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

Erdal Imamoglu*
Department of Mathematics
Florida State University
Tallahassee, FL 32306, USA
eimamogl@math.fsu.edu

Mark van Hoeij*
Department of Mathematics
Florida State University
Tallahassee, FL 32306, USA
hoeij@math.fsu.edu

ABSTRACT
Let \( L \) be a second order differential equation with coefficients in \( \mathbb{C}(x) \). The goal of this paper is to find solutions of \( L \) in the form

\[
\exp\left( \int r \, dx \right) \cdot _2F_1(a_1, a_2; b_1; f)
\]

where \( r, f \in \mathbb{Q}(x) \), and \( a_1, a_2; b_1 \in \mathbb{Q} \).

Categories and Subject Descriptors
I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms; G.4 [Mathematics of Computing]: Mathematical Software

General Terms
Algorithms

Keywords
Symbolic Computation, Differential Equations, Closed Form Solutions, Hypergeometric Solutions

1. INTRODUCTION
Consider a second order homogenous linear differential equation with rational function coefficients \( A_i \in \mathbb{C}(x) \)

\[ A_2 y'' + A_1 y' + A_0 y = 0. \]

which corresponds to the differential operator

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

where \( \partial = \frac{d}{dx} \). Then \( (2) \) is the equation \( L(y) = 0 \).

This paper gives a (heuristic) algorithm to find a solution of \( (2) \) in the form of \( (1) \). This form is both more and less general than in prior work. Less general in the sense that papers \( 2, 7 \) considered 3 transformations instead of the 2 in section \( 2.3 \) and more general in the sense that prior work was restricted to either a specific number of singularities (4 in \( 9 \) and 5 in \( 6 \)) or specific degrees (degree 3 in \( 7 \) and a degree-2 decomposition in \( 2 \)). Moreover, our program can also find algebraic functions \( f \) in \( (1) \) (although at the moment this requires additional user inputs).

We assume that \( (2) \) has no Liouvillian solutions (this implies it is irreducible), otherwise one can solve it with Kovacic’s algorithm \( 5 \). The goal of this paper is: Given a second order operator \( L_{inp} \in \mathbb{C}(x)[\partial] \), regular singular without Liouvillian solutions, find a solution of form \( (1) \) if it exists. This means finding \( a_1, a_2, b_1 \in \mathbb{Q} \) and finding transformations (sections \( 2.3 \) and \( 3.2 \)) that send \( L_B \) to the input equation \( L_{inp} \), where \( L_B \) is the minimal operator of \( _2F_1(a_1, a_2; b_1; x) \).

Two crucial steps of this task are: (1) find (candidates for) \( a_1, a_2, b_1 \) and (2) find the pullback function \( f \) (after that, finding \( r \) becomes easy). Given \( a_1, a_2, b_1 \) (or equivalently, \( L_B \)), by comparing quotients of formal solutions of \( L_B \) and \( L_{inp} \), we can compute \( f \) if we know the value of a certain constant \( c \). We have no direct formula for \( c \); to obtain it with a finite computation, we take a prime number \( \ell \). Then, for each \( c \in \{1, \ldots, \ell - 1\} \) we try to compute \( f \) modulo \( \ell \). If this succeeds, then we lift \( f \) modulo a power of \( \ell \), and try reconstruction.

Example 1. Rational Pullback Function

\[ L = 21x(x-1)(x+1)^2 + (38x^2 - 6x - 14)\partial + \frac{20x - 5}{7} \]

has a \( _2F_1 \)-type solution

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

where

\[
\exp\left( \int r \, dx \right) = (x+1)^{-\frac{5}{24}} \quad \text{and} \quad f = \frac{4x}{(x+1)^2} \quad (3)
\]

Here the degree of the pullback function \( f \) is 2. We can find this solution with the quotient method in remark \( 7 \) below. In the quotient method, the parameters \( a_1, a_2, b_1 \) (here \( \frac{5}{24}, \frac{11}{42}, \frac{2}{3} \)) and the degree of \( f \) (here 2) are taken as an input. We implemented section \( 3.2 \) which computes candidates for

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]

\[ \]
has a $2F_1$-type solution

$$Y(x) = \exp(-\frac{1}{2} \int r \, dx) \, 2F_1(\frac{1}{3}, \frac{1}{3}; 1; f)$$

where

$$r = \frac{-x^5 + 22 \, x^4 - 55 \, x^3 + 343 \, x^2 + 6 \, x \left(x^2 - 7 \, x + 1\right) \sqrt{x^2 - 34 \, x + 1} + 58 \, x - 1}{x \left(x^2 - 41 \, x + 1206 \, x^4 - 41 \, x + 1\right) \left(x + 1\right)}$$

and

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

Problem Description: Given a second order linear differential operator $L_{inp} \in \mathbb{C}(x)[\partial]$, 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 $E_1$. 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 $L_B \xrightarrow{\rho_{CE}} \xrightarrow{E} L_{inp}$.

