Computing Hypergeometric Solutions of Second Order Linear Differential Equations using Quotients of Formal Solutions and Integral Bases

Erdal Imamoglu [1]

Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA.

Mark van Hoeij [1]

Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA.

Abstract

We present two algorithms for computing hypergeometric solutions of second order linear differential operators with rational function coefficients. Our first algorithm searches for solutions of the form

$$\exp(\int r \, dx) \cdot {}_2F_1(a_1, a_2; b_1; f) \tag{1}$$

where $r, f \in \overline{\mathbb{Q}(x)}$, and $a_1, a_2, b_1 \in \mathbb{Q}$. It uses modular reduction and Hensel lifting. Our second algorithm tries to find solutions in the form

$$\exp(\int r \, dx) \cdot \left(r_0 \cdot {}_2F_1(a_1, a_2; b_1; f) + r_1 \cdot {}_2F_1{}'(a_1, a_2; b_1; f) \right) \tag{2}$$

where $r_0, r_1 \in \overline{\mathbb{Q}(x)}$, as follows: It tries to transform the input equation to another equation with solutions of type (1), and then uses the first algorithm.

Key words: Symbolic Computation, Linear Differential Equations, Closed Form Solutions, Hypergeometric Solutions, Integral Bases.

1. Introduction

A second order homogeneous linear differential equation with rational function coefficients $A_i \in \mathbb{Q}(x)$

$$A_2 y'' + A_1 y' + A_0 y = 0 \tag{3}$$

corresponds to the differential operator

$$L = A_2 \partial^2 + A_1 \partial + A_0 \in \mathbb{Q}(x)[\partial]$$

where $\partial = \frac{d}{dx}$. Another representation of (3) is

$$L(y) = 0.$$

This paper gives two heuristic (see Remarks 1.2 and 3.4) algorithms to find a hypergeometric solution of (3) in the form (1) and (2). The form (2) is more general than in prior works Fang and van Hoeij (2011); Kunwar and van Hoeij (2013); van Hoeij and Vidunas (2015); Kunwar (2014). Papers van Hoeij and Vidunas (2015) and Kunwar (2014) were restricted to a specific number of singularities (4 in van Hoeij and Vidunas (2015) and 5 in Kunwar (2014)). Papers Kunwar and van Hoeij (2013) and Fang and van Hoeij (2011) were restricted to specific degrees (degree 3 in Kunwar and van Hoeij (2013) and a degree-2 decomposition in Fang and van Hoeij (2011)). Our algorithms are not restricted to a specific number of singularities or a specific degree. Moreover, our algorithms can find algebraic functions f in (1) and (2).

Our first algorithm, Algorithm 3.1, tries to find solutions of (3) in the form of (1). Our second algorithm, Algorithm 4.1, tries to reduce equations with solutions of form (2) to equations and then calls our first algorithm.

We assume that (3) has no Liouvillian solutions and hence irreducible. Otherwise one can solve (3) with Kovacic's algorithm Kovacic (1986).

Let $L_{inp} \in \mathbb{Q}(x)[\partial]$ (L input) be a second order linear differential operator, regular singular (details in Section 2.1) and without Liouvillian solutions:

- Goal 1: (Algorithm 3.1) Find a solution of form (1) if it exists.
- Goal 2: (Algorithm 4.1) Try to transform L_{inp} to a simpler operator (which hopefully has a solution in form (1)).

The crucial steps for Goal 1 are to find (candidates for) a_1, a_2, b_1 and the *pullback function* f. Finding the parameters a_1, a_2, b_1 is the combinatorial part; Theorem 3.2 helps us to eliminate the vast majority of cases. Given a_1, a_2, b_1 (or equivalently a base operator L_B), *if we know the value of a certain constant c*, by comparing *quotients of formal solutions* of L_B and L_{inp}, we can compute f. We have no direct formula for c; to obtain it with a finite computation, we take a prime number ℓ. Then, for each $c \in \{1, \ldots, \ell - 1\}$ we try to compute f modulo ℓ. If this succeeds, then we lift f modulo a power of ℓ, and try rational number reconstruction.

Goal 2 is to find a transformation to convert L_{inp} to a simpler operator $\tilde{L}_{inp}$. The key idea is to follow the strategy of the POLRED algorithm in Cohen and Diaz Y Diaz (1991). It takes as input a polynomial $L_{inp} \in \mathbb{Q}[x]$ and finds an element $\mathcal{G} \in \mathbb{Q}[x]/\mathbb{Q}[x]L_{inp}$ whose minimal polynomial is close to optimal. It works as follows:

(1) "Finite points": Compute an integral basis.
(2) "Valuations at infinity": Find an integral element $\mathcal{G}$ with (near) optimal absolute values.

Our key idea is to apply `POLRED`'s strategy to $L_{inp} \in \mathbb{Q}(x)[\partial]$. First compute an integral basis (introduced in Kauers and Koutschan (2015)). Then *normalize at infinity* (following (Trager, 1984, Section 2.3) where this is done for function fields) so we can select an element with minimal valuations at infinity.

Example 1.1 (Rational Pullback Function). The differential operator

$$L_{inp} = 147x(x-1)(x+1)\partial^2 + (266x^2 - 42x - 98)\partial + 20x - 5$$

has a $_2F_1$-type solution in the form of (1), which is

$$Y(x) = \exp\left(\int r\, dx\right) \cdot {}_2F_1\left(\frac{5}{42}, \frac{11}{42}; \frac{2}{3}; f\right)$$

where

$$\exp\left(\int r\, dx\right) = (x+1)^{-\frac{5}{21}} \quad \text{and} \quad f = \frac{4x}{(x+1)^2} \tag{4}$$

Section 3.4 shows how to find the parameters $a_1, a_2, b_1 = \frac{5}{42}, \frac{11}{42}, \frac{2}{3}$. Then f is computed with the quotient method:

Remark 1.1 (The Quotient Method). The hypergeometric function

$$_2F_1\left(\frac{5}{42}, \frac{11}{42}; \frac{2}{3}; x\right)$$

is a solution of the Gauss hypergeometric differential operator

$$L_B = \partial^2 + \frac{(29x - 14)}{21x(x-1)}\partial + \frac{55}{1764x(x-1)}.$$

L_B has two solutions at $x = 0$. They are

$$y_1(x) = {}_2F_1\left(\frac{5}{41}, \frac{11}{42}; \frac{2}{3}, x\right) = 1 + \frac{55}{1176}x + \ldots,$$

$$y_2(x) = x^{\frac{1}{3}}\left(1 + \frac{475}{2352}x + \frac{1941325}{19361664}x^2 + \ldots\right).$$

The so-called *exponents* of L_B at $x = 0$ are the exponents of x in the dominant terms of the solutions y_1 and y_2, so the exponents are $e_{0,1} = 0$ and $e_{0,2} = \frac{1}{3}$. The minimal operator for $y(f)$ has the following solutions at $x = 0$:

$$y_1(f) = 1 + \frac{55}{294}x - \frac{4939}{86436}x^2 + \frac{16135823}{304946208}x^3 + \ldots,$$

$$y_2(f) = c \cdot x^{\frac{1}{3}}\left(1 + \frac{83}{588}x + \frac{6805}{1210104}x^2 + \ldots\right)$$

for some constant c that depends on f. The exponents are again 0 and $\frac{1}{3}$, because $x = 0$ is a root of f with multiplicity $e_0 = 1$ (Theorem 2.1). Let

$$Y_1(x) = \exp\left(\int r\, dx\right) y_1(f) = 1 - \frac{5}{98}x + \frac{439}{9604}x^2 + \ldots, \tag{5}$$

$$Y_2(x) = \exp\left(\int r\, dx\right) y_2(f) = c \cdot x^{\frac{1}{3}}\left(1 - \frac{19}{196}x + \ldots\right). \tag{6}$$

(5) and (6) form a basis of solutions of L_{inp}. Here $\exp(\int r\,dx)$ is the same as in (4). Denote the quotients of the formal solutions of L_B and L_{inp} by

$$q = \frac{y_1(x)}{y_2(x)}$$

$$Q = \frac{Y_1(x)}{Y_2(x)} = \frac{y_1(f)}{y_2(f)} = q(f)$$

respectively. It follows that $q^{-1}(Q(x))$ gives a series expansion of f at $x = 0$. Given enough terms we can compute f with rational function reconstruction. This *Quotient Method* was already used in (van Hoeij and Vidunas, 2015, Section 5.1). In order to turn this into an algorithm for solving differential equations we need to answer the following questions:

Q1. How many terms are needed to reconstruct f? This is equivalent to finding a degree bound for f.

