Finding Hyperexponential Solutions of Linear ODEs by Numerical Evaluation

Fredrik Johansson[*]
RISC
Johannes Kepler University
4040 Linz, Austria
fjohanss@risc.jku.at

Manuel Kauers[*]
RISC
Johannes Kepler University
4040 Linz, Austria
mkauers@risc.jku.at

Marc Mezzarobba
Inria, Univ. Lyon, AriC, LIP[†]
ENS de Lyon, 46 allée d'Italie
69364 Lyon Cedex 07, France
marc@mezzarobba.net

ABSTRACT

We present a new algorithm for computing hyperexponential solutions of linear ordinary differential equations with polynomial coefficients. The algorithm relies on interpreting formal series solutions at the singular points as analytic functions and evaluating them numerically at some common ordinary point. The numerical data is used to determine a small number of combinations of the formal series that may give rise to hyperexponential solutions.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation—*Algorithms*

General Terms

Algorithms

Keywords

Closed form solutions, D-finite equations, Effective analytic continuation

1. INTRODUCTION

We consider linear differential operators

$$P = p_r D^r + p_{r-1} D^{r-1} + \cdots + p_0$$

where $p_0, \dots, p_r$ are polynomials and D represents the standard derivation $\frac{d}{dx}$. Such operators act in a natural way on elements of a differential ring containing the polynomials. An object y is called a solution of the operator if P applied to y yields zero. We are interested in finding the hyperexponential solutions of a given operator. An object y is called hyperexponential if the quotient $D(y)/y$ can be identified with a rational function. Typical examples are rational functions (e.g. $(5x+3)/(3x+5)$), radicals (e.g. $\sqrt{x+1}$), exponentials (e.g. $\exp(3x^2-4)$ or $\exp(1/x)$), or combinations of these (e.g. $\sqrt{x+1}\exp(x^9/(x-1))$). Equivalently, y is called hyperexponential if there is some first order operator $q_1 D + q_0$ with q_0, q_1 polynomials which maps y to zero. If we regard differential operators as elements of an operator algebra $C(x)[D]$, then there is a one-to-one correspondence between the hyperexponential solutions y of an operator P and its first order right hand factors. In other words, if y is a hyperexponential term with $(q_1 D + q_0) \cdot y = 0$, then y is a solution of P if and only if there exist rational functions $u_0, \dots, u_{r-1}$ such that

$$P = (u_{r-1} D^{r-1} + u_{r-2} D^{r-2} + \cdots + u_0)(q_1 D + q_0).$$

Algorithms for finding the hyperexponential solutions of a linear differential equation (or equivalently, the first order right hand factors of the corresponding operators) are known since long. They are needed as subroutine in algorithms for factoring operators or for finding Liouvillian solutions. See Chapter 4 of [12] for details and references.

Classical algorithms first compute "local solutions" at singular points (cf. Section 2.3 below) and then test for each combination of local solutions whether it gives rise to a hyperexponential solution. This leads to a combinatorial explosion with exponential runtime. The situation is similar to classical algorithms for factoring polynomials over $\mathbb{Q}$, which first compute the irreducible factors modulo a prime and then test for each combination whether it gives rise to a factor in $\mathbb{Q}[x]$.

The algorithm of van Hoeij [13] avoids the combinatorial explosion as follows. It picks one local solution and considers the operator $Q = q_1 D + q_0$ with $q_1, q_0 \in C((x))$ which annihilates it. This operator is a right factor of P, though not with rational coefficients. The algorithm then constructs (if possible) a left multiple B of Q with rational coefficients of order at most $r-1$. This leads to a nontrivial factorization $P = AB$ in $C(x)[D]$. The procedure is then applied recursively to A and B until a complete factorization is found. The first order factors in this factorization give rise to at most r hyperexponential candidate solutions (possibly up to multiplication by a rational function). These are then checked in a second step. Van Hoeij's algorithm reminds of the polynomial factorization algorithm of Lenstra, Lenstra, Lovász [7, 16], which picks one modular factor and constructs (if possible) a multiple of this factor with integer

[*]Supported by the Austrian Science Fund (FWF) grant Y464-N18.
[†]UMR 5668 CNRS – ENS Lyon – Inria – UCBL

ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.

coefficients but smaller degree than the original polynomial. This multiple is then a proper divisor in $\mathbb{Q}[x]$.

The algorithm we propose below avoids the combinatorial explosion in a different way. We start from the local solutions and regard them as asymptotic expansions of complex functions. By means of effective analytic continuation and arbitrary-precision numerical evaluation, we compute the values of these functions at some common ordinary reference point. Then a linear algebra algorithm is used to determine a small list of possible combinations of local solutions that may give rise to hyperexponential ones, possibly up to multiplication by a rational function. These are then checked in a second step. Our approach was motivated by van Hoeij's polynomial factorization algorithm [15], which associates to every modular factor a certain vector and then uses lattice reduction to determine a small list of combinations that may give rise to proper factors.

Although our algorithm avoids the combinatorial explosion problem, we do not claim that it runs in polynomial time. Indeed, no polynomial time algorithm can be expected because there are operators P which have hyperexponential solutions y that are exponentially larger than P. Also van Hoeij [13] makes no formal statement about the complexity of his algorithm. It is clear though that his algorithm is superior to the naive algorithm. Similarly, we believe that our algorithm has chances to outperform van Hoeij's algorithm, at least in examples that are not deliberately designed to exhibit worst case performance. The reason is partly that during the critical combination phase we only work with floating point numbers of moderate precision while van Hoeij's algorithm in general needs to do arithmetic in algebraic number fields whose degrees may grow during the computation. Another advantage of our algorithm is that it is conceptually simpler than van Hoeij's, at least if we take for granted that we can compute high-precision evaluations of D-finite functions.

In Section 2 we recall known results and definitions which we use. Section 3 contains a global description of the whole algorithm for finding hyperexponential solutions. Section 4 contains an abstract description of the algorithm for the combination phase, and in Section 5 we explain how this algorithm can be implemented using numerical evaluation. A detailed example is given in Section 6.

2. PRELIMINARIES

In this section, we recall some results from the literature and introduce notation that will be used in subsequent sections.

2.1 Differential Fields and Operator Algebras

A differential ring/field is a pair (K, D) where K is a ring/field and $D\colon K \to K$ is a derivation on K, i.e., a map satisfying $D(a+b) = D(a)+D(b)$ and $D(ab) = D(a)b+aD(b)$ for all $a, b \in K$. Throughout this paper, we consider the differential field $K = C(x)$, where C is some (computable) subfield of $\mathbb{C}$, together with the derivation $D\colon K \to K$ defined by $D(c) = 0$ for all $c \in C$ and $D(x) = 1$. For simplicity, we assume throughout that C is algebraically closed.

A differential ring/field E is called an extension of K if $K \subseteq E$, and the derivation of E restricted to K agrees with the derivation of K.

By $K[D]$ we denote the set of all polynomials in the indeterminate D with coefficients in K. Addition in $K[D]$ is defined in the usual way, and multiplication is defined subject to the commutation rule $Da = aD + D(a)$ for $a \in K$. The elements of $K[D]$ are called operators, and they act on the elements of some extension E of K in the obvious way: If $P = p_0 + p_1 D + \cdots + p_r D^r$ is an operator of order r and $y \in E$, then $P \cdot y := \sum_{i=1}^{r} p_i D^i(y) \in E$. The noncommutative multiplication is compatible with operator application in the sense that we have $(PQ) \cdot y = P \cdot (Q \cdot y)$ for all $P, Q \in K[D]$ and all $y \in E$.