The general outline is as follows.

Algorithm Outline: find_2f1

Input:
- $L_{inp}$, a second order differential operator.
- At the moment we only handle coefficients in $\mathbb{Q}$. If $f$ in $E_1$ is algebraic, then our current implementation needs three more inputs which are
  - $L_B$, a candidate GHDO,
  - $a_f$, an algebraic degree bound for $f$,
  - $d_f$, degree bound for $f$.

Output:
- A list of basis elements of solutions of $L_{inp}$ in form $E_1$, or an empty list $\emptyset$.

2. If $L_B,a_f,d_f$ are not provided in the input, then use section 3.2 (at the moment this only covers rational $f$’s, i.e. $a_f = 1$) to compute candidates for $L_B$ and $d_f$.
3. For a candidate GHDO $L_B$, compute formal solutions of $L_B$ and $L_{inp}$ at a non-removable singularity (see remark 3 in section 2.3) up to precision $a \geq 2(a_f + 1)(d_f + 1) + 3$. Take the quotients of formal solutions and compute series expansions for $q^{-1}$ and $Q$ (in order to compute $f = q^{-1}(cQ(x))$ in the next step).
4. Choose a good prime number $\ell$, and try to find $c$ mod $\ell$ by looping $c = 1, 2, \ldots, \ell - 1$ as in section 4.3. If no solution is found, then proceed with the next candidate GHDO (if any) in step 3. If no candidates remain, then return an empty list $\emptyset$.
5. Compute $f$ mod $(x^\alpha, \ell)$ and then use Hensel lifting to find $f$ mod higher powers of $\ell$. After each lifting try rational reconstruction. If it does not fail, then we have $f$.
6. Compute the parameter $r$ of the exp-product transformation (section 4.5).
7. Return a basis of $2F_1$-type solutions of $L_{inp}$.
3.1 General Degree Bound

Let X and Y be two algebraic curves with genus \( g_X \) and \( g_Y \), and let \( f : X \to Y \) be a non-constant analytic map. The Riemann-Hurwitz formula says

\[
2g_X - 2 = \deg(f)(2g_Y - 2) + \sum_{p \in X} (e_p - 1). \tag{6}
\]

Here \( p \) is a branching point and \( e_p \) is its ramification order. In this paper \( f : \mathbb{P}^1 \to \mathbb{P}^1 \) so \( g_X = g_Y = 0 \) and

\[
\sum_{p \in \mathbb{P}^1} (e_p - 1) = 2\deg(f) - 2. \tag{7}
\]

In section 3.1.1 and 3.1.2 we compute a degree bound for a rational pullback function \( f \) from formula (7). In section 3.1.3 we use it to compute a formula for \( \alpha_0 + \alpha_1 + \alpha_\infty \), the sum of the exponent-differences of \( L_B \).

3.1.1 Bound for Logarithmic Cases

Let \( L_B \) be a GDHO with at least one logarithmic singularity. Assume that \( L_B \rightarrow_{\mathcal{D}} L_{\text{inp}} \). Let \( d_f = \deg(f) \). The number of elements in the set \( T = f^{-1}(\{0,1,\infty\}) \) can be at most \( 3d_f \).

\[
\#T = \sum_{p \in T} 1 = \sum_{p \in T} (e_p - 1) = 3d_f - \sum_{p \in T} (e_p - 1). \tag{8}
\]

From (7), we have

\[
0 \leq \sum_{p \in T} (e_p - 1) = \sum_{p \in \mathbb{P}^1} (e_p - 1) = 2d_f - 2
\]

where the latter sum is taken over all branching points of \( f \). Hence \( d_f + 2 \leq \#T \leq 3d_f \).

The set of true singularities of \( L_{\text{inp}} \) is a subset of \( T \) and these two sets do not need to be equal. Points in \( T \) come from \( p \) comes from \( s \) when \( f(p) = s \) the singular points \( \{0,1,\infty\} \) of \( L_B \). Such points need not be singular, for instance, if \( L_B \) has exponents \( 0,1/3 \) at \( x = 0 \) and \( f \) has a root \( p \) of order \( e_p = 3 \), then the exponents at \( x = p \) will be \( 3 \cdot \{0,1/3\} = \{0,1\} \) and \( x = p \) will be a regular point (a "disappeared singularity"). We define the set of disappeared singularities as \( T - \text{Sing}(L_{\text{inp}}) \). Logarithmic singularities do not disappear; if \( s \in \{0,1,\infty\} \) is a logarithmic singularity of \( L_B \), then every point \( p \) above \( s \) is a logarithmic singularity as well.