Q2. How to find the parameters a_1, a_2, b_1? This is the combinatorial part of our algorithm.

Q3. The exponents 0, $\frac{1}{3}$ of L_{inp} at $x = 0$ only determine $\frac{Y_1}{Y_2}$ up to a constant factor (see Remark 2.2 in Section 2.3). This means the quotient $\frac{y_1(f)}{y_2(f)}$ is only known up to a constant c. How to find this constant?

Q4. What if L_{inp} has logarithmic solutions at $x = 0$?

Q5. What if f is an algebraic function?

Q6. What if L_{inp} does not have solutions in the form of (1), but has solutions in the form of (2)?

Remark 3.2, Section 3.4, Section 3.6, and Section 3.5.2 provide answers to Q1, Q2, Q3, and Q4 respectively. Section 3.6 answers Q5. The parts Q1,...,Q5 are already in our ISSAC 2015 paper Imamoglu and van Hoeij (2015). Algorithm 4.1, which finds solutions of form (2), is new compared to Imamoglu and van Hoeij (2015). So Q6 is the main new part in this paper. It will be discussed in Section 4. Example 1.3 will illustrate to Q6.

Remark 1.2. Both algorithms are very effective in practice but they are not proven. For completeness for Goal 1 we still need a theorem for *good prime numbers*. A *good prime* is a prime for which reconstruction will work.

Example 1.2 (Algebraic Pullback Function)**.** The differential operator

$$L_{inp} = \partial^2 + \frac{1}{4}\frac{x^4 - 44\,x^3 + 1206\,x^2 - 44\,x + 1}{\left(x^2 - 34\,x + 1\right)^2 x^2}$$

has a $_2F_1$-type solution in the form of (1), which is

$$Y(x) = \exp(-\frac{1}{2}\int r\,dx) \cdot {}_2F_1\left(\frac{1}{3},\frac{2}{3};1;f\right)$$

where $r =$

$$\frac{-x^5 + 22\,x^4 - 55\,x^3 - 343\,x^2 + 58\,x - 1 + 6\,x\left(x^2 - 7\,x + 1\right)\sqrt{x^2 - 34\,x + 1}}{x\left(x^4 - 41\,x^3 + 240\,x^2 - 41\,x + 1\right)(x + 1)}$$

and

$$f = \frac{1}{2}\frac{1 + 30\,x - 24\,x^2 + x^3 - \left(x^2 - 7\,x + 1\right)\sqrt{x^2 - 34\,x + 1}}{1 + 3\,x + 3\,x^2 + x^3}.$$

Here the pullback function f is an algebraic function: $\mathbb{Q}(x, f)$ is an algebraic extension of $\mathbb{Q}(x)$ of degree $a_f = 2$ (a_f is 1 if and only if f is a rational function, as in Example 1.1). Algorithm 3.1 can find this solution.

Example 1.3 (Finding Solutions in the form of (2) using an Integral Basis). Consider the differential operator[2]

$$L_{inp} = \partial^2 - \frac{512\,x^5 + 384\,x^4 - 64\,x^3 - 88\,x^2 - 10\,x - 1}{x\,(4\,x - 1)\,(4\,x + 1)\,(16\,x^3 + 24\,x^2 + 5\,x + 1)}\partial$$
$$+ \frac{512\,x^5 + 64\,x^4 - 128\,x^3 - 60\,x^2 - 8\,x - 1}{x^2\,(4\,x - 1)\,(4\,x + 1)\,(16\,x^3 + 24\,x^2 + 5\,x + 1)}.$$

Algorithm 3.1 can not solve L_{inp}. We try to transform L_{inp} to simpler operator $\tilde{L}_{inp}$. First we compute an integral basis. Then we normalize the basis at infinity and obtain $[B_0, B_1]$ where

$$B_0 = \frac{16\,x^4 - x^2}{(16\,x^3 + 24\,x^2 + 5\,x + 1)\,x}\partial$$
$$+ \frac{-34359738400\,x^3 - 51539607556\,x^2 - 10737418241\,x - 2147483648}{(16\,x^3 + 24\,x^2 + 5\,x + 1)\,x}$$

and

$$B_1 = \frac{16\,x^3 - x}{(16\,x^3 + 24\,x^2 + 5\,x + 1)\,x}\partial + \frac{-32\,x^2 - 4\,x - 1}{(16\,x^3 + 24\,x^2 + 5\,x + 1)\,x}.$$

We try to find a suitable $\mathcal{G} \in \mathbb{Q}(x)[\partial]/\mathbb{Q}(x)[\partial]L_{inp}$. It should be a combination of B_0 and B_1. For this example, we take $\mathcal{G} = B_1$. This $\mathcal{G}$ is called a *gauge* transformation. It maps solutions of L_{inp} to solutions of

$$\tilde{L}_{inp} = \partial^2 + \frac{48\,x^2 - 1}{x\,(16\,x^2 - 1)}\partial + \frac{16}{16\,x^2 - 1}.$$

$\tilde{L}_{inp}$ has a solution in the form of (1),

$$y(x) = {}_2F_1\left(\frac{1}{2}, \frac{1}{2}; 1; 16x^2\right)$$

which is easy to find with Algorithm 3.1. Then we apply the inverse *gauge* transformation and obtain a solution of L_{inp} in the form of (2), which is

$$Y(x) = \left(4x^3 + x^2 + \frac{x}{2}\right){}_2F_1\left(\frac{1}{2}, \frac{1}{2}; 1; 16x^2\right) + \left(32x^5 - 2x^3\right){}_2F_1\left(\frac{3}{2}, \frac{3}{2}; 2; 16x^2\right).$$

2. Preliminaries

This section recalls the concepts needed in later sections.

---

[2] Prof. Jean-Marie Maillard sent us this differential operator.

2.1. Differential Operators, Singularities, Formal Solutions

We start with some classical definitions which can be also found in Ince (1926); van der Put and Singer (2003); Kunwar (2014); Fang (2012); Debeerst (2007); Yuan (2012).

Definition 2.1. Let $L = \sum_{i=0}^{n} A_i \partial^i \in \mathbb{C}(x)[\partial]$ be an operator of order n.
  (i) A point $p \in \mathbb{C}$ is called a *singularity* of L if it is a zero of the leading coefficient of L or a pole of any other coefficient of L. The point $p = \infty$ is called a singularity if $p = 0$ is a singularity of $L_{1/x}$. Here $L_{1/x}$ is the differential operator obtained from L via a change of variables $x \mapsto \frac{1}{x}$ (note that $x \mapsto f$ sends ∂ to $\frac{1}{f'}\partial$).
  (ii) If $x = p$ is not a singularity, then it is called a *regular point* of L.
  (iii) A singularity $p \in \mathbb{C}$ is called a *regular singularity* if $(x - p)^i \frac{A_{n-i}}{A_n}$ is analytic at $x = p$ for $1 \leq i \leq n - 1$. The point $p = \infty$ is a regular singularity if $p = 0$ is a regular singularity of $L_{1/x}$.
  (iv) L is *regular singular* if all its singularities are regular singular.