The elements $y \in E$ such that $P \cdot y = 0$ form a C-vector space V with $\dim V \leq r$. By making E sufficiently large it can always be assumed that $\dim V = r$.

2.2 Hyperexponential Terms

Let E be an extension of K. An element $h \in E \setminus \{0\}$ is called *hyperexponential* over K if $D(h)/h \in K$. Equivalently, h is hyperexponential if $Q \cdot h = 0$ for some nonzero first order operator $Q \in K[D]$.

Two hyperexponential terms h_1, h_2 are called *equivalent* if $h_1/h_2 \in K$. For example, the terms $\exp(3x^2 - x)$ and $(1-2x)^2 \exp(3x^2 - x)$ are equivalent, but $\exp(3x^2 - x)$ and $(1-2x)^{\sqrt{2}} \exp(3x^2 - x)$ are not. (Here and below, we use standard calculus notation to refer to elements of some extension E on which the derivation acts as the notation suggests, e.g. $D(\exp(3x^2 - x)) = (6x - 1)\exp(3x^2 - x)$.)

Every hyperexponential term can be written in the form $h = \exp(\int v)$, where v is a rational function. The additive constant of the integral amounts to a multiplicative constant for h, which is irrelevant in our context, because $P \cdot h = 0$ if and only if $P \cdot (ch) = 0$ for every $c \in C \setminus \{0\}$. If we consider the partial fraction decomposition of v and integrate it termwise, we obtain something of the form

$$g + \sum_{i=1}^{n} \gamma_i \log(p_i)$$

with $g \in K$, $\gamma_1, \ldots, \gamma_n \in C$ and monic square free pairwise coprime polynomials $p_i \in C[x]$. In terms of this representation, two hyperexponential terms are equivalent if the difference of the corresponding rational functions g is a constant and any two corresponding coefficients γ_i differ by an integer.

The equivalence class of a hyperexponential term h is called the *exponential part* of h. The motivation for this terminology is that when we are searching for some hyperexponential solution h of P and we already know its equivalence class, then we can take an arbitrary element h_0 from this class and make an ansatz $h = uh_0$ for some rational function $u \in K$. The operator $\tilde{P} := P \otimes \left(D - \frac{D(1/h_0)}{1/h_0}\right) \in K[D]$ then has the property that u is a solution of $\tilde{P}$ if and only if uh_0 is a solution of P. This reduces the problem to finding rational solutions, which is well understood and will not be discussed here [1, 12].

2.3 Local Solutions

Consider an operator $P \in C(x)[D]$ of order r. By clearing denominators, if necessary, we may assume that $P \in C[x][D]$, say $P = p_r D^r + \cdots + p_0$ with $p_r \neq 0$. A point $z \in \mathbb{C} \cup \{\infty\}$ is called *singular* if z is a root of p_r, or $z = \infty$. A point which is not singular is called *ordinary*. Note that there are only finitely many singular points, and that we include the "point at infinity" always among the singular points.

If $z = 0$ is an ordinary point then P admits r linearly independent power series solutions. If $z = 0$ is a singular point, it is still possible to find r linearly independent generalized series solutions of the form

$$x^\alpha \exp(u(x^{-1/s})) \sum_{k=0}^{m} b_k(x^{1/s}) \log(x)^k \tag{1}$$

where $\alpha \in C$, $u \in C[x]$ with $u(0) = 0$, $s \in \mathbb{N}$, $m \in \mathbb{N}$ and $b_0, \dots, b_m \in C[[x]]$. These solutions are called the *local solutions* at 0. Their computation is well-known and will not be discussed here. (See [14, 12] and the references given there for details.)

Two series as in (1) are called equivalent if they have the same u and s and the difference of the respective values of α is in $\frac{1}{s}\mathbb{Z}$. The equivalence classes of generalized series under this equivalence relation are called the *exponential parts* of the series. Adopting van Hoeij's notation and defining $\mathrm{Exp}(e) := \exp(\int \frac{e}{x})$ for $e \in C[x^{-1/s}]$, we have that $\mathrm{Exp}(e_1)$ and $\mathrm{Exp}(e_2)$ are equivalent iff $e_1 - e_2 \in \frac{1}{s}\mathbb{Z}$. Note that if $m = 0$ and $s = 1$, two series are equivalent iff their quotient can be identified with a formal Laurent series. We will from now on make no notational distinction between $\mathrm{Exp}(e)$ and its equivalence class.

A point $z \neq 0$ can be moved to the origin by the change of variables $\tilde{x} = x - z$ (if $z \in C$) or $\tilde{x} = 1/x$ (if $z = \infty$). If $\tilde{P}$ is the operator obtained from P by replacing x by $\tilde{x} + z$ or $1/\tilde{x}$, then a local solution of $P \in C[x][D]$ at z is defined as the local solution of $\tilde{P} \in C[\tilde{x}][D]$ at 0.

Throughout the rest of this paper, we will use the following notation. P is some operator in $C[x][D]$ of order r, by $z_1, \dots, z_{n-1} \in C$ we denote its finite singular points, $z_n = \infty$. We write $\tilde{x}_i = x - z_i$ $(i = 1, \dots, n-1)$ and $\tilde{x}_n = 1/x$ for the variables with respect to which the singularities at z_i appear at the origin. For $i = 1, \dots, n$, we consider the vector space V_i generated by all local solutions at z_i. There may be solutions with different exponential parts, say ℓ_i different parts $\mathrm{Exp}(e_{i,1}), \dots, \mathrm{Exp}(e_{i,\ell_i})$ for $e_{i,j} \in C[\tilde{x}_i^{-1/s_{i,j}}]$. By

$$V_{i,j} = V_i \cap \mathrm{Exp}(e_{i,j}) C((\tilde{x}_i^{1/s_{i,j}}))[\log \tilde{x}_i]$$

we denote the vector space of all local solutions of P at z_i with exponential part (equivalent to) $\mathrm{Exp}(e_{i,j})$. Our $V_{i,j}$ are written $V_{e_{i,j}}(P)$ in van Hoeij's papers [14, 13].

The condition in the definition of equivalence that the difference of corresponding values of α be an integer (rather than, say, requiring exactly the same value of α) ensures that the $V_{i,j}$ are indeed vector spaces, because if some $V_{i,j}$ contains, for example, the two series

$$x^\alpha(1 + x + x^2 + \cdots) \quad \text{and} \quad x^\alpha(1 + x + 3x^2 + \cdots)$$

then it must also contain their difference $x^\alpha(2x^2 + \cdots) = x^{\alpha+2}(2 + \cdots)$.

2.4 Analytic Solutions

It is classical that the formal power series solutions $\hat{y}$ of P at an ordinary point $z \in \mathbb{C}$ actually converge in a neighbourhood of z and thus give rise to analytic function solutions y of P. The correspondence is one-to-one. For any other ordinary point $z' \in \mathbb{C}$ and a path $z \rightsquigarrow z'$ avoiding singular points there exists a matrix $M_{z \rightsquigarrow z'} \in \mathbb{C}^{r \times r}$ such that

$$\left(D^j y(z')\right)_{j=0}^{r-1} = M_{z \rightsquigarrow z'} \left(D^j y(z)\right)_{j=0}^{r-1}$$

for every solution y analytic near z. There are algorithms [5, 9] for efficiently computing the entries of $M_{z \rightsquigarrow z'}$ for any given polygon path $z \rightsquigarrow z'$ with vertices in $\bar{\mathbb{Q}}$ to any desired precision. In other words, we can compute arbitrary precision approximations of y and its derivatives at every ordinary point ("effective analytic continuation").

