal{A} \in R[\partial]$ with $\mathrm{lc}(\mathcal{A}) \in S$: $\mathcal{A} \cdot f = g$.*
3. *The R_S-module $\langle \partial^n(f) : n \in \mathbb{N} \rangle_{R_S}$ is finitely generated.*

Proof. (1) $\Rightarrow$ (2): taking $g = 0 \in \mathrm{D}_E(R, S)$ proves it.

(2) $\Rightarrow$ (1): let $\mathcal{B} \in R[\partial]$ with $\mathrm{lc}(\mathcal{B}) \in S$ such that $\mathcal{B} \cdot g = 0$. Then we have that $(\mathcal{B}\mathcal{A}) \cdot f = 0$ and $\mathrm{lc}(\mathcal{B}\mathcal{A}) = \mathrm{lc}(\mathcal{B})\,\mathrm{lc}(\mathcal{A}) \in S$.

(1) $\Rightarrow$ (3): let $\mathcal{A} \in R[\partial]$ with $\mathrm{lc}(\mathcal{A}) \in S$ be such that it annihilates f, i.e., $\mathcal{A} \cdot f = 0$. Assume that $d = \deg_\partial(\mathcal{A})$. Then it is clear that, for all $k \in \mathbb{N}$ we have that the operator $(\partial^k \mathcal{A})$ has the same leading coefficient (in S) and order $d + k$. Thus, by induction, $\partial^{d+k}(f) \in \langle f, \ldots, \partial^{d-1}(f) \rangle_{R_S}$, and the R_S-module generated by f and its derivatives is finitely generated.

(3) $\Rightarrow$ (1): let the module $\langle f, \partial(f), \ldots \rangle_{R_S}$ be finitely generated. There is $N \in \mathbb{N}$ such that $\partial^n(f) \in \langle f, \ldots, \partial^N(f) \rangle$ for all $n \in \mathbb{N}$, namely, we take N as the maximal derivative appearing in a set of generators. In particular, we have that, for $n = N + 1$:

$$\partial^n(f) = \frac{r_0}{s_0} f + \ldots + \frac{r_N}{s_N} \partial^N(f), \tag{1}$$

so taking $\tilde{r_i} = r_i\,(s_0 \cdots s_{i-1} s_{i+1} \cdots s_N)$ we have that

$$\mathcal{A} = s\partial^n - \tilde{r_N}\partial^N - \ldots - \tilde{r_0},$$

satisfies $\mathcal{A} \in R[\partial]$, $\mathrm{lc}(\mathcal{A}) = s \in S$ and $\mathcal{A} \cdot f = 0$, i.e., $f \in \mathrm{D}_E(R, S)$. □

This characterization relates a differential property with linear algebra, more precisely, with module theory. Moreover, if we add the Noetherianity condition to R, we can prove the closure properties of addition, multiplication and derivation. Hence, we obtain that $\mathrm{D}_E(R, S)$ is a differential extension of R contained in E.

We denote by $M_{R_S}(f)$ the R_S-module generated by f and all its derivatives, omitting R_S when the ring R and the set S can be understood from the context.

Theorem 2.6. *Let (R, ∂) be a Noetherian differential integral domain, $E \supset R$ a differential extension and $S \subset R$ a multiplicatively closed set. Let $f, g \in \mathrm{D}_E(R, S)$. Then:*

- *$f + g \in \mathrm{D}_E(R, S)$.*
- *$fg \in \mathrm{D}_E(R, S)$.*
- *$\partial(f) \in \mathrm{D}_E(R, S)$.*
- *$\int f \in \mathrm{D}_E(R, S)$.*

In particular, $\mathrm{D}_E(R, S)$ is a differential extension of R contained in E.

Proof. Using the basic properties of the derivation, we can easily prove the following inclusions of modules:

$$M(f+g) \subset M(f)+M(g), \qquad M(fg) \subset M(f)M(g),$$
$$M(\partial(f)) \subset M(f),$$

where $M(f)M(g)$ is the module generated by the product of elements of $M(f)$ and $M(g)$.

Since R is Noetherian, R_S is also Noetherian and then $M(f)$ and $M(g)$ are Noetherian modules. This implies that $M(f)+M(g)$ and $M(f)M(g)$ are also Noetherian [2, Chapter 6]. Since $M(f+g)$, $M(fg)$ and $M(\partial(f))$ are submodules of Noetherian modules, they are finitely generated, proving, by Theorem 2.5, that

$$f+g,\ fg,\ \partial(f) \in \mathrm{D}_E(R,S).$$

For the antiderivative $\int f$, we have a direct formula for the resulting differential equation since for any operator $\mathcal{A} \in R[\partial]$ such that $\mathcal{A} \cdot f = 0$ we obtain $(\mathcal{A}\partial) \cdot \left(\int f\right) = 0$.

We can also check that $R \subset \mathrm{D}_E(R,S)$. Let $r \in R$ and consider the module $M(r)$. This is a submodule of R_S (since R is a differential ring) and it is finitely generated because R is Noetherian. By Theorem 2.5, we conclude that $r \in \mathrm{D}_E(R,S)$. □

This proof is very similar to the proof of closure properties of differentially definable functions [12, Propositions 5 and 6]. However, the Noetherianity condition is necessary to guarantee that the modules $M(f+g), M(fg)$ and $M(\partial(f))$ are finitely generated.

An important difference to the case of differentially definable functions is that here we do not have an explicit bound for the order of the resulting differential equation. The methods that we propose here are based on an exhaustive search of annihilating operators increasing the order by one in every step. Termination is guaranteed, but without an a priori bound.