2.2. Gauss Hypergeometric Differential Operator and $_2F_1$ Function

Definitions also can be found in Wang and Guo (1989); Kunwar (2014); Fang (2012). Let a_1, a_2, $b_1 \in \mathbb{Q}$. The operator

$$L_B = x(1-x)\partial^2 + (b_1 - (a_1 + a_2 + 1)x)\partial - a_1 a_2$$

is called *Gauss hypergeometric differential operator* (GHDO). The solution space of L_B in a *universal extension* Fang (2012) has dimension 2 because the order of L_B is 2. One of the solutions of L_B at $x = 0$ is the *Gauss hypergeometric function*. It is denoted by $_2F_1$ and defined by the Gauss hypergeometric series

$$_2F_1(a_1, a_2; b_1; x) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k}{(b_1)_k k!} x^k.$$

Here $(\lambda)_k$ denotes the *Pochammer symbol*. It is defined as $(\lambda)_k = \lambda(\lambda + 1) \ldots (\lambda + k - 1)$ and $(\lambda)_0 = 1$.

L_B has three regular singularities at the points $x = 0$, $x = 1$, and $x = \infty$, with exponents $\{0, 1 - b_1\}$, $\{0, b_1 - a_1 - a_2\}$, and $\{a_1, a_2\}$ respectively. We denote the exponent differences of a GHDO as $\alpha_0 = |1 - b_1|$, $\alpha_1 = |b_1 - a_1 - a_2|$, $\alpha_\infty = |a_1 - a_2|$. We may assume $\{a_1, a_2, b_1 - a_1, b_1 - a_2\} \cap \mathbb{Z} = \emptyset$, otherwise L_B is reducible (it has exponential solutions).

Let d_i be ∞ if $\alpha_i \in \mathbb{Z}$, and the denominator of α_i if $\alpha_i \in \mathbb{Q} - \mathbb{Z}$. We will only consider a_1, a_2, b_1 for which L_B has no Liouvillian solutions. From the Schwarz list Schwarz (1873) one finds that this is equivalent to $\frac{1}{d_0} + \frac{1}{d_1} + \frac{1}{d_\infty} < 1$.

2.3. Transformations and Singularities

We summarize properties of transformations in this section. These properties can also be found in Kunwar (2014); Fang (2012); Debeerst (2007); Yuan (2012).

Let $L_1 \in \mathbb{C}(x)[\partial]$ be a differential operator of order 2, and let y be a solution of L_1. We consider the following *transformations* that send solutions of L_1 to solutions of another second order differential operator L_2.

6

(1) *Change of variables*:
$y(x) \longrightarrow y(f)$, where $f \in \overline{\mathbb{Q}(x)}$.
For L_1 this means substituting $(x, \partial) \mapsto (f, \frac{1}{f'}\partial)$.

Notation: $L_1 \xrightarrow{f}_C L_2$.
(2) *Exp-product*:
$y(x) \longrightarrow \exp\left(\int r\,dx\right)y(x)$, where $r \in \overline{\mathbb{Q}(x)}$.
For L_1 this means $\partial \mapsto \partial - r$.
Notation: $L_1 \xrightarrow{r}_E L_2$.
(3) *Gauge transformation*:
$y(x) \longrightarrow r_0 \cdot y(x) + r_1 \cdot y(x)$, where $r_0, r_1 \in \mathbb{Q}(x)$.
For L_1 this means computing the least common left multiple of L_1 and $r_1\partial + r_0$, and right-dividing it by $r_1\partial + r_0$.
Notation: $L_1 \xrightarrow{r_0, r_1}_G L_2$.

Remark 2.1. Transformations can affect singularities and exponents.
  (i) If a transformation $\xrightarrow{r}_E$ can send a singular point $x = p$ to a regular point $x = p$, then we call $x = p$ a *false singularity*.
 (ii) A singularity $x = p$ is a false singularity Debeerst (2007) if and only if $x = p$ is not logarithmic and the exponent difference is 1.
(iii) If $x = p$ is a singularity of L_1 and if transformation $\xrightarrow{r}_E$ can send L_1 to an equation L_2 for which all solutions of L_2 are analytic at $x = p$, then we call $x = p$ a *removable singularity*.
 (iv) A point $x = p$ is removable Debeerst (2007) if and only if $x = p$ is not logarithmic and the exponent difference is an integer. Non-removable singularities are called *true singularities*.
  (v) A point $x = p$ is a true singularity if and only if the exponent difference is not an integer *or* $x = p$ is logarithmic.
 (vi) If $L_1 \xrightarrow{r_0, r_1}_G L_2$, then L_1 and L_2 are called *gauge equivalent*. If V_1 and V_2 are the solution spaces of L_1 and L_2 respectively, then $\mathcal{G} = r_1\partial + r_0$ maps V_1 to V_2, i.e., $\mathcal{G}(V_1) = V_2$.

Remark 2.2. The quotient method (Remark 1.1 in Section 1) can only use true singularities, otherwise, $\frac{Y_1}{Y_2}$, the quotients of solutions of L_{inp}, would only be known up to a Möbius transformation instead of up to a constant.

Remark 2.3. At the moment, Algorithms 3.1 and 4.1 are only implemented for rational function coefficients. However, if f, r, r_0, r_1 are algebraic, then the three transformations may turn an operator with rational function coefficients into an operator with algebraic function coefficients.

Theorem 2.1. Bostan et al. (2011) Let the GHDO L_B have exponent differences α_0 at $x = 0$, α_1 at $x = 1$, and α_∞ at $x = \infty$. Let $L_B \xrightarrow{f}_C L_{inp}$. If $f(p) \in \{0, 1, \infty\}$, then L_{inp} has the following exponent difference at $x = p$:
  - $\alpha_0 \cdot e_p$ if f has a zero at $x = p$ with multiplicity e_p,
  - $\alpha_1 \cdot e_p$ if $f - 1$ has a zero at $x = p$ with multiplicity e_p,
  - $\alpha_\infty \cdot e_p$ if f has a pole at $x = p$ with order e_p.

3. Computing Solutions of a Second Order Linear Differential Operator in the form of (1) by using Quotients of Formal Solutions

This section gives our first algorithm, which looks for solutions of the form of (1).

3.1. Problem Statement

Given a second order linear differential operator $L_{inp} \in \mathbb{Q}(x)[\partial]$, irreducible and regular singular, we want to find a $_2F_1$-type solution of the differential equation $L_{inp}(y) = 0$ of the form of (1). This is equivalent to finding transformations 1 and 2 from a GHDO L_B to L_{inp}. Therefore, we need to find
 (1) L_B (i.e., find a_1, a_2, b_1),
 (2) parameters f and r of the change of variables and exp-product transformations such that $L_B \xrightarrow{f}_C \xrightarrow{r}_E L_{inp}$.

Algorithm 3.1. General Outline of `find_2f1`.
 INPUT: $L_{inp} \in \mathbb{Q}(x)[\partial]$ and (optional) $a_f max$ where
 L_{inp} = a second order regular singular irreducible operator,
 $a_f max$ = bound for the algebraic degree a_f (See Example 1.2). If omitted,
 then $a_f max = 2$ which means our implementation tries $a_f = 1$ and $a_f = 2$.
 OUTPUT: Solutions of L_{inp} in the form of (1), or an empty list.

 For each $a_f \in \{1, \ldots, a_f max\}$:
 (1) Use Section 3.4 to compute candidates for L_B and d_f. This is the combinatorial part of the algorithm.
 (2) For a candidate (L_B, d_f), compute formal solutions of L_B and L_{inp} at a non-removable singularity (see Remark 2.2 in Section 2.3) up to precision $a \geq 2(a_f + 1)(d_f + 1) + 6$. Take the quotients of formal solutions and compute series expansions for q^{-1} and Q which will be used to compute

$$f = q^{-1}(c\, Q(x)) \qquad (7)$$

 in the next step.
 (3) Choose a good prime number ℓ and try to find c mod ℓ by looping $c = 1, 2, \ldots, \ell - 1$ as in Section 3.5. For each c:
   3.1. Compute f mod (x^a, ℓ) from equation (7) and use it to reconstruct f mod ℓ (the image of f in $\mathbb{F}_\ell(x)$). If it fails for every c, then proceed with the next candidate GHDO (if any) in Step 2. If no candidates remain, then return an empty list.
   3.2. If rational reconstruction in Step 3.1 succeeds for some c values, then apply Hensel lifting (Section 3.6) to find f mod a power of ℓ. Then try rational number reconstruction. If it does not fail for at least one c value, then we have f. If no solution is found (see Remark 3.4 in Section 3.6.1), then proceed with the next candidate GHDO (if any) in Step 2. If no candidates remain, then return an empty list.
   3.3. Use Section 2.3 to compute the parameter r of the exp-product transformation.
 (4) Return a basis of $_2F_1$-type solutions of L_{inp}.