Assume now that 0 is a singular point, and consider the case $s = 1$ and $m = 0$, i.e., let $\hat{y} = \mathrm{Exp}(e)b$ for some $e \in C[x^{-1}]$ and $b \in C[[x]]$ be a formal solution of P. To give an analytic meaning to $\mathrm{Exp}(e) = \exp(\int \frac{e}{x}) = \exp(u + \alpha \log x) = x^\alpha \exp(u)$ (for suitable $\alpha \in C$ and $u \in C[x^{-1}]$) amounts to making a choice for a branch of the logarithm. Every choice gives rise to the same function up to some multiplicative constant.

Since $\mathrm{Exp}(e)b$ is a solution of P iff b is a solution of the operator $P \otimes (D + \frac{e}{x})$, we may assume that $e = 0$. Then the problem remains that the formal power series $\hat{y} = b$ may not be convergent if 0 is a singular point. However, by resummation theory [2, 3] it is still possible to associate to $\hat{y}$ an analytic function y defined on some sector

$$\Delta = \Delta(d, \varphi, \rho) := \{z \in \mathbb{C} : 0 < |z| \leq \rho \wedge |d - \arg z| \leq \varphi/2\}$$

(with $d \in [0, 2\pi]$, $\rho, \varphi > 0$) such that $\hat{y}$ is the asymptotic expansion of y for $z \to 0$ in Δ.

The precise formulation of this result is technical and not really needed for our purpose (see [3, Chap. 6, 10, and 11] or [2, Chap. 5–7] for full details). It will be more than sufficient to know the following facts:

- For every $\boldsymbol{k} = (k_1, \dots, k_q) \in \mathbb{Q}^q$ with $k_1 > \cdots > k_q$ and every $\boldsymbol{d} = (d_1, \dots, d_q) \in [0, 2\pi]^q$ such that
$$|d_{j+1} - d_j| \leq (k_{j+1}^{-1} - k_j^{-1})\tfrac{\pi}{2}, \quad j = 1, \dots, q-1,$$
one constructs [3, §10.2] a differential subring $\mathbb{C}\{x\}_{\boldsymbol{k},\boldsymbol{d}}$ of $\mathbb{C}[[x]]$ [3, Theorems 51 and 53] which contains the ring $\mathbb{C}\{x\}$ of all convergent power series.

- There is a differential ring homomorphism [3, Theorems 51 and 53] $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}$ from $\mathbb{C}\{x\}_{\boldsymbol{k},\boldsymbol{d}}$ to the germs of analytic functions defined on sectors of the form $\Delta(d_1, \varphi, \rho)$ for suitable $\varphi, \rho > 0$, with the property that for every $\hat{y} \in \mathbb{C}\{x\}_{\boldsymbol{k},\boldsymbol{d}}$ the function $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(\hat{y})$ has $\hat{y}$ as its asymptotic expansion for $z \to 0$ [3, §10.2, Exercice 2]. The $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}$ map convergent formal power series to their sum in the usual sense [3, Lemmas 8 and 20].

- For a given operator $P \in C[x][D]$ of order r, one can compute a tuple $\boldsymbol{k}$ and finite subsets $\mathcal{D}_1, \dots, \mathcal{D}_q$ of $[0, 2\pi]$ such that any $\hat{y} \in \mathbb{C}[[x]]$ with $P \cdot \hat{y} = 0$ belongs to $\mathbb{C}\{x\}_{\boldsymbol{k},\boldsymbol{d}}$ for all $\boldsymbol{d}$ as above with $d_1 \notin \mathcal{D}_1, \dots, d_q \notin \mathcal{D}_q$. Additionally, given such a $\boldsymbol{d}$, one can compute $\varphi, \rho > 0$ such that each $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(\hat{y})$ is defined on $\Delta(d_1, \varphi, \rho)$.

- Furthermore, given a point $z \in \Delta(d_1, \varphi, \rho)$, a precision $\varepsilon > 0$, and $\hat{y} \in \mathbb{C}[[x]]$ with $P \cdot \hat{y} = 0$, one can efficiently compute an approximation Y_ε of the vector $Y(z) = (D^j \mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(\hat{y}))_{j=0}^{r-1}$ such that $\|Y(z) - Y_\varepsilon\| \leq \varepsilon$.

The computational part of the last two items is a special case of Theorem 7 of van der Hoeven [11]. As an application, van der Hoeven [10] shows how to factor differential operators using numerical evaluation. Note that our k_j correspond to $1/k_j$ in van der Hoeven's articles, and the components of the tuples $\boldsymbol{k}$ and $\boldsymbol{d}$ appear in reverse order.

Also observe that in the last item, z is an ordinary point, so that from there we can use effective analytic continuation to compute values of $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(\hat{y})$ and its derivatives at any other ordinary point.

3. OUTLINE OF THE ALGORITHM

A hyperexponential term h can be expanded as a generalized series at every point $z \in \mathbb{C} \cup \{\infty\}$, in particular at its singularities. The resulting generalized series are local solutions of P if h is a solution of P. If $h = \exp(\int v)$ is a hyperexponential solution where $v \in \mathbb{C}(x)$, and if we write the partial fraction decomposition of v in the form

$$v = \frac{e_1}{x - z_1} + \frac{e_2}{x - z_2} + \cdots + \frac{e_n}{1/x},$$

where the e_i are polynomials in $\tilde{x}_i^{-1}$, then expanding this h at z_i yields a generalized series in $\tilde{x}_i$ whose exponential part matches $\operatorname{Exp}(e_i)$. The components e_i in the decomposition of v must hence show up among the exponential parts of the local solutions of P.

If $\operatorname{Exp}(e_{i,1}), \ldots, \operatorname{Exp}(e_{i,\ell_i})$ are (representatives of) the different exponential parts that appear among the local solutions at z_i, then any hyperexponential solution must be equivalent to the term $\exp(\int (\frac{e_{1,j_1}}{\tilde{x}_1} + \cdots + \frac{e_{n,j_n}}{\tilde{x}_n}))$ for some tuple $(j_1, \ldots, j_n)$. It then remains to check for each of these candidates whether some element of its equivalence class solves the given equation. The basic structure of the algorithm for finding hyperexponential solutions is thus as follows.

Algorithm 1. *Input: a linear differential operator* $P = p_0 + p_1 D + \cdots + p_r D^r$, $p_r \neq 0$, *with coefficients in* $C[x]$. *Output: all the hyperexponential terms* h *with* $P \cdot h = 0$.