Let \( n_{\text{diss}} \) be the number of disappeared singularities of \( L_{\text{inp}} \). For a GDHO with exponent differences \( \{0,1/2,1/3\} \) at \( 0,1,\infty \) respectively, \( n_{\text{diss}} \leq \frac{1}{2}d_f + \frac{1}{3}d_f \), with equality if and only if every point above \( s \) with exponent difference \( \alpha = 1/2 \), respectively \( \alpha = 1/3 \) disappears (i.e., \( e_p = 2 \), respectively \( e_p = 3 \)). So, if the total number of true singularities of \( L_{\text{inp}} \) is \( n_{\text{true}} \), then

\[
n_{\text{true}} = \#T - n_{\text{diss}} = \left(3d_f - \sum_{p \in S} (e_p - 1) \right) = n_{\text{diss}}
\]

\[
\geq \left[3d_f - (2d_f - 2) \right] - n_{\text{diss}} = d_f + 2 - n_{\text{diss}}
\]

\[
\geq d_f + 2 - \left(\frac{1}{2}d_f + \frac{1}{3}d_f \right) = \frac{1}{6}d_f + 2
\]

and so

\[
d_f \leq 6(n_{\text{true}} - 2). \tag{9}
\]

Inequality (9) is an upper bound for \( d_f \) in all cases with at least one logarithmic singularity. This is because \( \frac{1}{6}d_f + \frac{1}{3}d_f \) is an upper bound for the number of disappeared singularities in the logarithmic case (the GDHO cannot have two singularities with exponent difference \( \frac{1}{2} \) if it is irreducible, this makes \( \frac{1}{6}d_f + \frac{1}{3}d_f \) the maximum possible value for \( n_{\text{diss}} \) in the logarithmic case).

3.1.2 Bound for Non-Logarithmic Cases

In the non-logarithmic case one could have disappeared singularities above all three singularities \( \{0,1,\infty\} \) of the GDHO. The maximal degree bound is achieved at exponent differences \( \{\frac{1}{2},\frac{1}{2},\frac{1}{2}\} \). All \( L_B \)'s with a higher bound such as \( \{\alpha_0,\alpha_1,\alpha_\infty\} = \{\frac{1}{2},\frac{1}{2},\frac{1}{2}\} \), etc, are either reducible or appear in Schwarz's list \( \mathcal{S} \), which means they have Liouvillian solutions.

The maximum number of disappeared singularities for \( \{\frac{1}{2},\frac{1}{2},\frac{1}{2}\} \) is not \( \frac{1}{2}d_f + \frac{1}{3}d_f \) because this contradicts the formula (7). The maximum number consistent with (7) is

\[
\left(\frac{1}{2} + \frac{1}{3}\right)d_f + \frac{1}{7} - \frac{2 - \frac{2}{3}d_f}{2} - \frac{3 - \frac{1}{3}d_f}{3}
\]

and it leads to

\[
d_f \leq 36 \left(n_{\text{true}} - \frac{7}{3}\right). \tag{10}
\]

We use inequality (10) as an a priori upper bound for \( d_f \) for all cases with no logarithmic singularity.

Therefore, an a priori degree bound for a rational pullback function \( f \) is

\[
d_f \leq \begin{cases} 
6(n_{\text{true}} - 2), & \text{logarithmic case,} \\
36(n_{\text{true}} - \frac{7}{3}), & \text{non-logarithmic case.} 
\end{cases} \tag{11}
\]

Our algorithm uses this degree bound only as a starting point; additional restrictions are computed during the algorithm that may lower the degree.

3.1.3 Riemann-Hurwitz Type Formula

The differential operators \( L_B \) and \( L_{\text{inp}} \) are in \( \mathbb{C}(x)[\partial] \), i.e., they are defined on \( \mathbb{P}^1 \). The function field of \( \mathbb{P}^1 \) is \( \mathbb{C}(x) \). Denote \( D_{\mathbb{C}(x)} = \mathbb{C}(x)[\partial] \). So \( L_B, L_{\text{inp}} \subseteq D_{\mathbb{C}(x)} \).