Step 2 is explained in Sections 3.2 and 3.4. Step 3 is the quotient method, see Section 3.5 for more. Steps 3.2 and 3.3 are explained in Sections 3.6 and 3.7 respectively. A Maple implementation of Algorithm 3.1 and some examples can be found at Imamoglu (2015a).

Theorem 3.1 (Riemann-Hurwitz Formula)**.** Let X and Y be two algebraic curves with genera g_X and g_Y respectively. If $f : X \longrightarrow Y$ is a non-constant morphism, then

$$2g_X - 2 = \deg(f)(2g_Y - 2) + \sum_{p \in X}(e_p - 1). \tag{8}$$

Here e_p denotes the ramification order at $p \in X$. See Hartshorne (1977) for more details.

Let $L_B \in \mathbb{Q}(x)[\partial]$ be a GHDO, $d_f := \deg(f)$, and assume that

$$L_B \xrightarrow{f:\mathbb{P}^1 \longmapsto \mathbb{P}^1}_C \xrightarrow{r}_E L_{inp}.$$

Section 3.1 of Imamoglu and van Hoeij (2015) gives an a priori bound for d_f,

$$d_f \leq \begin{cases} 6(n_{true} - 2), & \text{logarithmic case,} \\ 36\left(n_{true} - \frac{7}{3}\right), & \text{non-logarithmic case.} \end{cases} \tag{9}$$

where n_{true} is the number of true singularities of L_{inp}. Algorithm 3.1 uses this only as an initial degree bound.

3.3.   *Riemann-Hurwitz Type Formula For Differential Equations*

Remark 3.1. Let X be any algebraic curve and $\mathbb{C}(X)$ be its function field. The ring $D_{\mathbb{C}(X)} := \mathbb{C}(X)[\partial_t]$ is the ring of differential operators on X. Here $t \in \mathbb{C}(X) \setminus \mathbb{C}$. An element $L \in D_{\mathbb{C}(X)}$ is a differential operator defined on the algebraic curve X.

Theorem 3.2. (Baldassari and Dwork, 1979, Lemma 1.5) Let X, Y be two algebraic curves with genera g_X, g_Y, and function fields $\mathbb{C}(X)$, $\mathbb{C}(Y)$ respectively. Let $f : X \longrightarrow Y$ be a non-constant morphism. The morphism f corresponds to a homomorphism $\mathbb{C}(Y) \longrightarrow \mathbb{C}(X)$, which in turn corresponds to a homomorphism $D_{\mathbb{C}(Y)} \longrightarrow D_{\mathbb{C}(X)}$. If $L_1 \in D_{\mathbb{C}(Y)}$ with $\text{ord}(L_1) = 2$ and L_2 is the corresponding element in $D_{\mathbb{C}(X)}$, then

$$\text{Covol}(L_2, X) = \deg(f) \cdot \text{Covol}(L_1, Y) \tag{10}$$

where

$$\text{Covol}(L, X) := 2g_X - 2 + \sum_{p \in X}(1 - \Delta(L, p))$$

and where $\Delta(L, p)$ is the absolute value of the exponent difference of L at p.

Proof. Following Baldassari and Dwork (1979), take finite sets $S \subseteq Y$ and $T = f^{-1}(S)$ in such a way that all singularities of L_1 are in S, all singularities of L_2 are in T, and all branching points in X are in T as well.

$$\#T = \sum_{p \in T} 1 = \sum_{p \in T} e_p + \sum_{p \in T}(1 - e_p) \tag{11}$$

$$= \deg(f) \cdot \#S + \sum_{p \in X}(1 - e_p) \tag{12}$$

$$= \deg(f) \cdot \#S - (2g_X - 2 - \deg(f)(2g_Y - 2)). \tag{13}$$

From (12) to (13) we used (8). Then,

$$\sum_{p\in X}(1-\Delta(L_2,p)) = \sum_{p\in T}(1-\Delta(L_2,p)) \tag{14}$$

$$= \sum_{p\in T}1 \ - \ \sum_{p\in T}\Delta(L_2,p) \tag{15}$$

$$= \#T \ - \ \deg(f)\sum_{s\in S}\Delta(L_1,s). \tag{16}$$

Then, combine (13) and (16) and get

$$2g_X - 2 + \sum_{p\in X}(1-\Delta(L_2,p)) = \deg(f)\left(2g_Y - 2 + \sum_{s\in Y}(1-\Delta(L_1,s))\right) \tag{17}$$

which is the same as (10). □

Corollary 3.1. Let $X = Y = \mathbb{P}^1$ and suppose that $L_B \xrightarrow{f:\mathbb{P}^1\to\mathbb{P}^1}_C \xrightarrow{r}_E L_{inp}$ where $L_B \in \mathbb{C}(x)[\partial]$ is a GHDO with exponent differences $[\alpha_0,\alpha_1,\alpha_\infty]$ at $\{0,1,\infty\}$. Since an exp-product transformation does not affect exponent differences, Theorem 3.2 gives the following equation for $\mathrm{Covol}(L_{inp},\mathbb{P}^1)$:

$$-2 + \sum_{p\in\mathbb{P}^1}(1-\Delta(L_{inp},p)) = \deg(f)\left(-2 + \sum_{i\in\{0,1,\infty\}}(1-\alpha_i)\right). \tag{18}$$

Corollary 3.2. Let L_B and L_{inp} be as in Corollary 3.1. Both have rational function coefficients. This time, suppose that f,r in $L_B \xrightarrow{f}_C \xrightarrow{r}_E L_{inp}$ are algebraic functions. Then $f : X \to \mathbb{P}^1$ for an algebraic curve X whose function field $\mathbb{C}(X) = \mathbb{C}(x,f)$ is an algebraic extension of both $\mathbb{C}(x) \cong \mathbb{C}(\mathbb{P}^1)$ and $\mathbb{C}(f) \cong \mathbb{C}(\mathbb{P}^1)$. Let a_f and d_f denote the degrees of these extensions.

$$
\begin{array}{ccc}
 & \mathbb{C}(x,f) & \\
a_f \diagup & & \diagdown d_f \\
\mathbb{C}(x) & & \mathbb{C}(f)
\end{array}
$$

Applying (10) to both field extensions gives:

$$\mathrm{Covol}(L_{inp},\mathbb{P}^1) = \frac{d_f}{a_f}\left(-2 + \sum_{i\in\{0,1,\infty\}}(1-\alpha_i)\right). \tag{19}$$

3.4.   Candidate Exponent Differences

This section explains how to obtain exponent differences for candidate GHDOs.

Algorithm 3.2. General Outline of `find_expdiffs`.

INPUT: e_{inp}, e_{app}, and a_f where

e_{inp} = the list of exponent differences of L_{inp} at its true singularities,

e_{rem} = the (possibly empty) list of exponent differences of L_{inp} at its removable singularities,

a_f = candidate algebraic degree.

OUTPUT: A list of all lists $e_B = [\alpha_0, \alpha_1, \alpha_\infty, d]$ of integers or rational numbers where $[\alpha_0, \alpha_1, \alpha_\infty]$ is a list of candidate exponent differences and d is a candidate degree d_f for f such that:

(i) For every exponent difference m in e_{inp} there exists $e \in \mathbb{Q}$ with $e \cdot a_f \in \{1, \ldots, d\}$ such that $m = e \cdot \alpha_i$ for some $i \in \{0, 1, \infty\}$.

(ii) The multiplicities e are consistent with (8), and their sums are compatible with d, see the last paragraph in Step 2.