3 SIMPLE D-FINITE FUNCTIONS

In this section, we take $R = \mathbb{K}[x]$ and $E = \mathbb{K}[[x]]$. For any multiplicatively closed set $S \subset \mathbb{K}[x]$, the ring $\mathrm{D}_{\mathbb{K}[[x]]}(\mathbb{K}[x], S)$ will be a differential subring of the D-finite functions and we call them *S-simple D-finite functions.*

The set S controls the possible singularities of the S-simple D-finite functions:

Lemma 3.1. *Let $S \subset \mathbb{K}[x]$ be a multiplicatively closed set and $f(x)$ be an S-simple D-finite function. If $\alpha \in \mathbb{C}$ is a singularity of $f(x)$, then there is $s(x) \in S$ such that $s(\alpha) = 0$.*

Proof. Since $f(x)$ is S-simple, there is a differential operator $\mathcal{A} \in \mathbb{K}[x][\partial]$ such that $\mathcal{A} \cdot f(x) = 0$ and $\mathrm{lc}(\mathcal{A}) \in S$. If $\alpha \in \mathbb{C}$ is a singularity of $f(x)$, then $\mathrm{lc}(\mathcal{A})(\alpha) = 0$. □

The following sets for S are worth of consideration:

- $S = \mathbb{K} \setminus \{0\}$: these functions have no singularities.
- $S = \left\{(x-\alpha_1)^{i_1} \cdots (x-\alpha_n)^{i_n} \ :\ i_j \in \mathbb{N}\right\}$: these functions can only have singularities on $\alpha_1, \ldots, \alpha_n$.
- $S = \mathbb{K}[x] \setminus \mathfrak{p}$ where $\mathfrak{p}$ is a prime ideal: these functions *avoid* singularities on the zero set of the ideal.

In particular, if $f(x), g(x)$ are two D-finite functions that satisfy linear differential equations with leading coefficients $p_f(x), p_g(x) \in \mathbb{K}[x]$, then we can consider $S = \{p_f(x)^i p_g(x)^j \ :\ i,j \in \mathbb{N}\}$ and show that any polynomial combination of $f(x)$ and $g(x)$ is annihilated by an S-simple differential operator.

Example 3.2 (Adding analytic functions). Let $f(x)$ and $g(x)$ be two D-finite functions annihilated by the differential operators

$$\mathcal{A} = \partial_x^2 + 1, \quad \mathcal{B} = \partial_x^2 - x.$$

Consider the function $h(x) = f(x) + g(x)$. With classical computations, we get that $h(x)$ is annihilated by the differential operator

$$\begin{aligned}\mathcal{F} = \ & (x+1)^2\partial_x^4 - 2(x+1)\partial_x^3 - (x^3+x^2-x-3)\partial_x^2 - \\ & 2(x+1)\partial_x - (x^3+2x^2+x-2).\end{aligned} \tag{2}$$

However, this operator has a non-constant leading coefficient. Applying Theorem 2.6 with $R = \mathbb{K}[x]$ and $S = \mathbb{K}^*$, we know that there is an operator with constant leading coefficient that annihilates $h(x)$.

If we search for it with an ansatz, we need to compute a $\mathbb{K}[x]$-linear combination that yields $h^{(n)}(x)$ for some $n \in \mathbb{N}$. We can express these derivatives in term of the derivatives of $f(x)$ and $g(x)$ in the following way:

- $h(x) = f(x) + g(x)$.
- $h'(x) = f'(x) + g'(x)$.
- $h''(x) = -f(x) + xg(x)$.
- $h'''(x) = -f'(x) + g(x) + xg'(x)$.
- $h^{(4)}(x) = f(x) + x^2g(x) + 2g'(x)$.
- $h^{(5)}(x) = f'(x) + 4xg(x) + x^2g'(x)$.

For the case $n = 4$ (which was the original bound for the D-finite computation), the ansatz is as follows:

$$h^{(4)}(x) - \alpha_3 h'''(x) - \alpha_2 h''(x) - \alpha_1 h'(x) - \alpha_0 h(x) = 0,$$

and after substituting the previous equalities, we obtain the following linear system for $(\alpha_0, \ldots, \alpha_3)$ that has to be solved in $\mathbb{K}[x]$:

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & x & 1 \\ 0 & 1 & 0 & x \end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix} = \begin{pmatrix}1\\0\\x^2\\2\end{pmatrix}.$$

This equation has a unique solution in $\mathbb{K}(x)$ which leads to the equation (2), so there is no solution where all the α_i are polynomials. If we increase the order of the ansatz by one:

$$h^{(5)}(x) - \alpha_4 h^{(4)}(x) - \alpha_3 h'''(x) - \alpha_2 h''(x) - \alpha_1 h'(x) - \alpha_0 h(x) = 0,$$

we obtain the following linear system for $\alpha_0, \ldots, \alpha_4$:

$$\begin{pmatrix} 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & x & 1 & x^2 \\ 0 & 1 & 0 & x & 2 \end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\\ \alpha_4\end{pmatrix} = \begin{pmatrix}0\\1\\4x\\x^2\end{pmatrix}.$$

This system has a solution in $\mathbb{K}[x]$:

- $\alpha_0 = -(x^2 + x - 3)$,
- $\alpha_1 = x - 2$,
- $\alpha_2 = 4 - x^2$,
- $\alpha_3 = x - 3$,
- $\alpha_4 = x + 1$,

which means that $h(x)$ is annihilated by the $\mathbb{K}$-simple differential operator:

$$\partial_x^5 - (x+1)\partial_x^4 - (x-3)\partial_x^3 - \\ (4-x^2)\partial_x^2 - (x-2)\partial_x + (x^2+x-3).$$