In general, let \( X \) be any algebraic curve and \( X \) be its function field. The ring \( D_{\mathbb{C}(X)} := \mathbb{C}(X)[\partial] \) is the ring of differential operators on \( X \). Here \( t \in \mathbb{C}(X) \) with \( t' \neq 0 \). An element \( L \in D_{\mathbb{C}(X)} \) is a differential operator defined on the algebraic curve \( X \).

**Theorem 2.** Let \( X, Y \) be two algebraic curves with genus \( g_X, g_Y \); and function fields \( \mathbb{C}(X), \mathbb{C}(Y) \). Let \( f : X \to Y \) be a non-constant morphism with \( \deg(f) = d \). The morphism \( f \) corresponds to a homomorphism \( \mathbb{C}(Y) \to \mathbb{C}(X) \), which induces a homomorphism \( D_{\mathbb{C}(Y)} \to D_{\mathbb{C}(X)} \). If \( L_1 \subseteq D_{\mathbb{C}(Y)} \) with \( \text{ord}(L_1) = 2 \) and, \( L_2 \) is the corresponding element in \( D_{\mathbb{C}(X)} \), then

\[
2 - 2g_X + \sum_{p \in X} (\Delta(L_2,p) - 1) = d(2 - 2g_Y + \sum_{s \in Y} (\Delta(L_1,s) - 1)).
\]

**Proof.** Let \( S \subseteq Y \) be a finite set and \( T = f^{-1}(S) \) such that \( \text{Sing}(L_1) \subseteq S \), \( \text{Sing}(L_2) \subseteq T \), and all branching points in \( X \) are in \( T \). There are infinitely many points in \( X \setminus T \) and for each \( p \in X \setminus T \), we have \( \Delta(L_2,p) = 1 \) and \( e_p = 1 \). There
are infinitely many points in $Y \setminus S$ and for each $s \in Y \setminus S$, we have $\Delta(L_1, s) = 1$.

\[
\#T = \sum_{p \in T} 1 = \sum_{p \in T} e_p - \sum_{p \in T} (e_p - 1) = d \cdot \#S - \sum_{p \in X} (e_p - 1) = d \cdot \#S - (2g_x - 2 - d(2g_y - 2)).
\]

From (13) to (14) we used (6). Then,

\[
\sum_{p \in X} (\Delta(L_2, p) - 1) = \sum_{p \in T} (\Delta(L_2, p) - 1) = \sum_{p \in T} \Delta(L_2, p) - \sum_{p \in T} 1 = d \sum_{s \in S} \Delta(L_1, s) - \#T.
\]

Combine (14) and (17) to obtain

\[
\sum_{p \in X} (\Delta(L_2, p) - 1) = d \sum_{s \in S} \Delta(L_1, s) - d \cdot \#S + (2g_x - 2 - d(2g_y - 2)).
\]

Therefore,

\[
2 - 2g_x + \sum_{p \in X} (\Delta(L_2, p) - 1) = d(2 - 2g_y + \sum_{s \in Y} (\Delta(L_1, s) - 1)).
\]

We use differential operators $L_B, L_{\text{inp}} \in \mathbb{C}(x)[\partial]$. So $X = Y = \mathbb{P}^1$ and $g_x = g_y = g_{p_1} = 0$. Suppose that

\[
L_B \xrightarrow{\partial} C \xrightarrow{\partial} L_{\text{inp}}
\]

where $f : \mathbb{P}^1 \to \mathbb{P}^1$ and $L_B$ is a GHDO with exponent differences $[\alpha_0, \alpha_1, \alpha_\infty]$ at $\{0, 1, \infty\}$. Since the exp-product transformation does not affect exponent-differences, formula (18) gives us:

\[
2 + \sum_{p \in \mathbb{P}^1} (\Delta(L_{\text{inp}}, p) - 1) = \deg(f)(2 + \sum_{i \in \{0, 1, \infty\}} (\alpha_i - 1)).
\]

We will use formula (19) in section 3.2.

### 3.2 Candidate Exponent Differences

This section explains a method of computing exponent differences for candidate GHDOs.