(1) Let $\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3 = \alpha_0, \alpha_1, \alpha_\infty$. After reordering we may assume that $\overline{\alpha}_1, \ldots, \overline{\alpha}_k \in \mathbb{Z}$ and $\overline{\alpha}_{k+1}, \ldots, \overline{\alpha}_3 \notin \mathbb{Z}$ for $k \in \{0, 1, 2, 3\}$. For each $k \in \{0, 1, 2, 3\}$ we use `CoverLogs` in Imamoglu (2015a) to compute candidates for $\overline{\alpha}_1, \ldots, \overline{\alpha}_k \in \mathbb{Z}$.

Algorithm `CoverLogs` computes candidates that meet these requirements:

- Logarithmic singularities are true singularities with integer exponent differences. If L_{inp} has at least one logarithmic singularity s with exponent difference $\Delta(L_{inp}, s)$, then a candidate L_B must have at least one logarithmic singularity; at least one of the $\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3$ must be an integer that divides $a_f \cdot \Delta(L_{inp}, s)$, and for every $\overline{\alpha}_i \in \mathbb{Z}$ there must be at least one s such that $\overline{\alpha}_i$ divides $a_f \cdot \Delta(L_{inp}, s)$.
- $\Delta(L_{inp}, s) = 0$ for some s $\iff$ $0 \in \{\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3\}$.
- Theorem 2.1.

If $\overline{\alpha}_1 + \cdots + \overline{\alpha}_k \neq 0$, then algorithm `CoverLogs` also computes the exact degree d_f of f using Theorem 2.1 which shows that $d_f(\overline{\alpha}_1 + \cdots + \overline{\alpha}_k)/a_f$ must be the sum of the logarithmic exponent differences of L_{inp}. Otherwise, it uses (9) to compute a bound for d_f, and uses it as d_f to compute a candidate degree.

(2) We will explain only the case $a_f = 1$, and only $k = 1$, which is the case $[\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3] = [\alpha_0, \alpha_1, \alpha_\infty]$, where $\alpha_0 \in \mathbb{Z}$ and $\alpha_1, \alpha_\infty \notin \mathbb{Z}$.

Let $k = 1$. Let $\alpha_0 \in \mathbb{Z}$ be one of the candidates from algorithm `CoverLogs`. We need to find candidates for α_1 and α_∞.

The logarithmic singularities of L_{inp} come from the point 0. Non-integer exponent differences of L_{inp} must be multiples of α_1 or α_∞. Let S_N be the set of non-logarithmic exponent differences of L_{inp} and S_R be the set of exponent differences of L_{inp} at its removable singularities. Consider the set

$$\Gamma_1 = \begin{cases} \Gamma_A = \{\frac{\max(S_N)}{b} : b = 1, \ldots, d_f\} & \text{if } S_N \neq \emptyset, \\ \Gamma_B = \{\frac{a}{b} : a \in S_R \cup \{1\}, b = 1, \ldots, d_f\} & \text{otherwise.} \end{cases}$$

α_1 (or α_∞, but if so, we may interchange them) must be one of the elements of Γ_1. We loop over all elements of Γ_1. Assume that a candidate for α_1 is chosen. Let $\Omega = S_N \setminus \alpha_1 \mathbb{Z}$. Now consider the set

$$\Gamma_\infty = \begin{cases} \Gamma_A \cup \Gamma_B & \text{if } \Omega = \emptyset, \\ \{\frac{g}{b} : g = \gcd(\Omega) : b = 1, \ldots, d_f\} & \text{otherwise.} \end{cases}$$

11

Now take all pairs (α_∞, d) satisfying (19), $\alpha_\infty \in \Gamma_\infty$, $1 \leq d \leq d_f$, with additional restrictions on d, as follows:

For every potential non-zero value v for one of the α_i's we pre-compute a list of integers N_v by dividing all exponent differences of L_{inp} by v and then selecting the quotients that are integers. Next, let D_v be the set of all $1 \leq d \leq d_f$ that can be written as the sum of a sublist of N_v. Each time a non-zero value v is taken for one of the α_i, it imposes the restriction $d \in D_v$. This means that we need not run a loop for $\alpha_\infty \in \Gamma_\infty$, instead, we run a (generally much shorter) loop for d (taking values in the intersection of the D_v's so far) and then for each such d compute α_∞ from (19). We also check if $d \in D_{\alpha_\infty}$.

(3) Return the list of candidate exponent differences with a candidate degree, the list of lists $[\alpha_0, \alpha_1, \alpha_\infty, d]$, for candidate GHDOs.

3.5. Quotient Method

In this section, we explain a method to recover the pullback function f. We will explain our algorithm for rational pullback functions. For algebraic pullback functions, the only difference is the lifting algorithm, which is explained in Section 3.6. Note that we can always compute the formal solutions of a given differential equation $L_{inp}(y) = 0$ up to a finite precision.

3.5.1. Non-logarithmic Case

Let the second order differential equation $L_{inp}(y) = 0$ be given. Let L_B be a GHDO such that $L_B \xrightarrow{f}_C \xrightarrow{r}_E L_{inp}$. Let $f : \mathbb{P}^1_x \mapsto \mathbb{P}^1_z$ and $L_1 \xrightarrow{f}_C L_2$. If $x = p$ is a singularity of L_2 and $z = s$ is a singularity of L_1, then we say that p comes from s when $f(p) = s$.

After a change of variables we can assume that $x = 0$ is a singularity of L_{inp} that comes from the singularity $z = 0$ of L_B. This means $f(0) = 0$ and we can write $f = c_0 x^{v_0(f)} (1 + \dots)$ where $c_0 \in \mathbb{C}$, $v_0(f)$ is the multiplicity of 0, and the dots refer to an element in $x\mathbb{C}[[x]]$.

Let y_1 and y_2 be the formal solutions of L_B at $x = 0$. The following diagram shows the effects of the change of variables and exp-product transformations on the formal solutions of L_B,

$$y_i(x) \xrightarrow{f}_C y_i(f) \xrightarrow{r}_E Y_i(x) = \exp\left(\int r dx\right) y_i(f), \quad i \in \{1, 2\}$$

where Y_1 and Y_2 are solutions of L_{inp}.

Let $q = \frac{y_1}{y_2}$ be a quotient of formal solutions of L_B. The change of variables transformation sends x to f, and so q to $q(f)$. Therefore, $q(f)$ will be a quotient of formal solutions of L_{inp}.

The effect of exp-product transformation disappears under taking quotients. In general, a quotient of formal solutions of L_B at a point $x = p$ is only unique up to Möbius transformations $\frac{y_1}{y_2} \mapsto \frac{\alpha y_1 + \beta y_2}{\gamma y_1 + \eta y_2}$.

If $x = p$ has a non-integer exponent difference, then we can choose q uniquely up to a constant factor c. So if we likewise compute a quotient Q of formal solutions of L_{inp}, then we have $q(f) = c \cdot Q(x)$ for some unknown constant c. Then

$$f(x) = q^{-1}(c \cdot Q(x)). \tag{20}$$

If we know the value of this constant c, then (20) allows us to compute an expansion for the pullback function f from expansions of q and Q. To obtain c with a finite computation, we take a prime number ℓ. Then, for each $c \in \{1, \ldots, \ell - 1\}$ we try to compute f modulo ℓ in $\mathbb{F}_\ell(x)$ using series-to-rational function reconstruction. If this succeeds, then we lift f modulo a power of ℓ, and try to find $f \in \mathbb{Q}(x)$ with rational number reconstruction. Details of lifting are in Section 3.6.

Remark 3.2. We compute formal solutions up to a precision $a \geq (a_f + 1)(d_f + 1) + 6$. This suffices to recover the correct pullback function with a few extra terms to reduce the number of false positives.

Algorithm 3.3. General Outline of `case_1: non_logarithmic case`.

    INPUT: L_{inp}, L_B, d_f, a_f, where

        L_{inp} = input differential operator,

        L_B = candidate GHDO,

        d_f = candidate degree for f,

        a_f = candidate algebraic degree for f.