Example 3.3 (Preserving singularities in the equation). Now consider $f(x)$ and $g(x)$ the D-finite functions annihilated by the differential operators

$$\mathcal{C} = (x+1)\partial_x^2 + \partial_x \quad \text{and} \quad \mathcal{D} = \partial_x - 1,$$

respectively, and let $h(x) = f(x) + g(x)$. Using classical computations as D-finite functions, we get that $h(x)$ is annihilated by the differential operator

$$(x+1)(x+2)\partial_x^3 - (x^2+2x-1)\partial_x^2 - (x+3)\partial_x.$$

The leading coefficient vanishes at $x = -1$ and $x = -2$, adding one apparent singularity to the resulting differential operator. Applying Theorem 2.6, we know there is a differential operator whose leading coefficient only vanishes at $x = -1$. In fact, using the ansatz method above and solving the corresponding system in the polynomial ring $\mathbb{K}[x]$ localized in the set $\{(x+1)^n \;:\; n \in \mathbb{N}\}$, we obtain that

$$4(x+1)\partial_x^4 + (x^5+4x^4+6x^3+4x^2-9x+2)\partial_x^3 \\ -(x^5+3x^4+2x^3-7x+13)\partial_x^2 - (x^4+4x^3+4x^2+2x-7)\partial_x$$

annihilates $h(x)$ and, as desired, its leading coefficient only vanishes at $x = -1$.

4 IMPLEMENTATION

In this section we detail how the computations described in Examples 3.2 and 3.3 can be generalized to compute the closure properties of Theorem 2.6 for any simple D-finite function. Since some methods depend on the closure property we compute, we indicate this by an asterisk to be replaced by "addition", "multiplication" or "derivation" respectively, and by adjusting the input, i.e., providing two differential operators for the addition and multiplication and just one operator for the derivation.

The idea of the implementation is to use an ansatz method [13] adapted accordingly to the simple case, namely, solving the systems in the localized ring $\mathbb{K}[x]_S$.

We consider $h(x) \in \mathrm{D}(\mathbb{K}[x], S)$ that is either the sum or product of two other functions $f(x), g(x) \in \mathrm{D}(\mathbb{K}[x], S)$ or the derivative of a function $f(x) \in \mathrm{D}(\mathbb{K}[x], S)$ from which we know an S-simple annihilating operator. Recall from proof of Theorem 2.6 that the module

$$M(h) = \langle h^{(m)}(x) \;:\; m \in \mathbb{N}\rangle_{\mathbb{K}[x]_S}$$

is included in a finitely generated $\mathbb{K}[x]_S$-module M that depends directly on the modules $M(f)$ and $M(g)$ (and can be explicitly computed). Let $\phi_1, \ldots, \phi_k$ denote the generators of M. Hence, we can express all the derivatives of $h(x)$ as a linear combination of these generators:

$$h^{(m)}(x) = v_{m,1}\phi_1 + \ldots + v_{m,k}\phi_k,$$

where $v_{m,l} \in \mathbb{K}[x]_S$. The fact that $M(h)$ is finitely generated implies that there is $n \in \mathbb{N}$ such that $h^{(n)}(x)$ is a $\mathbb{K}[x]_S$-linear combination of $h(x)$ and its first $n-1$ derivatives. We can translate this into a problem in the module M.

Let $\mathbf{v}_m^T = (v_{m,1}, \ldots, v_{m,k}) \in \mathbb{K}[x]_S^k$. We can consider the following inhomogeneous ansatz system:

$$(\mathbf{v}_0 | \ldots | \mathbf{v}_{n-1})\, \boldsymbol{\alpha} = \mathbf{v}_n.$$

Computing a solution on $\mathbb{K}[x]_S$ yields an S-simple equation for $h(x)$ since:

$$h^{(n)}(x) - \alpha_{n-1}h^{(n-1)}(x) - \ldots - \alpha_0 h(x) = 0,$$

and we can then clear denominators as we did in (1), obtaining a leading coefficient in S.

This system may have no solution in $\mathbb{K}[x]_S$, as we saw in Example 3.2 for $n = 4$. In this case, we increase the value of n and repeat the process. This method always terminates, since Theorem 2.6 guarantees that the module $M(h)$ is finitely generated.

Hence, in order to implement this ansatz method we need:

(1) For each operation, an algorithm to compute the vectors $\mathbf{v}_m$.
(2) A complete solver of linear systems $A\boldsymbol{\alpha} = \mathbf{b}$ in localized rings $\mathbb{K}[x]_S$.

Algorithm 1 implements the complete process of getting the differential equation for $h(x)$.

Algorithm 1: `get_equation_for_*`

Input : Equations for the operands ($\mathcal{A}, \mathcal{B}$ for addition and product and $\mathcal{A}$ for derivation)
Output : Differential equation for the result $h(x)$
result ←*No solution*;
$i \leftarrow 1$;
while result **is** *No solution* **do**
  $A, \mathbf{b} \leftarrow$ `get_system_for_*`$(*, i)$;
  result ← `solve_system`$(A, \mathbf{b})$;
  $i \leftarrow i + 1$;
$(\alpha_0, \ldots, \alpha_m), T \leftarrow$ result;
$s = \mathrm{lcm}($`denominator`$(\alpha_i), i = 0, \ldots, m)$;
for $i = 0, \ldots, m$ **do**
  $s_i \leftarrow s/$`denominator`(α_i);
  $r_i \leftarrow -s_i *$ `numerator`(α_i);
return $r_0 + r_1\partial + \ldots + r_m\partial^m + s\partial^{m+1}$;

4.1 Computing the ansatz system

Computing the ansatz system requires to compute the representation of $h(x)$ and its derivatives in a $\mathbb{K}[x]_S$-module M generated by the elements $\phi_1, \ldots, \phi_k$. In [11], the same problem was solved for vector spaces. In fact, the theory showed for vector spaces can be easily adapted to differential modules.