**Remark 4.** Consider the operator $L_{\text{inp}}$ in example 3. It has 4 true singularities, so (11) gives us $d_f = 60$. For a candidate $L_B$ having exponent differences $[\alpha_0, \alpha_1, \alpha_\infty]$, we have

\[
\alpha_0, \alpha_1, \alpha_\infty \in \{a/b : a \in \mathbb{Q}, b \in \mathbb{N} \setminus \{1, 2, 3\}, 1 \leq b \leq d_f\}.
\]

Here $S_T$ is the set of exponent differences of $L_{\text{inp}}$ at its true singularities and $S_R$ is the set of exponent differences of $L_{\text{inp}}$ at its removable singularities. There are 176 elements in the set (20). This leaves too many candidates for $[\alpha_0, \alpha_1, \alpha_\infty]$. Algorithm find_expdiffs is designed to skip most combinations (formula (19) is particularly effective). In about 0.25 seconds find_expdiffs returns all different candidates: $[\frac{2}{7}, \frac{1}{3}, \frac{1}{7}, \frac{1}{2}], [\frac{1}{7}, \frac{1}{3}, \frac{1}{2}, 20]$. The first candidate gives a pullback function of degree 2 and the second candidate gives a pull-back function of degree 20.

**Algorithm: find_expdiffs**

**Input:**
- $c_{\text{inp}}$, a list of exponent differences of $L_{\text{inp}}$ at its true singularities.
- $c_{\text{rem}}$, a (possibly empty) list of exponent differences of $L_{\text{inp}}$ at its removable singularities.

**Output:**
- List of candidate exponent differences for candidate GHDOs.

Output is a list of all lists $e_B = [a_0, a_1, \alpha, \infty]$ of integers or rational numbers where $[a_0, a_1, \alpha, \infty]$ is a list of candidate exponent differences and $d$ is a candidate degree for $f$ such that:

- For every exponent difference $m$ in $c_{\text{inp}}$ there exists $e \in \{1, 2, \ldots, d\}$ such that $m = e\alpha_i$ for some $i \in \{0, 1, \infty\}$.
- The multiplicities $e$ are consistent with (7), and $\sum e = 1$.

The logarithmic singularities of $L_{\text{inp}}$ come from the point 0. Non-integer exponent differences of $L_{\text{inp}}$ must be multiples of $\alpha_1$ or $\alpha_\infty$. Let $S_N$ be the set of non-logarithmic exponent differences of $L_{\text{inp}}$ and $S_R$ be the set of exponent differences of $L_{\text{inp}}$ at its removable singularities. Consider the set

\[
\Gamma_1 = \{ \Gamma_A = \{ \frac{\max(S_N)}{b} : b = 1, \ldots, d_f \} \text{ if } S_N \neq \emptyset, \quad \\
\{ \Gamma_B = \{ \frac{a}{b} : a \in S_R \cup \{1\}, b = 1, \ldots, d_f \} \text{ otherwise.} \}
\]

Let $k = 1$. So we have $a_0 \in \mathbb{Z}$. We need to find rational numbers $a_1$ and $a_\infty$.

The logarithmic singularities of $L_{\text{inp}}$ must be multiples of $\alpha_1$ or $\alpha_\infty$. Let $S_N$ be the set of non-logarithmic exponent differences of $L_{\text{inp}}$ and $S_R$ be the set of exponent differences of $L_{\text{inp}}$ at its removable singularities. Consider the set

\[
\Gamma_1 = \{ \Gamma_A = \{ \frac{\max(S_N)}{b} : b = 1, \ldots, d_f \} \text{ if } S_N \neq \emptyset, \quad \\
\{ \Gamma_B = \{ \frac{a}{b} : a \in S_R \cup \{1\}, b = 1, \ldots, d_f \} \text{ otherwise.} \}
\]

Let $k = 1$. So we have $a_0 \in \mathbb{Z}$. We need to find rational numbers $a_1$ and $a_\infty$.

Now take all pairs $(\alpha_\infty, d)$ 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 $\alpha_i$’s we pre-compute a list of integers $N_v$ by dividing all exponent-differences of $L_{\text{inp}}$ 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 $N_x$. Each time a non-zero value $v$ is taken for one of the $\alpha_i$, 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 $\alpha_{\infty}$ 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.

Once we have the list of candidate exponent differences, then each of the elements of this list gives a candidate GHDO. If $L_{inp}$ has a $2F_1$-type solution in form (1), then it is among the candidate GHDOs that we computed, via a change of variables and exp-product transformations. This answers question Q2.

### 3.3 Quotient Method

In this section, we explain a method to recover the pullback function $f$, which is the most crucial part of our algorithm. 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 3.4. Before starting this section, note that we can always compute the formal solutions of a given differential equation $L_{inp}(y) = 0$ up to a fine precision.

#### 3.3.1 Non-Logarithmic Case

Let the second order differential equation $L_{inp}(y) = 0$ be given. Let $L_B$ be a GHDO such that $L_B \overset{\gamma}{\rightarrow}_E L_{inp}$. Let $f : \mathbb{P}_x \rightarrow \mathbb{P}_x$ and $L_1 \overset{\gamma}{\rightarrow}_C L_2$. 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 $f(y) = s$.