    OUTPUT: $[f, r]$ or 0, where

        f = pullback function,

        r = parameter of exp-product transformation.

(1) Compute formal solutions y_1, y_2 of L_B and Y_1, Y_2 of L_{inp} up to precision $a \geq (a_f + 1)(d_f + 1) + 6$.

(2) Compute $q = \frac{y_2}{y_1}$, $Q = \frac{Y_2}{Y_1}$, and q^{-1}.

(3) Select a prime ℓ for which these expansions can be reduced mod ℓ.

(4) For each c_0 in $\{1, \ldots, \ell - 1\}$:

    4.1. Evaluate $\overline{f}_{1,c_0} = q^{-1}(c_0 \cdot Q) \in \mathbb{Z}[x]/(\ell, x^a)$.

    4.2. If $a_f = 1$ then try rational function reconstruction for $\overline{f}_{1,c_0}$ (the case $a_f > 1$ is explained in Section 3.6.2).

       • If rational function reconstruction succeeds and produces f_{1,c_0}, then store c_0 and f_{1,c_0}.

       • If rational function reconstruction fails for every c_0, then return 0.

(5) For n from 2 (see Remark 3.4 in Section 3.6.1):

    For each stored c_0:

    5.1. Using the techniques explained in Section 3.6 lift f_{n-1,c_0} to f_{n,c_0}.

    5.2. f_{n,c_0} is a candidate for f mod ℓ^n. Try to obtain f from this with rational number reconstruction. If this succeeds, compute M such that $L_B \xrightarrow{f}_C M$. Compute r such that $M \xrightarrow{r}_E L_{inp}$, if it exists (see Section 3.7). If so, return f and r.

3.5.2. *Logarithmic Case*

A logarithm may occur in one of the formal solutions of L_{inp} at $x = p$ if exponents at $x = p$ differ by an integer. We may assume that L_{inp} has a logarithmic solution at the singularity $x = 0$.

Let y_1, y_2 be the formal solutions of L_B at $x = 0$. Let y_1 be the non-logarithmic solution (it is unique up to a multiplicative constant). Then $\frac{y_2}{y_1} = c_1 \cdot \log(x) + h$ for some $c_1 \in \mathbb{C}$ and $h \in \mathbb{C}[[x]]$. We can choose y_2 such that

$$c_1 = 1 \quad \text{and} \quad \text{constant term of } h = 0. \tag{21}$$

That makes $\frac{y_2}{y_1}$ unique. If h does not contain negative powers of x then define

$$g = \exp\left(\frac{y_2}{y_1}\right) = x \cdot (1 + \dots) \tag{22}$$

where the dots refer to an element of $x\mathbb{C}[[x]]$.

Remark 3.3. If we choose y_2 differently, then we obtain another $\tilde{g} = \exp\left(\frac{y_2}{y_1}\right)$ that relates to g in (22) by $\tilde{g} = c_1 g^{c_2}$ for some constants c_1, c_2. If h contains negative powers of x, then the formula for g is slightly different (we did not implement this case, instead we use Section 4 to transform equations.).

We do likewise for the formal solutions Y_1, Y_2 of L_{inp} and denote

$$G = \exp\left(\frac{Y_2}{Y_1}\right) = x \cdot (1 + \dots). \tag{23}$$

Write $f \in \mathbb{C}(x)$ as $c_0 x^{v_0(f)} \cdot (1 + \dots)$. Then $g(f) = c \cdot x^{v_0(f)}(1 + \dots)$. Note that g, G are not intrinsically unique, the choices we made in (21) implies that

$$g(f) = c_1 \cdot G^{c_2} \tag{24}$$

for some constants c_1, c_2. Here $c_1 = c$ and $c_2 = v_0(f)$.

If $\Delta(L_{inp}, 0) \neq 0$, then find $v_0(f)$ from $\Delta(L_B, 0)v_0(f) = \Delta(L_{inp}, 0)$. Otherwise we loop over $v_0(f) = 1, 2, \dots, d_f$. That leaves one unknown constant c. We address this problem as before, choose a good prime number ℓ, try $c = 1, 2, \dots, \ell - 1$. Then calculate an expansion for f with the formula

$$f = g^{-1}\left(c \cdot G^{v_0(f)}\right). \tag{25}$$

Algorithm 3.4. General Outline of `case_2: logarithmic case`.
      INPUT: L_{inp}, L_B, d_f, a_f, where
            $L_{inp} = $ input differential operator,
            $L_B = $ candidate GHDO,
            $d_f = $ candidate degree for f,
            $a_f = $ candidate algebraic degree for f.
      OUTPUT: $[f, r]$ or 0, where
            $f = $ pullback function,
            $r = $ parameter of exp-product transformation.
(1) Compute the exponents of L_{inp} and L_B. If $\Delta(L_{inp}, 0) = 0$, then replace L_{inp} with L defined in Remark (3.3) above. Otherwise let $L = L_{inp}$.
(2) Compute formal solutions y_1, y_2 of L_B and Y_1, Y_2 of L up to precision $a \geq (a_f + 1)(d_f + 1) + 6$.
(3) Compute $q = \frac{y_2}{y_1}$, $Q = \frac{Y_2}{Y_1}$. Compute g, G from (24) and (25) respectively, and g^{-1}.
(4) Same as in Algorithm 3.3 Step 3.
(5) Compute $v_0(f)$ and search for c_0 value(s) such that c could be $\equiv c_0 \mod \ell$ by looping over $c_0 = 1, \dots, \ell - 1$. If $\Delta(L_{inp}, 0) = 0$, then also simultaneously loop over $v_0(f) = 1, \dots, d_f$ to find $v_0(f)$.
        For each c_0 in $\{1, \dots, \ell - 1\}$:
        5.1. Evaluate $\overline{f}_{1,c_0} = g^{-1}\left(c_0 \cdot G^{v_0(f)}\right) \in \mathbb{Z}[x]/(\ell, x^a)$.

14

5.2. Try rational function or algebraic function reconstruction for $\overline{f}_{1,c_0}$ as in Algorithm 3.3 Step 4.2.

(6) Same as in Algorithm 3.3 Step 5.

3.6. *Lifting: Recovering the Pullback Function*

We explain lifting by using the formula (20) for the pullback function, which occurs in the non-logarithmic case. The algorithm for the formula (25) in the logarithmic case is similar.

3.6.1. *Lifting for a Rational Pullback Function*

By using the formula (20), which is $f(x) = q^{-1}(c \cdot Q(x))$, we can recover the rational pullback function f, if we know the value of the constant c. We do not have a direct formula for c. However, if we know c_0 such that

$$c \equiv c_0 \mod \ell$$

for a good prime number ℓ, then we can recover the pullback function f. This can be done via *Hensel lifting techniques.*

Let ℓ be a good prime number and consider

$$h : \mathbb{Q} \longrightarrow \mathbb{Q}[x]/(x^a)$$
$$h(c) \equiv q^{-1}(c \cdot Q(x)) \mod x^a.$$

By looping on $c_0 = 1, \ldots, \ell - 1$ and trying rational function reconstruction for $h(c_0)$ mod (ℓ, x^a), we can compute the image of $f \in \mathbb{F}_\ell(x)$ from its image in $\mathbb{F}_\ell[x]/(x^a)$. If a is high enough, then for correct value(s) of c_0, rational function reconstruction will succeed and return a rational function $\frac{A_0}{B_0}$ mod ℓ. This c_0 is the one satisfying $c \equiv c_0$ mod ℓ.