Then, the main goal is to compute a *derivation matrix* C of M w.r.t the generators $\phi_1, \ldots, \phi_k$, meaning that, if $p(x) = p_1\phi_1 + \ldots + p_k\phi_k$ and $p'(x) = \hat{p_1}\phi_1 + \ldots \hat{p_k}\phi_k$, then

$$\begin{pmatrix}\hat{p_1}\\ \vdots \\ \hat{p_k}\end{pmatrix} = C\begin{pmatrix}p_1\\ \vdots \\ p_k\end{pmatrix} + \begin{pmatrix}\partial(p_1)\\ \vdots \\ \partial(p_k)\end{pmatrix}.$$

These derivation matrices can be easily computed if we know the derivatives of the generators $\phi_1, \ldots, \phi_k$. Namely, the ith column of the derivation matrix is the list of coordinates of $\partial(\phi_i)$ w.r.t. the same set of generators $\phi_1, \ldots, \phi_k$.

Example 4.1. Let $f(x)$ be an S-simple D-finite function annihilated by the differential operator $\mathcal{A} = p_0(x)+\ldots+p_{d-1}(x)\partial_x^{d-1}+s\partial^d$. We know that $M(f) = \langle f(x), \ldots, f^{(d-1)}(x)\rangle_{\mathbb{K}[x]_S}$, and a derivation matrix of $M(f)$ is

$$C_f = \begin{pmatrix} 0 & 0 & \ldots & 0 & \frac{-p_0(x)}{s} \\ 1 & 0 & \ldots & 0 & \frac{-p_1(x)}{s} \\ 0 & 1 & \ldots & 0 & \frac{-p_2(x)}{s} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & \frac{-p_{d-1}(x)}{s} \end{pmatrix}.$$

This matrix is also known as the *companion matrix of the operator* $\mathcal{A}$ and all its coefficients are in $\mathbb{K}[x]_S$.

If we know the vector $\mathbf{v}_0$ that represents the actual function w.r.t. the generators $\phi_1, \ldots, \phi_k$, we can easily build the ansatz systems by recursively computing the vectors $\mathbf{v}_n$ with the formula:

$$\mathbf{v}_{n+1} = C\mathbf{v}_n + \kappa_{\partial_x}(\mathbf{v}_n),$$

where κ_∂ is the termwise derivation of the vector $\mathbf{v}_n$.

Algorithm 2: `get_system_for_*`

Input: Equations for the operands ($\mathcal{A}, \mathcal{B}$ for addition and product and $\mathcal{A}$ for derivation) and size n of the system

Output: Ansatz system with n columns and the inhomogeneous term

$C \leftarrow$ `derivation_matrix_for_*`$(*)$;
$m \leftarrow$ `ncols`(C);
$\mathbf{v}_0 \leftarrow$ `initial_vector_for_*`$(*)$;
for $i = 1, \ldots, n$ **do**
  $\mathbf{v}_i = C\mathbf{v}_{i-1} + \kappa_{\partial_x}(\mathbf{v}_{i-1})$;
return $(\mathbf{v}_0|\ldots|\mathbf{v}_{n-1}), \mathbf{v}_n$;

For each operation, the derivation matrices can be computed from the companion matrices of the operands [11]. Assume that $f(x)$ and $g(x)$ are S-simple D-functions of orders d_1 and d_2 respectively. Then:

- A derivation matrix of $M(f)$, as in Example 4.1, is C_f.
- A derivation matrix of $M(f) + M(g)$ is the direct sum of the companion matrices $C_f \oplus C_g$.
- A derivation matrix of $M(f)M(g)$ is the Kronecker sum of the companion matrices, denoted by $C_f \boxplus C_g$, and mimics the Leibniz rule of derivation with matrices:

$$C_f \otimes \mathcal{I}_{d_2} + \mathcal{I}_{d_1} \otimes C_g,$$

where $\mathcal{I}_m$ is the identity matrix of size m.

On the other hand, the computation for the vector $\mathbf{v}_0$ can be done for each operation as follows:

- For the derivation, $h(x) = f'(x)$, we get
$$\mathbf{v}_0^T = \mathbf{e}_{d_1,2}^T = (0, 1, 0, \ldots, 0).$$
- For the addition, $h(x) = f(x) + g(x)$, we get
$$\mathbf{v}_0^T = \mathbf{e}_{d_1+d_2,1}^T + \mathbf{e}_{d_1+d_2,d_1+1}^T = (1, 0, \ldots, 0, 1, 0, \ldots, 0)$$
- For the product, $h(x) = f(x)g(x)$, we get
$$\mathbf{v}_0^T = \mathbf{e}_{d_1,1}^T \otimes \mathbf{e}_{d_2,1}^T = (1, 0, \ldots, 0)$$

4.2 Linear systems on localized rings

In order to guarantee the termination of our implementation of the ansatz method, we need a complete solver for linear systems over localized rings, in particular, for localized rings over the polynomial ring $\mathbb{K}[x]$. Here, a complete solver means that we can compute all the solutions to the system.

This problem has been studied in the case of coherent rings [19], which includes the case of the polynomial ring $\mathbb{K}[x]$. Here we describe the approach that is the actual implementation provided in the Sage package `dd_functions`.

First, let us consider one linear equation with coefficients in $\mathbb{K}[x]_S$:

$$v_0\alpha_0 + \ldots + v_{n-1}\alpha_{n-1} = v_n, \tag{3}$$

where α_i are the unknowns. Since $\mathbb{K}[x]$ is a Euclidean domain, so is $\mathbb{K}[x]_S$. Thus, this equation has a solution with all $\alpha_i \in \mathbb{K}[x]_S$ if and only if v_n is in the ideal $(v_0, \ldots, v_{n-1})_{\mathbb{K}[x]_S}$ or, equivalently,

$$\gcd(v_0, \ldots, v_{n-1}) \mid v_n.$$

In fact, using the extended Euclidean algorithm, we can obtain a particular solution to the equation or a message guaranteeing there is no solution.