1. *Let* $z_1, \ldots, z_{n-1} \in \mathbb{C}$ *be the roots of* p_r *in* $\mathbb{C}$, *and let* $z_n = \infty$.
2. *For* $i = 1, \ldots, n$ *do*
3. *Find the exponential parts* $\operatorname{Exp}(e_{i,1}), \ldots, \operatorname{Exp}(e_{i,\ell_i})$ *of the local solutions of* P *at* z_i.
4. *Determine a set* $U \subseteq \{1, \ldots, \ell_1\} \times \cdots \times \{1, \ldots, \ell_n\}$ *s.t. for every hyperexponential solution* h *equivalent to* $\exp\bigl(\int \sum_{i=1}^n \frac{e_{i,j_i}}{\tilde{x}_i}\bigr)$ *we have* $(j_1, \ldots, j_n) \in U$.
5. *For each* $(j_1, \ldots, j_n) \in U$ *do*
6. *Let* $h_0 := \exp\bigl(\int \sum_{i=1}^n \frac{e_{i,j_i}}{\tilde{x}_i}\bigr)$, *and compute the operator* $\tilde{P} := P \otimes (D - \frac{D(1/h_0)}{1/h_0})$.
7. *Compute a basis* $\{u_1, \ldots, u_m\} \subseteq C(x)$ *of the vector space of all rational solutions of* $\tilde{P}$, *and output* $u_1 h_0$, $\ldots$, $u_m h_0$.

There is some freedom in step 4 of this algorithm. A naive approach would simply be to take all possible combinations, i.e., $U = \{1, \ldots, \ell_1\} \times \cdots \times \{1, \ldots, \ell_n\}$. This is a finite set, but its size is in general exponential in the number of singular points. For finding a smaller set U, Cluzeau and van Hoeij [6] use modular techniques to quickly discard unnecessary tuples. Our algorithm, explained in the following section, addresses the same issue. It computes a set U of at most r tuples.

4. THE COMBINATION PHASE

In general, the differential operator P may have several different solutions with the same exponential part, i.e., the dimension of the vector spaces $V_{i,j}$ might be greater than one. In this case, it might be that $V_{i,j}$ contains some series which is the expansion of a hyperexponential solution h at z_i as well as some other series which are not. If we compute some basis of $V_{i,j}$, we cannot expect it to contain the expansion of h. Instead, each basis element will in general be the linear combination of this series and some other one. Now, if the expansion of h at some other singular point $z_{i'}$ belongs to the space $V_{i',j'}$ (which possibly also has higher dimension), then, in some sense, h must belong to the intersection of the vector spaces $V_{i,j}$ and $V_{i',j'}$.

Our algorithm is based on testing which intersections are nontrivial. To make these intersections meaningful, we must first map the vector spaces we want to intersect into a common ambient space W. Let E be some differential ring containing $C(x)$ as well as all the hyperexponential solutions of P, and let $W \subseteq E$ be the C-vector space generated by solutions of P in E. For each i, let π_i be some vector space homomorphism

$$\bigoplus_{j=1}^{\ell_i} \operatorname{Exp}(e_{i,j})\mathbb{C}((\tilde{x}_i^{1/s_{i,j}}))[\log \tilde{x}_i] \supseteq V_i \xrightarrow{\pi_i} W$$

with the following properties:

1. The sum $\pi_i(V_{i,1}) + \cdots + \pi_i(V_{i,\ell_i})$ is direct.
2. If $h \in W$ is hyperexponential, then $\pi_i^{-1}(h)$ contains the formal series expansion $\hat{h}$ of h at z_i, possibly up to a multiplicative constant.

Define $W_{i,j} := \pi_i(V_{i,j})$. If h is some hyperexponential solution of P, say with exponential part

$$\exp\Bigl(\int \Bigl(\frac{e_{1,j_1}}{\tilde{x}_1} + \frac{e_{2,j_2}}{\tilde{x}_2} + \cdots + \frac{e_{n,j_n}}{\tilde{x}_n}\Bigr)\Bigr),$$

then $h \in W_{i,j_i}$ for all i, and hence the vector space $W_{\boldsymbol{j}} := W_{1,j_1} \cap \cdots \cap W_{n,j_n}$ is not the zero subspace (because it contains at least h). Our main observation is that there can be at most r tuples $\boldsymbol{j} = (j_1, \ldots, j_n)$ for which $W_{\boldsymbol{j}} \neq \{0\}$, and that they can be computed efficiently once we have bases of the $W_{i,j}$.

Postponing the discussion of making the π_i constructive to the next section, assume for the moment that W is some vector space over C, let $r = \dim W < \infty$ be its dimension, and suppose we are given n different decompositions of subspaces of W into direct sums:

$$\begin{gathered} W_{1,1} \oplus W_{1,2} \oplus \cdots \oplus W_{1,\ell_1} \subseteq W, \\ W_{2,1} \oplus W_{2,2} \oplus \cdots \oplus W_{2,\ell_2} \subseteq W, \\ \vdots \\ W_{n,1} \oplus W_{n,2} \oplus \cdots \oplus W_{n,\ell_n} \subseteq W. \end{gathered}$$

Without loss of generality, we may make the following assumptions:

- Each direct sum $\bigoplus_{j=1}^{\ell_i} W_{i,j}$ is in fact equal to W. If not, add one more vector space to the sum.
- $\ell_1 = \ell_2 = \cdots = \ell_n =: \ell$. If not, pad the sum with several copies of $\{0\}$.

- $\ell \le r$. If not, then because the sums are supposed to be direct, each decomposition must contain at least $\ell - r$ copies of $\{0\}$, which can be dropped.

Lemma 2. *There are at most* $\dim W = r$ *different tuples*

$$\boldsymbol{j} = (j_1, \ldots, j_n) \in \{1, \ldots, \ell\}^n$$

such that $W_{\boldsymbol{j}} := W_{1,j_1} \cap W_{2,j_2} \cap \cdots \cap W_{n,j_n} \ne \{0\}$.

Proof. Induction on n. For $n = 1$, there are only $\ell \le r$ different tuples altogether: $(1), (2), \ldots, (\ell)$, so the claim is obviously true. Suppose now that the claim is shown for the case when $n - 1$ decompositions of some vector space are given. Let $U \subset \{1, \ldots, \ell\}^n$ be a set of tuples $\boldsymbol{j}$ with $W_{\boldsymbol{j}} \ne \{0\}$. Partition the elements of U according to their first components,

$$U = U_1 \,\dot{\cup}\, U_2 \,\dot{\cup} \cdots \dot{\cup}\, U_\ell,$$

i.e., U_k is the set of all tuples $\boldsymbol{j}$ whose first component is k, for $k = 1, \ldots, \ell$.

For all $\boldsymbol{j} = (k, j_2, \ldots, j_n) \in U_k$ we have $\{0\} \ne W_{\boldsymbol{j}} \subseteq W_{1,k}$. Therefore, $(j_2, \ldots, j_n) \in \{1, \ldots, \ell\}^{n-1}$ is a valid solution tuple for the modified problem with $W'_{i,j} := W_{i+1,j} \cap W_{1,k}$ ($i = 1, \ldots, n-1$, $j = 1, \ldots, \ell$) in place of $W_{i,j}$ ($i = 1, \ldots, n$, $j = 1, \ldots, \ell$). By induction hypothesis, since the $W'_{i,j}$ form $n - 1$ decompositions of the space $W_{1,k}$, there are at most $\dim W_{1,k}$ tuples $(j_2, \ldots, j_n)$ with $W_{(j_2,\ldots,j_m)} \ne \{0\}$. Consequently, there are altogether at most $\sum_{k=1}^{\ell} \dim W_{1,k} = \dim W = r$ different tuples for the original space W. ∎

The desired index tuples can be computed efficiently using dynamic programming, as shown in the following algorithm.