Write

$$c \equiv c_0 + \ell c_1 \mod \ell^2$$

for $0 \leq c_1 \leq \ell - 1$. Taylor series expansion of h gives us

$$h(c) = h(c_0 + \ell c_1) \equiv h(c_0) + \ell c_1 h'(c_0) \mod (\ell^2, x^a). \tag{26}$$

Substitute $c_1 = 0$, $c_1 = 1$, respectively, in (26) and compute

$$h(c_0) \mod (\ell^2, x^a), \tag{27}$$
$$h(c_0 + \ell) \equiv h(c_0) + \ell h'(c_0) \mod (\ell^2, x^a). \tag{28}$$

Subtracting (27) from (28) gives

$$\ell h'(c_0) \equiv [h(c_0 + \ell) - h(c_0)] \mod (\ell^2, x^a).$$

Let

$$E_{c_1} = h(c_0) + c_1 \ell h'(c_0) \tag{29}$$

where c_1 is an unknown constant. Suppose $f = \frac{A}{B}$ in characteristic 0. We do not know what A and B are. However, from applying rational function reconstruction for $h(c_0)$, we obtain A_0, B_0 with $f \equiv \frac{A_0}{B_0}$ mod (ℓ, x^a). It follows that

$$f = \frac{A}{B} \equiv \frac{A_0}{B_0} \equiv E_{c_1} \mod (\ell, x^a).$$

From this equation we have

$$A \equiv B E_{c_1} \mod (\ell, x^a). \tag{30}$$

Now let

$$f = \frac{A}{B} \equiv \frac{A_0 + \ell A_1}{B_0 + \ell B_1} \mod (\ell^2, x^a) \tag{31}$$

where

$$A_1 = a_0 + a_1 x + \cdots + a_{\deg(A_0)} x^{\deg(A_0)}$$
$$B_1 = b_1 x + \cdots + b_{\deg(B_0)} x^{\deg(B_0)}$$

are unknown polynomials. Here we are fixing the constant term of B. We need values of $\{a_i, b_j\}$ to find $f \mod (\ell^2, x^a)$. From (30), we have

$$(A_0 + \ell A_1) \equiv (B_0 + \ell B_1) \cdot E_{c_1} \mod (\ell^2, x^a). \tag{32}$$

Now, solve the linear system (32) for unknowns $\{a_i, b_j, c_1\}$ in $\mathbb{F}_\ell$. From (31) find $f \mod (\ell^2, x^a)$ and $c \equiv c_0 + \ell c_1 \mod \ell^2$.

Try rational number reconstruction after each Hensel lift. If it succeeds, then check if this rational function is the one that we are looking for as in the last step of Algorithm 3.3. If it is not, then lift $f \mod (\ell^2, x^a)$ to mod (ℓ^3, x^a) (or (ℓ^4, x^a) if an implementation for solving linear equations mod ℓ^n is available). After a (finite) number of steps, we can recover the rational pullback function f.

Remark 3.4. Our implementation gives up when the prime power becomes "too high"; (a proven bound is still lacking, but would be needed for a rigorous algorithm).

3.6.2. Lifting for an Algebraic Pullback Function

We can recover algebraic pullback functions in a similar way. However, we need to know $a_f = [\mathbb{C}(x, f) : \mathbb{C}(x)]$. The idea is to recover the minimal polynomial of f.

Let $d_f = [\mathbb{C}(x, f) : \mathbb{C}(f)]$. Consider the polynomial in y

$$\sum_{j=0}^{a_f} A_j y^j \mod (\ell, x^a) \tag{33}$$

with unknown polynomials

$$A_j = \sum_{i=0}^{d_f} a_{i,j} x^i$$

where $j = 0, \ldots, a_f$.

First we need to find the value of c_0 such that $c_0 \equiv c \mod \ell$. As before, by looping on $c_0 = 1, \ldots, \ell - 1$, we can compute the corresponding f_{c_0} which is a candidate for $f \mod (x^a, \ell)$ in $\mathbb{F}_\ell[x]/(x^a)$. The polynomial (33) should be congruent to 0 mod (ℓ, x^a) if we plug in f_{c_0} for y. Solve the system

$$\sum_{j=0}^{a_f} A_j f_{c_0}^j \equiv 0 \mod (\ell, x^a)$$

over $\mathbb{F}_\ell$ and find the unknown polynomials $A_j \mod \ell$. Then let

$$c \equiv c_0 + \ell c_1 \mod \ell^2.$$

16

Now let E_{c_0} be as in (29) and consider the system

$$\sum_{j=0}^{a_f}(A_j + \ell\tilde{A}_j)E_{c_0}^j \equiv 0 \mod (\ell^2, x^a).$$

Solve it over $\mathbb{F}_\ell$ to find c_1 and the unknown polynomials $\tilde{A}_j$. After a finite number of lifting steps and rational reconstruction, we will have the minimal polynomial of $\sum A_j y^j$ of f in $\mathbb{Q}[x, y]$.

3.7. Recovering the Parameter of Exp-product

After finding f, we can compute the differential operator M, such that

$$L_B \xrightarrow{f}_C M \xrightarrow{r}_E L_{inp}.$$

Then we can compare the second highest terms of M and L_{inp} to find the parameter r of the exp-product transformation: If $M = \partial^2 + B_1\partial + B_0$ and $L_{inp} = \partial^2 + A_1\partial + A_0$, then

$$r = \frac{B_1 - A_1}{2}.$$

4. Computing an Integral Basis for a Linear Differential Operator

We tested Algorithm 3.1 on many examples, including from the Online Encyclopedia of Integer Sequences, (https://oeis.org). Another source of examples comes from Bousquet-Melou and Mishna (2010); Bostan and Kauers (2009); Bostan et al. (in Progress) (see Imamoglu (2015b) for these operators). Four of them have solutions in the form of (1) and Algorithm 3.1 finds these solutions. However, Algorithm 3.1 does not solve the other operators from that list. We know from Bousquet-Melou and Mishna (2010) and Bostan and Kauers (2009) that these operators do have solutions in the form of (2). It means that these operators must be gauge equivalent to operators with solutions in the form of (1). The question is *how can we find these gauge transformations?* As mentioned in the introduction the key idea is to follow Cohen and Diaz Y Diaz (1991) POLRED's strategy; compute an integral basis and then normalize it at infinity. Then we can select an element with minimal valuations at infinity. This element gives us a gauge transformation.

We modified the algorithm explained in Kauers and Koutschan (2015), which is an analogue of the algorithm in van Hoeij (1994), and implemented our own version of the integral basis procedure for second order regular singular linear differential operators. Our integral basis algorithm first finds local integral bases for each finite singularity of L_{inp}, then combines all of these local bases, and at the end normalizes the basis at infinity in the sense of (Trager, 1984, Section 2.3) (see Section 4.2).

4.1. Integral Bases

Definition 4.1. The *local parameter* t_p of a point $p \in \mathbb{C}\cup\{\infty\}$ is defined as $t_p = x - p$ if $p \neq \infty$ and $t_p = \frac{1}{x}$ otherwise. If $L \in \mathbb{C}(x)[\partial]$ is regular singular, then all of the solutions of L at a point $x = p$ are in the form

$$f = t_p^{\nu_p}\sum_{i=0}^{\infty} P_i t_p^i$$

where $\nu_p \in \mathbb{C}$ and $P_i \in \mathbb{C}[\log(t_p)]$ with $\deg(P_i) < \operatorname{ord}(L)$.

Definition 4.2. Let y be a solution of L at $x = 0$,

$$y = x^{\nu_0} \sum_{i=0}^{\infty} P_i x^i$$

where $P_i \in \mathbb{C}[\log(x)]$. The *valuation* of y at $x = 0$ is defined as follows:

$$v_0(y) := \nu_0 + \inf\{i \mid P_i \neq 0\}. \tag{34}$$

We say that y is *integral* at $x = 0$ if $\operatorname{Re}(v_0(y)) \geq 0$.

Definition 4.3. If $p \in \mathbb{C} \cup \{\infty\}$ and y is a solution of L at $x = p$, then the valuation of y at $x = p$ is defined as in (34) with x replaced by t_p.

Definition 4.4. Let $L \in \mathbb{C}(x)[\partial]$ and $G \in \mathbb{C}(x)[\partial]$. The operator G is called *integral* for L if

$$\forall_{p \in \mathbb{C}} \operatorname{Re}(v_p(G(y))) \geq 0. \tag{35}$$

We may assume that $\operatorname{ord}(G) < \operatorname{ord}(L)$ because $G = QL + R$ for some $Q, R \in \mathbb{C}(x)[\partial]$ such that $\operatorname{ord}(R) < \operatorname{ord}(L)$, and we may replace G by R without changing $G(y)$ in (35), i.e., we may interpret G as an element of $\mathbb{C}(x)[\partial]/\mathbb{C}(x)[\partial]L$.