In order to compute all the solutions to the equation, consider two particular solutions $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$. It is clear that:

$$v_0(\alpha_{1,0} - \alpha_{2,0}) + \ldots + v_{n-1}(\alpha_{1,n-1} - \alpha_{1,n-1}) = 0.$$

The set of solutions to the homogeneous equation is known as the *syzygy* module of the generators $(v_0, \ldots, v_{n-1})$. This syzygy module can be described with a matrix $T \in \mathbb{M}_{n\times p}(\mathbb{K}[x]_S)$ (where p is the dimension of the syzygy module). Then, for any particular solution $\boldsymbol{\alpha}$ and any vector $\boldsymbol{\beta} \in \mathbb{K}[x]_S^p$, the vector

$$\boldsymbol{\gamma} = \boldsymbol{\alpha} + T\boldsymbol{\beta} \in \mathbb{K}[x]_S^n$$

is a solution to the linear equation (3) and, more importantly, all solutions to equation (3) are of this form.

These two computations can be performed simultaneously when computing a Hermite Normal Form. Let g be the greatest common divisor of $v_0, \ldots, v_{n-1}$. Then there is a unimodular matrix $U \in \mathbb{M}_{n\times n}(\mathbb{K}[x]_S)$ such that

$$U\begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Here, the first row of U times v_n/g is the particular solution $\boldsymbol{\alpha}$ and the other rows transposed are exactly the syzygy matrix T.

This approach can always be performed when we work within a Euclidean domain. This is implemented in Algorithm 3.

Now, consider a linear system $A\boldsymbol{\alpha} = \mathbf{b}$. If we look to the last equation, we can solve it using the procedure described above,

Algorithm 3: `solve_equation`

Input : Coefficients $(v_0, \dots, v_{n-1}) \in \mathbb{K}[x]_S$ and an inhomogeneous term v_n
Output : Solution space for $\mathbf{v}\boldsymbol{\alpha} = v_n$
$U, H \leftarrow \mathtt{hermite\_form}((v_0, \dots, v_{n-1})^T)$;
if not $h_{0,0}$ **divides** v_n **then**
  return *No solution*
$\boldsymbol{\alpha}_0 \leftarrow \frac{v_n}{h_{0,0}}(u_{0,0}, \dots, u_{0,n-1})^T$;
$T \leftarrow \mathtt{transpose}\left(\left(u_{i,j}\right)_{i=1,\dots,n-1}^{j=0,\dots,n-1}\right)$;
return $(\boldsymbol{\alpha}_0, T)$;

giving that the solution vector $\boldsymbol{\alpha}$ has a particular shape $\boldsymbol{\alpha}_0 + T\boldsymbol{\beta}$. If we plug this into the original system, we obtain:

$$A\boldsymbol{\alpha}_0 + AT\boldsymbol{\beta} = \mathbf{b},$$

and moving the particular solution to the right-hand side of the equation we obtain a new system:

$$AT\boldsymbol{\beta} = \mathbf{b} - A\boldsymbol{\alpha}_0.$$

By definition of A, T and $\boldsymbol{\alpha}_0$, the matrix AT has the last row equal to zero and $\mathbf{b} - A\boldsymbol{\alpha}_0$ has its last coordinate equal to zero too. Hence, we have a smaller system where we can iterate the process.

We iterate solving one equation each time and returning *No solution* if such an equation has no solution and the whole solution set otherwise. At the end, either the system has no solution or we have solved all the equations, obtaining that all the solutions are of the form:

$$\boldsymbol{\alpha}_0 + T_0\left(\boldsymbol{\alpha}_1 + T_1\left(\dots\left(\boldsymbol{\alpha}_q + T_q\boldsymbol{\beta}\right)\right)\right),$$

for any $\boldsymbol{\beta}$ with coefficients in $\mathbb{K}[x]_S$.

We can then obtain a particular solution with the formula

$$\boldsymbol{\alpha}_0 + T_0\boldsymbol{\alpha}_1 + T_0T_1\boldsymbol{\alpha}_2 + \dots + T_0\cdots T_{q-1}\boldsymbol{\alpha}_q,$$

and we can adapt the solution by adding any vector obtained by multiplying the matrix $T = T_0T_1\cdots T_q$ with any vector $\boldsymbol{\beta}$.

This method is implemented in Algorithm 4.

Algorithm 4: `solve_system`

Input : Matrix A for the system and the inhomogeneous term $\mathbf{b}$
Output : Solution space for $A\boldsymbol{\alpha} = \mathbf{b}$
solution $\leftarrow (0, 0, \dots, 0)$;
$T \leftarrow \mathcal{I}$;
for $i = \mathtt{nrows}(A), \dots, 0$ **do**
  if not $\mathtt{row}(A, i) = (0, \dots, 0)$ **then**
    aux $\leftarrow \mathtt{solve\_equation}(\mathtt{row}(A, i), b_i)$;
    if aux **is** *No solution* **then**
      return *No solution*;
    $\boldsymbol{\alpha}, \tilde{T} \leftarrow$ aux;
    `// Updating the whole solution`
    solution $\leftarrow$ solution $+ T\boldsymbol{\alpha}$;
    $T \leftarrow T\tilde{T}$;
    `// Updating the system`
    $\mathbf{b} \leftarrow \mathbf{b} - A\boldsymbol{\alpha}$;
    $A \leftarrow A\tilde{T}$;
return (solution,T);

5 SIMPLE D^n-FINITE FUNCTIONS

Up to this point, we can compute several operations such as addition and multiplication preserving the zeros of the leading coefficients of the resulting operators. This means that the set of points for possible singularities is fixed throughout all the computations.