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 $f = c_0 x^{\gamma_0}(1 + \ldots)$ where $c_0 \in \mathbb{C}$, $\gamma_0(f)$ is the multiplicity of 0, and the dots refer to an element in $xC[[x]]$.

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 $L_B$.

\[
y_1(x) \overset{\gamma}{\rightarrow}_C y_1(f) \overset{\gamma}{\rightarrow}_E Y_1(x) = \exp(\int rdx)y_1(f),
\]
\[
y_2(x) \overset{\gamma}{\rightarrow}_C y_2(f) \overset{\gamma}{\rightarrow}_E Y_2(x) = \exp(\int rdx)y_2(f),
\]

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

Let $q = \frac{y_2}{y_1}$ 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 $y_2 \mapsto \alpha y_2 + \beta y_2$.

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
\[
f(x) = q^{-1}(c \cdot Q(x)).
\]

If we know the value of this constant $c$, then we can 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 $\ell$. Then, for each $c \in \{1, \ldots, \ell - 1\}$ we try to compute $f$ modulo $\ell$. If this succeeds, then we lift $f$ modulo a power of $\ell$, and try reconstruction. Details of lifting is explained in section 3.4.

Remark 5. Here we should compute the formal solutions up to a precision $a \geq (a_f + 1)(d_f + 1) + 3$. This precision is enough to recover the correct pullback function with a few extra terms for checking. This answers Q1.

Algorithm: case1 (non-logarithmic case)

**Input:**
- $L_{inp}$, a second order differential operator with non-logarithmic solutions,
- $L_B$, a candidate GHDO,
- $d_f$, degree bound for $f$.

**Output:**
- The rational pullback function $f$, or 0 (in this case there is no rational pullback function).

1. Compute expansions of the formal solutions $y_1, y_2$ of $L_B$ and $Y_1, Y_2$ of $L_{inp}$ up to precision $a \geq 2d_f + 5$.
   Select a prime $\ell$ for which these expansions can be reduced mod $\ell$.
2. $q \leftarrow \frac{y_2}{y_1}$, $Q \leftarrow \frac{Y_2}{Y_1}$, then compute $q^{-1}$.
3. Search for $c_0$ such that $c \equiv c_0 \mod \ell$ by looping over $c_0 = 1, \ldots, \ell - 1$. If there is no such $c_0$, then return 0.
4. Compute $f_1 = q^{-1}(c_0 \cdot Q) \in \mathbb{Z}[x]/(\ell, x^a)$.
5. Lift $f_1$ to $f_1 \in \mathbb{Z}[x]/(\ell, x^a)$ for a suitable $\ell \in \mathbb{N}$, and then reconstruct the rational pullback function $f$ from $f_1$ (we still need to address remark 2).
6. Return $f$.

#### 3.3.2 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

\[
c_1 = 1 \quad \text{and constant term of } h = 0.
\]

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

\[
g = \exp\left(\frac{y_2}{y_1}\right) = x \cdot (1 + \ldots)
\]

where the dots refer to an element of $xC[[x]]$.

\footnote{For details see the section 3.4}
Remark 6. 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 \(^{(23)}\) 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 have not implemented this case yet).

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

\[
G = \exp \left( \frac{Y_2}{Y_1} \right) = x \cdot (1 + \ldots). \tag{24}
\]

Write \( f \in \mathbb{C}(x) \) as \( c_0 x^{v_0(f)} \cdot (1 + \ldots) \). Then \( g(f) = c \cdot x^{v_0(f)}(1 + \ldots) \). Note that \( g, G \) are not intrinsically unique, the choices we made in \(^{(22)}\) implies that

\[
g(f) = c_1 \cdot G^{c_2} \tag{25}
\]

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 \( v_0(f) = 1, 2, \ldots, df \). That leaves one unknown constant \( c \). We address this problem as before, choose a good prime number \( \ell \), try \( c = 1, 2, \ldots, \ell - 1 \). Then calculate an expansion for \( f \) with the formula

\[
f = g^{-1} \left( c \cdot x^{v_0(f)} \right). \tag{26}
\]

Then we lift \( f \) modulo a power of \( \ell \), and try reconstruction. The discussion in this section answers Q4.

Algorithm: case2 (logarithmic case)

---

Input:
- \( L_{inp} \), a second order differential operator with at least one logarithmic solutioan.
- \( \ell \), a candidate GHDO.
- \( d_f \), degree bound for \( f \).