Definition 4.5. Let $L \in \mathbb{C}(x)[\partial]$ and

$$M_L = \{G \in \mathbb{C}(x)[\partial] \mid G \text{ is integral for } L \text{ and } \operatorname{ord}(G) < \operatorname{ord}(L)\}.$$

A basis of M_L as $\mathbb{C}[x]$-module is called an *integral basis*.

4.2. Normalization at Infinity

Assume that we computed an integral basis $[B_0, B_1]$ for a second order regular singular $L \in \mathbb{Q}(x)[\partial]$. One can normalize $[B_0, B_1]$ at infinity as follows:

Compute the formal solutions Y_1, Y_2 of L_{inp} at $x = \infty$. Compute the pole orders (the pole order is the valuation as in (34) multiplied by -1) of

$$B_0(Y_1), \quad B_0(Y_2), \quad B_1(Y_1), \quad B_1(Y_2) \tag{36}$$

at the point $x = \infty$. Let the maximum of the pole orders be m and let it come from $B_i(Y_j)$ where $i \in \{0, 1\}$ and $j \in \{1, 2\}$. Form the ansatz

$$\mathfrak{B} = B_i(Y_j) - C \cdot x^{(m-n)} B_k(Y_j).$$

Here $k \in \{0, 1\}$, $k \neq i$, and n is the pole order of $B_k(Y_j)$ at $x = \infty$. For a suitable C, the pole order of $\mathfrak{B}$ at $x = \infty$ will be less than m. Find this C. Update $B_i = B_i - C \cdot x^{(m-n)} B_k$. Now we have the updated basis $[B_0, B_1]$. Compute the pole orders, for this updated basis, as in (36). The possibilities for the updated basis $[B_0, B_1]$ are

(1) one of the pole orders decreases and none of them increases,
(2) one of the other pole orders increases.

If 1 occurs, then it means that we made an improvement, we are making the pole orders smaller. Repeat this process until there is no improvement possible (case 2). Then return the normalized basis $[B_0, B_1]$.

4.3. Finding a Suitable Gauge Transformation

By using an integral basis $[B_0, B_1]$, which is normalized at infinity, for L_{inp}, we want to find a gauge transformation $\mathcal{G}$ such that $\mathcal{G}$ transforms L_{inp} to $\tilde{L}_{inp}$ such that $L_B \xrightarrow{f}_C \xrightarrow{r}_E \tilde{L}_{inp}$. We observed that for the operators coming from Bousquet-Melou and Mishna (2010) and Bostan and Kauers (2009), one of the basis elements always gives such a gauge transformation $\mathcal{G}$. We tested our main algorithm on other examples as well and it turns out that this approach is very effective.

Algorithm 4.1. General Outline of `hypergeometricsols`.

INPUT: $L_{inp} \in \mathbb{Q}(x)[\partial]$ and (optional) $a_f max$ where

L_{inp} = a second order regular singular irreducible operator,

$a_f max$ = bound for the algebraic degree a_f. If omitted, then $a_f max = 2$ which means our implementation tries $a_f = 1$ and $a_f = 2$.

OUTPUT: Solutions of L_{inp} in the form of (2), or an empty list.

(1) Try to find solutions of L_{inp} in the form of (1) by using the Algorithm 3.1 in Section 3. If none are found go to Step 2.

(2) Compute an integral basis $[B_0, B_1]$ for L_{inp} and normalize this basis at infinity by using the method given in Section 4.2. Each basis element B_k, ($k \in \{0, 1\}$), is a candidate gauge transformation. Transform L_{inp} to another operator using B_k. Try to find solutions of the new operator in the form of (1) by using Algorithm 3.1. If this new operator has solutions of type (1), then apply the inverse of the gauge transformation to these solutions to form the solutions of L_{inp} of type (2), and return them. Otherwise return an empty list.

A Maple implementation of Algorithm 4.1 and examples can be found at Imamoglu (2015b).

References

Baldassari, F., Dwork, B. 1979. On Second Order Linear Differential Equations with Algebraic Solutions. American Journal of Mathematics 101, 42–76.

Bostan, A., Chyzak, F., Kauers, M., Pech, L., van Hoeij, M., in Progress. Explicit Differentiably Finite Generating Functions of Walks with Small Steps in the Quarter Plane, in Progress.

Bostan, A., Chyzak, F., van Hoeij, M., Pech, L., 2011. Explicit Formula for Generating Series of Diagonal 3d Rook Paths. Seminaire Lotharingien de Combinatorie.

Bostan, A., Kauers, M., 2009. Automatic Clasification of Restricted Lattice Walks. FPSAC'09 Proceedings.

Bousquet-Melou, M., Mishna, M., 2010. Walks with Small Steps in the Quarter Plane. Contemporary Mathematics 520, 1–39.

Cohen, H., Diaz Y Diaz, F., 1991. A Polynomial Reduction Algorithm. Journal de théorie des nombres de Bordeaux, 351–360.

Debeerst, R., 2007. Solving Differential Equations in terms of Bessel Functions. Master's thesis, Universität Kassel.

Fang, T., 2012. Solving Linear Differential Equations in terms of Hypergeometric Functions by 2-Descent. Ph.D. thesis, Florida State University.

Fang, T., van Hoeij, M., 2011. 2-Descent for Second Order Linear Differential Equations. ISSAC'11 Proceedings, 107–114.

Hartshorne, R., 1977. Algebraic Geometry. Springer.

Imamoglu, E., 2015a. Implementation of `find_2f1`. `www.math.fsu.edu/~eimamogl/find_2f1`.

Imamoglu, E., 2015b. Implementation of `hypergeometricsols`. `www.math.fsu.edu/~eimamogl/hypergeometricsols`.

Imamoglu, E., van Hoeij, M., 2015. Computing Hypergeometric Solutions of Second Order Linear Differential Equations using Quotients of Formal Solutions. ISSAC'15 Proceedings, 235–242.

Ince, E. L., 1926. Ordinary Differential Equations. Dover Publications, Inc., New York.

Kauers, M., Koutschan, C., 2015. Integral D-Finite Functions. ISSAC'15 Proceedings, 251–258.

Kovacic, J., 1986. An Algorithm for Solving Second Order Linear Homogeneous Equations. Journal of Symbolic Computation 2 (1), 2–43.

Kunwar, V. J., 2014. Hypergeometric Solutions of Linear Differential Equations with Rational Function Coefficients. Ph.D. thesis, Florida State University.

Kunwar, V. J., van Hoeij, M., 2013. Second Order Differential Equations with Hypergeometric Solutions of Degree Three. ISSAC'13 Proceedings, 235–242.

Schwarz, H. A., 1873. Ueber diejenigen Fälle, in welchen die Gaussische hypergeometrische Reihe eine algebraische Function ihres vierten Elementes darstellt. Journal für die Reine und Angewandte Mathematik, 292–335.

Trager, B. M., 1984. Integration of Algebraic Functions. Ph.D. thesis, Massachusetts Institute of Technology.

van der Put, M., Singer, M. F., 2003. Galois Theory of Linear Differential Equations. Vol. 328 of Grundlehren der mathematischen Wissenschaften. Springer.

van Hoeij, M., 1994. An Algorithm for Computing an Integral Basis in an Algebraic Function Field. Journal of Symbolic Computation 18 (4), 353–363.

van Hoeij, M., Vidunas, R., 2015. Belyi Functions for Hyperbolic Hypergeometric-to-Heun Transformations. Journal of Algebra 441, 609–659.

Wang, Z. X., Guo, D. R., 1989. Special Functions. World Scientific.

Yuan, Q., 2012. Finding All Bessel Type Solutions for Linear Differential Equations with Rational Function Coefficients. Ph.D. thesis, Florida State University.