However, all these methods and results are based on Theorem 2.6, which requires that the ring of coefficients is Noetherian. In the case of D-finite functions, we know that $\mathbb{K}[x]$ is Noetherian (in particular, it is a Euclidean domain). In order to extend these results to DD-finite functions or, even further, to D^n-finite functions, we would need to prove that the ring of D-finite functions (or, in general $\mathrm{D}_E(R)$) is a Noetherian ring. This is, currently, not known (although we expect D-finite functions are not a Noetherian ring).

Let us recall the concept of D^n-finite functions (given in [10, 12]): the set of D^n-finite functions are then nth step in an iterative construction of differentially definable functions over the polynomial ring $\mathbb{K}[x]$, i.e., $\mathrm{D}^n(\mathbb{K}[x]) = \mathrm{D}(\mathrm{D}^{n-1}(\mathbb{K}[x]))$. With this notation, D-finite functions are D^1-finite functions and DD-finite functions are D^2-finite functions.

Consider S to be a multiplicatively closed subset of $\mathrm{D}^{n-1}(\mathbb{K}[x])$. We are going to avoid the proof of Noetherianity for $\mathrm{D}^n(\mathbb{K}[x])$, but still extend the result of Theorem 2.6 to these functions for any $n \in \mathbb{N}$ and multiplicatively closed set S.

Note that, when we compute with these functions, we do not need to use the whole ring of D^{n-1}-finite functions, but just a smaller ring generated mainly by the coefficients of the differential equations.

Lemma 5.1. *Let (R, ∂) be a Noetherian differential integral domain, $E \supset R$ a differential extension and $F = Q(E)$ the total ring of fractions of E (i.e., the localization of E w.r.t. the set of all non-zero divisors). Let $f_1, \dots, f_m \in \mathrm{D}_E^n(R)$ for some $n \geq 1$. Then there is a Noetherian differential extension $R \subset T \subset F$, such that $f_1, \dots, f_m \in T$.*

Proof. We proceed by induction on n. We start with the base case $n = 1$. Let $\mathcal{A}_i = r_{i,0} + \dots + r_{i,d_i}\partial^{d_i} \in R[\partial]$ be such that $\mathcal{A}_i \cdot f_i = 0$ for all $i = 1, \dots, m$. Consider the following set

$$D = \left\{\prod_{i=1}^{m} r_{i,d_i}^{p_i} \ :\ p_1, \dots, p_m \in \mathbb{N}\right\}.$$

It is clear that $D \subset R$ is a multiplicatively closed set. Consider the ring where we localize R with respect to D and add all the elements f_i and enough of their derivatives in order to get a differential ring. Namely,

$$T = R_D[f_1, \dots, \partial^{d_1-1}(f_1), \dots, f_m, \dots, \partial^{d_m-1}(f_m)] \subset F$$

This ring is Noetherian (since it is a polynomial ring over a Noetherian ring) and we can easily check that $\partial^j(f_i) \in T$ for all $i = 1, \dots, m$ and $j \in \mathbb{N}$. Hence, T is a differential extension of R such that $f_1, \dots, f_m \in T$, finishing the proof of this case.

For the case $n > 1$, we consider $\mathcal{A}_i = g_{i,0} + \dots + g_{i,d_i}\partial^{d_i} \in \mathrm{D}_E^{n-1}(R)[\partial]$ to be such that $\mathcal{A}_i \cdot f_i = 0$ for all $i = 1, \dots, m$. By the induction hypothesis, there is a Noetherian differential extension

$\tilde{T} \subset F$ that contains all the coefficients $g_{i,j}$ for $i = 1, \dots, m$ and $j = 0, \dots, d_i$. By definition of $D_F(\tilde{T})$, it is clear that $f_1, \dots, f_m \in D_F(\tilde{T})$. We can apply now the case $n = 1$, obtaining a Noetherian differential extension $\tilde{T} \subset T \subset F$ that contains all the elements $f_1, \dots, f_m$. □

This lemma guarantees that we can build a Noetherian ring given any finite set of D^n-finite functions. However, we did some localizations over some elements that are the leading coefficients of the differential operators involved. In order to get simple D^n-finite functions we need to take care of those elements and keep track of them, knowing that at the end, we can clear denominators.

Definition 5.2. Let $S \subset \mathbb{K}[[x]]$ be multiplicatively closed. We denote the set of S-simple D^n- finite functions by $D^n(\mathbb{K}[x], S)$, and define them recursively as follows:

- $D^1(\mathbb{K}[x], S) = D(\mathbb{K}[x], S \cap \mathbb{K}[x])$.
- $D^n(\mathbb{K}[x], S) = D(D^{n-1}(\mathbb{K}[x], S), S \cap D^{n-1}(\mathbb{K}[x], S))$.

Observe that for $n = 1$ we obtain the S-simple D-finite functions defined in Section 3. Also, in this definition the set $S \subset \mathbb{K}[[x]]$. In order to fit to the setting of Definition 2.4, we intersect in each layer with $D^{n-1}(\mathbb{K}[x], S)$, so the set to localize is included in the ring of coefficients for the differential operators.

Theorem 5.3. *Let $S \subset \mathbb{K}[[x]]$ be a multiplicatively closed set and $f(x), g(x) \in D^n(\mathbb{K}[x], S)$. Then*

- *$f(x) + g(x) \in D^n(\mathbb{K}[x], S)$.*
- *$f(x)g(x) \in D^n(\mathbb{K}[x], S)$.*
- *$f'(x) \in D^n(\mathbb{K}[x], S)$.*
- *$\int_x f(t)dt \in D^n(\mathbb{K}[x], S)$.*

In particular, the set $D^n(\mathbb{K}[x], S)$ is a differential subring of $\mathbb{K}[[x]]$.