Algorithm 3. *Input: a vector space* W *of dimension* r*, and a collection of subspaces* $W_{i,j}$ $(i = 1, \ldots, n;\ j = 1, \ldots, \ell)$ *such that* $W = \bigoplus_{j=1}^{\ell} W_{i,j}$ *for* $i = 1, \ldots, n$ *and* $\ell \le r$.
Output: the set U *of all tuples* $\boldsymbol{j} = (j_1, \ldots, j_n)$ *with the property* $W_{\boldsymbol{j}} = \bigcap_{i=1}^{n} W_{i,j_i} \ne \{0\}$.

1. $U := \{\, (j) : W_{1,j} \ne \{0\} \,\}$
2. *For* $i = 2, \ldots, n$ *do*
3. $\quad U_{new} := \emptyset$
4. $\quad$ *For* $j = 1, \ldots, \ell$ *do*
5. $\quad\quad$ *For* $\boldsymbol{k} \in U$ *do*
6. $\quad\quad\quad$ *If* $W_{\boldsymbol{k}} \cap W_{i,j} \ne \{0\}$ *then*
7. $\quad\quad\quad\quad U_{new} := U_{new} \cup \{\mathrm{append}(\boldsymbol{k}, j)\}$
8. $\quad U := U_{new}$
9. *Return* U

Proposition 4. *Algorithm 3 is correct and needs no more than* $\mathrm{O}(nr^5)$ *operations in* C*, if the bases of the* $W_{\boldsymbol{k}}$ *are cached.*

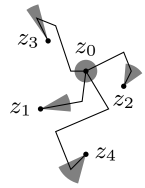
Proof. Correctness is obvious by line 6 and the fact that whenever $\boldsymbol{k} = (k_1, \ldots, k_n)$ is such that $W_{\boldsymbol{k}} \ne \{0\}$ then we necessarily also have $W_{(k_1,\ldots,k_{n-1})} \ne \{0\}$.

For the complexity, observe that all the vector space intersections can be performed with a number of operations which is at most cubic in r. Taking also into account that we always have $|U| \le r$ by Lemma 2, and that $\ell \le r$, the bound $\mathrm{O}(n\ell r^4) = \mathrm{O}(nr^5)$ follows immediately. ∎

By a more careful analysis it can be shown that the complexity is actually bounded by $8nr^4$. We omit the details of the corresponding argument as we will not need the tighter estimate in what follows.

5. NUMERICAL EVALUATION AT A REFERENCE POINT

We now turn to the question of how to construct the morphisms π_i. The basic idea is to choose a reference point z_0 that is an ordinary point of P, and let W be the space of analytic solutions of the equation in a neighborhood of z_0.



For each singular point z_i, let Δ_i be a sector rooted at z_i for which all formal power series appearing in the generalized series solutions of P at z_i admit an interpretation as analytic functions via some operator $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}$ (depending on i, but not on the series), as described in Section 2.4. Such sectors exist and can be computed explicitly. Next, let γ_i $(i = 1, \ldots, n)$ be polygonal paths from z_i to z_0 avoiding singular points and leaving the startpoint through Δ_i (meaning that for some $\varepsilon > 0$ all the points on γ_i with a distance to z_i less than ε should belong to Δ_i). Such paths exist. The analytic interpretations of the generalized series solutions at the singular points z_i defined in Δ_i admit a unique analytic continuation along the paths γ_i to the neighborhood of z_0.

We define $\pi_i \colon V_i \to W$ as follows. Let $V^0_{i,j}$ be the subspace of $V_{i,j}$ consisting of generalized series (1) with $s = 1$ and $m = 0$, and let $V'_{i,j}$ be a linear complement of $V^0_{i,j}$ in $V_{i,j}$. If $\hat{y} \in V^0_{i,j}$ i.e., if $\hat{y} = \mathrm{Exp}(e_{i,j})b$ with $e_{i,j} \in \mathbb{C}[\tilde{x}_i^{-1}]$ and $b \in \mathbb{C}[[\tilde{x}_i]]$, define $\pi_i(\hat{y})$ to be the unique analytic continuation of the function $\mathrm{E}(e_{i,j})\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(b)$ along γ_i to z_0, where $\mathrm{E}(e_{i,j})$ refers to the function $z \mapsto \exp(\int_{z_0}^{z} e_{i,j}/\tilde{x}_i)$ with some arbitrary but fixed choice of the branch of the logarithm, and $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}$ is as described in Section 2.4. Set $\pi_i(\hat{y}) = 0$ for $\hat{y} \in V'_{i,j}$, and then extend π_i to V_i by linearity. The precise values of $\pi_i(V_{i,j})$ depend on the choice of Δ_i and $\boldsymbol{d}$ (which is arbitrary, within the limits indicated in Section 2.4), but, as shown below, the properties of these spaces used in the algorithm do not.

Proposition 5. *The functions* π_i *defined above satisfy the two requirements imposed in Section 4: (1)* $\pi_i(V_{i,1}) + \cdots + \pi_i(V_{i,\ell_i})$ *is a direct sum; (2) if* h *is a hyperexponential term, then* $\pi_i^{-1}(h)$ *contains the formal series expansion of* h *at* z_i*, possibly up to a multiplicative constant.*

Proof. 1. Without loss of generality, we assume $z_i = 0$. Let $\hat{y}_j \in V_{i,j}$ $(j = 1, \ldots, \ell_i)$ and consider $\hat{y} = \sum_{j=1}^{\ell_i} \hat{y}_j$. Write $\hat{y}_j = x^{\alpha_j} \exp(u_j) b_j + \hat{y}'_j$ where $\hat{y}'_j \in V'_{i,j}$, the (α_j, u_j) are pairwise distinct, $u_j(0) = 0$, and $b_j(0) \ne 0$ unless the series b_j is zero. Writing $u_j = \sum_k u_{j,k} x^{-k}$, choose a direction θ such that $\rho \mathrm{e}^{\mathrm{i}\theta} \in \Delta_i$ for small ρ and any two unequal $u_{j,k}^{1/k} \mathrm{e}^{\mathrm{i}\theta}$ have different real parts.

By changing x to $\mathrm{e}^{-\mathrm{i}\theta} x$, we can assume that $d = 0$. This tranforms u_j into $\sum_k (u_{j,k} \mathrm{e}^{\mathrm{i}k\theta}) x^{-k}$, so that the real parts of two polynomials u_j can be the same only if the u_j themselves are equal. Hence, we can reorder the nonzero terms in the expression of $\hat{y}$ by asymptotic growth rate, in such a way that the nonzero terms come first, $u_1 = \cdots = u_t$ and $\operatorname{Re} \alpha_1 = \cdots = \operatorname{Re} \alpha_t$, while

$$z^{\operatorname{Re} \alpha_1} \mathrm{e}^{\operatorname{Re} u_1(z)} \gg z^{\operatorname{Re} \alpha_p} \mathrm{e}^{\operatorname{Re} u_p(z)}, \quad z \to 0, z > 0$$

for all $p \geq t+1$ such that $y_p \neq 0$. Using the definition of π_i and the fact that $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}(b_j)(z)$ tends to $b_j(0)$ as $z \to 0$ in the positive reals, it follows that

$$z^{-\operatorname{Re}\alpha_1} \exp(-u_1(z))\, \pi_i(\hat{y})(z) = \sum_{j=1}^{t} c_j b_j(0) z^{\mathrm{i}\operatorname{Im}\alpha_j} + \mathrm{o}(1)$$

(as $z \to 0$, $z > 0$) for some nonzero constants c_j. Since the (α_j, u_j) are pairwise distinct by assumption and the $(\operatorname{Re}\alpha_j, u_j)$ are equal for $j = 1, \ldots, t$, the $\operatorname{Im}\alpha_j$ are pairwise distinct for $j = 1, \ldots, t$.