Output:
- The rational pullback function \( f \), or 0 (in this case there is no rational pullback function).

1. Compute the exponents of \( L_{inp} \) and \( L_B \).
2. Compute expansions of the formal solutions \( y_1, y_2 \) of \( L_B \) and \( Y_1, Y_2 \) of \( L \) up to precision \( a \geq 2df + 5 \). Select a prime \( \ell \) for which these expansions can be reduced mod \( \ell \).
3. \( g \leftarrow \frac{y_2}{y_1}, Q \leftarrow \frac{Y_2}{Y_1} \), and compute \( g \) and \( G \) from \(^{(23)}\) and \(^{(24)}\) respectively. Then compute \( g^{-1} \).
4. Select (compute if \( \Delta(L_{inp},0) \neq 0 \), loop otherwise) \( v_0(f) \) and search for \( c_0 \) such that \( c \equiv c_0 \mod p \) by looping over \( 1, \ldots, \ell - 1 \). If there is no such \( c_0 \) (which means there is no rational pullback function for this candidate \( L_B \)), then return 0.
5. Compute \( f_1 = g^{-1} \left( c_0 \cdot G^{v_0(f)} \right) \in \mathbb{Z}[x]/(\ell, x^a) \).
6. Lift \( f_1 \) to \( f_l \in \mathbb{Z}[x]/(\ell^l, x^a) \) for a suitable \( l \in \mathbb{N} \), and reconstruct the rational pullback function \( f \) from \( f_l \) (we still need to address remark 2).
7. Return \( f \).

Remark 7. Algebraic Pullback Functions

Let \( L_{inp} \) have a \( 2F_1 \)-type solution in the form \(^{(1)}\) where \( f \) is an algebraic function. We do not have a degree bound for this case, nor the analogue of the algorithm from section \(^{3.2}\). Therefore, for this case, the current version of our implementation needs extra inputs: a candidate GHDO, a degree bound for \( f \), and an algebraic degree bound for \( f \). Then we can find the algebraic pullback function via the quotient method. The only difference is the lifting algorithm which is explained in section \(^{3.4}\). An algebraic degree bound is needed for lifting. This remark together with section \(^{3.4}\) answer question Q5.

3.4 Lifting: Recovering the Pullback Function

We introduce two lifting algorithms, one for rational functions, one for algebraic functions. We explain lifting by using the formula \(^{(21)}\) for the pullback function, which occurs in the non-logarithmic case. The algorithm for the formula \(^{(26)}\) in the logarithmic case is similar. The discussion in this section answers Q3.

3.4.1 Lifting for a Rational Pullback Function

By using the formula \(^{(21)}\), 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 \( \ell \), then we can recover the pullback function \( f \). This can be done via Hensel lifting techniques.

Let \( \ell \) be a good prime number and consider

\[
h : Q \to \mathbb{Q}[x]/(x^a)
\]

By looping on \( c_0 = 0, 1, \ldots, 1 - 1 \) and trying rational function reconstruction for \( h(c_0) \) mod \( (\ell, x^a) \), we can compute the image of \( f \) in \( F_\ell/(x^a) \). 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 \( (\ell, x^a) \). This \( c_0 \) is the one satisfying \( c \equiv c_0 \mod \ell \).

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

\[
h(c) = h(c_0 + \ell c_1) \equiv h(c_0) + \ell h'(c_0) \mod (\ell^2, x^a). \tag{27}
\]

Substitute \( c_1 = 0, c_1 = 1 \), respectively, in \(^{(27)}\) and compute

\[
h(c_0) \mod (\ell^2, x^a), \tag{28}
\]

\[
h(c_0 + \ell) \equiv h(c_0) + \ell h'(c_0) \mod (\ell^2, x^a). \tag{29}
\]

Subtracting \(^{(28)}\) from \(^{(29)}\) gives

\[
\ell h'(c_0) \equiv [h(c_0 + \ell) - h(c_0)] \mod (\ell^2, x^a). \tag{30}
\]

Let

\[
S = \left\{ h(c_0) + \ell h'(c_0) : c_1 = 0, \ldots, \ell - 1 \right\}. \tag{30}
\]

Let \( f = \frac{A}{B} \) 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 \( f \equiv \frac{A_0}{B_0} \mod (\ell, x^a) \). It follows that \( \frac{A}{B} \equiv \frac{A_0}{B_0} \equiv E_{c_1} \mod (\ell, x^a) \) for an element \( E_{c_1} \in S \) defined in \(^{(30)}\). From this equation we have

\[
A \equiv B E_{c_1} \mod (\ell, x^a). \tag{31}
\]