Proof. We proceed by induction on n. The case $n = 1$ is exactly Theorem 2.6 with $R = \mathbb{K}[x]$ and $S = \mathbb{K}[x] \cap S$.

Now, let $n > 1$. Assume that $f(x)$ and $g(x)$ are annihilated respectively by the operators in $D^{n-1}(\mathbb{K}[x], S)[\partial_x]$:

$$\mathcal{A} = s_f \partial_x^{d_1} + \alpha_{d_1-1}\partial_x^{d_1-1} + \dots + \alpha_0,$$

$$\mathcal{B} = s_g \partial_x^{d_2} + \beta_{d_2-1}\partial_x^{d_2-1} + \dots + \beta_0,$$

where $s_f, s_g \in S$.

Using Lemma 5.1, there is a Noetherian differential ring T that contains all the elements α_i, β_j, s_f and s_g. Moreover, following the proof of that Lemma, we know that this ring T is of the form

$$T = \mathbb{K}[x, \gamma_1, \dots, \gamma_k]_D,$$

where $D = \{\eta_1^{k_1} \cdots \eta_m^{k_m} \ : \ k_l \in \mathbb{N}\}$ and the elements η_l are leading coefficients of some linear differential operators (i.e., all the η_l are in $S \cap D^{n-1}(\mathbb{K}[x], S)$) and $\gamma_i \in D^{n-1}(\mathbb{K}[x], S)$.

Consider the set

$$\tilde{S} = \left\{ s_f^i s_g^j \ : \ i, j \in \mathbb{N} \right\} \subset S \cap D^{n-1}(\mathbb{K}[x], S).$$

This set is multiplicatively closed. It is clear now that $f(x), g(x) \in D(T, \tilde{S})$. Hence, applying Theorem 2.6, we have that $f(x) + g(x)$, $f(x)g(x)$, $f'(x)$ and $\int_x f(t)dt$ are also elements in $D(T, \tilde{S})$. They satisfy a differential equation of the shape

$$sh^{(p)}(x) + a_{p-1}h^{(p-1)}(x) + \dots + a_0 h(x) = 0,$$

where $s \in \tilde{S}$ and $a_k \in T$. We can clear the denominators (which are elements of D) and obtain a linear differential equation whose leading coefficient is a product of elements in $S \cap D^{n-1}(\mathbb{K}[x], S)$. Since S is multiplicatively closed, the remaining leading coefficient is in S. On the other hand, by induction, the leading coefficients is in $D^{n-1}(\mathbb{K}[x], S)$, finishing the proof. □

Example 5.4 ($\mathbb{K}[x]$-simple DD-finite functions). The set of DD-finite functions satisfying linear differential equations with polynomial leading coefficients is a differential subring of $\mathbb{K}[[x]]$. For proving that, we apply Theorem 5.3 with $S = \mathbb{K}[x] \setminus \{0\}$ and $n = 2$.

In this ring we can find some special functions such as $e^{e^x - 1}$ and the Mathieu functions [8, Chapter 28]. We can always compute the singularities of the functions included here since we can compute the singularities of the D-finite coefficients and the zeros of the leading coefficient (a polynomial in this case).

Example 5.5 ($\cos(x)$-simple DD-finite functions). Consider now the set of DD-finite functions that satisfy a linear differential equation with a power of $\cos(x)$ as leading coefficient. By Theorem 5.3, this set is a differential subring taking $S = \{\cos(x)^n \ : \ n \in \mathbb{N}\}$.

In order to be in this ring, a function must be annihilated by a differential operator where the leading coefficient is a power of $\cos(x)$ and the other coefficients are $\mathbb{K}$-simple D-finite functions. In this ring we can find special functions as the tangent ($\tan(x)$) or compositions such as $\sin(\sin(x))$.

If $f(x)$ is a function in this ring, the singularities of $f(x)$ are strictly contained in the set $\{(2k+1)\pi/2 \ : \ k \in \mathbb{Z}\}$, which is the zero set of $\cos(x)$.

Example 5.6. In Example 5.5, the coefficients allowed in the operators were not all D-finite functions but only the $\mathbb{K}$-simple functions. If we want to extend the possible coefficients to all the D-finite functions, we need to allow polynomials in the leading coefficient too.

This new set is also a differential ring taking

$$S = \{p(x)^n \cos(x)^m \ : \ p(x) \in \mathbb{K}[x], n, m \in \mathbb{N}\}.$$

This ring is an extension of D-finite functions that includes some special functions and compositions such as $Ai(\sin(x))$, where $Ai(x)$ is an Airy function, i.e., a formal power series annihilated by $\partial_x^2 - x$.

The singularities in this ring can also be computed: they are included in the zeros of the cosine, the zeros of the polynomial factor of the leading coefficient and the singularities of the D-finite coefficients.

6 CONCLUSIONS

In this paper we have shown how we can algebraically control the singularities that are present in differential operators after performing several operations that, classically, do not guarantee that the singularity set of the resulting equation is fixed. This can be applied to the manipulation of differential operators without apparent singularities. Using the methods described in this paper, we can compute directly new differential operators without apparent singularities using only linear algebra. The algorithms for the D-finite case are included in the package `dd_functions`for SageMath.

Furthermore, we have extended this theory to the set of D^n-finite functions. In order to extend the implementation to this wider class, we first need to adapt the algorithms for a multivariate setting, which can be done using Gröbner basis. We then need to compute the algebraic relations between the coefficients of the differential equation. This is a problem only solved for generating functions of C-finite sequences (generating functions whose coefficients satisfy a linear recurrence with constant coefficients) [15].