Now assume that $\pi_i(\hat{y}) = 0$. Then, for all $\lambda > 0$, the expression $\sum_{j=1}^{t} c_j b_j(0)(\lambda z)^{\mathrm{i}\operatorname{Im}\alpha_j}$ tends to 0 as $z \to 0$, $z > 0$. Choosing $\lambda = \mathrm{e}^p$ for $p = 1, \ldots, t$, it follows that if not all the $b_j(0)$ were zero, the $t \times t$ determinant

$$\det\Big((\mathrm{e}^p z)^{\mathrm{i}\operatorname{Im}\alpha_q}\Big)_{p,q} = z^{\mathrm{i}\operatorname{Im}(\alpha_1+\cdots+\alpha_t)} \det\Big((\mathrm{e}^{\mathrm{i}\operatorname{Im}\alpha_q})^p\Big)_{p,q}$$

would tend to zero as well, which however is not the case. Therefore $b_j(0) = 0$ for $j = 1, \ldots, t$, and therefore $\hat{y}_j = 0$ for $j = 1, \ldots, t$, and therefore $\hat{y}_j = 0$ for $j = 1, \ldots, \ell_i$.

2. Let $h \in W$ be hyperexponential. Then the expansion $\hat{h}$ of h at z_i is clearly a local solution, so $\hat{h} \in V_{i,j}$ for some j. We show that $\pi_i(\hat{h}) = ch$ for some $c \in \mathbb{C}$. The map π_i is a differential homomorphism because $\mathcal{S}_{\boldsymbol{k},\boldsymbol{d}}$ is (as remarked in Section 2.4) and the (formal) exponential parts $\operatorname{Exp}(e_{i,j})$ are mapped to analytic functions satisfying the same differential equations. Since h is hyperexponential, it satisfies a first order linear differential equation. Since $\hat{h}$ is the expansion of h, it satisfies the same equations as h. Since π_i is a differential homomorphism, $\pi_i(\hat{h})$ satisfies the same equations as $\hat{h}$. Hence $\pi_i(\hat{h})$ and h satisfy the same first-order differential equation. The claim follows. ∎

The definition of the maps $\pi_1, \ldots, \pi_m$ as outlined above relies on analytic continuation, a concept which is only available if $C = \mathbb{C}$. For actual computations, we must work in a computable coefficient domain. At this point, we use numerical approximations. By van der Hoeven's result quoted in Section 2.4, we are able to compute for every given $\hat{y} \in V_{i,j}$ and every given $\varepsilon > 0$ a vector $Y_\varepsilon \in \mathbb{Q}(\mathrm{i})^r$ with

$$\Big\| \Big(D^k \pi_i(\hat{y})(z_0)\Big)_{k=0}^{r-1} - Y_\varepsilon \Big\|_\infty < \varepsilon.$$

Using these approximations, the linear algebra parts of Algorithm 3 are then performed with ball arithmetic to keep track of accumulating errors during the calculations. The test in line 6 of this algorithm requires to check whether a certain matrix has full rank. There are two possible outcomes: If during the Gaussian elimination we can find in every iteration an entry which is definitely different from zero, then the rank of the matrix is definitely maximal and the intersection of the vector spaces is definitely empty. We are then entitled to discard the possible extension of the partial tuple under consideration. On the other hand, if during the Gaussian elimination we encounter a column in which all the entries are balls that contain zero, this can either mean that the intersection is really nonempty, or that the accuracy of the approximation was insufficient. In this case, in order to be on the safe side, we must consider the intersection as nonempty and include the corresponding tuple.

Regardless of which initial accuracy ε is used, this variant of Algorithm 3 produces a set of tuples that is guaranteed to contain all correct ones, but may possibly contain additional ones. With sufficiently high precision, the number of tuples in the output that actually have an empty intersection will drop to zero. We don't need to know in advance which precision is sufficient in this sense, because it is not dramatic to have some extra tuples in the output as long as they are not too many. Unfortunately, we don't have an implementation yet with which we could experimentally determine a good strategy for resolving this situation. An ad-hoc way to balance precision and output size might be to start the algorithm with some fixed precision ε and let it abort and restart with doubled precision whenever $|U|$ exceeds $2r$, say. Another idea, suggested by a referee, is to use our algorithm in combination with the modular approach from [6] for cross checking tuples for which the numerical data is not conclusive.

In any case, we want to stress that we use numerical approximations only to determine the (small) tuple set U, and not to somehow reconstruct the (long) exact symbolic hyperexponential solutions from it. We therefore don't expect to need very high precision in typical situations.

6. A DETAILED EXAMPLE

Consider the operator

$$P = p_0 + p_1 D + p_2 D^2 + p_3 D^3 \in \mathbb{Q}[x][D]$$

where

$$\begin{gathered} p_0 = -105x^{20} + 3570x^{19} - 58026x^{18} + 556216x^{17} - 3456830x^{16} + \\ 14810744x^{15} - 45667732x^{14} + 104614932x^{13} - 182764261x^{12} + \\ 249940430x^{11} - 276371642x^{10} + 257839924x^9 - 211785148x^8 + \\ 154714472x^7 - 95675216x^6 + 45214304x^5 - 13863936x^4 + \\ 1685888x^3 + 424960x^2 - 182784x + 20480, \end{gathered}$$

$$\begin{gathered} p_1 = (x-1)x(105x^{19} - 3150x^{18} + 51456x^{17} - 489796x^{16} + \\ 2938210x^{15} - 11903624x^{14} + 34247824x^{13} - 72603516x^{12} + \\ 116974957x^{11} - 148046826x^{10} + 153582952x^9 - 137261696x^8 + \\ 109046080x^7 - 75250624x^6 + 41559168x^5 - 16084864x^4 + \\ 3278080x^3 + 163840x^2 - 231424x + 32768), \end{gathered}$$

$$\begin{gathered} p_2 = -4(x-2)^2(x-1)^3x^2(30x^{15} - 693x^{14} + 7314x^{13} - 42905x^{12} + \\ 155930x^{11} - 378483x^{10} + 649718x^9 - 828795x^8 + 820160x^7 - \\ 645092x^6 + 398200x^5 - 182384x^4 + 54656x^3 - 5696x^2 - 2944x + 1024), \end{gathered}$$

$$\begin{gathered} p_3 = 4(x-2)^4(x-1)^5x^4(15x^{10} - 258x^9 + 1492x^8 - 4446x^7 + \\ 8309x^6 - 10972x^5 + 10520x^4 - 6456x^3 + 1552x^2 + 480x - 256). \end{gathered}$$

The leading coefficient p_3 has 13 distinct roots in $\mathbb{C}$, but those coming from the degree-10-factor turn out to be apparent, so we can ignore them. (See [4] and the references there for more on apparent singularites.) It thus remains to study the singular points $z_1 := 0$, $z_2 := 1$, $z_3 := 2$, and $z_4 := \infty$.