Now let

\[
f = \frac{A}{B} \equiv \frac{A_0 + \ell A_1}{B_0 + \ell B_1} \mod (\ell^2, x^a) \tag{32}
\]
where $A_1 = a_0 + a_1 x + \ldots + a_{\deg(A_0)} x^{\deg(A_0)}$ and $B_1 = b_0 x + \ldots + b_{\deg(B_0)} x^{\deg(B_0)}$ are unknown polynomials. Here we are fixing the constant term of $B$. If we can find the unknowns $\{a_i, b_i\}$, then find $f \mod (\ell^2, x^a)$. Then, from \((33)\), we have

\[(A_0 + \ell A_1) \equiv (B_0 + \ell B_1) [b(c_0) + \ell c_1 h'(c_0)] \mod (\ell^2, x^a). \tag{33}\]

Now, solve the linear equation \((33)\) for unknowns $\{a_i, b_i, c_i\}$ in $F_\ell$, and from \((33)\) find $f \mod (\ell^2, x^a)$ and $c \equiv c_0 + \ell c_1 \mod \ell^2$. Then try rational number reconstruction. If it succeeds, then check if this rational function is the one that we are looking for or not (apply change of variables transformation and try to find the parameter of the exp-product transformation). If it is not, then use the same algorithm to lift $f \mod (\ell^2, x^a)$ to mod $(\ell^3, x^a)$ (or $(\ell^4, x^a)$ if an implementation for solving linear equations mod $\ell^3$ is available). After a (finite) number of steps, we can recover the rational pullback function $f$.

### 3.4.2 Lifting for an Algebraic Pullback Function

We can also recover algebraic pullback functions with a very similar method as explained in the previous section. However, in the algebraic pullback case we need to know an algebraic degree bound for $f$. The idea here is to recover the minimal polynomial of the algebraic pullback function $f$.

Let $d_j$ be a degree bound, and $a_j$ be an algebraic degree bound for $f$. Consider the below polynomial in $y$,

$$
\sum_{j=1}^{a_j} A_j y^j \mod (\ell, x^a),
$$

with unknown polynomials $A_j = \sum_{i_j=0}^{d_j} a_{i,j} x^i$, $(j = 1, \ldots, a_j)$.

First we need to find the value of $c_0$ such that $c_0 \equiv c \mod \ell$. Similarly, by looping on $c_0 = 1, \ldots, \ell - 1$, we can compute the corresponding $f \equiv f_1 \in F_\ell/(x^a)$. For this $f_1$, the polynomial \((34)\) will be congruent to 0 mod $(\ell, x^a)$ if we plug $f_1$ in $y$. So, solve the equation

$$
\sum_{j=1}^{a_j} A_j f_1^j \equiv 0 \mod (\ell, x^a)
$$

in $F_\ell$ and find the unknown polynomials $A_j$. After finding $c \equiv c_0 \mod \ell$ and polynomials $A_j$, then let $c \equiv c_0 + \ell c_1 \mod \ell^2$. Then $f_1$ also satisfies the polynomial

$$
\sum_{j=1}^{a_j} (A_j + \ell \tilde{A}_j) y^j \mod (\ell^2, x^a).
$$

in $F_\ell$ for unknown polynomials $\tilde{A}_j$. Similarly, find the $c_1$ and unknown polynomials $\tilde{A}_j = \sum_{i_j=0}^{d_j} \tilde{a}_{i,j} x^i$, $(j = 1, \ldots, a_j)$. After a finite number of lifting steps, and rational reconstruction, we will have the minimal polynomial of an algebraic pullback function $f$.

### 3.5 Recovering the Parameter of Exp-product

After finding $f$, we can compute the differential operator $M$, such that $L_B \frac{\partial}{\partial C} M \approx_B L_{\text{inp}}$. Then we can compare the second highest terms of $M$ and $L_{\text{inp}}$ to find the parameter $r$ of the exp-product transformation: If $M = \partial^2 + B_1 \partial + B_0$ and $L_{\text{inp}} = \partial^2 + A_1 \partial + A_0$, then $r = \frac{B_1 - A_1}{2}$.

### 4. FUTURE WORK

We plan to work on finding a method to compute a degree bound and an algebraic degree bound for an algebraic pullback function as well as finding a method to compute candidate GHDOs for algebraic cases. We also plan to use \cite{Kauers2011} to find a method to reduce equations involving gauge transformation to equations involving only change of variables and exp-product transformations.

### 5. REFERENCES

\begin{thebibliography}{9}
\end{thebibliography}