These results can also be used to build sets of functions and equations where we can explicitly compute their singularities (as in Examples 5.4, 5.5 and 5.6). This is fundamental to build appropriate analytic continuation paths that are used to compute certified numerical evaluations of D-finite functions [16, 17]. Future research in this direction will involve the study of the nature of these singularities.

Acknowledgment. Many thanks to the anonymous referees for their careful reading and detailed comments. Their suggestions helped to significantly improve the quality of the paper.

REFERENCES

[1] Sergei A. Abramov, Moulay A. Barkatou, and Mark van Hoeij. 2006. Apparent singularities of linear difference equations with polynomial coefficients. *Applicable Algebra in Engineering, Communication and Computing* 17, 2 (April 2006), 117–133. https://doi.org/10.1007/s00200-005-0193-9

[2] Michael F. Atiyah and I. G. MacDonald. 1969. *Introduction to commutative algebra.* Addison-Wesley-Longman. I–IX, 1–128 pages.

[3] Moulay A. Barkatou and Suzy S. Maddah. 2015. Removing Apparent Singularities of Systems of Linear Differential Equations with Rational Function Coefficients. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation* (Bath, United Kingdom) *(ISSAC '15)*. Association for Computing Machinery, New York, NY, USA, 53–60. https://doi.org/10.1145/2755996.2756668

[4] Manuel Bronstein. 1997. *Symbolic Integration I.* Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-03386-9

[5] Shaoshi Chen, Maximilian Jaroschek, Manuel Kauers, and Michael F. Singer. 2013. Desingularization Explains Order-Degree Curves for Ore Operators. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation* (Boston, Maine, USA) *(ISSAC '13)*. Association for Computing Machinery, New York, NY, USA, 157–164. https://doi.org/10.1145/2465506.2465510

[6] Shaoshi Chen, Manuel Kauers, Ziming Li, and Yi Zhang. 2019. Apparent singularities of D-finite systems. *Journal of Symbolic Computation* 95 (2019), 217 – 237. https://doi.org/10.1016/j.jsc.2019.02.009

[7] Shaoshi Chen, Manuel Kauers, and Michael F. Singer. 2016. Desingularization of Ore operators. *Journal of Symbolic Computation* 74 (2016), 617 – 626. https://doi.org/10.1016/j.jsc.2015.11.001

[8] DLMF [n.d.]. *NIST Digital Library of Mathematical Functions.* http://dlmf.nist.gov/, Release 1.0.16 of 2017-09-18. http://dlmf.nist.gov/ F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.

[9] Antonio Jiménez-Pastor. 2019. DD-Finite Functions Implemented in Sage. In *Mathematical Aspects of Computer and Information Sciences*, Daniel Slamanig, Elias Tsigaridas, and Zafeirakis Zafeirakopoulos (Eds.). Springer International Publishing, Cham, 457–462. https://doi.org/10.1007/978-3-030-43120-4_36

[10] Antonio Jiménez-Pastor. 2020. *A computable extension for holonomic functions: DD-finite functions.* Ph.D. Dissertation. Doctoral Program Computational Mathematics, Johannes Kepler University. https://epub.jku.at/urn:nbn:at:at-ubl:1-39720/

[11] Antonio Jiménez-Pastor and Veronika Pillwein. 2018. Algorithmic Arithmetics with DD-Finite Functions. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation* (New York, NY, USA) *(ISSAC '18)*. Association for Computing Machinery, New York, NY, USA, 231–237. https://doi.org/10.1145/3208976.3209009

[12] Antonio Jiménez-Pastor and Veronika Pillwein. 2019. A computable extension for D-finite functions: DD-finite functions. *Journal of Symbolic Computation* 94 (2019), 90 – 104. https://doi.org/10.1016/j.jsc.2018.07.002

[13] Manuel Kauers. 2013. The Holonomic Toolkit. In *Texts & Monographs in Symbolic Computation.* Springer Vienna, 119–144. https://doi.org/10.1007/978-3-7091-1616-6_5

[14] Manuel Kauers, Maximilian Jaroschek, and Fredrik Johansson. 2015. Ore Polynomials in Sage. In *Lecture Notes in Computer Science.* Springer International Publishing, 105–125. https://doi.org/10.1007/978-3-319-15081-9_6

[15] Manuel Kauers and Burkhard Zimmermann. 2006. Computing the algebraic relations of C-finite sequences and multisequences. *Journal of Symbolic Computation* 2007 (2006).

[16] Marc Mezzarobba. 2010. NumGfun: a Package for Numerical and Analytic Computation with D-finite Functions. In *ISSAC '10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation* (Munich, Germany), Stephen M. Watt (Ed.). ACM, 139–146. https://doi.org/10.1145/1837934.1837965

[17] Marc Mezzarobba. 2011. *Autour de l'évaluation numérique des fonctions D-finies.* Thèse de doctorat. École polytechnique. http://tel.archives-ouvertes.fr/pastel-00663017/

[18] Veronika Pillwein. 2008. *Computer Algebra Tools for Special Functions in High Order Finite Element Methods.* Ph.D. Dissertation. Johannes Kepler University Linz.

[19] Sebastian Posur. 2018. Linear systems over localizations of rings. *Archiv der Mathematik* 111, 1 (April 2018), 23–32. https://doi.org/10.1007/s00013-018-1183-z

[20] Bruno Salvy and Paul Zimmermann. 1994. GFUN: A Maple Package for the Manipulation of Generating and Holonomic Functions in One Variable. *ACM Trans. Math. Softw.* 20, 2 (June 1994), 163–177. https://doi.org/10.1145/178365.178368

[21] William A. Stein et al. 2020. *Sage Mathematics Software (Version 9.1).* The Sage Development Team. http://www.sagemath.org.