For each singular point, we find three linearly independent generalized series solutions with two distinct exponential parts:

$$\begin{aligned} V_{1,1} &= \mathbb{C}\hat{y}_{1,1} & V_{1,2} &= \mathbb{C}\hat{y}_{1,2} + \mathbb{C}\hat{y}_{1,3}, \\ V_{2,1} &= \mathbb{C}\hat{y}_{2,1} & V_{2,2} &= \mathbb{C}\hat{y}_{2,2} + \mathbb{C}\hat{y}_{2,3}, \\ V_{3,1} &= \mathbb{C}\hat{y}_{3,1} & V_{3,2} &= \mathbb{C}\hat{y}_{3,2} + \mathbb{C}\hat{y}_{3,3}, \\ V_{4,1} &= \mathbb{C}\hat{y}_{4,1} & V_{4,2} &= \mathbb{C}\hat{y}_{4,2} + \mathbb{C}\hat{y}_{4,3} \end{aligned}$$

where

$$\hat{y}_{1,1} = \exp(\tfrac{1}{x})\Big(1 - \tfrac{4}{9}x + \tfrac{37}{32}x^2 + \tfrac{83}{384}x^3 + \cdots\Big),$$

$$\hat{y}_{1,2} = \sqrt{x}\Big(1 - x - \tfrac{25}{24}x^3 + \cdots\Big),$$

$$\hat{y}_{1,3} = \sqrt{x}\Big(x^2 - \tfrac{7}{4}x^3 + \tfrac{9}{32}x^4 + \cdots\Big),$$

$$\hat{y}_{2,1} = (x-1)^3 + (x-1)^5 - \tfrac{4}{3}(x-1)^6 + \cdots,$$
$$\hat{y}_{2,2} = \exp(\tfrac{1}{x-1})\Big(1 + \tfrac{1}{2}(x-1) + \tfrac{19}{120}(x-1)^3 + \cdots\Big),$$
$$\hat{y}_{2,3} = \exp(\tfrac{1}{x-1})\Big((x-1)^2 + \tfrac{23}{30}(x-1)^3 + \cdots\Big),$$
$$\hat{y}_{3,1} = 1 - \tfrac{3}{4}(x-2) + \tfrac{39}{32}(x-2)^2 - \tfrac{673}{384}(x-2)^3 + \cdots,$$
$$\hat{y}_{3,2} = \tfrac{1}{(x-2)^2}\exp(\tfrac{1}{x-2})\Big(1 + \tfrac{11}{4}(x-2) + \cdots\Big),$$
$$\hat{y}_{3,3} = \tfrac{1}{(x-2)^2}\exp(\tfrac{1}{x-2})\Big((x-2)^3 + \tfrac{1}{4}(x-2)^4 + \cdots\Big),$$
$$\hat{y}_{4,1} = x\big(1 + 3x^{-1} + 9x^{-2} + \tfrac{79}{3}x^{-3} + 74x^{-4} + \cdots\big),$$
$$\hat{y}_{4,2} = \sqrt{x}\Big(1 + x^{-1} + \tfrac{3}{2}x^{-2} + \tfrac{13}{6}x^{-3} + \cdots\Big),$$
$$\hat{y}_{4,2} = \sqrt{x}\Big(x^3 + x + \tfrac{19}{6}x^{-1} + \tfrac{283}{30}x^{-2} + \cdots\Big).$$

Let us choose $z_0 = 3$ as ordinary reference point and take the branch of the logarithm for which $\sqrt{x}$ is positive and real on the positive real axis. The example was chosen in such a way that all the power series are convergent in some neighborhood of the expansion point, so that we do not need to worry about sectors and resummation theory but can use the somewhat simpler algorithm for effective analytic continuation in the ordinary case to compute the values of the analytic functions $y_{i,j} := \pi_i(\hat{y}_{i,j})$ ($i = 1,\ldots,4$; $j = 1,2,3$). The vectors $\big(y_{i,j}(z_0), Dy_{i,j}(z_0), D^2y_{i,j}(z_0)\big)$ to five decimal digits of accuracy are as follows.

$$W_{1,1}=\left[\begin{pmatrix}-200.15\\322.46\\-1184.8\end{pmatrix}\right],\quad W_{1,2}=\left[\begin{pmatrix}-70.513\\-46.308\\-101.17\end{pmatrix},\begin{pmatrix}-156.55\\-91.322\\-205.47\end{pmatrix}\right],$$

$$W_{2,1}=\left[\begin{pmatrix}30.349\\-48.896\\179.66\end{pmatrix}\right],\quad W_{2,2}=\left[\begin{pmatrix}12.494\\5.2891\\13.066\end{pmatrix},\begin{pmatrix}77.105\\44.216\\99.931\end{pmatrix}\right],$$

$$W_{3,1}=\left[\begin{pmatrix}.74285\\-.061904\\.14960\end{pmatrix}\right],\ W_{3,2}=\left[\begin{pmatrix}15.580\\-31.307\\105.26\end{pmatrix},\begin{pmatrix}4.5433\\2.6503\\5.9631\end{pmatrix}\right],$$

$$W_{4,1}=\left[\begin{pmatrix}30.349\\-48.896\\179.66\end{pmatrix}\right],\quad W_{4,2}=\left[\begin{pmatrix}2.8557\\-.23797\\.57510\end{pmatrix},\begin{pmatrix}63.199\\41.308\\90.353\end{pmatrix}\right].$$

We now go through Algorithm 3. Start with the partial tuples (1) and (2) corresponding to the vector spaces $W_{1,1}$ and $W_{1,2}$, respectively. To compute the intersection of $W_{1,1}$ and $W_{2,1}$ we apply Gaussian elimination to the 3×2-matrix whose columns are the generators of $W_{1,1}$ and $W_{2,1}$:

$$\begin{pmatrix}-200.15 & 30.349\\ 322.46 & -48.896\\ -1184.8 & 179.66\end{pmatrix}\longrightarrow\begin{pmatrix}-200.15 & 30.349\\ & 0.00\\ & 0.00\end{pmatrix}$$

The notation 0.00 refers to some complex number z with $|z| < 5\cdot 10^{-3}$, which may or may not be zero, while the blank entries in the left column signify exact zeros that have been produced by the elimination. As the remaining submatrix does not contain any entry which is certainly nonzero, we regard the intersection as nonempty, which in this case means $W_{1,1} = W_{2,1}$. The partial tuple (1) is extended to $(1,1)$.

The intersections $W_{1,1}\cap W_{2,2}$ and $W_{1,2}\cap W_{2,1}$ turn out to be trivial, as they have to be if we really have $W_{1,1} = W_{1,2}$, because the sums $W_{1,1}\oplus W_{1,2}$ and $W_{2,1}\oplus W_{2,2}$ are direct. It thus remains to consider the intersection $W_{1,2}\cap W_{2,2}$. Applying Gaussian elimination to the 3×4-matrix whose columns are the generators of $W_{1,2}$ and $W_{2,2}$, we find

$$\begin{pmatrix}-70.513 & -25.596 & 12.494 & 77.105\\ -46.308 & 2.1330 & 5.2891 & 44.216\\ -101.17 & -5.1548 & 13.066 & 99.931\end{pmatrix}$$
$$\longrightarrow\begin{pmatrix}-70.513 & -25.596 & 12.494 & 77.105\\ & -17.50 & 4.440 & 9.777\\ & & 0.00 & 0.00\end{pmatrix},$$

which suggests that we have $W_{1,2} = W_{2,2}$. We extend the partial tuple (2) to $(2,2)$. At the end of the first iteration, we have $U = \{(1,1),(2,2)\}$.

In the second iteration, we find $W_{(1,1)}\cap W_{3,1} = \{0\}$ and $W_{(1,1)}\subseteq W_{3,2}$, so we extend the partial tuple $(1,1)$ to $(1,1,2)$ and record $W_{(1,1,2)} = W_{(1,1)} = W_{1,1}$. Furthermore we find $W_{3,1}\subseteq W_{(2,2)}$, so we extend $(2,2)$ to $(2,2,1)$ and record $W_{(2,2,1)} = W_{3,1}$. Finally, there is a nontrivial intersection between $W_{(2,2)}$ and $W_{3,2}$:

$$\begin{pmatrix}12.494 & 77.105 & 15.580 & 4.5433\\ 5.2891 & 44.216 & -31.307 & 2.6503\\ 13.066 & 99.931 & 105.26 & 5.9631\end{pmatrix}$$
$$\longrightarrow\begin{pmatrix}12.494 & 77.105 & 15.580 & 4.5433\\ & -27.34 & 89.53 & -1.72\\ & & 216. & 0.00\end{pmatrix}$$

suggests a common subspace of dimension 1 generated by the second listed generator of $W_{3,2}$. We therefore extend the partial tuple $(2,2)$ to $(2,2,2)$ and record $W_{(2,2,2)} = [(4.5433, 2.6503, 5.9631)]$. At the end of the second iteration, we have $U = \{(1,1,2),(2,2,1),(2,2,2)\}$.

For the final iteration, we see by inspection that $W_{4,1} = W_{2,1} = W_{(1,1,2)}$, so we extend $(1,1,2)$ to $(1,1,2,1)$. Because $\dim W_{4,1} = 1$ and the sums of the vector spaces are direct, the other two partial tuples cannot also have a nontrivial intersection with $W_{4,1}$, nor can $W_{(1,1,2)}\cap W_{4,2}$ be nontrivial. We do however have $W_{(2,2,1)}\subseteq W_{4,2}$ and $W_{(2,2,2)}\subseteq W_{4,2}$, so the algorithm terminates with the output $U = \{(1,1,2,1),(2,2,1,2),(2,2,2,2)\}$.

At this point we know that every hyperexponential solution of the operator P must have one of the following three exponential parts:

$$\frac{1}{(x-2)^2}\exp\Big(\frac{1}{x}+\frac{1}{x-2}\Big)\qquad\text{from (1,1,2,1)}$$
$$\sqrt{x}\exp\Big(\frac{1}{x-1}\Big)\qquad\text{from (2,2,1,2)}$$
$$\sqrt{x}\exp\Big(\frac{1}{x-1}+\frac{1}{x-2}\Big)\qquad\text{from (2,2,2,2).}$$

Following the steps of Algorithm 1, it remains to check whether some rational function multiples of these terms are solutions of P. The important point is that we have to do this only for three different candidates, while the naive algorithm would have to go through all $2^4 = 16$ combinations. Indeed, it turns out that P has the following three hyperexponential solutions:

$$\frac{(x-1)^3}{(x-2)^2}\exp\Big(\frac{1}{x}+\frac{1}{x-2}\Big),\quad \sqrt{x}\exp\Big(\frac{1}{x-1}\Big),$$

$$(x-2)x^2\sqrt{x}\exp\Big(\frac{1}{x-1}+\frac{1}{x-2}\Big).$$

7. CONCLUDING REMARKS

Our algorithm as described above takes advantage of the fact that series expansions of hyperexponential terms cannot involve exponential terms with ramification ($s > 1$) or logarithms ($m > 0$), by letting the morphisms π_i map all these irrelevant series solutions to zero. As a result, we get smaller vector spaces $W_{i,j}$, which not only reduces the expected computation time per vector space intersection but also makes it somehow more likely for intersections to be empty, thus decreasing the chances of getting tuples that do not correspond to hyperexponential solutions.

As a further refinement in this direction, it would be desirable to exploit the fact that if $\hat{h} = \mathrm{Exp}(e)b$ is the expansion of some hyperexponential term h, then the formal power series b must be convergent in some neighborhood of the expansion point. Instead of the vector spaces $W_{i,j}$ used above, it would be sufficient to consider the subspaces $W'_{i,j} \subseteq W_{i,j}$ corresponding to generalized series solutions involving only convergent power series. Besides the advantage of having to work with even smaller vector spaces, an additional advantage would be that the numerical evaluation becomes simpler because algorithms for the regular case [5, 9] become applicable. Implementations of these algorithms are available [8], which to our knowledge is not yet the case for van der Hoeven's general algorithm for the divergent case [11]. Unfortunately however, it is not obvious how to compute from a given basis of $W_{i,j}$ a basis of the subspace $W'_{i,j}$. Miller's algorithm [17] numerically solves a similar problem, but so far we have not been able to turn the underlying convergence statements into explicit error bounds that would yield an algorithm producing output with certified precision.

Finally, it would of course be also interesting to see an analog of our algorithm for finding hypergeometric solutions of linear recurrence equations with polynomial coefficients. A translation is not immediate because there is no notion of local solution around a finite singularity in this case.

8. REFERENCES

[1] S.A. Abramov and K.Yu. Kvashenko. Fast algorithms to search for the rational solutions of linear differential equations with polynomial coefficients. In *Proceedings of ISSAC'91*, pages 267–270, 1991.

[2] Werner Balser. *From Divergent Power Series to Analytic Functions*, volume 1582 of *Lecture Notes in Mathematics*. Springer-Verlag, 1994.

[3] Werner Balser. *Formal power series and linear systems of meromorphic ordinary differential equations*. Springer, 2000.

[4] Shaoshi Chen, Maximilian Jaroschek, Manuel Kauers, and Michael F. Singer. Desingularization explains order-degree curves for univariate Ore operators. In *these ISSAC'13 proceedings*, 2013.

[5] David V. Chudnovsky and Gregory V. Chudnovsky. Computer algebra in the service of mathematical physics and number theory. In David V. Chudnovsky and Richard D. Jenks, editors, *Computers in Mathematics*, volume 125 of *Lecture Notes in Pure and Applied Mathematics*, pages 109–232, Stanford University, 1986. Dekker.

[6] Thomas Cluzeau and Mark van Hoeij. A modular algorithm to compute the exponential solutions of a linear differential operator. *Journal of Symbolic Computation*, 38:1043–1076, 2004.

[7] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Annals of Mathematics*, 126:515–534, 1982.

[8] Marc Mezzarobba. NumGfun: a package for numerical and analytic computation with d-finite functions. In *Proceedings of ISSAC'10*, 2010.

[9] Joris van der Hoeven. Fast evaluation of holonomic functions. *Theoretical Computer Science*, 210(1):199–216, 1999.

[10] Joris van der Hoeven. Around the numeric-symbolic computation of differential galois groups. *Journal of Symbolic Computation*, 42:236–264, 2007.

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

[12] Marius van der Put and Michael Singer. *Galois Theory of Linear Differential Equations*. Springer, 2003.

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

[14] Mark van Hoeij. Formal solutions and factorization of differential operators with power series coefficients. *Journal of Symbolic Computation*, 24(1):1–30, 1997.

[15] Mark van Hoeij. Factoring polynomials and the knapsack problem. *Journal of Number Theory*, 95:167–189, 2002.

[16] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[17] Jet Wimp. *Computing with Recurrence Relations*. Pitman Publishing Ltd., 1984.